



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Neural Network-Based Channel Estimation and Detection in Spatial Modulation VLC Systems

Citation for published version:

Palitharathna, K, Suraweera, HA, Godaliyadda, R, Herath, VR & Thompson, J 2022, 'Neural Network-Based Channel Estimation and Detection in Spatial Modulation VLC Systems', *IEEE Communications Letters*, vol. 26, no. 7, pp. 1598-1602. <https://doi.org/10.1109/LCOMM.2022.3166221>

Digital Object Identifier (DOI):

[10.1109/LCOMM.2022.3166221](https://doi.org/10.1109/LCOMM.2022.3166221)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

IEEE Communications Letters

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Neural Network-Based Channel Estimation and Detection in Spatial Modulation VLC Systems

Kapila W. S. Palitharathna, *Member, IEEE*, Himal A. Suraweera, *Senior Member, IEEE*, Roshan I. Godaliyadda, *Senior Member, IEEE*, Vijitha R. Herath, *Senior Member, IEEE*, and John S. Thompson, *Fellow, IEEE*

Abstract—We consider a spatial modulation aided indoor visible light communication system with user mobility and random receiver orientation. Two artificial neural networks (ANNs) are proposed which are able to predict the channel state information (CSI) with high accuracy and resolution. These architectures use estimated CSI at pilot instances obtained using least square or minimum mean square error estimation and predict CSI at intermediate locations. Moreover in ANN 2, predicted user position information is used to improve the performance. Numerical results show that the proposed ANNs deliver a better bit error rate compared to a benchmark spline interpolation-based method. Further, ANN 2 is shown to perform robustly in a high mobility scenario.

Index Terms—Visible light communication, spatial modulation, artificial neural network, user mobility.

I. INTRODUCTION

In the recent developments of wireless systems, visible light communication (VLC) has become a promising technology to provide high-speed reliable communication using the existing indoor lighting infrastructure [1]. Moreover, spatial modulation (SM), which is a promising technique to enhance the spectral efficiency, has been studied in the context of VLC [2], [3]. SM is a technique that enables modulation over space, across different antennas at a transmitter. Specifically, directive light patterns of light-emitting diodes (LEDs) and LED/photodiode (PD) positioning can be exploited to obtain distinct channel gains that are essential for SM.

In the literature several papers have applied SM to VLC systems [3]–[7]. In [3], the bit error ratio (BER) performance of a generalized SM scheme that activates multiple LEDs per channel use has been analyzed. In [5], a precoding scheme for SM has been presented. Adaptive SM in which an arbitrary number of transmitters can be used has been studied in [4]. The work in [6] proposed a novel LED array structure to enable reliable SM transmission by manipulating the transmitter geometry. In [7], a SM aided system has been proposed to cope with receiver orientation errors and device blockages. In [8], a group-based LED selection method is proposed for generalized SM under user mobility and receiver orientation.

User mobility is a key aspect in VLC systems since movement causes time variations of the channel gains. In addition, random receiver orientations due to human activities such as sitting and walking also influence the channel gains. While most of the literature has studied static receivers, some

papers have considered user mobility and random receiver orientation [9], [10]. Typically, designing a receiver for robust communication under such conditions is challenging. Moreover, accurate channel estimation in high mobility is difficult to achieve and can cause errors at the data detection phase.

In a VLC receiver, channel estimation is performed prior to the data detection phase. For that purpose, interleaved pilot symbols in the time and/or frequency domain are sent [11]. Conventional channel estimation techniques such as least squares (LS) and minimum mean square error (MMSE) estimation have been explored in the context of VLC systems [11]–[14]. To finely estimate the channel with high mobility, an additional amount of pilot symbols should be inserted. In order to predict the channel state information (CSI), interpolation techniques can be used. Among them, linear, spline, and piece-wise cubic Hermite polynomial interpolation are well established techniques [15]. Depending on the pilot spacing and non-linear channel variation due to mobility, interpolated channel gains can be mismatched with the actual channel gains. On the other hand, machine learning based techniques have gained attention to perform transmitter and receiver operation in VLC systems [14]. However, fluctuations of the channel gain matrix due to the mobility and receiver orientation in indoor settings can cause a significant performance deterioration in the considered SM VLC system. To the best of authors knowledge, work on suitable artificial neural network (ANN) architectures for channel estimation and detection of a SM VLC systems with practical conditions such as user mobility and random receiver orientation have not been developed. To this end, we propose novel ANN based solutions with path prediction incorporated. In this paper, we present ANN based channel prediction method which can reduce the number of pilot symbols and result in reduced BER for SM aided indoor VLC systems. We consider a realistic user mobility model based on random way-points and a random receiver orientation model based on measurements. SM detectors are sensitive to effects such as user mobility and random receiver orientation errors. The goal of the proposed technique is to recover the transmitted signals based on accurate CSI estimation. ANN based CSI estimation can capture random and realistic channel conditions. Therefore, it is effective in many cases of practical scenarios. Our contributions are summarized as follows.

