AFRL-RV-PS-TR-2019-0135

AFRL-RV-PS-TR-2019-0135

SATELLITE DIGITAL CHANNELIZER BEAMFORMING WITH PREDISTORTED HIGH-POWER AMPLIFIER

Hyuck M. Kwon, et al.

Wichita State University
Department of Electrical Engineering and Computer Science
1845 N. Fairmount Ave.
Wichita, KS 67260-0083

10 November 2019

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

DTIC COPY

NOTICE AND SIGNATURE PAGE

AFRL-RV-PS-TR-2018-0135 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//	//SIGNED//
Dr. Khanh Pham	Dr. Thomas R. Caudill, Chief
Program Manager, AFRL/RVB	AFRL Geospace Technologies Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
10-11-2019

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
04 Apr 2018 – 05 Nov 2019

4. TITLE AND SUBTITLE
Satellite Digital Channelizer Beamforming with Predistorted High-Power Amplifier

5a. CONTRACT NUMBER
FA9453-18-1-0102

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63401F

6. AUTHOR(S)
Hyuck M. Kwon, Madhuprana Goswami, and Vidhi Rana

5d. PROJECT NUMBER
6368

5e. TASK NUMBER
PPM00093307

5f. WORK UNIT NUMBER
EF130891

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Wichita State University
Department of Electrical Engineering and Computer Science
1845 N. Fairmount Ave.
Wichita, KS 67260-0083

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Avenue SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVBY

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TR-2019-0135

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report studies for the first time a digital channelizer combined with a digital high-power amplifier (HPA) and a digital predistorter (PD) for future satellite onboard processing. A digital channelizer, called a polyphaser, takes multiple access user (MAU) frequency-division (FD) sub-band signals, converts the multiple MAU sample streams into a single sample stream, and then feeds it into a digital PD followed by an memoryless HPA. Hence, no overlapping in the time domain among MAU samples occurs, so undesirable intermodulation (IM) products can be avoided, whereas an analog HPA operating near a saturation point causes significant IM product degradation at the HPA output when a sum of MAU sub-band signals are fed into it. The proposed digital PD can compensate almost perfectly for both phase and amplitude distortion caused by the HPA's nonlinearity. Then, individual digital beamforming (BF) is applied after the HPA for each MAU sub-band signal sample stream to direct to each individual target terminal. Finally, this report verifies the performance of the proposed digital channelizer combined with the digital PD and HPA by presenting both the analytical and simulation bit error rate (BER) results, thereby concluding that the overall degradation is negligible.

15. SUBJECT TERMS
beamforming, digital channelizer, high-power amplifier, poly phasor, predistortion, satellite communication

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	Unlimited	62	Dr. Khanh Pham
					19b. TELEPHONE NUMBER *(include area code)*

This page is intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA9453-18-1-0102. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

DISCLAIMER

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

This page is intentionally left blank.

1.0 SUMMARY

This report achieved the objective stated in the original proposal during the period from August 8, 2018, to August 10, 2019, by doing the following:

"Investigate an adaptive digital satellite beamforming in combination of a digital channelizer onboard satellite transponder for each channel per beam or combined several channels per beam. Configure channel beam steering with consideration of channel bandwidths, carrier frequencies, shapes of beams and main lobe directions. Design a digital channel beamforming (DCB) codebook consisting of channel beam steering vectors that will be employed at a reconfigurable satellite digital channelizer beam former. Study how many users can be supported per carrier by the DCB for power and spectrum efficiency as well as interference mitigation."

And the following attached conference paper was accepted for presentation and publication:

Madhuprana Goswami, Vidhi Rana, Hyuck M. Kwon, Khanh Pham, and James Lyke, "Satellite Digital Channelizer with Predistorted High-Power Amplifier," IEEE Military Communications Conference, Norfolk, VA, November 12-14, 2019.

2.0 INTRODUCTION

Motivation

The presence of undesirable intermodulation (IM) products in a satellite link is due to the nonlinear components, e.g., an analog high-power amplifier (HPA) in a satellite transponder. Specifically, this occurs when multiple access user (MAU) frequency-division (FD) sub-band signals are summed and fed into an analog HPA. This also occurs when the HPA operates near its saturation point. Typically, a geostationary earth orbit (GEO) satellite waveform traveling distance can be more than 36,000 km, and high transmitting power (e.g., 20 dBW = 100 W) is required. Consequently, undesirable IM products are generated at the HPA output, thereby causing non-negligible system performance degradation. As shown in Fig. 1, the amplitude of input to the HPA is = 0.93, for example, which is almost 1, i.e., the HPA is operating near its saturation point. Due to nonlinear characteristics of the HPA operating near its saturation point, undesirable IM products are generated at the HPA output. Carrier frequencies for the user 1 transmitted signal is 0.2 MHz, for the user 2 transmitted signal is 0.4 MHz, and for the user 3 transmitted signal is 0.6 MHz. Due to the HPA's nonlinear characteristics, we can observe in Fig. 1 undesirable spikes at 0.19 MHz, 0.21 MHz, 0.39 MHz, 0.41 MHz, 0.59 MHz, 0.8 MHz, and so on. Fig. 1 also shows in red that, when the HPA is not present, the undesirable IM products are not observed.

Therefore, the motivation of this report is the following question: Can we remove these undesirable IM products that cause non-negligible system performance degradation for satellite communication system with HPA?

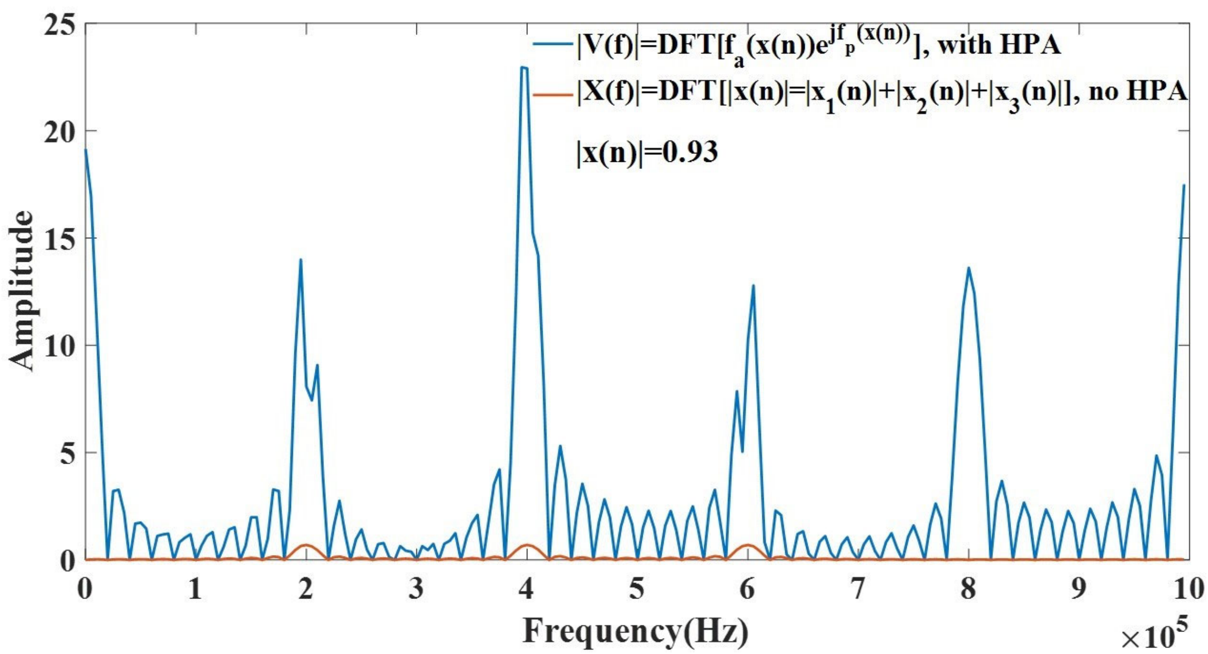


Fig. 1. Intermodulation products generated due to nonlinear HPA

Objective

The objective of this report is to find a communication system solution that shows no IM products, even when an HPA takes multiple FD sub-band signals as its input and operates near its saturation point. To solve the IM product problem and to achieve the objective, this report considers a memoryless HPA, the input and output of which are a sequence of sample streams in discrete time. Throughout the report, digital refers to the discrete time sample with an infinite quantization level. A finite quantization level could be used, but this report does not consider any quantization loss. An ideal HPA, as assumed in this report, is not yet available in practice. However, a practical digital device to control the analog HPA outputs digitally has been discussed [1].

In addition, some of the literature discusses the output of the power amplifier modeled as a discrete time sample [2]. Therefore, this report assumes that such a HPA exists. Here, the authors cannot consider all practical implementation issues related to the proposed method, e.g., when an analog HPA must be employed because no HPA exists now, so we employ a digital-to-analog (D/A) converter to change the predistorter (PD) digital output sequence into an analog signal and then feed it into an existing analog HPA. This D/A conversion will introduce loss, such as quantization loss, which is not investigated here. Rather, the authors propose a feasible concept about how to solve the IM product problem theoretically for a future satellite HPA. All other related issues are left for future study in order for the proposed method to be utilized in practice.

