Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us
what having access to this work means to you and why it's important to you. Thank you.

An Adaptive DPPM for Efficient and Robust Visible Light Communication Across the Air-Water Interface

Md Shafiqul Islam
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, Maryland, USA
mdislam1@umbc.edu

Mohamed Younis
Department of Computer Science and Electrical
EngineeringUniversity of Maryland Baltimore County
Baltimore, Maryland, USA
younis@umbc.edu

Abstract—The scarcity of the optical power is the main challenge for underwater visible light communication. It becomes worst for communication across the air-water interface because of the reflection of light from the air-water interface. Differential pulse position modulation (DPPM) is one of the power efficient modulation techniques. In L-DPPM a block of $M = log_2 L$ input data is mapped into one of the L distinct waveforms containing only one 'on' chip. The size of the DPPM packet is variable and depends on the value of input data and L, which makes error detection quite challenging. In this paper, we propose a frame structure that efficiently enables error detection within a packet for various symbol length, L, of DPPM. We also propose an algorithm using such a frame structure to enable effective detection of packet errors and for adaptively changing the value of L for optimal power efficiency while meeting a certain bound on the packet error rate (PER). We have named our proposed protocol as adaptive differential pulse position modulation (ADPPM). The Bit rate and PER have been studied for different signal-to-noise ratio (SNR) through simulation. A comparison between ADPPM and OOK, DPPM with fixed L is provided.

Keywords—Differential pulse position modulation; Underwater networks; Free space optics; Visible light communication.

I. INTRODUCTION

Underwater communication is gaining increased attention from the research community due to its vast applications such as oceanic studies, search and rescue, sea floor observation, and security surveillance, etc. In many of these applications, it is necessary to provide access to the underwater node from ground-based remote center. However, there are no physical signals which work smoothly in both air and water medium. Therefore, interaction with remote centers is supported through a gateway which is placed in the water surface. Such a gateway will employ two distinct transceivers, one for reaching underwater nodes, e.g., using acoustic or optical links, and another radio-based to reach the ground center. Generally deploying surface nodes is logistically challenging, lacks responsiveness, and is undesirable in security-sensitive applications. Alternatively, the gateway can be airborne where visible light communication (VLC) is pursued. Light signal propagates well in both air and water. However, with the increase of underwater depth visible light's signal becomes weaker. Therefore, an efficient modulation technique is crucial in order to support reliable communication and maximize the underwater reach from the airborne unit. In our previous work, we studied and analyzed coverage area and intensity of visible light through the air water interface [1]. In this paper we leverage such work to find the most energy efficient modulation technique for air to underwater communication.

There are various modulation techniques for optical communication. For simplicity, underwater optical communication usually uses intensity-based modulation with direct detection technique (IM / DD). The most common modulation technique is on-off keying (OOK) with NRZ or RZ encoding [2][3][4]. Although the bandwidth efficiency and bit rate are very high using OOK-NRZ or OOK-RZ, power efficiency is not good. In underwater environments, power efficiency is very crucial for optical communication due to the absorption and scattering loss of the optical signal. The pulse position modulation (PPM) is one of the most popular power efficient techniques for optical communication [5][6]. In PPM, each M bits are sent over a symbol $L = 2^M$ time chips and only one pulse is sent in L for the chip position, corresponding to the value of the M bits. However, PPM requires very accurate clock synchronization between the transmitter and receiver, which is quite challenging in underwater environments. Also, PPM is not bandwidth efficient, as discussed in detail later in the paper.

In order to achieve better bandwidth efficiency, a number of modified versions of PPM have been proposed, such as overlapping PPM (OPPM) [7][8], multiple PPM (MPPM), differential PPM (DPPM) [9][10], pulse-interval modulation (DPIM) [11][12], and dual-header pulse-interval modulation (DH–PIMα) [13]. Among these PPM variants, we focus on DPPM in this paper. DPPM starts the next symbol after sending the pulse, i.e., before the elapse of the remaining time chips of the symbol L. Thus, in DPPM the transmitter and receiver do not need to have tightly synchronized clocks. Nonetheless, the bit error detection in DPPM is very complex as the number of chips in a frame are variable after modulation, i.e., the frame size is not constant. Only few studies have focused on error detection for DPPM. In [14], a marker code is used to identify insertion/deletion errors and a Reed–Solomon code is used to correct burst errors and erasures. Yet, such maker code requires a fixed M, and thus will not work if the value of M changes.