- We present two ANNs which can predict CSI with higher accuracy and resolution with user mobility and random receiver orientation. ANN 1 uses only estimated channels at known pilot instances as inputs while ANN 2 uses both estimated channels and predicted user locations as inputs to generate results.
- Results reveal that the presented architectures are capable of achieving significant performance improvement compared to the benchmark spline interpolation method.

K. W. S. Palitharathna is with the Centre for Telecommunication Research, School of Engineering, Sri Lanka Technological Campus, Padukka, 10500, Sri Lanka. (Email: kapilap@sltc.ac.lk)

H. A. Suraweera, R. I. Godaliyadda, and V. R. Herath are with the Department of Electrical and Electronic Engineering, University of Peradeniya, Peradeniya, 20400, Sri Lanka. (Email: {himal, roshangodd, vijitha}@ee.pdn.ac.lk)

J. S. Thompson is with the School of Engineering, Institute for Digital Communications, The University of Edinburgh, Edinburgh, EH9 3JL, UK. (Email: j.s.thompson@ed.ac.uk)

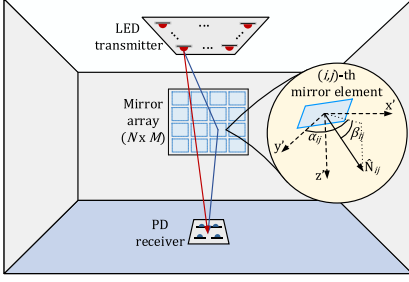


Fig. 1: Indoor SM aided VLC system. The azimuth and polar angles of the device are also marked.

Specifically, ANN 2 is well suited for high mobility conditions since it captures the mobility and requires a small window size for prediction.

II. SYSTEM MODEL

Fig. 1 shows the indoor SM aided VLC system with N_t ceiling mounted LEDs and N_r receiver PDs. We assume on-off keying for the transmission. Each LED is either OFF or emits light of intensity $\{I_0, I_1\} \in \mathcal{I}$ per channel use, where \mathcal{I} is the set of all possible intensity levels [3]. Hence, the number of bits transmitted in a given channel use is $\eta_{SM} = \lfloor \log_2(N_t) \rfloor + \log_2(M_I)$ where the set of intensity levels M_I for on-off keying is two and $\lfloor \cdot \rfloor$ is the floor function. Let $\mathbf{s}[n]$ denote the $N_t \times 1$ transmit vector for n -th time slot. There are $2^{\eta_{SM}}$ possible activation patterns of $\mathbf{s}[n]$ denoted as \mathbf{s}_{i_s} where $i_s = \{1, \dots, 2^{\eta_{SM}}\}$. Therefore, the received real-valued signal vector at the receiver for n -th time slot can be expressed as

$$\mathbf{y}[n] = r\mathbf{H}[n]\mathbf{s}[n] + \eta[n], \quad (1)$$

where r is the responsivity of the PD, $\mathbf{H}[n]$ is the $N_r \times N_t$ optical channel matrix for n -th time slot, $\eta[n]$ is the noise vector of dimension $N_r \times 1$. The (k, l) -th element of the $\mathbf{H}[n]$ matrix contains the channel gain from l -th LED to the k -th PD, $h_{k,l}[n]$. However, we assume that channel doesn't vary for a block of k_c small number of time slots and, hence, $h_{k,l}[n] = h_{k,l}^{i_c}$ for $n = \{(i_c - 1)k_c + 1, \dots, i_c k_c\}$ where i_c is the block number. Each element of $\eta[n]$ is distributed as real additive white Gaussian noise with zero mean and variance σ_η^2 .

A. Channel Model

We assume that the channel gain from a LED transmitter to a PD receiver is determined through line-of-sight (LoS) geometric loss which is deterministic for a fixed LED/PD. The LoS channel gain can be evaluated with the help of the well known Lambertian radiation pattern equation as [7]

$$h_{k,l}^{i_c} = \frac{(m+1)A}{2\pi(l_{k,l}^{i_c})^2} \cos^m(\theta_{k,l}^{i_c}) \cos(\phi_{k,l}^{i_c}) Gc(\phi_{k,l}^{i_c}), \quad (2)$$

where $m = -\ln(2)/\ln(\cos(\theta_{1/2}))$ is the Lambertian order of the LED, A is the aperture area of the PD, $l_{k,l}^{i_c}$ is the Euclidean distance between the l -th LED and the k -th PD, and $\theta_{1/2}$ is the half power angle of the LED. $\theta_{k,l}^{i_c}$ is the irradiance angle of the (l, k) -th LED/PD pair, $\phi_{k,l}^{i_c}$ is the incident angle of the (l, k) -th LED/PD pair, G is the gain of the trans-impedance amplifier (TIA) at the receiver, and $c(\phi_{k,l}^{i_c})$ is the concentration gain expressed as

$$c(\phi_{k,l}^{i_c}) = \begin{cases} \frac{\rho^2}{\sin^2(\Phi_{FoV})}, & 0 \leq \phi_{k,l}^{i_c} \leq \Phi_{FoV}, \\ 0, & \phi_{k,l}^{i_c} > \Phi_{FoV}, \end{cases} \quad (3)$$

where ρ is the internal refractive index of the concentration and Φ_{FoV} is the field-of-view (FoV). It is noted that $\theta_{k,l}^{i_c}$, $\phi_{k,l}^{i_c}$ and $l_{k,l}^{i_c}$ in (2) vary with the time due to the user mobility and random receiver orientation as explained in the sequel.

