Received May 16, 2019, accepted June 4, 2019, date of publication June 19, 2019, date of current version July 16, 2019.
*Digital Object Identifier 10.1109/ACCESS.2019.2923644*

# Evaluation of a Joint Detector Demodulator Decoder (JDDD) Performance With Modulation Schemes Specified in the DVB-S2 Standard

**KHEONG SANN CHAN[1], (Senior Member, IEEE), ASHISH JAMES[2], AND SUSANTO RAHARDJA[3], (Fellow, IEEE)**
[1]Nanjing Institute of Technology, Nanjing 211167, China
[2]Institute for Infocomm Research (I2R), A*STAR, Singapore 138632
[3]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
Corresponding author: Susanto Rahardja (susanto@nwpu.edu.cn)

**ABSTRACT** With the rapid increase in the growth of digital broadcast technologies, there has been a demand for high data rate, power, and bandwidth efficient transmission, which, today, is aided by the high-performing low-density parity check (LDPC) codes. In this paper, we explore a novel alternative scheme to the LDPC decoder that performs detection, demodulation, and decoding jointly, coined the joint detector demodulator decoder (JDDD). We test the JDDD through simulation using the modulation schemes defined in the DVB-S2 satellite broadcasting standard. The JDDD is a more recently developed algorithm to the sum–product algorithm (SPA) that is used in decoding the LDPC codes. The JDDD is optimal over a modulated additive white Gaussian noise/intersymbol interference (ISI) channel when resources are sufficient, with the main constraint limiting the algorithm being the availability of computing resources. In this paper, we compare the performance of the system using the JDDD against that of the LDPC decoder. The main result is that the JDDD is able to outperform the iterative detector decoder (IDD) at shorter codeword lengths, when resource requirement is smaller, while the increase in computational requirements tends to favor the IDD at longer CWLs.

**INDEX TERMS** Joint detector demodulator decoder, map detector-demodulator, iterative detector-decoder, ML decoder, JVDD, ISI channel, DVB-S2, AWGN, LDPC.

## I. INTRODUCTION

Limited availability and high costs for licensing of the radio spectrum has led to the demand for more bandwidth efficient communication systems. Advanced modulation schemes such as those used in the DVB-S2 standard [1]–[4] have been developed to accommodate higher data-rates and correct errors with advanced low-density parity check (LDPC) coding. LDPC codes [3], [5]–[14], are used in the DVB-S2 standard for their high performance and low complexity, providing around 30% performance improvement over their predecessors [15], [16].

In the current paper, we investigate the performance of an alternative algorithm to the iterative detector/decoder (IDD), tested over the code rates and modulation schemes used in the DVB-S2 standard. The code-rates used in the DVB-S2 standard are: 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 and 9/10, while the modulation schemes are selected from QPSK, 8PSK, 16APSK and 32APSK, which are chosen according to the transmission conditions. The novel scheme has been named the joint demodulator/detector/decoder (JDDD) and it performs the detection and decoding jointly instead of sequentially or iteratively as with the IDD. The motivation behind exploring a joint approach is the hope/expectation that the integrated system may outperform its iterative counterpart that exchanges information between component modules.

Much of the pre-existing literature focuses on testing and improving aspects of the LDPC code and/or decoder within the communication system. In [8]–[10], the authors look at FTN (faster than Nyquist) signaling which is a new feature optionally included in the latest extended version of the standard: DVB-S2x. The proposal with FTN is to increase the rate at which symbols are passed through the medium *without* increasing the bandwidth of transmission. This is

The associate editor coordinating the review of this manuscript and approving it for publication was Soon Xin Ng.

achieved by transmitting the next pulse before the previous pulse has completed, meaning that the pulses overlap in the time-domain and are therefore no longer orthogonal to each other. This results in a certain known amount of intersymbol interference (ISI) in the transmitted signal which can then be addressed with an appropriate trellis-based receiver. The current work attempts to achieve similar design goals as the FTN approach, as bandwidth, SNR, BER and transmission time can be traded-off one against the other. In this work, we explore the path of improved performance by using the joint-detection/decoding scheme proposed.

In [11] the authors have analyzed the performance of quasi-cyclic (QC) LDPC codes over the DVB-S2 channel measuring the performance gap from the Shannon capacity limit and explaining this gap in terms of losses caused by the base-matrix, the number of decoding iterations and the fact that the codeword length is not infinite, and in [12], the authors study performance of the LDPC codes used in DVB-S2, comparing them to those used in the ATSC 3.0 standard, in terms of their gaps to capacity.

Some previous work on joint detection and decoding has been done in [17]–[19]. In the usual IDD, detection is performed on a trellis, while decoding is performed on a Tanner graph [14] using the belief-propagation algorithm (BPA). In order to achieve the detection and decoding operations jointly, one could take the approach in [18], [20], in which they convert the detector to operate over a Tanner graph and use the BPA to perform both the detection and decoding. With the JDDD we are taking, in a sense, the opposite approach, where we convert the algorithm to operate entirely over a trellis. The advantage of the Tanner-graph based approach is that its computational complexity is lower, while the advantage of the trellis based approach adopted in [19] is its conditional optimality, producing the maximum likelihood (ML) bit sequence when the noise is white. In the current work, a more comprehensive set of simulations have been performed for the JDDD, varying both the code-rate and modulation schemes.

ML decoding has been previously analyzed for two classes of codes, namely block codes and convolutional codes. For convolutional codes, the Viterbi algorithm (VA) [21], [22] returns the minimum metric path through the code trellis. However, optimal ML decoding of linear block codes has proven to be an NP-hard problem [23], [24]. There has been much effort in the direction of developing detection/decoding algorithms for such categories of codes with acceptable complexity [25]–[28].

The joint Viterbi detector/decoder (JVDD) [28] was formulated based on the ML criterion for linear block codes. It targets to return the minimum metric legal codeword (MMLC), or the legal codeword that has the smallest path metric in the trellis. The JVDD however was formulated to operate over the magnetic recording channel which operates at baseband frequencies. In the current work, we extend this algorithm to operate over a modulated system and test it with conventional modulation schemes.