In this paper, we introduce a new variant of DPPM to suit communication across the air-water interface. We show that the symbol size L affects both reliability and bandwidth efficiency. We argue that by varying the value of M in DPPM, we can control the packet error rate (PER) and implicitly bit rate of DPPM, and trade it off with the bandwidth of a VLC link. To

enable the use of different values of *M*, we design and analyze a novel frame structure for DPPM which contains the information of *M* so that a receiver can detect the errors in the frame. As information about *M* is embedded with the transmitted frame, a transmitter can change the value of *M* any time without hindering correct reception. Using our proposed frame, we further develop an algorithm to transmit data from an airborne node to an underwater node while changing the value of *M* depending on the packet error rate. We name this as adaptive differential pulse position modulation (ADPPM). To the best of our knowledge, no prior work has studies VLC modulation for communication across two mediums. Specifically, the contributions of this paper are: 1) propose a novel ADPPM frame structure to enable tradeoff between bandwidth and error rate, 2) design algorithms to create and decode such a frame, 3) develop an algorithm to dynamically change the value of *M* based on certain PER criteria, and 4) analyze the bit rate and PER for various underwater depth.

The paper is organized as follows. In section II, system model is discussed. Section III presents our proposed adaptive DPPM approach. Section IV presents the validation results. The paper is concluded in Section V.

II. SYSTEM MODEL AND PRELIMINARIES

A. Channel Model

The objective of this paper is to develop a suitable modulation technique for communication through the air-water interface. Figure 1 illustrates the application scenario. A light transmitter with beam angle, Θ is placed in the air at point *S* which is d_w meter above the water surface. Light is transmitted through the air-water interface and reaches at the coverage area, *CD*, where an underwater photodetector is placed to receive the modulated data from the source. This coverage area is located at d_w meter depth from the water surface. The light intensity at underwater point *C*, can be calculated as follows:

$$I = \frac{2\pi P}{4\pi\theta} \cdot \tau \cdot \frac{1}{SA^2} \cdot e^{-k\cdot(AC)} \qquad (1)$$

A detailed analysis of how to get the length of *SA* and *AC* and use them to calculate the light intensity for flat and wavy water surface can be found in [1]. This paper leverages such analysis to decode the received signal in the validation of our approach.

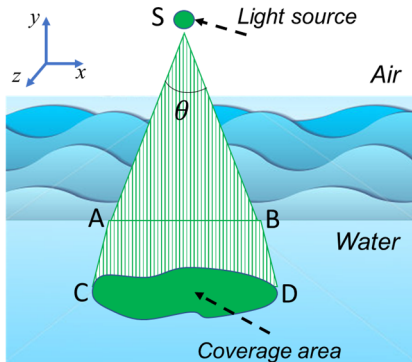While light travels across the air-water interface, the power of the light signal decreases due to reflection at the water surface and absorption and attenuation while propagating underwater. As the underwater distance d_w increases, the light intensity diminishes. By increasing the power of the transmitted light, we can increase the received signal strength in the underwater. However, it is not always possible to increase the transmission power due to application requirements and/or constraints imposed on the design of the airborne unit. In this paper we handle such a constraint by proposing a power-efficient and robust modulation technique. Our approach is based on PPM and DPPM, which are widely used power efficient modulation techniques. In the next subsection we will provide an overview for PPM and DPPM.

B. PPM and DPPM

In PPM, information is encoded in the pulse position; the pulse position depends on the value represented by the corresponding *M* input data. In L-PPM, a block of $M = log_2 L$ input data is mapped into one of the *L* distinct waveforms containing only one 'on' chip and the remaining L-1 chips are off, where *M* > 0. A pulse $p(t)$ is transmitted on that 'on' chip (time slot). Figure 2 explains the PPM with an example along with other modulation technique. In this example, actual data is 9 bits long. Here, we have chosen *M* =3 which means $L = 2^M = 8$; consequently, the main input data needs to be partitions into groups of 3 bits. In this example, the decimal value of first three input bits is 2, so the pulse position is also 2 in the first *L* time slots. In a similar way we can schedule that other pulses based on the decimal value of each group of 3 bits in the data payload.