B. User Mobility and Random Receiver Orientation

The user mobility and the random receiver orientation have direct relationships with the azimuth angle, α^{i_c} and the polar angle, β^{i_c} of the receiver. In an indoor environment, the variation of β^{i_c} is faster than the walking speed. Hence, we select the value of k_c such that the difference between β^{i_c} and β^{i_c+1} is less than a pre-defined threshold.

1) *User Mobility*: We consider the random way-point based model in [7] and [9] to account for user mobility. The key concept of this model is to consider a set of random points and fill the intermediate points to obtain a walking path. Such intermediate points can be accurately modeled using a walking human trajectory model as presented in [16]. In our work, a set of random points, $\mathbf{p}_{i_p} = (x_{i_p}, y_{i_p})$, where $i_p = \{1, \dots, N_p\}$, and N_p is the number of points which lie on the floor area were selected. Next, the path for the movement between points \mathbf{p}_{i_p} and \mathbf{p}_{i_p+1} was obtained by solving the optimization problem in [16] that can be expressed as

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot), T_{i_p+1}} \int_{T_{i_p}}^{T_{i_p+1}} \phi_f(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (4)$$

under the constraints:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) & \text{Dynamic constraint} \\ \mathbf{x}(T_{i_p}) = \mathbf{x}_{i_p} & \text{Initial condition} \\ \mathbf{x}(T_{i_p+1}) = \mathbf{x}_{i_p+1} & \text{Final condition} \end{cases}$$

where $\mathbf{x}(t) = (x(t), y(t), \alpha(t), v_{forw}, v_{orth}, \omega)^T$ and $\mathbf{u}(t) = (u_1(t), u_2(t), u_3(t))^T$ are the state and control variables. In here, $(x(t), y(t))$ is the coordinate of the user at time t , $\alpha(t)$ is the azimuth angle at time t , v_{forw} is the velocity for the forward direction, v_{orth} is the velocity for the orthogonal direction, ω is the angular velocity, T_{i_p} is the travel time to the i_p -th point, $\phi_f(\mathbf{x}(t), \mathbf{u}(t))$ is the objective function, respectively. $\phi_f(\mathbf{x}(t), \mathbf{u}(t))$ is given by [16]

$$\begin{aligned} \phi_f(\mathbf{x}(t), \mathbf{u}(t)) = & a_0 + a_1 u_1^2(t) + a_2 u_2^2(t) + a_3 u_3^2(t) \\ & + a_4 \psi(\mathbf{x}(t), \mathbf{x}(T_{i_p+1}))^2 + a_5 ((x(T_{i_p+1}) - x(t))^2 + \\ & (y(T_{i_p+1}) - y(t))^2) + a_6 (\alpha(T_{i_p+1}) - \alpha(t))^2, \end{aligned} \quad (5)$$

where $a_0 = 1$, $a_1 = 1.2$, $a_2 = 1.7$, $a_3 = 0.7$, $a_4 = 5.2$, $a_5 = 5$, $a_6 = 8$ in our system. The dynamic equation of the locomotion system is $\dot{\mathbf{x}}(t) = (\dot{x}(t), \dot{y}(t), \dot{\alpha}(t), \dot{v}_{forw}, \dot{v}_{orth}, \dot{\omega})^T$ where $\dot{x}(t) = \cos(\alpha(t))v_{forw} - \sin(\alpha(t))v_{orth}$, $\dot{y}(t) = \sin(\alpha(t))v_{forw} + \cos(\alpha(t))v_{orth}$, $\dot{\alpha}(t) = \omega$, $\dot{v}_{forw} = u_1$, $\dot{v}_{orth} = u_2$, $\dot{\omega} = u_3$. In this model, $(T_{i_p+1} - T_{i_p})$ is not fixed due to variable time of movement between two random points. To obtain T_{i_p+1} for each \mathbf{p}_{i_p+1} point, an iterative search can be used. We start by setting

$$T_{i_p+1} - T_{i_p} = \frac{\sqrt{(x(T_{i_p+1}) - x(T_{i_p}))^2 + (y(T_{i_p+1}) - y(T_{i_p}))^2}}{v_n}, \quad (6)$$

where v_n is the nominal speed. Next, we increase $(T_{i_p+1} - T_{i_p})$ until $\phi_f(\mathbf{x}(t), \mathbf{u}(t))$ is smaller than a pre-defined threshold. Using this mobility model, the azimuth angle $\alpha(t)$, and user coordinates $(x(t), y(t))$ for time t are recorded. Such signals

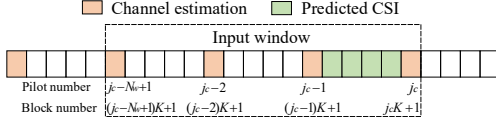


Fig. 2: Time frame of the proposed channel estimation and prediction protocol.

are sampled at $i_c k_c T_s$ where T_s is the time duration between two time slots to obtain the user coordinates, (x^{i_c}, y^{i_c}) and azimuth angle, α^{i_c} throughout the movement.