In Section II we describe the system architecture and the JDDD algorithm, followed by the simulation parameters and results in Section III with the conclusion and future works in Section IV.

## II. THE JOINT DETECTOR DEMODULATOR DECODER (JDDD) ALGORITHM

Fig. 1 shows the system block diagram for the JDDD over a modulated channel where the modulation schemes are the ones used the DVB-S2 standard. The data is protected by an LDPC code in the standard channel, after which the bits are converted into symbols by the modulation scheme. The symbols are subsequently passed through an ISI channel with additive white Gaussian noise (AWGN). The output of the channel is then processed by the 2 competing schemes. In the top branch, is the conventional iterative detector/demodulator/decoder, while the bottom branch has the novel joint detector/demodulator/decoder.

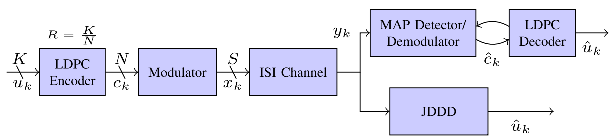



**FIGURE 1.** Communication system block diagram depicting an iterative detector/demodulator-decoder on the top path and the competing joint detector demodulator decoder (JDDD) below.

### A. SYSTEM MODEL

The input of the LDPC encoder is a length $K$ information vector consisting of $\mathbf{u} = [u_1\ u_2 \cdots u_K]$ which is encoded using the linear block LDPC code. The LDPC encoder operates at a code rate $R$ which generates a codeword of length $N$: $\mathbf{c} = [c_1\ c_2 \cdots c_N]$, where the code-rate $R = K/N$. The corresponding $(N, K)$ code is represented by a parity check matrix $\mathbf{H} = [h_{ij}]_{M\times N}$, where $M = N - K$ is the number of parity check bits and $N$ is the number of coded bits. The coded bits are then modulated onto symbols $\mathbf{x} = [x_1\ x_2 \cdots x_S]$ where the number of symbols $S = N/b$, with $b$ bits being represented by one symbol in the constellation, which is subsequently transmitted over the ISI channel. An ISI channel with AWGN is used in this work with the received signal $y_k$ given by:

$$y_k = \sum_{i=0}^{L-1} f_i x_{k-i} + w_k \tag{1}$$

where $f_i$ is the channel response of length $L$, $x_k$ is the to-be-transmitted encoded signal and $w_k$ is a zero mean AWGN process of variance $\sigma^2$. This work assumes the more general case where the terms in (1) are complex-valued, with the real and imaginary portions representing the quadrature and in-phase components of the modulated signal respectively. It is further assumed that the receiver has knowledge of the channel response coefficients $f_i$. The received signal is then passed to the JDDD algorithm that determines the most likely sequence of user bits given the observation.

### *B. JDDD*

The most likely estimate of the bits is achieved by deciding on the symbol sequence $x_k$, that maximizes the probability $\Pr(x_k|y_k)$, known as the *maximum a posteriori* (MAP) criterion [29]. When the input symbols have equal probabilities, the MAP criterion is equivalent to the ML criterion that maximizes the probability $\Pr(y_k|x_k)$. When the noise is white, this corresponds to finding the sequence with the minimum Euclidean distance. However, the minimum distance bit-sequence, which corresponds to optimum detection/demodulation only, may not correspond to a valid codeword. Thus, the optimal detector/demodulator/decoder needs to find the minimum distance codeword rather than the minimum distance sequence. The JDDD performs detection, demodulation and decoding jointly by finding the *codeword* with the minimum Euclidean distance to the received waveform. We call this codeword the minimum metric legal codeword (MMLC). As with the regular VA, the metric is defined as the Euclidean distance between the received (noisy) waveform and the candidate sequence associated with each the survivor path. The JDDD imposes constraints derived from the parity check matrix, on the survivors to ensure that the survivor paths correspond to legal codewords.

The JDDD operates on a trellis and has three main operations in addition to the regular trellis-based algorithms: metric thresholding, parity checking and capping. Their purpose is to ensure the bit sequence of the survivor is a legitimate codeword, with a reasonable number of survivors in the trellis. The sequences in the JDDD consist of symbols from the constellation defined in the modulator. Therefore, the number of states ($N_s$) in the JDDD trellis is determined by $b$, the number of bits constituting a symbol in the modulation constellation and $L$, the length of the channel impulse response and is given by

$$N_s = \begin{cases} 2^{b(L-1)} & \text{if } L > 1 \\ 2^b & \text{otherwise} \end{cases} \tag{2}$$

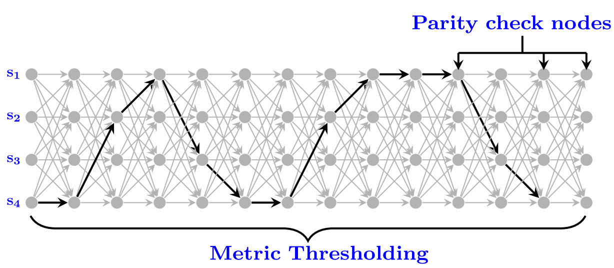



**FIGURE 2. An example trellis for the JDDD over a QPSK modulated channel with 1-tap of ISI, corresponding to 4-states in the trellis. The metric thresholding occurs at every time-instance while the parity checking occurs only at specific time-instances corresponding to the location of the last one in a certain row of the parity check matrix H.**

Fig. 2 shows an example JDDD trellis corresponding to QPSK modulation. Branch metrics between two legal states at consecutive times in the trellis are computed as:

$$\gamma_{k,(i,j)} = (y_k - s(i,j))^2 \tag{3}$$

where $k$ is the time index in the trellis, and the transition is from state $i$ to state $j$, $y_k$ is the received noisy waveform from the channel, and $s(i,j)$ is the noise-free value computed for the particular transition from state $i$ to state $j$.

