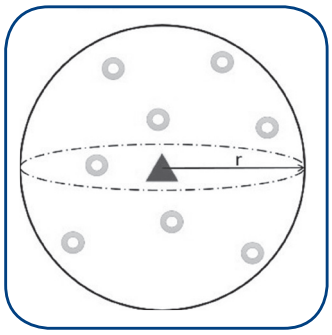
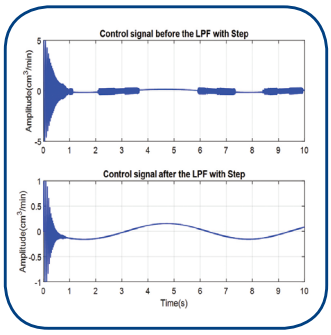
Volume 13, Number 6, 2022 ISSN 1847-6996

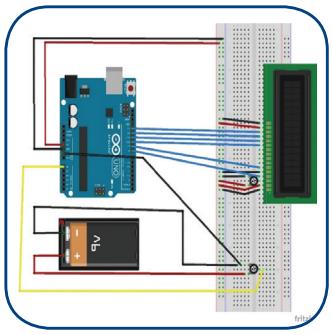
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY **OSIJEK**

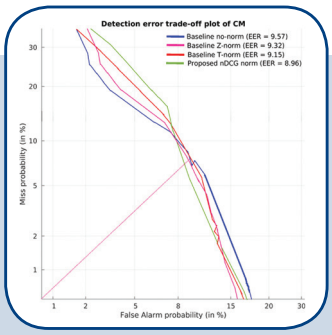
**IJECES** International Journal of Electrical and Computer Engineering Systems

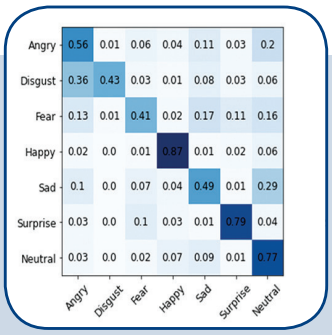
# International Journal of Electrical and Computer Engineering Systems

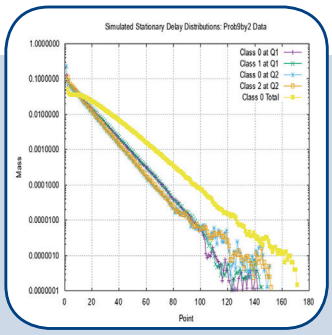




ISSN: 1847-6996

# INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Croatia

Osijek, Croatia | Volume 13, Number 6, 2022 | Pages 417 - 491

The International Journal of Electrical and Computer Engineering Systems is published with the financial support of the Ministry of Science and Education of the Republic of Croatia

**CONTACT**

**International Journal of Electrical and Computer Engineering Systems (IJECES)**

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b, 31000 Osijek, Croatia
Phone: +38531224600, Fax: +38531224605
e-mail: ijeces@ferit.hr

**Subscription Information**

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.
Giro account: 2390001 - 1100016777,
Croatian Postal Bank



**Journal is referred in:**

- Scopus
- Web of Science Core Collection (Emerging Sources Citation Index - ESCI)
- Google Scholar
- CiteFactor
- Genamics
- Hrčak
- Ulrichweb
- Reaxys
- Embase
- Engineering Village

**Bibliographic Information**

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003
Published: quarterly
Circulation: 300

**IJECES online**

https://ijeces.ferit.hr

# TABLE OF CONTENTS

# Delay Distributions in Discrete Time Multiclass Tandem Communication Network Models

Original Scientific Paper

**Nandyala D. Gangadhar**
Department of Computer Science and Engineering, Faculty of Engineering and Technology,
M. S. Ramaiah University of Applied Sciences
470-P Phase IV, Peenya, Bengaluru 560058, India
n_d_gangadhar@ieee.org; gangadhar.cs.et@msruas.ac.in

**Govind R. Kadambi**
Department of Electronics and Communication Engineering, Faculty of Engineering and Technology,
M. S. Ramaiah University of Applied Sciences
470-P Phase IV, Peenya, Bengaluru 560058, India
pvc.research@msruas.ac.in

***Abstract*** *– An exact computational algorithm for the solution of a discrete time multiclass tandem network with a primary class and cross-traffic at each queue is developed. A sequence of truncated Lindley recursions is defined at each queue relating the delays experienced by the first packet from consecutive batches of a class at that queue. Using this sequence of recursions, a convolve-and-sweep algorithm is developed to compute the stationary distributions of the delay and inter-departure processes of each class at a queue, delays experienced by a typical packet from the primary class along its path as well as the mean end-to-end delay of such a packet. The proposed approach is designed to handle the non-renewal arrival processes arising in the network. The algorithmic solution is implemented as an abstract class which permits its easy adaptation to analyze different network configurations and sizes. The delays of a packet at different queues are shown to be associated random variables from which it follows that the variance of total delay is lower bounded by the sum of variances of delays at the queues along the path. The developed algorithm and the proposed lower bound on the variance of total delay are validated against simulation for a tandem network of two queues with three classes under different batch size distributions.*

***Keywords****: Delay, Tandem, Communication, Network, Discrete Time, Queueing, Algorithm, Lindley Recursion, End-to-end*

## 1. INTRODUCTION

Communication networks carry packet traffic through connections between multiple source-destination pairs. Traffic from each connection passes through a network path consisting of a sequence of intermediate nodes and faces contention at each of these nodes from other connections. Delays experienced by the connection along the path are random, and their characterization and estimation are crucial measures of network performance. For this purpose, the connection can be modelled as a tandem network of queues [1-5].

In discrete time queues and their network models, the time axis is divided into equal intervals, termed "slots". In each slot, a batch of packets from each of the sources is generated and enters the queue or network and one or more packets leave the queue or network [3-4].

In a network, the joint distribution of queue lengths and delays as well as the end-to-end delay are useful measures of network performance but have proved to be hard problems to solve [2]. Analogous to the analysis of their continuous time counterparts, decomposition of the stationary joint distribution of the queue lengths at different queues in a discrete time network into a product of marginal distributions is shown for a large class of networks [6]. To employ this for obtaining the delays, one needs to establish arrival theorems which are known for only special models of networks [7]. However, product-form decomposition is employed as a heuristic approximation in general networks. This paper addresses the problem of computing the delays in a discrete tandem network with batch arrival processes and cross-traffic. An exact algorithm for computing the distributions of delays at each queue as well as the mean end-to-delay of a typical packet is developed, along with a lower bound for the variance of the end-to-end delay.

Different models of tandem queueing networks with and without arrivals and with and without departures at the intermediate nodes are studied in the literature. In [8], a tandem network of two queues in continuous

time with Poisson arrivals only at the first queue and identical and independent service times is studied. Joint delays in a continuous time tandem network without intermediate arrivals are analyzed in [9]. In [10], an approximate analysis of the queue lengths and busy periods at each node in a discrete time tandem network with arrivals and departures at intermediate nodes is carried out. The departures are modelled as random packet drops after service. The authors of [11] consider a model identical to the model analyzed in the current paper and propose recursive algorithms for the computation of delay distributions without presenting any results. In the current paper, a sequence of truncated Lindley processes is introduced for expressing the delays in the queues of the tandem network, and a convolve-and-sweep algorithm is devised for computing the delay distributions. This approach has the advantage that the algorithm does not require renewal arrival processes and hence can be applied to all the queues in the tandem network.

Discrete time tandem network with arrivals and departures at intermediate nodes under Furthest-to-Go service discipline is analyzed in [12] and expressions for queue length distributions are obtained. The components of end-to-end delay in a Software Defined Network (SDN) are modelled and experimentally estimated in [5].

Apart from computational approaches, simulation and bounds are also employed in the analysis of tandem networks. Simulation analysis of the performance of high-speed networks is carried out in [13-14]. In [15], the output process of a GI/GI/1 queue is approximated by a renewal process and this is applied to a tandem network of queues without interfering traffic. This approach is applied in [16] to analyze general discrete time networks. Worst case bounds on the end-to-end delay under active queue management scheduling algorithms are derived in [17]. A general network calculus approach to the end-to-end analysis of queueing networks when the inputs are modelled as deterministic or stochastic affine envelop processes is developed in [18-19] and the same has been applied to SDN in [20-21].

Computation of the end-to-end delay in a network is feasible for product-form networks wherein the stationary joint distribution of delays at various queues factors into the product of the marginal distributions. This follows if a stronger assumption that the delays in the queues are independent random variables is made. The product-form decomposition and independence assumption are commonly employed as a heuristic for end-to-end analysis. Lower bounds on the moments of total delay are established by showing that the delays in the individual queues are associated *random variables* [22]. This property is proved in [23] for a tandem network wherein the service times at all but the last server are all a constant. In [23] it is employed to derive an upper bound on the mean total delay. In the current paper, it is established for the tandem network model considered and is employed to obtain a lower bound on the variance of the total delay. The exact mean total delay is computed as the sum of computed marginal distributions.

The rest of the paper is organized as follows: Section 2 describes the model, notation employed and analysis of a first queue in the tandem network as a single discrete queue highlighting the need for careful analysis of the other queues. Section 3 begins by defining a sequence of truncated Lindley processes to recursively relate the delays experienced by the first packet from two consecutive batches of Class 0 at each queue in the tandem network. A computational algorithm is derived from these recursions by adapting the convolve-and-sweep algorithm to the truncated Lindley processes. The delays of a typical packet in the queues are shown to be associated random variables providing a lower bound on the variance of the end-to-end delay. This section also describes the details of implementation and benchmark simulations. The results of computational and their validation via simulation are presented in Section 4. Section 5 concludes the paper with some directions for future work. Appendix 1 presents a proof of Theorem 1 stated in Section 3.

## 2. MODEL, NOTATION AND PRELIMINARIES

This section presents the tandem discrete time queueing network model, notation employed and approaches to the analysis of a single discrete queue. The model consists of discrete time queues in series, each with arrivals and departures. The tandem network models the path of a packet stream of interest in a general network.

### 2.1 TANDEM NETWORK MODEL

The discrete time model analyzed in this paper consists of a tandem of one or more discrete time queues and multiple streams of traffic. Each stream can traverse multiple queues. Figure 1 depicts an instance of such a tandem network with two queues Q1 and Q2, and three streams of traffic that are designated as belonging to Classes 0, 1 and 2. Class 0 traffic enters the first queue Q1 and passes through both the queues before leaving the system. Traffic from Class $i$, $i$=1,2 enters the ith queue and leaves the system after service at that queue. Classes 1 and 2 can be considered as "cross-traffic" at queues Q1 and Q2, respectively.

Packets from all the classes are assumed to be of the same size and the servers at all the queues have identical service rates. Thus, all the jobs need the same amount of server time at the queues. This constant amount of time is taken as a unit of time and time is discretized into slots of this duration. The system works in discrete time: all the arrivals to a queue arrive at the beginning of the slot and the first job (after ordering for service) will be served at the beginning of the earliest slot that the server is free. The arrivals from each Class

occur as a single batch with a general batch size distribution and it is assumed that the batches are ordered for service according to a stationary policy. One can in principle remove this assumption at the cost of a cumbersome analysis.

The interval $[k,k+1)$ is termed the $k^{\text{th}}$ slot. The traffic from each class is assumed to arrive as batches per slot with the batch size following an iid distribution

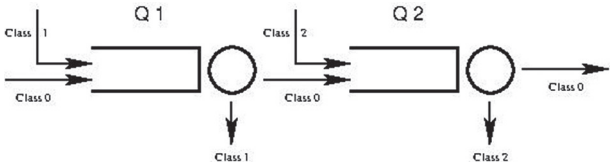



**Fig. 1.** A Tandem Queueing Network with Cross Traffic

### 2. 2 NOTATION

The following notation is used throughout the paper:

Qi : The $i^{\text{th}}$ Queue in the tandem

$X_{\text{k}}^{(\text{i})}$ : Number of Class $i$ arrivals in the $k^{\text{th}}$ slot

$A_{\text{j,k}}^{(\text{i})}$ : Arrival slot of the first job of Class $i$ at Qj

$\Delta_{\text{j,k}}^{(0)}$ : Interval between the arrival of the $k^{\text{th}}$ and $(k+1)^{\text{st}}$ batch of Class 0 at Qj

$W_{\text{i,k}}$ : Workload of Qi at the beginning of $k^{\text{th}}$ slot

$D_{\text{j,k}}^{(0)}$ : Delay of the first job of Class the $k^{\text{th}}$ batch of Class 0 at Qj

### 2.3. PRELIMINARIES

The workload $W_{1,\text{k}}$, evolves as follows:

$$W_{1,k+1} = \left(W_{1,k} + X_k^{(0)} + X_k^{(1)} - 1\right)^+ \tag{1}$$

where, $x^+=\max\{x,0\}$. Indeed, the total arrivals in the $k_{\text{th}}$ slot are $X_k^{(0)}+X_k^{(1)}$ and one packet gets served if there are a non-zero number of packets, leaving $W_{1,k+1}$ packets at the beginning of the $(k+1)^{\text{st}}$ slot. The Markov Chain $\{W_{1,\text{k}}, k\geq 0\}$ has a unique stationary distribution $\pi$ under the stability condition $\text{E}(X_k^{(0)}+X_k^{(1)})<1$ and can be computed using Ramaswamy Algorithm [25]. By the arrival theorem for geometric arrivals, $D_{\text{j,k}}^{(0)}=W_{1,k}\sim\pi$. The delay distribution of a typical Class 0 packet in the $k^{\text{th}}$ batch can then be obtained as

$$Y_{j,k}^{(0)} = D_{j,k}^{(0)} + \theta \tag{2}$$

where $\theta$ is the rank of the typical packet in its batch.

This analysis does not extend to the rest of the queues in the tandem since the arrival stream from Class 0 packets is non-renewal. Hence, a new approach for developing a computational solution for the tandem networks is needed, as developed in Section 3.

## 3. COMPUTATIONAL SOLUTION AND ALGORITHM DEVELOPMENT

Eq. (1) is a Lindley Recursion [5, 24] of the form

$$Z_{n+1} = (Z_n + X_n - 1)^+ \tag{3}$$

and the process $\{Z_{\text{n}}\}$ is called a Lindley process. The convolve-and-sweep algorithm for computing the distribution of the Lindley process is given by:

$$P(Z_{n+1} = m) = \begin{cases} P(Z_n = 0)P(X_b \leq 1) \\ \quad + P(Z_n = m)P(X_b = 0), \qquad m = 0 \\ \sum_{l=0}^{m+1} P(Z_n = l)P(X_b = m+1-l), \\ \qquad m > 0 \end{cases} \tag{4}$$

where it is assumed that $Z_0$ is independent of $\{X_k\}$.

In this paper, Lindley Recursion and the convolve-and-sweep algorithm will be used to derive a sequence of *truncated* Lindley processes defined by generalizing Eq. (3) (Cf. Eqs. (5)-(7) and (9)-(12)) and build a computational algorithm for obtaining the distributions of delays of each class of packets using the convolve-and-sweep algorithm expressed by Eq. (4).

The advantage of computing the stationary distributions of delays using the convolve-and-sweep algorithm is that no independence assumptions on the arrival processes are required. This is essential for computing the distributions of delays at Q2 and other downstream nodes where the arrival process from Class 0 is not a renewal process. The stationary distributions of delays are computed as limits of the transient distributions.

### 3.1 TRUNCATED LINDLEY PROCESSES

From the Lindley process $\{W_{1,k}\}$ of Eq. (1), a sequence of truncated Lindley processes is defined:

$$W_{1,k+1}(0) := D_{1,k}^{(0)} + X_k^{(0)} + X_k^{(1)} - 1 \tag{5}$$

$$W_{1,k+1}(l) := \left(W_{1,k+1}(l-1) + X_{A_{1,k}^{(0)}}^{(1)} - 1\right)^+ \tag{6}$$

for $l$=1, 2,..., $\Delta_{1,k}^{(0)}$-1, using the notation of Sec. 2. Then, the delay of the first packet in the $(k+1)^{\text{st}}$ batch of Class 0, $D_{1,k+1}^{(0)}$, is given by:

$$D_{1,k+1}^{(0)} = W_{1,k+1}\left(\Delta_{1,k}^{(0)} - 1\right) \tag{7}$$

Eqs. (5)-(7) can be used to compute the transient distributions of the delays $\{D_{1,k}^{(0)}; k=0,1,\ldots\}$ by extending the convolve-and-sweep algorithm for the standard Lindley Recursion to the sequence of truncated Lindley processes, $\{W_{1,k}(l)\}$. Since $\Delta_{1,k}^{(0)}$ is random, the computation is carried out by conditioning on $\Delta_{1,k}^{(0)}=l$, $l$=1,2,…, using the distribution of $\Delta_{1,k}^{(0)}$.

### 3.2 INTER-DEPARTURE DISTRIBUTION FROM Q1

For computing the delay distributions at Q2, the distribution of the inter-arrival time between the first jobs of Class 0 from two successive batches of arrivals at Q1 that have at least one Class 0 job is needed. For this, the following relation is used:

$$\begin{aligned}\Delta_{2,k}^{(0)} &= A_{1,k+1}^{(0)} + D_{1,k+1}^{(0)} - \left(A_{1,k}^{(0)} + D_{1,k}^{(0)}\right) \\ &= D_{1,k+1}^{(0)} - D_{1,k}^{(0)} + \Delta_{1,k}^{(0)}\end{aligned} \tag{8}$$

Eq. (8) follows from the fact that the departure times of the first packet from $k^{\text{th}}$ and $(k+1)^{\text{st}}$ Class 0 batches are $A_{1,k}{}^{(0)} + D_{1,k}{}^{(0)}$ and $A_{1,k+1}{}^{(0)} + D_{1,k+1}{}^{(0)}$, respectively, and $\Delta_{j,k}{}^{(0)} = A_{j,k+1}{}^{(0)} - A_{j,k}{}^{(0)}$, $j$=0,1, by definition.

### 3.3 DELAY AND INTER-DEPARTURE DISTRIBUTIONS AT Q2

As observed above, the computation of delay distributions at Q1 using Eqs. (5)-(7) can be applied even when the arrival processes are non-renewal as is the case for the Class 0 process at Q2. The corresponding truncated Lindley processes at Q2 are now derived.

To simplify the notation, it is assumed that the Class 0 jobs are arranged to be at the beginning of all the arrivals in a slot at Q2. In Q2, the evolution is given by slight modifications of Eqs. (5)-(7) since now Class 0 packets from its $k^{\text{th}}$ batch form a train of $X_{A_{1,k}^{(0)}}^{(1)}$ number arrivals at Q2 starting from slot $A_{2,k}{}^{(0)}$. The workload process evolution is characterized by the Eqs. (9)-(11) below (Cf. Eqs. (5)-(7)):

$$W_{2,k}(0) = D_{2,k}^{(0)} + Z_k^1 + X_{A_{2,k}^{(0)}+X_{A_{1,k}^{(0)}}^{(0)}}^{(1)} - 1 \tag{9}$$

where

$$Z_k^1 = X_{A_{2,k}^{(0)}}^{(1)} + X_{A_{2,k}^{(0)}+1}^{(1)} + \cdots + X_{A_{2,k}^{(0)}+X_{A_{1,k}^{(0)}}^{(0)}-1}^{(1)} \tag{10}$$

and

$$W_{2,k+1}(l) = \left( W_{2,k}(l-1) + X_{A_{2,k}^{(0)}+X_{A_{1,k}^{(0)}}^{(0)}+l}^{(1)} - 1 \right)^+ \tag{11}$$

for $l$=1,2,…,$\Delta_{2,k}{}^{(0)}$-1. Eq. (10) follows from the fact that, in slots $A_{2,k}^{(0)}, A_{2,k}^{(0)}+1, \ldots, A_{2,k}^{(0)}+X_{A_{1,k}^{(0)}}^{(0)}-1$, there is a single Class 0 packet arriving and being served at Q2. Then, as in Eq. (7), the delay is given by

$$D_{2,k+1}^{(0)} = W_{2,k}\left(\Delta_{2,k}^{(0)} - 1\right) \tag{12}$$

It can be observed that Eq. (10) has a variable number, $X_{A_{1,k}^{(0)}}^{(1)}$ of terms. In addition, $\Delta_{2,k}{}^{(0)}$ is positively correlated with this number, $X_{A_{1,k}^{(0)}}^{(1)}$. Hence, the computation of $D_{2,k+1}{}^{(0)}$ needs to condition on $X_{A_{1,k}^{(0)}}^{(1)}$. With this modification, the computational procedure developed for computing the stationary distribution of delays $D_{2,k+1}{}^{(0)}$ as well as that of the inter–departure times $\Delta_{3,k}{}^{(0)}$, at Q1, carries over to Q2.

### 3.4 SOLUTION OF TANDEM NETWORK

The computational solutions developed for delays and inter-departure distributions at Q2 can be employed at each of the downstream queues in the tandem network with changes in the input processes. The algorithm for computing the tandem network processes is listed as Algorithm 1.

**Algorithm 1**

**Input**: Batch size distributions for Classes with external arrivals, $X_0{}^{(i)}$ and initial queue lengths $W_0{}^{(i)}$

**Output**: Stationary distributions of delays $D_{j,\infty}{}^{(i)}$ and inter-departure times $\Delta_{j,\infty}{}^{(i)}$ at Qj

**Initialization**:

a) Set precision value EPSILON (typical: 1e-06)

b) Set the distributions $\{D_{j,0}{}^{(i)}\}$ and $\{\Delta_1{}^{(i)}\}$, for all Classes $i$ to degenerate distribution $\delta_0$

c) Initialize the distribution of batch inter-arrival time $\Delta_{1,\infty}{}^{(0)}$ to Geom($p$=$P(X_1{}^{(0)}>0)$)

**Iteration**:

For each queue Q$j$, $j$ = 1,2,…, in the tandem do:

For $k$=1,2,…until convergence do:

Compute the distribution of $D_{j,k+1}{}^{(0)}$ using

Eqs. (5)-(7) for Q1 or Eq. (9)-(12) for Q2, Q3, …

End

Compute the distribution of $\Delta_{j,\infty}{}^{(0)}$ using Eq. (8)

End

The iteration for the computation of the stationary distributions$\{D_{j,\infty}{}^{(0)}\}$ in Algorithm 1 is repeated until a) The CDF of the computed distribution is greater than 1.0-EPSILON, and b) The $l_\infty$ distance between the computed successive distributions is less than EPSILON.

### 3.5 ASSOCIATION OF DELAYS IN THE TANDEM

The *total delay* in the tandem network is an important performance measure in communication networks. To compute it, the distribution of the *sum* of delays at individual queues in the tandem is needed. Algorithm 1 does not provide this since only the *marginal* distributions of delays are computed.

On the other hand, most network monitoring and control applications employ the moments of the total delay. The mean of the total delay is the sum of mean delays in the individual queues; the latter quantities are readily computed from the obtained marginal distributions of the delays. Regarding the variance of the total delay, the following result is established to provide a computable lower bound:

**Theorem 1.** The random variables corresponding to the delays experienced by an arbitrary Class 0 job at Q1 and Q2 are associated random variables [24]; i.e., using Eq. (2), $\mathrm{Cov}(f(Y_{1,k}^{(0)}), g(Y_{2,k}^{(0)}) \geq 0$ for all increasing functions $f$ and $g$.

The theorem is proved in the Appendix.

In particular, the theorem implies $\mathrm{Cov}(Y_{1,k}^{(0)}, Y_{2,k}^{(0)}) \geq 0$. This is used to compute a lower bound on the variance of the total delay $Y_{1,k}^{(0)} + Y_{2,k}^{(0)}$:

$$\mathrm{Var}(Y_{1,k}^{(0)} + Y_{2,k}^{(0)}) \geq \mathrm{Var}(Y_{1,k}^{(0)}) + \mathrm{Var}(Y_{2,k}^{(0)}) \quad (13)$$

### 3.6 IMPLEMENTATION OF ALGORITHM 1

The computational Algorithm 1 for analysis at each queue in a tandem network is implemented in Python programming language. The iterative computation of the distributions of the delays $\{D_{j,k}^{(0)}\}$ at Q1 using the truncated Lindley recursions given by Eqs. (5)-(7) is implemented abstractly so that it can be employed for both Q1 and downstream queues with suitable parametrization. The abstraction is implemented in an Object-Oriented fashion as a generic Q class with methods tabulated in Table 1.

Q1, Q2, Q3, …, are realized by sub-classing the generic Q class. When instantiated as Q1, the given inter-batch arrival time distribution of Class 0 (resp, Class 1) traffic is returned by the method Delta1. The method Zn_Zn1 implements the Lindley Recursion Eq. (3) for computing the distribution of $Z_{n+1}$ from that of $Z_n$; it starts with a call to method Z0_Z1 for incorporating a call to the method Delta1. Using Zn_Zn1, the result of l-step convolve-and-sweep $D_{1,l}^{(0)}$ (resp. $D_{1,l}^{(1)}$), starting from $D_{1,0}^{(0)}$ (resp. $D_{1,0}^{(1)}$) and conditioned on the stationary distribution of $\Delta_{1,k}^{(0)}$ (resp. $\Delta_{1,k}^{(1)}$) being $l$, is carried out by Dn_Dn1. Dn_Dn1 is applied iteratively till convergence to compute the stationary distribution of the delays $D_{1,0}^{(0)}$ (resp. $D_{1,0}^{(1)}$). Finally, the inter-batch *departure* distribution $\Delta_{2,\infty}^{(0)}$ is computed by Delta2 from Eq. (8).

For Q2, this inter-batch arrival distribution of Class 2 is computed by Delta1 and the Algorithm 1 is executed with Class 0 and Class 2 as inputs. This procedure is repeated for the rest of the downstream nodes.

All the methods in Table 1 work with discrete probability distribution functions. To facilitate this, a Python library for definition and manipulation of discrete probability distributions with finite or infinite integer support (spanning positive and negative axis) is developed. The library consists of Dist Class and a set of functions on Dist objects as listed in Table 2.

Specific probability distributions are defined by subclassing Dist. Two of them are used in the results presented in this paper. Geom(10, 0.29825) is the truncated version of the Geometric distribution with parameter 0.29825 with support restricted to {0,1,…,10} and having a mean of 0.425007 and a variance of 0.605592. The mean of about 0.425 for the arrival distributions is chosen so that the average load at a queue with a stream of interest and a cross-traffic stream becomes 0.85 which makes the queue load typical of a communication network node.

Another distribution, termed Prob9by2, with the same support, mean of 0.425002, close to that of Gem(10,0.29825), but with a different variance of 1.31788, is also defined. The distributions are listed in Table 3 and are plotted in Figure 2. Prob9by2 has a heavier tail than Geom(10,0.29825) as can be inferred from Table 3 and Figure 2. The choice of these distributions allows a comparative analysis of the effect of light and heavy-tailed arrival distributions on the network delays.

**Table 1.** The Q Abstract Class

| Method | Purpose |
|---|---|
| Delta1 | For computing the Class 0 batch inter-arrival distribution to Q |
| Z0_Z1 | Basic Convolve-and-Sweep Recursion step for computing the distribution of $Z_1$ from that of $Z_0$ |
| Zn Zn1 | Basic Convolve-and-Sweep Recursion step for computing the distribution of $Z_{n+1}$ from that of $Z_n$ |
| D0_D0l | For computing the distribution of l-fold convolution of $D_0^{(0)}$ when $\Delta_{1,k}^{(0)} = l$ using Z0_Z1 and Zn_Zn1 repeatedly |
| Dn_Dn1 | Recursion step for computing the distribution of $D_{n+1}^{(0)}$ from that of $D_n^{(0)}$using Zn Zn1 repeatedly |
| D_stat | For computing the stationary delay distribution $D_\infty^{(0)}$, by repeatedly applying the Dn_Dn1 iteration |
| Delta2 | For computing the stationary inter-departure distribution from Q |

**Table 2.** Library for Working with Discrete Distributions

| | |
|---|---|
| **Dist Methods** | "name", "min_val", "max_val", "prob_mass", "variate", "moment", "var", "std_dev", "display_par_data", "set_name", "set_min_val""set_max_val", "set_prob_dist", "par_data", "set_par_data", "put_mass", "linf_norm", "cdf" |
| **Functions on Dists** | "scale_mass", "scale_dist", "plot", "add_dists", "linf_distance_dists", "convolve_dists", "normalize", "truncate_normalize" |

**Table 3.** Two Discrete Probability Distributions

| Point | **Mass** (Rounded to 4 decimals for display) | |
|---|---|---|
| | **Geom(10,0.29825)** | **Prob9by2** |
| 0 | 0.7018 | 0.7827 |
| 1 | 0.2093 | 0.1267 |
| 2 | 0.0624 | 0.0517 |
| 3 | 0.0186 | 0.0142 |
| 4 | 0.0056 | 0.0067 |
| 5 | 0.0017 | 0.0042 |
| 6 | 0.0005 | 0.0042 |
| 7 | 0.0001 | 0.0030 |
| 8 | 4.394e-05 | 0.0022 |
| 9 | 1.310e-05 | 0.0020 |
| 10 | 5.569e-06 | 0.0022 |

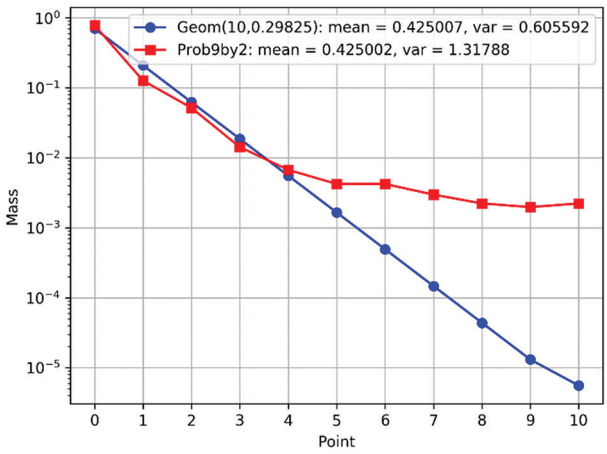



**Fig. 2.** Two Discrete Probability Distributions

Specific probability distributions are defined by subclassing Dist. Two of them are used in the results presented in this paper. Geom(10, 0.29825) is the truncated version of the Geometric distribution with parameter 0.29825 with support restricted to {0,1,…,10} and having a mean of 0.425007 and a variance of 0.605592. The mean of about 0.425 for the arrival distributions is chosen so that the average load at a queue with a stream of interest and a cross-traffic stream becomes 0.85 which makes the queue load typical of a communication network node.

Another distribution, termed Prob9by2, with the same support, mean of 0.425002, close to that of Gem(10,0.29825), but with a different variance of 1.31788, is also defined. The distributions are listed in Table 3 and are plotted in Figure 2. Prob9by2 has a heavier tail than Geom(10,0.29825) as can be inferred from Table 3 and Figure 2. The choice of these distributions allows a comparative analysis of the effect of light and heavy-tailed arrival distributions on the network delays.

### 3.7 SIMULATION FOR BENCHMARKING

The results obtained from the computational algorithms are validated against those from simulation. An event-driven simulation program is developed for this purpose. The event-driven program generates columnated output files recording the Queue Lengths and Waiting Times of different classes of jobs at each of the queues as well as the total waiting time in both the queues. The simulations are repeated for multiple runs and empirical quantities are computed using the time and ensemble statistics. Statistical routines for analyzing the data from individual runs as well as aggregated statistics from multiple runs are developed. The statistical quantities estimated include mean, variance and distribution of queue lengths and waiting times.

## 4. RESULTS AND DISCUSSION

The developed computational algorithm is validated using two instances of the tandem network shown in Figure 1, each with batch sizes given by Geom(10, 0.29825) and Prob9by2 distributions from Table 3 and Figure 2. The results are benchmarked against a simulation-based analysis of the same as outlined in Sec. 3.7.

The computational routines developed (Sec. 3.6) are executed with a precision of $10^{-6}$. Stationary distributions of waiting times of an arbitrary Class 0 job at Q1 and Q2 are computed. The computed results are used to obtain a lower bound on the end-to-end delay. For benchmarking, the simulation program is executed with the same arrival distributions and repeated for 25 runs. Results from the numerical computation and simulation are compared and discussed.

### 4.1 MARGINAL DELAY DISTRIBUTIONS

Figures 3-8 present the results of estimated delay distributions from simulation and computation; Figures 3-5 depict the results for the Geom(10,0.29825) batch size distribution and Figures 6-8 show the results for Prob9by2 batch size distribution.

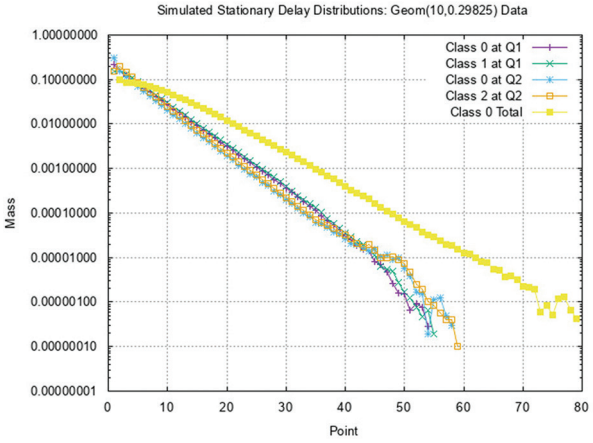



**Fig. 3.** Simulation Results: Geom(10, 0.29825) Data

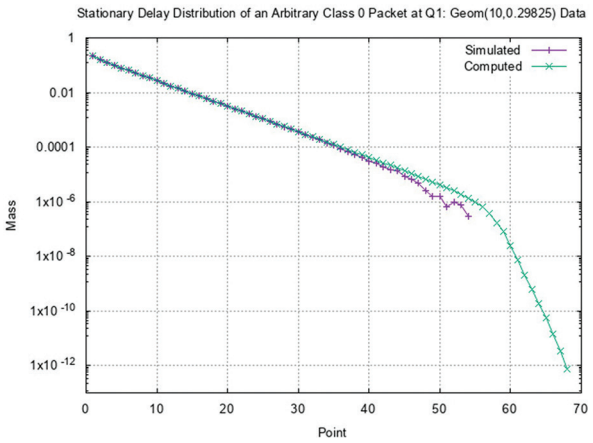



**Fig. 4.** Stationary Delay Distribution of Class 0 at Q1: Geom(10,0.2985) Batch Sizes

Figures 3 and 6 present the baseline simulation-based estimated distributions of delay experienced by a typical packet from each of the three classes at Q1 and Q2 as well as the total delay of a typical Class

0 packet in the two queue tandem, respectively for Geom(10,0.29825) and Prob9by2 distributions.

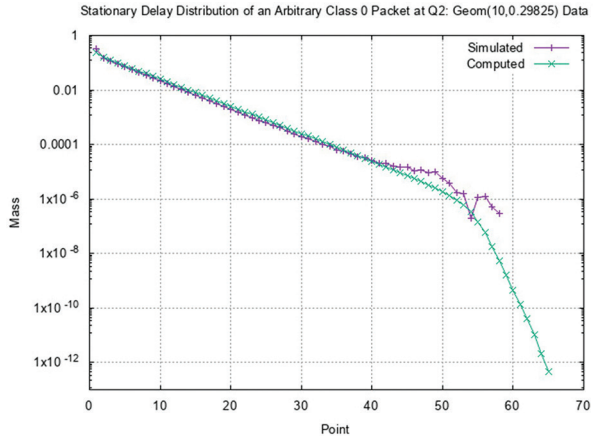



**Fig. 5.** Stationary Delay Distribution of Class 0 at Q2: Geom(10,0.2985) Batch Sizes

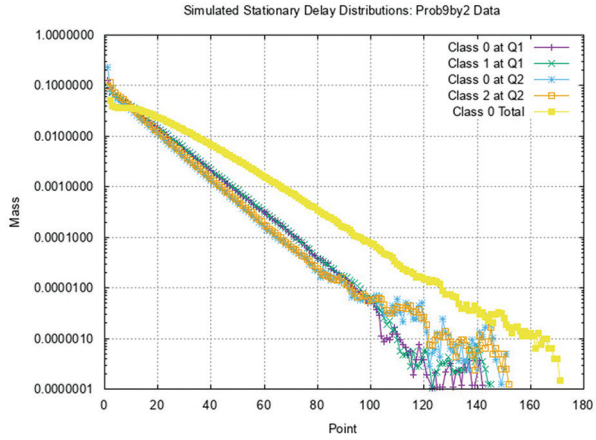



**Fig. 6.** Simulation Results: Prob9by2) Data

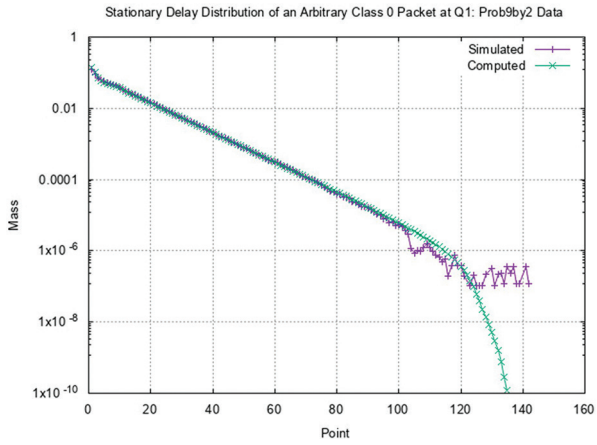



**Fig. 7.** Stationary Delay Distribution of Class 0 at Q1: Prob9by2 Batch Sizes

Figures 4 and 5 present the comparison of the computed and simulated delay distributions for a typical Class 0 packet in Q1 and Q2, respectively. For values of the delays whose probabilities are reliably estimated by the simulation, there is very good agreement between the simulated and computed quantities. Since large delays are rare events, their estimation via simulation is unreliable, as seen in the plots.

Figures 7 and 8 present the validation of computed delay distributions for a typical Class 0 packet at Q1 and Q2, against simulation for Prob9by2 data. Again, it can be observed that the computed results are validated by their simulation-based estimations.

Since Geom(10,0.29825) has a lighter tail than Prob-9by2 (cf. Figure 2), the delay distributions of the former case are lighter (stochastically less than) compared to those of the latter case, as observed in Figures 4 and 7 as well as in Figures 5 and 8.

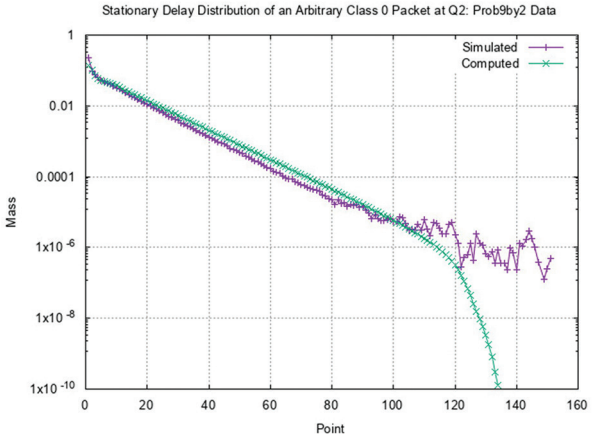



**Fig. 8.** Stationary Delay Distribution of Class 0 at Q2: Prob9by2 Batch Sizes

### 4.2 END-TO-END DELAYS

The associativity of the stationary delays at Q1 and Q2 is verified for results of numerical computation as well as simulation for Geom(10,0.29825) and Prob9by2 batch size distributions. The results are summarized in Tables 4 and 5.

In both cases, it is observed that the variance of the stationary total delay (estimated by simulation) is lower bounded by the sum of variances of these stationary delays at individual queues (for both simulation and numerical computation). This provides a validation of the lower bound given by Eq. (13). The variance of delays with Prob9by2 batch arrival distributions is larger than those with Geom(10,0.29825) distributions since the heavier tail of Prob9by2 leads to larger variances.

**Table 4.** Stationary Delay Variances: Geom(10,0.29825) Data

| | Simulated | Computed |
|---|---|---|
| Var [Delay at Q 1] | 21.0888 | 21.2862 |
| Var [Delay at Q 2] | 16.7995 | 18.6227 |
| Sum of Variances | **37.8883** | **39.9089** |
| Var [Total Delay] | **44.7317** | NA |

**Table 5.** Stationary Delay Variances: Prob9by2 Data

| | Simulated | Computed |
|---|---|---|
| Var [Delay at Q 1] | 107.5180 | 111.9067 |
| Var [Delay at Q 2] | 84.4776 | 108.3647 |
| Sum of Variances | **191.9956** | **220.2714** |
| Var [Total Delay] | **226.5120** | NA |

## 5. CONCLUSION

A computational approach to the delay distributions in multiclass discrete time tandem networks with general batch size distributions and interfering traffic at each node is developed. The delays of the first packets from consecutive batches of a class are related by using a sequence of truncated Lindley recursions. A convolve-and-sweep algorithm is developed to handle the non-renewal arrival processes and solve these recursions and compute the distributions of the delays and inter-departure distributions.