2) *Random Receiver Orientation*: To model random receiver orientation several techniques are used in the literature. In [10], the polar angle is assumed to be distributed in a uniform distribution. In [9], a measurement based realistic model has been presented. Based on the measurements the polar angle of the mobile device is independent of the azimuth angle and the user position. According to the model, the polar angle of the receiver for walking, β^{i_c} follows a Gaussian distribution and adjacent samples are correlated. A correlated Gaussian random process (RP) can be used generate samples of the polar angle. We use a first order linear auto-regressive (AR) model to generate the i_c -th sample of the correlated Gaussian RP, β^{i_c} , which is expressed as

$$\beta^{i_c} = c_0 + c_1 \beta^{i_c-1} + w^{i_c}, \quad (7)$$

where c_0 is the bias level, c_1 is the constant factor of the AR, and w^{i_c} is the white Gaussian noise with variance σ_w^2 [9]. The parameters of the AR model can be calculated using

$$c_0 = (1 - c_1) \mathbb{E}[\beta], \quad c_1 = \mathbb{R}_\beta \left(\frac{T_{c,\beta}}{T_s} \right)^{\frac{T_s}{T_{c,\beta}}}, \quad (8)$$

and $\sigma_w^2 = (1 - c_1^2) \sigma_\beta^2$, where $\mathbb{E}(\cdot)$ denotes the mean, $\mathbb{R}_\beta(\cdot)$ is the auto-correlation function of β^{i_c} , $T_{c,\beta}$ is the coherence time, and σ_β^2 is the variance of β^{i_c} , respectively.

The position of k -th PD can be obtained with 3D translations and rotations around the y and x -axes of the receiver's local coordinate frame. Hence, the coordinates of the k -th PD is

$$\begin{bmatrix} x_k^{i_c} \\ y_k^{i_c} \\ z_k^{i_c} \end{bmatrix} = \begin{bmatrix} x_{k,0} \cos(\alpha^{i_c}) \cos(\beta^{i_c}) - y_{k,0} \sin(\alpha^{i_c}) + x^{i_c} \\ x_{k,0} \sin(\alpha^{i_c}) \cos(\beta^{i_c}) + y_{k,0} \cos(\alpha^{i_c}) + y^{i_c} \\ -x_{k,0} \sin(\beta^{i_c}) \end{bmatrix},$$

where $(x_{k,0}, y_{k,0})$ is the coordinate of the k -th PD in the receiver's local coordinate frame. Next, $\theta_{k,l}^{i_c}$, $\phi_{k,l}^{i_c}$, and $l_{k,l}^{i_c}$ in (2) can be calculated as

$$\theta_{k,l}^{i_c} = \tan^{-1} \left(\frac{x_k^{i_c} - x_l}{z_k^{i_c} - z_l} \right), \quad (9)$$

$$\cos(\phi_{k,l}^{i_c}) = \left(\frac{x_l - x_k^{i_c}}{l_{k,l}^{i_c}} \right) \sin(\alpha^{i_c}) \cos(\beta^{i_c}) + \left(\frac{y_l - y_k^{i_c}}{l_{k,l}^{i_c}} \right) \sin(\alpha^{i_c}) \sin(\beta^{i_c}) + \left(\frac{z_l - z_k^{i_c}}{l_{k,l}^{i_c}} \right) \cos(\alpha^{i_c}), \quad (10)$$

and

$$l_{k,l}^{i_c} = \sqrt{(x_k^{i_c} - x_l)^2 + (y_k^{i_c} - y_l)^2 + (z_k^{i_c} - z_l)^2}, \quad (11)$$

where (x_l, y_l, z_l) is the coordinates of the l -th LED.

C. Channel Estimation and Detection

In this work, we present a channel estimation and prediction protocol as shown in Fig. 2. According to the protocol, pilot symbols are transmitted at an interval of K blocks or $K k_c T_s$ seconds. At the receiver, LS or MMSE channel estimation is used to obtain the CSI at pilot instances. Let

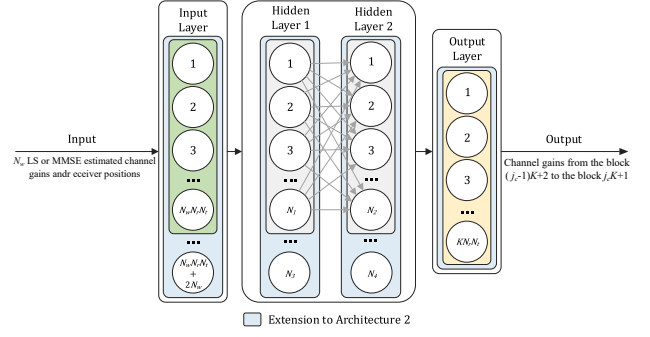


Fig. 3: Proposed ANN architecture.

