Diversity Schemes for a Large MIMO Acoustic Transmitter on a C2000 Micro-controller

Joseph Hall and Y. Rosa Zheng

Dept. of ECE, Lehigh University, Bethlehem, PA 18015, USA
{jwh318, yrz218}@lehigh.edu

***Abstract*—Two diversity schemes are implemented for a small form-factor, large MIMO acoustic transmitter for an underwater IoT application with a high reliability requirement and target data rate of 20 kbps using a Texas Instruments C2000 micro-controller. The first is linear precoding scheme which can yield both diversity and multiplexing gain and may use channel state information at the transmitter (CSIT). The second is the Space-Time Block Code which provides optimal diversity at the cost of reduced rate and no need of CSIT. The digital processing of the transmitter design, such as channel coding, interleaving, phase-shift-keying (PSK) bit to symbol mapping, and the diversity schemes are implemented in the software via the C2000 CPU and TMU (Trigonometry Multiplier Unit). The passband modulation is achieved by multiple EPWM (enhanced Pulse width modulation) channels of the C2000. The transmitter can encode frames at 21.3 kbps in the worst case and will allow the streaming of data. The carrier-modulated BPSK, QPSK, and 8-PSK symbols are recovered and shown to have an MSE of 3% or less with the ideal constellation. The operation of the precoder and STBC diversity schemes are verified from the generated passband waveform.**

I. Background

Underwater environmental monitoring and sensing applications require the deployment of many small devices, often without a wired connection to the shore or a base station. Recovery of data from these devices requires either manual retrieval of the device or wireless transmission of the data. In the underwater environment, RF signals are strongly attenuated, and other methods of wireless communication must be used. For wireless applications targeting medium to high range, acoustic communication performs well. However, the underwater acoustic channel is a difficult medium with severe frequency-selective fading and narrow bandwidth. Additionally, underwater acoustic transducers and driver circuits are often relatively large and unwieldy. This motivates the development of small form-factor, high-reliability acoustic transmitters for underwater sensor deployments.

In this work, a transmitter is designed for a flexible-PCB underwater sensor system where a low volume of data is generated, but reception must be reliable. Additionally, the flexible PCB imposes size and complexity constraints on the circuits that may be used in the transmitter, as well as the size of the transducer elements. A large MIMO system design with 1-12 outputs and support for diversity techniques was chosen to increase reliability of reception (i.e. the system was designed to maximize diversity gain and did not prioritize multiplexing gain), a carrier frequency of 200kHz was selected to reduce the size of the transmitting element, and the modulation steps were implemented digitally, allowing a small, simple transmitter front-end circuit. The TMS320F28379D, a DSP-derived microcontroller in the C2000 family, was chosen to implement the digital portion of the system. TI DSPs have seen use in many other underwater acoustic communication systems, and the TMS320F28379D's low power consumption and flexible, numerous PWM outputs make it particularly suitable.

The resulting transmitter is shown to achieve a low MSE with the ideal constellation points despite the simplifications in modulation, and the transmitter can encode frames fast enough to stream data at the target rate of 20kbps. Orthogonal space-time block codes and beamforming are currently supported as diversity schemes, with the ability to expand to arbitary precoding and STBC schemes with only changes to the modulation design.

II. Transmitter Diversity

The fundamental idea of transmitter diversity schemes is to transmit combinations of symbols drawn from complex-valued constellation on multiple elements and/or multiple time slots to compensate for the fading experienced by each individual path, lowering the error probability at the receiver. There are many schemes to exploit spatial-temporal diversity [1]. This work considers linear precoding with the special case of beamforming, and space-time block codes [2]. All diversity schemes are subject to a diversity-multiplexing tradeoff – as the data rate on the link is increased, the error probability will decay more slowly with increasing SNR [3]. Therefore, diversity techniques may be used to achieve higher data rate, lower error probability, or some tradeoff between the two.

One way to exploit diversity is linear precoding with a low rate [4]. For a system with N_t transmit elements across which the set of symbols $\mathbf{u}[\tau] = [u_1, u_2, \cdots u_{N_t}]$ is transmitted at each time tap τ, the precoded symbols $\mathbf{s}[t] = \mathbf{u}[\tau]\mathbf{P}$, where $\mathbf{P} \in \mathcal{C}^{N_t \times N_t}$ is the precoding matrix which is derived from channel state information at the transmitter (CSIT) $\mathbf{H}_t$ or its statistics. The time tap t is often aligned with time tap τ thus providing high multiplexing gain through N_t transmit elements. Many MIMO precoding schemes focus on multiplexing gain, or boosting the data rate that achieves a given tradeoff between error probability and SNR by transmitting

independent data streams on each Tx element. In the current application, we are primarily concerned with diversity gain, or achieving a better tradeoff between SNR and error probability for a fixed data rate. One precoding scheme focusing on this is beamforming [5] which adjusts the carrier phases on multiple Tx elements to achieve desired beam patterns. Note that to implement arbitrary precoding schemes, the transmitter must control both amplitude and phase. For systems where we do not have amplitude control, precoding is impossible, but beamforming may be employed. Beamforming schemes may be viewed as a special case of precoding, where the precoding matrix is of the form:

$$P = \begin{bmatrix} \exp(j\phi_1) & 0 & 0 & \dots & 0 \\ 0 & \exp(j\phi_2) & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \exp(j\phi_{N_t}) \end{bmatrix} \quad (1)$$

(i.e. each symbol undergoes a phase shift, possibly based on CSIT). This is achievable for a system that cannot vary its output amplitude, assuming the base constellation has symbols α_j with $|\alpha_j| = 1 \ \forall j$ (e.g. PSK).