Survivor paths are made up of a sequence of consecutive trellis branches, and the survivor metric is simply the summation from each of its constituent branch metrics

$$\alpha_k = \sum_{n=0}^{k} \gamma_n \tag{4}$$

where $\gamma_n$ are the metrics of the branches that make up the survivor path keeping in mind that the JDDD retains a time-varying list of such surviving paths. The path with the smallest metric through the trellis corresponds to the symbol sequence that produces the waveform closest to the received waveform, in the Euclidean distance sense. This is the sequence returned by the conventional VA. The JDDD on the other hand searches for the path with smallest metric that simultaneously satisfies the code constraint imposed at the encoder, that we refer to as the MMLC. This is achieved through keeping a list of the smaller metric survivors using a threshold, and removing survivors that are not legal codewords through parity checking. We now describe these operations in more detail.

#### 1) METRIC THRESHOLDING

The JDDD computes all survivor paths emanating from a particular node and uses a *threshold* to retain a subset of the smallest metric survivors. Let us denote the threshold as $\tau$, then only the survivors with path metrics $\alpha_{k,(j)} < \tau + \alpha^{min}_{k,(j)}$ are retained. The thresholding is done on a state-by-state basis wherein $\alpha^{min}_{k,(j)}$ is the minimum path metric at state $j$ and at time $k$ over all competing survivors at state $j$ and time $k$. Thus only survivors with metrics within the threshold of the minimum metric at each state and time are retained for each state and at each time. In this way, survivors propagate down the JDDD trellis by splitting at each node and then curtailing the numbers via the threshold. There always exists a trade-off between the performance and complexity for the metric thresholding: a smaller threshold results in fewer survivors and lower complexity but a larger probability of losing the MMLC resulting in poorer performance. Conversely, a larger threshold has larger complexity and better performance. The parity checking operation, described next, on the other hand, reduces complexity without a risk to the MMLC or a corresponding trade-off in performance, but it only occurs at certain nodes in the trellis. More frequent parity checking requires more parity check nodes (PCNs) and thus an increase in the coding overhead. Thus, the parity checking operation trades algorithm complexity off against code-rate in the JDDD.

#### 2) PARITY CHECKING

The survivors in the JDDD trellis can be split into 2 categories at each PCN, depending on whether or not they pass the parity check associated with the PCN. This distinction can be determined by calculating the checks defined in the parity check matrix using the bit sequences making up each survivor. Each row of the parity check matrix defines a check to be performed at specific time instances in the trellis. During the parity check operation, the symbol sequence for each survivor is demapped into a bit sequence which is subsequently checked via the parity constraint of the associated PCN. The parity check constraint evaluates the syndrome: $syn = \hat{\mathbf{c}}\mathbf{h}_i^T$ where $\hat{\mathbf{c}}$ is the detected/demodulated bit sequence of the candidate survivor and $\mathbf{h}_i$ is the $i$th row of the $M \times N$ parity check matrix $\mathbf{H}$ and multiplication is modulo 2. The survivor is retained if $syn = 0$ and discarded if $syn = 1$. This check can only occur when the bit corresponding to the last 1 in $\mathbf{h}_i$ has been reached. Hence the parity-checking operation in the JDDD occurs only at those times corresponding to the last 1 of some row in the parity check matrix $\mathbf{H}$.

Each syndrome check can determine whether or not a survivor is definitely invalid (and should thereby be discarded), with possibly valid sequences having $syn = 0$ to survive for future processing. Unlike the metric thresholding operation, the parity checking operation bears no risk to the MMLC as it is a legitimate codeword and will survive all parity checks. As the distribution of the last one in each row of $\mathbf{H}$ affects the number of survivors in the trellis, the design of the parity check matrix $\mathbf{H}$ should take into consideration the performance of the algorithm, if the code is to be used with the JVDD or JDDD. This is the main consideration in the design of the code in what has been termed "JVDD codes" [28].

### 3) CAPPING

At each time instant in the JDDD, each survivor is split, multiplying the number of survivors every $k$. The metric-thresholding operation brings down the number of survivors every $k$, typically at an average rate less than the rate at which the survivors are growing. The parity checking operation also helps to further limit the number of survivors in the trellis, but only occurs intermittently. Despite these 2 operations controlling the growth in the number of survivors, without the capping operation, there is no hard-limit on this number, which could potentially exceed the available resources. The capping operation places such a hard-limit on the number of survivors to ensure that the resource consumption does not exceed the pre-set limit. This comes at the cost of an increased likelihood of the MMLC being lost and therefore reduced optimality of the algorithm.

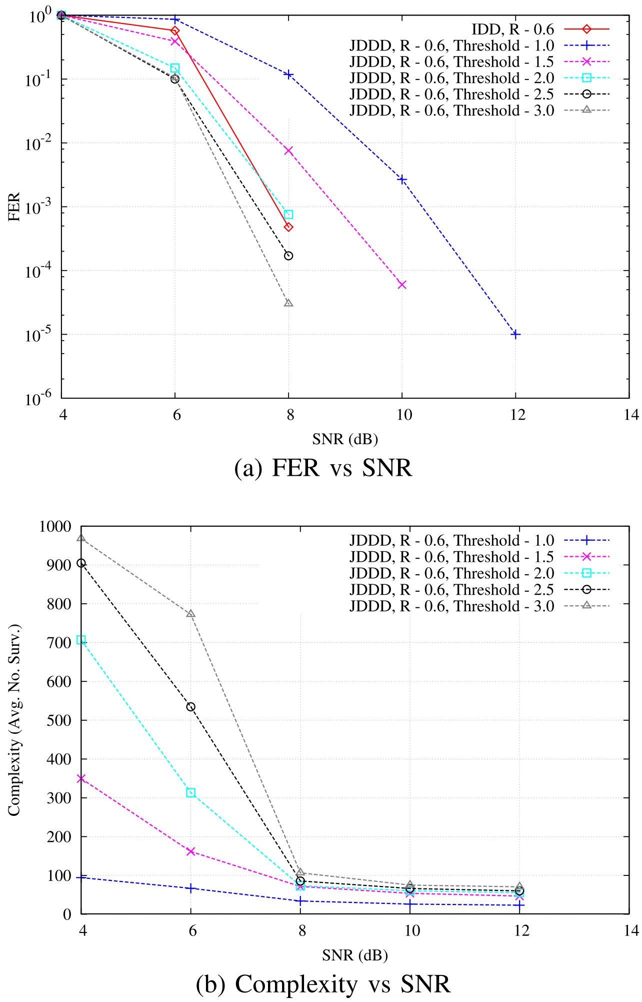
## III. SIMULATION RESULTS

In this Section, we present the results and analysis of the performance of the JDDD through Monte-Carlo simulations at varying codeword lengths, code rates and modulation schemes. The JDDD codes used in these simulations are taken from [30] and are referred to as variable gradient uniform-distribution linear diagonal (VGUDLD) codes. These are codes specifically designed for the JVDD/JDDD in which the last-one in each row of the parity check matrix is more evenly distributed so that the parity-checking operation has a more uniform impact on the number of survivors in the trellis.