The delays experienced by a typical packet at different queues are shown to be associated random variables. This is used to compute a lower bound on the variance of the end-to-end delay as the sum of variances of the individual delays experienced at the queues.

An object-oriented implementation of the developed algorithm is carried out facilitating the modelling of different network configurations. A library of routines to compute with discrete probability distributions is also developed to aid the computations.

The developed computational algorithm for the delay distributions and the lower bound on the variance of the total delay are validated using simulation for a tandem network of two queues with cross-traffic at each node under two different batch size distributions.

There are several directions for extending the work presented in this paper. Algorithm 1 can be applied to tandem networks with more than two classes at each node and each stream traversing several nodes. It can be extended to handle more general arrival processes such as Markov modulated arrivals. The solution approach may be extended to compute the joint distributions of the delays in different queues which will allow exact computation of the total end-to-end delays. A non-trivial extension of the developed approach would be to extend it to networks with routing and feedback.

## 6. APPENDIX

In this Appendix, Theorem 1 is proved using Mathematical Induction.

The delays experienced by an arbitrary Class 0 job at Q1 and Q2 are, respectively, $Y_{j,k}^{(0)} = D_{j,k}^{(0)} + \theta, j=1,2$ (cf. Eq. (2)). Since $\theta$ is independent of $D_{j,k}^{(0)}$, it is enough to prove that $D_{j,k}^{(0)}$, $j = 1,2$ are associated random variables.

The idea of the proof from [23] is adapted for showing that $D_{1,k}^{(0)}$ and $D_{2,k}^{(0)}$ are associated.

For any non-negative integers $x_0$ and $x_1$,

$$((x_0 - 1)^+ + (x_1 - 1)^+ \geq (x_0 - 2)^+ \quad (14)$$

Applying it iteratively on Eqs. (5)-(6), we obtain

$$D_{1,k+1}^{(0)} \geq \left(D_{1,k}^{(0)} - \Delta_{1,k}^{(0)}\right)^+ \quad (15)$$

from Eq. (7). Similarly, from Eqs. (9)-(12),

$$D_{2,k+1}^{(0)} \geq \left(D_{2,k}^{(0)} + Z_{2,k}^{(0)} - \Delta_{2,k}^{(0)}\right)^+ \quad (16)$$

Inserting Eq. (8) into Eq.(16) gives,

$$D_{2,k+1}^{(0)} \geq \left(D_{2,k}^{(0)} + Z_{2,k}^{(0)} - \Delta_{1,k}^{(0)} + D_{1,k}^{(0)} - D_{1,k+1}^{(0)}\right)^+ \quad (17)$$

Hence, from Eq. (10), it follows that

$$D_{2,k+1}^{(0)} \geq \left(D_{2,k}^{(0)} + X_{A_{1,k}^{(0)}}^{(0)} - \Delta_{1,k}^{(0)} + D_{1,k}^{(0)} - D_{1,k+1}^{(0)}\right)^+ \quad (18)$$

Now, assume $D_{1,k}^{(0)}$ and $D_{2,k}^{(0)}$ are associated random variables. From Eqs. (15) and (18), $D_{1,k+1}^{(0)}$and $D_{2,k+1}^{(0)}$ are increasing functions of the random vector

$$\left(D_{1,k}^{(0)}, D_{2,k}^{(0)}, X_{A_{1,k}^{(0)}}^{(0)}, -\Delta_{1,k}^{(0)}\right) \quad (19)$$

Since $X_{A_{1,k}^{(0)}}^{(1)}$ and $-\Delta_{1,k}^{(0)}$ are independent random variables, they are associated [22]. Also, as a pair, they are independent of pair of random variables $\{D_{1,k}^{(0)}, D_{2,k}^{(0)}\}$ which are assumed to be associated. Hence, the vector in Eq. (19) is a vector of associated random variables [22]. Since $D_{1,k+1}^{(0)}$and $D_{2,k+1}^{(0)}$ are increasing functions of this random vector, they are associated [22]. This proves the induction hypothesis.

For the system starting with zero packets, the same argument as in the above paragraph shows that $D_{1,1}^{(0)}$ and $D_{2,1}^{(0)}$ are associated random variables, establishing the basis case.

Hence, by Mathematical Induction, it follows that $D_{1,k}^{(0)}$ and $D_{2,k}^{(0)}$ are associated random variables, for all $k \geq 0$, completing the proof.

## 7. REFERENCES


[1] D. Bertsekas, R. Gallager, "Data Networks", 2nd Edition, Prentice-Hall, 1992.

[2] A. Kumar, D. Manjunath, J. Kuri, "Communication Networks: An Analytical Approach", Academic Press, 2004.

[3] H. Bruneel, B. G. Kim, "Discrete–Time Models for Communication Systems Including ATM", Kluwer Academic, 1993.

[4] A. S. Alfa, "Applied Discrete Time Queues", Springer, 2016.

[5] T. Zhang, B. Liu, "Exposing End-to-End Delay in Software-Defined Networking", International Journal of Reconfigurable Computing, Vol. 2019, 2019, p. 7363901.

[6] H. Daduna, "Discrete Time Networks with Product Form Steady States", Queueing Networks: A Fundamental Approach, Springer, 2011, pp. 269-312.

[7] H. Daduna, "Discrete Time Analysis of a State Dependent Tandem with Different Customer Types", Lecture Notes in Computer Science, Vol. 1337, Springer, 1997, pp. 287-296.

[8] O. J. Boxma, "On a Tandem Queueing Model with Identical Service Times at Both the Counters", Advances in Applied Probability, 1979, Parts I, pp. 616-643, Part II, pp. 644-659.

[9] O. P. Vinogradov, "On the Output Stream and the Joint Distribution of Sojourn Times in a Multiphase System with Identical Service", Theory of Probability and Applications, Vol. 40, 1995, pp. 581-588.

[10] S. K. Walley, A. M. Viterbi, "A Tandem of Discrete-Time Queues with Arrivals and Departures at Each Stage", Queueing Systems, Vol. 23, pp. 157-176, 1996.

[11] N. D. Gangadhar, V. Sharma, "Computational Analysis of a Discrete Time Tandem Network of Queues with Intermediate Arrivals and Departures", Proceedings of the Second Canadian Conference on Broadband Research, 1998, pp. 350-359.

[12] M. J. Neely, "Exact Queueing Analysis of Discrete Time Tandem Networks with Arbitrary Arrival Processes", Proceedings of the IEEE International Conference on Communications, Paris, France, 20-24 June 2004, pp. 2221-2225.

[13] W.-C. Lau, S.-Q. Li, "Traffic Distortion and Inter-source Cross-correlation in High Speed Integrated Networks", Computer Networks and ISDN Systems, Vol. 29, 1997, pp. 811-830.

[14] D. Yates, J. Kurose, D. Towsley, M. G. Hluchj, "On Per Session End to End Delay Distributions and the Call Admission Problem for Real Time Applications with QoS Requirements", Proceedings of ACM SIGCOMM, 1993, pp. 2-12.

[15] W. Whitt, "Approximations for Departure Processes and Queues in Series", Naval Research Logistics Quarterly, Vol. 31, 1984, pp. 499-521.

[16] G. Hasselinger, E. S. Rieger. "Analysis of Open Discrete Time Queueing Networks: A Refined Decomposition Approach", Journal of the Operations Research Society, Vol. 47, 1996, pp. 640-653.

[17] S. J. Golestani, "Network Delay Analysis of a Class of Fair Queueing Algorithms", IEEE Journal on Selected Areas in Communications, Vol. 13, 1995, pp. 1057-1070.

[18] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks", IEEE Journal of Selected Areas in Communications, Vol. 13, 1995, pp. 1048-1056.

[19] C. S. Chang, "Queue length and Delay of Deterministic and Stochastic Queueing Networks", IEEE Transactions on Automatic Control, Vol. 39, 1994, pp. 913-931.

[20] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, D. Simeonidou, "An Analytical Model for Software Defined Networking: A Network Calculus-based Approach", Proceedings of IEEE Global Communications Conference, Atlanta, GA, USA, 9-13 December 2013, pp. 1397-1402.

[21] C. Lin, C. Wu, M. Huang, Z. Wen, Q. Zheng, "Performance Evaluation for SDN Deployment: An Approach based on Stochastic Network Calculus. China Communications, Vol. 13 (Supplement 1), 2016, pp. 98-106.

[22] J. D. Esary, F. Proschan, D. W. Walkup, "Association of Random Variables with Applications", Annals of Mathematical Statistics, Vol. 38, 1967, pp. 1466-1474.

[23] S.-C. Niu, "Bounds for The Expected Delays in Some Tandem Queues", Journal of Applied Probability, Vol. 17, 1980, pp. 831-838.

[24] M. Zukerman, "Introduction to Queueing Theory and Stochastic Teletraffic Models", arXiv:1307.2968, 2021

[25] V. Ramaswami, "A Stable Recursion for the Steady State Vector in Markov Chains of M/G/1 Type", Stochastic Models, Vol. 4, 1990, pp. 151-161.

# Efficiency of integration between sensor networks and clouds

Original Scientific Paper

**Filip Tsvetanov**
South West University,
Faculty of Engineering, Department of Communication and computer engineering
Ivan Mihajlov street 56, Blagoevgrad, Bulgaria
ftsvetanov@swu.bg

**Martin Pandurski**
South-West University,
Faculty of Engineering, Department of Communication and computer engineering
Ivan Mihajlov street 56, Blagoevgrad, Bulgaria
tom1000@abv.bg

***Abstract*** *– Numerous wireless sensor networks (WSN) applications include monitoring and controlling various conditions in the environment, industry, healthcare, medicine, military affairs, agriculture, etc. The life of sensor nodes largely depends on the power supply type, communication ability, energy storage capacity and energy management mechanisms. The collection and transmission of sensor data streams from sensor nodes lead to the depletion of their energy. At the same time, the storage and processing of this data require significant hardware resources. Integration between clouds and sensor networks is an ideal solution to the limited computing power of sensor networks, data storage and processing. One of the main challenges facing systems engineers is to choose the appropriate protocol for integrating sensor data into the cloud structure, taking into account specific system requirements. This paper presents an experimental study on the effectiveness of integration between sensor networks and the cloud, implemented through three protocols HTTP, MQTT and MQTT-SN. A model for studying the integration of sensor network - Cloud with the communication models for integration - request-response and publish-subscribe, implemented with HTTP, MQTT and MQTT-SN. The influence of the number of transmitted data packets from physical sensors to the cloud on the transmitted data delay to the cloud, the CPU and memory load was studied. After evaluating the results of sensor network and cloud integration experiments, the MQTT protocol is the most efficient in terms of data rate and power consumption.*

***Keywords****: communication protocols, integration, sensor networks, cloud.*

## 1. INTRODUCTION

Modern technological advances in sensor architecture, device miniaturisation, and wireless networks have facilitated wireless sensor networks' design, distribution, and application (WSN). The sensor networks are self-organising and consist of many different types of sensors, equipped with tools for monitoring, processing and communication, which are located in a certain area for monitoring, controlling and transmitting data to each other via wireless communication. The applications of WSN are numerous and include monitoring and control of a wide variety of conditions in the environment, everyday life, industry, healthcare, medicine, military affairs, agriculture, etc. The life of sensor units largely depends on the power supply types, their ability to communicate, energy storage capacity and energy management mechanisms. The collection and transmission of sensor data from sensor nodes lead to the depletion of their energy. At the same time, the storage and processing of this data require significant hardware resources. Designing additional capability to process the collected data can significantly increase the cost of the sensor. Due to the lack of battery power and bandwidth, the sensor nodes cannot store and process extensive data [1]. Therefore, storing and processing raw data is a challenging task.

On the other hand, cloud structures provide enormous computing power and storage space. Integration between clouds and sensor networks is an ideal solution to the limited computing power of sensor networks, data storage and processing. A new paradigm called "Sensor Cloud Computing" has been formulated to achieve this integration. Therefore, the sensor cloud arises to perform many tasks that are not possible from sensor networks [2]. The widespread use of WSN in many processes poses more and more severe problems related to the ability of people to share and analyse

sensor data in real-time. This large volume of data is a prerequisite for the trend for many companies to prefer and switch to using cloud databases for data storage and processing. Therefore, the collected sensory data is not only stored and processed in the clouds but can be accessed anywhere, anytime. Maintaining and providing the resources to the end-users of the sensor cloud is a challenging and important task. Researchers from academia, industry and standards organisations continue to work and offer potential solutions to this challenge.

This paper presents the sensor cloud architecture, focusing on the sensor network-cloud integration process. Some of the most used integration protocols are briefly analysed. The goal is to conduct an integration efficiency implementation experiment with an actual physically built sensor network. That network will send data to the cloud and performs tests on the influence of the different parameters that transmit data packets via HTTP (Hypertext Transfer Protocol), MQTT (message queue telemetry transport) and MQTT-SN (sensor network) protocols.

The paper contains a representation of Communication protocols for sensor data integration, the impact of the protocols on the integration effectiveness, Models for studying the integration of sensor data into a cloud structure, Communication models for sensor network - cloud interaction, Experimental design, Experimental study of the parameters influence of the transmitted packets on the delay, Results and discussions, Conclusion.

## 2. LITERATURE REVIEW

The conclusions that the authors give in [4, 5, 8, 13, 14, 15,16,17] can be systematised like this: MQTT is more suitable over HTTP when the same connection is reused as much as possible. If connections are created and broken frequently to send individual messages, the performance is not considerable compared to HTTP.

Except the protocols message format, another important feature, determining the integration efficiency, is the protocols communication models.

MQTT uses Pub/Sub model with broker, which collect all data and sends particular messages, only to clients, that are subscribed for them. In this way the payload is reduced, and so it is better for WSN than HTTP Request/Response model [14,15,16,17,18,19,20].

The proposed methods [13,14,15,16,17,18,19,20,21] cannot be easily applied in many IoT applications due to the limitations of IoT devices. Depending on the functional requirements of each model, a suitable solution would be using gateway devices/software with higher processing/memory capabilities. The data is transmitted from end devices to the Gateway, where various optimisation methods can be applied before further transmissions to the cloud [14,15,16,17,18,19,20,21,22].

## 3. COMMUNICATION PROTOCOLS FOR INTEGRATION OF SENSOR DATA TO CLOUD

The biggest challenge in designing "sensor-cloud" systems is establishing a communication channel between devices, gateways, servers and cloud platforms. Therefore, this task requires the use of different protocols. The complete communication stack contains the protocols distributed in four different layers: application, transport, Internet and the channel layer [3]. Some characteristics of popular protocols for "sensor-cloud" integration are shown in Table 1.

**Table1.** Protocols for integration of WSN into the cloud

| Protocols | Characteristic | | | |
|---|---|---|---|---|
| | Communication model | Transport layer | QoS | Security |
| **HTTP** | Request - response | TCP | - | TLS/SSL |
| **MQTT** | Publish-subscribe | TCP | QoS-0, QoS-1 QoS-2 | TLS/SSL |
| **MQTT-SN** | Publish-subscribe | UDP | QoS-0, QoS-1 QoS-2 | TLS |

As can be seen from Table 1. the communication channel can be established by appropriate data transmission protocols. By selecting a protocol in the application layer, we can influence the settings in the transport and Internet layers protocols to be predefined. The channel layer is usually determined by the hardware solutions, including IEEE 802.15.4, Z- wave, 802.11 WiFi, Bluetooth Low Energy (BLE), Zigbee, etc.

In sensor and IoT (Internet of things) networks, many small data blocks from different devices are transferred across different networks. Although the Internet Protocol IP is accepted for most types of communication, it has some problems when applied to IoT sensor networks. Internet access requires application protocols running over TCP/IP (Transmission Control Protocol) or UDP/IP (User Datagram Protocol). In addition, IP addressing depends on the physical location, which causes the problem of network control complexity. To address these issues, various name-based architectures have been discussed, such as Named Data Networking (NDN), Content-Centric Networking (CCN), and Information-Centric Networking (ICN) [4], [5], [6]. MQTT is one of the most commonly used protocols in name-based architectures because it reduces high data transmission costs and provides highly efficient communication in IoT systems. It also uses Name-based routing, thus reducing the need for routing, compared to IP addresses, for IoT traffic flows.

### 3.1. HTTP FOR COMMUNICATION IN SENSOR NETWORKS

HTTP determines how messages are transmitted and formatted on the Internet and all websites. HTTP trans-

fers many small packets when communicating with sensor and IoT devices and provides reliable communication over TCP/IP. Connections established by TCP are released on each access, as the available data is transferred based on IP and URL address and their connection changes dynamically [7]. This communication feature in sensor and IoT devices causes severe costs and consumption of network resources, and long delays.

### 3.2. MQTT

MQTT is ideal for use in many situations, including limited environments, such as communication in M2M and IoT, requiring low power consumption, making it one of the most popular protocol solutions for data transmission in the limited environments [8]. The protocol works on TCP/IP, providing orderly, lossless two-way connections. The MQTT Publish/Subscribe paradigm is event-driven and allows messages to be moved between a broker and two MQTT clients (publisher/subscriber). The broker receives and processes all messages, separates the publisher from the subscriber and acts as a router for the messages, deciding where to send them [9]. The publisher, in turn, creates different topics in the broker, as shown in Fig.1.

The MQTT has three different levels of Quality of Service QoS 0, QoS 1 and QoS 2. The QoS level determines the delivery guarantee of a specific message.

MQTT offers SSL/TLS protocols and a client SSL certificate for the security of the transferred content. The MQTT protocol is not text-based, and without SSL/TLS, communication is fully open, and the password is the main concern.

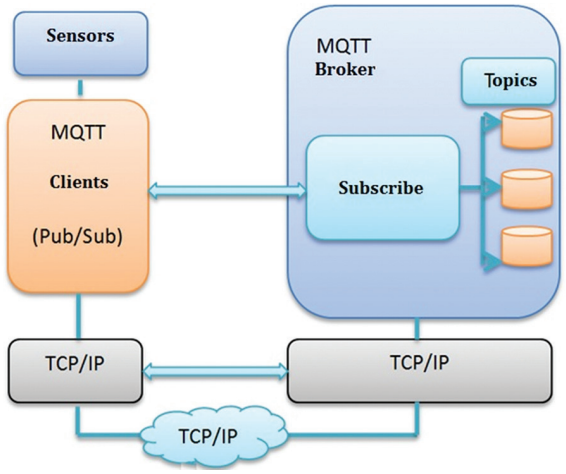



**Fig.1.** MQTT Architecture

Depending on the desired level of security, the MQTT protocol prescribes the following TCP channels:

- 1883 = non-encrypted MQTT and the channel should not be used for sensitive data.
- 8883 = encrypted MQTT, data is encrypted with SSL/TLS, and customer support is required to establish the connection.
- 8884 = encrypted MQTT + client certificate - this is the highest level of security available for MQTT communication. In addition to the encrypted data using the SSL/TLS protocol, the client must authenticate with a certificate issued by the broker. So far, however, this channel is maintained by only a few public brokers (e.g. Mosquitto - test.mosquitto.org server).

### 3.3. MQTT-SN

MQTT-SN is an adapted version of MQTT for WSN, making it suitable for sensor devices due to its low power, bandwidth limitation, and compact messaging. MQTT-SN uses UDP/IP transport communication protocol because it's lighter than TCP/IP. There are three types of MQTT-SN components: MQTT-SN clients, MQTT-SN GW gateways, and MQTT-SN forwarders [10].

MQTT-SN clients connect to the MQTT broker/server via the MQTT-SN GW using the MQTT-SN protocol. MQTT-SN forwarders are responsible for transporting messages to GW. The gateways used are of two types [11], [12]:

- *Transparent Gateway*, where each MQTT-SN connection has a corresponding MQTT connection. This is the most accessible type to implement.
- *The aggregating Gateway* represents multiple MQTT-SN connections that share a single MQTT connection.

## 4. INVESTIGATION OF THE IMPACT OF THE PROTOCOLS ON THE EFFECTIVENESS OF INTEGRATION

WSN faces many limitations and challenges related to the storage of large volumes of sensor data, their processing, scalability, security, accessibility, etc. Connecting the sensor network to the cloud structure solves the problem of storing, processing and transmitting large volumes of data generated by the sensor networks in real-time. This paradigm is known as "Sensor-Cloud" and can be implemented with physical and virtual sensors. Several advantages of using a "Sensor-Cloud" are described in [13]. The WSN - cloud communication can be realised through Gateway devices.

This study aims to assess the integration by examining the impact of the protocols type on the integration of sensor network data to Cloud - HTTP, MQTT and MQTT-SN. The integration evaluation can be done according to package number, topics per packet, and bit value criteria. The general requirement is reliable data transmission from sensor nodes to the database in the cloud. As a parameter's efficiency for the integration, we accept the delay of the transmitted data, the CPU (central processing unit) and RAM (random-access memory) load, showing the consumed energy degree.

### 4.1 MODEL FOR STUDYING THE INTEGRATION OF SENSOR DATA INTO A CLOUD STRUCTURE

The model from fig. 2 is in accordance with the scheme for data transmission between the sensor network and the Cloud via Gateway, analysed in detail in [14].

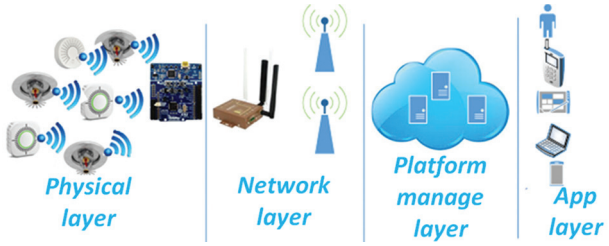



**Fig. 2.** Model for integrating sensor data to a cloud structure

The physical layer-includes different intelligent sensors that send data to the microcomputer.

The network layer includes a microcomputer that acts as a gateway and forwards data to a base station. The base station transports the received data to the cloud platform, which contains data storage and processing servers.

The platform management layer provides data storage and device management features.

The application service layer is connected to the cloud platform via the API (Application Programming Interface) to implement the function for online data requests and remote monitoring.

The advantage of the technology used is the possibility of remote control via mobile phone or tablet and the low cost.

### 4.2. COMMUNICATION MODELS FOR INTERACTION BETWEEN THE SENSOR NETWORK AND THE CLOUD STRUCTURE

The experiments were conducted with two communication models: "request-response" and "publish-subscribe". The communication model request-response is implemented with the HTTP protocol and publish/subscribe via MQTT and MQTT-SN.

### 4.3. DESIGN EXPERIMENT

The Experimental design includes

- An Xbee/Zigbee sensor network has been built.
- The sensor data is collected (via Routers in Mesh topology) and aggregated in the Coordinator.
- Then it's transmitted to the RPI4 microcomputer, which loads pre-developed code for the experiment. RPI4 transmits the sensor data to the ThingBoard Cloud [15] via MQTT, HTTP and MQTT-SN, Fig.4.

The ThingBoard Cloud is free code and supports various integration protocols. As can be seen from the documentation [15], the ThingBoard Cloud is not designed to access MQTT-SN data.


**Fig. 3.** Experiment design

MQTT-SN requires MQTT-SN Gateway, which acts as a protocol converter to convert MQTT-SN messages to MQTT messages [12].

### 4.4. EXPERIMENTAL STUDY OF THE INFLUENCE OF THE PARAMETERS FOR THE TRANSMITTED PACKETS ON THE DELAY.

The main focus of the proposed experiment is the influence of different parameters, such as number of packets, number of topics in packets and bits for each topic, on the speed (delay) of data transmission to the cloud via different protocols HTTP, MQTT and MQTT-SN. The parameter values can be changed via code settings. Many scenarios have been studied.

## 5. RESULTS AND DISCUSSIONS

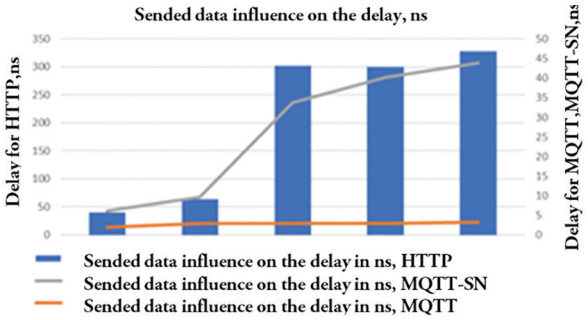



**Fig.4.** The number of transmitted packets Influences the delay

- A study of the number of transmitted packets influences the delivered data delay, Figure 4.

The study shows that we have the highest level of delay in HTTP due to the bigger header. Next in line is MQTT-SN due to gateway usage. MQTT offers a minor delay, a due smallest header of 2 bytes.

- A study of the number of transmitted packets influences RAM in MB

With packages increasing, the difference in protocol's impact on RAM load increases, in a way that HTTP shows the highest level of RAM stress, followed by MQTT-SN and MQTT. The conclusions of the results are based on the already discussed protocol's features.

- Number of transmitted packets Influence on CPU, MHz, Fig. 6.

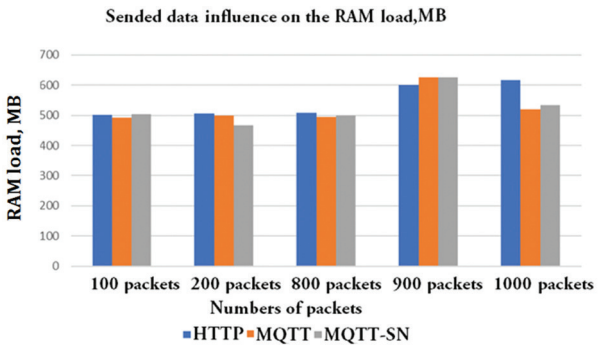



**Fig. 5.** Number of transmitted packets Influence on RAM

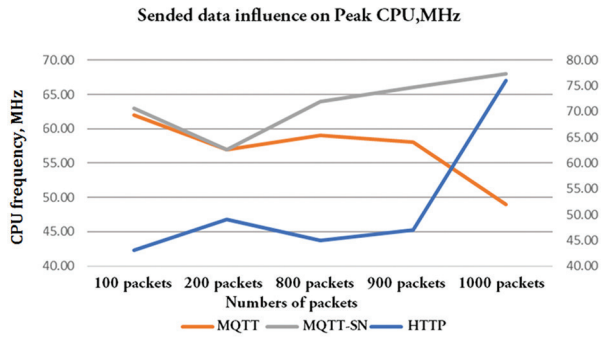



**Fig. 6.** Influence of the number of transmitted packets on the CPU

MQTT-SN shows the highest CPU load at the greatest packet numbers, followed by HTTP and the lowest load levels in MQTT. The results are based on the already discussed protocol's features.

- A study of the complex parameters influence the packets data delay

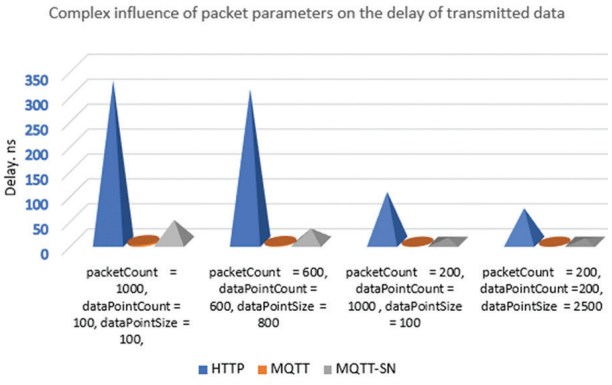



**Fig. 7.** The complex influence of packet parameters on the delay of transmitted data

This study considers the simultaneous influence of the three package parameters: number of packages, number of topics in the packages and bit value for each topic. Fig. 7 shows that in MQTT, the delay is almost unchanged, and MQTT-SN is in second place with minimal delay impact. It can be summarised that the biggest delay is in HTTP.

The limitations are on half of the ThingsBoard Cloud, which allows us to upload only a certain amount of data. Above that border, the cloud doesn't allow us to send more data.

- Verification of the transmitted data

The Wireshark software research tool performs the transmitted packets' destination and size verification for each examined protocol. Fig.8, fig 9 and fig.10 show the transmitted data via the HTTP, MQTT and MQTT-SN protocols.

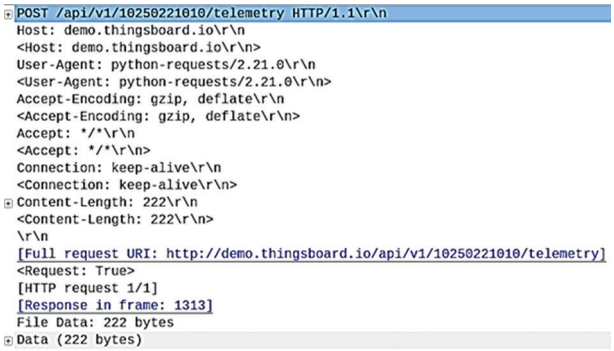



**Fig. 8.** HTTP POST commands

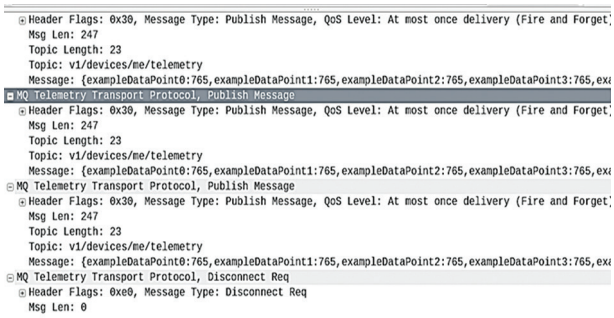



**Fig. 9.** MQTT PUBLISH and DISCONNECT commands

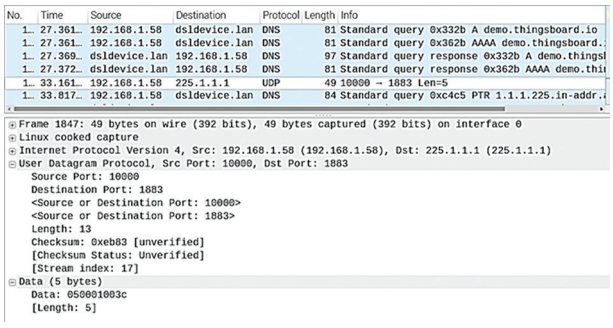



**Fig.10.** Send data via UTP for MQTT-SN

The analysis of the results, obtained from the experimental studies of the transmitted sensory data from physical sensors, gives grounds to draw the following conclusions:

MQTT provides the least delay in data transmission, the least CPU and RAM load, respectively, and requires

the least power consumption and the shortest time for data transmission.

MQTT-SN works with more significant delay and higher energy consumption. One of the reasons, in our opinion, is the additional processing that is implemented in the serialisation process using the MQTT-SN Gateway to convert data structures or objects to a byte stream, which explains the more significant delay.

Evaluating the results of integration experiments, the most efficient data rate and the energy consumption is MQTT.

The obtained results provide useful and practically applicable information for the designers of such systems on the efficiency of the transmitted data through the protocols for integrating sensor data HTTP, MQTT and MQTT-SN to the cloud structure.

The main contributions are the developed Python code for the experiments and the Gateway configurations in both hardware and software. In the general sense, Gateway converts different protocols at different levels. We can connect different devices/programs using gateways, working on different technologies, on single personally designed model platforms.

We plan to include more protocols, such as CoAP, because it works on UDP but uses a "request-response" model. Also, we plan to encrypt the protocols.

## 6. CONCLUSION

Integration between clouds and sensor networks is ideal for the limited computing power of sensor networks, storage, processing and access to sensor data anywhere and anytime. Designing, maintaining and providing end-user resources from the sensor cloud is a challenging and important task.

This paper is devoted to studying the integration between the sensor network and the cloud in transmitting sensor data to the cloud.

Has been created a model for studying the integration of sensor network - Cloud with the communication models for integration – "request-response" and "publish-subscribe", implemented with HTTP, MQTT and MQTT-SN.

An algorithm and Python code have been developed to conduct the experiments, the operability of which has been verified using the Wireshark tool.

The influence of the number of transmitted data packets from physical sensors to the cloud on the speed (delay) of the transmitted data to the Cloud, CPU and memory load was studied.

From evaluating the results of the experiments for integration between the sensor network and the cloud, the MQTT protocol is the most efficient in terms of data transfer rate and energy consumption.

## 7. REFERENCES:


[1] M. Pandurski, F. Tsvetanov, "Research of energy resources of the cluster sensor network ", Journal of Engineering Science and Technology Review, Vol. 13, 2020, pp. 32-36.

[2] F. A. Tsvetanov, "Storing Data from Sensors networks ", Journal IOP Conference Series: Materials Science and Engineering, Vol. 1032, 2021, p. 012012.

[3] M. Burhan, R. Rehman, B. Khan, B. Kim, "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey", Sensors, Vol. 9, 2018, pp. 1-34.

[4] T. Yokotani, S. Yuya. "Comparison with HTTP and MQTT on required network resources for IoT", Proceedings of the International Conference on Control, Electronics, Renewable Energy and Communications, Bandung, Indonesia, 13-15 September 2016, pp. 1-6.

[5] Y. Sasaki, T. Yokotani, "Performance Evaluation of MQTT as a Communication Protocol for IoT and Prototyping", Advanced Technology Innovations, Vol. 4, No. 1, 2019, pp. 21-29.

[6] A. Viswanathan, "Analysis of Power Consumption of the MQTT Protocol", https://www.semanticscholar.org/paper/Analysis-of-Power-Consumption-of-the-MQTT-Protocol-Viswanathan/f7c4c44eeb9ed5b2ec465ff35f78352e696c0724 (accessed: 2022)

[7] C. R. Garzon, "HTTP Request Methods – Get vs Put vs Post Explained with Code Examples", https://www.freecodecamp.org/news/http-request-methods-explained (accessed: 2022)

[8] J. Dizdarević, F. Carpio, A. Jukan, X. Masip-Bruin. "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration", ACM Computing Surveys, Vol. 1, 2018, 30 pages.

[9] Practical guide to MQTT and Mosquitto 2021, How to Install The Mosquitto MQTT Broker on Windows, http://www.steves-internet-guide.com/install-mosquitto-broker (accessed: 2022).

[10] C.-S Park, H.-M. Nam, "Security Architecture and Protocols for Secure MQTT-SN", IEEE Access, Vol. 8, 2020, pp. 226422-226436.

[11] B. Schütz, J. Bauer, N. Aschenbruck, "Improving Energy Efficiency of MQTT-SN in Lossy Environments Using Seed-Based Network Coding", Proceedings of the IEEE 42nd Conference on Local Computer Networks, Singapore, 9-12 October 2017, pp. 286-293.

[12] A. Stanford-Clark, H. L. Truong, "MQTT for sensor networks (MQTT-SN) protocol specification version 1.2.IBM", http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN spec v1.2.pdf (accessed: 2022)

[13] D. R. Kumar, S. Si. Kumar., "Integration of Wireless Sensor Networks with Cloud: A Review", Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering Confluence, Noida, India, 10-11 January 2019, pp. 114–119.

[14] F. Tsvetanov, M. Pandurski, "Some aspects for the integration of sensor networks in cloud structures", Proceedings of International Conference on High Technology for Sustainable Development HiTech , 2018, pp. 249-253.

[15] ThingsBoard Cloud Documentation https://thingsboard.io/docs/paas, (accessed: 2022)

[16] M. Collina, G. Corazza, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST", Proceedings of the 23rd Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Sydney, NSW, Australia, 9-12 September 2012.

[17] T. Yokotani, Y. Sasaki, "Comparison with HTTP and MQTT on Required Network Resources for IoT", Proceedings of the International Conference on Control, Electronics, Renewable Energy and Communications, Bandung, Indonesia, 13-15 September 2016.

[18] J. Dizdarevic, F. Carpio, "A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration", ACM Computing Surveys, Vol. 1, No. 1, 2018.

[19] Y. Sasaki, T. Yokotani, "Performance Evaluation of MQTT as a Communication Protocol for IoT and Prototyping", Advances in Technology Innovation, Vol. 4, No. 1, 2019, pp. 21-29.

[20] M. O. Reddy, J. Seventline, "Performance Analysis of QoS for the MQTT-SN Protocol with Industry Oriented MQTT-SN Gateway and Integration with Cloud MQTT-Server",. International Journal of Future Generation Communication and Networking Vol. 13, No. 3, 2020, pp. 2651-2673.

[21] F. Tsvetanov, M. Pandurski, "Security of the sensory data in the cloud", IOP Conference Series: Materials Science and Engineering, Vol. 1032, 2020, p. 012005.

[22] E. Davis, "Pub/Sub Protocol in WSN: Improved Reliability and Timeliness", Transactions on Internet and Information Systems Vol. 12, No. 4, 2018.

# A Quality of Service Aware Source Routing Based Protocol for Underwater Wireless Sensor Networks

Original Scientific Paper

**K. R. Narasimha Murthy**
M. S. Ramaiah University of Applied Sciences,
Faculty of Engineering and Technology, Department of Computer Science and Engineering
#470-P, Peenya Industrial Area, Bangalore, India
nmurthy.cs.et@msruas.ac.in

**Govind R. Kadambi**
M. S. Ramaiah University of Applied Sciences,
Faculty of Engineering and Technology, Department of Electronic and Communication Engineering
#470-P, Peenya Industrial Area, Bangalore, India
pvc.research.cs.et@msruas.ac.in

***Abstract*** *– Underwater Wireless Sensor Networks (UWSNs) handle many underwater applications such as environment monitoring, surveillance and navigation. These applications generate varied types of traffic such as continuous bit rate, sporadic and different packet sizes, leading to additional QoS requirements that are traffic and application dependent. This paper presents the development of a Quality of Service Aware Source Routing (QASR) protocol. QASR discovers multiple paths from the sources to the sinks and selects the most QoS compatible route among them. QASR is distinctive because it incorporates multiple QoS parameters such as Signal to Noise Ratio (SNR), latency and residual energy. Depending on which of these parameters are chosen, QASR has three variants, namely, QASR-Latency (QASR-L), QASR-Residual Energy (QASR-RE) and QASR-Signal to Noise Ratio (QASR-SNR). The performance of QASR protocol is compared against traditional source routing protocols, with simulations showing a reduction of about 10% to 20% in latency and about 5% to 10% lesser energy consumption than source routing. QASR protocol exhibits comparable performance to classic source routing protocols while simultaneously adhering to the QoS requirements of the application. It is also worth noting that the performance profile of all the three variants of QASR do not have sudden and drastic variations, with the performance profiles showing consistent trend-lines.*

***Keywords****: Underwater Wireless Sensor Network (UWSN), Underwater Acoustic Communication (UAC), source routing protocol, Quality of Service (QoS), application traffic, DESERT simulator*

## 1. INTRODUCTION

Underwater Wireless Sensor Networks (UWSNs) are wireless sensor networks deployed in an underwater environment. UWSNs are used extensively in ocean sampling networks, environmental monitoring, undersea exploration, disaster prevention, assisted navigation, distributed tactical surveillance and mine reconnaissance [1]. Depending on the application that USWNs are deployed for, there is a considerable variation in the type of traffic in application layer that they have to be modelled for. As shown in Fig. 1, the varied applications naturally imply that a UWSN generates different kinds of traffic.

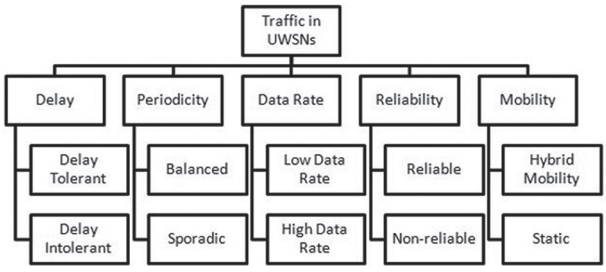



**Fig. 1.** Classification of traffic patterns in UWSNs

Application-based traffic patterns also demand an awareness of different Quality of Service (QoS) parameters. For example, pollution monitoring may

require the system to support sporadic data generation while allowing for slightly relaxed reliability. In contrast, applications that sense underwater seismic activity have more stringent reliability requirements and generate continuous data. Furthermore, military applications that transmit live audio/video streams demand that the links support enhanced bandwidth requirements.

The authors of this paper envision the research presented here as a first step toward a more generic and versatile underwater communication architecture capable of addressing significant system performance parameters such as node mobility and deployment strategy, medium access, routing, modulation techniques, energy efficiency and QoS requirements. A recurring theme in MAC and routing protocols is that they are usually designed for specific application only. In addition, there is a distinct lack of research on unified communication architectures for UWSNs which can handle the communication requirements of different applications.

This paper presents a Quality of service Aware Source Routing (QASR) protocol that can be configured to choose QoS parameters on the fly. In particular, QASR protocol is designed to incorporate three QoS parameters: residual energy, latency and Signal to Noise Ratio (SNR). The principle of source routing is used to discover multiple routes from the source node to the sink node, and the most QoS appropriate route is chosen from among them. QASR protocol achieves energy efficiency by optimizing the route discovery process, caching reusable routes and applying reactive routing. In addition, QASR protocol is extensively simulated to ensure that it conforms to the requirements of various applications such as density of node deployment, traffic characteristics, energy efficiency and data packet size and frequency.