Another means of exploiting the diversity in a MIMO system is by space-time block coding (STBC). Space-time block codes transmit blocks of input symbols, encoded over space, i.e. transmit element, and time. STBCs are defined by matrices whose entries the elements of $\mathbf{u}[\tau] = [u_1, u_2, \cdots u_{N_{sb}}]$ and whose row and column indices are timeslot and transmitter index respectively. Each N_t-length matrix row is transmitted simultaneously across the N_t transmit elements of the system. The matrix has N_{ts} rows, each corresponding to one time slot. A well-known STBC that has seen wide use in wireless standards is the Alamouti code [6]. The Alamouti code is described by the matrix:

$$\mathbf{P}_{alamouti} = \begin{bmatrix} u_1 & u_2 \\ -u_2^* & u_1^* \end{bmatrix} \quad (2)$$

This code is a special case of the orthogonal design-based STBCs described by Tarokh *et al.* [2] who develop the idea of generalized orthogonal designs and use them to construct codes that give full diversity for a given transmitter count. Any STBC matrix based on a real generalized orthogonal design may be used to form the STBC matrix for a complex signal constellation by concatenating an elementwise-conjugated copy of the original matrix to the bottom of the original matrix. For $N_t = 4$, $N_{sb} = 4$, and a complex-valued constellation, Tarokh's design criterion gives:

$$\mathbf{P}_{stbc} = [p_{r,c}]_{N_{ts} \times N_t} \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ -u_2 & u_1 & -u_4 & u_3 \\ -u_3 & u_4 & u_1 & -u_2 \\ -u_4 & -u_3 & u_2 & u_1 \\ u_1^* & u_2^* & u_3^* & u_4^* \\ -u_2^* & u_1^* & -u_4^* & u_3^* \\ -u_3^* & u_4^* & u_1^* & -u_2^* \\ -u_4^* & -u_3^* & u_2^* & u_1^* \end{bmatrix} \quad (3)$$

Other STBCs have been studied extensively. Xu *et al.* [7] surveys many STBC schemes including the orthogonal design scheme implemented in this work. The main advantages of more advanced schemes are in increasing the multiplexing gain while maintaining full diversity. This may be useful in later systems, but it is possible to meet our target data rate using rate-$\frac{1}{2}$ codes. The orthogonal STBC implementation is simple and achieves full diversity and at least rate-$\frac{1}{2}$ [2]. Other recent research on STBCs derives optimality criteria and new codes for mmWave channels where Rayleigh and Rician fading do not correctly capture the fading characteristics of the channel [8], and [9] combines spatial modulation with STBCs.

III. Hardware implementation of the large MIMO transmitter

Figure 1 shows the functional blocks of the MIMO transmitter. First, the input binary sequence $b[k]$ is convolutionally encoded, then the coded bits $c[n]$ are interleaved using a block interleaver. The resulting bits $c_i[n]$ are then grouped into binary values of length $\log_2(M)$, where M is the size of the M-PSK constellation. The grouped bits are mapped to the rectangular-form complex symbols $u[\tau]$ for the M-PSK constellation via lookup table. A diversity scheme is applied to $u[\tau]$ yielding multi-channel symbols $s_n[t]$, where subscript $n = 0, \cdots, N_t - 1$, and N_t is the number of transmit elements in the large MIMO system. Finally, the output of the diversity scheme is mapped from the rectangular form to the polar form and the phase $\angle s_n[t]$ of each symbol is used to set the compare-values CMPA and CMPB needed for each EPWM output. Each EPWM output has its own buffer to store the compare values and minimize the burden on the CPU while the data frame is being transmitted.

The structure of an encoded data frame is shown in Figure 2. A different maximum-length sequence (M-sequence) is simultaneously sent on each transmitter element using the BPSK modulation. A gap is inserted to mitigate inter-block interference, and then the payload is transmitted using the selected modulation order and diversity techniques. The resulting passband waveform is amplified and used to drive a ring-type piezoelectric PZT transducer, which produces an acoustic signal.

A. Baseband Processing

The convolutional encoder of rate 1/R encodes a block of information bits $b[k], k = 0, \cdots, K_b - 1$, to $2(K_b + K - 1)$ encoded bits, where K_b is the length of the information bit block and K is the constraint length of the encoder. Let the generator polynomials of the encoder be $G_1, G_2, \cdots, G_R$, where each generator is defined as $G_r = [g_{1,r}, \cdots, g_{k,r}, \cdots, g_{K,r}]$ with $g_{k,r}$ being the coefficients in GF(2). The encoder performs modulo-2 multiply and add between the input bits and the generator polynomials to yield the coded bits $c_1[k], c_2[k], \cdots, c_R[k]$. Since the C2000 only handles word sizes of 16 bits and larger, a uint16_t variable is used to implement the encoder's shift register and the

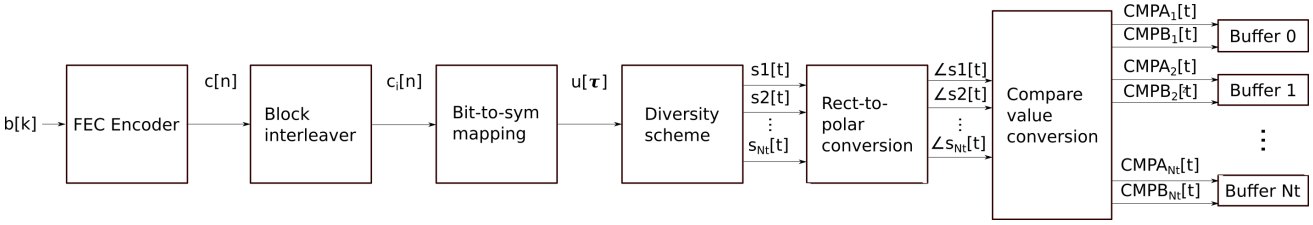


Fig. 1: Transmitter data processing of the large MIMO system

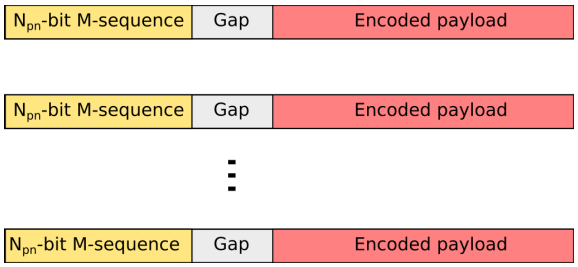


Fig. 2: Transmitted data frame structure

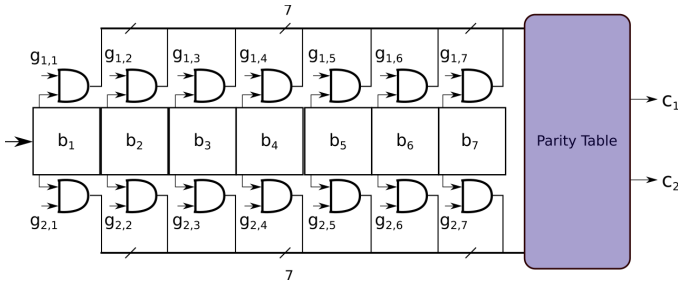


Fig. 3: Convolutional encoder implemented by parity table

input bits are also packed into uint16_t integers. The modulo-2 operation, if done directly, would require repeated integer divisions which are expensive in many microcontrollers. This paper utilizes the parity table method to reduce computational complexity. See the "Compute parity by lookup table" section of [10] for means to generate this table.