The VGUDLD codes have two independent parameters $dx$ and $dy$, which correspond to horizontal and vertical shifts of the main diagonal in the parity check matrix. The ones in $\mathbf{H}$ are uniformly distributed across the row, with the row weight

**TABLE 1. Simulation parameters.**

| | |
|---|---|
| **CWL** | 1024/2048 |
| **Code-rate (R)** | [0.5, 0.67, 0.8, 0.83, 0.89, 0.9] |
| **Signal-to-noise ratio (SNR) in dB** | [0-22] |
| **[$dx$ (% of $N$), $dy$ (% of $M$)]** | [0, 5] |
| **LDPC Code** | Random code (col. wt. 4) |
| **JDDD maxNoSurv** | 10000 |
| **ISI Channel Taps** | $\frac{1}{\sqrt{6}}[1, 2, 1]$ |
| **Modulation** | QPSK, 8PSK, 16APSK, 32APSK |



(a) FER vs SNR

(b) Complexity vs SNR

**FIGURE 3. Performance of JDDD for QPSK modulation with varying thresholds at different SNRs for code-rate $R = 0.6$. Solid curve is for the conventional IDD while the dotted curves are for the JDDD at various thresholds. In this figure we compare both the performance (FER) and complexity (average number of survivors) at different thresholds.**

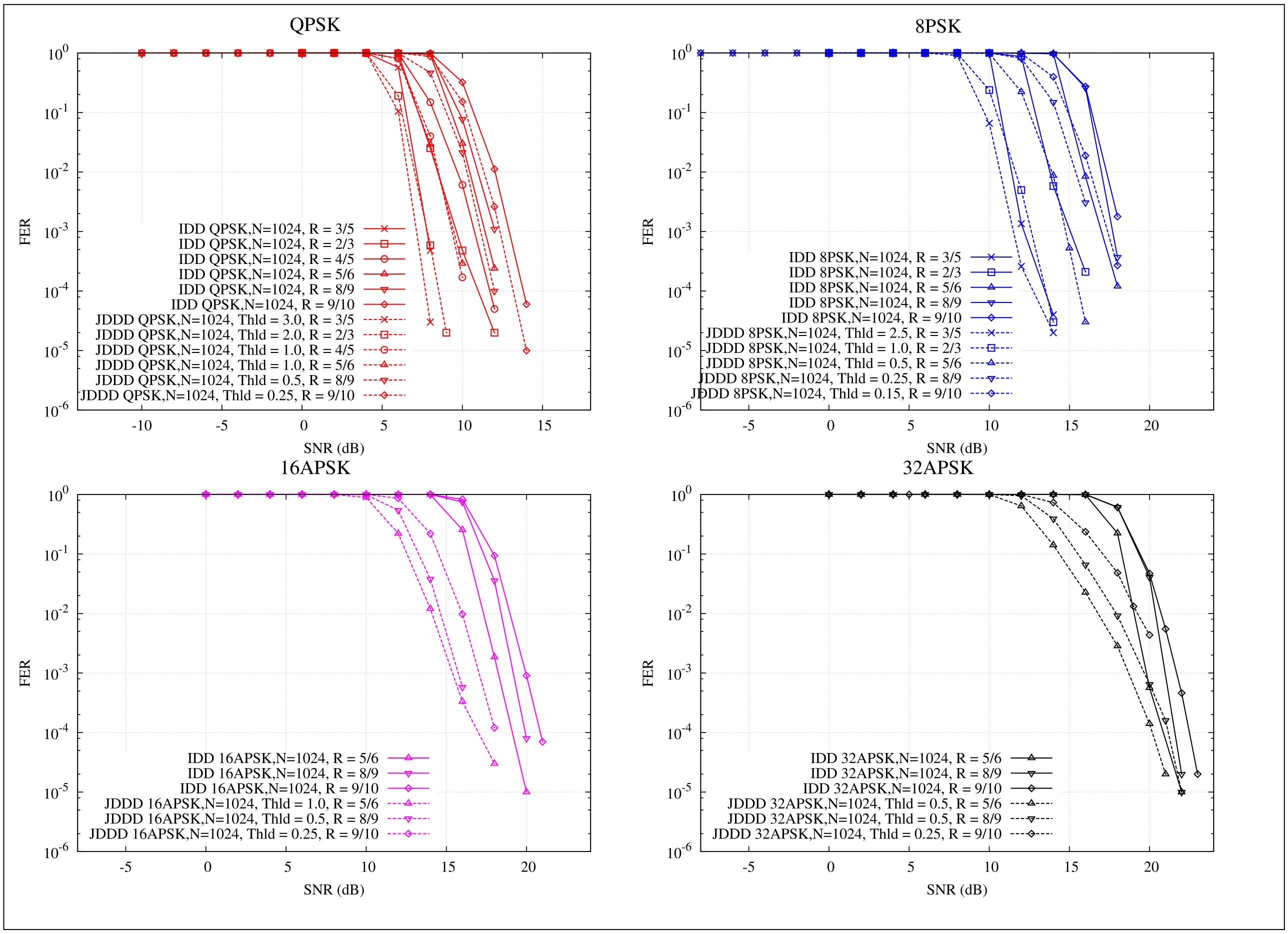



**FIGURE 4. Performance of the JDDD compared with the IDD for the various modulation schemes, and code rates used in the DVB-S2 standard, CWL = 1024.**

being set equal to 50% of the row width. In [30] it was found that increasing the parameter $dx$ leads to increased number of survivors in the trellis as the parity checking operations get delayed to start later in the trellis, thus the value for $dx$ was set at $dx = 0$. The parameter $dy$ on the other hand has to be more carefully chosen, as too small a value leads to under-protected bits at the end of the codeword, while too large a value reduces the number of parity checks occurring in the body of the trellis resulting in a more rapid growth in the number of survivors and therefore excessive complexity.