In the existing literature, QoS is viewed differently by different researchers. For example, in some research papers, the emphasis of QoS is mainly confined to traditional output parameters of the system such as throughput, latency, packet delivery and jitter. For UWSNs, another important QoS parameter of relevance is the energy efficiency. With the UWSNs increasingly catering to the more diverse and heterogeneous applications, the energy efficiency has moved from being a best-effort attempt to an essential parameter in taking decisions related to routing, clustering, medium access and even deployment strategies. Among others, the studies pertaining to energy efficient routing protocols are presented in [2] and [3].

In the recent past, void-aware and void mitigation routing has become a topic of considerable importance attracting significant interest from the researchers. Voids are black holes for data, resulting in overuse of the nodes in routing. This potentially causes a link outage because of the depletion of nodes in the network. The authors of [4] present the variants of void aware routing protocols and the associated challenges in their development.

The above referred research papers, and the literature reviewed in Section 2, clearly highlight an interesting shortcoming in the design of routing protocols for UWSNs. Designing a routing protocol for an acceptable single parameter of QoS, such as latency, energy efficiency, void avoidance or link quality, invariably results in an unresolved or unknown trade-off in other QoS parameters. Protocols are generally designed for specific traffic patterns, particular sensing environments and explicit QoS requirements. Therefore, it is a desirable and prudent wish that the needs of the varied applications of UWSNs and the traffic patterns they generate will be better served by a more generic and versatile routing protocol that supports multiple QoS parameters after its deployment also.

The primary emphasis of this paper is to address the following:

- Enhance the concept of source routing so that the source node is aware of all the available routes to reach the sink node.
- Use of this enhanced source routing and appropriate cross-layer information to track the QoS parameters as mandated by the developed routing protocol.
- Develop a source routing based protocol that can choose a path that supports the QoS requirements mandated by the application. For example, if the application changes the QoS parameter of interest, the protocol can handle that also.

The rest of the paper is structured as follows: Section 2 presents the related work, a brief description of some of the other QoS based routing protocols used in UWSNs. Section 3 describes the design of the QASR protocol, including the specifications and flowcharts used. Section 4 details the deployment scenario, subsequently covering the simulation results of QASR protocol and comparative analysis. Section 5 presents the conclusions arrived at in the paper. It also proposes some suggestions for future work in continuation of the research presented in this paper.

## 2. RELATED WORK

The review of literature presented in this section on the development of routing protocols for UWSNs looks at some research studies where specific QoS parameters such as link quality, traffic priority, reliability and channel awareness are considered.

In their study in [5], the authors enhance the Directional Flooding based Routing (DFR) and develop two variants that focus on end-to-end reliability as the QoS requirement. The two variants, namely, QoS-Aware DFR with Angle Adaption (QA_DFR_AA) and QoS-Aware

DFR with Threshold Adaption (QA_DFR_TA), are designed to adapt to node mobility, resulting in dynamic QoS requirements. The original DFR protocol does not accommodate mobile sources and sinks.

These protocols function by tracking the delivery ratio of each data flow and transmitting this information to the source and intermediate nodes of this flow. Since the sources are aware of link reliability using this information, further routing can be calibrated accordingly. Although this dynamic recalibration comes at the cost of increased overhead, there is an improvement in the delivery of packets, thereby satisfying the QoS requirement of reliability.

The authors in [6] propose a Channel Aware Routing Protocol (CARP) that considers the link quality as the QoS parameter of interest. Cross-layer principles are invoked to apply link quality to choose the next hop along the path to the sink. CARP also uses ready information, such as hop count, to route around voids while simultaneously focusing on nodes' residual energy and transmission power control to choose routes. CARP is analyzed by both simulations using Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing (SUNSET) [7] and real-life sea trials in the Mediterranean Sea. CARP is compared with a focused routing-based protocol, Focused Beam Routing (FBR) [8] and a flooding based protocol, EFlood. CARP is found to be at least 40% more energy efficient than FBR and EFlood. It has a significantly higher packet delivery ratio.

In [9], a QoS aware evolutionary routing protocol for underwater wireless sensor networks called QERP is presented. QERP is a greedy clustering-based routing protocol that increases packet delivery, reduces energy consumption, and decreases end to end delay. QERP protocol is based on the assumptions about location awareness, CSMA for medium access, power control and mobility patterns. QERP is designed to be evolutionary in nature, performing crossovers and mutations. A fitness function that considers the clustering cost and link quality cost is derived to perform route selection.

QERP is simulated using MATLAB, and its performance is compared with Depth Based Routing (DBR) protocol [10] and Vector Based Forwarding (VBF) [11] protocol. The delay was the least with QERP followed by DBR and VBF exhibiting the maximum delay. The clusters in QERP are smaller than other protocols that use the same concept, resulting in a better profile of energy consumption. The authors conclude that QERP improves the delay and reliability of data transfer in real-time scenarios.

QoSRP, proposed in [12], is a cross-layer QoS channel-aware routing protocol for the Internet of underwater acoustic sensor networks. This protocol is designed for cross-layer, QoS aware, multichannel routing to address time-critical marine monitoring applications. In addition, there are three mechanisms incorporated in it to aid data gathering to find vacant channels that can support high data rates while simultaneously avoiding congestion and balancing traffic.

QoSRP is simulated using NS-2 and AquaSim 2.0. Its performance is compared against Link quality-aware queue-based spectral clustering Routing Protocol (LRP) for underwater acoustic sensor networks [13], QERP and an energy efficient Multi-objective Evolutionary Routing Protocol (MERP) for reliable data gathering in the Internet of underwater acoustic sensor networks proposed in [14]. QoSRP performs better than the other protocols in conventional output parameters such as throughput, error rate, packet delivery and load balancing.

The research in [15] proposes a priority-based routing algorithm for underwater wireless sensor networks. This algorithm improves the QoS by classifying the traffic as high and low priority based on the delay tolerance. The network area is divided into logical cubes by network barriers. Low priority data is allowed to use only one side of the cube in its attempt to reach the sink, while high priority data uses all the sides of a cube. The nearest neighbor is chosen based on the Euclidian distance.

The performance of this protocol is compared with Geographic and opportunistic routing protocol with depth adjustment for mobile underwater sensor networks (GEDAR) protocol [16] using the OPNET network simulator. The simulations show that the high priority and low priority modes of both the protocols consistently perform better than GEDAR with respect to packet loss, latency, and residual energy.

A Delay-Intolerant Energy-Efficient Routing protocol with sink mobility in underwater wireless sensor networks, DIEER, is presented in [17]. This protocol assures data dissemination, even at the cost of energy efficiency. The study uses a multi-prong optimization approach to optimize sink mobility, data transmission and dissemination. This is applied to 3D mobile sensor networks characterized by dense deployment and mobile sinks. The authors claim that usage of a mobile sink reduces delay and energy consumption. Further, a mobile sink allows the optimization of transmission distance to reduce the number of retransmissions of data.

The DIEER protocol aims to maximize network lifetime and minimize end to end delay. Performance comparison is carried out with Mobicast [18]. The simulation results show that DIEER protocol performs better than Mobicast for dense deployments and achieves lower delays.

A summary of the literature reviewed is tabulated in Table 1. This summary highlights the QoS parameters considered, basic operation and limitations of the protocols reviewed and referred in this paper.

**Table 1.** Summary of Literature Reviewed

| Sl. No. | Name of Protocol | QoS parameter/s | Basic Principles | Limitations of Protocol |
|---|---|---|---|---|
| 1 | QoS Aware DFR [5] | Node mobility and link reliability | Enhancement of DFR that allows node mobility. Routes are established using links that have been proved reliable previously. | Increased overhead possibly resulting in higher latency. |
| 2 | Channel Aware Routing Protocol [6] | Link quality | Link quality is used to choose the next hop along the path to the sink. | Effect of node mobility on protocol performance is not evaluated. |
| 3 | QoS aware evolutionary routing protocol [9] | Clustering cost and link quality cost | A greedy approach is applied that increases packet delivery, reduces energy consumption, and decreases end to end delay. | There is an assumption that the protocol is aware of the locations of the nodes and their mobility patterns. |
| 4 | Cross-layer QoS channel-aware routing protocol [11] | Time-critical communication | Protocol uses channel detection, assignment and forwarding mechanisms to find vacant channels that can support high data rates while avoiding congestion and balancing traffic. | Effect on node mobility on protocol performance is not evaluated. |
| 5 | Priority-based routing algorithm [15] | Traffic classification based on Priority | Network area is divided into cubes and paths are assigned based on priority of data. | Only two levels of priority may not be appropriate for real world scenarios. |
| 6 | QASR (Proposed in this Paper) | Signal to Noise Ratio, Residual Energy and Latency | Concept of source routing is used to select routes based on the QoS parameter chosen. | The route discovery can be enhanced to directly choose the QoS compliant route instead of selecting one from multiple possibilities. |

## 3. DESIGN OF QASR PROTOCOL

The operation of the QASR protocol proposed in this paper is divided into three phases:

Phase 1: Selection of data haul node and data aggregation

Phase 2: Route establishment

Phase 3: Data transfer

These phases are detailed in this section after the introduction of network architecture for QASR.

### 3.1. NETWORK ARCHITECTURE

To facilitate the implementation of QASR, it is assumed that the nodes of UWSN are deployed in clusters in a stretch of ocean. The nodes are classified as sensing nodes, data haul nodes and sinks. Sensing nodes are static nodes responsible for sensing information and broadcasting it to the data haul nodes. The data haul nodes are mobile nodes, analogous to cluster heads, capable of movement and control of two aspects of the communication: They receive data from the sensing nodes in their cluster. This is done by moving within the cluster. The data haul nodes also create a multi-hop ad-hoc network among themselves, sending data collected from their clusters towards the sink. A single cluster is shown in Fig. 2, and a representation of the deployment of sensor nodes is depicted in Fig. 3.

Sensing and data-haul nodes are functionally interchangeable; the sensing node with the highest residual energy is chosen as the data haul node for a particular data flow. The protocol dictates that the data haul nodes be mobile while the sensing nodes are static. There are multiple sinks in the network architecture, and transmission is deemed successful if the data reaches any one of the sinks.

The following are the assumptions to facilitate the protocol design:

Assumption 1: The sensing nodes are aware of their positions in the Cartesian coordinate system.

Assumption 2: The sensing nodes of each cluster are fixed; by extension, the coverage area of each cluster is fixed.

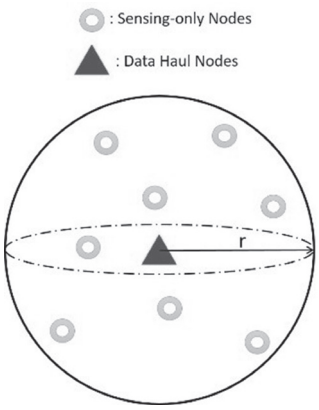



**Fig. 2.** A single cluster consisting of sensing nodes and one data haul node

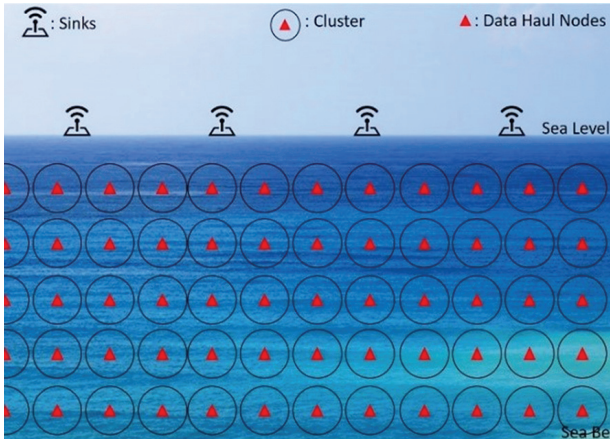



**Fig. 3.** Typical deployment scenario

Sensing and data-haul nodes are functionally interchangeable; the sensing node with the highest residual energy is chosen as the data haul node for a particular data flow. The protocol dictates that the data haul nodes be mobile while the sensing nodes are static. There are multiple sinks in the network architecture, and transmission is deemed successful if the data reaches any one of the sinks.

The following are the assumptions to facilitate the protocol design:

Assumption 1: The sensing nodes are aware of their positions in the Cartesian coordinate system.

Assumption 2: The sensing nodes of each cluster are fixed; by extension, the coverage area of each cluster is fixed.

### 3.2. SELECTION OF DATA HAUL NODE AND DATA AGGREGATION

One of the main assumptions of QASR protocol is that the area of interest covered by each cluster is fixed along with the members of the cluster. Each cluster can be viewed as an undirected, fully connected graph $G$ ($V$,$E$) where $V \in$ set of nodes in that cluster and $E$ represents the links. The weight of each link is calculated as the Euclidian distance between the pair of nodes it connects. For example, if $S_i$ and $S_j$ are two sensing nodes with coordinates $S_i$ ($x_i$, $y_i$, $z_i$) and $S_j$ ($x_j$, $y_j$, $z_j$), then the weight of the edge between them is given by Equation (1).

$$E_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (1)$$

The graph model of the cluster is illustrated in Fig. 4. This is a cluster with five sensing nodes, and each node is connected to each other.

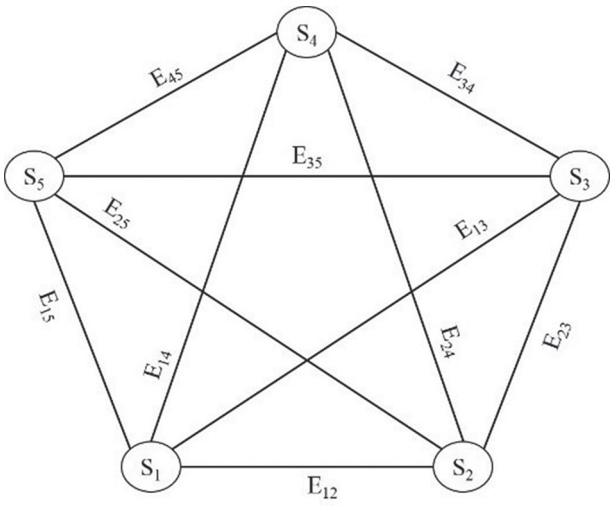



**Fig. 4.** A fully connected cluster showing the edges between the sensing nodes

The data haul node for each cluster is chosen as the node with the highest residual energy. The data haul aggregates the data by visiting each of the sensor nodes. Since one of the sensing nodes is designated as the data haul node, QASR protocol calculates the shortest path of all-pairs for the graph so that each node is aware of the shortest path to be taken to reach all the nodes.

The all-pairs shortest path involves finding the shortest path from all possible sources to destinations within the cluster. Since this is an undirected graph with no negative weights, the solution can be computed using Dijkstra's algorithm or the Floyd-Warshall algorithm. However, since Dijkstra's algorithm has a higher time complexity ($(ElogV)$) than the Floyd-Warshall algorithm ($O(V^3)$), the latter is chosen to find the all-pairs shortest path.

Floyd-Warshall algorithm uses dynamic programming to check if a given path from vertex $S_i$ to $S_j$ has a lower total weight alternative if the path goes through another vertex $S_k$, i.e., $S_i \rightarrow S_k \rightarrow S_j$ is a lower weight alternative to $S_i \rightarrow S_j$. The mathematical formulation of the Floyd-Warshall algorithm is given in Equation (2).

$$\mathrm{A[N]} \leftarrow D_{S_i,S_j} = \begin{cases} 0 & if\ S_i = S_j \\ E(S_i, S_j) & if\ S_i\ and\ S_j\ are\ connected \\ \infty & if\ S_i\ and\ S_j\ are\ not\ connected \end{cases} \qquad (2)$$

In Equation (2), A[N] refers to the cost matrix with N nodes, $D_{Si\,Sj}$ indicates the total distance between nodes $S_i$ and $S_j$ and $E(S_i, S_j)$ is the edge between the nodes $S_i$ and $S_j$.

This paper uses the Floyd-Warshall algorithm to create the array of routes R[N]. The algorithm is detailed in Algorithm 1 as shown below:

**Algorithm 1** Floyd-Warshall Algorithm

**Require:** All pair shortest paths $P[N]$
**Ensure:** Cost Matrix $A[N]$

```
R[N] ← path (S_i, S_j)=0        ∀ S_i, S_j ∈ N
do k=1,N
    do i=1,N
        do j=1,N
            if D(S_i, S_k)+D_(S_k, S_j)<D(S_i, S_j)
            then D(S_i, S_j)= D(S_i, S_k)+D(S_k, S_j)
                if path (S_i, S_k)=0
                then path (S_i, S_j)←k
                else path (S_i, S_j)← path (S_j, S_k)
                end if
            end if
        end do
    end do
end do
do i=1, N
    do j=1,N
        if path (S_i, S_j)=0 then path (S_i, S_j)←j
        end if
    end do
end do
```

This algorithm ensures that all the nodes have the cost matrix and know the shortest path to reach all the other nodes. This information is helpful once a node is designated as the data haul node.

To identify the data haul node, QASR protocol performs an intra-cluster MAC-level broadcast of the residual energy of each node in the cluster. Every node in the cluster broadcasts a *ResEn* frame that consists of its node ID and residual energy. This *ResEn* frame is stored by each node that receives it and subsequently forwarded to the other sensing nodes in the cluster. Each node continues to receive this frame from other nodes until they have the residual energy of all the nodes in the cluster.

The node with the highest residual energy is chosen as the data haul node. Mobility patterns are set up so that the data haul node moves to the vicinity of each sensing node according to the cost matrix derived. The cost matrix does not change based on energy or any other parameter, and it continues to be applied as a shortest path finder.

At the end of phase 1, the UWSN is ready to discover routes to the sink and use them according to the application requirements. Additionally, data haul nodes are identified, and mobility patterns have also been set up.

### 3.3. ROUTE ESTABLISHMENT

Route establishment in wireless networks can be primarily of two types. In reactive routing, a route is established between node A and node B only when there is data to be transmitted from node A to B. Conversely, proactive routing involves all the nodes in discovering a route to all other nodes at the initial start of the network irrespective of whether they will be needed. QASR protocol performs reactive routing by considering the mobile data haul nodes as a multi-hop ad-hoc network. This reactive routing is based on the Source routing for Underwater Networks (SUN) protocol [19].

A route request (RREQ) packet is broadcast from the source data haul node with the destination as the broadcast address of the sinks. This RREQ packet is forwarded by the intermediate data haul nodes till the RREQ reaches one of the sinks. Every intermediate node that forwards the route discovery packet adds its address to the header. When the sink receives the RREQ, it completely knows the entire path followed. The sink, as a destination, unicasts a route reply packet (RREP) along the same path to the source data haul node. Since the RREQ is broadcasted, the source receives multiple RREPs and can choose the 'best' route to transmit the data. If an intermediate node does not receive an acknowledgement for forwarding the data to the next hop, it initiates a route error (RERR) packet and transmits it towards the source so that all the intermediate hops can update their routing tables. Routes that are used successfully are cached so that they can be reused.

The QASR protocol proposed in this paper consists of two types of packets: Control packets and data packets. Control packets are used for route discovery and maintenance, while data packets are used for aggregated data at data haul nodes. The control packet format is shown in Fig. 5.

| Packet Type | Mode | Mode Value | Source Address | Packet ID | Destination Address | Path |
|---|---|---|---|---|---|---|

**Fig. 5.** Control packet format of QASR

In Fig. 5, Packet Type indicates whether the packet is RREQ, RREP or RERR. If the packet is an RREQ, the destination address is the broadcast group address of sinks, while the RREP uses the destination address field in unicast mode. The Path field is implemented as a structure to store a list of the nodes visited during route discovery.

QASR protocol achieves QoS awareness by working in three modes: SNR, residual energy and latency. In the SNR and residual energy modes, the mode value field stores the minimum SNR or residual energy encountered during propagation of RREQ. The same information is unicast back to the source in the RREP packet, enabling the source to choose the path based on the QoS requirement of the application. The latency mode is similar, but it tracks the cumulative latency in the entire path instead of hop to hop. This control packet structure enables the source data haul node to choose a path with minimum SNR, residual energy or end-to-end delay. The caching of routes allows them to be reused without going through route discovery overhead.

#### Cross Layer Interactions

For QASR protocol to achieve the different modes of operation and to identify the data haul node itself, a significant amount of information is used that is not conventionally available at the network layer. For example, the requirements of the application based on which the routes are chosen are available at the application layer. The process of computing the minimum SNR and residual energy requires information typically available at the Physical layer. QASR protocol works on the principle of cross-layer optimization, assimilating all this information at the network layer and taking routing decisions based on them. Fig. 6 depicts the exchange of information in the QASR protocol.

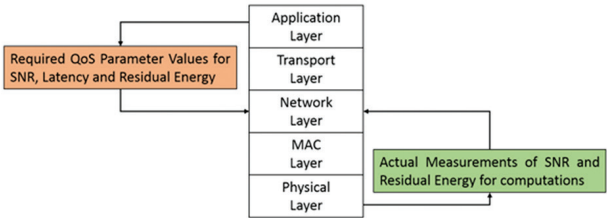



**Fig. 6.** Cross layer interactions of QASR

**Route Discovery**

RREQ packets are used if the cache does not have a route from source to destination, or a RERR is encountered. The flowchart for the generation of RREQ is shown in Fig. 7, and its propagation at the intermediate nodes is shown in Fig. 8.

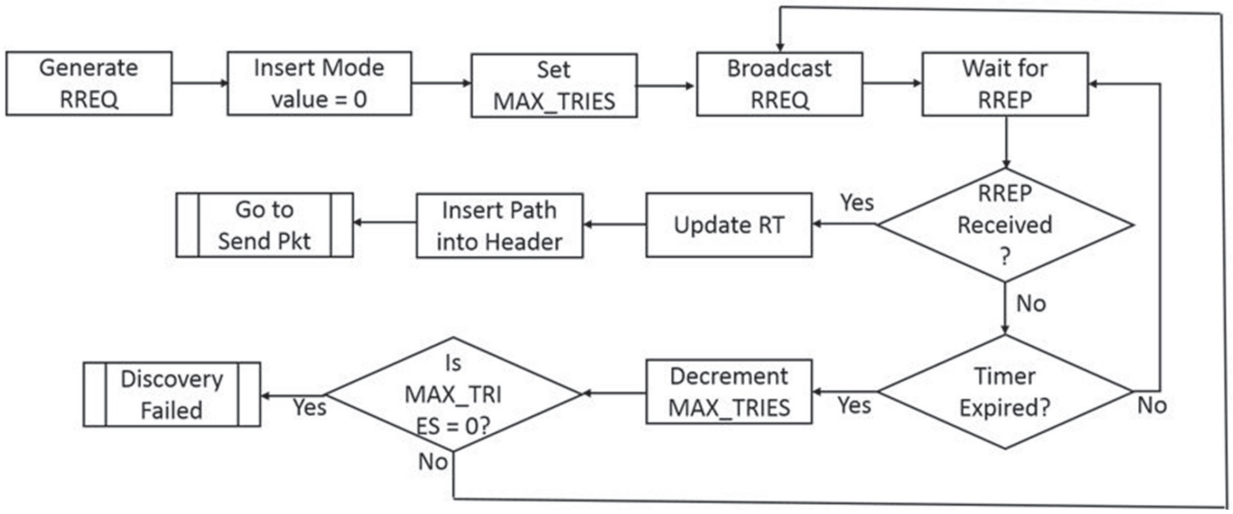



**Fig. 7.** Generation of RREQ

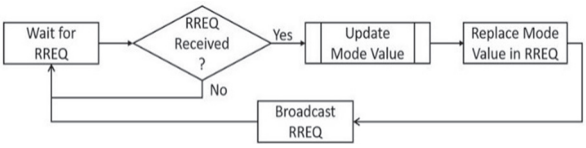



**Fig. 8.** Propagation of RREQ

When the RREQ is generated initially, the mode value is set to zero. When an intermediate node receives an RREQ, it extracts the mode value from it. Then the intermediate node calculates the SNR of the incoming packet or the residual energy of itself from the physical layer. If the calculated SNR/residual energy is lesser than the mode value in the packet, the intermediate node updates the mode value. This ensures that the mode value indicates the minimum SNR or residual energy at every hop until that point in the propagation. The intermediate node also updates the timestamp field in RREQ to reflect the time of receipt of the packet. Once the RREQ reaches the sink, mode values and timestamp information are transmitted back to the source data haul node. The algorithm for updating the mode value is shown below.

**Algorithm 2** Mode Value Update

**Require:** Updated mode value in RREQ
**Ensure:** Existing mode information *RREQ.mode*, Existing mode value *RREQ.modeval*, Existing time *RREQ.time*, timestamp *time*, Signal to Noise Ratio *SNR* and residual energy *RE*

*1:* **do while** *RREQ.received* is **TRUE**
*2:* **if** *RREQ.mode=SNR* **then**
3: **if** *RREQ.modeval* > *PHY.SNR* **then**
*RREQ.modeval* = *PHY.SNR*
4: **end if**
5: **if** *RREQ.mode* = *RE* **then**
6: **if** *RREQ.modeval* > *PHY.RE* **then**
*RREQ.modeval* = *PHY.RE*
7: **end if**
8: *RREQ.time* = *time*
9: **end do**

**Route Reply**

The transmitted RREQs reach the destination with information about minimum SNR or residual energy and the end-to-end delay in the route that was taken. The destination copies these values into the RREP packet and unicasts it using the path information already present to transmit this to the source. The algorithm for the generation and propagation of RREP is presented in Fig. 9.

After the route establishment phase, QASR protocol is aware of all possible routes from source to destination. The data is aggregated at the data haul nodes, ready for transmission towards the sink.

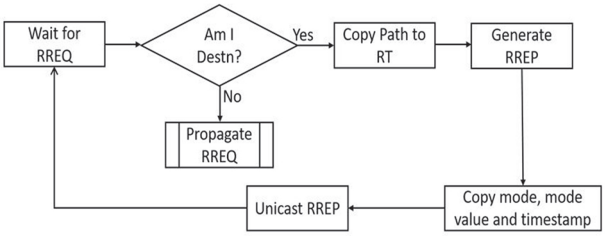



**Fig. 9.** Generation and propagation of RREP

Once the route establishment is completed, the source must choose the best route among the available options. This is carried out by considering the ap-

plication requirements of either SNR, latency or residual energy. Depending on the required mode of operation, QASR protocol initializes the respective path and inserts it into the data packet header.

The packet is subsequently transmitted to the next hop. The functioning of QASR protocol at the source data haul node is illustrated in Fig. 10.

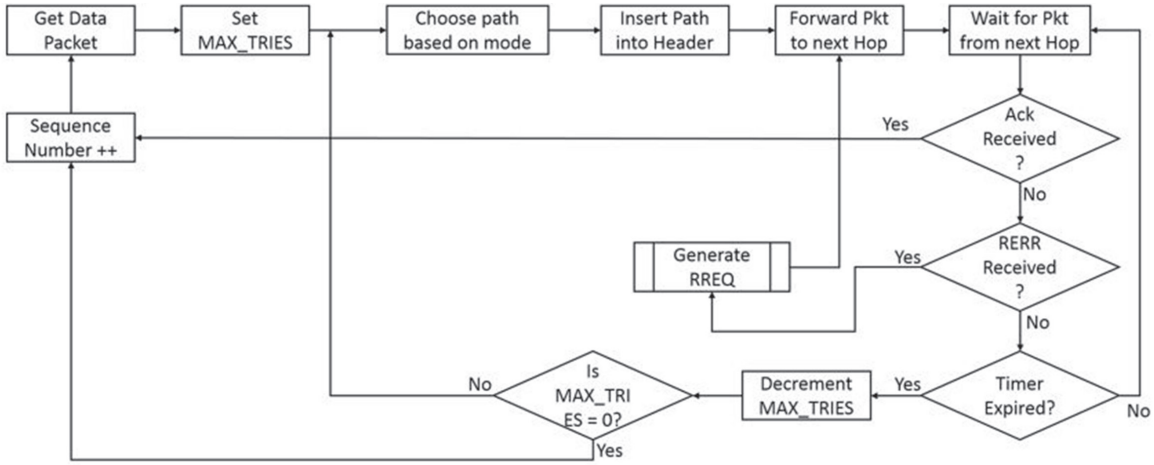



**Fig. 10.** QASR at source data haul node

## 4. SIMULATION ANALYSIS

To simulate QASR protocol, the DESERT Underwater Framework is used, which is an NS-2 based framework equipped with extensions to facilitate cross layer communication and multiple radio interfaces and underwater acoustics. The performance of QASR protocol is compared with Source routing for Underwater Networks (SUN) [19] and Information Carrying based Routing Protocol (ICRP) [20]. The performance of these protocols is evaluated in terms of their Packet Delivery Ratio (PDR), latency and energy consumption. The variable parameters are node mobility, node density and data rate. Subsequently, an analysis of the energy consumption is presented with respect to the improvement noticed in the energy efficiency of the QASR protocol.

### 4.1. DETAILS OF SCENARIO FOR SIMULATION STUDIES

The coverage area is 3000m3. Every cluster is assumed to have ten sensing nodes. A multi-sink architecture is considered, with 30 nodes acting as sinks. Where used, the mobility model will be random waypoint. The simulation parameters are listed in Table 2.

**Table 2.** Simulation parameters

| Parameter | Default | Minimum | Maximum |
|---|---|---|---|
| Coverage Area (m) | | 3000*3000*3000 | |
| Data Haul Nodes | 20 | 5 | 50 |
| Traffic (kbps) | 25 | 10 | 100 |
| Mobility (ms-1) | Static | 0.3 | 3 |
| Mobility | Based on the data aggregation algorithm | | |
| MAC Protocol | Carrier Sense Multiple Access | | |

### 4.2. PERFORMANCE WITH VARIABLE MOBILITY

In this set of simulations, the mobility of the nodes is varied. The performance graphs of the different routing protocols in terms of PDR, latency and energy consumption are presented in Fig. 11 to Fig. 13 respectively. Even though QASR protocol has been designed for mobility, the performance is relatively poor when the mobility is low. This means that the data haul nodes cannot reach all the sensing nodes in time to aggregate the information, leading to reduced performance initially. Once data haul nodes are fast enough to reach the sensors, there is a significant improvement in the performance of QASR protocol. The fundamental behavior of the QASR protocol is the same irrespective of whether it is operating in the QASR-L, QASR-RE or QASR-SNR mode. The only difference is in the value carried in the mode field. Hence, it is noted that there is not much variation in the performance of the three variants of QASR.

The results of Fig. 11 show that although SUN has a higher packet delivery ratio than the variants of QASR protocol for static nodes, as mobility increases, PDR of QASR is about 10% higher than SUN. The variation of latency with change in mobility is shown in Fig. 12. ICRP has a higher latency (about 10% to 15%) with increased mobility, while the other protocols show comparable performance.

Fig. 13 shows the energy consumption of the protocols being compared. Since SUN and ICRP are not designed for mobile data haul nodes, their energy consumption is higher for static nodes. This is attributed to the fact that the network does not converge completely when the data haul nodes are static. On the other hand, variants of QASR protocol show a predictable linear increase in the energy consumed by the nodes as the mobility of nodes increases.

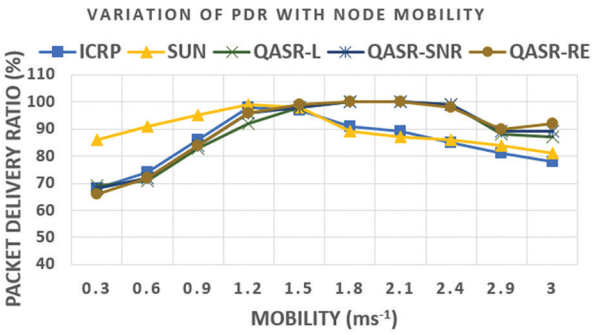



**Fig. 11.** Variation of PDR with mobility

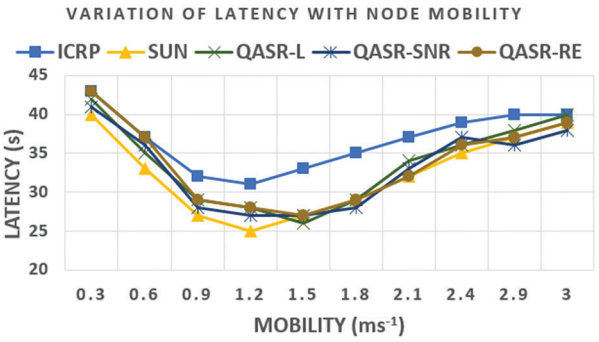



**Fig. 12.** Variation of latency with mobility

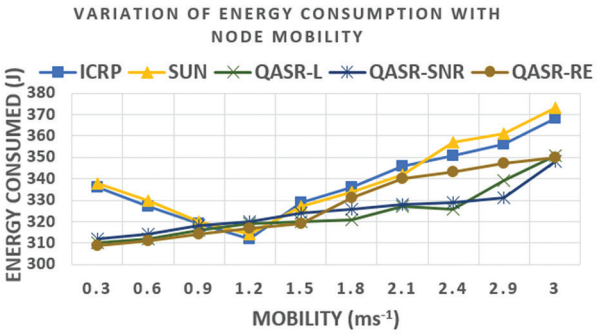



**Fig. 13.** Variation of energy consumption with mobility

### 4.3. PERFORMANCE WITH VARIABLE NODE DENSITY

This set of results is obtained by varying the density of the nodes by changing the number of clusters. For the coverage area size considered, the number of clusters is increased from 5 to 100, with each cluster containing one data haul node. Fig. 14 shows that as the number of clusters increases, the PDR of SUN and ICRP reduces drastically. In addition, there is a reduction of about 25% in the number of packets delivered. Since the data haul nodes of QASR protocol are mobile, they can accommodate larger clusters, thereby reducing overheads and increasing PDR. The results shown in Fig. 15 exhibit a similar trend, with the latency of SUN and ICRP about 150% to 200% more than QASR protocol. This increase is attributable to the increase in the overhead communication in the network because of increasing clusters. Fig. 16 shows a fairly linear increase in energy consumption of all the protocols. The increase in node density affects all the protocols in a similar fashion.

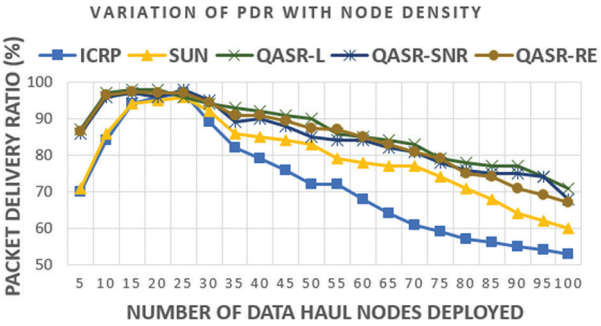



**Fig. 14.** Variation of PDR with node density

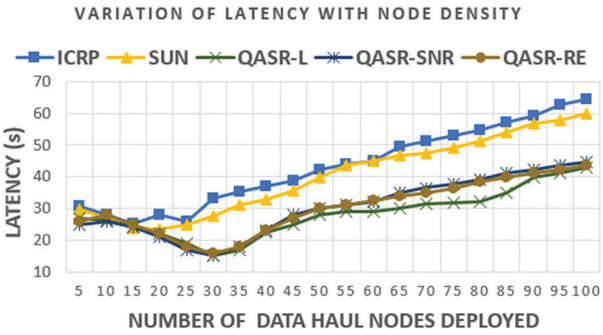



**Fig. 15.** Variation of latency with node density

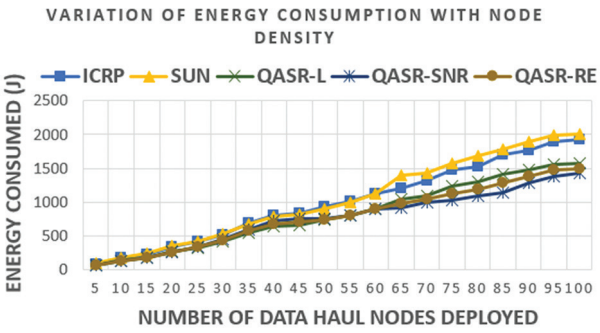



**Fig. 16.** Variation of energy consumption with node density

### 4.4. PERFORMANCE WITH VARIABLE DATA RATE

In this set of simulations, the data rate of the traffic is changed. The performance graphs of the different routing protocols in terms of PDR, latency and energy consumption are presented in Fig. 17 to Fig. 19 respectively.

Fig. 17 illustrates the deterioration of PDR with increasing data rate. While all the protocols show a comparable decline, SUN and ICRP perform comparatively worse for extremely high data rates of 90 kbps to 100 kbps. This is because of the increasingly frequent route discovery and maintenance carried out by SUN and ICRP when the data rate increases. QASR protocols shows a similar trend since it is also based on reactive routing, with the routing overhead increasing with data rate.

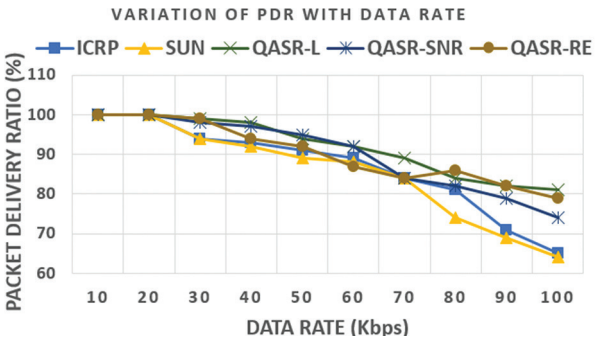



**Fig. 17.** Variation of PDR with data rate

Fig. 18 shows the effect of increasing data rate on the latency of the routing protocols. It is noted here that as the data rate increases, the latency also shows a corresponding increase of about 20%, with ICRP showing the maximum latency among the protocols. The latency values of QASR protocol are comparable to the other protocols. Fig. 19 shows the energy consumption, which is by and large linear. In comparison with QASR protocol, SUN and ICRP show an increase of about 10% to 40% of the energy consumed, with an increasing data rate.

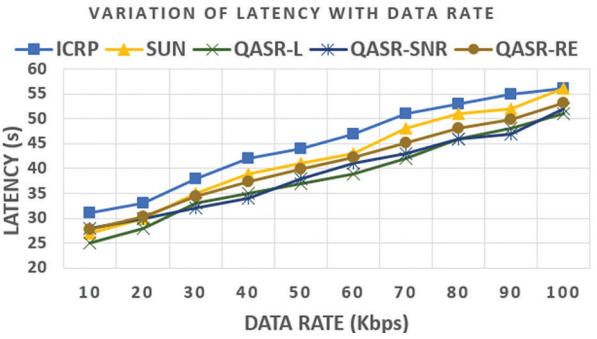



**Fig. 18.** Variation of latency with data rate

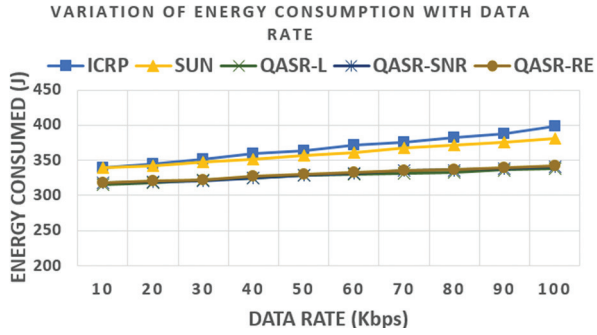



**Fig. 19.** Variation of energy consumption with data rate

## 5. CONCLUSION AND FUTURE DIRECTIONS

This paper presents the development and simulation analysis of QASR protocol, a QoS aware source routing protocol for UWSNs. QASR can choose the best routes based on QoS requirements of SNR, latency or residual energy. Depending upon the parameters based on which the routes are chosen, QASR protocol works in QASR-L, QASR-RE or QASR-SNR modes of operation. The intra-cluster data aggregation is handled by the sensing nodes' broadcasting information when the data haul node visits them. Inter-cluster communication is achieved by building an ad-hoc multi-hop network to reach the sinks.

QASR protocol is developed to be QoS aware, choosing different QoS parameters. For example, it can select between latency, residual energy and SNR. Such a provision for the choice in the selection of QoS parameters allows QASR protocol to be reconfigured while executing to consider any one of these parameters. This is a significant improvement over other QoS-based protocols optimized for one QoS parameter only.

QASR performs particularly well pertaining to latency, showing a reduction of about 10% to 20% for varying mobility and data rate. For increasing node density, the performance of SUN and ICRP deteriorate drastically by a factor of 150% to 200% compared to QASR protocol. The energy consumption of QASR protocol also deteriorates gradually, without any sudden drops and sharp variations. In some cases of increasing node density and data rate, the energy consumption of QASR protocol is about 5% to 10% lesser than SUN and ICRP. It is observed that QASR protocol exhibits performance comparable to source routing in underwater networks while simultaneously ensuring that the QoS requirements of the application are met.