PPM is a power efficient modulation technique because we are sending less 'on' pulses than other modulation techniques like OOK-NRZ, OOK-RZ, PWM. This is a key advantage for the energy constrained applications like those involving underwater wireless optical communication. However, the bandwidth efficiency of PPM is not as good as OOK because the symbol is longer, and more time is needed to transmit the same data than OOK. Another disadvantage of PPM is the need for very tight clock synchronization between the transmitter and receiver since accurate pulse positioning is crucial for successful reception in PPM. These two issues are addressed in DPPM, which is a modified version of PPM. DPPM improves power efficiency as well as bandwidth efficiency by removing the extra zeros after the pulse position. Figure 2 also shows the
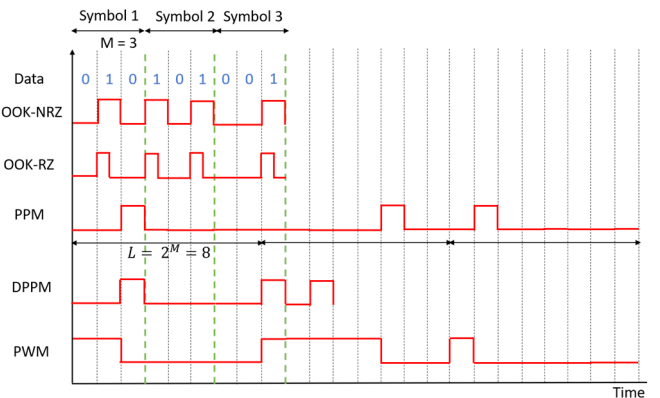

Fig. 1. 3D view of a coverage area inside the water

Fig. 2. Different modulation techniques for VLC communication

DPPM waveform, where the extra zeros after the pulse have been omitted from the PPM waveform. Thus, the average number of slots per symbol in DPPM is:

$$\bar{L}_{DPPM} = (L+1)/2 \qquad (2)$$

Such optimization enables DPPM to outperform PPM in terms of bandwidth efficiency, and also eliminate the need for tight clock synchronization. By knowing the time difference between two 'on' pulses, a receiver can extract the data.

Although DPPM's omission of zeros after a pulse improves the bandwidth efficiency, it makes the modulated data size variable. In Figure 2 we can see all modulated waveforms have the same size for a fixed length of data except DPPM, which has variable number of time slots in the waveform depending on the input data. This variable time slot count in the output waveforms makes it quite difficult to decode if there are any errors in the received waveform. Any insertion/deletion of a "1" chip in the modulated waveform causes wrong decoding. Therefore, the number of ones in the modulated waveform is very important to decode the received signal correctly.

Another key parameter of DPPM is M. For a fixed data size, a larger M would increase energy efficiency by sending fewer 'on' pulses, yet at the same time it would decrease the bandwidth efficiency because we need more time slots to send the same data. Depending on the communication properties we can change the setting of M to get better results. For example, if in Figure 1 the underwater depth, d_w, is very small, i.e., the light intensity at the receiver is high enough to establish a communication link, we can keep M as low as possible so that bandwidth efficiency remains high. On the other hand, a large d_w would dramatically diminish the detectability of the light signal at the receiver; in this case we can use a bigger value of M so that fewer 'on' pulses are sent with high power.

Thus, changing M dynamically is very important for the VLC links across the air-water interface. However, if the transmitter module changes the value of M dynamically, it will not be possible for the receiver to demodulate the DPPM data. In the next section we design a novel frame structure which contains the information of M so that the receiver module can decode the received signals. This frame structure also helps in detecting error within a frame. We call this adaptive DPPM (ADPPM) frame since it contains the information of M and changing the value of M will not hinder correct reception.

III. ADAPTIVE DPPM

In this section we first design a frame structure for the proposed ADPPM protocol and then develop an algorithm for fitting the encoded data in such a frame structure. We also show how this frame structure can be decoded. Finally, we propose an additional algorithm for determining the appropriate value of M, and consequently the symbol size L.