The parity table contains 1 at odd-parity indices and 0 at even-parity indices, and must have at least 2^K entries. The contents of the shift register are bitwise ANDed with the generator polynomials to yield the parity indices for each output bit, as shown in the example in Figure 3 for the parameters $K = 7$ and $R = 2$. Each entry in the parity table, for any parity index, is the mod-2 sum of the bits in the index. While only the least significant bit of each uint16_t is used, the encoder output bits $c_1[k], c_2[k], \cdots, c_R[k]$ are stored in separate uint16_t variables for each time index k and bit index r. This format makes it easy to implement the remaining functional blocks.

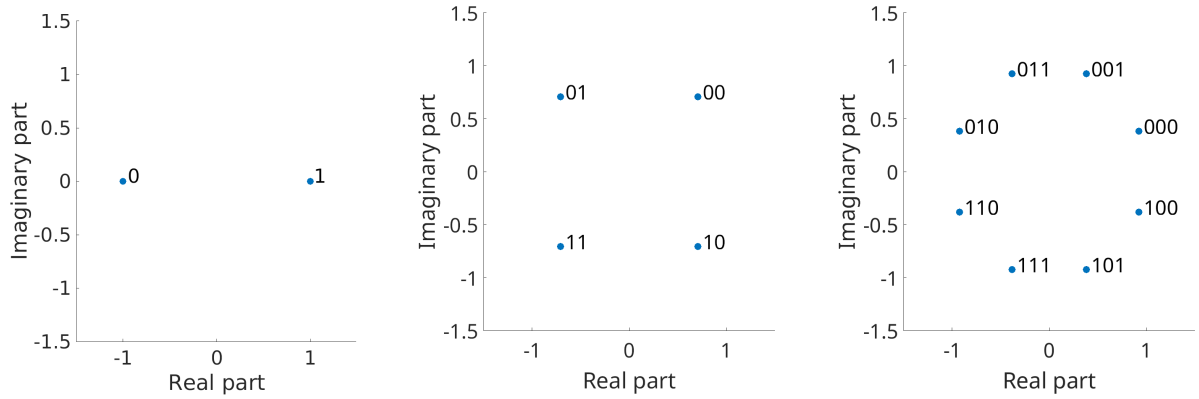
This parity table method is suitable for constraint length K up to 16 because the smallest native data type in the C2000 is 16 bits, which covers most of the practically used convolutional codes. If longer constraint length is required, it is possible to define the shift register as a 32 or 64-bit type, but the parity table will consume more ROM space. The advantage of the parity table method is its flexibility of implementing different parameters with the same structure. With the same parity table of size 2^K, a convolutional code with different generator polynomials, or different constraint length less than K, or different coding rate can be created easily. For example, a rate 1/3 convolutional code simply adds another branch of the AND gates to the rate 1/2 encoder. Alternatively, the convolutional encoder may be implemented by a trellis table directly storing the out-edges and R-bit branch words. For a constraint length K code, the number of trellis states is 2^{K-1}. This method would require at least 2 words per trellis state if the output bits are packed together, yielding $2 \cdot 2^{K-1}$ words of memory space. The parity table method also requires 2^K entries. While the memory constraint for the C2000 is not particularly stringent, the biggest difference between the trellis method and the parity table method is that the trellis table has to be constructed for each specific code with specific parameters, such as constraint length, generator polynomial, etc., meaning if rate adaptation is required, multiple tables must be stored, increasing memory complexity. In contrast, the parity table method's memory complexity does not increase when using multiple convolutional codes (other than the negligible increase from the polynomials, rates, and constraint lengths themselves, and the required buffer space for the lower-rate encoded payload). The advantage of the trellis table method is negligibly faster processing.

A block-type interleaver with arbitrarily configurable width and depth converts the encoder output $c[n]$ to a re-ordered output $c_i[n]$ by reindexing and copying $c[n]$ to a new buffer. Then the interleaver output bits $c_i[n]$ are repacked into groups of $log_2(M)$ bits where M is the modulation order.

Bit-to-symbol mapping is also implemented by a lookup table containing the gray-coded constellation reindexed by the grouped interleaver output bits. The output of this block is the real and imaginary parts of the symbols, $Re\{u[\tau]\}$ and $Im\{u[\tau]\}$, corresponding to the rectangular-format complex symbol $Re\{u[\tau]\} + jIm\{u[\tau]\}$ with $j = \sqrt{-1}$. For example, the 8-PSK gray-coded LUT is shown in Table I. The resulting PSK constellations are shown in Figure 4(c).

B. Diversity Schemes

This work implements two diversity schemes: the first is linear precoding and the second is STBC [2]. With N_t transmit elements, the diversity schemes convert the serial