Literature Survey

Digital beamforming (BF), or precoding, has been applied recently for terrestrial communications [3] but has not been investigated much for satellite links by including a digital channelizer and HPA. In [4], the authors introduced a BF technique and power allocation scheme and mentioned a digital channelizer BF model. However, the authors in [4] did not consider any combination of the digital channelizer, the digital PD, and the HPA investigated in this current report, and did not present any simulation results. In [5], the authors developed a mathematical and simulation model to improve satellite communication on-the-move operating in the X band, Ku band, and Ka band. The authors further included a digital channelizer beamforming model with a 500-MHz channel bandwidth (BW) and subdivided it into four 125 MHz channels. These subchannels were further subdivided into other subchannels. Although digital channelizer beamforming is discussed in [5], the authors did not include the HPA for their system model and simulations. Hence, in this current report, we introduce for the first time a combined method of a multiuser digital channelizer, digital PD, HPA, and BF.

In a general terrestrial multiple-input-multiple-output (MIMO) communication system with many antenna elements, multiple digital BF algorithms can be executed at the digital baseband instead of phase array antenna (PAA) BF at the radio frequency (RF). For satellite BF, the feasibility of such a terrestrial baseband digital MIMO BF has not yet been thoroughly investigated. This is mainly because the angle spread width from a GEO satellite to multiple terminals on the ground are very small, e.g., 0.2 degrees. Hence, the rank of a satellite MIMO channel per carrier can be very small, such as one. The authors were unable to find any literature on the rank study for a MIMO GEO-satellite channel. Therefore, this report applies the known PAA BF.

The conventional Saleh model, a memoryless high-power travelling-wave tube amplifier (TWTA) model, can still be used, even if a HPA is considered in this report. This is because the Saleh HPA is memoryless and exhibits nonlinear distortions in both amplitude (AM/AM conversion) and phase (AM/PM conversion) at a given sample time [6]. Therefore, the Saleh model will be used in this report.

Some existing works pertain to pre-distortion techniques in order to compensate for the nonlinear impairments due to the HPA. For example, in [7], the authors have applied a digital PD to a complex baseband signal in the form of a complex multiplicative gain. They also employed an algorithm that uses a set of parameters to store the PD gain and phase information, which are further used to build a look-up table (LUT) containing complex gains as a function of input power. In [8], a family of adaptive digital signal PD schemes compensate for nonlinear distortion with memory. The authors in [8] used recursion of the stochastic approximation type for finding the zero-crossing such that nonlinear distortion drives toward zero. In [8], the discrete time Volterra model was used for representation of a nonlinear system with memory. In [9], the authors combined a TWTA model with a linearizer such that the gain increase of the linearizer cancels the amplifier's gain decrease, and the phase change of the linearizer cancels the phase change of the amplifier. In [10], the authors used an LUT-based method to compensate for HPA nonlinearities with memory.

3.0 METHODS

In this report the following two system models are studied:

A. **Satellite Digital Channelizer with Predistorted HPA and No BF; Collocated Receivers;**

B. **Satellite Digital Channelizer BF with Predistorted HPA; Spatially Distributed Receivers;**

In both the above system models, our proposed predistorter (PD) method takes each sample $x(n)$ from a digital channelizer output stream, and computes two variables: an amplitude PD and a phase PD. These two variables are designed so that the overall HPA output with the proposed PD has zero phase distortion and the same amplitude as the HPA output for the no-PD case for any given input sequence $x(n)$.

The simple digital PD design proposed by us helps to pre-compensate for the phase and amplitude distortion by the HPA. The proposed PD does not employ any LUT as done in majority of existing PDs but rather takes each sample from the digital channelizer output and then simply computes how much phase and amplitude should be pre-compensated for each sample so that the output of the HPA may have the same amplitude as the no-PD case and zero phase distortion after the HPA. These pre-compensation values are computed for each input sample amplitude by using known HPA amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) characteristics.

Also, we use the Saleh's memoryless HPA AM/AM and AM/PM characteristics to compute the two variables. Our numerical results demonstrate that the proposed PD can compensate for the HPA nonlinear impairments almost perfectly, even when the multiuser digital channelizer output signals are fed as inputs to the PD followed by HPA. The proposed PD method does not use any LUT, and hence its complexity is simpler than any existing PD methods, e.g., those in [7]-[10]. This is because our PD method computes only two simple equations.

This PD method is feasible because the input and output of a HPA are samples. In addition, there is no memory in a HPA. If the digital PD can make the HPA amplitude including PD equal to that of the HPA of no PD and if the digital PD can make the overall phase distortion from the digital PD to the HPA to be zero, then the output of the HPA will show no IM products. This is because only one sample is fed into the HPA for one sample output. No sum of samples coming from different sub-band signals are fed into the HPA. Furthermore, the frequency component of each sample is already separated from adjacent samples in the frequency domain by using the BPFs, and therefore no intra-and inter-interference will be experienced from a sample to a next sample.

Multiple HPAs are not necessary to amplify the multiple access signals individually because the digital channelizer output is a single sample stream. The HPA amplifies each input sample by sample. Then the discrete output of the HPA is converted to a continuous time signal and up-converted with a main carrier frequency for a down-link transmission from the transponder to multiple access ground users.

In addition, distortion from the PD input to the HPA output is negligible. Furthermore, the overall distortion from the digital channelizer transmitter input to the digital channelizer receiver output, including practical transmit and receive BPFs, PD, and HPA under an additive white Gaussian noise (AWGN) environment, is negligible. In other words, the overall bit error rate

(BER) versus the bit energy-to-noise power-spectral density ratio E_b/N_0 curve for the proposed scheme shows the same theoretical BER for the considered 8-ary phase-shift keying (8PSK) modulation under the same AWGN environment when the theoretical system assumes no digital channelizer, no BPFs, no PD, and no HPA.

We also implement phase array antenna BF with the digital channelizer, PD, and HPA. We observe that the BER versus E_b/N_0 curve for the proposed digital channelizer with predistorted HPA shows that the BER is 14 dB better than that of the BER for the considered 8PSK modulation under AWGN environment when no BF is involved and when the theoretical system assumes no digital channelizer, no BPFs, no BFs, no PD and no HPA. This is because we use five antenna elements per each MAU, and the PAA power gain is 14 dB.

Finally, to achieve the objective, as stated in section 2.0, we propose for the first time a system model that combines a digital channelizer (poly-phaser or filter bank), a digital pre-distorter (PD), a HPA and a PAA BF.

4.0 ASSUMPTIONS

A digital channelizer employs multiple up-samplers at the transmit (TX) side and multiple down-samplers at the receive (RX) side. The number of pairs is equal to the number of MAU FD sub-bands. Also, the digital channelizer uses a pair of corresponding digital band-pass filters (BPFs) for each MAU branch: one at the TX side, and the other at the RX side. The up-samplers can be located at the output of a decode-and-forward (DF) satellite transponder front end and before a common HPA. The digital channelizer at the TX side will convert MAU FD digital sub-band input samples in parallel into a single serial sample stream with no overlap in time. Also, a digital channelizer down-sampler at the receive side at a common terminal receiver or distributed multiple terminals on the ground can convert the serial sample sequence back into multiple MAU output sample sequences in parallel, and each MAU receiver can pick and collect its desired samples without errors when the sampling time is synchronized. Overall, the output sequence of a digital channelizer can be regarded as a frequency-multiplexed version of the set of MAU sample sequences [3], [4].

In this report, we consider a digital channelizer for MAU FD signals like the one in [3], [5]. This digital channelizer using multiple polyphase bandpass filter banks is equivalent to a multichannel digital transmitter that simultaneously up-converts a multiple number of baseband signals into a set of frequency-division-multiplexed (FDM) channels. Similarly, a multichannel receiver simultaneously down-converts a set of FDM channels using polyphase bandpass filter banks [2]. The digital channelizer can be applied for a future digital satellite system design. For example, the multiple access FDM-like signals can be transmitted from a ground controller to a DF satellite transponder, which down-converts its received signal into multiple baseband signals in parallel at the transponder front end. Then the digital channelizer placed at the transponder front end converts the multiple access baseband signals into a single sample stream and feeds it into a digital PD followed by a single digital HPA before the satellite down-link transmission.

Finally, the proposed digital channelizer does not expand the bandwidth compared to the frequency-division-multiplexed channel with no digital channelizer. If the FDM channel with no digital channelizer requires a BW equal to the number of MAU times B Hz (i.e., BW per user), then the proposed digital channelizer also requires the same BW because it can employ the up-samplers at a rate equal to the number of MAUs times the original complex value sampling rate B Hz per user. Therefore, a future satellite communication system using the proposed digital channelizer, PD, digital HPA, and PAA BF can have high-frequency utilization efficiency, and multicast and broadcast capabilities, and can gain control for each subchannel. PAA BF technology will help the receivers with an improved signal quality.