A. Frame Design

Figure 3 shows our proposed ADPPM frame structure. In this frame, the payload is the modulated version of the actual message data. If D is the data size, then the maximum and minimum payload can be calculated as follows:

$$P_{max} = 2^M \times \frac{D}{M} \quad \text{and} \quad P_{min} = \frac{D}{M} \qquad (3)$$

Fig. 3. ADPPM frame structure

Delimiters: A guard band, G, is added before and after the payload. At the end of the frame, another guard band is added to inform the receiver about the frame boundary; recall the frame size of DPPM is variable. Theoretically, the bit pattern for G could be anything as long as the transmitter and receiver know it. Obviously, a similar bit pattern may appear in the payload and additional information may be included to distinguish guards from data, as we explain below. Generally, the size of G, denoted $|G|$, is subject to tradeoff. Using a large bit pattern for G will diminish the probability of having a similar pattern with the payload, and vice versa. In our approach, if the same pattern appears in the payload, an additional zero is appended to such a pattern in the payload. For example, if G is picked to be '0110', any time the pattern '0110' appears in the payload, an additional zero is inserted in the middle making it '01010' instead. A field, called $EZPs$, is included in the frame to identify the added zeros due to matching the guard G within the payload. Basically, $EZPs$ contains the position of all those extra 0's. If ezp represents the position of only one extra 0, EZPs can be expressed as follows:

$$EZPs = ezp \times N_0 \qquad (4)$$

Where N_0 denotes the number of extra 0's. The size of each ezp depends on the P_{max} since it represents the maximum size of the payload. For example, if maximum size of payload, $P_{max} = 128$ bits, we need at least 7 bits to represent each extra 0 position. We can express this as follows:

$$ezp = \log_2(P_{max}) = \log_2\left(2^M \times \frac{D}{M}\right) \qquad (5)$$

Thus, the size of $EZPs$ varies depending on the value of N_0. The maximum value of N_0, denoted $N_{0,max}$, corresponds to the case that all consecutive groups of $|G|$ bits in the payload matches the bit pattern of G.

$$EZPs_{min} = 0 \quad and \quad EZPs_{max} = ezp \times N_{0,max} \qquad (6)$$

It is important to note that distinguishing the payload from G is essential only for the guard band after the payload in the frame since it can cause confusion about when the payload field actually ends. Therefore, overlapped matches of G within the payload is not warranted. Thus,

$$N_{0,max} = \left\lfloor \frac{D}{|G|} \right\rfloor \qquad (7)$$

Determining M: In addition to $EZPs$, we also introduce an extra field, namely, NOP to enable correct demodulation. NOP, which stands for number of 1's in the data payload. Such a field is necessary to inform the receiver about the value of M used by the transmitter. Assume that N_1 is the number of 1's that the data payload. Since DPPM sends only one "on" pulse for each M bits, N_1 and the size of NOP can be calculated as follows:

$$N_1 = \frac{D}{M} \qquad (8)$$

$$NOP = \log_2\left(\frac{D}{M}\right) + 1 \qquad (9)$$

The size of the NOP is fixed and is determined based on the maximum value of N_1, which corresponds to the minimum

value of M that will be used in communication. If D is fixed, by knowing the decimal value in the *NOP* field in the frame, the receiver can estimate the value of M from Eq. (8).

B. Protocol Efficiency

In the frame, anything other than the payload is called overhead. Thus, the overhead for our proposed frame structure is:

$$Overhead = 3|G| + NOP + EZPs \qquad (10)$$

Substituting *NOP* and *EZPs* from Eq. (4), and (9) we get:

$$Overhead = 3|G| + \log_2\left(\frac{D}{M}\right) + 1 + ezp \times N_0 \qquad (11)$$

By knowing the payload and overhead we can calculate the protocol efficiency, η of the communication system as follows:

$$\eta = \frac{Data\ size}{Payload + Overhead} \qquad (12)$$

This protocol efficiency, η is very important to understand the maximum value of message data, D that should send in one frame. It also tells us how effectively we can design a data frame. We want this as big as possible. The largest η is when the payload, and overhead is at minimum. From Eq. (3) we can get the minimum payload. Eq. (6) indicates that the overhead is minimum when $EZPs = 0$, i.e., when payload contain no guard pattern, G. Thus,

$$\eta_{max} = \frac{D}{\frac{D}{M} + 3|G| + \log_2\left(\frac{D}{M}\right) + 1} \qquad (13)$$

Meanwhile, the minimum value of η corresponds to maximum overhead and payload, and will be:

$$\eta_{min} = \frac{Data\ Size}{Payload_{max} + Overhead_{max}} \qquad (14)$$

Using Eq. (6), (7), and (11), the maximum overhead.

$$overhead_{max} = 3|G| + \log_2\left(\frac{D}{M}\right) + 1 + ezp \times \left\lfloor\frac{D}{|G|}\right\rfloor \qquad (15)$$