In this work, we have chosen the value of $dy$ at 5% of $M$ in the simulations which has been found to be a good trade-off between the protection for the end bits in the code and having sufficient checks in the main body of the trellis. The performance is compared with the IDD which consists of the Bahl, Cocke, Jelinek, and Raviv (BCJR) [31] algorithm coupled with the sum-product algorithm (SPA) for the LDPC decoder [14] benchmark. The BCJR outputs the log likelihood ratios (LLRs) of the coded bits which are then passed to the SPA. The codes for the LDPC decoder used in this paper are constructed randomly with a column weight of 4 and with the maximum number of local and global iterations set to 250 and 5 respectively. Table 1 shows the simulation parameters.

### A. PERFORMANCE OF JDDD VARYING THE THRESHOLD

Initially, we evaluate the performance of the JDDD while varying the threshold parameter in comparison to the IDD over a QPSK channel at $R = 0.6$. We also make an estimate of the computational complexity of the JDDD by keeping track of the average number of survivors in the trellis. The larger the number of survivors, the more computational resources and memory the algorithm is consuming. Fig. 3a shows the performance (FER) while Fig. 3b shows the complexity (in number of survivors) at different SNRs and thresholds. The average number of survivors in the trellis is a good measure of the complexity for the JDDD, but unfortunately it cannot be used to directly measure the complexity of the standard IDD that is comprised of both a trellis-based portion, as well as a factor-graph based portion. Nevertheless, the number of survivors gives us an indication of the potential feasibility of implementing this algorithm in hardware, as the channel-chip companies currently implement trellis-based detectors today, in which there is one survivor per state. The JDDD on the

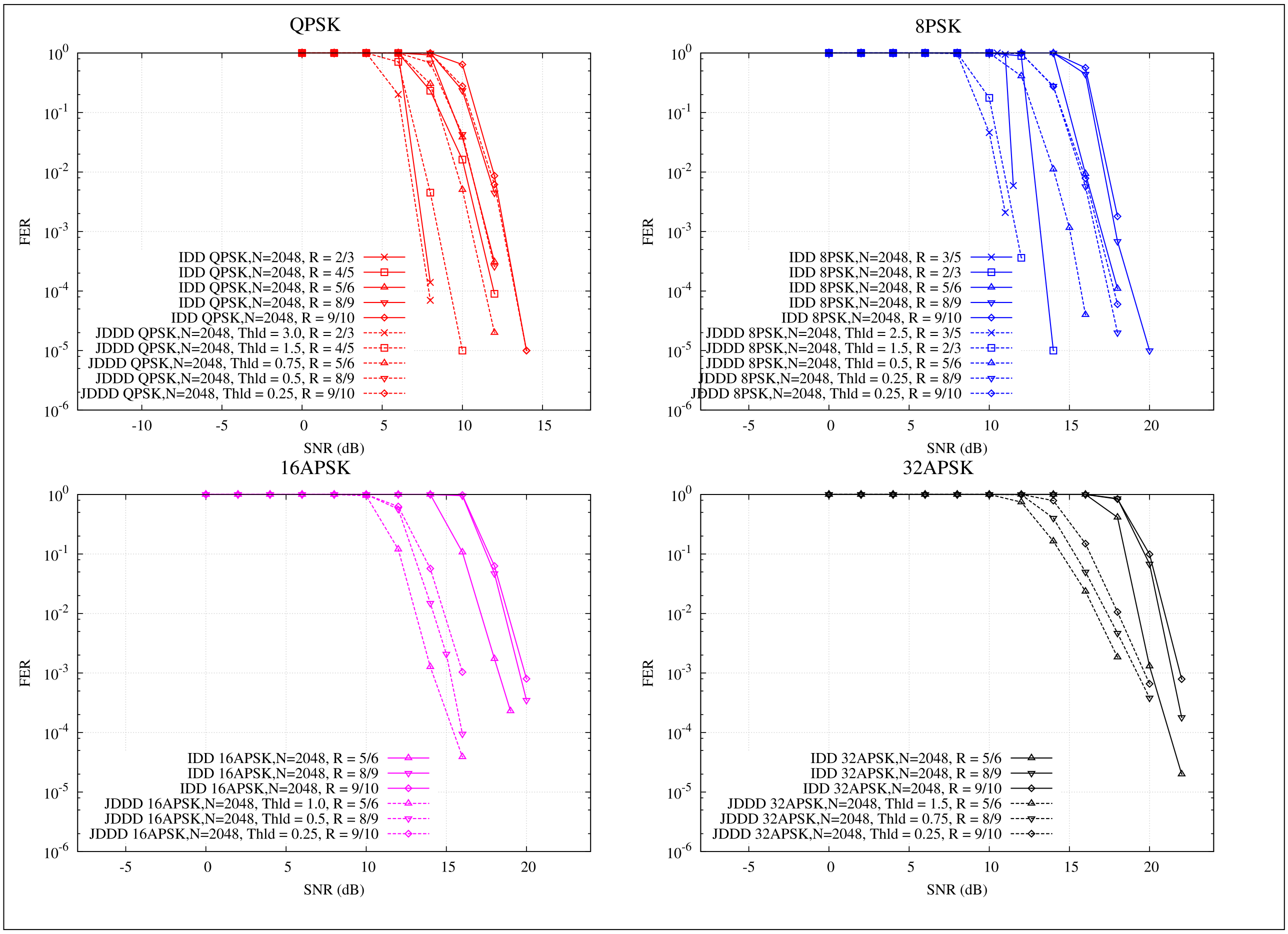



**FIGURE 5. Performance of the JDDD compared with the IDD for the various modulation schemes, and code rates used in the DVB-S2 standard, CWL = 2048.**

other hand, does not have an equal number of survivors and states, but the number of survivors is an indicator of the feasibility that the algorithm can be implemented in hardware, to channel-chip manufacturers.