The performance of the QASR protocol shows that it works fairly reliably for scenarios with varying mobility, node density and data rate. The results show that the protocol performs gracefully, without any kinks, sudden drops or unexpected behavior. Based on the various simulation studies presented in this paper, it is reasonable to infer and conclude that the QASR protocol has performed well within the generally expected and acceptable trends of performance metrics. This in turn lends QASR protocol a desirable attribute of reliability. The performance characteristics of QASR protocol are adequate to accommodate different types of traffic patterns and other application specific requirements.

It is typically observed that while designing energy efficient communication architectures for UWSNs, other QoS parameters are relegated to lesser importance. This is a valid decision since energy efficiency becomes an overarching design requirement. Therefore, the authors propose to use QASR protocol as a building block of a more generic and versatile underwater communication architecture that can be applied to scenarios requiring QoS and energy efficiency.

One of the proposed enhancements to the QASR protocol is to further optimize the process of route discovery. Currently the QASR protocol chooses the route that conforms to the QoS requirement specified during route discovery. This step involves looking at all the routes discovered and then selecting the best option among them. The authors propose to enhance route discovery such that the non-QoS compatible routes can be elimi-

nated in this phase. This will subsequently ensure that the best route that satisfies the required QoS parameter is discovered in a straightforward manner.

## 6. REFERENCES


[1] I. Akyildiz, D. Pompili, T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges", Ad Hoc Networks, Vol. 3, No. 3, 2005, pp. 257-259.

[2] S. Khisa, S. Moh, "Survey on Recent Advancements in Energy-Efficient Routing Protocols for Underwater Wireless Sensor Networks", IEEE Access, Vol. 9, 2021, pp. 55045-55062.

[3] C. Nakas, D. Kandris, G. Visvardis, "Energy Efficient Routing in Wireless Sensor Networks: A Comprehensive Survey", Algorithms, Vol. 13. No. 3, 2020, pp. 72-137.

[4] M. Khasawneh, L. Abualigah, M. Al Shinwan, "Void Aware Routing Protocols in Underwater Wireless Sensor Networks: Variants and challenges", Journal of physics: conference series, Vol. 1550, No. 3, 2020, pp. 32145-32151.

[5] J. Park, S. Lee, D. Kim, Y. Hong, "QoS-aware Directional Flooding-based Routing for Underwater Wireless Sensor Networks", Proceedings of the International Conference on Underwater Networks & Systems, Rome, Italy, 12-14 November 2014, pp. 1-2.

[6] S. Basagni, C. Petrioli, R. Petroccia, D. Spaccini, "CARP: A Channel-aware Routing Protocol for Underwater Acoustic Wireless Networks", Ad Hoc Networks, Vol. 34, 2015, pp. 92-104.

[7] C. Petrioli, R. Petroccia, J. Potter, D. Spaccini, "The SUNSET Framework for Simulation, Emulation and at-sea Testing of Underwater Wireless Sensor Networks", Ad Hoc Networks, Vol. 34, 2015, pp. 224-238.

[8] M. Jornet, J. Stojanovic, M. Zorzi, "Focused Beam Routing Protocol for Underwater Acoustic Networks", Proceedings of the Third ACM International Workshop on Underwater Networks, San Francisco, CA, USA, 15 September 2008, pp. 75-82.

[9] M. Faheem, G. Tuna, V. Gungor, "QERP: Quality-of-Service (QoS) Aware Evolutionary Routing Protocol for Underwater Wireless Sensor Networks", IEEE Systems Journal, Vol. 12, No. 3, 2018, pp. 2066-2073.

[10] H. Yan, Z. J. Shi, J. Cui, "DBR: Depth-Based Routing for Underwater Sensor Networks", Proceedings of the 7th International IFIP-TC6 Networking Conference, Singapore, 5-9 May 2008, pp. 72-86.

[11] P. Xie, J. Cui, L. Lao, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks", Proceedings of the 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, 15-19 May 2006, pp. 1216-1221.

[12] M. Faheem, R. Butt, B. Raza, H. Alquhayz, M. Ashraf, S. Shah, M. Ngadi, V. Gungor, "QoSRP: A Cross-layer QoS Channel-Aware Routing Protocol for the Internet of Underwater Acoustic Sensor Networks", Sensors, Vol. 19. No. 21, 2019, pp. 4762-4796.

[13] M. Faheem, G. Tuna, V. Gungor, "LRP: Link Quality-aware Queue-based Spectral Clustering Routing Protocol for Underwater Acoustic Sensor Networks", International Journal of Communication Systems, Vol. 30, No. 12, 2016, p. 3257.

[14] M. Faheem, M. Ngadi, V. Gungor, "Energy Efficient Multi-objective Evolutionary Routing Scheme for Reliable Data Gathering in Internet of Underwater Acoustic Sensor Networks", Ad Hoc Networks, Vol. 93. No. 3, 2019, p. 101912.

[15] A. Ali Rezaee, M. H. Zahedi, "A Priority-based Routing Algorithm for Underwater Wireless Sensor Networks (UWSNs)", Journal of Communication Engineering, Vol. 197, No. 2, 2018, pp. 61-79.

[16] R. Coutinho, A. Boukerche, L. Vieira, A. Loureiro, "GEDAR: Geographic and Opportunistic Routing Protocol with Depth Adjustment for Mobile Underwater Sensor Networks", Proceedings of the IEEE International Conference on Communications, Sydney, Australia, 10-14 June 2014, pp. 251-256.

[17] K. Latif, N. Javaid, I. Ullah, Z. Kaleem, Z. A. Malik, L. D. Nguyen, "DIEER: Delay-Intolerant Energy-Efficient Routing with Sink Mobility in Underwater Wireless Sensor Networks", Sensors, Vol. 20, No. 13, 2020, pp. 1-21.

[18] Y. Chen, Y. Lin, "Mobicast Routing Protocol for Underwater Sensor Networks", IEEE Sensors Journal, Vol. 13, No. 2, 2013, pp. 737-749.

[19] G. Toso, R. Masiero, P. Casari, "Revisiting Source Routing for Underwater Networking: The SUN Protocol", IEEE Access, Vol. 6, 2017, pp. 1525-1541.

[20] W. Liang, H. Yu, L. Liu, B. Li, C. Che, "Information-Carrying Based Routing Protocol for Underwater Acoustic Sensor Network", Proceedings of the International Conference on Mechatronics and Automation, Harbin, China, 5-8 August 2007, pp.729-734.

# Facial expression recognition via a jointly-learned dual-branch network

Original Scientific Paper

**Yamina Bordjiba**

University of Badji Mokhtar,
Faculty of Technology,
Department of Computer Science
BP 12, Annaba, Algeria
bordjiba.yamina@univ-guelma.dz

University of 8 mai 1945,
Faculty of Mathematics, computer science and sciences of matter,
Department of Computer Science, Labstic Laboratory
BP 401, Guelma, Algeria

**Hayet Farida Merouani**

University of Badji Mokhtar,
Faculty of Technology, Department of Computer Science, LRI Laboratory
BP 12, Annaba, Algeria
hayet.merouani@univ-annaba.org

**Nabiha Azizi**

University of Badji Mokhtar,
Faculty of Technology, Department of Computer Science, Labjed Laboratory
BP 12, Annaba, Algeria
azizi@labged.net

***Abstract*** *– Human emotion recognition depends on facial expressions, and essentially on the extraction of relevant features. Accurate feature extraction is generally difficult due to the influence of external interference factors and the mislabelling of some datasets, such as the Fer2013 dataset. Deep learning approaches permit an automatic and intelligent feature extraction based on the input database. But, in the case of poor database distribution or insufficient diversity of database samples, extracted features will be negatively affected. Furthermore, one of the main challenges for efficient facial feature extraction and accurate facial expression recognition is the facial expression datasets, which are usually considerably small compared to other image datasets. To solve these problems, this paper proposes a new approach based on a dual-branch convolutional neural network for facial expression recognition, which is formed by three modules: The two first ones ensure features engineering stage by two branches, and features fusion and classification are performed by the third one. In the first branch, an improved convolutional part of the VGG network is used to benefit from its known robustness, the transfer learning technique with the EfficientNet network is applied in the second branch, to improve the quality of limited training samples in datasets. Finally, and in order to improve the recognition performance, a classification decision will be made based on the fusion of both branches' feature maps. Based on the experimental results obtained on the Fer2013 and CK+ datasets, the proposed approach shows its superiority compared to several state-of-the-art results as well as using one model at a time. Those results are very competitive, especially for the CK+ dataset, for which the proposed dual branch model reaches an accuracy of 99.32, while for the FER-2013 dataset, the VGG-inspired CNN obtains an accuracy of 67.70, which is considered an acceptable accuracy, given the difficulty of the images of this dataset.*

***Keywords****: facial expression recognition, deep learning, CNN, VGGnet, transfer learning, EfficientNet, dual branch network, features fusion.*

## 1. INTRODUCTION

Based on a cross-cultural study, Ekman et al. [1] defined six basic emotional expressions: disgust, anger, fear, happiness, sadness, and surprise. Since these expressions are universal among human beings, they demonstrate that certain basic emotions are perceived in the same way among human beings, independently of their culture. While recent advanced research in neuroscience and psychology has indicated that the six basic emotions model is not universal [2], but culture-specific, most studies in the field of facial expression recognition focus on this model.

During the last decades, facial expression recognition has emerged as an important and challenging topic in several fields such as computer vision, artificial intelligence, and human-computer interaction. Generally, traditional works on facial expression recognition were conducted in two steps: first, expression features

are extracted to represent the given image/video, and then a classification stage is carried out to recognize the different expressions from the extracted features. Most conventional methods rely on handcrafted features or shallow learning, such as Neural Network [3], Bayesian Network (BN) [4], Support Vector Machine (SVM) [5], Adaboost [6], and Random Forest [7].

For conventional facial expression recognition systems, the extraction of facial features is a very crucial step, and it affects the later classification decision. Generally, it should be noted that the employed methods to extract these handcrafted features use labeled data in the context of supervised learning. In addition, these handcrafted features such as the representation of LBP and Gabor wavelets capture low-level information on facial images, except the high-level representation of facial images [8]. In addition to that, conventional approaches require relatively less computational and memory power than approaches based on deep learning. For these reasons, these approaches are still under study for use in real-time embedded systems because of their low computing complexity and high accuracy [9]. By investigating several approaches for facial expression recognition, deep neural networks generally offer better classification performance and achieve very good results in terms of accuracy in facial expression recognition compared to conventional approaches; this is due to their automatic and intelligent feature extraction. Researches show that facial expression recognition is significantly and efficiently influenced by extracted facial features. The challenge in the process of training such networks is the limit of available samples in the facial expression recognition datasets. Focusing on performance enhancement of facial expression recognition systems (FER), we propose a hybrid model combining CNN's-based extracted features to ensure complementarity and diversity, and transfer learning advantages in classification for FER applications.

Our contributions for this paper are as follows:

1. A dual-branch model based on a novel simple and efficient CNN inspired by VGGnet architecture and a pre-trained CNN, which is an efficient network is proposed to compensate for the lack of training samples, by merging their extracted features and to enhance recognition accuracy.
2. A joint training strategy is designed for the proposed dual-branch model.
3. Two datasets which are Fer-2013 and CK+ are employed to validate the effectiveness of our architectures. CK+ is a classic facial expression dataset and FER2013 provides samples of faces captured in the real world.

The rest of this document is organized as follows. Section 2 details the proposed FER approach. Section 3 presents and discusses the obtained results of the experiment. Finally, conclusions are presented in Section 4.

## 2. PROPOSED METHOD

This paper proposes a novel dual-branch system architecture for leveraging both types of CNN learning, namely learning from scratch and transfer learning, to diversify the feature maps fed to the classifier. The proposed FER system, as illustrated in Fig. 1, involves the following steps: two convolutional branches, generating each one its own feature maps. Those outputs will be combined to represent the responsible feature vector for classification step. The first branch extracts features from the input image through the convolutional layers of a VGG-inspired CNN; while the second branch extracts features from the same image through a pre-trained network. Finally, feature maps are merged using concatenation, and classification is performed using a fully connected layer with a Softmax activation function to recognize expressions. These steps will be presented in detail in the next sections.

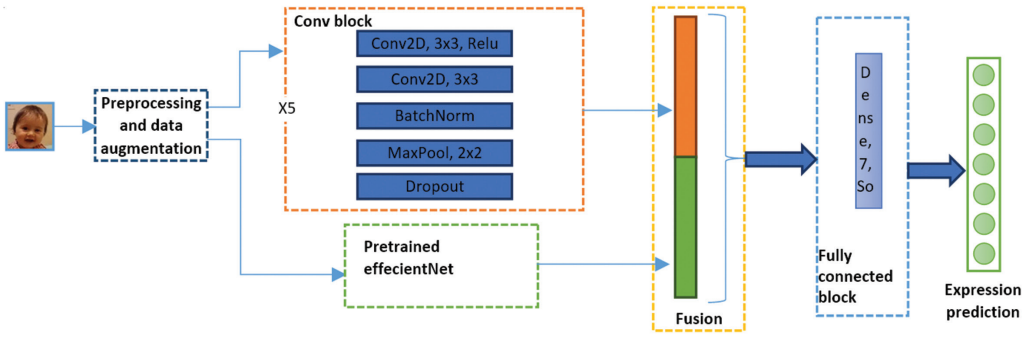



**Fig. 1.** Proposed method overview

### 2.1. VGG INSPIRED BRANCH

In 2014, Simonyan and Zisserman [10] proposed a very deep network called VGG16. In their work, they evaluate networks, increasing the depth and with very small convolution filters (3x3). This architecture won first and second place in the location and classification tracks, respectively, at the ImageNet 2014 Challenge. Based on this architecture, and after testing and experimentation with several configurations, we propose two simple and deep models, VGGinspiredCNN1 and VGGinspiredCNN2, enriched by batch normalization to improve generaliza-

tion and optimization and dropout layers. Their architecture is composed of five convolutional blocks and a fully connected block. Each convolution block is composed of two convolution layers (all used filters are 3x3 size like the VGGNet models) followed by batch normalization and a max-pooling layer (with a kernel size of 2x2) and a dropout layer. Each convolution layer is equipped with a non-linear rectification (Relu). The fully connected block of the first model is composed of two fully connected layers with 512 and 7 outputs respectively. The fully connected layer of the second model is only one layer with seven outputs. The VGG-inspired CNN architectures are described in Fig. 2 and Table 1.

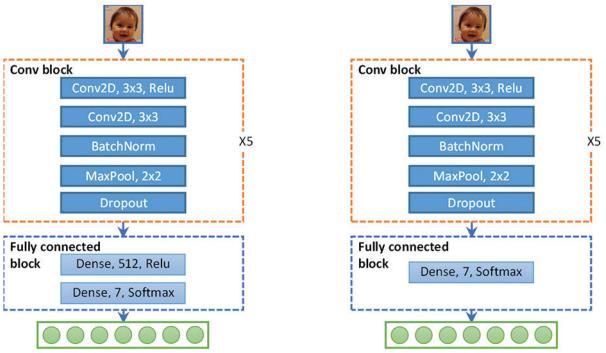



**Fig. 2.** The proposed VGGinspiredCNN architecture.

**Table 1.** The convolutional layer structure of the two proposed models inspired by VGGnet.

| | Input | Conv+Relu | Pooling | Conv+Relu | Pooling | Conv+Relu | Pooling | Conv+Relu | Pooling | Conv+Relu | Pooling |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Kernel size** | 48 x 48 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| **Stride** | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| **Pad** | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **# filters** | | 64 | | 128 | | 256 | | 512 | | 512 | |
| **#Replications** | | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |

## 2.2 PRETRAINED BRANCH

To enable CNNs to learn and extract features and achieve high accuracy, millions of samples must be used in their training base, however, existing facial expression data sets contain only just a few hundred or thousands of samples. This insufficient size is one of the main problems in CNN-based FER. To overcome this limit, the use of transfer learning will be a possible solution; it is a common practice where the network is first initialized with a set of pre-formed weights (and biases) based on a large-scale data set from one task and these parameters are then recycled to another new target task.

In order to obtain better accuracy than traditional CNNs, authors in [11] have proposed a family of models, EfficientNets, which can be systematically scaled according to available resources. A balance between network dimensions is obtained by simply scaling up them with a constant ratio. EfficientNets models transfer well to data sets such as CIFAR-100 [11], fruits [12], etc. with fewer parameters. Eight models of EfficientNetB0-EfficientNetB7 were examined for their efficiency and performance. In the transfer branch, preformed weights from the ImageNet dataset are used because it contains a large number of person images [13], about 952K images and this is very relevant for classifying the Fer-2013 and Ck+ facial expression datasets used in the evaluation. So, these pre-trained network parameters are used for initialization. Then, the model is trained, and fine-tuning will be performed to extract more specific features. Fig. 3 shows the architecture of the pre-trained CNN network.

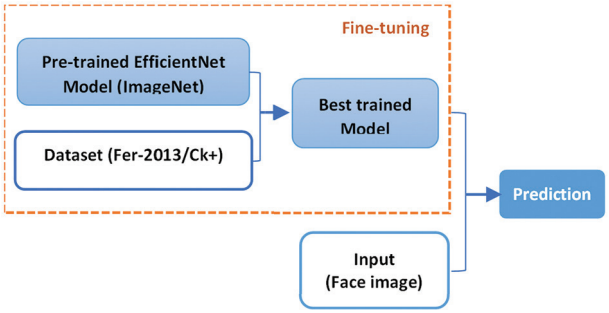



**Fig. 3.** The pre-trained CNN network architecture.

The base model EfficientNet-B0 consists of 18 convolution layers (with a kernel size of 3x3 or 5x5). Then, a flatten layer follows the max pooling as described in Table 2, its main building block is mobile inverted bottleneck MBConv, to which they also add squeeze-and-excitation optimization [11]. The other EfficientNet configurations, i.e., B1 - B7, are scaled from the basic configuration EfficientNet-B0 with different compound coefficients. A new classification layer replaces the last fully connected layers with seven classes (corresponding to seven expressions).

**Table 2.** EFFICIENTNET-B0 baseline network [11]

| Stage i | Operator Fi | Input Resolution Hi x Wi | Output Channels Ci | Layers Li |
|---|---|---|---|---|
| 1 | Conv3x3 | 224 x 224 | 32 | 1 |
| 2 | MBConv1, k3x3 | 112 x 112 | 16 | 1 |
| 3 | MBConv6, k3x3 | 112 x 112 | 24 | 2 |

| Stage i | Operator Fi | Input Resolution Hi x Wi | Output Channels Ci | Layers Li |
|---|---|---|---|---|
| 4 | MBConv6, k5x5 | 56 x 56 | 40 | 2 |
| 5 | MBConv6, k3x3 | 28 x 28 | 80 | 3 |
| 6 | MBConv6, k5x5 | 14 x 14 | 112 | 3 |
| 7 | MBConv6, k5x5 | 14 x 14 | 192 | 4 |
| 8 | MBConv6, k3x3 | 7 x 7 | 320 | 1 |
| 9 | Conv1x1 & Pooling & FC | 7 x 7 | 1280 | 1 |

### 2.3 FEATURE MAP FUSION MODULE

The feature vector concatenation is commonly used to merge and integrate multiple channels or branches in several architectures [14], [15]. The operation that combines features extracted from the VGGinspired branch and features extracted from the pre-trained model is defined as the following formula:

$$(x_1^V, x_2^V, \ldots)^T \oplus (x_1^P, x_2^P, \ldots)^T = (x_1^V, x_2^V, \ldots, x_1^P, x_2^P, \ldots)^T \quad (1)$$

Where: '$\oplus$' denotes vector concatenation operator, $(x_1^V, x_2^V, \ldots)^T$ denotes features extracted from VGGinspired branch and $(x_1^P, x_2^P, \ldots)^T$ denotes features extracted from the pretrained model.

## 3. EXPERIMENTAL RESULTS AND DISCUSSION

### 3.1 USED DATASETS AND PREPROCESSING

To demonstrate the performance of the proposed models in facial expression recognition, two widely used facial expression recognition databases, i.e., the Fer-2013 database [16] and the Cohn Kanade database [17], are used.

The Facial Emotion Recognition 2013 (FER-2013) dataset was created by Pierre Luc Carrier and Aaron Courville and was introduced in the ICML 2013 workshop's facial expression recognition challenge [16]. The data set is composed of 35887 facial images, most of them in wild environments. It consists of three parts: the original training data (OTD), which consists of 28709 images, the public test data (PTD), which includes 3589 images, and the final test data (FTD), which includes 3589 images used to score the final models.

The extended Cohn Kanade database (CK+) [17] is the most widely used laboratory-controlled database for the evaluation of FER systems. It consists of 593 video sequences obtained from 123 subjects. Among them, 327 sequences from 118 subjects are labeled as one of seven expressions. For each sequence, only the last frame is labeled. The last three frames are extracted from each sequence in the CK+ dataset, which contains 981 facial expressions. The distribution of samples of the two datasets used in experiments are shown in Table 3.

**Table 3.** Number of images per each expression in FER-2013 and CK+ datasets.

| Expression | Images number | |
|---|---|---|
| | FER-2013 | CK+ |
| anger | 4953 | 135 |
| disgust | 547 | 177 |
| fear | 5121 | 75 |
| happiness | 8989 | 207 |
| sadness | 6077 | 84 |
| surprise | 4002 | 249 |
| neutral | 6198 | / |
| contempt | / | 54 |

All face images are resized to $48 \times 48$ pixels, then normalized to have zero mean and unit variance. To make the proposed model more robust to slight transformations and noise, data augmentation is applied using different linear transformations which are: rotation, horizontal flipping, zooming, and skewing of the central area.

### 3.2. EXPERIMENT SETTINGS

In order to provide evidence of the performance of the proposed dual-branch model, three different learning experiments are conducted: the classical CNN model based on the VGG architecture, transfer learning of all EfficientNet models, and joint learning of the dual-branch model.

For training, the images from the CK+ datasets are randomly shuffled and are split as follows: 85% training, 15% test. For the FER-2013 dataset, the entire training set (28,709) and the public test set (3,589) are used for training and validation, respectively.

The total loss function is optimized during the backpropagation using the Adam optimizer, it should be noted that different optimizers were tested, even stochastic gradient descents, and Adam appeared to perform better.

The implementation is based on the Keras library [18] with TensorFlow backend [19]. OpenCV [20] is used for all image operations. All the experiments have been executed with PyTorch and trained using Google Colaboratory[21]. In light of the limitations of a free Google Colab account, such as a maximum of 12 hours per training session, the number and type of GPUs, or VRAM capacity, the training phase has been carried out using several parameters as shown in Table 4 for both datasets Fer-2013 and Ck+.

**Table 4.** Experimental configurations for FER-2013 and CK+datasets.

| | Network | Fer-2013 | | | CK+ | | |
|---|---|---|---|---|---|---|---|
| | | epochs | Batch size | Learning rate | epochs | Batch size | Learning rate |
| Learning from scratch | VGGinspiredCNN1 | 60 | 64 | 0,001 | 60 | 8 | 0,001 |
| | VGGinspiredCNN2 | 60 | 64 | 0,001 | 100 | 16 | 0,001 |
| Transfer learning | EfficientNet-B0 | 80 | 32 | 0.00001 | 80 | 16 | 0.0001 |
| | EfficientNet-B1 | 100 | 32 | 0.00001 | 80 | 16 | 0,0001 |
| | EfficientNet-B2 | 100 | 32 | 0.00001 | 80 | 8 | 0.0001 |
| | EfficientNet-B3 | 100 | 32 | 0.00001 | 80 | 8 | 0.0001 |
| | EfficientNet-B4 | 80 | 32 | 0.00001 | 80 | 8 | 0.0001 |
| | EfficientNet-B5 | 80 | 32 | 0.00001 | 80 | 16 | 0.0001 |
| | EfficientNet-B6 | 80 | 32 | 0.00001 | 80 | 16 | 0.0001 |
| | EfficientNet-B7 | 80 | 32 | 0.00001 | 80 | 16 | 0.0001 |
| Dual branch | CNN+EfficientNet-B0 | 60 | 64 | 0.001 | 120 | 8 | 0.001 |
| | CNN+EfficientNet-B1 | 60 | 64 | 0.001 | 100 | 8 | 0.0001 |
| | CNN+EfficientNet-B2 | 50 | 64 | 0.001 | 80 | 8 | 0.0001 |
| | CNN+EfficientNet-B3 | 40 | 64 | 0.0002 | 100 | 8 | 0.0001 |
| | CNN+EfficientNet-B4 | 40 | 128 | 0.0001 | 120 | 8 | 0.0001 |
| | CNN+EfficientNet-B5 | 40 | 64 | 0.0001 | 120 | 16 | 0.0001 |
| | CNN+EfficientNet-B6 | 30 | 64 | 0.0001 | 120 | 8 | 0.0001 |
| | CNN+EfficientNet-B7 | 40 | 64 | 0.0002 | 120 | 16 | 0.0001 |

### 3.3 RESULTS AND DISCUSSION

In this section, details of the achieved results are provided. As previously mentioned, experiments were conducted to determine the effectiveness of the proposed dual-branch CNN model compared to CNN by training from scratch and EfficientNet models by transfer learning. So, the two CNN models inspired by VggNet, the transfer learning of the EfficientNet with its eight configurations, and the proposed dual branch model are tested on two widely used FER datasets: CK + and FER-2013. The FER-2013 dataset includes the expressions of seven labels: anger, disgust, fear, happiness, sadness, surprise, and neutral, while the CK+ dataset includes the same expressions except the neutral expression, and also includes the expression contempt.

**1- Fer-2013dataset evaluation:**

Experimental results of the CNN models trained from scratch, the pre-trained EfficientNet models as well as the proposed dual branch model on the FER-2013 dataset are given in Fig. 4 to Fig 6 and Table 5.

Accuracy of all proposed models compared to other state-of-the-art models is listed in Table 5, and Fig. 4 to Fig. 6 show the corresponding normalized confusion matrices. For the Fer-2013 dataset, the VGG-inspired CNN models give competitive performance compared to the results of state-of-the-art models and surpass human-level accuracy, based on the accuracy rates achieved by the experimentation. As indicated by the study conducted by [22] on deep learning-based facial expression recognition, the most accurate tests on the FER-2013 dataset using a single CNN network are in the range of 67-71% and the models that accomplish it are very performant.

**Table 5.** Proposed models versus other models' performance on the Fer-2013 dataset.

| Network | | Accuracy rate |
|---|---|---|
| Learning from Scratch | VGGinspiredCNN1 | 66.71 |
| | VGGinspiredCNN2 | 67.70 |
| Transfer learning | EfficientNet-B0 | 56.23 |
| | EfficientNet-B1 | 57.48 |
| | EfficientNet-B2 | 57.43 |
| | EfficientNet-B3 | 58.32 |
| | EfficientNet-B4 | 57.13 |
| | EfficientNet-B5 | 57.70 |
| | EfficientNet-B6 | 57.17 |
| | EfficientNet-B7 | 60.10 |
| Dual Branch | CNN+EfficientNet-B0 | 63.36 |
| | CNN+EfficientNet-B1 | 62.88 |
| | CNN+EfficientNet-B2 | 62.77 |
| | CNN+EfficientNet-B3 | 63.80 |
| | CNN+EfficientNet-B4 | 62.25 |
| | CNN+EfficientNet-B5 | 62.09 |
| | CNN+EfficientNet-B6 | 62.31 |
| | CNN+EfficientNet-B7 | 62.22 |
| State of art models | [23] | 66.4 (Top-1) |
| | [25] | 66 |
| | [26] | 65.2 |
| | [24] | 71.14 |

It can be observed from the confusion matrices that the expressions 'happy' and 'surprise' are easier to recognize, with an accuracy of more than 80%, while the expression 'fear' and the expressions 'sad' and 'disgust' are the most difficult to recognize with our best model VGGinspiredCNN2, with an accuracy of 44%, 51%, and 52% respectively. It is important to be mentioned that sometimes, as a human being, it is difficult to recognize whether an expression of sadness or fear, this is due to the fact that people do not all express their emotions in the same manner, and the low accuracy rate of the expression 'disgust' is due to the small number of samples of this expression in the Fer-2013 dataset.

On the other hand, the pre-trained EfficientNet models achieved acceptable accuracy rates for this dataset, with the EfficientNet -B7 model achieving its highest accuracy of 60.10%. Note that across all EfficientNet models, the expression 'happy' is always the most recognized with the highest accuracy (more than 76%), while the expression 'disgust' is always the less recognized by the different EfficientNet models, with an accuracy between 21% and 35%, except for the EfficientNet-B7 model, where this class reaches an accuracy of 53%

**Table 5.** Proposed models versus other models' performance on the Fer-2013 dataset.

| Network | | Accuracy rate |
|---|---|---|
| Learning from Scratch | VGGinspiredCNN1 | 93.40 |
| | VGGinspiredCNN2 | 98.48 |
| Transfer learning | EfficientNet-B0 | 99.32 |
| | EfficientNet-B1 | 97.30 |
| | EfficientNet-B2 | 97.97 |
| | EfficientNet-B3 | 97.97 |
| | EfficientNet-B4 | 97.30 |
| | EfficientNet-B5 | 96.62 |
| | EfficientNet-B6 | 96.62 |
| | EfficientNet-B7 | 98.65 |
| Dual Branch | CNN+EfficientNet-B0 | 99.32 |
| | CNN+EfficientNet-B1 | 93.92 |
| | CNN+EfficientNet-B2 | 94.59 |
| | CNN+EfficientNet-B3 | 95.94 |
| | CNN+EfficientNet-B4 | 95.94 |
| | CNN+EfficientNet-B5 | 97.30 |
| | CNN+EfficientNet-B6 | 98.65 |
| | CNN+EfficientNet-B7 | 95.94 |
| State of art models | [23] | 93.2(Top-1) |
| | [27] | 72.1 |
| | [28] | 96.8 |
| | [24] | 95.29 |
| | [29] | 93.24 |

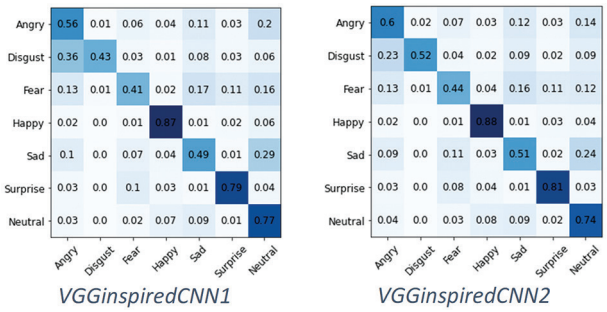



**Fig. 4.** Confusion matrices for VGGinspiredCNN models on the Fer-2013 database.

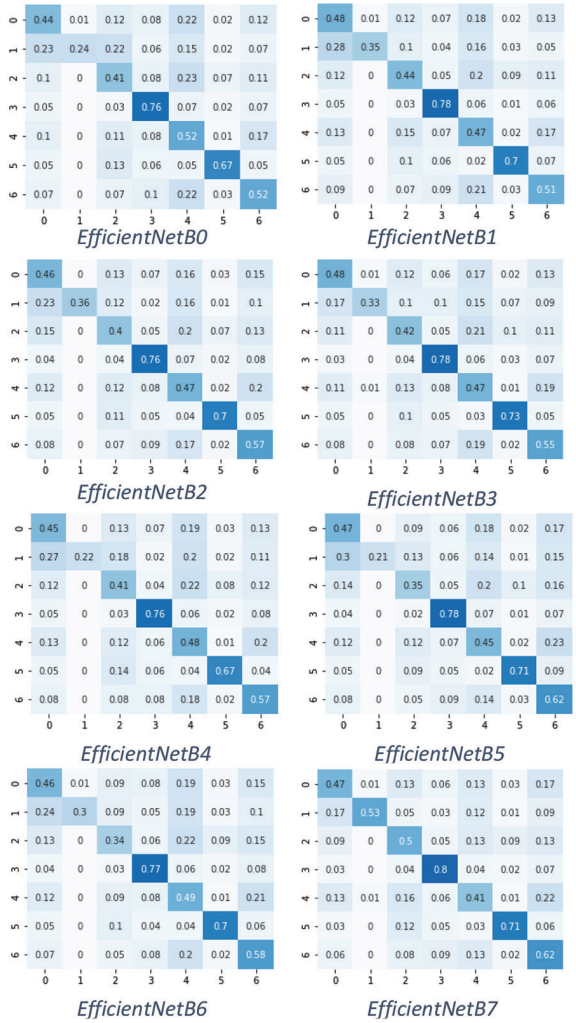



**Fig. 5.** Confusion matrices for EfficientNet models on the Fer-2013 database.

While the proposed dual-branch model provides a significant improvement for all EfficientNet configurations, especially for the EfficientNet B0 model, which gains 7.13% in accuracy, but the best performance is achieved by the EfficientNet-B3 model, with an accuracy of 63.80%. Confusion matrices of the dual-branch models show an improvement in the recognition rate of the expression 'disgust', which ranges from 47% to 70%, achieved by the best dual-branch model with EfficientNet-B3 for the Fer-2013 dataset, while no improvement is recorded on the recognition rate of the expression 'happy'.

Certainly, the proposed dual-branch model has improved the results for this dataset, however, Table 5 reveals also that the achieved recognition accuracy is not performing as well as the state-of-the-art models [43, 45] which were designed specifically for unconstrained facial expression recognition. In the proposed dual branch model, only one fully connected layer is utilized in order to obtain an efficient network, thus limiting its performance when handling the unconstrained FER task.

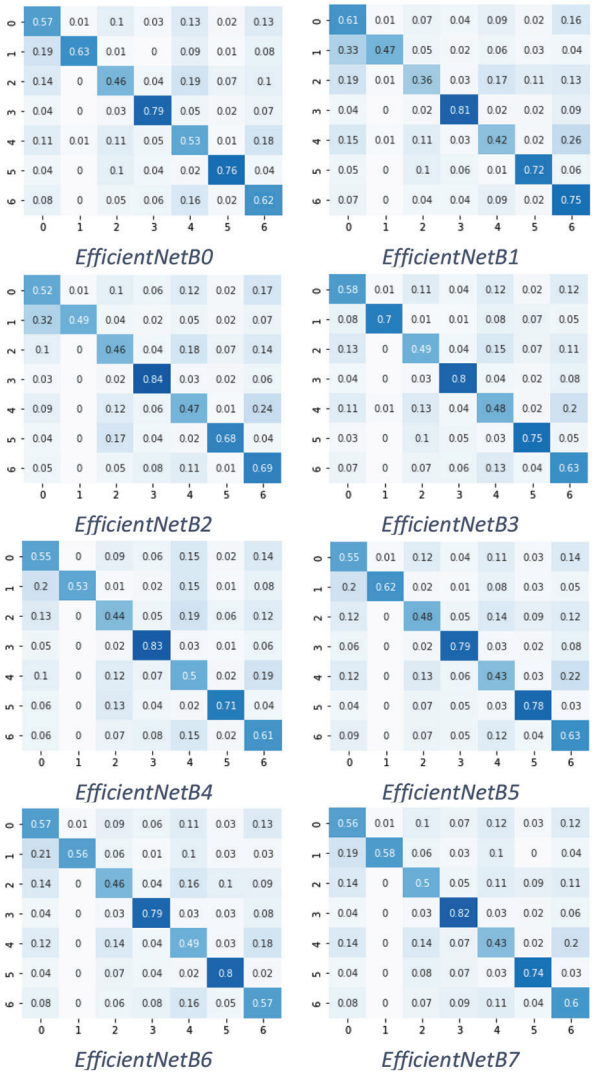



**Fig. 6.** Confusion matrices for the proposed dual branch model on Fer-2013 database.

**2 - Ck+ dataset evaluation:**

In the same way, as for the evaluation of the Fer-2013 dataset, confusion matrices for each expression and each of the proposed models on the CK+ dataset are presented in figures Fig. 7, Fig 8, and Fig.9, and the result of the comparison with other competing models is given in Table 6. CNN models inspired by VGG obtain very interesting and competitive results for the CK+ dataset, compared to the state-of-the-art models, especially, the VGGinspiredCNN2 model achieves a 98.48% accuracy rate which is better than all the reference works cited above.


**Fig. 7.** Confusion matrices for VGGinspiredCNN models on Ck+ database.

According to the normalized confusion matrix, 4 of 7 expressions ('anger', 'fear', 'happy', 'sadness') are recognized at 100%, two other expressions ('disgust' and 'surprise') are recognized at 98%, and only 'contempt' expression is recognized at 89%, which is confused with the expression 'sad'.

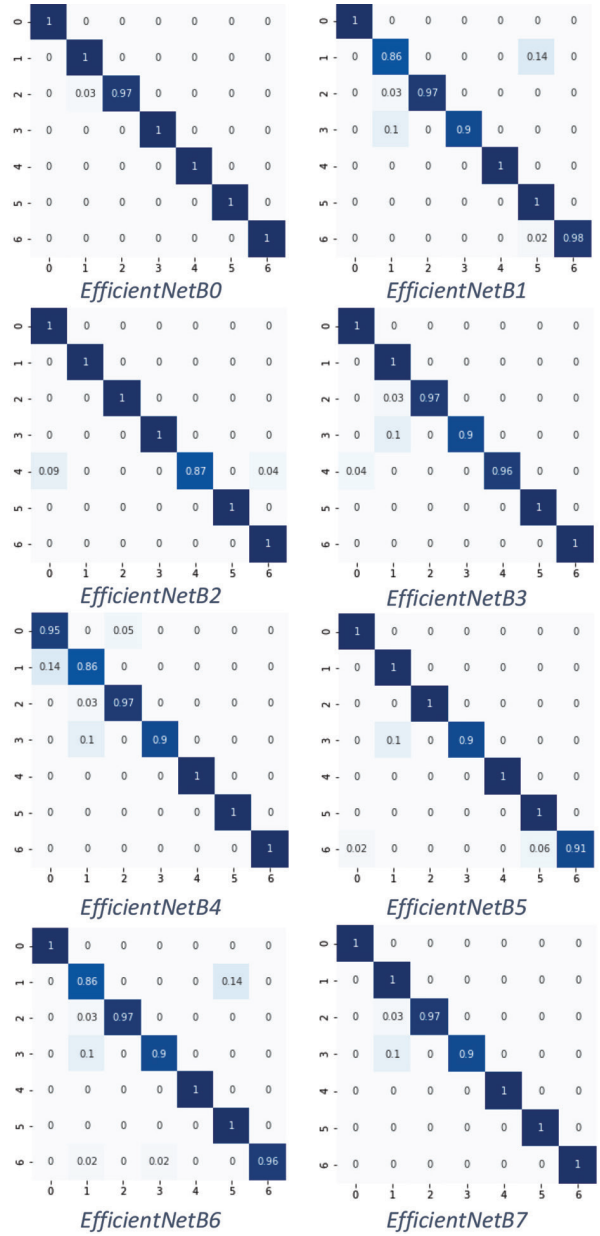



**Fig. 8.** Confusion matrices for EfficientNet models on CK+ database.

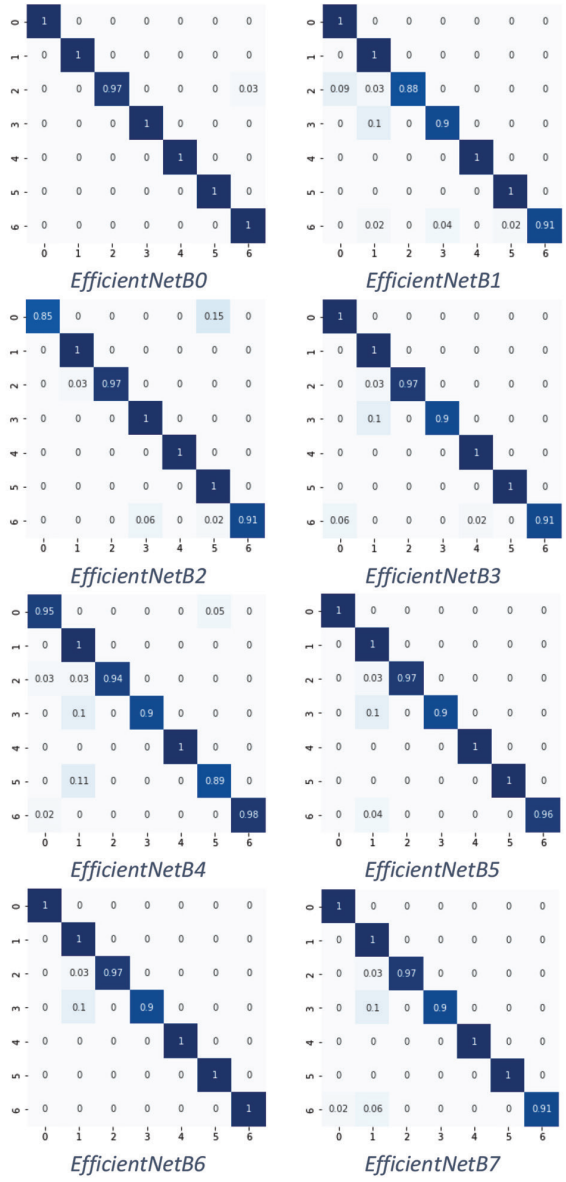



**Fig. 9.** Confusion matrices for the proposed dual branch model on the CK+ database

According to the tables, Table 5 and Table 6, comparing the results obtained by the proposed models and some reference models, the use of the dual-branch approach improves the results for the CK+ dataset, but not as much as for the Fer-2013 dataset. While the accuracy for the seven expressions in the CK+ dataset is high, the Fer2013 dataset has low classification accuracy due to mislabelling in the test set, except for the "happy" category. According to table 6, it can be seen that all EfficientNet models achieve an accuracy of more than 96%. EfficientNet-B5 and EfficientNet-B6 achieve the lowest accuracy of 96.62%, while EfficientNet-B0 realizes the best performance with an accuracy of 99.32%. The same accuracy rate is obtained in the proposed dual branch model.