(a) BPSK constellation (b) QPSK constellation (c) 8-PSK constellation

Fig. 4: Ideal constellations for implemented PSK bit-to-symbol mapping

TABLE I: Symbol lookup table for gray-coded 8-PSK

i	Bit pattern	**Packed for TMU**	Constellation $\alpha_{\mathbf{i}}$
0	000	0x00EC0061	$(0.9239 + 0.3827j)$
1	001	0x006100EC	$(0.3827 + 0.9239j)$
2	010	0xFF140061	$(-0.9239 + 0.3827j)$
3	011	0xFF9F00EC	$(-0.3827 + 0.9239j)$
4	100	0x00ECFF9F	$(0.9239 - 0.3827j)$
5	101	0x0061FF14	$(0.3827 - 0.9239j)$
6	110	0xFF14FF9F	$(-0.9239 - 0.3827j)$
7	111	0xFF9FFF14	$(-0.3827 - 0.9239j)$

symbol stream $u[\tau]$ into a size-N_t vector stream (or N_t parallel streams) and then modify the vector to the transmit signal $\mathbf{s}[t] = [s_1[t], \cdots, s_{N_t}[t]]$. Note the time indices at the input and output of the diversity scheme are different as the diversity schemes may change the transmitting rate. For linear precoding, the index t is $1/N_t$ of the symbol rate of $u[\tau]$. In contrast, the STBC schemes may have a rate less than 1. For example, a rate 1/2 STBC requires $s[t]$ to transmit twice as fast as $u[\tau]$ despite the use of multiple transmit elements. Only a perfect STBC scheme transmits at exactly the same rate as the single transmitter system (rate-1), yielding time tap t aligned with the same the time tap τ. Denote the N_{sb}-length output vector of the serial-to-parallel converter as $\mathbf{u}[\tau] = [u_1, u_2, \cdots, u_{N_{sb}}]$. Then the index τ is advanced as $\tau := \tau + N_{sb}$ for the next vector.

The complex symbols are stored as 32-bit words where the upper and lower 16 bits represent the real and imaginary parts of the symbols, respectively. This is the format expected by the C2000 VCU-II complex math unit with $CPACK = 0$. The matrix multiplication is accomplished via an assembly routine using the VCU-II's VCMAC instruction – this instruction takes 2 pipelined cycles, one active cycle and one delay slot where the VCU registers may not be touched. To fill the delay slot, the prior VCMAC real/imag outputs, each a uint32_t, can be cast to float.

In contrast, for a space-time block coding scheme with a rate $R_{stbc} = \frac{N_{sb}}{N_{ts}}$, the diversity scheme will output a matrix of size $N_t \times N_{ts}$, as given in [2]. The STBC matrix is stored as three real matrices containing real part signs $\eta_{r,c}$, imaginary part signs $\zeta_{r,c}$, and symbol permutation indices $\gamma_{r,c}$, respectively. The real and imaginary parts of each symbol are mapped as 32 bits per symbol and are stored in consecutive memory locations to form the N_{sb}-length vectors $\mathbf{u}[\tau]$. To encode one block, successive permutations of $\mathbf{u}[\tau]$ with the elements $u^p_{r,c}$ are generated, where $c \in \{1, 2, \cdots N_t\}$ and $r \in \{1, 2, \cdots N_{ts}\}$ are the column and row indices of the precoding matrix. Note the permuted row may not have all elements of $\mathbf{u}[\tau]$ for every r, and elements may be repeated. Next, according to the conjugate and sign operators in the STBC matrix, the real and imaginary parts of each $u^p_{r,c}$ are multiplied by the sign values $\eta_{r,c}$ and $\zeta_{r,c}$, which produces the symbols $s_n[t]$, $n \in \{1, 2 \cdots N_t\}$.

If we take the $N_t = 4$, complex-constellation case from (3) as an example, the matrix of symbol permutation indices would be

$$\mathbf{\Gamma} = [\gamma_{r,c}]_{4\times 4} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \tag{4}$$

and the matrices of real-part signs and imaginary-part signs would, respectively, be

$$\mathbf{Z} = [\zeta_{r,c}]_{8\times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{W} = [\eta_{r,c}]_{8\times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \quad (5)$$

The STBC processing flow's block diagram is shown in Figure 5. At each value of τ, the row index r increments through the values $1, 2, \cdots N_{ts}$, generating a total of $N_t N_{ts}$ symbols per block of N_{sb} input symbols.

C. Carrier Modulation

The Rect-to-polar and Compare-value conversion blocks prepare the baseband symbols for carrier modulation. The detailed implementation of the two blocks is shown in Figure 6. After symbol mapping and diversity scheme, the symbols $s_n[t]$ must be converted to the polar form (magnitude and phase) for carrier modulation. The TMS320F28379D contains a trigonometric math unit (TMU) that provides accelerated trigonometric instructions for floating point numbers. Floating point casts cost 2 cycles. To find the magntiude of each symbol, the FSQRT32 instruction is available as a compiler intrinsic. It takes 5 pipelined cycles to complete. The atan2 operation is accomplished with a combination of the QUADF32 and ATANPUF32 instructions, where QUADF32 computes the per-unit quadrant offset and the ratio of the real and imaginary parts. QUADF32 executes in 5 pipelined cycles. ATANPUF32 computes the per-unit arctangent of the ratio which can then have the quadrant offset from QUADF32 added to produce the per-unit ATAN2 value (i.e. correct quadrant selection). These two instructions are combined in a compiler intrinsic. ATANPUF32 executes in 4 pipelined cycles.

The polar-form complex symbols are modulated with a carrier by the EPWM peripherals. Figure 7 shows the C2000 hardware used to carry out the modulation task. The current frontend amplifier hardware can only adjust phase, and the symbols are constrained to lie on the unit circle. Therefore, the phase value is the only part used. However, all steps of the pipeline before to the modulation may be reused with a scheme that allows amplitude control. The modulator consists of one C2000 EPWM module, running at the carrier frequency f_{cc} with symbols modulated to the carrier at a rate D. The two compare registers CMPA and CMPB determine the phase by controlling rising and falling edges respectively. We define TPWM as the number of PWM counts in a carrier period. CMPA and CMPB are always kept exactly TPWM/2 apart. CMPA is computed from the phase ϕ by:

$$CMPA = \lfloor \frac{\phi}{2\pi} TPWM \rfloor \quad (6)$$

From CMPA, CMPB is computed by:

$$CMPB = \left(CMPA + \frac{TPWM}{2} \right) \% TPWM \quad (7)$$

where % is the modulo operator. As CMPA and CMPB are computed for each symbol period, they are stored in per-channel buffers according to their index in the symbol stream. Once this is complete, the frame is ready to transmit. Figure 8 shows how the phase for gray-coded QPSK constellation symbols is generated using the PWM configuration in this work. Each new phase is loaded for all channels at the EPWM1 compare interrupt and applied synchronously.