In Figure 3, we see as expected, that at the lower thresholds the performance suffers, while the complexity is low. To get better performance, larger thresholds are required, leading to more survivors in the trellis. The number of survivors is also seen to be SNR-dependent, increasing at the lower SNR's. This is because at low SNR's, more competing survivors have the chance to make it within the threshold of the minimum metric, increasing the number of survivors. At the same time, lower SNR's increase the probability that the metric of the MMLC is pushed outside of the threshold from the minimum metric, thereby necessitating larger thresholds to retain the MMLC. However, we still observe the JDDD outperforming the IDD at lower SNR's and at higher thresholds. At the highest threshold of 3.0 tested, the JDDD is outperforming the IDD by close to 1dB under these conditions. At the same time, we observe the computational complexity of the JDDD coming down with increasing SNR, as less noise leads to fewer survivors being pushed within the threshold of a competing survivor path, and with decreasing threshold, which is the usual performance/complexity trade-off.

Another observation from Fig. 3a is that not only do the curves shift to the left with increasing threshold, but the gradient also increases meaning that the JDDD is able to perform better in the low FER region. This double win is a needed property for the JDDD to compete with and outperform the IDD and is the reason the JDDD will need to be operated at larger thresholds in practical applications.

### B. PERFORMANCE OF JDDD WITH DIFFERENT MODULATION SCHEMES

Next, we investigate the performance of the JDDD algorithm with different modulation schemes and code rates. The results are shown in Figure 4, with the IDD and JDDD being depicted by solid and dotted curves respectively. In the cases tested, the JDDD curves are outperforming the respective IDD curves by varying degrees. At lower code rates, the performance improves due to the additional parity bits included into the transmitted bit stream. However this comes at a cost of higher bandwidth requirements. Secondly, we can again observe that at higher modulation schemes, more SNR is

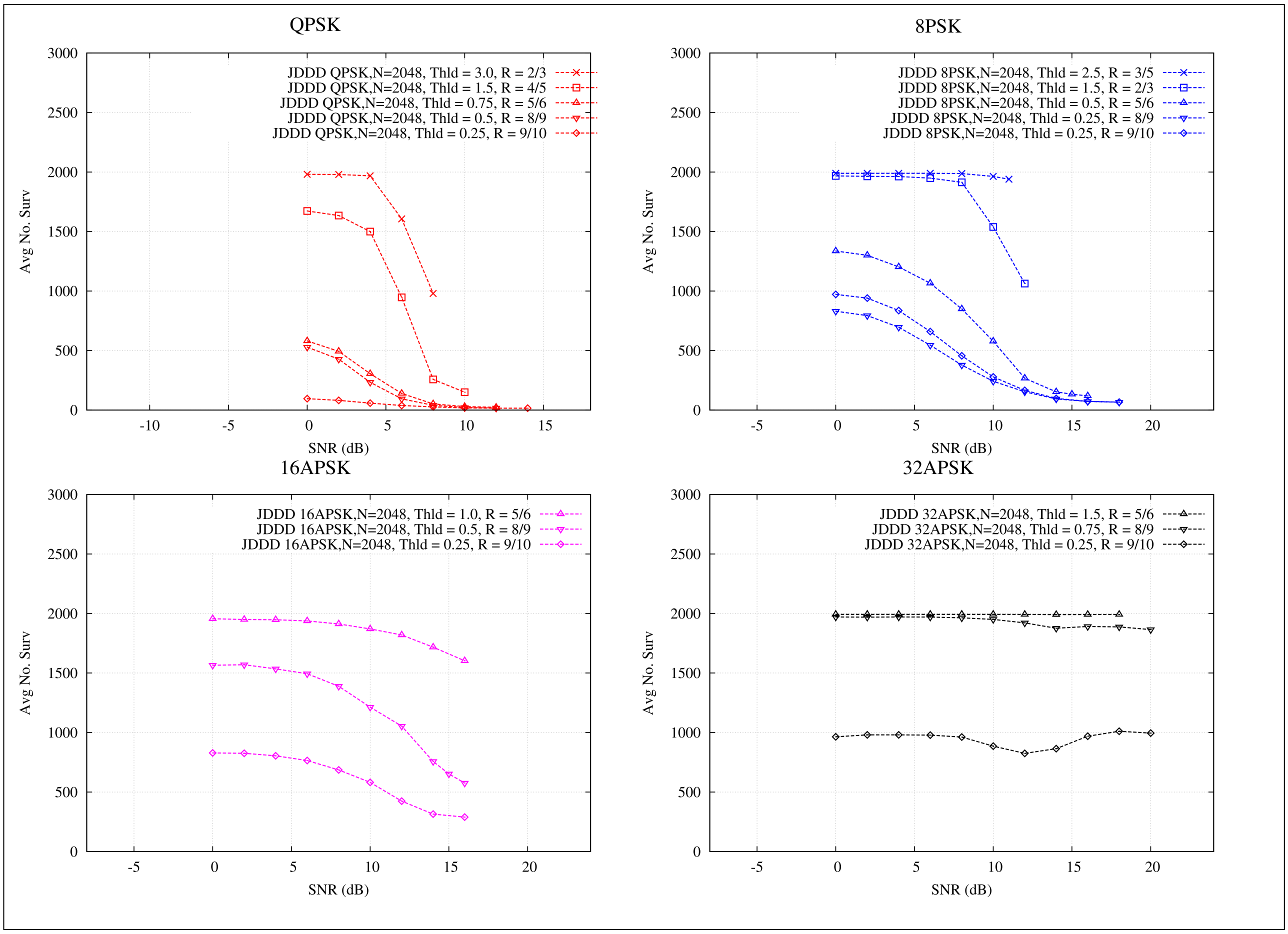



**FIGURE 6.** **Computational complexity of the JDDD measured by the average number of survivors in the trellis, CWL = 2048.**

needed to achieve the same level of performance, as the distance between constellation points is closer, but this is in exchange for higher bandwidth efficiency.