From Eq. (3) the maximum size of payload is $2^M \times \frac{D}{M}$. However, we need to note that the payload may contains zeros due to matches of G with the data bits; in the worst cases the bit pattern of the data is simply a sequence of matches of G and $N_{0,max}$ bits are added. Therefore, the maximum payload is
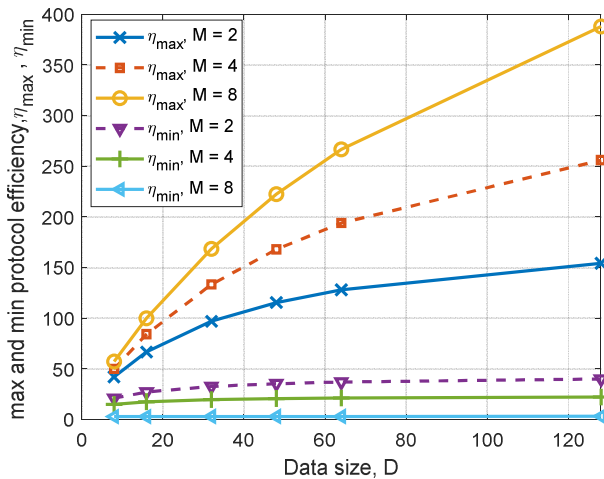

Fig. 4. Effect of data size, D and M on protocol efficiency, η

further updated to factor in the situation when N_0 is maximum, i.e., using Eq. (7). Thus,

$$Payload_{max} = 2^M \times \frac{D}{M} + \left\lfloor\frac{D}{|G|}\right\rfloor \qquad (16)$$

Substituting $overhead_{max}$ and $Payload_{max}$ from Eq. (6), (15) and (16) into Eq. (14), we can derive formula for η_{min}.

$$\eta_{min} = \frac{D}{3|G| + \log_2\left(\frac{D}{M}\right) + 1 + \log_2\left(2^M \times \frac{D}{M}\right)\left\lfloor\frac{D}{|G|}\right\rfloor + 2^M \times \frac{D}{M} + \left\lfloor\frac{D}{|G|}\right\rfloor} \qquad (17)$$

From eq. (13) and (17) we can note that the maximum and minimum protocol efficiency mainly depends on the message data length, D and M. Such a relation is captured in Figure 4. From this figure we can see that for large values of D, η_{min} and η_{max} do not depend that much on M. On the other hand, for smaller D, growing M boosts η_{max} and diminishes η_{min}. At this point we need to keep in mind that though higher D give us slightly better protocol efficiency, it also makes the frame size bigger which means if there is any error, we need to send that bigger frame again which eventually decreases the bandwidth efficiency. Therefore, we need to choose D as low as possible which also provides reasonable protocol efficiency. For example, if $D = 32$ bits, we get reasonable protocol efficiency; when $D < 32$ the maximum efficiency drops drastically. Hence, in the validation (Section IV) we choose $D = 32$ bits.

C. Algorithms and Illustraive Examples

Based on the frame design and analysis above, we develop two algorithms to: (i) generate the ADPPM frame structure at the sender, and (ii) demodulate the frame at the receiver. We will also propose another algorithm to dynamically change the value of M to maintain certain packet error rate (PER).

Algorithm 1: A pseudo code summary of the steps for creating an ADPPM frame is shown in Algorithm 1. We will explain such an algorithm using the example shown in Figure 5.

Step 1: The key parameters, namely, the size of input data, D, the guard pattern, G and the modulation index, M, are to be determined. In the example, $D = 32$ bits and $M = 4$, i.e., $L = 2^M = 16$. We also assume that $G = '0110'$.

Step 2: ezp and *NOP* are calculated based on eq. (6) and (10), respectively. In the example *NOP* = 4 bits and *ezp* = 7 bits.

Step 3: Apply DPPM to modulate on input data.

Step 4: Check the data for matches to the guard pattern, G. Anytime the pattern of G is encountered in the payload an extra 0 is added and EZPs are updated, as shown in Figure 5.