In this report we study, for the first time, the feasibility of a digital channelizer phase array antenna beamforming application for a digital satellite transponder to communicate frequency division multiple access user sub-band signals via a nonlinear high-power amplifier and a simple pre-distorter under an additive white Gaussian noise environment. The same conclusion under the other practical satellite channel environments such as a Rician fading channel, i.e., negligible overall bit error rate degradation when BF is not involved and an approximated 14 dB BER improvement when BF is involved with five antenna elements per user, can be expected.

Throughout this report, lowercase boldface letters denote column or row vectors. All italicized letters denote scalar quantities. The symbol $|x(n)|$ denotes the amplitude of a complex variable $x(n)$, and $E[X]$ stands for the expectation of a random variable X.

5.0 PROCEDURE (THEORETICAL ANALYSIS)

A. Satellite Digital Channelizer with Predistorted HPA and No BF; Collocated Receivers;

In this section, assume that one ground hub terminal receives the digital channelized signal sent from a GEO satellite and demodulates all MAU sub-band signals together and distributes each MAU signal individually via ground communication networks. Assume also that there is no beamforming. Fig. 2 shows the corresponding system block diagram for this case: no BF, multiuser digital channelizer, digital PD, and digital HPA under AWGN. The AWGN noise vector is denoted by $\boldsymbol{q}$ of a data block size n_{total}. The digital PD is placed before the digital HPA. In Fig. 2, N denotes the up-sampling and down-sampling factors, while M denotes the number of channels or the number of sub-band users. For a practical communication system, $N > M$ should be satisfied. As the ratio $N > M$ increases, the bandwidth allocation of each sub-band user signal decreases, but the interference from neighboring sub-band user signals decreases. In other words, the ratio $N > M$ can control the bandwidth efficiency versus the performance degradation due to the neighboring sub-band signal interference [11].

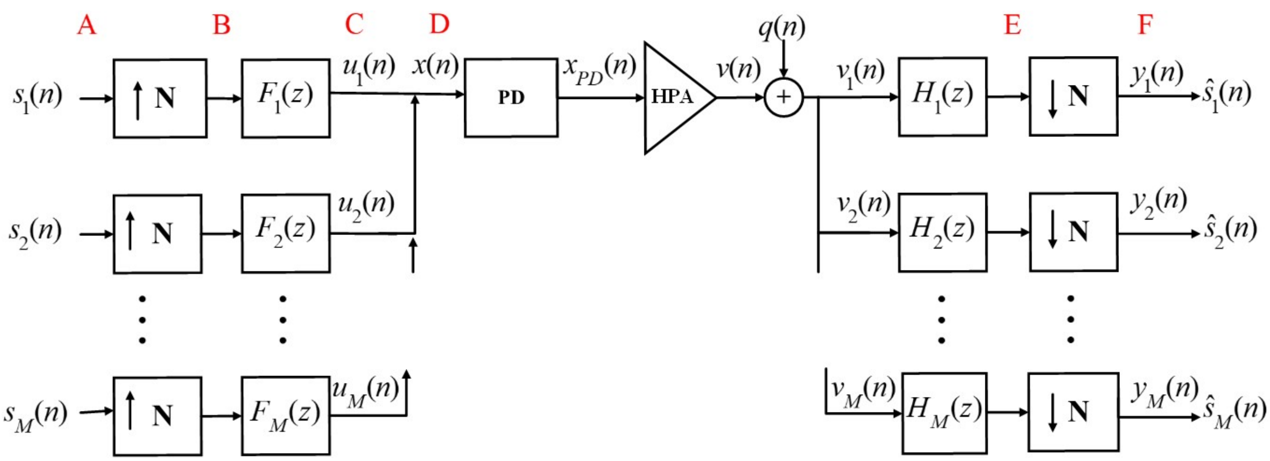


Fig. 2: Multiuser digital channelizer, digital PD, digital HPA, and no PAA BF, where all MAU signals are demodulated by common receiver

Vector $\boldsymbol{s_k} = \left[s_k\left(n\right), s_k\left(n+1\right), \cdots, s_k\left(n+n_{total}-1\right)\right]^T$ of a data block size n_{total}, denotes a vector of 8PSK modulated input symbols for each user $k = 1, \cdots, M$. At the transmitter in the DF transponder, the up-sampled input signal is passed through the transmit BPFs with z-transfer function $\boldsymbol{F_k}(z)$ for each user sub-band signal, $k = 1, \cdots, M$. Each up-sampler takes a 8PSK baseband sample $s_k\left(n\right)$ and inserts $\left(N-1\right)$ number of zeros before next sample $s_k\left(n+1\right)$. Hence the bandwidth of each up-sampled signal is reduced by factor N, and the spectrum is copied into N number of images in the angular frequency range from 0 to 2π equally. The BPF $\boldsymbol{F_k}(z)$ takes an image spectrum located at the *k*-th desired sub-band and generates output sample sequence $u_k\left(n\right)$. Then a multiplexer converts the M parallel sequences into a single sequence stream $x\left(n\right)$.

This system is called a synthesis filter bank because it combines a set of signals $\boldsymbol{s_k}$ into a single sample stream sequence signal $x(n)$ [11] and works as a polyphase filter bank. The $x(n)$ is fed into the PD followed by HPA. The output $v(n)$ of HPA is transmitted to a ground hub terminal through a satellite channel medium under an AWGN environment $q(n)$.

At the receiver side, i.e., an analysis filter bank, a de-multiplexer converts back the single stream signal vector $\boldsymbol{v}$, which is the output of the HPA, into M individual parallel components for M channels, denoted by $\boldsymbol{v_k}$ of vector size equal to the data block size n_{total}. In Fig. 2, the analysis filter z-transfer function is also a BPF denoted by $\boldsymbol{H_k}(z)$ which is equal to $\boldsymbol{F_k}(z)$, for $k=1,\cdots,M$ receiver [11]. The output of the BPF is down-sampled with the same factor N. Hence all zeros which were inserted at the transmitter side up-sampler are removed. The down-sampler expands the BW by factor N. However, there is no overall bandwidth change from the up-sampler to the down-sampler. The demodulated symbol at sample time n is denoted by $\hat{\boldsymbol{s}}_{\boldsymbol{k}}(n)$ for $k=1,\cdots,M$ receivers and is obtained after demodulating the down-sampled signal vector $\boldsymbol{y_k}$ of vector size equal to the data block size n_{total}.

The block diagram in Fig. 2 also shows our proposed digital PD scheme followed by the digital HPA, where $x_{PD}(n)$ denotes the PD output sample at time n and is the input to the HPA. We want to design the digital PD so that the amplitude of digital HPA output may be equal to the amount of the original HPA amplitude gain $f_a\left(|x(n)|\right)$ times the input sample magnitude $|x(n)|$ and the phase of the digital HPA output may be equal to the phase of $x(n)$. Let $A_{PD}(x(n))$ and $P_{PD}(x(n))$ denote, respectively, the pre-distortion amplitude and phase amount necessary to achieve the PD design goal.

First, we consider the digital HPA output sample $v(n)$ at time n when no digital PD is employed as

$$v(n) = f_a(|x(n)|)e^{jf_p(|x(n)|)}. \tag{1}$$

The amplitude and phase of the original sample $x(n)$ are distorted by the HPA according to the Saleh's nonlinear and memoryless HPA model, i.e., the AM/AM $f_a(|x(n)|)$ and AM/PM $f_p(|x(n)|)$ characteristics in [6] with (α_a= 2, β_a= 1) and (α_p=4, β_p=9), where

$$f_a(|x(n)|) = \frac{\alpha_a |x(n)|}{1+\beta_a |x(n)|^2} \tag{2}$$

and

$$f_p(|x(n)|) = \angle x(n) + \frac{\alpha_p |x(n)|^2}{1+\beta_p |x(n)|^2}. \tag{3}$$

Second, we consider including the digital PD before the digital HPA. The input to the digital HPA (i.e., PD output) can be expressed as

$$x_{PD}(n) = x(n)A_{PD}(x(n))e^{jP_{PD}(x(n))}. \tag{4}$$

The output of the HPA including PD can be expressed as

$$v(n) = f_a\left(|x_{PD}(n)|\right)e^{jf_p\left(|x_{PD}(n)|\right)}. \tag{5}$$

Also, from (5), the output of HPA in terms of $x(n)$ can be expressed as

$$v(n) = x(n)A_{PD}\left(x(n)\right)e^{jP_{PD}(x(n))} \times \frac{\alpha_a}{1+\beta_a\left|x(n)A_{PD}\left(x(n)\right)\right|^2} \times e^{j\left[\frac{\alpha_p\left|x(n)A_{PD}(x(n))\right|^2}{1+\beta_p\left|x(n)A_{PD}(x(n))\right|^2}\right]}. \quad (6)$$