Another interesting trend to be noted is that for the higher modulation schemes, the JDDD appears to outperform the IDD as a group, rather than individually at each code-rate. At 32APSK, the phenomenon is accentuated in that the JDDD curves start to drop at a much lower SNRs, around 10-12 dB while the IDD curves only start to drop after 16dB. However the JDDD curves drop more slowly and get closer to the IDD curves towards the higher SNR/lower FER region. This phenomenon occurs because the codeword length is relatively short in the simulations at CWL=1024. With short CWL's, there is a higher probability that even at the lower SNR's the MMLC can make it through the trellis with the help from the code, especially at the lower code rates (due to the parity checking function) and with the larger thresholds. This is one area where the JDDD has been found to outperform by larger margins. However the abovementioned phenomena exists because of the relatively short choice of CWL=1024. At longer CWLs for example, we would not expect the JDDD to succeed at the low SNR's because the longer the trellis, the higher the probability the MMLC is lost during thresholding.

In order to check whether the above explanation holds, the simulations were rerun at CWL=2048 and the results are shown in Figure 5. The separation of the JDDD curves from the IDD curves are again conspicuously visible for the higher order modulation schemes. The improvement from CWL=1024 to 2048 for the IDD however, appears to be fairly minimal, and there is a slight tendency observable in the trend for the JDDD curves towards the steeper gradients for CWL=2048 as was predicted previously for the 32APSK curves.

The gains observed of the JDDD over the IDD in Figs. 4 and 5 are due to the choice of the threshold parameter used, that controls the trade-off between performance and complexity. In Figure 6 we show the average number of survivors of the JDDD in each situation as a measure of its complexity for CWL=2048. Here we reiterate the value of using the average number of survivors in the JDDD. Although this metric is not as suitable for the IDD, since the IDD has a trellis based and a factor-graph based portion, channel companies that implement trellis-based algorithms in hardware

are able to gauge from these numbers whether the algorithm might be viable for implementation in silicon. In previous discussions with channel-chip manufacturers, it has been said that typical soft-output trellis detectors could have 32 states, and thereby 32 survivors in the trellis. With the evolution of chip technology since that time, 64 survivors is most probably viable at reasonable cost. In addition, the IDD has some extra computations needed to evaluate the factor graph portion, although admittedly the bulk of the computations within the IDD are going to be consumed by the trellis portion. We can therefore use 64 as a ballpark benchmark of the number of survivors that could be acceptable to the industrial channel chip manufacturers, and probably 100 survivors could be managed, if the additional performance could justify the additional cost. The results in Fig. 6 are showing that there exist regimes of operation at higher SNR and lower modulation schemes where the average number of survivors are within a reasonable range, and yet still able to outperform the IDD as demonstrated in Fig. 5.

## IV. CONCLUSIONS

In this paper, we have proposed a novel scheme that performs the detection, demodulation, and decoding of signals coming from a modulated channel, jointly. The JDDD algorithm is found to outperform the conventional MAP detector/demodulator with LDPC decoder at the tested code rates, modulation schemes and codeword lengths when the threshold is adequately selected. However, the JDDD tends to require greater memory and computational resources for tracking the survivors in the trellis.

The JDDD performance improves with increasing threshold parameter, corresponding to allowing more survivors in the trellis and thereby demanding more resources. Having more survivors benefits the JDDD algorithm by both shifting the FER curves to the left, as well as increasing the steepness of the waterfall region. We also have shown the average number of survivors in the JDDD trellis for each of the schemes investigated at CWL=2048 and found that there did exist regions where the JDDD was performing with a number of survivors that are in the same ball-park as conventional detectors employed today.

When comparing the JDDD with the IDD at different code rates and for different modulation schemes, we find the usual waterfall curves shifting to the right with increasing modulation order (QPSK up through 32APSK) and to the left with increasing coding overhead. The steepness of the JDDD curves is found to get less with increasing modulation schemes, with 32APSK producing the slowest dropping curves. However, to compensate, the JDDD curves start their waterfall drop at lower SNR's than the corresponding IDD curves. We attribute this to the poorer performance of the IDD at lower SNR's and higher modulation schemes.

The main challenge for the JDDD is its performance at larger block sizes, as the longer CWL increases the chances that the MMLC will be discarded (degrading performance) thereby necessitating larger thresholds to compensate (increasing complexity). In a future work, we will investigate the expected performance of this algorithm as CWL grows, how close to optimal the JDDD can perform, and the computations that this will cost, to evaluate its potential for applications at longer CWL's.

## REFERENCES

[1] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proc. IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006.

[2] *Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, document EE307-1, DVB-S2, ETSI, Oct. 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302300_302399/302307/01.02.01_40/en_302307v010201o.pdf

[3] S. Papaharalabos, M. Papaleo, P. T. Mathiopoulos, M. Neri, A. Vanelli-Coralli, and G. E. Corazza, "DVB-S2 LDPC decoding using robust check node update approximations," *IEEE Trans. Broadcast.*, vol. 54, no. 1, pp. 120–126, Mar. 2008.

[4] D. Theodoropoulos, N. Kranitis, and A. Paschalis, "An efficient LDPC encoder architecture for space applications," in *Proc. IEEE 22nd Int. Symp. On-Line Test. Robust Syst. Des. (IOLTS)*, Jul. 2016, pp. 149–154.

[5] M. Y. Zinchenko, A. M. Levadniy, and Y. A. Grebenko, "Development of the LDPC coder-decoder of tHE DVB-S2 standard on FPGA," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun.*, Jul. 2018, pp. 1–3.

[6] D. J. Patel and P. Engineer, "Design and implementation of quasi cyclic low density parity check (QC-LDPC) code on FPGA," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 181–185.

[7] A. Gupta, M. E. Scholar, A. Jain, and P. D. Vyavahare, "Performance analysis of concatenated LDPC codes for video broadcast satellite system," in *Proc. IEEE Radio Antenna Days Indian Ocean (RADIO)*, Sep. 2017, pp. 1–2.