$\mathbf{S}_p^{j_c}$ be the pilot matrix of the dimension $N_t \times k_c$ which contains the known pilot symbols for j_c -th pilot instance where $j_c = \lfloor (i_c - 1)/K + 1 \rfloor$, and $\mathbf{Y}_p^{j_c}$ be the received signal matrix of $N_r \times k_c$. The estimated channel gain matrix using LS estimation can be expressed as

$$\hat{\mathbf{H}}_{LS}^{j_c} = \mathbf{Y}_p^{j_c} (\mathbf{S}_p^{j_c})' \left(\mathbf{S}_p^{j_c} (\mathbf{S}_p^{j_c})' \right)^{-1}, \quad (12)$$

where $\hat{\mathbf{H}}_{LS}^{j_c} \in \mathbb{R}^{N_r \times N_t}$ is the estimated channel matrix [12], [13]. Similarly, the channel gain matrix for MMSE is

$$\hat{\mathbf{H}}_{MMSE}^{j_c} = \mathbf{R}_H \left(\mathbf{R}_H + \sigma_\eta^2 (\mathbf{S}_p^{j_c} (\mathbf{S}_p^{j_c})')^{-1} \right)^{-1} \hat{\mathbf{H}}_{LS}^{j_c}, \quad (13)$$

where \mathbf{R}_H is the co-variance matrix of the channel [11], [12]. After LS and MMSE estimation, channel gains for the blocks from $((j_c - 1)K + 1)$ to $(j_c K + 1)$ are predicted using the ANN architectures presented in Sec. III. Hence, the estimated and predicted channel gain matrix for the n -th time slot, $\hat{\mathbf{H}}_j[n]$ can be obtained where $j \in \{LS, MMSE\}$ denotes the channel estimation method.

We use ML detection to decode the received message stream [3], i.e.,

$$\hat{s}[n] = \arg \min_{s_i} \|\mathbf{y}[n] - r \hat{\mathbf{H}}_j[n] \mathbf{s}_i\|^2. \quad (14)$$

III. ANN BASED CHANNEL ESTIMATION AND PREDICTION

In this section, we present two ANNs that make use of CSI and user position information, to obtain higher resolution channel estimates. Recall that in our system, pilots are sent at a block interval of K followed by estimating the channels at these pilot positions using methods such as LS or MMSE. For this purpose, a window comprising N_w pilot instances are considered as shown in Fig. 2. The estimated CSI and predicted device positions from $(j_c - N_w + 1)$ to j_c pilot instances are used as inputs to the proposed ANNs. The output of ANNs are the predicted values of the CSI for blocks from $((j_c - 1)K + 2)$ to $(j_c K + 1)$.

A. Proposed ANN Architecture

Fig. 3 shows the ANN Architecture (referred to as ANN 1), which has an input layer, two hidden layers, and an output layer. The input layer consists of $N_w \times N_r \times N_t$ neurons, the values of which are taken from N_w estimated channel matrices of the input window. We adopt two fully connected hidden layers with N_1 , and N_2 neurons in each layer. The widely used rectified linear unit (ReLU) function $f_{ReLU}(x) = \max(0, x)$ is used as the activation function. We denote the output for the hidden layers as $\hat{\mathbf{x}}^{(2)} \in \mathbb{R}^{N_1}$, and $\hat{\mathbf{x}}^{(3)} \in \mathbb{R}^{N_2}$ and can be written as $\hat{\mathbf{x}}^{(2)} = f_{ReLU}(\mathbf{W}_1 \hat{\mathbf{x}}^{(1)} + \mathbf{k}_1)$, and $\hat{\mathbf{x}}^{(3)} =$

$f_{ReLU}(\mathbf{W}_2 \hat{\mathbf{x}}^{(2)} + \mathbf{k}_2)$, where $\mathbf{W}_1 \in \mathbb{R}^{N_1 \times (N_w \times N_r \times N_t)}$ and $\mathbf{W}_2 \in \mathbb{R}^{N_2 \times N_1}$ are the weight vectors for the first and second hidden layers, $\mathbf{k}_1 \in \mathbb{R}^{N_1}$ and $\mathbf{k}_2 \in \mathbb{R}^{N_2}$ are the bias vectors for the first and second hidden layers, and $\hat{\mathbf{x}}^{(1)}$ is input to the first layer, respectively. The output layer, which consists of $K \times N_r \times N_t$ neurons, is used to obtain the predicted K channel gains from $((j_c - 1)K + 2)$ to $(j_c K + 1)$ blocks. No activation function is used at the output layer. Hence, the output can be expressed as $\hat{\mathbf{y}} = \mathbf{W}_o \hat{\mathbf{x}}^{(3)}$, where $\mathbf{W}_o \in \mathbb{R}^{(K \times N_r \times N_t) \times N_2}$ is the weight vector.