Input: Message data, D
Output: ADPPM frame
----
1. *initialize: M, D, G*
2. *set NOP = M/D*
3. *set bit size for each extra zero position (ezp)*
4. *payload = mod_data = DPPM modulation of the input data*
5. **if** *(mod_data contains G)*
6.   *add extra zero in the payload and update EZPs*
7. *ADPPM frame = G + payload + G + NOP + EZPs + G*

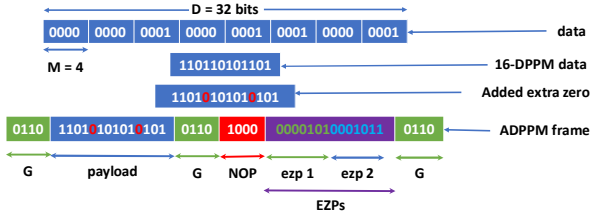**Algorithm 1.** Steps for generating an ADPPM frame

Fig. 5. An example ADPPM frame

Step 5: Form a frame following the format shown in Figure 3.

Algorithm 2: The outlined steps explain how a receiver can decode an ADPPM frame and extract the message data, *D*. Figure 6 illustrates the algorithm using an example.

Step 1: A receiver, Rx, looks first for the guard pattern, *G* within the frame. Since according to Figure 3, a transmitter, Tx, would have added *G* three times, Rx should find at least three guard patterns. Otherwise, the frame should be resent by Tx. On the other hand, since Tx added extra 0's in the payload, Rx should not receive more than three guard patterns. However, it may happen, because the *EZPs* field itself can contain a pattern that matches *G*. Thus, if Rx observes more than three instances of *G*, it does not conclude that an error has occurred.

Step 2: Rx will extract the payload and the control segment within the frame. The latter will further be divided into two parts, corresponding to *NOP* and *EZPs*.

Step 3: Depending on the decimal value of bits in the *EZPs*, extra 0's will be discarded from the payload. After discarding them, the payload will reflect DPPM modulated data.

Step 4: The number of 1's in the DPPM data should match the value of NOP. In the example *NOP* = 8. We can also see DPPM data contains 8 1's. If this two fields do not match there are errors in the packet. Given that the Rx knows that D=32, the value of *M* can be concluded by diving *D* by *NOP*.

Step 5: Rx will then demodulate the DPPM data to get the actual message data, *D*. While decoding Rx will use the value of *M* inferred from the previous step.

Algorithm 3: Leveraging Algorithms 1 and 2, this algorithm calculates PER and dynamically changes value of *M* depending to such PER. In Algorithm 3, the initial value of *M* may be decreased or increased depending on PER.

IV. VALIDATION RESULTS

Algorithms 1, 2 and 3 have been implemented in MATLAB to analyze PER, bit rate for different SNR and underwater depth, d_w. Figures 7 and 8 capture the effect of *M* in bit rate and PER calculation for different SNR. Figure 7 shows that if the value of *M* is increased the bit rate decreases. This is expected due to the fact that the modulated data size grows with the increase of *M,* which ultimately decreases the bit rate; recall that the symbol *L* exponentially grows with the increase in *M*. On the other hand, Figure 8 shows that PER decreases with the increase of *M*. Because higher values of *M* imply that we are sending less 'on' chips and we can provide more power on those 'on. Chips; consequently, the probability of bit errors diminishes. In Figure 8, PER becomes almost close to zero for SNR value above 15 dB. Hence, if the signal strength at the
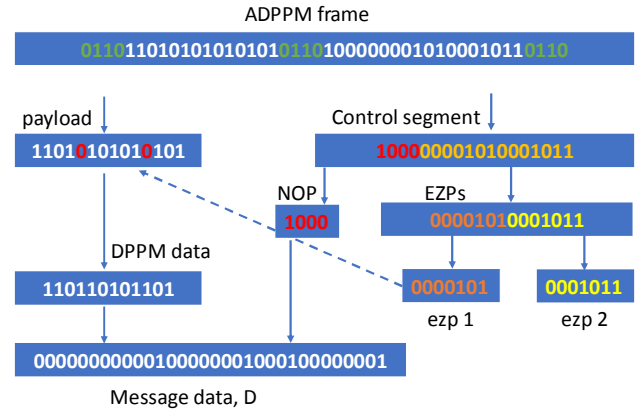