[8] G. Y. Mihaylov, T. B. Iliev, E. P. Ivanova, I. S. Stoyanov, and L. Iliev, "Performance analysis of low density parity check codes implemented in second generations of digital video broadcasting standards," in *Proc. 39th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May/Jul. 2016, pp. 499–502.

[9] J.-A. Lucciardi, N. Thomas, M.-L. Boucheret, C. Poulliat, and G. Mesnager, "Trade-off between spectral efficiency increase and PAPR reduction when using FTN signaling: Impact of non linearities," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[10] H. Kwon, M.-S. Baek, J. Yun, H. Lim, and N. Hur, "Design and performance evaluation of DVB-S2 system with FTN signaling," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2016, pp. 1210–1212.

[11] S. Chen, K. Peng, J. Song, and Y. Zhang, "Performance analysis of practical QC-LDPC codes: From DVB-S2 to ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 172–178, Mar. 2018.

[12] K.-J. Kim, S. Myung, S.-I. Park, J.-Y. Lee, M. Kan, Y. Shinohara, J.-W. Shin, and J. Kim, "Low-density parity-check codes for ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 189–196, Mar. 2016.

[13] S. T. Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[14] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, 1997.

[15] L. F. Abusedra, A. M. Daeri, and A. R. Zerek, "Implementation and performance study of the LDPC coding in the DVB-S2 link system using MATLAB," in *Proc. 17th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. (STA)*, Dec. 2017, pp. 669–674.

[16] A. Jain and R. Singhai, "Comparative analysis of FEC subsystem in fixed satellite broadcasting," in *Proc. Int. Conf. Recent Innov. Signal Process. Embedded Syst. (RISE)*, Oct. 2017, pp. 30–32.

[17] C. K. Matcha and S. G. Srinivasa, "Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR," *IEEE Trans. Magn.*, vol. 51, no. 10, Oct. 2015, Art. no. 3101215.

[18] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1410–1422, Jun. 2002.
[19] A. James and K. S. Chan, "Joint detector demodulator decoder (JDDD) over ISI channels," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–5.
[20] C. K. Matcha, S. Roy, M. Bahrami, B. Vasic, and S. G. Srinivasa, "2-D LDPC codes and joint detection and decoding for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 54, no. 2, Feb. 2018, Art. no. 3100111.
[21] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
[22] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommun. Conf. Exhib. Commun. Technol.*, Nov. 1989, pp. 1680–1686.
[23] S. G. Wilson, *Digital Modulation and Coding*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[24] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 3, pp. 384–386, May 1978.
[25] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966.
[26] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.
[27] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995.
[28] K. S. Chan, S. S. B. Shafiee, E. M. Rachid, and Y. L. Guan, "Optimal joint viterbi detector decoder (JVDD) over AWGN/ISI channel," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 282–286.
[29] H. Sasaoka, Ed., *Mobile Communications*. Amsterdam, The Netherlands: IOS Press, 2001.
[30] A. James and K. S. Chan, "Multi-pass joint Viterbi detector decoder (MP-JVDD) over AWGN/ISI channels," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–5.
[31] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.


**KHEONG SANN CHAN** received the bachelor's degree from Northwestern University, in 1996, and the Ph.D. degree from the National University of Singapore, in 2000. He is currently a Professor with the Nanjing Institute of Technology, China. In 2017, he was transferred to the Nanjing Institute of Technology. His research interests include coding and signal processing for recording channels, channel modeling, and detection/decoding for wireless communications. His research work has contributed to the information storage industrial consortium (INSIC) and advanced storage technology consortium (ASTC) in USA. He was a recipient of the INSIC Technical Achievement Award for his work on advanced channel modeling, in 2009, and the Jiangsu Distinguished Professor Award at the Nanjing Institute of Technology. In 2014, he received the NRF Grant to explore novel joint detection/decoding schemes for satellite channels. He has been granted two recent ASTC sponsored projects, in 2015 and 2016.

**ASHISH JAMES** received the B.S. degree from the College of Engineering Trivandrum, Thiruvananthapuram, India, and the Ph.D. degree from Nanyang Technological University, Singapore, where he was a Postdoctoral Fellow with the School of Computer Science and Engineering, from 2012 to 2014. He is currently a Scientist with the Institute for Infocomm Research (I2R), one of the research institutes under Agency for Science Technology and Research (A*STAR), Singapore. His research interests include machine learning, deep learning, coding and signal processing techniques for future communication systems, cooperative communications, multiple-access techniques, and femtocells.

**SUSANTO RAHARDJA** received the B.Eng. degree from the National University of Singapore, in 1991, and the M.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 1993 and 1997, respectively. He is currently a Chair Professor with Northwestern Polytechnical University, under the Thousand Talent Plan of Peoples Republic of China. He contributed to the development of a series of audio compression technologies, such as Audio Video Standards, AVS-L and AVS-2, ISO/IEC 14496-3:2005/Amd.2:2006, and ISO/IEC 14496-3:2005/Amd.3:2006, in which some have been licensed to several companies. He has over 15 years of experience in leading research team for media related research that cover areas in signal processing (audio coding and video/image processing), media analysis (text/speech, image, and video), media security (biometrics, computer vision, and surveillance), and sensor networks. His research interests include multimedia, signal processing, wireless communications, and discrete transforms and signal processing algorithms and implementation. He was a recipient of numerous awards, including the IEE Hartree Premium Award, the Tan Kah Kee Young Inventors' Open Category Gold Award, the Singapore National Technology Award, the A*STAR Most Inspiring Mentor Award, the Finalist of the 2010 World Technology and Summit Award, the Nokia Foundation Visiting Professor Award, and the ACM Recognition of Service Award. He was a past Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing and IEEE Transactions on Multimedia and a past Senior Editor of the IEEE Journal of Selected Topics in Signal Processing. He is currently serving as an Associate Editor for the *Journal of Visual Communication and Image Representation* (Elsevier) and IEEE Transactions on Multimedia. He was the Conference Chair of 5th ACM SIGGRAPHASIA, in 2012, and the APSIPA 2nd Summit and Conference, in 2010, as well as other conferences in ACM, SPIE, and IEEE.