From (6), we establish two simultaneous equations such that the total phase distortion due to the HPA including the PD is zero and the total amplitude distortion is equal to the original HPA output amplitude with no PD. The total HPA output amplitude including the PD is preserved as the HPA's only output amplitude. In other words, the total amplitude at the HPA output including the PD is set to the original input amplitude $|x(n)|$ times the slope of the original HPA output amplitude with no PD over the original input amplitude $|x(n)|$ using the HPA AM/AM characteristics. The slope is calculated from (2) for the case with no predistortion and is denoted by c. The two simultaneous equations are written as follows: using the phase of (6),

$$P_{PD}\left(x(n)\right) + \frac{\alpha_p\left|x(n)A_{PD}\left(x(n)\right)\right|^2}{1+\beta_p\left|x(n)A_{PD}\left(x(n)\right)\right|^2} = 0, \quad (7)$$

and using the amplitude of (6),

$$A_{PD}\left(x(n)\right)\frac{\alpha_a}{1+\beta_a\left|x(n)A_{PD}\left(x(n)\right)\right|^2} = c. \quad (8)$$

Solving simultaneously two equations (7) and (8) by the substitution method, we find the amplitude predistortion factor, $A_{PD}\left(x(n)\right)$ and the phase predistortion factor, $P_{PD}\left(x(n)\right)$ as follows:

$$A_{PD}\left(x(n)\right) = \frac{\alpha_a + \sqrt{\alpha_a^{\,2} - 4\beta_a c^2\left|x(n)\right|^2}}{2\left|x(n)\right|^2\beta_a c} \quad (9)$$

and by substituting the value of $A_{PD}\left(x(n)\right)$ in (7), we can get

$$P_{PD}\left(x(n)\right) = -\frac{\alpha_p\left|x(n)A_{PD}\left(x(n)\right)\right|^2}{1+\beta_p\left|x(n)A_{PD}\left(x(n)\right)\right|^2} \quad (10)$$

For a given input sample $x(n)$, we can compute the slope c from (2), then the PD compensation amplitude factor $A_{PD}\left(x(n)\right)$ from (9), and then the PD phase compensation factor $P_{PD}\left(x(n)\right)$ from (10). This complexity is so simple and requires only a few multiplication and division procedures. Our simulation results, which are discussed in the next section, show that the phase predistortion factor as obtained through (9) and (10) compensates for the nonlinearity effect due to the HPA almost perfectly for any input symbol $x(n)$ at time n, even for $|x(n)| > 1$ case.

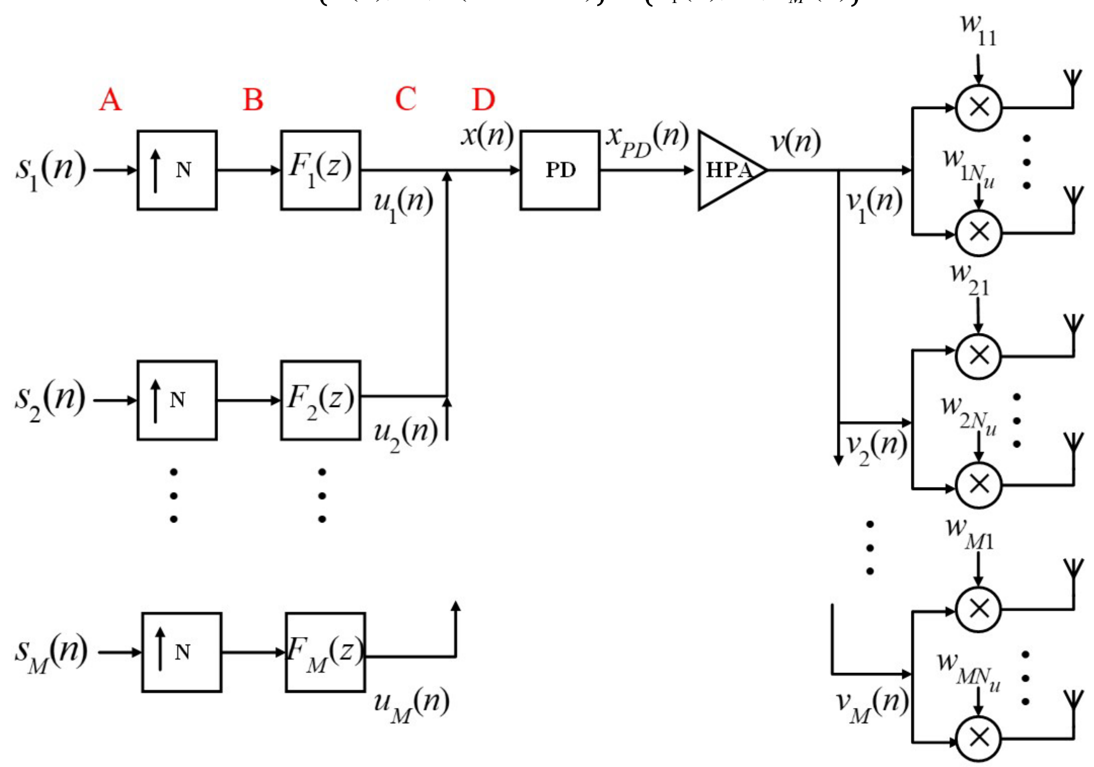
B. Satellite Digital Channelizer BF with Predistorted HPA; Spatially Distributed Receivers;

In a practical scenario, the MAU receive terminals on the ground may want to demodulate each target MAU signal by themselves rather than receiving their targeted signal via a common receive ground terminal. However, they can be located at different directions of arrival angles and at different distances. Fig. 3 demonstrates this case when the receiver ground terminals are placed at different locations. Assume that the transmitter in the satellite transponder knows each MAU location, i.e., each MAU direction of arrival angles. Therefore, the satellite transmitter can employ individual BF for each individual MAU digital channelized down-sample stream with N_t number of PAA elements using the known arrival angle information. In addition, we can assume that each MAU receiver is synchronized in time using its pilot symbol sequence so that it can pick up each MAU's desirable down-sampled polyphase sequence. Other MAU signal streams will be received at a desirable target MAU receiver with a much weaker SNR than the desirable target MAU stream, due to the PAA BF. We consider N_t total number of transmit antennas. In Fig. 3a, the number of antennas per user is denoted by N_u and can be calculated as

$$N_u = \frac{N_t}{M}. \tag{11}$$

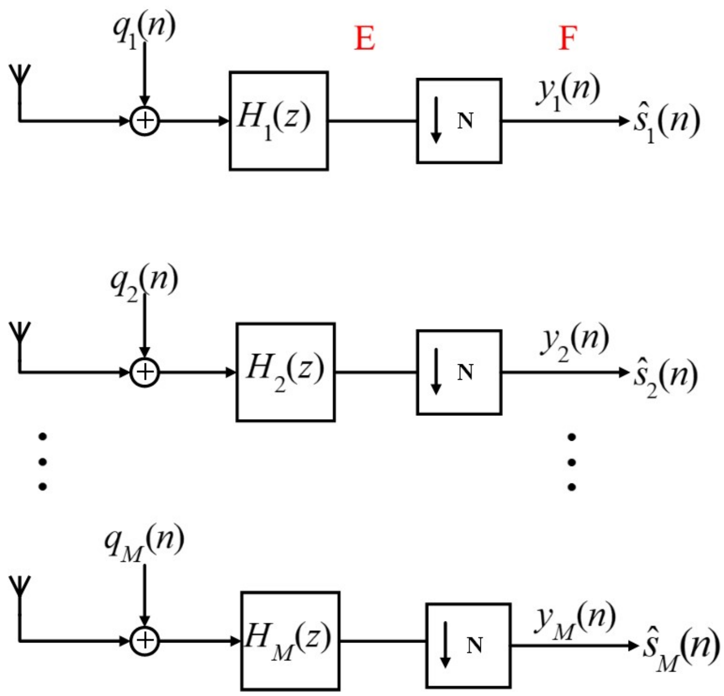
As explained in the previous section II-A, the synthesis filter bank at the transmitter combines a set of signals $\boldsymbol{u_k}$ into a single sample stream sequence signal $x(n)$ [11] and works as a polyphase filter bank. Thus, after multiplexing M user signals, the sequence $x(n)$ represents

$$\{x(n),\cdots,x(n+M-1)\} = \{u_1(n),\cdots,u_M(n)\}. \tag{12}$$



(a) TX: MAU digital channelizer, digital PD, digital HPA, and PAA BF for individual MAU

Fig. 3: (a) Transmitter and (b) receiver for distributed MAU with digital channelizer, digital PD, digital HPA, and individual PAA BF



(b) RX: distributed MAU individual ground terminal receiver

Fig. 3 (cont.): (a) Transmitter and (b) receiver for distributed MAU with digital channelizer, digital PD, digital HPA, and individual PAA BF