Nevertheless, the Fer-2013 dataset is the most commonly used dataset for facial expression recognition. It should be noted that the human eye can hardly distinguish the appropriate emotion for some of them.

## 4. CONCLUSION

In this paper, the proposed dual-branch model is designed to take advantage of the commonly used learning approaches like learning from scratch and transfer learning, to recognize human facial expressions in the wild and under controlled laboratory conditions. Experiments and evaluation of the models using two reference datasets, Fer-2013 and CK+, showed very interesting and motivating results for both datasets. These results are competitive and outperform existing works. The most important challenge with the FER datasets is the limit of its size, and the unbalanced images of different classes, which do not favor deep learning. To overcome this limitation, two types of learning in the same model are employed simultaneously. In this approach, the training of the EfficientNets models is refined; in fact, those models are already trained for the Imagenet dataset, on the FER-2013 and CK+ datasets, and then the feature vector obtained is concatenated to that obtained from a classical CNN based on a very well-known and robust VGG architecture.

The proposed dual-branch model improved the accuracy of all expressions except "disgust", which is a bit weak like other methods on CK+. In the case of Fer-2013, the proposed dual branch model has a significant improvement in "disgust", which improves the accuracy to 70%. This demonstrates the efficiency and effectiveness of the proposed approach.

In future work, and to surmount some limitations of the proposed system, an investigation based generative adversarial network for data augmentation will be done; to improve the outcome of transfer learning, a two-step fine-tuning approach will be studied.

## 5. REFERENCES:

[1] P. Ekman, W. V. Friesen, "Constants across cultures in the face and emotion", Journal of personality and social psychology, Vol. 17, No. 2, 1971, pp. 124-129.

[2] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara, P. G. Schyns, "Reply to Sauter and Eisner: Differences outweigh commonalities in the communication of emotions across human cultures", Proceedings of the National Academy of Sciences, Vol. 110, No. 3, 2013, pp. E181-E182.

[3] H. Boughrara, M. Chtourou, C. B. Amar, L. Chen, "Facial expression recognition based on a mlp neural network using constructive training algorithm", Multimedia Tools and Applications, Vol. 75, No. 2, 2016, pp. 709-731.

[4] I. Cohen, N. Sebe, F. G. Gozman, M. C. Cirelo, T. S. Huang, "Learning Bayesian network classifiers for facial expression recognition both labeled and unlabeled data", Proceedings of the IEEE Computer

Society Conference on Computer Vision and Pattern Recognition, 18-20 June 2003, Vol. 1, pp. 595-604.

[5] M. Szwoch, P. Pieniążek, "Facial emotion recognition using depth data", Proceedings of the 8th International Conference on Human System Interaction, 25-27 June 2015, pp. 271–277.

[6] Y. Wang, H. Ai, B. Wu, C. Huang, "Real time facial expression recognition with adaboost", Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004, Vol. 3, pp. 926-929.

[7] A. Dapogny, K. Bailly, S. Dubuisson, "Pairwise conditional random forests for facial expression recognition", Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7-13 December 2015, pp. 3783-3791.

[8] X. Zhao, X. Shi, S. Zhang, "Facial Expression Recognition via Deep Learning", IETE technical review, Vol. 32, No. 5, 2015, pp. 347-355.

[9] M. Suk, B. Prabhakaran, "Real-time mobile facial expression recognition system-a case study", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23-28 June 2014, pp. 132–137.

[10] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv:1409.1556v6, 2014.

[11] M. Tan, Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", arXiv:1905.11946v5, 2019.

[12] L. T. Duong, P. T. Nguyen, C. Di Sipio, D. Di Ruscio, "Automated fruit recognition using EfficientNet and MixNet", Computers and Electronics in Agriculture, Vol. 171, 2020, p. 105326.

[13] ImageNet, http://www.image-net.org/about-stats (accessed: 2020)

[14] J. A. Aghamaleki, V. A. Chenarlogh, "Multi-stream CNN for facial expression recognition in limited training data", Multimedia Tools and Applications, Vol. 78, No. 16, 2019, pp. 22861-22882.

[15] W. Zou, D. Zhang, D.-J. Lee, "A new multi-feature fusion based convolutional neural network for facial expression recognition", Applied Intelligence, Vol. 52, No 3, 2022, pp. 2918-2929.

[16] I. J. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests", proceedings of the 20th International Conference on Neural Information Processing, Daegu, Korea, 3-7 November 2013, pp. 117-124.

[17] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, San Francisco, CA, USA, 13-18 June 2010, pp. 94-101.

[18] Keras, https://github.com/keras-team/keras (accessed: 2020)

[19] M. Abadi et al. "TensorFlow: A System for Large-Scale Machine Learning", Proceedings of the 12th USENIX Conference Operating Systems Design and Implementation, Savannah, GA, USA, 2-4 November 2016, pp. 265–283.

[20] OpenCV, https://opencv.org/ (accessed: 2020)

[21] Google Colaboratory, https://colab.research.google.com/notebooks/intro.ipynb (accessed: 2020)

[22] S. Li, W. Deng, "Deep Facial Expression Recognition: A Survey", IEEE transactions on affective computing, pp. 1–1, 2020.

[23] A. Mollahosseini, D. Chan, M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks", Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, USA, 7-10 March 2016, pp. 1-10.

[24] J. Shao, Y. Qian, "Three convolutional neural network models for facial expression recognition in the wild", Neurocomputing, Vol. 355, 2019, pp. 82-92.

[25] O. Arriaga, M. Valdenegro-Toro, P. Plöger, "Real-time Convolutional Neural Networks for Emotion and Gender Classification", arXiv:1710.07557v1, 2017.

[26] P. Giannopoulos, I. Perikos, I. Hatzilygeroudis, "Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013", Advances in hybridization of intelligent methods. Springer, 2018, pp. 1-16.

[27] R. Breuer, R. Kimmel, "A deep learning perspective on the origin of facial expressions", arXiv:1705.01842v2, 2017.

[28] H. Ding, S. K. Zhou, R. Chellappa, "FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition", Proceedings of the 12th IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 30 May-3 June 2017, pp. 118-126.

[29] D. K. Jain, P. Shamsolmoali, P. Sehdev, "Extended deep neural network for facial emotion recognition", Pattern Recognition Letters, Vol. 120, 2019, pp. 69-74.

# An improved normalized Gain-based score normalization technique for the spoof detection algorithm

Original Scientific Paper

**Ankita Chadha**
School of Computer Science,
Taylor's University,
Subang Jaya, Selangor, Malaysia 47500
chadhaankitanaresh@sd.taylors.edu.my

**Azween Abdullah**
School of Computer Science,
Taylor's University,
Subang Jaya, Selangor, Malaysia 47500
azween.abdullah@taylors.edu.my

**Lorita Angeline**
School of Computer Science,
Taylor's University,
Subang Jaya, Selangor, Malaysia 47500
lorita.angeline@taylors.edu.my

***Abstract*** *– A spoof detection algorithm supports the speaker verification system to examine the false claims by an imposter through careful analysis of input test speech. The scores are employed to categorize the genuine and spoofed samples effectively. Under the mismatch conditions, the false acceptance ratio increases and can be reduced by appropriate score normalization techniques. In this article, we are using the normalized Discounted Cumulative Gain (nDCG) norm derived from ranking the speaker's log-likelihood scores. The proposed scoring technique smoothens the decaying process due to logarithm with an added advantage from the ranking. The baseline spoof detection system employs Constant Q-Cepstral Co-efficient (CQCC) as the base features with a Gaussian Mixture Model (GMM) based classifier. The scores are computed using the ASVspoof 2019 dataset for normalized and without normalization conditions. The baseline techniques including the Zero normalization (Z-norm) and Test normalization (T-norm) are also considered. The proposed technique is found to perform better in terms of improved Equal Error Rate (EER) of 0.35 as against 0.43 for baseline system (no normalization) wrt to synthetic attacks using development data. Similarly, improvements are seen in the case of replay attack with EER of 7.83 for nDCG-norm and 9.87 with no normalization (no-norm). Furthermore, the tandem-Detection Cost Function (t-DCF) scores for synthetic attack are 0.015 for no-norm and 0.010 for proposed normalization. Additionally, for the replay attack the t-DCF scores are 0.195 for no-norm and 0.17 proposed normalization. The system performance is satisfactory when evaluated using evaluation data with EER of 8.96 for nDCG-norm as against 9.57 with no-norm for synthetic attacks while the EER of 9.79 for nDCG-norm as against 11.04 with no-norm for replay attacks. Supporting the EER, the t-DCF for nDCG-norm is 0.1989 and for no-norm is 0.2636 for synthetic attacks; while in case of replay attacks, the t-DCF is 0.2284 for the nDCG-norm and 0.2454 for no-norm. The proposed scoring technique is found to increase spoof detection accuracy and overall accuracy of speaker verification system.*

***Keywords****: spoof detection, speaker verification, score normalization, replay attack, voice conversion, speech processing.*

## 1. INTRODUCTION

The voice of a speaker is a unique way of identifying an individual and signifies various traits of the speaker such as his pitch, pauses, breathiness, and vocal tract length. The authentication based on the voice has gained importance to secure our biometric systems such as phone banking, person identification, voice command devices, voice assistants, and many more [1]. These applications require Automatic Speaker Verification (ASV) to detect enrolled and unknown speakers [2]. The spoof detection algorithm intends to detect the imposter attacks

on the ASV system. Hence, the aim of the spoof detection algorithm is to accurately classify the incoming speech sample as spoofed or genuine speech. These spoofing attacks may be categorized as Logical Access (LA) and Physical Access (PA) [3]. The development of a spoof detection algorithm includes feature representation, model training, and decision making as the major steps. While ASV also implicates score normalization for obtaining standardized scores which is indeed a crucial step in decision making [4]. Hence, considering score normalization for spoof detection is equally important for improvising the detection scores similar to the ASV framework [5]. In absence of score normalization, the variations are seen in the distribution of genuine and spoofed scores for more than one model. This happens for every speaker enrolled during the training. This leads to difficulty in choosing a unique threshold for all the enrolled speaker models. Moreover, a single enrolled speaker is likely to have variation in test utterance distribution due to changed environmental conditions such as acoustic variations, recording environment, language, and gender variations. Thus, developing a scoring technique that overcomes the mismatched conditions observed in the test speech is the essential for contributing to accurate detection of unknown test speech.

The elementary speaker verification is shown in Fig. 1 with two major sub-tasks: training and testing phase. During the training phase, the feature extraction represents the enrolled samples for various types of attacks along with the genuine speech utterances. The commonly used renowned features for spoof detection in ASV framework are Linear Prediction Residual [6], Glottal Flow parameters (GFP) [7], CQCC [3], Line Frequency Cepstral Coefficient (LFCC) [3], Phase based features like Modified Group Delay (MGD) [8] and Deep features [9]. These features have shown significant improvement in the EER. The speaker-specific features are then trained using appropriate machine learning algorithms such as GMM [3], Support Vector Machines (SVM) [10], and deep learning models like Recurrent Neural Networks (RNN) [11], Convolution Neural Network (CNN) [12], [13], Residual Networks [14] etc. During the testing, the unknown utterances are classified as genuine or spoofed speech using the target and imposter models.

## 2. RELATED WORK

The testing phase may include normalizing the scores by comparing the claimant score to the trained model score. The score distributions from the imposter and genuine speakers are normalized to improve the overall accuracy of the detection system under mismatched conditions [4]. The dissimilarities in scores are observed due to intra-speaker and inter-speaker variations [15]. This work focuses on using a unique score normalization technique for improving the EER and t-DCF of a spoof detection system.

The score normalization process works on a similar principle as that of the basis function in the wavelets, where we scale up or down and shift the score distributions according to individual speaker models for tuning the threshold to a single value. It is also widely used in other speech applications such as speaker recognition [16], [17] and outbreak classification [18]. The main work began in speaker recognition where Z-norm was employed to select speech segments [19]. Since, Z-norm does not consider handset variations, its variant, Handset normalization (H-norm) was also proposed [4]. Following this, the T-norm was used which was based on the test speech signals [20]. Other kinds of normalizations employed include ZT-norm [21], [22], HT-Norm [20], Cellular normalization (C-norm) [23], Symmetric normalization (S-norm) [16] and Distance normalization (D-norm) [16]. In [16], adaptive score normalization has also been proposed for speaker recognition and is found to perform equally well as the S-norm using NIST 2016 dataset. Although the importance of score normalization has proven to be evident in improving the accuracy of ASV, research in spoof detection is scarce [5]. Table 1 shows the research done in score normalization based on the dataset and area of application.

The study of these various normalization schemes yields three important conclusions as highlighted below:

- For score normalization, if prior knowledge about the speech samples is available then it may prove to be beneficial for that scoring technique. To elaborate on this, consider handset, language, and gender-related information to be available; then H-norm and HT-norm may help in boosting the performance of the speaker verification and recognition system. But HT norm also requires high computational time [15]. On the contrary, the knowledge about the handset, language and gender is less likely to be known for an unknown test utterance.
- Some scoring techniques are based on speaker or imposter-centric approaches where cohorts are chosen either closer to the target speaker or imposter speaker. The selection of imposters plays a vital role in such normalization schemes [16]. Still, the prior knowledge about the imposter is rarely known and the cohort selection is incomplete without imposter information.
- The normalization techniques are generally based on the estimation of mean and variance for score distributions which include Z-norm and T-norm [24].

So far, there is no specific set of rules for selecting a score normalization technique and the dataset used for scoring is not uniform due to its application in various domains (as seen from Table 1). Based on the authors' knowledge, there is no scoring technique based on cumulative gain using the rank of the speaker and has not been used for spoof detection task. Hence in this work, the nDCG-norm is used to regulate the loglikelihood scores which show promising results with a reduction in EER and t-DCF scores. The nDCG-norm is based on the goodness of ranking as well as the cumulative accumulation of relevance of the scores.

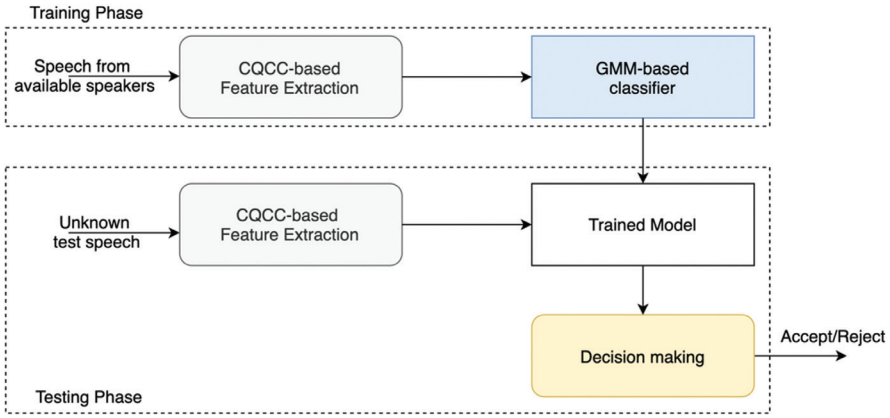



Fig. 1 Generic Speaker verification system.

**Table 1.** Research work in score normalization wrt datasets and various speech domains.

| Speech Application | Author | Year | Normalization | Datasets |
|---|---|---|---|---|
| Speaker Verification | Auckenthaler et. al [20] 2000 | T-norm | NIST 1997 | |
| | Castro et. al [25] | 2006 | Kullback-Leiber – T-norm | NIST 2005 |
| | Kenny et. al [21] | 2008 | T-norm, Z-norm, ZT-norm | NIST 2006 |
| | Villalba et. al [26] | 2011 | ZTnorm | NIST SRE 2008 |
| | Kinnunen et. al [27] | 2012 | ZT norm | NIST 2006 |
| | Kons et. al [22] | 2013 | ZT norm | WF corpus |
| | Alegre et. al [28] | 2014 | T-norm | NIST 2005 and NIST 2006 |
| | Khemiri et. al [29] | 2016 | T-norm | RSR 2015 |
| | Li and Wang [30] | 2016 | Cohort scores | CSLT- DSDB |
| | Tong et. al [31] | 2020 | Adaptive (A) scoring | CH Data, Voxceleb2 and FFSV 2020 |
| | Sahidullah et. al [32] | 2020 | AS-norm | SdSv challenge dataset |
| | Zhao et. al [33] | 2021 | S-norm | Voices 2019 |
| Speaker Recognition | Matejka et. al [16] | 2017 | S-norm | NIST 2016 |
| | Swart and Brummer [17] | 2017 | Generative scoring | RSR 2015 |
| Spoof detection - Replay Attack | Shang, Stevenson [5] | 2010 | Mean, standard deviation | Custom made |

Thus, the objectives of this work are three-fold:

I. Exploring nDCG-norm for computing normalized scores for LA and PA attacks.

II. Investigating the performance of nDCG-norm using EER, t-DCF, and Detection Error Tradeoff (DET) curve for the spoof detection task.

III. Comparing the proposed score normalization technique with baseline no normalization, state-of-the-art Z-norm, and T-norm-based scoring algorithms.

This article is structured as follows: Section 3 describes the Baseline Techniques and Section 4 includes Proposed Score Normalizing technique respectively. Section 5 discusses the Experimental setup and results of this work. Lastly, the conclusion of this work can be found in Section 6.

## 3. BASELINE TECHNIQUES

The effect of score normalization is visible on the decision accuracy. Although the literature in score normalization is two decades old, its progress is sluggish with lack of work done in spoof detection domain. Hence, there is a vivid need to explore score normalization for spoof detection task as well. The objective of the score normalizing technique is to decrease intra-speaker variations which leads to better accuracy, score calibration, and improved threshold selection. This section elaborates the baseline CQCC -GMM detection along with baseline score normalization techniques including Z-norm and T-norm as shown in Fig. 2.

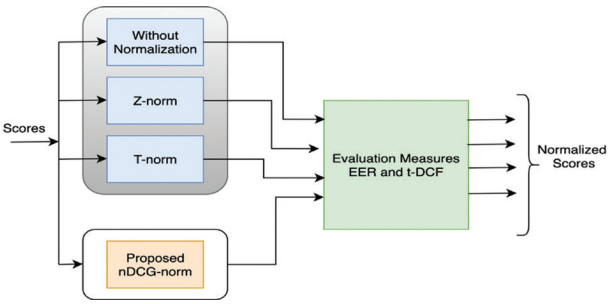



**Fig. 2.** Score Normalization Process – no normalization, Z-norm, T-norm and Proposed nDCG norm algorithm.

### 3.1 CQCC-GMM SPOOF DETECTION

The CQCC features were introduced after the ASVspoof 2015 challenge to detect the S10 attack (an attack found to be difficult to detect in the ASVspoo2015 challenge) and were also the state-of-the-art features for the ASVspoof 2019 challenge [34]. These features are based on CQT rather than discrete Fourier Transform as they promote temporal content present at higher frequencies. This is an important criterion for distinguishing genuine speech from spoofed speech.

The state-of-the-art GMM classifier is commonly used in spoof detection scheme due to its ability to perform well and capture generality in the data [35]. The task of a GMM classifier or detector is to categorize the input unknown test sample as genuine or spoofed. This is done by computing log-likelihood scores from the individual trained model – genuine speech ($\Theta_{gen}$) and spoofed speech model ($\Theta_{spoof}$). Hence, while testing the unknown test speech ($s$), the difference in log-likelihood ($l$) can be computed using equation (1).

$$Score\ (s) = \ell(\Theta_{gen}(s)) - \ell(\Theta_{\text{spoof}}(s)) \tag{1}$$

### 3.2 SCORE NORMALIZATION

The general scores are resultant of enrolled ($r$) and test speech which is denoted as score($r$,$s$). The likelihood of a speaker model $\Theta$ (speaker model consists of mixture weights) with the extracted feature set $Y=\{y_1, y_2, ..., y_m\}$ where m is the number of utterances, is given in equation (2).

$$\text{score} = \mathcal{L}(Y|\Theta) = \sum_{m=1}^{M} \log p(y_m|\Theta) \tag{2}$$

#### 3.2.1 Zero Normalization (Z-norm)

The most reliable and simplest form of normalization that is based on the estimation of mean and variance for the genuine or target speaker distribution is Z-norm [29]. The important highlight of the Z-norm is that it doesn't need to perform online permutations during the training process. The trained speaker model is compared to the subset of enrolled samples following which mean $\mu_r$ and variance $\delta_r$ are estimated. The Z-norm score $C_{z\text{-}norm}$ normalized can be computed as shown in equation (3)

$$C_{z-norm} = \frac{\log p(Y|\Theta) - \mu_r}{\delta_r} \tag{3}$$

#### 3.2.2 Test-Normalization (T-norm)

The T-norm is based on a similar principle to Z-norm except for the imposter score distribution [16]. This arrangement boosts the accurate distribution of cohort samples because of the variance and can be computed as shown in equation (4)

$$C_{t-norm} = \frac{\log p(Y|\Theta) - \mu_s}{\delta_s} \tag{4}$$

Where, $\mu_s$ and $\delta_s$ are mean and variance of imposter cohort score distribution.

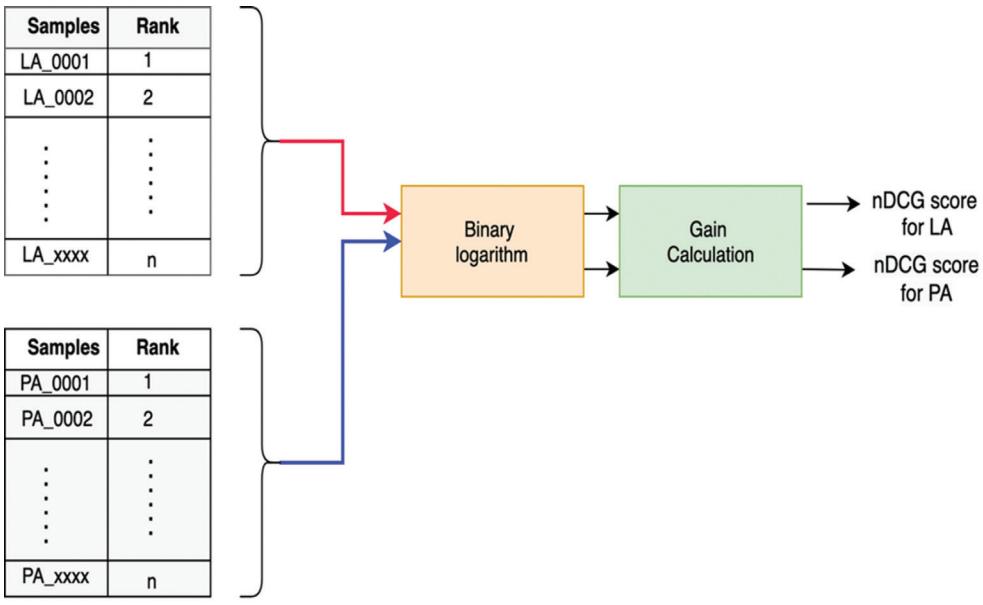



**Fig. 3** Steps to compute proposed nDCG normalization.

## 4. PROPOSED SCORE NORMALIZING TECHNIQUES

The experiments conducted in the score normalization have revealed the difficulty of various parameters that need to be considered before selecting an appropriate scoring technique. Some of these parameters are the number of speakers, the number of utterances, language dependency, handset reliability, challenges of pseudo imposters, speaker dependency, gender variations, and test data dependency which influence the performance of the scoring technique. Hence, there is a need for a more reliable scoring technique that consid-

ers variations between the training and testing phase for both known as well as unknown speakers. In this work, we propose an nDCG-norm that works on a similar principle to reduce false acceptance ratios and cumulate the score relevance through the ranking of the speaker samples. To elaborate further, Fig. 3 depicts the computation steps for the proposed scoring technique.

The scores for LA and PA attacks are normalized separately. The scores are ranked based on their degree of score value and then scaled using the binary logarithm. The nDCG-norm can be calculated as shown in equations (5) and (6) [36].

$$DCG = \sum_{i=1}^{K} \frac{N_i}{log_2(i+1)} \quad (5)$$

$$nDCG = \frac{DCG}{IDCG} \quad (6)$$

Where $K$ is the number of test samples, $N_i$ is the rank of ith sample and inverse DCG is the reverse order rank DCG of the score distribution. The nDCG-norm does not require cohort score and hence, the difficulty of choosing the cohort data is averted in contrast to state-of-the-art Z-norm and T-norm. Furthermore, the nDCG-norm when used as a part of spoof detection framework may reduce EER subsequently. So to evaluate its performance, the t-DCF score and DET are also employed.

## 5. EXPERIMENTAL SETUP AND RESULTS

The baseline ASV for developing a spoof detection system is the CQCC-GMM algorithm whose log-likelihood scores are considered for normalization. The 30 coefficient CQCC includes the delta and double delta coefficients and GMM is used as a two-class classifier with 512 components [37]. To evaluate the proposed and baseline normalization techniques, we used the ASV spoof 2019 dataset [3] which includes all three attacks including voice converted speech, text-to-speech (TTS) [36], and replay speech. For objective evaluation, the EER [39] and t-DCF are used to measure the performance of the score normalization techniques along with the DET curve [15] on the test dataset. The corpus and results are elaborated below in sub-sections.

### 5.2 ASV SPOOF 2019 CORPUS

The ASV spoof 2019 [3] corpus is adapted from the VCTK dataset [40] [41] which comprises of a separate data for LA and PA attacks. The LA dataset has synthetic speech while the PA dataset includes the replay speech. The corpus is split into three parts: training subset with 20 speakers (12 Female, 8 Male), development subset with 10 speakers (6 Female, 4 Male), and unknown speaker-based test data with nearly 48 speakers (27 Female, 21 Male). In this work, the baseline spoof detection system is trained using a training and development subset of the data and evaluated using unknown test data.

### 5.3 EXPERIMENTAL RESULTS

The spoof detection algorithm needs to be evaluated for measuring its performance and susceptibility to various attacks. This is possible through objective measures including EER and t-DCF functions. The EER measures the ratio of false positives to the false negatives and its value must be as low as possible. The t-DCF is the most important metric that calculates the error between the speaker verification system and its counter-measure or spoof detection system.

**Table 2.** EER and t-DCF for LA and PA attack for Baseline and Proposed normalization schemes.

| Type of Attack | Type of Scoring | Development Dataset | | Evaluation Dataset | |
|---|---|---|---|---|---|
| | | EER | t-DCF | EER | t-DCF |
| LA | Baseline with no normalization | 0.4311 | 0.01564 | 9.57 | 0.2366 |
| | Baseline Z-norm | 0.4302 | 0.01298 | 9.32 | 0.2298 |
| | Baseline T-norm | 0.4299 | 0.01267 | 9.15 | 0.2207 |
| | Proposed nDCG-norm | 0.3571 | 0.01037 | 8.96 | 0.1989 |
| PA | Baseline with no normalization | 9.87 | 0.1953 | 11.04 | 0.2454 |
| | Baseline Z-norm | 9.51 | 0.1921 | 10.87 | 0.2395 |
| | Baseline T-norm | 8.76 | 0.1865 | 10.66 | 0.239 |
| | Proposed nDCG-norm | 7.83 | 0.1782 | 9.79 | 0.2284 |

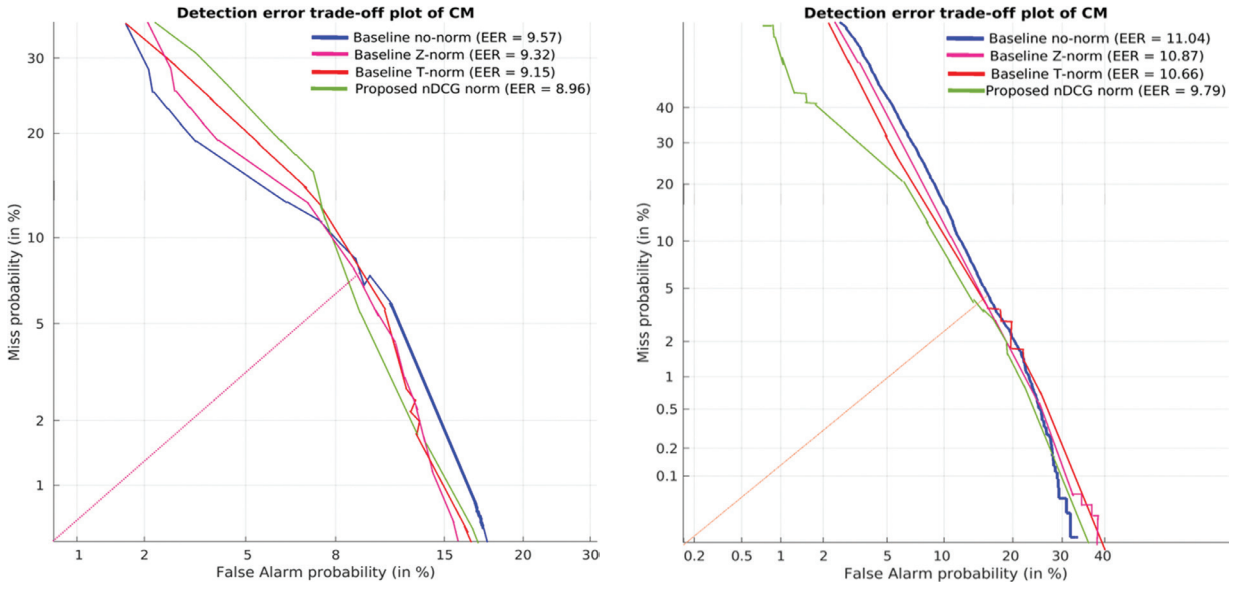
Its value must lie between 0 and 1 where 0 implies error-free between verification and counter-measure while 1 means no further improvement can be seen in spoof detection [38].

Table 2 shows EER and t-DCF scores for LA and PA attacks using development and evaluation dataset. The baseline scores include non-normalized scores, Z-norm, and T-norm scores as against the proposed nDCG-based scoring technique.

The Z-norm and T-norm scores show negligible improvements in the decision accuracy of the spoofing algorithm. This might be due to the lower number of speakers in the training data. On the contrary, the nDCG-norm does not dependent on the number of

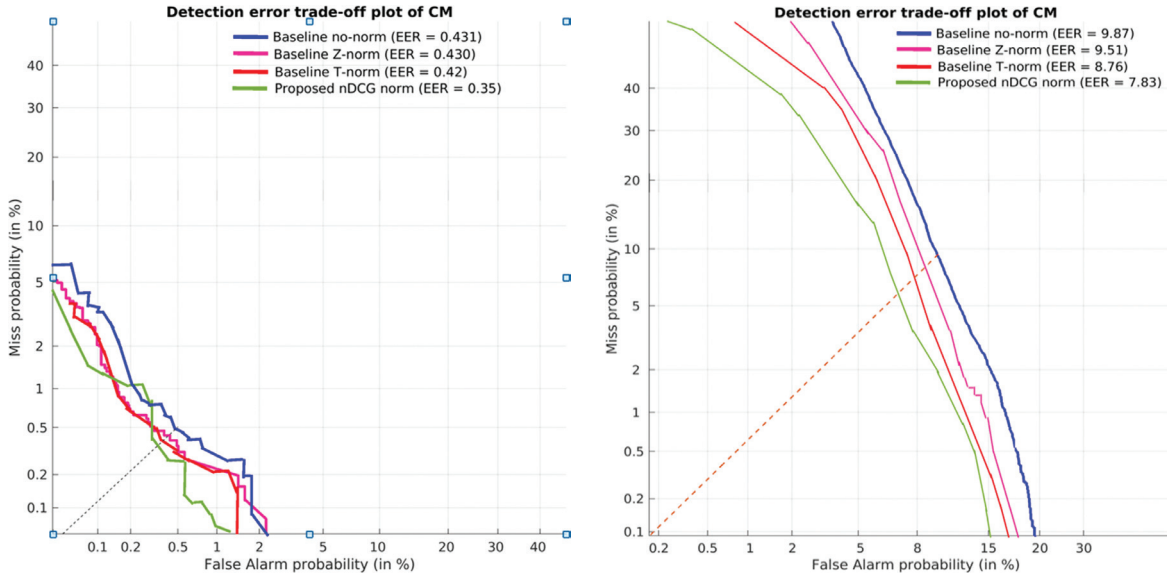
speakers. During the development stage for LA attacks, the EER for nDCG-norm is 0.35 in contrast to 0.43 with no normalization while t-DCF is 0.0103 for nDCG-norm and 0.015 for no-norm. Similarly, during the evaluation stage for LA attacks, the EER for nDCG-norm is 8.96 as against 9.57 with no-norm while t-DCF is 0.1989 for nDCG-norm and 0.2366 with no-norm. Furthermore, during development stage for PA attacks, the EER for nDCG-norm is 7.83 in contrast to 9.87 with no-norm while the t-DCF scores are 0.1782 for nDCG-norm and 0.1953 for no-norm. Similarly, during the evaluation stage for PA attacks, the EER for nDCG-norm is 9.79 as against 11.04 with no-norm while t-DCF is 0.2284 for nDCG-norm and 0.2454 with no-norm.

The nDCG-norm performs better than the system with no normalization and the baseline scoring techniques. The main reason for improvements is due to no involvement of cohort in nDCG computation; simply, the speaker-based ranking and gain computations are carried out. The selection of cohort is laborious and involves no ground rule but surely depends on the number of spoofed speakers that sound more like the genuine speakers individually. To support the above objective measures, the DET plots are used to show the relation between False-Acceptance Ratio (FAR) and False Rejection Ratio (FRR) as shown in Fig. 4 and Fig. 5 for LA and PA attacks using development and evaluation dataset respectively.




(a) (b)

**Fig. 4.** DET plot for Baseline no norm, Z-norm, T-norm and Proposed nDCG norm spoof detection system using evaluation data based on – (a) LA attacks (b) PA attacks.




(a) (b)

**Fig. 5.** DET plot for Baseline no norm, Z-norm, T-norm and Proposed nDCG norm spoof detection system using development data based on – (a) LA attacks (b) PA attacks.

In case of LA attack (Fig. 4(a) and Fig. 5(a)), the DET curves for no normalization, Z-norm and T-norm show slight variation in slope and operating point of the system i.e. the EER. The nDCG-norm shows significant improvement in lowering the false positives as compared to the other three baseline techniques. While on the other hand, for PA attack (Fig. 4(b) and Fig. 5(b) ), the slope for no-norm and Z-norm are similar, with minute variation in slope is observed for T-norm. The nDCG-norm has an improved slope implying reduced false positives ad increased in true values. Overall EER and t-DCG scores are reduced for proposed normalization as against Z-norm and T-norm score. Moreover, the normalization of scores is proven to influence the accuracy of the spoof detection system than with no normalization. It is also rightful to state that the overall ASV performance is thus improved.

## 6. CONCLUSION

The task of spoof detection is challenging yet crucial for stimulating secure environments for imposter-resistant networks including the ASV framework. The score normalization is not a compulsory but necessary step in improving the decision accuracy of the ASV. In this work, a unique score normalization technique is proposed for the spoof detection task. The proposed nDCG-norm is found to perform equally well in contrast to state-of-the-art normalization schemes. Moreover, the EER and t-DCF for all the baseline techniques are higher than the proposed scoring technique including LA and PA attacks. In the case of LA attacks, the nDCG-norm achieved an EER of 0.35 and t-DCF of 0.01 which is superior to the EER of 0.43 and t-DCF of 0.015 for the baseline technique with no normalization during the development stage. Further nDCG-norm achieved an EER of 8.96 and t-DCF of 0.198 which is superior to the EER of 9.57 and t-DCF of 0.236 for the baseline technique with no normalization during the evaluation stage. Additionally, considering PA attacks, the EER is 9.87 and t-DCF is 0.19 for no-norm in the development stage, with no major variations observed for Z-norm and T-norm; while a significant reduction in EER of 7.83 and t-DCF of 0.17 are observed for nDCG-norm. Similarly, during the evaluation stage, the EER is 11.04 and t-DCF is 0.245 for no-norm, whereas improved EER of 9.79 and t-DCF of 0.228 are obtained for proposed nDCG-norm.

The overall objective of improving accuracy by reducing the false positives is achieved by the proposed score normalization technique. Moreover, the simplicity of extraction of nDCG-norm and lower computation complexity makes it potentially viable in the post-processing stage of the spoof detection algorithm. In the future, this work can be extended for feature normalization and investigating an alternative for rank selection.

## 7. REFERENCES


[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, H. Li, "Spoofing and countermeasures for speaker verification: A survey", Speech Communication, Vol. 66, 2015, pp. 130–153.

[2] M. Sahidullah, H. Delgado, M. Todisco, T. Kinnunen, N. Evans, J. Yamagishi, K.-A. Lee, "Introduction to Voice Presentation Attack Detection and Recent Advances", Advances in Computer Vision and Pattern Recognition, 2019, pp. 321–361.

[3] J. Yamagishi et al. "ASVspoof 2019: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan", 2019, https://www.asvspoof.org/asvspoof2019/asvspoof2019_evaluation_plan.pdf (accessed: 2022)

[4] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification", Proceedings of the 5th European Conference on Speech Communication and Techology, 22-25 September 1997.

[5] W. Shang, M. Stevenson, "Score normalization in playback attack detection", Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14-19 March 2010, pp. 1678–1681.

[6] K. Phapatanaburi, L. Wang, S. Nakagawa, M. Iwahashi, "Replay Attack Detection Using Linear Prediction Analysis-Based Relative Phase Features", IEEE Access, Vol. 7, 2019, pp. 183614–183625.

[7] A. Chadha, A. Abdullah, L. Angeline, "A Unique Glottal Flow Parameters based Features for Anti-spoofing Countermeasures in Automatic Speaker Verification", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 8, 2021, pp. 827–835.

[8] I. Saratxaga, J. Sanchez, Z. Wu, I. Hernaez, E. Navas, "Synthetic speech detection using phase information", Speech Communication, Vol. 81, Jul. 2016, pp. 30–41.

[9] Y. Qian, N. Chen, K. Yu, "Deep features for automatic spoofing detection", Speech Communication, Vol. 85, 2016.

[10] B. Nasersharif, M. Yazdani, "Evolutionary fusion of classifiers trained on linear prediction based fea-

tures for replay attack detection", Expert Systems, Vol. 38, No. 3, 2021.

[11] A. Gómez Alanís, A. M. Peinado, J. A. Gonzalez, A. Gomez, "A Deep Identity Representation for Noise Robust Spoofing Detection", Proceedings of Interspeech, Hyderabad, India, 2-6 September 2018, pp. 676–680.

[12] I. Himawan, S. Madikeri, P. Motlicek, M. Cernak, S. Sridharan, C. Fookes, "Voice Presentation Attack Detection Using Convolutional Neural Networks", Advances in Computer Vision and Pattern Recognition, 2019, pp. 391–415.

[13] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, A. Larcher, "End-to-End anti-spoofing with RawNet2", Proceedings of Interspeech, 2021, pp. 6369–6373.

[14] Y. Zhang, F. Jiang, Z. Duan, "One-class Learning Towards Synthetic Voice Spoofing Detection", IEEE Signal Processing Letters, Vol. 28, 2020, pp. 937–941.

[15] F. Bimbot et al. "A Tutorial on Text-Independent Speaker Verification", EURASIP Journal on Advances in Signal Processing, Vol. 2004, No. 4, 2004, p. 101962.

[16] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, J. Černocký, "Analysis of Score Normalization in Multilingual Speaker Recognition", Proceedings of Interspeech, Stockholm, Sweden, 20-24 August 2017, pp. 1567–1571.

[17] A. Swart, N. Brümmer, "A Generative Model for Score Normalization in Speaker Recognition", Proceedings of Interspeech, Stockholm, Sweden, 20-24 August 2017, pp. 1477–1481.