Fig. 6. An example of decoding received ADPPM frame

Input: Received ADPPM frame
Output: Message data, *D*
----
1. *Check the guard positions, G in the received data*
2. **if** *(no. of G <3 or no G found in the beginning and end)*
3.   *ack = 0*
4. **else**
5.   *extract payload and control segment*
6.   **if** *(no. of bits in the control seg. < NOP)*
7.   *ack =0*
8.   **else**
9.   *extracts NOP and EZPs*
10.   **if** *(mod(EZPs, ezp) ≠ 0)*
11.   *ack = 0*
12.   **else**
13.   *ack =1*
14.   **if** *(EZPs ≠ 0)*
15.   *get the decimal value of each ezp*
16.   **if** *(max(decimal value of each ezp)< no. of payloads)*
17.   *delete extra zeros from the payload*
18.   **if** *(no. of 1's in the payload = NOP)*
19.   *demodulate payload*
20.   **else**
21.   *ack = 0*

Algorithm 2. A demodulation of an ADPPM frame

Input: Data, M
Output: Packet error rate, PER
----
1. *Initialize: M, D and threshold_PER*
2. *Call Algorithms 1 and 2*
3. *Calculate PER*
4. **if** *PER > threshold_PER*
5.   *Increase M*
6. **else**
7.   *Keep same M*
8.   *Go to next frame*
9. *calculate PER*

Algorithm 3. Changing value of *M* to optimize the PER

receiver is very good all values of *M* works perfectly. In this kind of situation settings, a lower value of *M* is desired, because it will give better bit rate. However, if the signal strength is low

such that SNR is less than 15 dB, a higher value of M gives better PER at the expense of lower bit rate and consequently lower bandwidth efficiency as shown earlier in Figure 4. To conclude, depending on the signal strength at the receiver, we can dynamically change the value of M to achieve power and the bandwidth efficiency using Algorithm 3.

Figure 9 shows PER verses underwater depth for different values of M. We have also studied the performance of OOK modulation, since it is the most popular modulation techniques for underwater optical communication; thus, we use OOK as a baseline to compare the result of our proposed ADPPM modulation scheme. The results in Figure 9 indicate that PER grows with the increase of underwater depth d_w; this is because the SNR diminishes with the increase of underwater depth due to signal absorption and scattering. PER increases rapidly for small values of M. For OOK, PER also increases with the increase of d_w. For example, at depth 20 meter, the value of PER is almost zero for $M = 4, 8, 16$ and approximately 0.5 for $M = 2$. Hence, at this point, setting $M = 4$, 8, or 16 will yield the same error performance. However, keeping the value of M as low as possible will increase the bit rate as we discussed earlier. By using Algorithm 3 we can dynamically adjust M to optimize the bit rate and PER. For example, for a PER threshold of 0.3, Algorithm 3 will adjust M if PER exceeds a threshold. The green line in Figure 9 shows the simulation result for Algorithm 3. The results plotted in the figure indicate that PER remains below 0.3 for $d_w = [5, 15]$, and consequently a setting of $M = 2$ is the best option. With the increase of d_w, Algorithm 3 dynamically would boost M to meet the PER requirement. Figure 9 also shows that our proposed approach outperforms OOK for increased underwater depth.

V. Conclusion

This paper has studied modulating data for communication over VLC links from an airborne unit to an underwater node across the air-water interface. We have presented a novel frame structure for DPPM modulation that enables dynamic adjustment of the symbol size to cope with varying channel conditions and transmission power constraints. The proposed frame provisions for specifying key DPPM attributes in order to inform the receiver about how to demodulate the transmitted signals. We have also developed an algorithm to dynamically change those DPPM attributes to optimize the bit rate and PER. Through simulation we have shown that our proposed Adaptive

DPPM scheme with adjustable symbol length, provides better power and bandwidth efficiency than the DPPM with fixed symbol sizes. Our modulation scheme also provides better PER than traditional VLC modulation techniques such as OOK.

References

[1] M. S. Islam and M. F. Younis, "Analyzing visible light communication through air–water interface," *IEEE Access*, 7, pp. 123830-123845, 2019.

[2] J. Wang, C. Lu, S. Li, and Z. Xu, "100 m/500 Mbps underwater optical wireless communication using an NRZ-OOK modulated 520 nm laser diode," *Optics Express*, Vol. 27, pp. 12171-12181, 2019.

[3] X. Liu, S. Yi, R. Liu, L. Zheng and P. Tian, "34.5 m underwater optical wireless communication with 2.70 Gbps data rate based on a green laser with NRZ-OOK modulation," *Proc. of the 14th China Int'l Forum on Solid State Lighting,* China, Beijing, pp. 60-61, 2017.