Also, $x_{PD}(n)$ denotes the input to the HPA when PD is applied. The output of the HPA is denoted by $v(n)$ and represents

$$\{v(n),\cdots,v(n+M-1)\}=\{v_1(n),\cdots,v_M(n)\}. \tag{13}$$

The component of the PAA BF weight vector with N_u number of antenna elements/user is denoted by $w_{kl}(n)$, where subscript k denotes user index $k=1,\cdots,M$ and subscript l denotes antenna element index $l=1,\cdots,N_u$. We consider a linear one-dimensional PAA BF. This is because a geo-satellite communication link has a very small angle spread, e.g., less than 0.2 degrees, typically due to a long waveform traveling distance, and it has not been verified yet whether a satellite communication link can have a rank larger than one or not. This is simple to extend for other BF arrays such as a parabolic, circular array, patch, or MIMO array. Thus, the PAA BF weight vector component $w_{kl}(n)$ can be expressed as

$$w_{kl}(n)=e^{j\frac{2\pi}{\lambda}(l-1)d\left(\phi_k(n),\theta_k(n)\right)}, \tag{14}$$

where λ is the wavelength, and $d\left(\phi_k(n),\theta_k(n)\right)=d[\sin\left(\theta_k(n)\right)\cos\left(\phi_k(n)\right)$ $+\sin\left(\theta_k(n)\right)\sin\left(\phi_k(n)\right)+\cos\left(\theta_k(n)\right)]$ is the effective distance between the adjacent linear array antenna elements for a given pair of azimuth and elevation angles $\left(\phi_k(n),\theta_k(n)\right)$ in the user k direction. The adjacent antenna distance d is chosen as

$$d = \frac{\lambda}{2}. \tag{15}$$

Note that $w_{kl}(n)$ can be time-varying. We assume a pair of constant azimuth and elevation angles (ϕ_k, θ_k) and a constant w_{kl} during a block of data in simulation. As shown previously in Fig. 3b, we consider a single receiver antenna per user. The received signal $y_k(n)$ can be written as

$$y_k(n) = s_k(n)\left[\sum_{l=1}^{N_u} e^{-j\frac{2\pi}{\lambda}(l-1)d(\phi_k(n)\theta_k(n))w_{kl}(n)}\right] + q_k(n) \tag{16}$$

for $k = 1, \cdots, M$ users, where the phase in $e^{-j\frac{2\pi}{\lambda}(l-1)d(\phi_k(n)\theta_k(n))}$ is due to the phase delay difference among the linear 1D PAA elements at the transmitter, and $q_k(n)$ denotes the identically and independently distributed (i.i.d.) AWGN noise with probability distribution $N(0, N_0/2)$ at each individual receiver, $k = 1, \cdots, M$.

6.0 RESULTS AND DISCUSSION

Throughout our simulations, n_{total} denotes the total number of samples in a data block, f_s denotes the sampling frequency, and T_s denotes the sampling time. Then the sub-carrier frequency is normalized by $f_s = 1/T_s$ for user k, i.e., the k-th polyphase channel frequency out of M polyphase channels normalized by the sampling rate f_s and can be written as

$$f_k = \frac{k}{N} f_s T_s = \frac{k}{N}. \tag{17}$$

In order to prevent distortion from the discrete Fourier transform (DFT) and to ensure that each channel can transmit simultaneously with equal bandwidth at the normalized sub-carrier frequency f_k, we choose the normalized symbol rate $R_s < f_k$. The true un-normalized sub-carrier frequency would be $f_k \times f_s = (k/N) \times f_s$. For our simulation, the symbol rate is chosen as

$$R_s = \frac{f_1}{10} \tag{18}$$

for all users. Since f_1 is 1/5 from (17), and $R_s = 1/50$, i.e., the number of samples per symbol is 50. One way of choosing a reasonable R_s is to use the greatest common divider (GCD) out of a set of f_k which is equal to $f_1/10$, for $k = 1, \cdots, M$ users. If another symbol rate is chosen, then all users do not have an equal magnitude frequency spectrum. Since the GCD is $f_1/10$, we used this in (18) for each user. Also, in order to achieve a signal spectrum with equal magnitude for all users at each step A, B, C, D, E, and F in Fig. 2, we need to ensure that n_{total}/N is an integer. This is because each DFT exponential waveform should have an integer number of cycles per data block time unit, i.e., the size of the DFT when it is used to find the frequency spectrum of the signal by a linear combination of DFT exponential wave forms [12].

We limited our simulations for up to 10^5 symbols and 5×10^6 samples. We observed that for any n_{total} number of samples that makes n_{total}/N an integer, each DFT waveform has an integer number of cycles per data block time unit. Thus, we can achieve a signal spectrum with equal amplitude at each step shown in Fig. 2. The sampling frequency f_s for our simulation is taken as 2 MHz.

We have divided our simulation results into three subsections. In subsection A, we check the digital channelizer for our proposed system model in the discrete time-domain at each step A, B, C, D, E and F, as shown in Figs. 2 and 3. The only difference between the system models in Figs. 2 and 3 is whether the PAA BF vectors are applied or not at the transmitter side. The digital channelizer performs in the same way for both models in Figs. 2 and 3. In subsection B, we check our results for the multiuser digital channelizer model in Fig. 2 in the frequency domain by plotting the frequency spectrum at each step A, B, C, D, E, and F. Also, we check the BER plot to see if the PAA BF reduces the BER for the multiuser digital channelizer, digital PD, and digital HPA. In subsection C, we check the performance of our digital HPA and digital PD separately, according to the block diagrams in Figs. 2 and 3. Our simulation results clearly show that the proposed digital PD can compensate for the phase distortion due to the digital HPA, while the desirable HPA amplitude can be preserved almost completely.

6.01 Discrete Time Domain Plots for Multiuser Digital Channelizer, Digital HPA, and Proposed Digital PD but No BF

To check the discrete time domain outputs for our proposed multiuser digital channelizer, digital HPA, and digital proposed PD, 50 discrete-time sampling points are plotted in Figs. 4–10 for demonstration purposes only. For our MATLAB simulation, we used $n_{total} = 200$ samples and 4 symbols, so that we have 50 samples per symbol. As shown in Fig. 4, we consider $M = 3$ FDM sub-channels, for $k = 1, \cdots, M$ users. The symbol streams $s_k(n)$ are generated from 8-PSK modulated signals. For simulation, the multiplexing is performed at the beginning stage for convenience instead the last moment of the transmitter without loss of generality. In Fig. 4, we plot the discrete time symbol streams $s_k(n)$ for $k = 1, \cdots, M$ users at step A in Figs. 2 and 3. As clearly shown in Fig. 4, each user signal is separated from the previous user signal with a unit sample time delay.

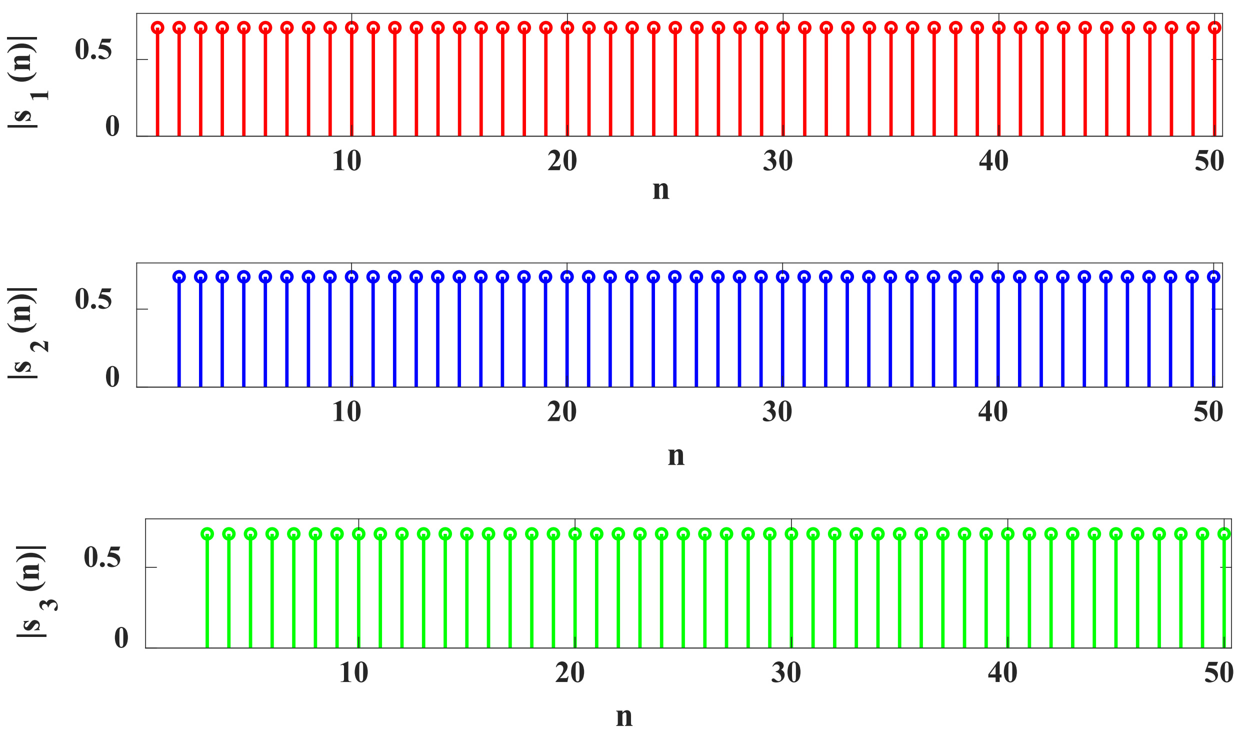


Fig. 4: Discrete time domain representation at step A in Figs. 2 and 3.