[18] Z. Mustaffa, Y. Yusof, "A comparison of normalization techniques in predicting dengue outbreak", Proceedings of the International Conference on Business and Economics Research, Kuala Lumpur, Malaysia, Vol. 1, 2011.

[19] K.-P. Li, J. E. Porter, "Normalizations and selection of speech segments for speaker recognition scoring", Proceedings of the International Conference on Acoustics, Speech, Signal Processing, New York, NY, USA, 11-14 April 1988, pp. 595–598.

[20] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems", Digital Signal Processing: A Review Journal, Vol. 10, No. 1, 2000.

[21] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, P. Dumouchel, "A study of interspeaker variability in speaker verification", IEEE Transactions on Audio, Speech and Language Processing, Vol. 16, No. 5, 2008.

[22] Z. Kons, H. Aronowitz, "Voice transformation-based spoofing of text-dependent speaker verification systems", Proceedings of Interspeech, Lyon, France, 25-29 August 2013, pp. 945–949.

[23] D. A. Reynolds, "Channel robust speaker verification via feature mapping", Proceedings of the International Conference on Acoustics, Speech, Signal Processing, Vol. 2, Hong Kong, China, 6-10 April 2003, pp. II-53–6.

[24] A. Mittal, M. Dua, "Automatic speaker verification systems and spoof detection techniques: review and analysis", International Journal of Speech Technology, 2021.

[25] D. Ramos-Castro, J. Fierrez-Aguilar, J. Gonzalez-Rodriguez, J. Ortega-Garcia, "Speaker verification using speaker- and test-dependent fast score normalization", Pattern Recognition Letters, Vol. 28, No. 1, 2007, pp. 90–98.

[26] J. Villalba, E. Lleida, "Detecting replay attacks from far-field recordings on speaker verification systems", Lecture Notes in Computer Science, Vol. 6583, 2011.

[27] T. Kinnunen, Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech", IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25-30 March 2012.

[28] F. Alegre, A. Janicki, N. Evans, "Re-assessing the threat of replay spoofing attacks against automatic speaker verification", Lecture Notes in Informatics, Vol. P-230, 2014.

[29] H. Khemiri, D. Petrovska-Delacretaz, "Cohort selection for text-dependent speaker verification score normalization", Proceedings of the 2nd International Conference on Advanced Technologies for Signal and Image Processing, Monastir, Tunisia, 21-23 March 2016, pp. 689–692.

[30] L. Li, D. Wang, C. Zhang, T. F. Zheng, "Improving Short Utterance Speaker Recognition by Modeling Speech Unit Classes", IEEE/ACM Transactions on Audio, Speech, Language Processing, Vol. 24, No. 6, 2016, pp. 1129–1139.

[31] Y. Tong, W. Xue, S. Huang, L. Fan, C. Zhang, G. Ding, X. He, "The JD AI Speaker Verification System for the FFSVC 2020 Challenge", Proceedings of Interspeech, Shanghai, China, 25-29 October 2020, pp. 3476–3480.

[32] M. Sahidullah et al. "UIAI System for Short-Duration Speaker Verification Challenge 2020", Proceedings of the IEEE Spoken Language Technology Workshop, Shenzhen, China, 19-22 January 2021, pp. 323–329.

[33] L. Zhao, M.-W. Mak, "Channel Interdependence Enhanced Speaker Embeddings for Far-Field Speaker Verification", Proceedings of the 12th International Symposium on Chinese Spoken Language Processing, Hong Kong, China, 24-27 January 2021, pp. 1–5.

[34] H. Tak, J. Patino, A. Nautsch, N. Evans, M. Todisco, "An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification", Proceedings of the The Speaker and Language Recognition Workshop, Tokyo, Japan, 1-5 November 2020, pp. 333–340.

[35] M. G. Kumar, S. R. Kumar, M. S. Saranya, B. Bharathi, H. A. Murthy, "Spoof Detection Using Time-Delay Shallow Neural Network and Feature Switching", Proceedings of the Automatic Speech Recognition and Understanding Workshop, Singapore, 14-18 Dcember 2019, pp. 1011–1017.

[36] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, T. Y. Liu, "A theoretical analysis of NDCG ranking measures", Journal of Machine Learning Research, Vol. 30, 2013.

[37] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, A. Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge", Proceedings of the Annual Conference of the International Speech Communication Association, Graz, Austria, September 2019, pp. 1033–1037.

[38] S. Mo, H. Wang, P. Ren, T.-C. Chi, "Automatic Speech Verification Spoofing Detection", arXiv:2012.08095, 2020.

[39] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, B. L. Sturm, "Ensemble Models for Spoofing Detection in Automatic Speaker Verification", Proceedings of the Annual Conference of the International Speech Communication Association, September 2019, pp. 1018–1022.

[40] C. Veaux, J. Yamagishi, K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit", The Centre for Speech Technology Research, 2016, https://datashare.ed.ac.uk/handle/10283/3443 (accessed: 2022)

[41] J. Yamagishi et al. "ASVspoof 2021 : accelerating progress in spoofed and deepfake speech detection", Proceedings of the ASVspoof 2021 Workshop - Automatic Speaker Verification and Spoofing Coutermeasures Challenge, September 2021.

# Semiconductor Losses Calculation of a Quasi-Z-Source Inverter with Dead-Time

Original Scientific Paper


**Ivan Grgić**
University of Split,
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Ruđera Boškovića 32, Split, Croatia
igrgic00@fesb.hr

**Mateo Bašić**
University of Split,
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Ruđera Boškovića 32, Split, Croatia
mabasic@fesb.hr

**Dinko Vukadinović**
University of Split,
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Ruđera Boškovića 32, Split, Croatia
dvukad@fesb.hr

**Matija Bubalo**
University of Split,
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
Ruđera Boškovića 32, Split, Croatia
mbubalo@fesb.hr



***Abstract*** *– A quasi-Z-source inverter (qZSI) belongs to the group of single-stage boost inverters. The input dc voltage is boosted by utilizing an impedance network and so called shoot-through (ST) states. In pulse-width modulations utilized for the qZSI, the dead-time is commonly omitted. However, unintended ST states inevitably occur as a result of this action, due to the non-ideality of the switching devices, causing the unintended voltage boost of the inverter and an increase in the switching losses. Hence, the implementation of the dead-time is desirable with regard to both the controllability and efficiency of the qZSI. This paper deals with the calculation of semiconductor losses of the three-phase qZSI with implemented dead time. An algorithm available in the literature was utilized for that purpose. The algorithm in question was originally proposed and applied for the qZSI with omitted dead-time, where the occurrence of unintended, undetected ST states combined with the errors in the switching energy characteristics of the insulated gate bipolar transistor (IGBT) provided by a manufacturer led to errors in the obtained results. However, these errors were unjustifiably ascribed solely to the errors in the switching energy characteristics of the IGBT. In this paper, a new, corrected multiplication factor is experimentally determined and applied to the manufacturer-provided IGBT switching energies. The newly-determined multiplication factor is expectedly lower than the one obtained in the case of omitted dead time. The loss-calculation algorithm with the new multiplication factor was experimentally evaluated for different values of the qZSI input voltage, the duty cycle, and the switching frequency.*



***Keywords****: dead-time, loss-calculation algorithm, quasi-Z-source inverter, semiconductor losses, switching energies*


## NOMENCLATURE

| Symbol | Description |
|---|---|
| $C_f$ | output filter capacitors |
| $C_1$, $C_2$ | impedance network capacitors |
| $D_0$ | shoot-through state duty cycle |
| $D_1$ | impedance network diode |
| $e_{Dcond}$, $e_{Drr}$ | conduction energy and the reverse recovery energy of the free-wheeling diode, respectively |
| $e_{D1cond}$, $e_{D1rr}$ | conduction energy and the reverse recovery energy of the impedance network diode, respectively |
| $e_{Tcond}$ | conduction energy of the transistor |
| $e_{Toff}$ | turn-off energy of the transistor |
| $e_{Ton}$ | turn-on energy of the transistor |
| $f_L$ | frequency of the load voltage |
| $i_{ce}$ | collector current |
| $i_{D1}$ | current of the impedance network diode |
| $i_L$ | current of the impedance network inductor |
| $i_{ph}$, $I_{ph}$ | instantaneous value and the RMS value of the phase current, respectively |
| $k_{sw}$ | multiplication factor |
| $k_T$ | coefficient representing voltage dependence of transistor switching losses |
| $L_{f1}$, $L_{f2}$ | output filter inductors |

| | |
|---|---|
| $L_1$, $L_2$ | impedance network inductors |
| $m_a$ | amplitude modulation index |
| $p$ | non-ST state switching pulses of the transistor |
| $P_{Dcond}$ | conduction losses of the free-wheeling diode |
| $P_{D1cond}$ | conduction losses of the impedance network diode |
| $P_{Drr}$, $P_{D1rr}$ | reverse recovery losses of the free-wheeling diode and the impedance network diode, respectively |
| $P_{in}$ | input inverter power |
| $P_L$ | losses of the impedance network inductors |
| $P_{measured}$ | measured semiconductor losses |
| $P_{out}$ | output inverter power |
| $P_{Tcond}$ | transistor conduction losses |
| $P_{Toff}$ | turn-off losses of the transistor |
| $P_{Ton}$ | turn-on losses of the transistor |
| $R_{ac}$ | output load resistance |
| $R_{ce}$ | forward resistance of the IGBT |
| $R_D$, $R_{D1}$ | forward resistance of the free-wheeling diode and the impedance network diode, respectively |
| $R_d$ | damping resistance |
| $R_g$ | gate resistance |
| $R_{L1}$, $R_{L2}$ | parasitic resistances of the impedance network inductors |
| $ST_{signal}$ | shoot-through state signal |
| $T_j$ | junction temperature |
| $T_{sw}$ | switching period |
| $t_w$ | energy accumulation time window |
| $T_0$ | shoot-through state period |
| $V_{ac}$ | output voltage RMS value |
| $v_{ce}$ | collector-emitter voltage |
| $V_{ce,0}$ | threshold voltage of the transistor |
| $v_D$ | voltage across free-wheeling diode |
| $V_{D,0}$ | threshold voltage of the free-wheeling diode |
| $v_{D1}$ | voltage across impedance network diode |
| $V_{D1,0}$ | threshold voltage of the impedance network diode |
| $v_{in}$, $V_{in}$ | Instantaneous value and the mean value of the inverter input voltage, respectively |
| $V_{pn}$ | peak value of inverter bridge input voltage |
| $V_{ref}$ | reference voltage |
| $v_{refA/B/C}$ | reference phase voltages |
| $v_{trian}$ | carrier triangular signal |
| $\tau_d$ | dead-time of the pulse-width modulation |

## 1. INTRODUCTION

The quasi-Z-source inverter (qZSI), proposed in 2008 [1], is a single-stage inverter with boost capability. It represents a modification of an originally proposed Z-source inverter topology, ensuring continuous input current and lower voltage rating of one of the impedance-network capacitors [2]. The impedance network of the qZSI combined with the additional shoot-through (ST) switching state enables boost of the inverter input voltage. During the ST state, the inverter bridge of the qZSI is short circuited, which is forbidden in conventional voltage-source inverters. Therefore, in order to apply conventional pulse-width modulations (PWMs) for the qZSI, the injection of the ST states has to be additionally enabled.

The control of the qZSI is usually achieved by utilizing the sinusoidal PWM (SPWM) or the space-vector PWM (SVPWM), both with the injected ST states. The most common qZSI-compatible SVPWMs, presented in [3], differ by a number of the inverter legs simultaneously utilized for the ST state injection and by a number of the ST state occurrences within a single switching period. The SVPWM with regard to SPWM achieves higher ac voltage at the inverter bridge output for a given input dc voltage. However, this disadvantage of the SPWM may be overcome by injecting 1/6 of the 3rd harmonic component into the respective modulation signals. The qZSI-compatible SPWMs may be divided into two groups. The first comprises SPWMs in which the ST state duty cycle ($D_0$) is determined by the amplitude modulation index ($m_a$). The commonly utilized SPWMs in this group are the simple boost control [4], the maximum boost control [4], and the maximum constant boost control [5]. The second group [6-8] comprises SPWMs which allow the $D_0$ value to vary regardless of the $m_a$ value as long as the $D_0$ value is lower than the maximum allowed, which is, in turn, defined by the applied $m_a$ value [5]. In [6], the ST state signal is generated based on the comparison of two dc reference signals (positive and negative) with the carrier signal. In this way, the start of the ST state is unsynchronized with the start of the zero-switching PWM state. This results in the additional zero-switching state occurring between the ST state and the preceding active PWM state, which causes additional switching losses. To overcome this problem, a so-called zero-sync SPWM was proposed in [7], where the start of the ST state is synchronized with the start of the zero-switching state. In the same study, an additional unintended voltage boost was noted in the case of the SPWM with omitted dead-time, which is a consequence of the unintended ST states caused by the non-ideality of the utilized transistors. Consequently, it was in [7] proposed to implement the dead-time within the SPWM so as to eliminate the unintended ST states, resulting in the new dead-time zero-sync (DTZS) SPWM.

The power losses of the qZSI consist of the semiconductor losses and passive component losses. The latter include the inductors' losses, which may be determined as in [9, 10], and the capacitors' losses which are generally considered negligible. The calculation of the semiconductor losses represents a challenging task and many methods have been proposed with this regard [11-17]. In [11], the semiconductor losses were calculated based on the measured voltage and current waveforms of the utilized transistors and diodes. This required sensors with high frequency bandwidth due to the fast transients in the current and voltage of the semiconductor devices. On the other hand, in [12-17], the semiconductor losses were calculated based on the characteristics provided by the semiconductor device

manufacturer. In [12], the switching losses were calculated based on the switching energies determined according to the corresponding switching times and the semiconductor device current and voltage. This approach implies linear change of the current and voltage of the semiconductor device during the switching transition. Another possible approach is to utilize the switching energies characteristics [13-17]. These characteristics are defined as a function of the semiconductor current and are typically approximated by utilizing the linear fitting, whereas more accurate approximation is achieved by utilizing the cubic fitting as in [15]. The losses caused by the unintended ST states were not considered in [12-17], but it was observed in [15] that the loss-calculation error increases with the switching frequency, with the transistor switching losses taking up more than 90% of the total semiconductor losses. This was ascribed to the differences between the actual transistor switching energies and those provided by the manufacturer – determined based on the double-pulse test – which may have been caused by the differences between the test circuitry and the utilized laboratory setup, including parasitic capacitance and additional loop resistance/inductance [18]. Consequently, a multiplication factor ($k_{sw}$) was introduced for the transistor switching energies to minimize the errors between the measured and calculated losses. However, in this way, the losses caused by the unintended ST states were also ascribed to the errors in the switching energy characteristics, resulting in a presumably overestimated value of $k_{sw} = 1.530$.

This paper deals with the calculation of the semiconductor losses of the qZSI with the DTZS SPWM. The loss-calculation algorithm (LCA), originally proposed and denoted LCA2 in [15], is utilized for that purpose. A new experimentally determined $k_{sw}$ value is utilized for the transistor switching energies, following the same procedure as described in [15]. Finally, the semiconductor losses provided by the LCA with the newly-determined $k_{sw}$ are compared with the experimentally obtained values as well as with the values obtained by another, competing algorithm available in the literature.

## 2. POWER LOSSES OF THE QZSI

The stand-alone qZSI-based control system is shown in Fig. 1. A symmetrical impedance network is considered in this study, i.e. $L_1 = L_2 = L$, $C_1 = C_2 = C$, $R_{L1} = R_{L2} = R_L$. The three-phase inverter bridge is composed of six insulated-gate bipolar transistors (IGBTs) with integrated free-wheeling diodes (FWDs). The LCL filter, composed of the inductors ($L_{f1}$, $L_{f2}$), capacitors ($C_f$), and damping resistances ($R_d$), is connected to the inverter output. The qZSI supplies the three-phase resistive load ($R_{ac}$), whereas the control system maintains the required RMS value of the fundamental load phase voltage through the adjustment of $m_a$. The peak value of the inverter bridge input voltage is defined according to the qZSI input voltage ($V_{in}$) as follows [1]:

$$V_{pn} = \frac{V_{in}}{1-2D_0} = \frac{V_{in}}{1-2\dfrac{T_0}{T_{sw}}} \tag{1}$$

where $T_0$ and $T_{sw}$ represent the ST state period and the switching period, respectively, whereas $D_0$ represents the ST duty cycle.

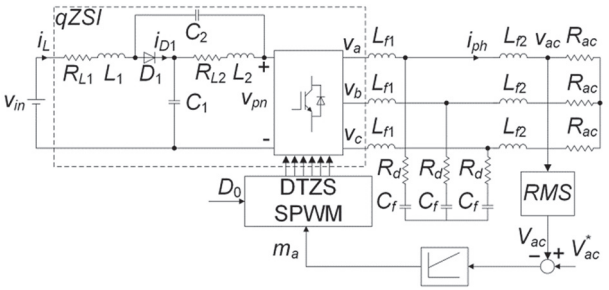



**Fig. 1.** Stand-alone qZSI-based control system [15]

Fig. 2 shows the waveforms of the reference voltages ($v_{refA}$, $v_{refB}$, $v_{refC}$), the carrier triangular signal ($v_{trian}$), the ST state signal ($ST_{signal}$), and the pulses for all the IGBTs ($S_{A+}$, $S_{A-}$, $S_{B+}$, $S_{B-}$, $S_{C+}$, $S_{C-}$) in the case of the DTZS SPWM. The letter in the subscript of "*S*" denotes the corresponding phase, whereas + and – denote the upper and the lower IGBT, respectively. The ST state occurs right at the beginning of each zero-switching state (denoted by the dashed lines). During the ST state, the pulses of all the IGBTs are set to 1. The dead-time (yellow segments) is introduced to postpone the IGBT turn-on pulse ($\tau_d = 0.7$ μs in this study). The considered PWM implied the utilization of the corresponding circuitry, details available in [7].

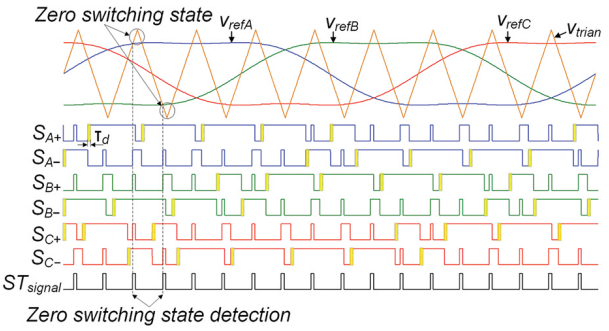



**Fig. 2.** Waveforms of the dead-time zero-sync SPWM [7]

The power losses of the qZSI represent the difference between the inverter input ($P_{in}$) and output ($P_{out}$) powers. These losses include the semiconductor losses and losses of the impedance network inductors and capacitors. The semiconductor power losses are divided into the losses of the IGBTs, the FWDs, and the impedance network diode. As for the IGBTs, the conduction, the switching, and the blocking losses exist, whereas for the diodes the conduction, the reverse recovery, the turn-on, and the reverse losses exist. Generally, the blocking losses of the IGBTs along with the turn-on and reverse losses of the diodes may be considered negligi-

ble. In this study, the semiconductor losses were calculated by utilizing the LCA described in the next section.

## 3. CALCULATION OF SEMICONDUCTOR LOSSES

The considered LCA was originally applied in [15], where it was denoted LCA2. It enables the calculation of the IGBT and FWD losses in the three-phase inverter bridge and the losses of the impedance network diode. The inverter bridge losses are determined as the losses of a single upper IGBT-FWD pair multiplied by six, based on the assumption of symmetry that holds in the case of the symmetrical output load as is the one utilized in this study. The losses are calculated based on the corresponding energies accumulated in the time window ($t_w$), as shown in the flow chart in Fig. 3. These energies are obtained from the I–V characteristics and switching energy characteristics of the IGBTs and diodes provided by the semiconductor device manufacturer. The flow chart shown in Fig. 3 may be divided into two parts. The blue-colored part is utilized for the calculation of the IGBT conduction energy accumulated during the ST states and the switching energies of the IGBT and diodes accumulated during the switching transitions between the ST state and the non-ST states. The yellow-colored part of the flow chart shown in Fig. 3 is utilized for the calculation of the energies accumulated during the non-ST states.

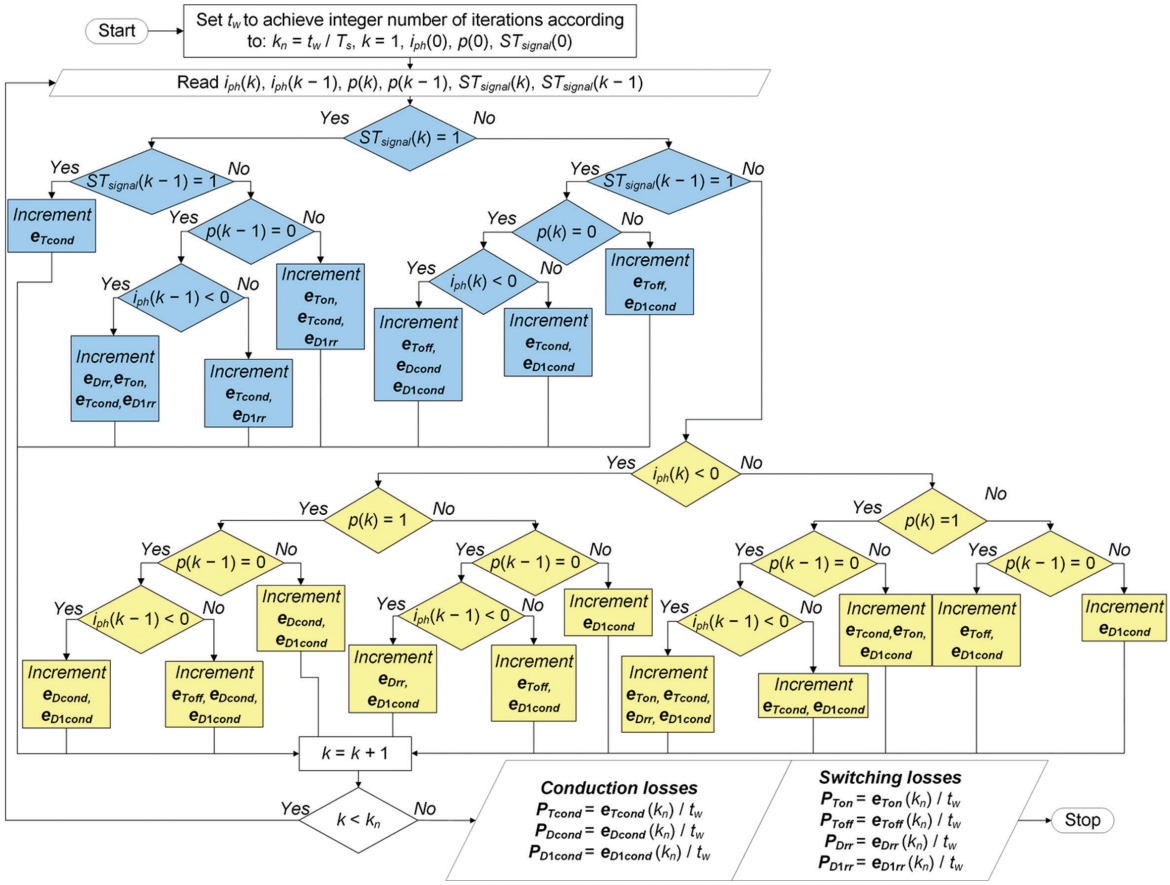



**Fig. 3.** Flow chart of the LCA [15]

That comprises the conduction and switching energies of the IGBT along with the conduction and reverse recovery energies of the FWD and impedance network diode. The LCA determines which energies should be increased by considering the instantaneous values of the following variables in $k^{th}$ and $(k-1)^{st}$ instants: the non-ST state switching pulses of the IGBT ($p$), the ST state signal ($ST_{signal}$), and the phase current ($i_{ph}$). The IGBT conduction energy ($e_{Tcond}$) is calculated based on the collector current ($i_{ce}$) and the collector-emitter voltage ($v_{ce}$), as follows:

$$e_{Tcond}(k) = e_{Tcond}(k-1) + v_{ce}(k)\left|i_{ce}(k)\right|\left[t(k) - t(k-1)\right]$$
$$v_{ce}(k) = V_{ce,0} + R_{ce}\left|i_{ce}(k)\right| \quad (2)$$

In (2), $V_{ce,0}$ and $R_{ce}$ represent the IGBT threshold voltage and the IGBT forward resistance, respectively (values given in Appendix). During the ST state ($ST_{signal} = 1$), $i_{ce} = 1/2 i_{ph} + 2/3 i_L$, where $i_L$ represents the impedance network inductor current, whereas otherwise $i_{ce} = i_{ph}$.

The conduction energy of the FWD is defined based on the diode forward voltage ($v_D$) and phase current, as follows:

$$e_{Dcond}(k) = e_{Dcond}(k-1) + v_D(k)\left|i_{ph}(k)\right|\left[t(k) - t(k-1)\right]$$
$$v_D(k) = V_{D,0} + R_D\left|i_{ph}(k)\right| \quad (3)$$

where $V_{D,0}$ and $R_D$ represent the FWD threshold voltage and the FWD forward resistance, respectively (values given in Appendix).

The conduction energy ($e_{D1cond}$) of the impedance network diode is calculated based on the corresponding threshold voltage ($V_{D1,0}$) and the forward resistance ($R_{D1}$) (values given in Appendix), as

$$\begin{aligned} e_{D1cond}(k) &= e_{D1cond}(k-1) + \\ &+ v_{D1}(k)\left|i_{D1}(k)\right|\left[t(k)-t(k-1)\right] \\ v_{D1}(k) &= V_{D1,0} + R_{D1}\left|i_{D1}(k)\right| \end{aligned} \quad (4)$$

where $v_{D1}$ and $i_{D1}$ represent the impedance network diode forward voltage and the diode current, respectively.

The IGBT switching losses are calculated based on the corresponding switching energy characteristics. The datasheet of the IGBT-FWD pair utilized in this paper contains the turn-on and turn-off switching energies of the IGBT vs. $i_{ce}$. These energies are provided for two junction temperatures ($T_j$ = 25 °C and $T_j$ = 150 °C) and for a specific reference value of the inverter bridge input voltage ($V_{ref}$ = 600 V) and the gate resistance ($R_g$ = 10 Ω). Fig. 4 shows the extraction of the characteristics that describe the IGBT turn-on energy ($e_{Ton}$) vs. $i_{ce}$ by utilizing the cubic fitting. Four coefficients $a_0$, $a_1$, $a_2$, $a_3$ (values given in Appendix) were obtained by averaging the coefficients obtained for the two provided temperatures. In this study, $R_g$ = 10 Ω was utilized, which corresponds to the value for which the datasheet characteristics were determined.

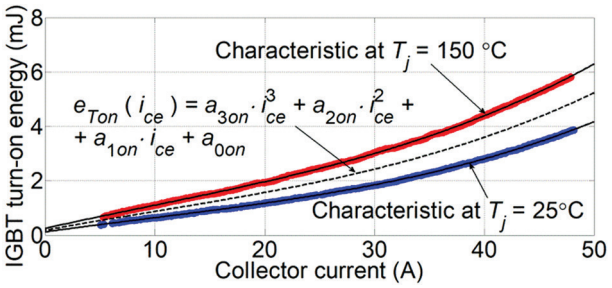



**Fig. 4.** Turn-on characteristics of the utilized IGBT [15]

The utilized inverter bridge input voltage ($V_{pn}$) differs from $V_{ref}$ in the datasheet. Therefore, the calculated IGBT turn-on energy was scaled by the ratio $(V_{pn}/V_{ref})^{k_T}$, according to the recommendations in [19], where $k_T$ is the exponent representing the voltage dependance of the IGBT switching losses ranging from 1 to 1.4. Note that the $V_{pn}$ value is determined based on the $V_{in}$ value, as per (1). Finally, by considering all the facts mentioned above, $e_{Ton}$ is obtained as follows:

$$e_{Ton}(i_{ce}) = k_{sw}\left(\frac{V_{pn}}{V_{ref}}\right)^{k_T}\left(a_3 i_{ce}^3 + a_2 i_{ce}^2 + a_1 i_{ce} + a_0\right) \quad (5)$$

In (5), $k_{sw}$ represents a multiplication factor, initially set to 1, introduced to correct the IGBT switching energies, as described in Introduction.

The IGBT turn-off characteristics were extracted in the same way as the turn-on characteristics. The corresponding polynomial is defined as follows:

$$e_{Toff}(i_{ce}) = k_{sw}\left(\frac{V_{pn}}{V_{ref}}\right)^{k_T}\left(b_3 i_{ce}^3 + b_2 i_{ce}^2 + b_1 i_{ce} + b_0\right) \quad (6)$$

where values of $b_0$, $b_1$, $b_2$, $b_3$ are given in Appendix.

The cumulative values of $e_{Ton}$ and $e_{Toff}$, obtained based on the flow chart shown in Fig. 3, are calculated as follows:

$$e_{Ton/off}(k) = e_{Ton/off}(k-1) + e_{Ton/off}\left(\left|i_{ce}(k)\right|\right) \quad (7)$$

The reverse recovery energies of the FWD ($e_{Drr}$) and the impedance network diode ($e_{D1rr}$) are calculated as follows:

$$e_{Drr}(k) = e_{Drr}(k-1) + e_{Drr}\left(\left|i_{ph}(k)\right|\right) \quad (8)$$

$$e_{D1rr}(k) = e_{D1rr}(k-1) + e_{D1rr}\left(\left|i_{D1}(k)\right|\right) \quad (9)$$

Note that, $e_{Drr}$ and $e_{D1rr}$ utilized in (8) and (9), respectively, are given in Appendix.

## 4. EXPERIMENTAL INVESTIGATION

Fig. 5 shows the laboratory setup of the stand-alone qZSI-based system. The same measurement equipment and the same sampling procedure as in [15] were utilized for the LCA implementation. Other details about the utilized experimental setup are given in [15].

The measured semiconductor losses ($P_{measured}$) were obtained as $P_{in} - P_{out} - P_L$. The input inverter ($P_{in}$) power was obtained as the mean value of $v_{in}\cdot i_L$, whereas the output inverter power ($P_{out}$) was measured by means of the power analyzer Norma 4000 (Fluke). The losses of the inductors ($P_L$) were calculated as in [9], whereas the losses of the utilized polypropylene capacitors were neglected due to the low ESR value of 7.8 mΩ

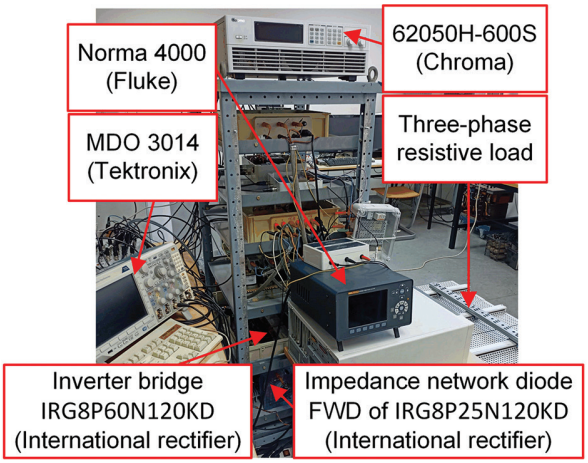



**Fig. 5.** Laboratory setup of the stand-alone qZSI-based system

### 4.1 DETERMINATION OF THE MULTIPLICATION FACTOR

The first part of the investigation was carried out to determine the new $k_{sw}$ value applicable for the qZSI

with the DTZS SPWM. For that purpose, the same procedure as in [15] was utilized. The following $I_{ph}$ values were considered: 1.33 A, 1.72 A, 2.12 A, and 2.61 A. During the measurements, $f_{sw}$ and $D_0$ were set to 5 kHz and 0.22, respectively, whereas $V_{in}$ was set to 450 V, resulting in the constant $V_{pn}/V_{ref}$ ratio with $V_{pn} \approx 800$ V. The reference RMS value ($V^*_{ac}$) and frequency ($f_L$) of the fundamental load phase voltage were set to 230 V and 50 Hz, respectively.

The IGBT switching energies were multiplied by $k_{sw}$ to annul the LCA error with respect to $P_{measured}$ for each of the considered $I_{ph}$ values. In this way, the $k_{sw}$ values in the range 1.09 – 1.34 were obtained. The final average value of $k_{sw}$ was calculated as 1.197. As expected, due to the elimination of the unintended ST states and thus the corresponding losses, the new $k_{sw}$ value is about 22% lower than the value previously obtained in [15]. The fact that the new $k_{sw}$ value is still higher than zero speaks in support of the hypothesis that the datasheet values of $e_{Ton}$ and $e_{Toff}$ may differ from the actual ones, as stated in [15]. Note that during the experimental investigation, $k_T$ in (5) and (6) was set to 1.4, which is the highest recommended value [19]. The lower $k_T$ value would result in the higher $k_{sw}$ value.

The comparison between the LCA prior and after the correction of the IGBT switching energies and the measured semiconductor losses is shown in Fig. 6 (left column).

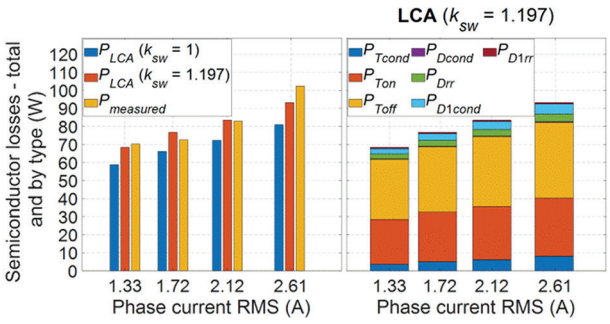



**Fig. 6.** Measured and calculated semiconductor losses with respect to RMS phase current

In comparison to the results given in [15], the absolute error of the LCA with $k_{sw}$ = 1 (i.e., before the correction) is reduced by approximately 30% solely due to the dead-time implementation. After the application of the corrected factor $k_{sw}$ = 1.197, the highest absolute error was further reduced from 22 W (21%) to 6 W (6%). Note that the error remaining after the correction is probably not related to the conduction losses calculation error, which depends on the chosen LCA sampling period. In this study, the sampling period is set to 2 µs since it has been shown in [15] that the additional reduction of the LCA sampling period does not enhance the LCA's accuracy. The introduced dead-time does not interfere with the LCA's operation because the LCA input variable $p$ represents the actual switching pulses with implemented dead-time.

The right column in Fig. 6 shows the semiconductor losses distribution of the LCA with the applied $k_{sw}$ value of 1.197. The turn-on ($P_{Ton}$) and turn-off ($P_{Toff}$) losses of the IGBTs are dominant with the share higher than 90% in the total semiconductor losses. These losses are followed by the conduction losses of the IGBTs ($P_{Tcond}$) and the conduction losses of the impedance network diode ($P_{D1cond}$) and FWDs ($P_{Dcond}$). The reverse recovery losses of FWDs ($P_{Drr}$) and the impedance network diode ($P_{D1rr}$) have the lowest share in the total semiconductor losses.

### 4.2 EXPERIMENTAL EVALUATION OF THE LCA

The second part of the investigation was carried out to evaluate the LCA with the applied $k_{sw}$ in operation ranges that result in different distribution of the semiconductor losses. The semiconductor losses were also calculated by means of another algorithm proposed in [15], where it was denoted LCA1. The corresponding IGBT switching energies were also multiplied by $k_{sw}$ = 1.197.

Fig. 7 shows the corrected calculated and measured semiconductor losses with respect to the switching frequency ($f_{sw}$), the qZSI input voltage ($V_{in}$), and the duty cycle ($D_0$).

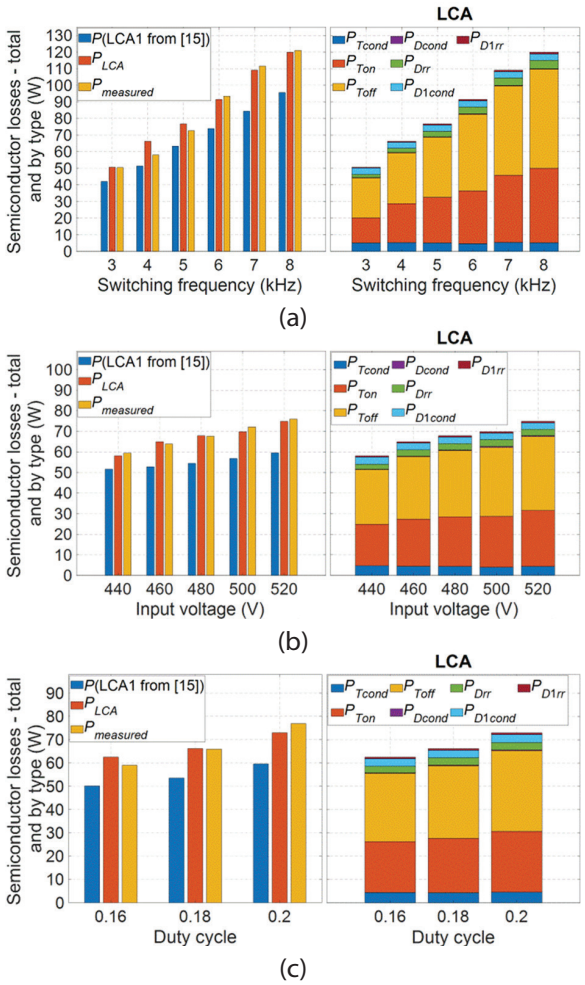



**Fig. 7.** Corrected calculated (*ksw* = 1.197) and measured semiconductor losses with respect to: switching frequency (a), input voltage (b), and duty cycle (c)

The measurements carried out for this purpose corresponded to the measurements carried out in [15]. The only difference was the utilization of the DTZS SPWM instead of the zero-sync SPWM with omitted dead-time. $V_{ac}^{*}$ and $f_L$ were set to 230 V and 50 Hz, respectively. During the variation of the switching frequency, other parameters were set to the constant values: $D_0$ = 0.22, $I_{ph}$ = 1.72 A, $V_{in}$ = 450 V. Similarly, the switching frequency was set to 5 kHz when the values of $V_{in}$ and $D_0$ were varied. The variation of $V_{in}$ was carried out with $D_0$ = 0.18 and $I_{ph}$ = 1.72 A, whereas the variation of $D_0$ was carried out with $I_{ph}$ = 1.72 A, $V_{in}$ = 470 V. Note that during the variation of $V_{in}$, the utilization of $D_0$ = 0.18 ensured the highest tolerable $V_{pn}$ value (including transients), which is lower than the maximum allowed value of 1200 V. For the same reason, $V_{in}$ was set to 470 V during the variation of $D_0$.

The left columns in Fig. 7 indicate that the accuracy of the LCA1 from [15] is lower for all the considered measurement points, with the highest noted error amounting to 25 W (23%), compared to only 9 W (12%) by the LCA. The semiconductor losses distribution of the LCA with the applied $k_{sw}$ = 1.197 is shown in Fig. 7 (right columns). This distribution corresponds to the distribution shown in Fig. 6, with dominant $P_{Ton}$ and $P_{Toff}$.

The second part of the investigation was carried out in order to evaluate the LCA over wide ranges of $V_{in}$ and $D_0$. The main aim was to consider qZSI applications where $V_{in}$ and $V_{pn}$ vary significantly, such as in the case of a photovoltaic-fed qZSI. In this part of the investigation, the $m_a$ value was set to 0.8, meaning that the output voltage was not controlled, whereas the output load resistance ($R_{ac}$) was adjusted to maintain 1 kW output power ($P_{out}$).

Fig. 8 shows the measured and calculated semiconductor losses with respect to $V_{in}$ for the $D_0$ values of 0.1, 0.15, 0.2, and 0.25.

The results shown in the left columns in Fig. 8 indicate that the semiconductor losses obtained by the LCA ($P_{LCA}$) closely correspond to the measured losses ($P_{measured}$), with the highest noted error amounting to 11 W (12%). The accuracy of the LCA1 from [15] is lower for all the measurement points, with the highest noted error amounting to 27 W (30%). These results confirm the superior accuracy of the LCA, as previously observed in [15].

The semiconductor losses distribution of the LCA depends on the $V_{in}$ value. In the case of $V_{in}$ = 200 V, the conduction losses amount to approximately 60% of the total semiconductor losses for $D_0$ = 0.1. The increase in $V_{in}$ or $D_0$ causes the increase in $V_{pn}$, as per (1), and thus the increase in the load voltage due to the constant $m_a$. Since $P_{out}$ is controlled, higher load voltage implies lower load current and thus lower current through the semiconductors. Therefore, the share of the conduction losses in the total semiconductor losses decreases with the increase in $V_{in}$ or $D_0$. On the other hand, the share of the switching losses, especially $P_{Ton}$ and $P_{Toff}$, notably increases with $V_{in}$ and $D_0$ due to the increase in $V_{pn}$, as per (1). For example, in the case of applied $V_{in}$ = 470 V and $D_0$ = 0.25 the IGBT switching losses amount to 87% of the total semiconductor losses.