For MIMO transmission, the EPWMs' phases (the phases of the carriers) must be synchronized and have known relationships to each other. Three features of the C2000 EPWMs facilitate this: inter-module synchronization, the global sync event, and shadow registers. First, multiple or all of the EPWMs may be made to respond to a single synchronization signal by tying the EPWMxSYNCO output of EPWMx to the EPWMxSYNCI input of module y. Synchronized EPWMs share the same timebase as the generator of the sync signal. Additionally, using the EPWMxLINK bits, compare register updates may be performed simultaneously for all linked EPWMs. Second, a synchronized global load signal may be provided by software using the OSHTLD bit of the GLDCTL2 register. This global one-shot load may be synchronized across all linked registers. These together provide a guarantee that the EPWM compare registers in all modules will be updated simultaneously. Third, the compare registers in the C2000 EPWMs are shadowed, or consist of two copies, known as shadow and active. The active copy is used by the peripheral itself, but cannot be written to directly. The shadow copy is not used by the peripheral, but may be written to. Its value is dumped to the active copy on some predetermined event; here the software-generated global load sync from the OSHTLD bit is used. The result of this configuration is that all symbol changes are simultaneous, and all transmitter outputs have a defined phase relationship to each other.

In a typical PWM module, the compare interrupt must fire every PWM period, or equivalently in this work, the carrier period. The carrier frequency $f_{cc} = 200$ kHz is a rather high frequency for an interrupt in the C2000. Using the interrupt prescale feature of the EPWMs, we reduce the frequency of the interrupt to the symbol rate D. The interrupt prescale may be set from 1 to 15 – the number set determines how many times the interrupt fires is raised before entry into the interrupt handler occurs. If we set this to T_{ps}, as given by:

$$T_{ps} = \lceil \frac{f_{cc}}{D} \rceil \quad (8)$$

we will only execute the interrupt handler at the symbol period boundary. This frees CPU time for other tasks or entry to IDLE mode to reduce current consumption.

One may question how the system can be modified to support both amplitude and phase control and enable arbitrary precoding schemes and the use of constellations such as QAM. To achieve amplitude control and diversity schemes that modify amplitude, no changes to the baseband processing flow are required. The system can already multiply an arbitrary precoding matrix to the input symbols and obtain phase and

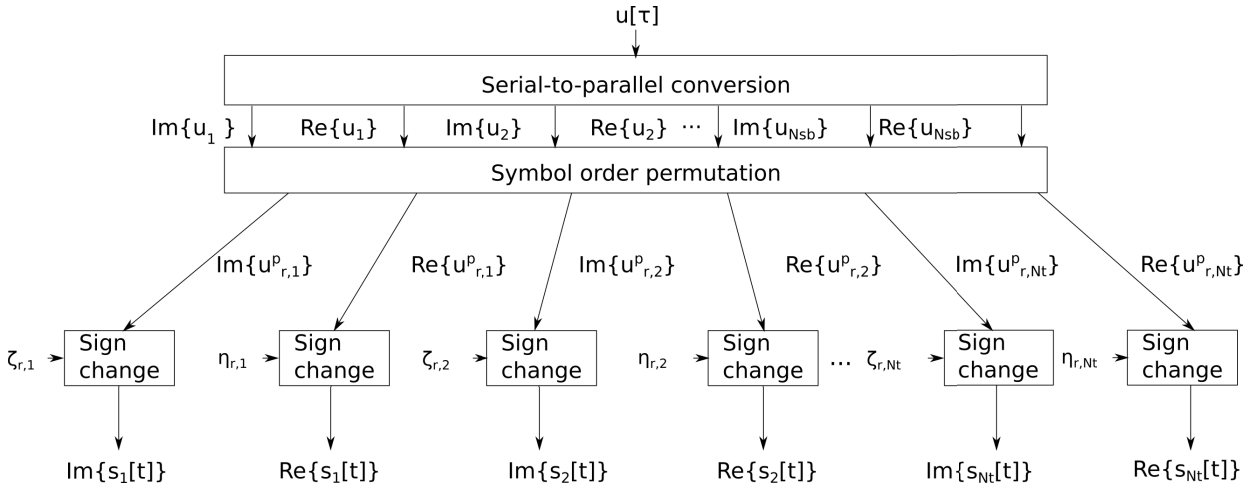


Fig. 5: STBC encoding implementation flow

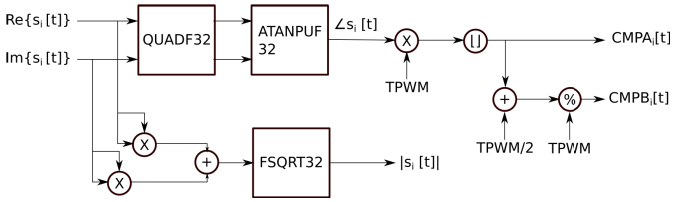


Fig. 6: Algorithm flow from bit-to-symbol mapping to compare value generation

magnitude for each. However, the carrier modulation and analog frontend must be modified. The analog frontend should be a class-D amplifier with a low-pass characteristic. The amplifier should take as input a PWM signal at some multiple of the carrier frequency f_{cc}, $f_{pwm} = N_{spt} f_{cc}, N_{spt} \in \mathbb{Z}$, where each PWM cycle's duty corresponds to a sample from the sinusoidal carrier. The intended structure for one channel of this arrangement is shown in figure 9. In this figure, the phase $\angle s_i[t]$ and magnitude $|s_i[t]|$ from the diversity scheme are used to generate one period of modulated carrier samples $CMPA_i[m]$, which are written to lookup table PING while the prior generated carrier is being written to the EPWM from the lookup table PONG. The carrier sample index m is N_{spt} times faster than the input index t and each buffer has N_{spt} entries which will be sent T_{ps} times before the next buffer switch, as defined in eq. 8. The C2000 DMA controller feeds the EPWM compare register the compare values associated with the prior symbol, $CMPA_{i-1}[m]$, from PONG via a DMA channel configured in circular mode. The source and destination addresses in the DMA module are shadowed, and each DMA channel is configured to generate an interrupt at the beginning of the transfer. In the interrupt, the shadowed source address can be switched from PONG to PING and a flag triggering regeneration of the carrier can be set. The compare values $CMPA_i[m]$ are then transferred T_{ps} times from PING while the values $CMPA_{i+1}[m]$ are generated and written to PONG. The DMA controller's address wrapping feature allows multiple cycles of the carrier to be sent in one transfer without CPU interaction. After T_{ps} transfers the source address from the last interrupt will become active, the next symbol will be sent, and regeneration of the carrier samples will be triggered. The C2000 TMU implements the COSPUF32 instruction, which accelerates the computation of cosine values to 2-3 pipelined cycles depending on the next instruction, reducing generation time for the next symbol's modulated carrier. When regeneration of the inactive table is triggered, this instruction is used to recompute all samples of the carrier with the phase and magnitude of the next symbol in the transmit element's stream applied. If the EPWMs are started synchronous to each other using TBCLKSYNC and never disabled as is already done for the beamforming-only modulation design, and the DMA is fast enough to continuously feed carrier samples for all channels, they will remain synchronized for the full frame. This technique should be implemented in the future to enable the use of more complex diversity schemes that require constellations whose symbols vary in magnitude (for example QAM or HEX).