B. ANN Architecture with User Position Information

In addition to the estimated CSI, prior knowledge of the receiver position can be used to refine the channel prediction. This key idea has been incorporated in extension to the ANN architecture (referred to as ANN 2) as shown in Fig. 3. Let position and azimuth angle of the device given by $\bar{\mathbf{c}} = (\bar{x}(t), \bar{y}(t), \bar{\alpha}(t))^T$, are recorded with a sampling interval of T_R at time instances T_m , where $(T_{m+1} - T_m) = T_R > K T_s$. $\bar{\mathbf{c}}$ is piece wise continuous on each $[T_m, T_{m+1}]$ interval. The traveled trajectory of the device is given by $(\bar{\mathbf{c}}_0, \bar{\mathbf{c}}_1, \dots, \bar{\mathbf{c}}_m)^T$. Considering $\bar{\mathbf{c}}_m$ as the initial location and $\bar{\mathbf{c}}_{m+1}$ as the final location, intermediate device positions are predicted exploiting the path prediction technique in [17]. Next, it is sampled at $K k_c T_s$ to obtain device locations for $(j_c - N_w + 1)$ to j_c pilot instances. The N_w number of predicted (x, y) coordinates are used as the additional inputs to the ANN 2. Accurate path prediction can be realized using techniques such as video surveillance cameras and wall/floor mounted sensors leading to a rich statistical analysis of the movement data. Our ANN architectures are sufficiently flexible, for example the training phase can easily accommodate all real-world data sets.

In ANN 2, the input layer consists of $(N_w \times N_r \times N_t + 2 \times N_w)$ neurons where additional $2 \times N_w$ neurons than ANN 1 are used to provide predicted (x, y) coordinates of the device at N_w pilot instances. Two hidden layers consists of N_3 and N_4 neurons in each layer. It is noted that a better result can be obtained, even with $N_3 < N_1$ and $N_4 < N_2$ due to the use of device location prediction. The ReLU function is used as the activation function. Similar to ANN 1, $K \times N_r \times N_t$ neurons are used at the output layer.

In the training phase of both ANNs, initial values for the network weights, $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_o, \mathbf{k}_1, \mathbf{k}_2\}$ are set. Forward propagation is applied to obtain $\hat{\mathbf{y}}$ for a selected input. In this letter, the mean squared error between $\hat{\mathbf{y}}$ and a vector containing perfect CSI for K blocks, \mathbf{y}_0 is used as the loss function which is expressed as $\phi(\theta) = \frac{1}{B} \sum_{i_m=1}^B \|\hat{\mathbf{y}} - \mathbf{y}_0\|^2$, where B is the mini batch size of the training. An optical power measurement technique can be used to measure received power, and hence, \mathbf{y}_0 can be calculated. Hardware imperfections may introduce random effects for channel gains such as in optical OFDM, while in this work, training data is processed with the help of (2) under the assumption that the VLC channel gains are deterministic. θ is updated for training batches using the stochastic gradient descent algorithm that can be expressed as $\theta^+ := \theta - \epsilon \nabla \phi(\theta)$, where ϵ is the learning rate. Note that for training we use the Adam optimizer to minimize the loss

Algorithm 1 Proposed algorithm for ANN 2

Inputs:

Initial conditions, $\mathbf{x}(0)$
 Final conditions, $\mathbf{x}(T_f)$
 Trained ANN model for channel prediction

Initialize:

$\mathbf{x}(T_{i_p}) = \mathbf{x}(0)$

1: repeat

- 2: Predict $\mathbf{x}(K k_c T_s((j_c - N_w + 1)K + 1))$ to $\mathbf{x}(K k_c T_s(j_c K + 1))$
 - 3: Estimate channel gains at N_w pilot instances using LS or MMSE
 - 4: Predict channel gains from $((j_c - 1)K + 2)$ to $(j_c K + 1)$ intermediate time slots using ANN 2
 - 5: **until** $\mathbf{x}[n] = \mathbf{x}(T_f)$
 - 6: **return** Estimated channel gains matrices, $\hat{\mathbf{H}}_j[n]$
-

function. The algorithm that performs channel prediction using this architecture is shown in Algorithm 1.

ANN 1 and ANN 2 involves a trade-off between complexity and performance. In terms of complexity, ANN 1 requires $(2N_w N_r N_t N_1 + 2N_1 N_2 + 2N_2 K N_r N_t)$ floating point operations (flops) to produce a set of CSI outputs while it is $(2(N_w N_r N_t + 2N_w)N_3 + 2N_3 N_4 + 2N_4 K N_r N_t)$ flops for ANN 2. Moreover, the latter requires the device position prediction which increase the complexity.