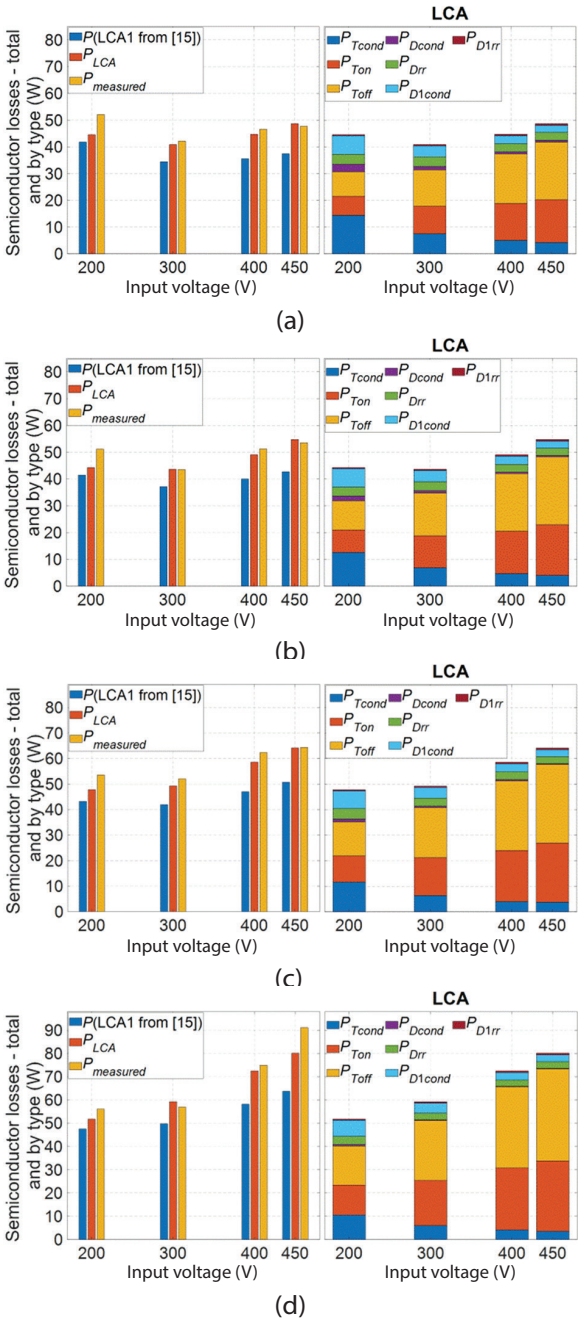



**Fig. 8.** Corrected calculated ($k_{sw}$ = 1.197) and measured semiconductor losses with respect to input voltage for duty cycles of: 0.1 (a), 0.15 (b), 0.2 (c), 0.25 (d)

## 5. CONCLUSION

In this study, the LCA available in the literature has been successfully applied for the qZSI with the DTZS SPWM. The implementation of the dead-time did not interfere with the LCA's operation, whereas it caused the decrease of the actual inverter losses. As a result, the

corrective multiplication factor for the IGBT switching energies was reduced from 1.530 (omitted dead-time) to 1.197 (implemented dead-time). However, the fact that the new multiplication factor is still different from 1 supports the hypothesis that, in some cases, the manufacturer-provided IGBT switching energies need to be adjusted. Also note that the highest recommended value was utilized for the scaling factor that describes the voltage dependence of the IGBT switching losses. Any reduction of this factor would require higher multiplication factor. Finally, the LCA was experimentally evaluated and compared to another algorithm available in the literature, with the same multiplication factor applied for both the considered algorithms. It turned out that the LCA is overall more accurate with the relative error not exceeding 12%, as opposed to the 30% obtained for the considered competing algorithm.

## 6. APPENDIX

Parameters and coefficients in (2)-(6)

$R_{ce} = 0.066105\ \Omega$, $V_{ce,0} = 0.6823\ \text{V}$, $R_D = 0.0862\ \Omega$

$V_{D,0} = 0.774\ \text{V}$, $R_{D1} = 0.1225\ \Omega$, $V_{D1,0} = 0.999\ \text{V}$

$a_0 = 0.18$, $a_1 = 0.074$, $a_2 = -7.2 \cdot 10^{-4}$, $a_3 = 2.53 \cdot 10^{-5}$

$b_0 = 0.258$, $b_1 = 0.081$, $b_2 = -1.41 \cdot 10^{-4}$, $b_3 = 0$

Reverse recovery energies in (8), (9)

$$e_{Drr}(i_{ph}) = \left(\frac{V_{pn}}{600}\right)^{0.6} \begin{pmatrix} 9.9 \cdot 10^{-7} i_{ph}^3 - 3.76 \cdot 10^{-4} i_{ph}^2 + \\ + 0.04 i_{ph} + 0.036 \end{pmatrix}$$

$$e_{D1rr}(i_{D1}) = \left(\frac{V_{pn}}{600}\right)^{0.6} \begin{pmatrix} 5.34 \cdot 10^{-6} i_{D1}^3 - 0.0012 i_{D1}^2 + \\ + 0.052 i_{D1} + 0.0145 \end{pmatrix}$$

## 7. REFERENCES:


[1] J. Anderson, F. Z. Peng: "Four quasi-Z-Source inverters", Proceedings of the 2008 IEEE Power Electronics Specialists Conference, Rhodes, Greece, pp. 2743-2749, 2008.

[2] P. Fang Zheng, "Z-source inverter", IEEE Transactions on Industry Applications, Vol. 39, No. 2, 2003, pp. 504-510.

[3] P. Liu, J. Xu, Y. Yang, H. Wang, F. Blaabjerg, "Impact of Modulation Strategies on the Reliability and Harmonics of Impedance-Source Inverters", IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol. 8, No. 4, 2020, pp. 3968-3981.

[4] P. Fang Zheng, S. Miaosen, Q. Zhaoming, "Maximum boost control of the Z-source inverter", IEEE Transactions on Power Electronics, Vol. 20, No. 4, 2005, pp. 833-838.

[5] N. Singh, S. K. Jain, "A novel strategy for indirect control of peak dc-link voltage of grid-connected qZS inverter fed through renewable energy sources", Electrical Engineering, Vol. 102, No. 2, 2020, pp. 611-625.

[6] A. Ayad, R. Kennel, "A comparison of quasi-Z-source inverters and conventional two-stage inverters for PV applications", EPE Journal, Vol. 27, No. 2, 2017, pp. 43-59.

[7] I. Grgić, D. Vukadinović, M. Bašić, M. Bubalo, "Efficiency Boost of a Quasi-Z-Source Inverter: A Novel Shoot-Through Injection Method with Dead-Time", Energies, Vol. 14, No. 14, 2021, pp. 4216.

[8] L. Monjo, L. Sainz, J. J. Mesas, J. Pedra, "State-Space Model of Quasi-Z-Source Inverter-PV Systems for Transient Dynamics Studies and Network Stability Assessment", Energies, Vol. 14, No. 14, 2021, pp. 4150.

[9] I. Grgić, M. Bubalo, D. Vukadinović, M. Bašić: "Power Losses Analysis of a Three-phase Quasi-Z-Source Inverter", Proceedings of the 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, pp. 1-5, 2020.

[10] R. Iijima, N. Kamoshida, R. A. B. Cardenas, T. Isobe, H. Tadano: "Evaluation of Inductor Losses on Z-source Inverter Considering AC and DC Components", Proceedings of the 2018 International Power Electronics Conference (IPEC-Niigata 2018 -ECCE Asia), Niigata, Japan, pp. 1111-1117, 2018.

[11] N. Kamoshida, R. Iijima, T. Isobe, H. Tadano, "Loss analysis of quasi Z-source inverter with Superjunction-MOSFET", Electrical Engineering in Japan, Vol. 205, No. 2, 2018, pp. 54-61.

[12] S. Stepenko, O. Husev, D. Vinnikov, A. Fesenko, O. Matiushkin, "Feasibility Study of Interleaving Approach for Quasi-Z-Source Inverter", Electronics, Vol. 9, No. 2, 2020, pp. 277.

[13] M. Zdanowski, D. Peftitsis, S. Piasecki, J. Rabkowski, "On the Design Process of a 6-kVA Quasi-Z-inverter Employing SiC Power Devices", IEEE Transactions on Power Electronics, Vol. 31, No. 11, 2016, pp. 7499-7508.

[14] Q. Lei, F. Z. Peng, L. He, S. Yang: "Power loss analysis of current-fed quasi-Z-source inverter", Proceedings of the 2010 IEEE Energy Conversion Congress and Exposition, pp. 2883-2887, 2010.

[15] I. Grgić, D. Vukadinović, M. Bašić, M. Bubalo, "Calculation of Semiconductor Power Losses of a Three-Phase Quasi-Z-Source Inverter", Electronics, Vol. 9, No. 10, 2020, pp. 1-19.

[16] A. Abdelhakim, P. Davari, F. Blaabjerg, P. Mattavelli, "Switching Loss Reduction in the Three-Phase Quasi-Z-Source Inverters Utilizing Modified Space Vector Modulation Strategies", IEEE Transactions on Power Electronics, Vol. 33, No. 5, 2018, pp. 4045-4060.

[17] M. Meraj, S. Rahman, A. Iqbal, N. A. Emadi, "Novel Level Shifted PWM Technique for Unequal and Equal Power Sharing in Quasi Z-Source Cascaded Multilevel Inverter for PV Systems", IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol. 9, No. 1, 2021, pp. 937-948.

[18] B. Cougo, H. Schneider, T. Meynard: "Accurate switching energy estimation of wide bandgap devices used in converters for aircraft applications", Proceedings of the 2013 15th European Conference on Power Electronics and Applications (EPE), Lille, France, pp. 1-10, 2013.

[19] Semikron: "Application Manual Power Semiconductors". 2015: ISLE Verlag: Ilmenau, Germany.

# Adaptive Sliding Mode Control Based on Fuzzy Logic and Low Pass Filter for Two-Tank Interacting System

Original Scientific Paper

**Thanh Tung Pham**
Vinh Long University of Technology Education,
Faculty of Electrical and Electronic Engineering
Nguyen Hue Street, Vinh Long City, Vietnam
tungpt@vlute.edu.vn

**Chi-Ngon Nguyen**
Can Tho University,
College of Engineering Technology
3/2 Street, Can Tho City, Vietnam
ncngon@ctu.edu.vn

***Abstract*** *– An adaptive sliding mode control (SMC) based on fuzzy logic and low pass filter is designed in this research. The SMC is one of the most widely accepted robust control techniques. However, the main disadvantage of the SMC is chattering phenomena, which inhibits its usage in many practical applications. Fuzzy logic control has supplanted conventional techniques in many applications. A major feature of fuzzy logic is the ability to express the amount of ambiguity in individual perception and human thinking. In this study, a fuzzy inference system is applied to approximate the function in the SMC law. A low pass filter is used to reduce chattering phenomena around the sliding surface. The stability of the control system is proved by the Lyapunov theory. The proposed controller is tested to position tracking control for two-tank interacting system. This system has been applied in process industries like petroleum refineries, chemical, paper industries, water treatment industries. Simulation results in MATLAB/Simulink show that the proposed algorithm is more effective than the sliding mode control, sliding mode control using conditional integrators and fuzzy control without steady-state error, the overshoot is 0 (%), the rising time achieves 2.187 (s) and the settling time is about 3.9133(s).*

***Keywords****: sliding mode control, adaptive, fuzzy logic, low pass filter, two-tank interacting*

## 1. INTRODUCTION

A sliding mode control (SMC) methodologies emerged as an effective tool to tackle uncertainty and disturbances, which are inevitable in most of the practical systems [1]. The most significant advantage of the SMC method is the ability to eliminate the effects of uncertainties caused by model errors and unwanted disturbances that affect the system response. Therefore, different and hybrid structures of the SMC method, which is known as robust control technique [2]. It is evident that real-time implementation of SMC is comparatively easier in contrast to other types of nonlinear controllers [3-5] and is applied to both linear and nonlinear systems [6, 7]. However, for the amplitude of the sliding mode control law, if not selected properly, it will cause chattering phenomenon [8, 9]. Chattering can be described as the phenomenon of finite-frequency, finite-amplitude oscillations appearing in systems with sliding mode control [8]. Chattering phenomenon due to imperfections and time delays in switching, due to small time constant actuators, power circuits are prone to overheating leading to damage [10, 11]. Many research have provided solutions to overcome chattering phenomenon in the SMC, such as a signum function of the SMC is replaced by saturation function and a Recurrent Elman Neural Network was developed for optimal determination of the switching gain of the Smoothed Sliding Mode Control was given in [8], the feed forward neural network was used [12], an adaptive terminal sliding mode control was given [13], a Smooth Hyperbolic Tangent Function was utilized to replace the discontinuous signum function [14, 15] was used a nearly optimal sliding mode controller, a robust multi-channel control system based on SMC was given [16], and an adaptive finite time robust control methods were employed [17] based on SMC method.

Liquid level control of the two-tank interacting system is widely used in the chemical industry [18, 19], paper chemical, mixing treatment industries,

pharmaceutical and food processing industries [19], nuclear power plants, and automatic liquid dispensing and replenishment devices [20]. Several researchers have investigated the problem of liquid level control for this system, such as the Fuzzy-PID Controller [18, 20, 21] was used the SMC using conditional integrators, [22] was used an intelligent self-tuning fuzzy-PID controller, the PID controller was given [23], the PID Controller and SMC were given [24] and [25] was used the Fuzzy-Optimized model reference adaptive control based on MIT and Lyapunov rules.

This study proposes to use fuzzy logic combined with a low pass filter to reduce chattering phenomenon of the control signal around the sliding surface. The aim of reducing chattering is not to lead to problems such as saturation, high energy consumption and heat for mechanical parts and also, high wear and tear of moving mechanical parts and high heat losses in electrical power circuits. The proposed controller is test to control the liquid level of the two-tank interacting system with the effects of external disturbance and changes the time constant of the filter.

The paper contribution is 1 - the SMC controller is designed to ensure that the actual liquid level position follows the desired position; 2 - the fuzzy system is used to approximate the function $f(\mathbf{x})$ in the SMC law; 3 - the low pass filter is used to reduce chattering phenomenon around the sliding surface. Simulation results in MATLAB of the proposed controller are compared with the SMC, the SMC using conditional integrators and the fuzzy controller.

This paper is organized in 5 sections: section 2 presents the mathematical model of the two-tank interacting system, the adaptive sliding mode control based on fuzzy logic and low pass filter is presented in section 3, the results and discussion are presented in the section 4 and section 5 is the conclusion.

## 2. MATHEMATICAL MODEL OF THE TWO-TANK INTERACTING SYSTEM

The model of the two-tank interacting system [25] shows in Fig. 1. Where the height of the liquid level is $h_1$ (cm) in tank 1 and $h_2$ (cm) is tank 2. Volumetric flow into tank 1 is $q_{in}$ ($cm^3$ /min), the volumetric flow rate from $q_1$ ($cm^3$ /min), and the volumetric flow rate from tank 2 is $q_2$ ($cm^3$ /min). Cross sectional area of tank 1 is $A_1$ ($cm^2$) and area of tank 2 is $A_2$ ($cm^2$).

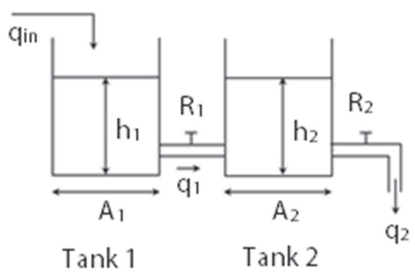



**Fig. 1.** The model of the two-tank interacting system

The state space representation of the system [25] as (1):

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\boldsymbol{x}) + \frac{a}{b} q_{in} + d(t) \end{aligned} \tag{1}$$

where:

$$f(\boldsymbol{x}) = -\frac{1}{b} x_1 - \frac{c}{b} x_2 \tag{2}$$

$a = R_2; b = T_1 T_2; c = T_1 + T_2 + A_1 R_2$; $T_2 = A_2 R_2$ and $T_1 = A_1 R_1$ are the time constant of tank 1 and 2, respectively. $x_1 = h_2$ is the actual liquid level, $x_2 = \dot{x}_1 = \dot{h}_2$ is the velocity of the liquid level and $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ is the state vector of the system; $d(t)$ is the unknown disturbance, $|d(t)| \le D$

The control reality shows that the $f(\boldsymbol{x})$ component in (2) is not easily measured. Therefore, this study is aimed at approximating $f(\boldsymbol{x})$ by a fuzzy system.

## 3. DESIGN OF ADAPTIVE SLIDING MODE CONTROLLER BASED ON FUZZY LOGIC AND LOW PASS FILTER

### 3.1. DESIGN OF SLIDING MODE CONTROLLER

The structure of the SMC controller is presented in Fig. 2.

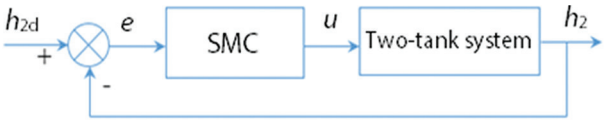



**Fig. 2.** The structure of the SMC controller

The sliding surface [10, 11] is described as (3):

$$s = ce + \dot{e} \tag{3}$$

Where $c$ must satisfy Hurwitz condition,

The tracking error and its derivative value as (4) and (5):

$$e = h_{2d} - h_2 \tag{4}$$

$$\dot{e} = \dot{h}_{2d} - \dot{h}_2 \tag{5}$$

Where $h_{2d}$ is the reference liquid level, $h_2$ is the reality position.

Taking the derivative of (3), we have (6):

$$\dot{s} = c\dot{e} + \ddot{e} = c\dot{e} + \ddot{h}_{2d} - \ddot{h}_2 \tag{6}$$

Substituting (1) into (6), we get (7):

$$\dot{s} = c\dot{e} + \ddot{h}_{2d} - \ddot{h}_2 = c\dot{e} + \ddot{h}_{2d} - f(\boldsymbol{x}) - \frac{a}{b} q_{in} - d(t) \tag{7}$$

The reaching law with constant rate [10] as (8):

$$\dot{s} = -\eta sign(s) \tag{8}$$

We get the SMC law for the two-tank interacting system as (9):

$$u_{SMC} = q_{in} = \frac{b}{a}\left[ c\dot{e} + \ddot{h}_{2d} - f(\boldsymbol{x}) + \eta sign(s) \right] \tag{9}$$

Now, (7) becomes (10):

$$\begin{aligned} \dot{s} &= c\dot{e} + \ddot{h}_{2d} - \ddot{h}_2 = c\dot{e} + \ddot{h}_{2d} - f(\boldsymbol{x}) - \frac{a}{b} q_{in} - d(t) \\ &= -\eta sign(s) - d(t) \end{aligned} \tag{10}$$

If $\eta \geq D$, we have (11):

$$s\dot{s} = -\eta |s| - sd(t) \leq 0 \tag{11}$$

In this paper, the fuzzy system will be use to approximate the function *f(**x**)* in (9).

### 3.2 APPROXIMATION USING A FUZZY SYSTEM

Using the universal approximation theorem, we will replace *f(**x**)* with the fuzzy system $\hat{f}(\boldsymbol{x})$ to realize feedback control. Three steps [10] are designed as follows:

Step 1. For $x_1$ and $x_2$, define five fuzzy sets for $A_1^{l_i}$ and $A_2^{l_i}$ respectively, $l_i = 1,2,\ldots,5$;

Step 2. Design $\prod_{i=1}^{n} p_i = p_1 \times p_2 = 25$ fuzzy rules to construct fuzzy system $\hat{f}(\boldsymbol{x}|\boldsymbol{\theta})$ as (12):

$$\begin{aligned} &R^{(1)}: \text{if } x_1 \text{ is } A_1^1 \text{ and } \ldots \text{ and } x_2 \text{ is } A_2^1 \text{ then } \hat{f} \text{ is } B^1 \\ &\vdots \\ &R^{(25)}: \text{if } x_1 \text{ is } A_1^5 \text{ and } \ldots \text{ and } x_2 \text{ is } A_2^5 \text{ then } \hat{f} \text{ is } B^{25} \end{aligned} \tag{12}$$

Where $l_i = 1,2,3,4,5; i = 1,2; p_1 = p_2 = 5$

Step 3. Using fuzzy inference, the output of fuzzy system as (13)

$$\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{\sum_{l_1=1}^{5}\sum_{l_2=1}^{5} \bar{y}_f^{l_1 l_2} \left( \prod_{i=1}^{2} \mu_{A_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{5}\sum_{l_2=1}^{5} \left( \prod_{i=1}^{2} \mu_{A_i^{l_i}}(x_i) \right)} \tag{13}$$

Where $\mu_{A_i^{l_i}}(x_i)$ is membership function of $x_i$

Let $\bar{y}_f^{l_1 l_2}$ to be freedom parameter and be put in the set $\hat{\boldsymbol{\theta}} \in \boldsymbol{R}^{(25)}$. Column vector $\boldsymbol{\xi}(\boldsymbol{x})$ is introduced and (13) can be written as (14):

$$\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}^T \boldsymbol{\xi}(\boldsymbol{x}) \tag{14}$$

Where

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \tag{15}$$

$\boldsymbol{\xi}(\boldsymbol{x})$ is $\prod_{i=1}^{n} p_i = p_1 \times p_2 = 25$ dimensional column vector, and $l_1, l_2$ elements are, respectively,

$$\boldsymbol{\xi}_{l_1 l_2}(\boldsymbol{x}) = \frac{\prod_{i=1}^{2} \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{p_1}\sum_{l_2=1}^{p_2} \left( \prod_{i=1}^{2} \mu_{A_i^{l_i}}(x_i) \right)} \tag{16}$$

The membership functions are needed to be selected according to experiences. Moreover, all the states must be known.

### 3.3 ANALYSIS AND DESIGN OF ADAPTIVE SLIDING MODE CONTROLLER BASED ON FUZZY LOGIC

Suppose the optimal parameter [10] as (17):

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Omega} \left[ \sup_{x \in \boldsymbol{R}^n} \left| \hat{f}(\boldsymbol{x}|\boldsymbol{\theta}) \right| - f(\boldsymbol{x}) \right] \tag{17}$$

Where $\Omega$ is the set of $\theta$, i.e., $\theta \in \Omega$.

The term *f(**x**)* [10] can be expressed as (18):

$$f(\boldsymbol{x}) = \boldsymbol{\theta}^{*T} \xi(\boldsymbol{x}) + \varepsilon, \tag{18}$$

where $\boldsymbol{x}$ is the input signal of the fuzzy system, where is the fuzzy vector, $\boldsymbol{\xi}(\boldsymbol{x})$ is approximation error of fuzzy system, and $\varepsilon \leq \varepsilon_N$.

The fuzzy system is used to approximate *f(**x**)*. The fuzzy system input is selected as $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, and the output [10] of the fuzzy system as (19):

$$\hat{f}(\boldsymbol{x}|\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}^T \boldsymbol{\xi}(\boldsymbol{x}) \tag{19}$$

We get the adaptive fuzzy SMC (AFSMC) law for the two-tank interacting system as (20):

$$u_{AFSMC} = \frac{b}{a}\left[ c\dot{e} + \ddot{h}_{2d} - \hat{f}(\boldsymbol{x}) + \eta sign(s) \right] \tag{20}$$

Substituting (20) into (10), we get (21):

$$\begin{aligned} \dot{s} &= c\dot{e} + \ddot{h}_{2d} - f(\boldsymbol{x}) - \frac{a}{b} q_{in} - d(t) \\ &= -f(\boldsymbol{x}) + \hat{f}(\boldsymbol{x}) - \eta sign(s) - d(t) \\ &= -\tilde{f}(\boldsymbol{x}) - \eta sign(s) - d(t) \end{aligned} \tag{21}$$

Since

$$\begin{aligned} \tilde{f}(\boldsymbol{x}) &= f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) \\ &= \boldsymbol{\theta}^{*T} \xi(\boldsymbol{x}) + \varepsilon - \hat{\boldsymbol{\theta}}^T \xi(\boldsymbol{x}) = \tilde{\boldsymbol{\theta}}^T \xi(\boldsymbol{x}) + \varepsilon \end{aligned} \tag{22}$$

where

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}} \tag{23}$$

Lyapunov function [10] is defined as (24):

$$V = \frac{1}{2} s^2 + \frac{1}{2} \gamma \tilde{\boldsymbol{\theta}}^T \tilde{\boldsymbol{\theta}}, \gamma > 0 \tag{24}$$

With a derivative *V*, and from (21), we have (25):

$$\begin{aligned} \dot{V} &= s\dot{s} + \gamma \tilde{\boldsymbol{\theta}}^T \dot{\tilde{\boldsymbol{\theta}}} \\ &= s\left( -\tilde{f}(\boldsymbol{x}) - d(t) - \eta sign(s) \right) - \gamma \tilde{\boldsymbol{\theta}}^T \dot{\hat{\boldsymbol{\theta}}} \\ &= s\left( -\tilde{\boldsymbol{\theta}}^T \boldsymbol{\xi}(\boldsymbol{x}) - \varepsilon - d(t) - \eta sign(s) \right) - \gamma \tilde{\boldsymbol{\theta}}^T \dot{\hat{\boldsymbol{\theta}}} \\ &= -\tilde{\boldsymbol{\theta}}^T \left( s\boldsymbol{\xi}(\boldsymbol{x}) + \gamma \dot{\hat{\boldsymbol{\theta}}} \right) - s\left( \varepsilon + d(t) + \eta sign(s) \right) \end{aligned} \tag{25}$$

Let the adaptive law [10] as (26):

$$\dot{\hat{\theta}} = -\frac{1}{\gamma} s\xi(\boldsymbol{x}) \tag{26}$$

Then,

$$\dot{V} = -s\left(\varepsilon + d(t) + \eta sign(s)\right) = -s\left(\varepsilon + d(t)\right) - \eta|s| \tag{27}$$

Due to the approximation error $\varepsilon$ is sufficiently small, if we design $\eta \geq \varepsilon_N$, we can obtain approximately $\dot{V} \leq 0$.

Using LaSalle's invariance principle, if $t \to \infty$, then $s \to 0$, i.e., $e \to 0$ and $\dot{e} \to 0$.

### 3.4 ADAPTIVE SLIDING MODE CONTROL BASED ON FUZZY LOGIC AND LOW PASS FILTER

The adaptive fuzzy sliding mode control (AFSMC) in (19) still exhibits the chattering phenomenon around the sliding surface because of high level switching frequency in control input signal. To overcome this phenomenon, the study proposes to combine the adaptive fuzzy sliding mode control with the low pass filter (LPF). The proposed controller is aimed to drive the sliding variable 's' to zero in the presence of system uncertainties. The structure of the AFSMC system with LPF (AFSMC_LPF) is shown in Fig. 3.


**Fig. 3.** Structure of the AFSMC_LPF

Where $\tau$ is the AFSMC output and $u$ is the actual control input. The LPF transfer function [11] is given as (28):

$$Q(s) = \frac{\lambda}{s + \lambda} \tag{28}$$

Where $\lambda$ is the time constant of the filter.

We get the AFSMC_LPF for two-tank interacting as (29):

$$u_{ASMCF\_LPF} = \left(\frac{b}{a}\left[-c\dot{e} + \ddot{h}_{2d} - \hat{f}(\boldsymbol{x}) - \eta sign(s)\right]\right)Q(s) \tag{29}$$

Equation (29) provides the necessary control input for a system with filter function Q(s) along with the AFSMC.

## 4. RESULTS AND DISCUSSION

The structure of the AFSMC _ LPF in MATLAB/Simulink is presented as Fig. 4 with $d(t)=5sin(t)$.

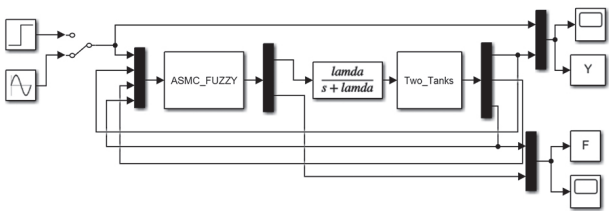



**Fig. 4.** Structure of the AFSMC _ LPF for two-tank interacting in MATLAB/Simulink

The parameters of the two-tank interacting system are presented in Table 1. The fuzzy logic rules are shown in Table 2 with NM - Negative Medium, NS - Negative Small, Z - zero, PS - Positive Small and PM - Positive Medium.

**Table 1.** The parameters of the two-tank interacting system

| Parameters | A1 | A2 | R1 | R2 |
|---|---|---|---|---|
| Value | 0.0145 | 0.0145 | 1478.57 | 642.86 |
| Unit | m2 | m2 | sec/m2 | sec/m2 |

**Table 2.** Fuzzy rule based table

| $\hat{f}(\boldsymbol{x})$ | | x1 | | | | |
|---|---|---|---|---|---|---|
| | | NM | NS | Z | PS | PM |
| x2 | NM | NM | NM | NM | NS | Z |
| | NS | NM | NM | NS | Z | PS |
| | Z | NM | NS | Z | PS | PM |
| | PS | NS | Z | PS | PM | PM |
| | PM | Z | PS | PM | PM | PM |

The membership functions are given as Fig. 5. The approximation result of the function $f(\boldsymbol{x})$ with the fuzzy system is shown as Fig. 6. The response in Fig. 6 shows that the fuzzy system has effectively approximated the function $f(\boldsymbol{x})$ with the approximation error approaches 0.

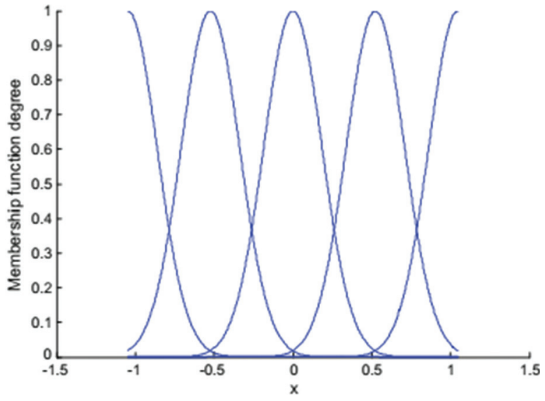



**Fig. 5.** Membership functions of *xi*

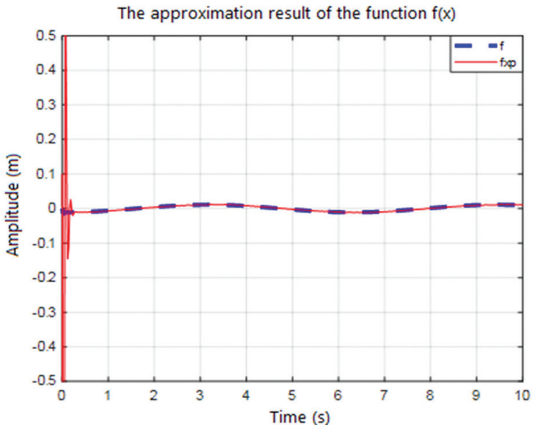



**Fig. 6.** Approximation result of the function $f(\boldsymbol{x})$ with fuzzy system

The step response (0.055m) and error of the system with the AFSMC _ LPF are shown as Fig. 7.

**Table 3.** The achieved quality criteria of the AFSMC _ LPF controller

| Quality criteria | Rising time (s) | Settling time (s) | Overshoot (%) | Steady state error (m) |
|---|---|---|---|---|
| AFSMC _LPF | 2.187 | 3.9133 | 0 | 0 |
| Fuzzy control [19] | 33 | 47.2 | 1.45 | 0 |
| Sliding mode control using conditional integrators [21] | 87.18 | 330 | - | 1.6 |
| Sliding mode control [24] | - | 7.6 | 0 | - |

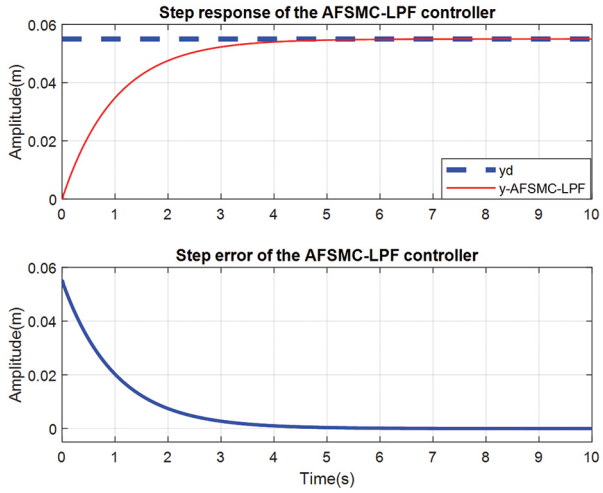



**Fig. 7.** Step response and error of the system with the AFSMC_LPF

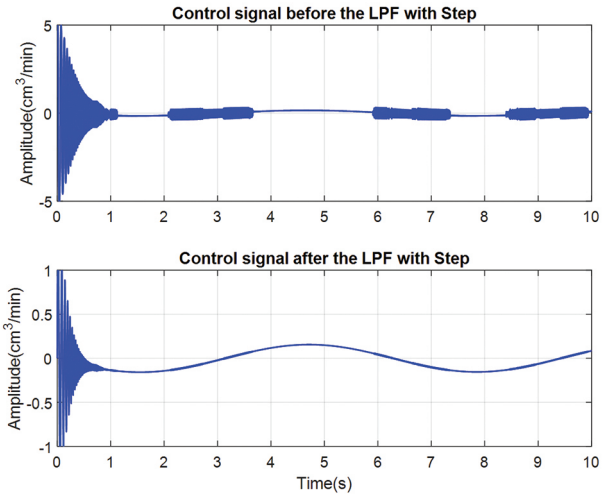



**Fig. 8.** Control signal of the AFSMC _ LPF before and after the LPF with Step

The actual liquid level (denoted as y-AFSMC-LPF) of the system tracks to the reference input (denoted as yd) in a finite time without the overshoot, the rising time achieves 2.187(s), the settling time is about 3.9133(s) and the steady state error converges to 0. These criteria are presented in Table 3 and compared with fuzzy control [19], SMC using conditional integrators [21] and SMC controller [24]. The amplitude of the control signal after passing the low pass filter is reduced about 5 times and significantly reduces the oscillation compared to before the low pass filter when the system response is at steady state. This signal is presented as Fig. 8.

The sine response and error of the AFSMC _ LPF with initial state is [0.1, 0], are shown as Fig. 9. The actual position of the system still tracks the desired input in a finite time with the steady state error converges to 0 and to reduces the chattering phenomenon in control signal as Fig. 10.

The step and sine response of the AFSMC _ LPF with Band Limited White Noise (0.0001w) is used to simulate sensor noise acting at the output of the system are shown in Fig. 11 and 12, respectively. The actual responses of the system still converge to the reference signal in the finite time. This proves that the AFSMC _ LPF controller is suitable to control the system.

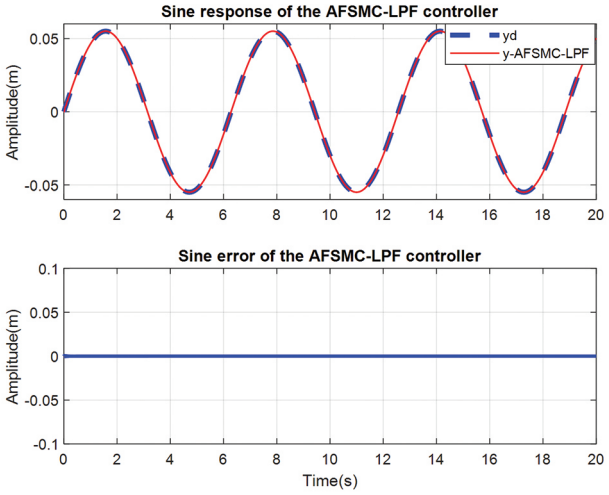



**Fig. 9.** Sine response and error of the AFSMC_LPF

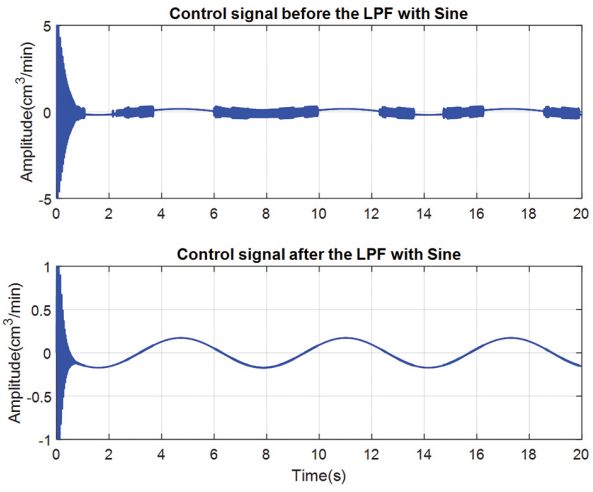



**Fig. 10.** Control signal of the AFSMC _ LPF before and after the filter with Sine

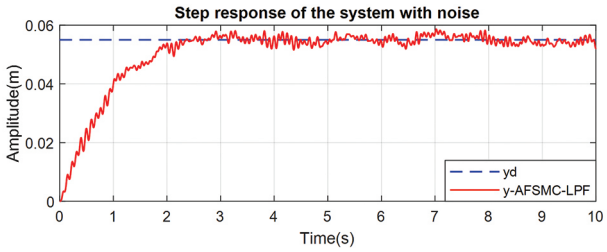



**Fig. 11.** Step response of the AFSMC _ LPF with Band Limited White Noise

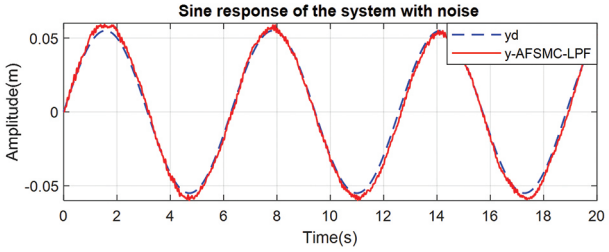



**Fig. 12.** Sine response of the AFSMC _ LPF with Band Limited White Noise

## 5. CONCLUSIONS

This paper presents the design of the adaptive sliding mode control based on the fuzzy logic and the low pass filter for the two-tank interacting system. The sliding mode controller is designed to ensure the actual liquid level of the system converges to the reference in a finite time. The fuzzy logic is used to approximate the function $f(\mathbf{x})$ in the SMC law and the LPF reduces the effect of chattering to a great extent. Responses and control input for AFSMC _ LPF with and without filter are presented. Simulation results with MATLAB/Simulink shown that the AFSMC _ LPF efficient, robust and suitable in liquid level control of the two-tank interacting system. The actual liquid level of the system tracks to the desired input in a finite time (Fig. 7) without the overshoot, the rising time achieves 2.187(s), the settling time is about 3.9133(s) and the steady state error converges to 0. These criteria (Table 3) of the proposed controller are better than the SMC controller, SMC using conditional integrators and fuzzy control. The amplitude of the control signal with yd=0.055m and yd=sine (±0.055m) after passing the low pass filter is reduced about 5 times (Fig. 8, 10). The step and sine response of the AFSMC _ LPF with Band Limited White Noise still converge to the reference signal in the finite time (Fig. 11, 12). In the future, this research will use the intelligent algorithms to optimize the fuzzy system and experiment with real models.

## 6. REFERENCES

[1] S Balamurugan, P Venkatesh, M Varatharajan, "Fuzzy sliding-mode control with low pass filter to reduce chattering effect: an experimental validation on Quanser SRIP", Indian Academy of Sciences, Vol. 42, No. 10, 2017, pp. 1693-1703.

[2] K. Orman, "Design of a Memristor-based Chattering Free Sliding Mode Controller and Speed Control of the BLDC Motor", Tehnicki Vjesnik, Vol. 28, No. 3, 2021, pp. 754-762.

[3] S. Nagammai, S. Latha, M. Varatharajan, "Soft computing-based fuzzy integral sliding mode control: a real-time investigation on a conical tank process", Soft Computing, Vol. 24, No. 7, 2020, pp. 1-12.

[4] M. Bekrani, S. T. Behrooz, M. Heydari, "An Adaptive Control Method using Interval Fuzzy Sliding Mode for Direct Matrix Converters", International Journal of Industrial Electronics, Control and Optimization, Vol. 3, No. 2, 2020, pp. 159-172.

[5] B. Zhang, K. Nie, X. Chen, Y. Mao, "Development of Sliding Mode Controller Based on Internal Model Controller for Higher Precision Electro-Optical Tracking System", Actuators, Vol. 11, No. 1, 2022, pp. 1-19.