D. Data Frame Control

The structure and timing of the transmitted data frame are shown in Figure 2. In this work, the transmitted N_t simultaneous BPSK-modulated m-sequences are used both as symbol synchronization and training symbols for channel estimation, followed by a configurable-length gap of N_G symbol periods to lower or eliminate inter-block interference (IBI), followed by the M-PSK payload, where M is the modulation order.

An m-sequence is a maximum length non-repeating binary sequence generated by a linear feedback shift register (LFSR) of a given size. The input bits to the LFSR are generated using the XOR of a subset of the bits in the register, called the "taps". For a p-bit LFSR, the taps generating a maximum length sequence are given by the primitive polynomials of degree p over GF(2). For SIMO transmission, any arbitrary polynomial can be chosen. For the MIMO case, we generate all p-length GF(2) polynomials that give an m-sequence by

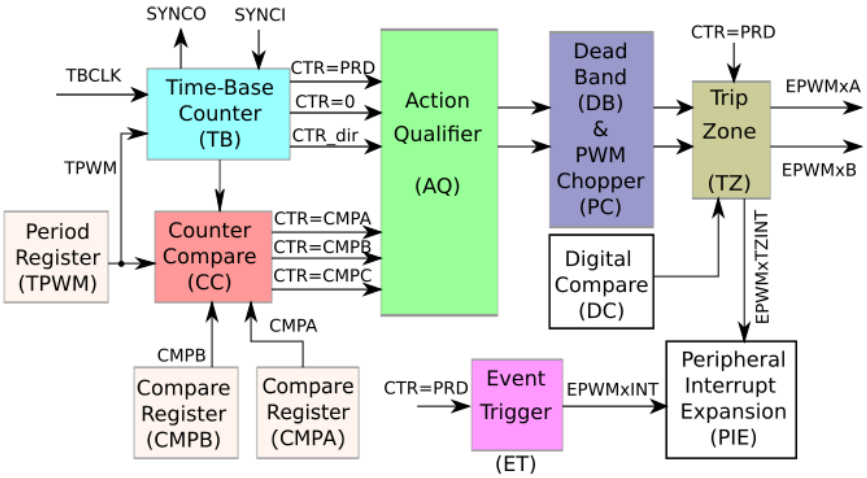


Fig. 7: EPWM submodule configuration for carrier modulation

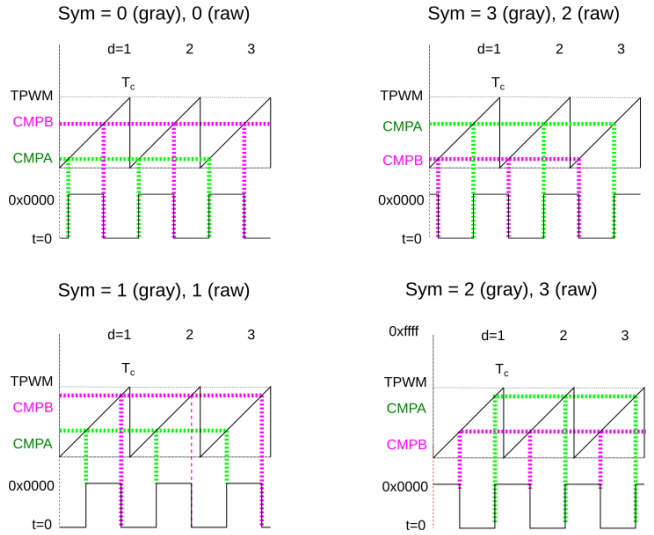


Fig. 8: Gray-coded QPSK EPWM waveform-generation diagram

finding the primitive polynomials of $GF(2^p)$, then select N_t polynomials such that any two of their generated m-sequences have 0.1 or lower normalized cross-correlation. Regardless of the modulation order selected for the payload, BPSK modulation is used to transmit the M-sequences. Simulations show that accurate symbol synchronization is still possible when individually, randomly delayed m-sequences are summed together (Figure 10). The N_t chosen m-sequences are reused as training symbols for the channel estimator. This process is controlled by a state machine in the EPWM1 interrupt.

To control the gap, the design simply sets $CMPA > TBMAX$ on the corresponding EPWM channels. This guarantees each channel stops on the same carrier cycle, and to restart transmission is to modulate the next symbol by writing new CMPA/CMPB values. This is preferred to the trip zone because the trip zone cannot be triggered simultaneously by software for multiple channels. Disabling only the output is preferred to stopping the timebase count because it allows the gap length to be tracked precisely in terms of symbol intervals, without arming another timer or other peripheral.

The payload is transmitted using phase-shift keying (PSK) with the selected modulation order. The phases of all channels' carriers are synchronized to each other, and the phase relationship between the carriers on all transmit elements should always be known. Preloading of new symbols is synchronized to EPWM1's symbol interval. When the prescaled EPWM1 interrupt fires, all channels are preloaded with compare values for the next symbol, and a one-shot global load trigger is sent. All channels are linked to EPWM1.