C. Spline Interpolation

In order to benchmark the performance of ANN based CSI estimation, we compare the results with an existing interpolation method in which spline interpolation is used to obtain the intermediate $(K - 1)$ blocks [15]. Accordingly, the spline interpolation for each (k, l) -th element of the channel gain matrix $\hat{\mathbf{H}}_j[n]$, $h_{k,l}[n]$ is done separately. Depending on the value of N_w , the order of the spline polynomial can be selected. The interpolation polynomial of the order N_w is expressed as

$$f_{N_w}(h_{ij}(t)) = \sum_{i=0}^{N_w} a_i h_{ij}(t)^i, \quad (15)$$

where $a_i, \forall i \in \{1, N_w\}$ are constants. The spline polynomial should pass through N_w points in the window. Hence, an N_w set of equations can be formed to find a_i . Optimal a_i values can be found for example using MATLAB[®] software. Further, $h_{ij}(t)$ is sampled at T_s to obtain $h_{k,l}[n]$.

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, numerical results are presented to illustrate the performance of the ANNs under different system and channel parameters. A room dimension of $5\text{m} \times 5\text{m} \times 3\text{m}$ is assumed. Unless stated explicitly otherwise in all simulations, we have set the parameter values as $\rho = 1.25$, $A = 10^{-3}\text{m}^2$, $\Phi_{FOV} = 90^\circ$, $\theta_{1/2} = 60^\circ$, $r = 0.5\text{A/W}$, $T = 1$, $N_t = 4$, $N_r = 4$, $P_t = 1\text{W}$, $N_w = 5$, $K = 10$, $T_c = 13\text{ms}$, $N_1 = 20$, $N_2, N_3 = 20$, $N_4 = 20$, $k_c = 10$, and SNR of 25 dB at pilot instances, respectively [7], [9]. A set of 10^7 channel matrices were generated. A series of simulations were conducted to obtain suitable percentages for training, validation, and testing and finally are selected as 40%, 20%, and 40%. The evaluation is performed by MATLAB and Python running in a Acer Aspire E5-574G-58WK with a 2.3 GHz Intel Core i5 CPU and a NVIDIA GeForce 920M GPU. The average execution times for ANN 1 and ANN 2 are 7.52 μs and 10.34 μs , respectively.

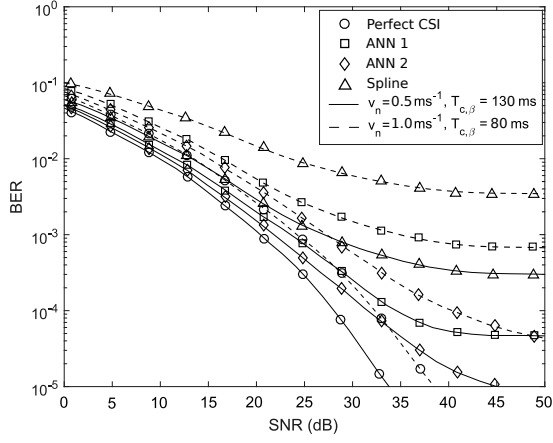


Fig. 4: BER versus SNR with ANN 1 and ANN 2.

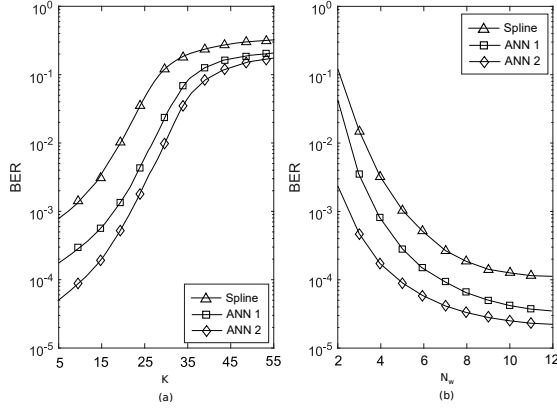


Fig. 5: (a) BER versus K . (b) BER versus N_w .

Fig. 4 shows the BER performance with ANNs, perfect CSI, and spline interpolation. These results corresponds to two mobility and receiver orientation conditions. $v_n = 1.0 \text{ ms}^{-1}$ and $T_{c,\beta} = 80 \text{ ms}$ corresponds to high mobility while $v_n = 0.5 \text{ ms}^{-1}$ and $T_{c,\beta} = 130 \text{ ms}$ is the low mobility case. Both ANN 1 and ANN 2 are capable of achieving a significant performance at moderate to high SNR when compared with the case of perfect CSI. Moreover, using the ANNs, the irreducible error floor due to imperfect channel estimation can be considerably reduced. At low mobility, both ANNs show a near optimal performance while there is a performance deterioration at high mobility. However, ANN 2 achieves a lower BER performance than ANN 1 and spline interpolation in both mobility conditions.