[6] N. Ullah, A. Aziz Al-Ahmadi, "A triple mode robust sliding mode controller for a nonlinear system with measurement noise and uncertainty", American Institute of Mathematical Sciences, Vol. 3, No. 2, 2020, pp. 81-99.

[7] B. Xu, J. Li, Y. Yang, H. Wu, O. Postolache, "A Novel Sliding Mode Control with Low-Pass Filter for Nonlinear Handling Chain System in Container Ports", Complexity, Vol. 2020, 2020, pp. 1-15.

[8] U. S. Hussein, B. M. Muhammed, A. Z. Tahir, T. S. Ahmed, T. Sadiq, A. G. Ahmadu, "Methods of Chattering Reduction in Sliding Mode Control: A Case Study of Ball and Plate System", Proceedings of the IEEE 7th International Conference on Adaptive Science & Technology, Acrraa, Ghana, 22-24 August 2018, pp. 1-8.

[9] C. B. Kadu, A. A. Khandekar, "Design of sliding mode controller with PI sliding surface for robust regulation and tracking of process control systems", Journal of Dynamic Systems, Measurement and Control, Vol. 140, No. 9, 2018, pp. 1-11.

[10] J. Liu, "Sliding mode control using MATLAB", Academic Press, 2017.

[11] J. Liu, X. Wang, "Advanced Sliding Mode Control for Mechanical Systems", Springer, 2012.

[12] N. Cibiraj, M. Varatharajan, "Chattering reduction in sliding mode control of quadcopters using neural networks", Proceedings of the 1st International Conference on Power Engineering, Computing and Control, Chennai, India, 2-4 March 2017, pp. 885-892.

[13] L. Wan, G. Chen, M. Sheng, Y. Zhang, Z. Zhang, "Adaptive chattering-free terminal sliding-mode control for full-order nonlinear system with unknown disturbances and model uncertainties", International Journal of Advanced Robotic Systems, Vol. 17, No. 3, 2020, pp. 1-11.

[14] J.-S. Fang, J. Tsai, J.-J. Yan, S.-M. Guo, "Adaptive Chattering-Free Sliding Mode Control of Chaotic Systems with Unknown Input Nonlinearity via Smooth Hyperbolic Tangent Function", Mathematical Problems in Engineering, Vol. 2019, 2019, pp. 1-9.

[15] L. Guo, H. Zhao, Y. Song, "A Nearly Optimal Chattering Reduction Method of Sliding Mode Control with an Application to a Two-wheeled Mobile Robot", arXiv:2110.12706v1, 2021.

[16] A. Can, H. Efstathiades and A. Montazeri, "Desing of a Chattering-Free Sliding Mode Control System for Robust Position Control of a Quadrotor", Proceedings of the International Conference Nonlinearity, Information and Robotics, Innopolis, Russia, 3-6 December 2020, p. 1-6.

[17] A. S. S. Abadi, P. A. Hosseinabadi, N. B. Soin, S. Mekhilef, "Chattering-Free Adaptive Finite-Time Sliding Mode Control for Trajectory Tracking of MEMS Gyroscope", Automatic Control and Computer Sciences, Vol. 54, No. 4, 2020, pp. 335-345.

[18] T. Mien, "Liquid Level Control of Coupled-Tank System Using Fuzzy-PID Controller", International Journal of Engineering Research & Technology, Vol. 7, No. 11, 2017, pp. 459-464.

[19] M. Changela, A. Kumar, "Designing a Controller for Two Tank Interacting System", International Journal of Science and Research, Vol. 4, No. 5, 2015, pp. 589-593.

[20] K Pravallika, G Vamsi Krishna, "Fuzzy-PID controller for coupled two tank interacting system", International Journal of Mechanical and Production Engineering Research and Development, Vol. 10, No. 3, 2020, pp. 8817-8830.

[21] S. B. Prusty, S. Seshagiri, U. C. Pati, K. K. Mahapatra, "Sliding Mode Control of Coupled Tank Systems Using Conditional Integrators", IEEE/CAA Journal of Automatica Sinica, Vol. 7, No. 1, 2020, pp. 118-125.

[22] D. Bhandare, N. Kulkarni, M. Bakshi, "An intelligent self-tuning fuzzy-PID controller to coupled tank liquid level system", International Journal of Information Technology, Vol. 54, 2021, pp. 1-8.

[23] A. Mostafa, A. Fellani, M. Gabaj, "PID Controller Design for Two Tanks Liquid Level Control System using Matlab", International Journal of Electrical and Computer Engineering, Vol. 5, No. 3, 2015, pp. 436-442.

[24] T. Toms, D. Hepsiba. "Comparison of PID Controller with a Sliding Mode Controller for a Coupled Tank System", International Journal of Engineering Research & Technology, Vol. 3, No. 2, 2014, pp. 151-154.

[25] D. Demilade, P. O. Oluseyi, "Fuzzy-Optimized model reference adaptive control of interacting and noninteracting processes based on MIT and Lyapunov rules", Turkish Journal of Engineering, Vol. 5, No. 4, 2021, pp. 141-153.

# Prototyping Design and Optimization of Smart Electric Vehicles/Stations System using ANN

Original Scientific Paper

**Mohamed A. Elkasrawy**
Renewable Energy Programme, Faculty of Engineering, and FabLab in the Center for Emerging Learning Technologies (CELT),
The British University in Egypt (BUE),
11387, Cairo, Egypt
pgmohamed91810003@bue.edu.eg

**Ahmed Hassan**
Electrical Engineering Department, Faculty of Engineering, and FabLab in the Center for Emerging Learning Technologies (CELT),
The British University in Egypt (BUE),
11387, Cairo, Egypt
ahmed186005@bue.edu.eg

**Sameh O. Abdellatif**
Electrical Engineering Department, Faculty of Engineering, and FabLab in the Center for Emerging Learning Technologies (CELT),
The British University in Egypt (BUE),
11387, Cairo, Egypt
Sameh.osama@bue.edu.eg

**Gamal Abdel Shafy Ebrahim**
Computer Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt
gamal.ebrahim@eng.asu.edu.eg

**Hani A. Ghali**
Electrical Engineering Department, Faculty of Engineering, and FabLab in the Center for Emerging Learning Technologies (CELT),
The British University in Egypt (BUE),
11387, Cairo, Egypt
hani.amin@bue.edu.eg

***Abstract*** *– This paper demonstrates an experimental attempt to prototype electric vehicle charging station's (EVCS) decision-making unit, using artificial neural network (ANN) algorithm. The algorithm acts to minimize the queuing delay in the station, with respect to the vehicle state of charge (SOC), and the expected arrival time. A simplified circuit model has been used to prototype the proposed algorithm, to minimize the overall queuing delay. Herein, the worst-case scenario is considered by having number of electric vehicles arriving to the station at the same time greater than the charging points available in the station side. Accordingly, the optimization technique was applied to reduce the mean charging time of the vehicles and minimize queuing delay. Results showed that this model can help in reducing the queuing delay by around 20% of the mean charging time of the station, while referring to a bare model without ANN algorithm as a reference.*

***Keywords***: *Real-time algorithms, Physical realization, Electric vehicles stations, Queuing delay optimization, Artificial neural network (ANN) algorithm.*

## 1. INTRODUCTION

Is the Electric Vehicles (EVs) market a booming market? Various studies have demonstrated such a question [1-4]. The electric vehicles deployment rate has recently risen worldwide and has become the conventional transportation for the cities as it reduces air pollution and fuel dependency [5]. Accordingly, queuing delay and charging cost problems appear, and many researchers tried to solve them using different optimization techniques. Previous works were investigated to reduce the charging cost for EVs, either by introducing new charging schemes, integrating renewable energy sources, or utilizing decentralized in-door charging [6-10]. The current study focuses more on the other problem, queuing delay. One of the solutions to minimize queuing delay is by increasing charging points with respect to electric vehicles demand in each zone [11]. Reducing electric vehicle charging time and queuing delay will enhance the usage of electric vehicles globally. As the increases in electric vehicle usage may lead to congestion in the electric vehicle charging stations (EVCS). Accordingly, optimization of high priority vehicles needs to be established, which will lead to rearrangement of vehicles

entering the station simultaneously [12, 13]. Fast charging modes like constant current constant voltage (CCCV) and multistage constant current (MSCC) may lead to a decrease in charging time, but also as the charging duration is longer than fueling time so that congestion may happen, and queuing delay will occur [14].

Another factor affecting the queuing delay is the state of charge (SOC) upon arriving at the station. The lower the SOC, the higher the charging priority. The battery management system is the primary key in controlling the SOC, as it contains many accurate algorithms that control the functional status of the battery [15]. Previous literature demonstrates various attempts to provide intelligent battery management systems in vehicles, seeking optimum charging, either time or cost. The work in [16] provided an attempt to utilize fuzzy logic to control a vehicle's battery management system. Another trial based on vehicle-to-vehicle technology was conducted in [17]. Moreover, the charging swapping methodology was utilized as in [18]. Alternatively, the integration of self-controlling and renewable resources were investigated [19].

Integrating machine learning models in optimization problems has attracted various researchers' attention. A study of mathematical optimization methods and machine learning to estimate optimized solutions by being trained on previously optimized solutions shows that the machine learning model is more accurate and efficient in obtaining the value of the optimized instances [20]. In [21], global solar radiation is required to design a proper energy conversion system due to the high cost of measurement several mathematical models are used linear and nonlinear in parallel with a machine learning model; result conducts that machine learning accuracy compared by actual reading shows more accurate results as error percentage at some time reaches less than 2%. In [22], scientists used artificial neural network (ANN) for medical purposes, and they faced some problems regarding feeding the network with data from a real-world, determination of the best structure as it needs many trials, eccentric network behavior as ANN has some limitations regarding providing clue how results are obtained when it provides a probe solution to specific problem and duration of the network is un-expectable beside to is an unknown clue. Overall, it can be concluded from previous work that ANN is more practicable, applicable, and accurate than conventional mathematical optimization models.

On the other hand, ANN has limitations regarding time, unknown behavior, feeding data translation, and hardware dependency. ANN's advantages outweigh its disadvantages as ANN are developing rapidly, and scientists are eliminating its disadvantage one by one, which shows a great perspective in the future. Provided all these attempts to effectively optimize the charging process of vehicles, in both dimensions cost and time, the physical realization and prototyping of such an intelligent system are still missing in the literature.

This paper proposes an accurate prototyped model implemented using an embedded system controller to emulate electric vehicles that communicate with electric vehicle stations. 12 EVs nodes (Figure 1) were fabricated with scaling down the EV battery by two 3-V batteries, the controlling unit is implemented using an Arduino controller, and the data is demonstrated using an LCD screen (cf. Figure 2). The intelligent point is considered in the Arduino kit with the ANN model running to minimize the queuing delay on the EV station side. Again, LCDs are mounted in the EV stations to demonstrate the predicted queuing delay for each arriving vehicle. A real-time energy management model is deployed on the Arduino kit for estimating the SOC and the expected charging time, with the aid of the AI model. The model pretends to estimate the queuing delay using the developed ANN model. A series of experimental measurements were recorded and compared with theoretical expectations.

## 2. PROTOTYPE DESIGN METHODOLOGY

As a prototype, 12 electric vehicles are assumed to arrive per two stations, with three charging points in each station (c.f. Figure 1). For the sake of decision-making on the station side, estimating the battery's state of charge (SOC) is an essential task. Our proposed prototype is a scaled-down model with two 3-V batteries connected in series for each EV, while Arduino is acting as a control unit [23, 24].

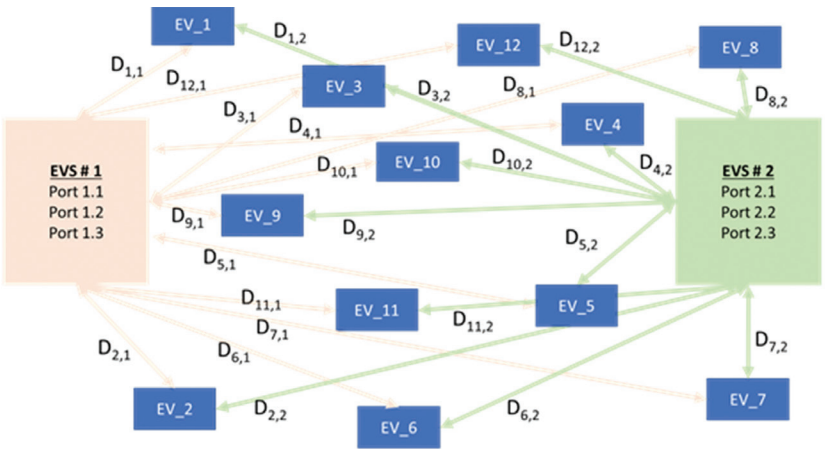



**Fig. 1.** Schematic diagram for the EVs scenario with two available EV stations. The distance between EV and EV station is denoted as Dm,n, where m is the EV number and n is the EV station (EVS) number.

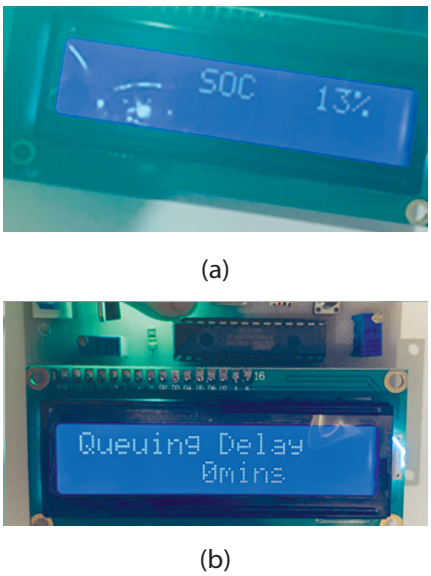
The Arduino is deployed with an AI model capable of estimating the SOC as well as charging time, assuming CCCV charging scheme, by:

$$CC_{Station} = CC_{full} - \frac{CC_{full}\, x\, SOC}{100} \quad (1)$$

$$T_{charging} = \frac{CC_{Station}}{DC_{max} x\, 0.8905}\, x\, 60 \quad (2)$$

where $CC_{Station}$ is the charging capacity of the battery upon reaching the station, $CC_{full}$ is the full charging capacity, $T_{charging}$ is the charging time and $DC_{max}$ is the maximum acceptable DC charging power in KW. Each EV is provided with LCD display to show the current SOC (see Figure 2-a). Alternatively, EV stations are prototyped as a receiving node with Arduino controller, see Figure 2-b. EV nodes and EVCS nodes are communicating via 433 MHz RF transceiver [23, 24].

For estimating the queuing delay, a real-time artificial intelligent energy management model is utilized as described in the flowchart demonstrated in Figure 3.




(a)

(b)

**Fig. 2.** (a) EV node showing the current SOC, and (b) EVS node showing the expected queuing delay.

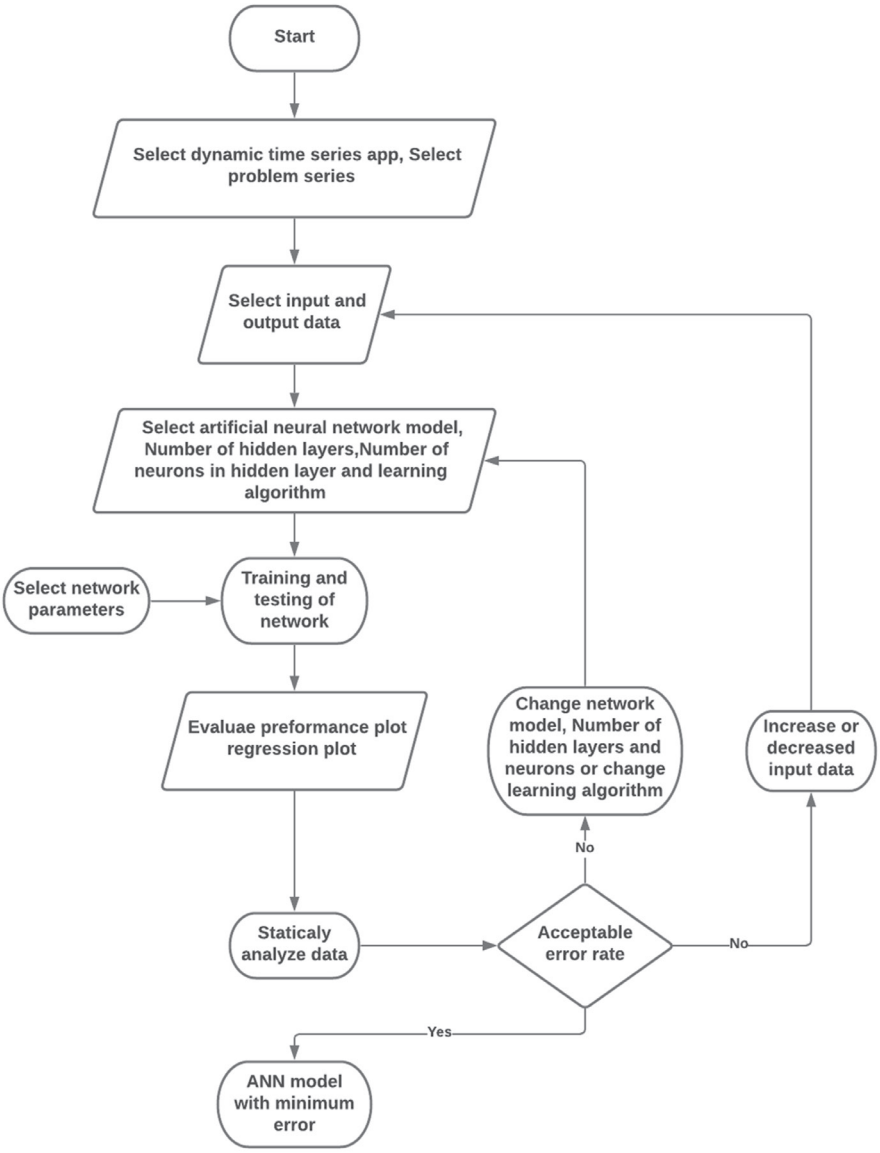



**Fig. 3.** Flow-chart for a real-time ANN model to simultaneously predict queuing delay in EV station with EVs arriving station.

Levenberg-Marquardt has been chosen in this work as it showed the highest validation among other algorithms, including Bayesian regularization and scaled conjugate gradient, as reported in [25].

A generated dataset of 12,234 point was developed, while utilizing our database of EVs, published in [12] (cf. table 1). To ensure the suitability of the chosen ANN model with respect to the optimization problem under test, the generated dataset was tested. The dataset has been divided into three subsets: training, validation, and testing [25, 26]. A 70 % of the dataset was included as a training data, where the model learning process is conducted. Another 15% of the dataset are used for validating the data in terms of the data range under test. Finally, the remaining 15% are seeded as new input to the system to examine the prediction capabilities of the learning process implemented with the first 70% of the dataset. The accuracy of the validation process showed around 93%, while 91% and 71% were observed for the validation and testing process, respectively. As an overall accuracy performance indication, 89% accuracy was recorded for the suggested ANN model. We consider this output as an acceptable accuracy for the model to be used in the current study, with the possibility for further improvement in the future work by limiting the relatively high variation in accuracy (71% to 93%).

As time-series app helps in utilizing the inputs and outputs to continuously predict the minimum expected EVs queuing delay, a real-time autoregressive nonlinear optimization algorithm [26] is used to rank EVs according to the SOC at the station, as well as the expected arrival time as a function in the distance to the station $D_{m,n}$. The model is considering a possibility for a queuing delay whenever the number of arriving vehicles is more than the available charging points at the station side, assumed to be three points per station.

As mentioned earlier, the illustrated model was deployed on the Arduino kit for physical realization. The Arduino kit supplies the circuit by 5 volts, the LCD is supplied by 10 volts total, and the data wires are connected to (8,9,10,11,12,13) digital pins with (RS, E, D4, D5, D6, D7). LEDA is connected to the positive side to provide better vision, and the batteries are connected to a potentiometer to vary the output voltage controlling the SOC, as shown in Figure 4.

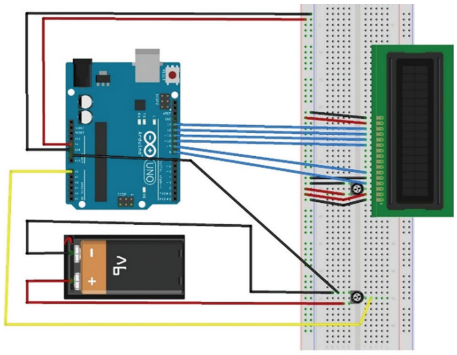
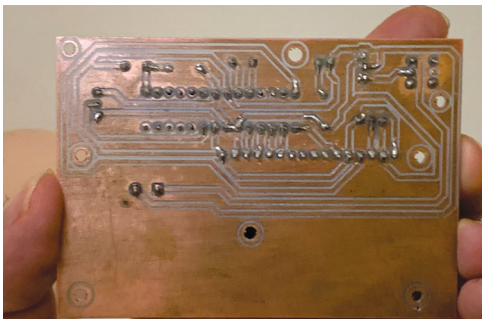



**Fig. 4.** Wiring diagram for the EV node

Consequently, the actual circuit realization is shown in Figures 5-a and b.


(a)



(b)

**Fig. 5.** (a) and (b) demonstrates the actual circuit model for the EV node.

## 3. RESULTS AND DISCUSSIONS

The hardware system described in section two is then utilized in capturing actual data concerning the change of the queuing delay with the SOC variation following the ANN model prediction. The model depends upon scaling down the vehicle battery, Fiat 500 Model 2018, with a battery capacity of 42 KW.hr, to a small battery used with a small capacity, as described in section two. This model helped show the SOC's effect on the charging time and queuing delay.

As mentioned in Figure 1, 12 EVs were prototyped to communicate with two EVs. The EV decides which EVCS to access based on the SOC recorded at the EV node, the distance between the EV node and EVCS node, and the queuing delay expected at EVCS based on the traffic. The conducted experiment is implemented by fixing all EVs in terms of distance to stations, as (Dm,n), while one EV is kept with variable SOC. Accordingly, the SOC variation can influence the decision taken and reflect the expected queuing delay.

Figure 6 shows that the queuing delay is directly proportional to the SOC, while the charging time is inversely proportional. The experiment is conducted so that a group of vehicles enters the charging stations when all vehicles' SOC remains static while one vehicle SOC varies in our prototype to show how the change in SOC affects the charging time and queuing delay problem.

It was observed that the queuing delay could be predicted only if the number of EVs arriving at the station exceeds the number of available charging points at the station. Herein, all vehicles were charged based on the DC constant current constant voltage (DC-CCCV) charging mode, despite any congestion in stations that may occur.

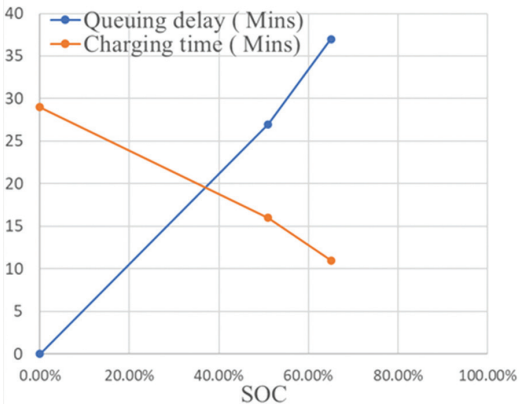



**Fig. 6.** Charging time and queuing delay variation against SOC for the worst-case scenario, including 12 arriving EVs, to two EV stations.

Conceptually, the optimization technique used in this paper is driven by arranging the vehicles with the highest charging time in descending order, as this technique will minimize the total mean charging time, which will decrease the mean charging time along the way day. Screenshots for the recorded SOC and queuing delay are demonstrated in Figure 7, all data are listed in Table 1. The figure illustrates three randomly selected EVs, showing the EV node side with the SOC, and the EV station side with charging time and queuing delay. The same 12 EVs scenario was run without performing the ANNN decision model on the EVs side to test the model's utility. The average queuing delay was calculated with and without the ANN model in both cases. It was shown that the average queuing delay with applying the ANN model is reduced by 20%, with respect to the bare case (24.5 mins to 29.2 mins, respectively).

The model shows that the queuing delay is directly proportional to the SOC as when the SOC increases, the queuing delay increases. This means that the optimization algorithm identifies the high-priority vehicles with the lowest SOC to minimize the queuing delay for those vehicles as it reaches zero. In contrast, vehicles with high SOC will have the longest queuing delay time even though they have the shortest charging time. Overall, the average mean charging time, including the queuing delay with applying optimization technique for those vehicles, will be less than the average mean charging time without applying this technique as calculated earlier. Based on the above, the proposed prototype validates the utility of the selected ANN model to minimize the queuing delay at rush hours. The model can be extended to include more EVs as well as EVCS. We considered such extension as a part of the future work.

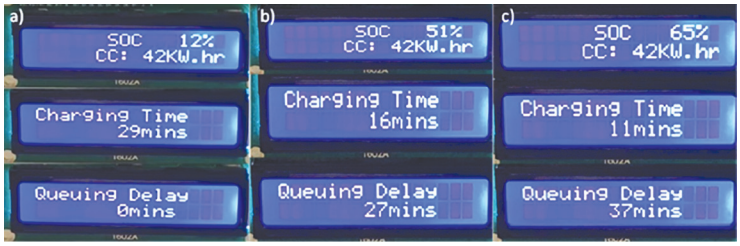



**Fig. 7.** Recorded SOC, charging time, and queuing delay for three samples out of the 12 EV prototyped (a) EV # 1, (b) EV # 5, and EV # 8.

| EV_# | Initial SOC (%) | Estimated SOC at Station | Predicted Charging Time (mins) | Dm,1 (m) | Dm,2 (m) | Queuing delay (mins) |
|---|---|---|---|---|---|---|
| EV_1 | 12 | 9.57545117 | 29.13756525 | 0.618588 | 1.42121 | 0 |
| EV_2 | 30 | 33.56969084 | 21.50423098 | 4.98713 | 5.19231 | 11.59742 |
| EV_3 | 40 | 28.32280701 | 36.57449544 | 4.17398 | 2.78219 | 18.55471 |
| EV_4 | 50 | 37.58541729 | 16.86148293 | 2.90895 | 4.98562 | 0 |
| EV_5 | 51 | 36.38462651 | 16.33263482 | 3.84365 | 4.25879 | 27.58459 |
| EV_6 | 60 | 43.85077205 | 16.42789885 | 4.77989 | 3.51666 | 0 |
| EV_7 | 70 | 74.27270927 | 21.07815914 | 1.001587 | 1.875104 | 9.54214 |
| EV_8 | 65 | 27.53896209 | 11.536542497 | 4.14293 | 4.35069 | 37.65812 |
| EV_9 | 80 | 78.63664373 | 3.808792758 | 1.316519 | 1.989016 | 0 |
| EV_10 | 25 | 21.94191819 | 24.03313005 | 0.877803 | 2.09306 | 0 |
| EV_11 | 35 | 26.22420666 | 38.30491932 | 1.64114 | 3.37592 | 7.11244 |
| EV_12 | 45 | 45.82565932 | 11.6138407 | 0.786987 | 1.854612 | 0 |

**Table. 1.** Recorded SOC, charging time, distance to stations and queuing delay for three samples out of the 12 EV prototyped. Selected station is underlined per EV

More towards assessing the utility of the proposed ANN model, the model is evaluated against our previous developed model in [12]. The work in [12] demonstrated meta-heuristic/ nonlinear programming optimization models that can estimate the EV queuing delay as well as the entrance to the station sequence. Applying the same boundaries defined in Figure 1, in terms of number of EVs and stations, the two models were compared. It is a fact that the ANN model adds some complexity to the system. However, the observed accuracy in capturing the global minimum and estimating the queuing delay was enhanced through the ANN with respect to the water cycle optimization technique (WCOT) in [12] by around 12.48%. In a detailed manner, the superiority of our proposed ANN model over the previous work in [12] can be observed in both minimizing queuing delay as well as maximizing EV charging capacity. The work in [12] acted to minimize the queuing delay through limiting the EV charging capacity by introducing the partial charging factor (σ). As a result, the queuing delay is minimized, while the charging capacity of the served EVs is not at maximum level. The suggested ANN model reached a global minimum point for the queuing delay, with all served EVs were charged up to 100%. Another added value is considered in building an accumulative learning curve through updating the ANN database per EV arrival. Herein, the ANN can make use of all previous history of both EV and station in making the decision, while the WCOT still rely only on the instantaneous data received from the EVs arriving at the same time.

## 4. CONCLUSIONS

Electric mobility is considered a vital model that plays a critical role in nations' development and directly impacts the environment and the economy. Green and sustainable electrification for the automation industry is considered a proactive model to minimize global pollution. However, providing optimum charging time in the centralized stations is still missing. Based on that, centralized electric vehicle stations glow up readily these days, seeking a green and clean transportation alternative. Long-charging time obstacle such market. This long charging time is attributed to charging technologies and queuing delay, especially for underestimated electric vehicles stations or overflow in rush hours.

With the aid of intelligent artificial intelligent decision-making systems, we provide an actual prototype for an electric vehicle station decision-making unit, that communicate with emulated vehicles. The model can estimate the expected queuing delay and arrange the vehicles in an optimized order in case of overflow. The current study demonstrated an actual physical realization of the model through prototyping 12 EVs communicating with two electric vehicles stations. The decision is taken, per each EV, based on the SOC, the distance to the station, and the available ports at the EVCS side. A bare model without running the ANN model on the EVCS side was also considered a reference. An overall 20% reduction in total charging time was estimated. Additionally, the model showed capabilities better than both meta-heuristic as well as hybrid nonlinear programming and meta-heuristic algorithms in capturing the minimum expected queuing delay. As future work, the current model can be enlarged either in terms of the prototyped elements or level of intelligence by utilizing other machine learning models and more reliable processing units. Additionally, assessing the current ANN model with respect to conventional scheduling algorithms can be part of the future work.

## 5. REFERENCES:


[1] C. A. Guevara, E. Figueroa, M. A. Munizaga, "Paving the road for electric vehicles: Lessons from a randomized experiment in an introduction stage market", Transportation Research Part A: Policy and Practice, Vol. 153, 2021, pp. 326-340.

[2] S. Fu, H. Fu, "A method to predict electric vehicles' market penetration as well as its impact on energy saving and CO2 mitigation", Science Progress, Vol. 104, 2021, p. 00368504211040286.

[3] A. H. K. Babar, Y. Ali, "Enhancement of electric vehicles' market competitiveness using fuzzy quality function deployment", Technological Forecasting and Social Change, Vol. 167, 2021, p. 120738.

[4] R. D. Arnott, B. Cornell, L. J. Wu, "Big Market Delusion: Electric Vehicles", SSRN, 3801052, 2021.

[5] E. A. Nanaki, C. J. Koroneos, "Climate change mitigation and deployment of electric vehicles in urban areas", Renewable energy, Vol. 99, 2016, pp. 1153-1160.

[6] M. U. Javed, N. Javaid, A. Aldegheishem, N. Alrajeh, M. Tahir, M. Ramzan, "Scheduling charging of electric vehicles in a secured manner by emphasizing cost minimization using blockchain technology and IPFS", Sustainability, Vol. 12, 2020, p. 5151.

[7] S. Hussain et al. "A Heuristic Charging Cost Optimization Algorithm for Residential Charging of Electric Vehicles", Energies, Vol. 15, 2022, p. 1304.

[8] R. Zhang, T. Hanaoka, "Deployment of electric vehicles in China to meet the carbon neutral target by 2060: Provincial disparities in energy systems, CO2 emissions, and cost effectiveness", Resources, Conservation and Recycling, Vol. 170, 2021, p. 105622.

[9] Z. Liu et al. "Comparing total cost of ownership of battery electric vehicles and internal combustion engine vehicles", Energy Policy, Vol. 158, 2021, p. 112564.

[10] N. M. Hassan, S. O. Abdellatif, "Assessing Centralized and Decentralized EV Charging Schemes using PV-Grid Connected System, Case Study in Egypt", Proceedings of the International Conference on Microelectronics, New Cairo City, Egypt, 19-22 December 2021, pp. 232-235.

[11] D. Said, H. T. Mouftah, "A novel electric vehicles charging/discharging management protocol based on queuing model", IEEE Transactions on Intelligent Vehicles, Vol. 5, 2019, pp. 100-111.

[12] P. Makeen, S. Memon, M. A. Elkasrawy, S. O. Abdullatif, H. A. Ghali, "Smart green charging scheme of centralized electric vehicle stations", International Journal of Green Energy, 2021, pp. 1-9.

[13] M. Elkasrawy, P. Makeen, S. O. Abdellatif, H. A. Ghali, "Optimizing electric vehicles station performance using AI-based decision maker algorithm", Proceedings of Emerging Topics in Artificial Intelligence, Vol. 114691, 24 August - 4 September 2020.

[14] M. A. Liaqat, Z. Hussain, Z. Khan, M. A. Akram, A. Shuja, "Effects of Ag doping on compact TiO2 thin films synthesized via one-step sol–gel route and deposited by spin coating technique", Journal of Materials Science: Materials in Electronics, Vol. 31, 2020, pp. 7172-7181.

[15] Z. Cen, P. Kubiak, "Lithium-ion battery SOC/SOH adaptive estimation via simplified single particle model", International Journal of Energy Research, Vol. 44, 2020, pp. 12444-12459.

[16] S. Hussain, Y. S. Kim, S. Thakur, J. G. Breslin, "Optimization of Waiting Time for Electric Vehicles Using a Fuzzy Inference System", IEEE Transactions on Intelligent Transportation Systems, 2022. (in press)

[17] H. Abualola, H. Otrok, R. Mizouni, S. Singh, "A V2V charging allocation protocol for electric vehicles in VANET", Vehicular Communications, Vol. 33, 2022, p. 100427.

[18] Y. Tao et al. "Data-driven on-demand energy supplement planning for electric vehicles considering multi-charging/swapping services", Applied Energy, Vol. 311, 2022, p. 118632.

[19] E. A. Kakillioglu, M. Y. Aktaş, N. Fescioglu-Unver, "Self-controlling resource management model for electric vehicle fast charging stations with priority service", Energy, Vol. 239, 2022, p. 122276.

[20] Y. Mokhtari, D. Rekioua, "High performance of maximum power point tracking using ant colony algorithm in wind turbine", Renewable energy, Vol. 126, 2018, pp. 1055-1063.

[21] Ö. Çelik, A. Teke, H. B. Yıldırım, "The optimized artificial neural network model with Levenberg–Marquardt algorithm for global solar radiation estimation in Eastern Mediterranean Region of Turkey", Journal of cleaner production, Vol. 116, 2016, pp. 1-12.

[22] D. A. Khalilov, N. A. K. Jumaboyeva, T. M. K. Kurbonova, "Advantages and Applications of Neural Networks", Academic research in educational sciences, Vol. 2, 2021.

[23] S. M. Maher, Z. M. Ali, H. H. Mahmoud, S. O. Abdellatif, M. M. Abdellatif, "Performance of RF underwater communications operating at 433 MHz and 2.4 GHz", Proceedings of the International Conference on Innovative Trends in Computer Engineering, Aswan, Egypt, 2-4 February 2019, pp. 334-339.

[24] S. M. Maher, Z. M. Ali, S. O. Abdellatif, M. M. Abdellatif, "Enhancing the Performance of 433 MHz Underwater WSN Using Handover Mechanisms", Journal of Communications, Vol. 15, 2020.

[25] R. Merjulah, J. Chandra, "Classification of myocardial ischemia in delayed contrast enhancement using machine learning", Intelligent Data Analysis for Biomedical Applications, 2019, pp. 209-235.

[26] T. Coleman, M. A. Branch, A. Grace, "Optimization toolbox", For Use with MATLAB. User's Guide for MATLAB 5, Version 2, Relaese II, 1999.

# INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia.

## About this Journal

The International Journal of Electrical and Computer Engineering Systems publishes original research in the form of full papers, case studies, reviews and surveys. It covers theory and application of electrical and computer engineering, synergy of computer systems and computational methods with electrical and electronic systems, as well as interdisciplinary research.

## Topics of interest include, but are not limited to:

- Power systems
- Renewable electricity production
- Power electronics
- Electrical drives
- Industrial electronics
- Communication systems
- Advanced modulation techniques
- RFID devices and systems
- Signal and data processing
- Image processing
- Multimedia systems
- Microelectronics
- Instrumentation and measurement
- Control systems
- Robotics
- Modeling and simulation
- Modern computer architectures
- Computer networks
- Embedded systems
- High-performance computing
- Parallel and distributed computer systems
- Human-computer systems
- Intelligent systems
- Multi-agent and holonic systems
- Real-time systems
- Software engineering
- Internet and web applications and systems
- Applications of computer systems in engineering and related disciplines
- Mathematical models of engineering systems
- Engineering management
- Engineering education

### Paper Submission

Authors are invited to submit original, unpublished research papers that are not being considered by another journal or any other publisher. Manuscripts must be submitted in doc, docx, rtf or pdf format, and limited to 30 one-column double-spaced pages. All figures and tables must be cited and placed in the body of the paper. Provide contact information of all authors and designate the corresponding author who should submit the manuscript to https://ijeces.ferit.hr. The corresponding author is responsible for ensuring that the article's publication has been approved by all coauthors and by the institutions of the authors if required. All enquiries concerning the publication of accepted papers should be sent to ijeces@ferit.hr.

The following information should be included in the submission:

- paper title;
- full name of each author;
- full institutional mailing addresses;
- e-mail addresses of each author;
- abstract (should be self-contained and not exceed 150 words). Introduction should have no subheadings;
- manuscript should contain one to five alphabetically ordered keywords;
- all abbreviations used in the manuscript should be explained by first appearance;
- all acknowledgments should be included at the end of the paper:
- authors are responsible for ensuring that the information in each reference is complete and accurate. All references must be numbered consecutively and citations of references in text should be identified using numbers in square brackets. All references should be cited within the text;
- each figure should be integrated in the text and cited in a consecutive order. Upon acceptance of the paper, each figure should be of high quality in one of the following formats: EPS, WMF, BMP and TIFF;
- corrected proofs must be returned to the publisher within 7 days of receipt.

### Peer Review

All manuscripts are subject to peer review and must meet academic standards. Submissions will be first considered by an editor-in-chief and if not rejected right away, then they will be reviewed by anonymous reviewers. The submitting author will be asked to provide the names of 5 proposed reviewers including their e-mail addresses. The proposed reviewers should be in the research field of the manuscript. They should not be affiliated to the same institution of the manuscript author(s) and should not have had any collaboration with any of the authors during the last 3 years.

### Author Benefits

The corresponding author will be provided with a .pdf file of the article or alternatively one hardcopy of the journal free of charge.

#### Units of Measurement

Units of measurement should be presented simply and concisely using System International (SI) units.

### Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003

Published: semiannually

### Copyright



### Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.

### Postal Address

Faculty of Electrical Engineering,
Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b
31000 Osijek, Croatia

# IJECES Copyright Transfer Form

(Please, read this carefully)

This form is intended for all accepted material submitted to the IJECES journal and must accompany any such material before publication.

**TITLE OF ARTICLE** (hereinafter referred to as "the Work"):

COMPLETE LIST OF AUTHORS:

The undersigned hereby assigns to the IJECES all rights under copyright that may exist in and to the above Work, and any revised or expanded works submitted to the IJECES by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the complete Work and all incorporated parts of the Work. Otherwise he/she warrants that necessary permissions have been obtained for those parts of works originating from other authors or publishers.

Authors retain all proprietary rights in any process or procedure described in the Work. Authors may reproduce or authorize others to reproduce the Work or derivative works for the author's personal use or for company use, provided that the source and the IJECES copyright notice are indicated, the copies are not used in any way that implies IJECES endorsement of a product or service of any author, and the copies themselves are not offered for sale. In the case of a Work performed under a special government contract or grant, the IJECES recognizes that the government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official government purposes only, if the contract/grant so requires. For all uses not covered previously, authors must ask for permission from the IJECES to reproduce or authorize the reproduction of the Work or material extracted from the Work. Although authors are permitted to re-use all or portions of the Work in other works, this excludes granting third-party requests for reprinting, republishing, or other types of re-use. The IJECES must handle all such third-party requests. The IJECES distributes its publication by various means and media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various collections, databases and other publications. The IJECES publisher requires that the consent of the first-named author be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes. If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IJECES publisher assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Authors of IJECES journal articles and other material must ensure that their Work meets originality, authorship, author responsibilities and author misconduct requirements. It is the responsibility of the authors, not the IJECES publisher, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it.

- The undersigned represents that he/she has the authority to make and execute this assignment.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
- The undersigned agrees to indemnify and hold harmless the IJECES publisher from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.

_____ _____

**Author/Authorized Agent** **Date**

**CONTACT**
**International Journal of Electrical and Computer Engineering Systems (IJECES)**
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b
31000 Osijek, Croatia
Phone: +38531224600,
Fax: +38531224605,
e-mail: ijeces@ferit.hr