IV. Results and Performance

Several tests was conducted at the transmitter output using an oscilloscope and logic analyzer. The phases of the transmitted symbols for the first set of encoded STBC symbols, and for one symbol value modulated on all 4 channels after phase shifts applied via the precoding matrix, are shown in figures 13 and 14 respectively. A logic analyzer was used to capture and check a full STBC-encoded data frame against a simulated data frame encoded with Tarokh's 4-transmitter STBC matrix and verify the carrier-modulated symbols were the same as the simulated encoded symbols for a predetermined input sequence and QPSK constellation.

A 512-bit pseudorandom sequence was generated, and the sequence was encoded, modulated, and sent using the implemented transmitter. An oscilloscope was attached to the EPWM pins of the microcontroller. Using a bandpass filtered version of the individual EPWM outputs, carrier demodulation was performed for the payload signal blocks and all bits of the payload for BPSK, QPSK, and 8-PSK were recovered. The bandpass filter has a passband spanning the interval [fcc - D, fcc + D], where D is the symbol rate of the system. This process is equivalent to passing the signal through an ideal channel with no intersymbol interference and very little noise. The complex baseband payload symbol constellations

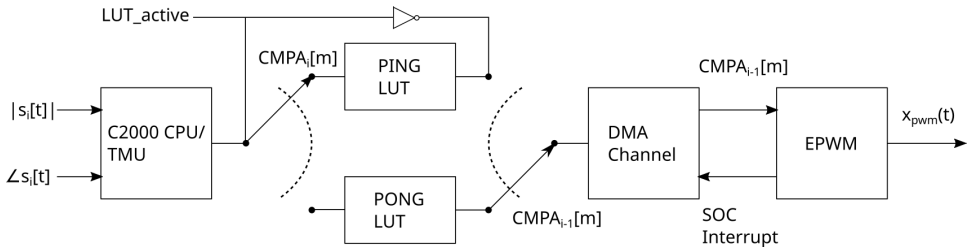


Fig. 9: Structure of potential PWM DAC modulator design (one channel branch)

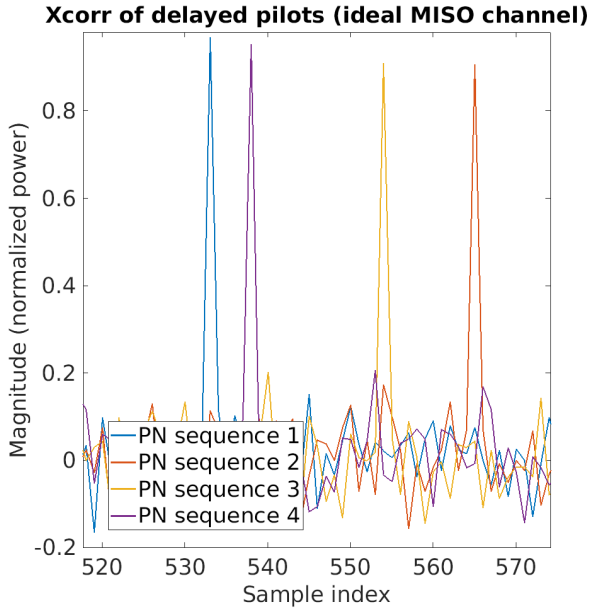


Fig. 10: Symbol synchronization with ideal MISO channels

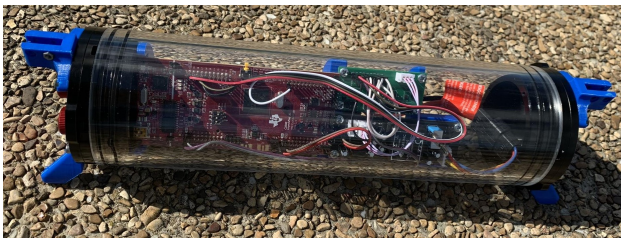


Fig. 11: Photograph of the transmitter hardware

obtained by this process are shown in Figure 12 for a 512-bit pseudorandomly-generated bit sequence in BPSK, QPSK, and 8-PSK, respectively. The mean squared error (MSE) of the recovered symbols from their nominal values is 0.0065 for BPSK, 0.0295 for QPSK, and 0.0250 for 8-PSK.

Measured via the Texas Instruments Code Composer Studio profiling clock, the time for a 512-bit frame with 4 channels to be encoded is 24 ms, which yields a maximum of 21.3 kbps throughput – above the target maximum rate of 20 kbps.

V. Conclusion

In this work, an underwater acoustic transmitter design that emphasizes small size and reliability through multi-channel operation was demonstrated. Convolutional encoding, interleaving, and diversity scheme processing is performed digitally, and the modulation is performed using the PWM channels of the microcontroller, operating at the carrier frequency f_{cc}. Only bandpass filtering, voltage boosting, and voltage shifting are done outside the microcontroller to produce the final output signal. The mean square error in the captured baseband symbols from the transmit side is computed and shown to be 3% or less in the worst case. The execution time to encode one frame is measured, demonstrating that the system achieves a higher worst-case encoding speed than the application's minimum throughput for continuous streaming. Constellations and validation of the frame synchronization and training procedures have been shown to further demonstrate the correct operation of the transmitter.

References

[1] J. Mietzner, R. Schober, L. Lampe, W. H. Gerstacker, and P. A. Hoeher, "Multiple-antenna techniques for wireless communications - a comprehensive literature survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 87–105, 2009.

[2] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1456–1467, 1999.

[3] L. Zheng and D. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, 2003.

[4] A. H. Mehana and A. Nosratinia, "Diversity of mimo linear precoding," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1019–1038, 2014.

[5] Y. Xin, Z. Wang, and G. Giannakis, "Space-time diversity systems based on linear constellation precoding," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 294–309, 2003.