Fig. 5(a) shows the BER versus K , when $N_w = 5$ and MMSE is used. A lower K value results in more accurate channel prediction and, hence a lower BER can be observed. In the considered setup, a BER better than 10^{-3} can be observed when $K < 15$ using both ANNs. In order to obtain 10^{-3} BER, the K value needs to be 6, 16, and 21 for in the cases of spline, ANN 1 and ANN 2, respectively. Fig. 5(b) shows the BER versus N_w , when $K = 10$ and MMSE is used. A higher value of N_w increases the prediction accuracy which in turn reduces the BER. Presented results are helpful to select a practical value for N_w . In our setup, a BER below 10^{-3} can be achieved when $N_w \geq 5$. The gap between ANNs reduces and settles to a constant value as N_w increases which shows the ANN 1 is able to mitigate the absence of position information

to some extent. Hence, ANN 2 is well suited for lower N_w values.

V. CONCLUSION

In this letter, we have considered channel estimation and performance evaluation of SM aided indoor VLC systems under mobility and receiver orientation. Specifically, two ANNs which use estimated CSI at pilot instances and user position information to obtain CSI with higher resolution have been proposed. Our ANNs are capable of achieving a low BER with reduced amount of pilot symbols as compared to existing solutions. For the considered setup, using our ANN solutions reduces the number of pilots required by 71%. Moreover, the presented ANN solution with user position information is robust against a wide range of user mobility cases and receiver orientations without much performance degradation, hence suitable for VLC applications.

REFERENCES

- [1] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE Commun. Surv. Tutor.*, vol. 17, pp. 2047–2077, Sep. 2015.
- [2] M. Di Renzo, H. Haas, A. Ghrayeb, S. Sugiura, and L. Hanzo, "Spatial modulation for generalized MIMO: Challenges, opportunities, and implementation," *Proceedings of the IEEE*, vol. 102, pp. 56–103, Jan. 2014.
- [3] S. P. Alaka, T. L. Narasimhan, and A. Chockalingam, "Generalized spatial modulation in indoor wireless visible light communication," in *Proc. IEEE GLOBECOM 2015*, Dec. 2015, pp. 1–7.
- [4] J.-Y. Wang *et al.*, "Adaptive spatial modulation for visible light communications with an arbitrary number of transmitters," *IEEE Access*, vol. 6, pp. 37 108–37 123, Jun. 2018.
- [5] Y. Li *et al.*, "Precoded optical spatial modulation for indoor visible light communications," *IEEE Trans. Commun.*, vol. 69, pp. 2518–2531, Apr. 2021.
- [6] T. Cogalan, H. Haas, and E. Panayirci, "Optical spatial modulation design," *Philos. Trans. Royal Soc. A*, vol. 378, Mar. 2020.
- [7] M. D. Soltani *et al.*, "Bidirectional optical spatial modulation for mobile users: Toward a practical design for LiFi systems," *IEEE J. Sel. Areas Commun.*, vol. 37, pp. 2069–2086, Aug. 2019.
- [8] Y. Chen, S. Gao, G. Tu, and D. Chen, "Group-based led selection for generalized spatial modulation in visible light communication," *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 3022–3026, Sep. 2021.
- [9] M. D. Soltani, A. A. Purwita, Z. Zeng, H. Haas, and M. Safari, "Modeling the random orientation of mobile devices: Measurement, analysis and LiFi use case," *IEEE Trans. Commun.*, vol. 67, pp. 2157–2172, Mar. 2019.
- [10] Y. Yapıcı and s. Güvenç, "NOMA for VLC downlink transmission with random receiver orientation," *IEEE Trans. Commun.*, vol. 67, pp. 5558–5573, Aug. 2019.
- [11] X. Chen and M. Jiang, "Adaptive statistical bayesian MMSE channel estimation for visible light communication," *IEEE Trans. Signal Process.*, vol. 65, pp. 1287–1299, Mar. 2017.
- [12] Y. S. Hussein, M. Y. Alias, and A. A. Abdulkafi, "On performance analysis of LS and MMSE for channel estimation in VLC systems," in *Proc. IEEE CSPA 2016*, 2016, pp. 204–209.
- [13] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing least square channel estimation using deep learning," in *Proc. IEEE VTC-Spring 2020*, May 2020, pp. 1–5.
- [14] T. Wang, F. Yang, and J. Song, "Deep learning-based detection scheme for visible light communication with generalized spatial modulation," *Opt. Express*, vol. 28, pp. 28 906–28 915, Sep. 2020.
- [15] O. Şaylı, H. Doğan, and E. Panayirci, "Spline interpolation based channel estimation for ACO-OFDM over visible light channels," in *Proc. IEEE SIU 2016*, May 2016, pp. 333–336.
- [16] I. Maroger, O. Stasse, and B. Watier, "Walking human trajectory models and their application to humanoid robot locomotion," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Syst. (IROS)*, Jan. 2020, pp. 3465–3472.
- [17] I. Maroger, N. Ramuzat, O. Stasse, and B. Watier, "Human trajectory prediction model and its coupling with a walking pattern generator of a humanoid robot," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 6361–6369, Oct. 2021.