Next, at step B in Figs. 2 and 3, the baseband signals from each user are up-sampled by factor $N = 5$. The up-sampled $s_k(n)$ can be seen as the components of a time-multiplexed signal $s(n)$ in Fig. 5. Each MAU transmitted signal samples are not overlapped in the time domain with any other MAU samples. For example, in Fig. 5 the up-sampled signal for user 1 carries information at sampling points 1, 5, 10, $\cdots$ and inserts zeros at the rest of sampling points, e.g., 2, 3, 4, 6, 7, 8, 9, $\cdots$. The up-sampled signal for user 2 carries information at sampling points 2, 6, 11, $\cdots$ and inserts zeros at the rest of sampling points. Since all MAU signals are transmitted in a sequence order by a common on-board transmitter processor in Fig. 2, we can assume that the MAU transmitted sequence will be received in the same manner at any distributed RX in Fig. 3. Therefore, each MAU receiver can take samples at the right sampling time when its information is transmitted. Here it is assumed that time synchronization is achieved with a known pilot symbol sequence.

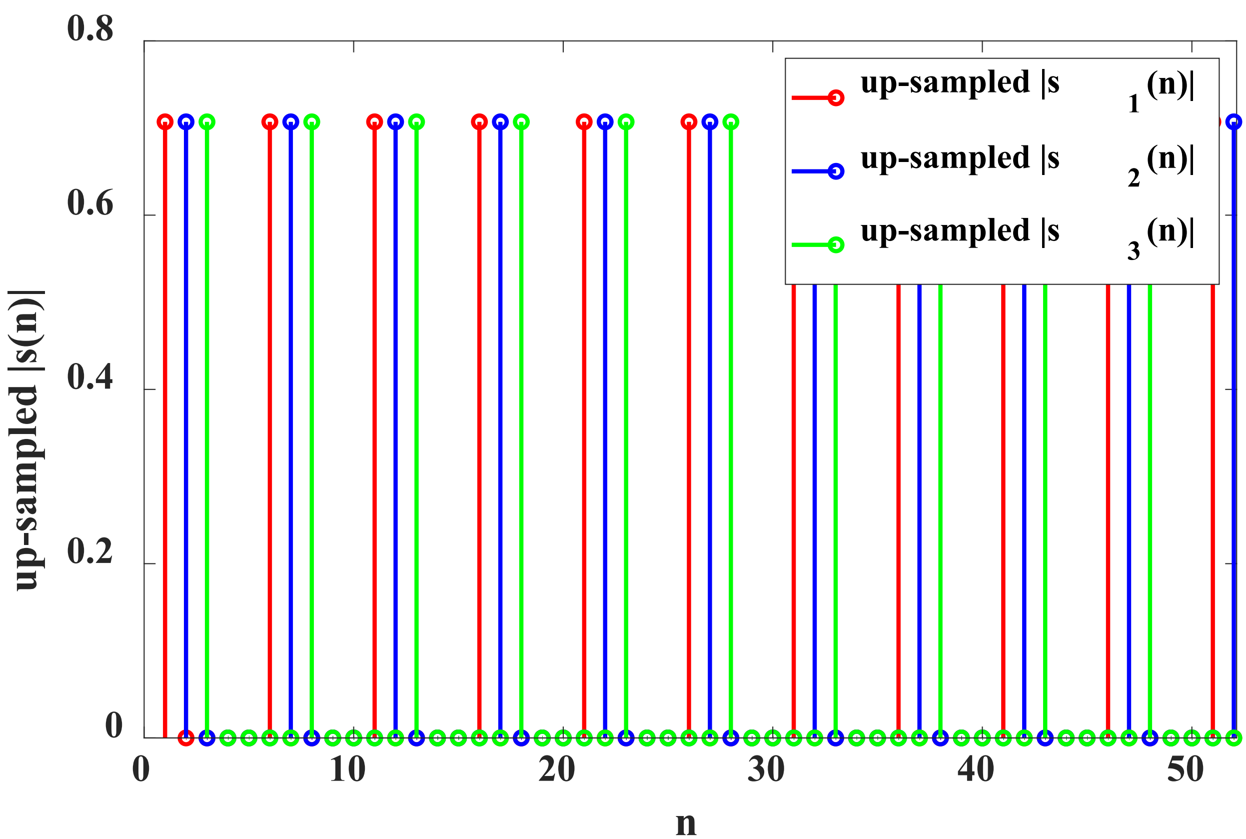


Fig. 5: Discrete time domain representation at step B in Figs. 2 and 3

Each MAU time-multiplexed signal component is passed through a synthesis filter bank consisting of M number of BPFs with z-transfer function $\boldsymbol{F}_k(z)$: one for each user sub-band signal, $k = 1, \cdots, M$. The transmit BPFs of order 39 are used for our simulation. Fig. 6 shows the impulse response of $\boldsymbol{F}_k(z)$ generated by the BPF filters, which are delayed by unit sample time for the two adjacent MAUs.

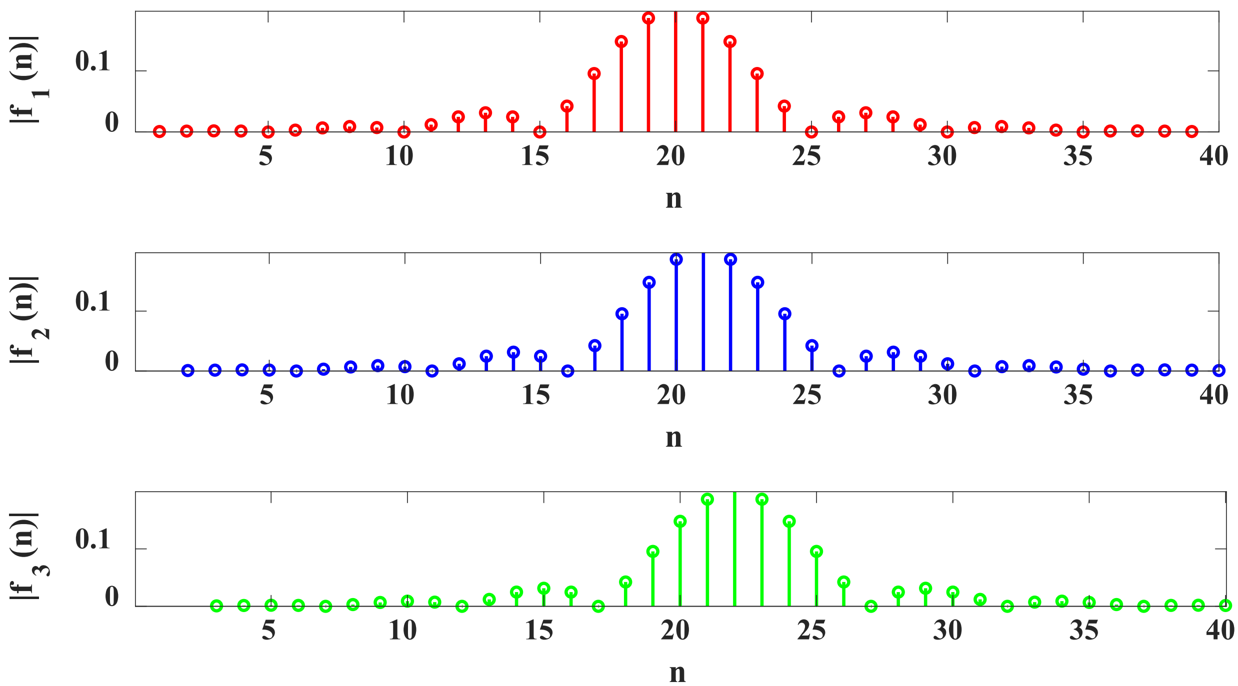


Fig. 6: Discrete time impulse response of transmit filters $F_k(z)$ in Figs. 2 and 3

Fig. 7 shows the output sequence from each transmitting filter $F_k(z)$ in Figs. 2 and 3 when the up-sampled signal $s_k(n)$ is fed into the BPF at step C. The filter takes a convolution of each input sample sequence of $s_k(n)$ with its impulse response $F_k(z)$, as explained in [11].