[4] D. Zhang, Y. Zhu, and Y. Zhang, "Multi-LED phase- shifted OOK modulation based visible light communication systems," *IEEE Photonics Technology Letters*, Vol. 25, December 2013.

[5] J. Liu, B. Zheng, L. Zhao and Z. Gong, "A design of underwater wireless laser communication system based on PPM modulating method," *Proc of OCEANS 2015 - MTS/IEEE Washington*, Washington, DC, pp. 1-6, 2015.

[6] S. Meihong, Y. Xinsheng and Z. Zhangguo, "The modified PPM modulation for underwater wireless optical communication," *Proc. of the Conf. on Comm. Software & Nets*, Macau, China, pp. 173-177, Feb. 2009.

[7] B. Bai, Z. Xu and Y. Fan, "Joint LED dimming and high capacity visible light communication by overlapping PPM," *Proc. 19th Annual Wireless and Optical Comm. Conf. (WOCC 2010)*, Shanghai, China, May 2010.

[8] J. E. Gancarz, H. Elgala, and T. D.C. Little, "Overlapping PPM for band-limited visible light communication and dimming," *Journal of Solid State Lighting*, vol. 2, no.1, May 2015.

[9] E. Elsayed, B. Yousif and M. Alzalabani, "Performance enhancement of the power penalty in DWDM FSO communication using DPPM and OOK modulation," Optical and Quantum Electronics, vol. 50, June 2018.

[10] D. Shiu and J. M. Kahn, "Differential pulse-position modulation for power-efficient optical communication," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1201-1210, Aug. 1999.

[11] G. A. Mahdiraji and E. Zahedi, "Comparison of selected digital modulation schemes (OOK, PPM and DPIM) for wireless optical communications," *Proc. of 4th Student Conference on Research and Development*, Selangor, Malaysia, pp. 5-10, June 2006.

[12] J. Ma, Y. Jiang, S. Yu, L. Tan and W. Du, "Packet error rate analysis of OOK, DPIM and PPM modulation schemes for ground-to-satellite optical communications," *Optics Comm.*, vol. 283, pp. 237-242, January 2010.

[13] N. M. Aldibbiat, Z. Ghassemlooy and R. McLaughlin, "Dual header pulse interval modulation for dispersive indoor optical wireless communication systems," *IEE Proceedings - Circuits, Devices and Systems*, vol. 149, no. 3, pp. 187-192, June 2002.

[14] U. Sethakaset and T. A. Gulliver, "Performance of differential pulse-position modulation (DPPM) with concatenated coding over optical wireless communications [optical wireless communications]," *IET Communications*, vol. 2, no. 1, pp. 45-52, January 2008.
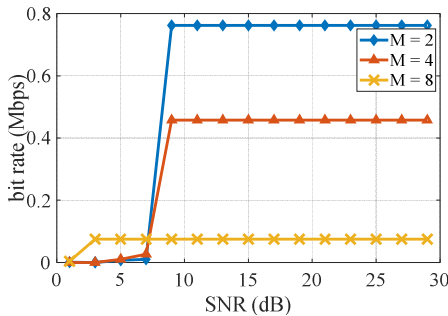
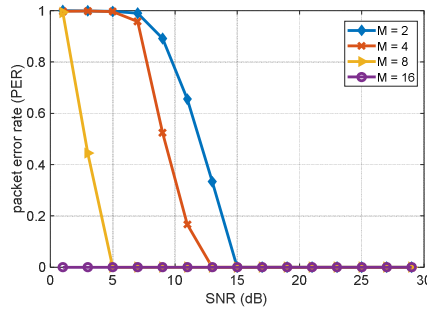**Fig. 7.** Bit rate vs SNR for different values of M

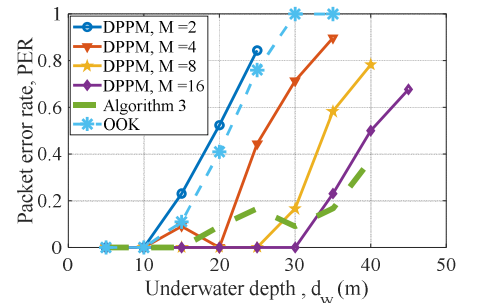**Fig. 8.** Packet error rate vs. SNR for different M in DPPM

Fig. 9. PER vs. underwater depth for various modulation techniques