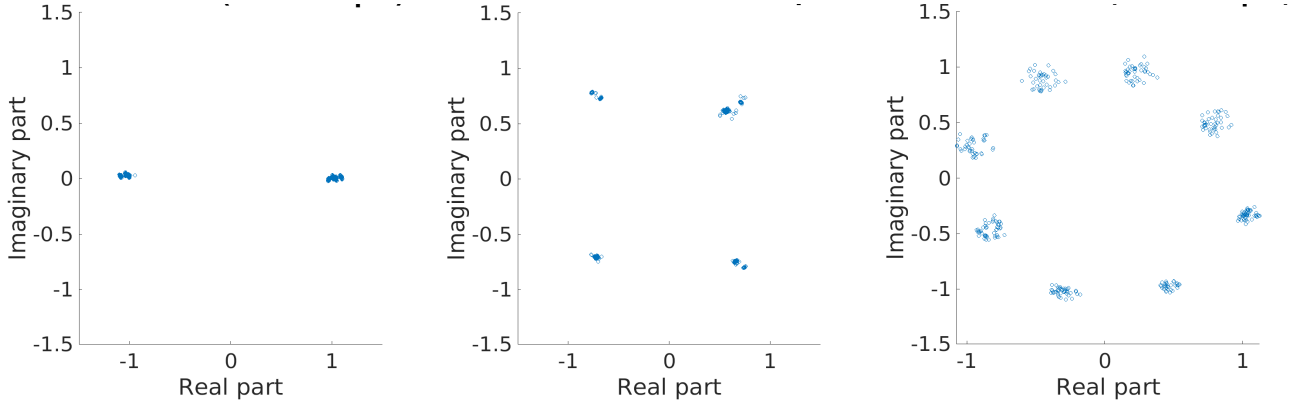
[6] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[7] C. Xu, N. Ishikawa, R. Rajashekar, S. Sugiura, R. G. Maunder, Z. Wang, L.-L. Yang, and L. Hanzo, "Sixty years of coherent versus non-coherent tradeoffs and the road from 5g to wireless futures," *IEEE Access*, vol. 7, pp. 178 246–178 299, 2019.

[8] A. Aboutaleb, W. Fatnassi, Z. Rezki, and A. Chaaban, "Optimal diversity and coding gains for millimeter-wave communication," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4601–4614, 2021.

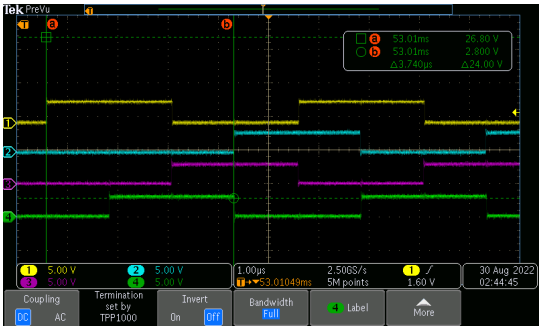
[9] M.-T. Le, T.-D. Nguyen, X.-N. Tran, and V.-D. Ngo, "On the combination of double space time transmit diversity with spatial modulation," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 170–181, 2018.

[10] S. E. Anderson. (2022) Bit twiddling hacks. [Online]. Available: https://graphics.stanford.edu/~seander/bithacks.html

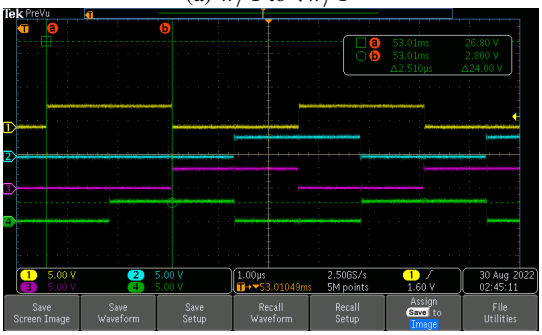


(a) BPSK (b) QPSK (c) 8-PSK

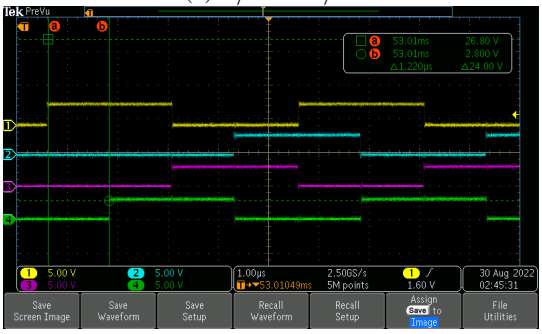
Fig. 12: Constellations of measured output at C2000 EPWM pin1 for the implemented transmitter



(a) $\pi/4$ to $7\pi/4$

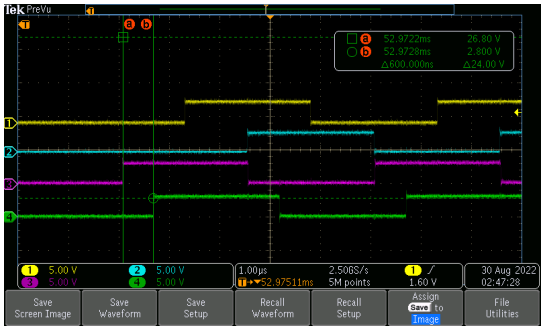


(b) $\pi/4$ to $5\pi/4$

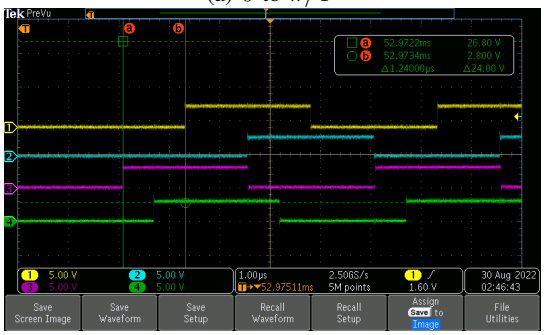


(c) $\pi/4$ to $3\pi/4$

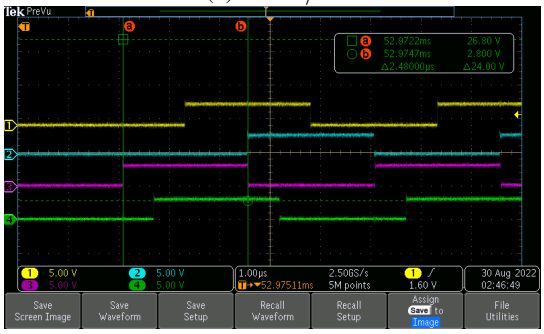
Fig. 13: Phase differences between first 4 encoded STBC symbols in test payload



(a) 0 to $\pi/4$



(b) 0 to $\pi/2$



(c) 0 to π

Fig. 14: Phase differences between 4 identical symbols phase shifted with beamforming method