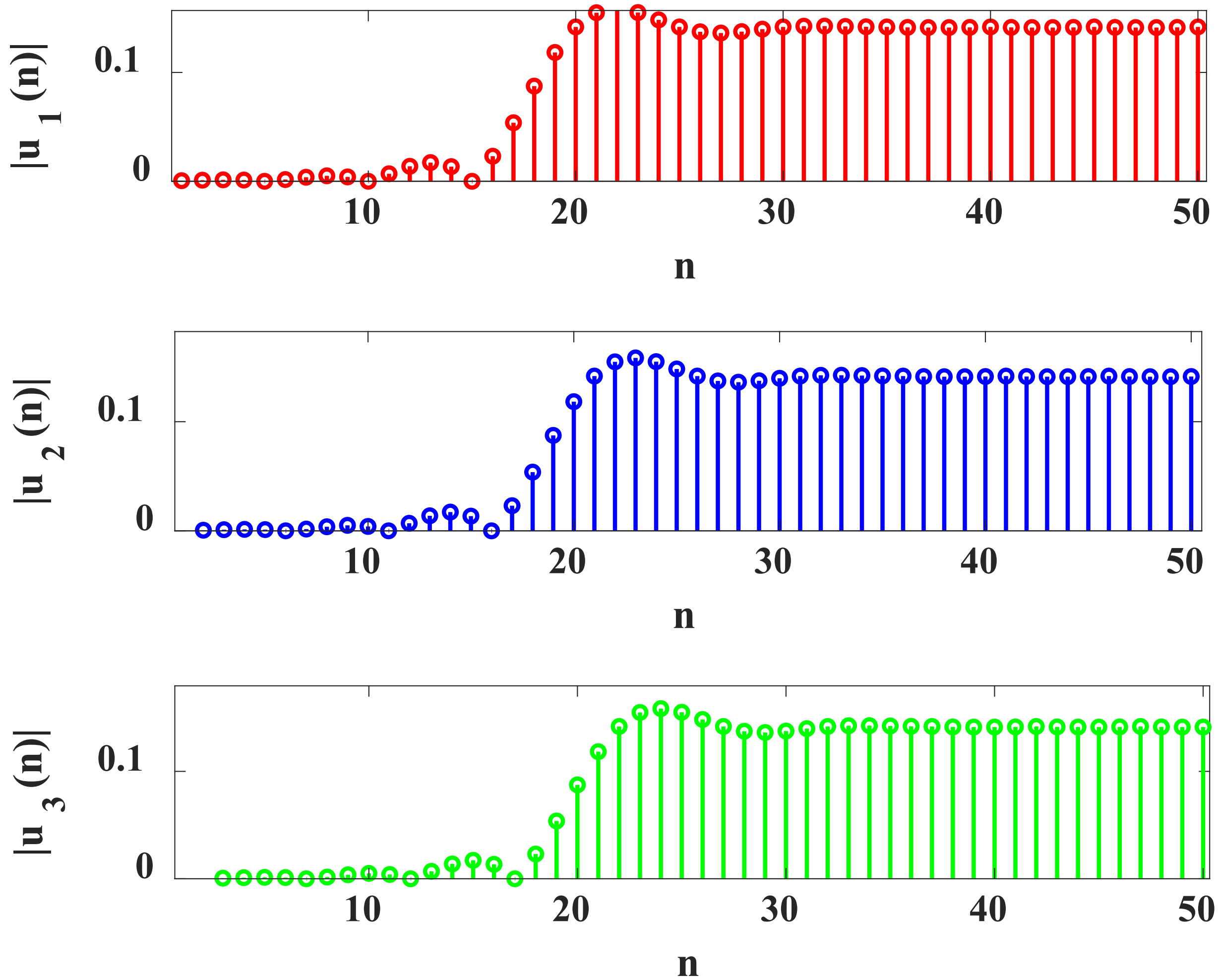


Fig. 7: Discrete time domain representation at step C in Figs. 2 and 3

Fig. 8 shows the magnitude of the multiplexed signal sample sequence $x(n)$ to be transmitted. This $x(n)$, which is the output of the transmitting filter bank, is sent over the digital PD and to the digital HPA before the D/A and main carrier up conversion, which are not shown in Figs. 2 and 3. The output of the digital HPA is passed through the AWGN channel with additive noise, $q(n)$.

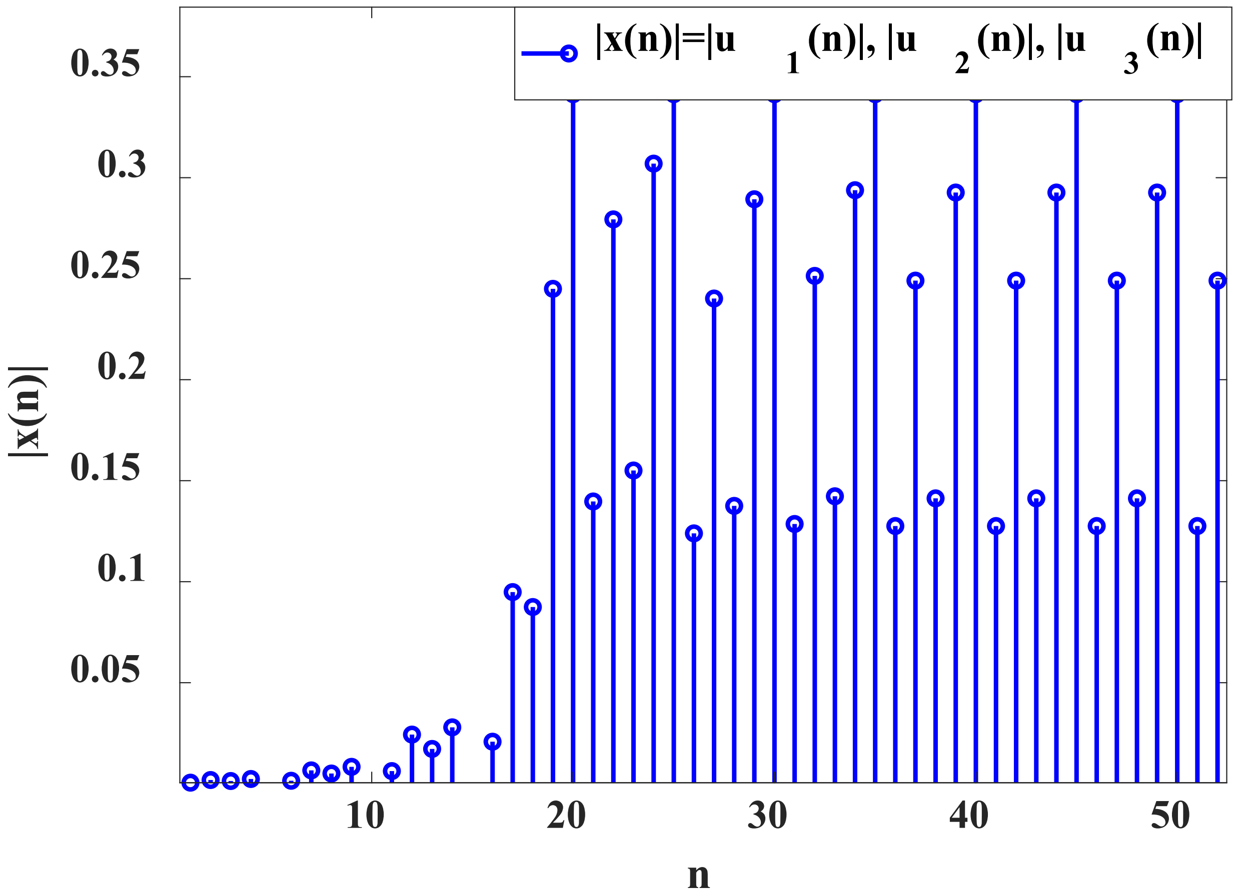


Fig. 8: Discrete time domain representation at step D in Figs. 2 and 3

The output $x(n)$ represents a single MAU signal $u_k(n)$, where $k = n\,\mathrm{mod}(M)$ for $k = 1, \cdots, M$. HPA $v(n)$ is the transmitted signal $u_k(n)$, where $k = n\,\mathrm{mod}(M)$ for $k = 1, \cdots, M$. At the receiver end, the analysis BPF filter z-transfer function, denoted by $\boldsymbol{H}_k(z)$, for $k = 1, \cdots, M$ receiver channels [11], has the task of passing only the k-th sub-band signal, and then a down-sampler with a factor of $N = 5$ removes the zeros inserted by the up-sampler in the transmitter. Each of the filters in the analysis filter bank is designed using MATLAB Remez BPFs on the order of 39, which behaves almost in the same way as the polyphase filter in [2] of a much higher filter order, such as 200. The discrete time domain representation at step E of Figs. 2 and 3 is shown in Fig. 9. The AWGN is not included in Fig. 9 because the focus here is observing the HPA distortion.

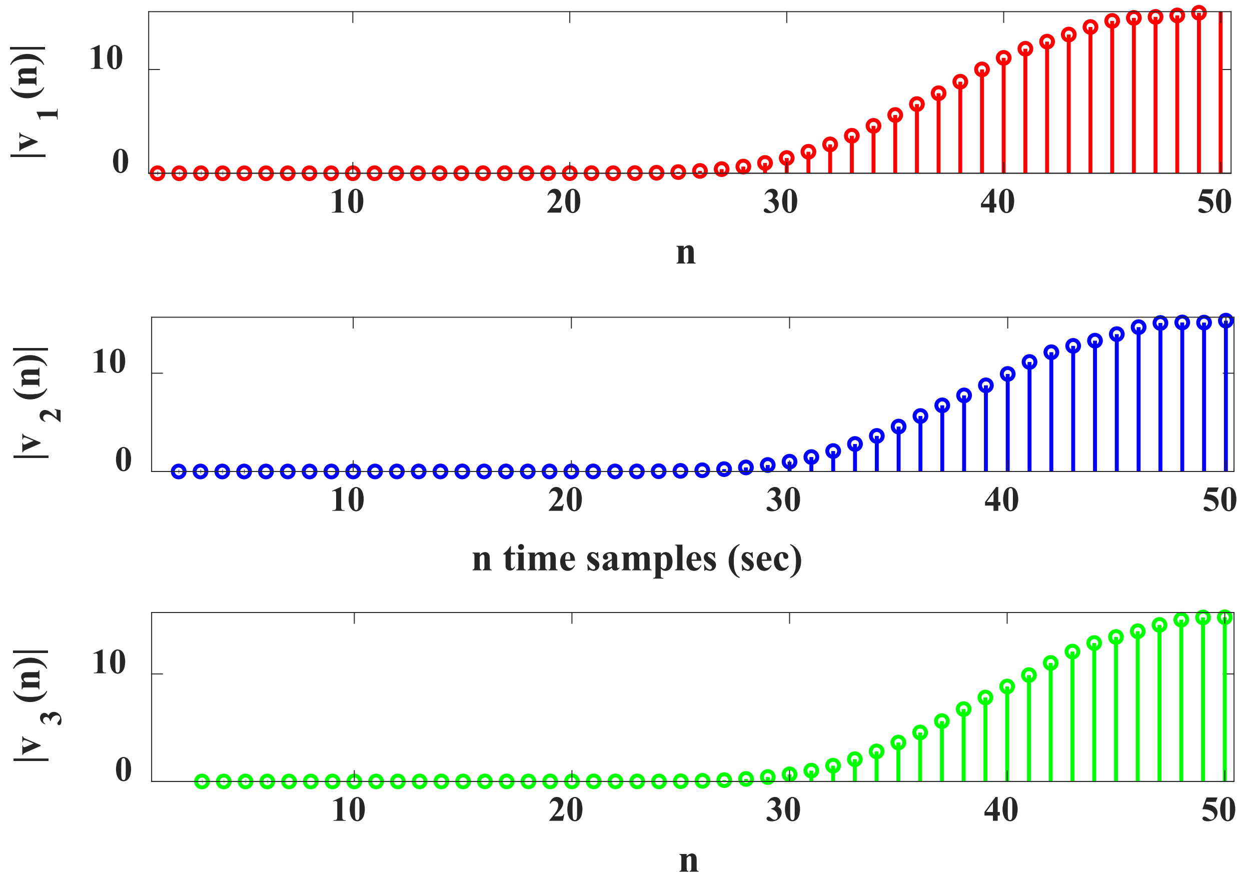


Fig. 9: Discrete time domain representation at step E in Figs. 2 and 3

Finally the transmitted symbols $s_k(n)$ are demodulated from the received signal $\hat{s}_k(n)$, which, in general, are distorted versions of the symbols $s_k(n)$ because of the combined effects of channel, noise, HPA, and filters. In our simulation, we use a simple pre-compensation to remove the deterministic distortions caused by filters, and the proposed PD successfully removes distortion due to the nonlinear HPA. To demonstrate this again we did not include AWGN in Fig. 10. It can be seen that the transmitted signal sample in Fig. 4 can be retrieved almost exactly at the receiver, as shown in Fig. 10. The discrete time domain representation at step F in Figs. 2 and 3 is shown in Fig. 10.

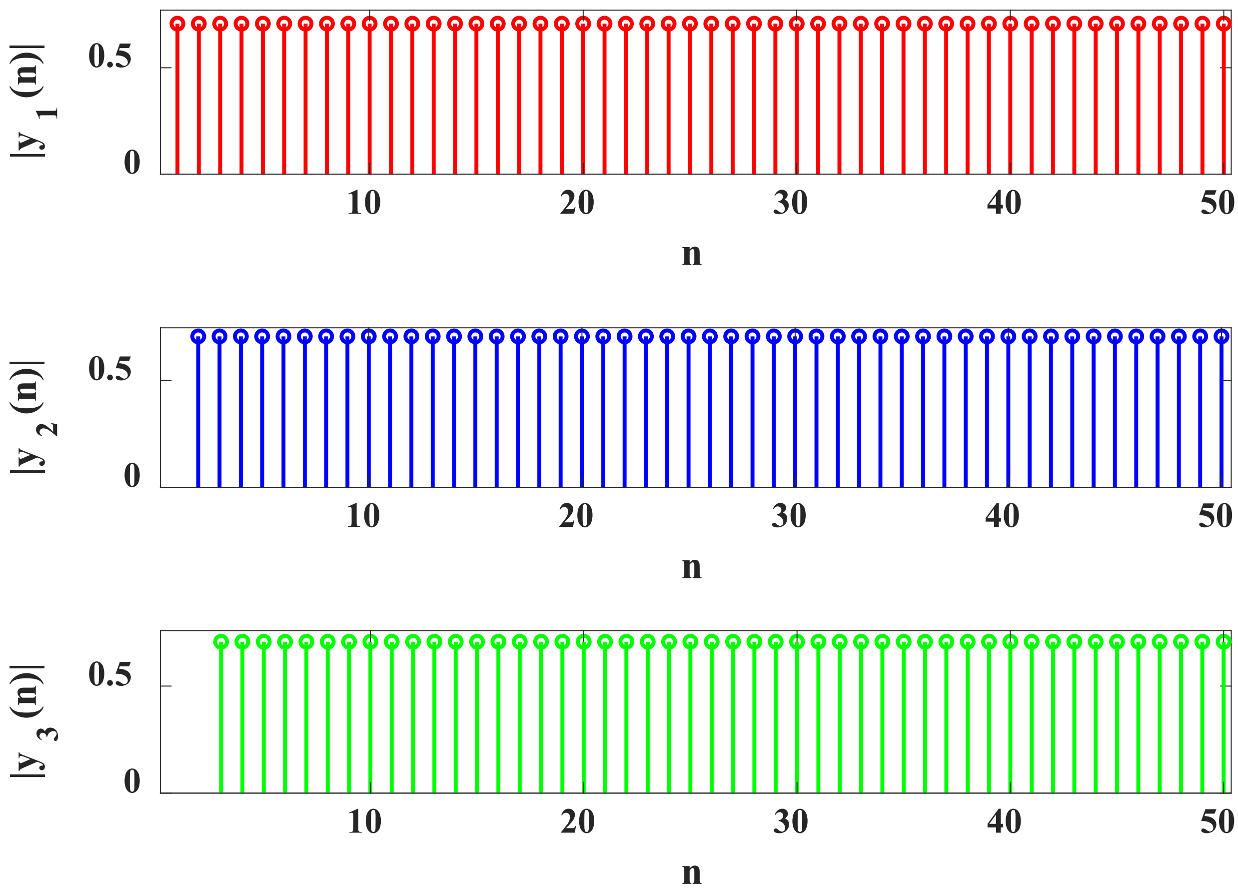


Fig. 10: Discrete time domain representation at step F in Figs. 2 and 3

6.02 Frequency Spectrum of Multiuser Digital Channelizer with HPA and Proposed PD

Similar to the time-domain representation in Figs. 4–10, the frequency domain representation for our proposed multiuser digital channelizer with HPA and PD is presented in this subsection. We used $n_{total} = 200$ samples and 4 symbols in Figs. 11–16, so that we have 50 samples per symbol.

Fig. 11 shows the 8-PSK modulated input signal spectrum for three MAUs at step A in Figs. 2 and 3. As shown in Fig. 11, the initial baseband signals with no sub-carrier have the same spectrum centered at 0 Hz for all three users because the $M = 3$ polyphase channels are not yet applied at this step. Next, at step B, the baseband signals from each user are up-sampled by factor $N = 5$. In our simulation, $N > M$ and we take $N = 5$ and $M = 3$. This means that up to $N = 5$ number of sub-band user signals can be transmitted, but only $M = 3$ number of sub-band user signals are transmitted to reduce the spectrum overlap.

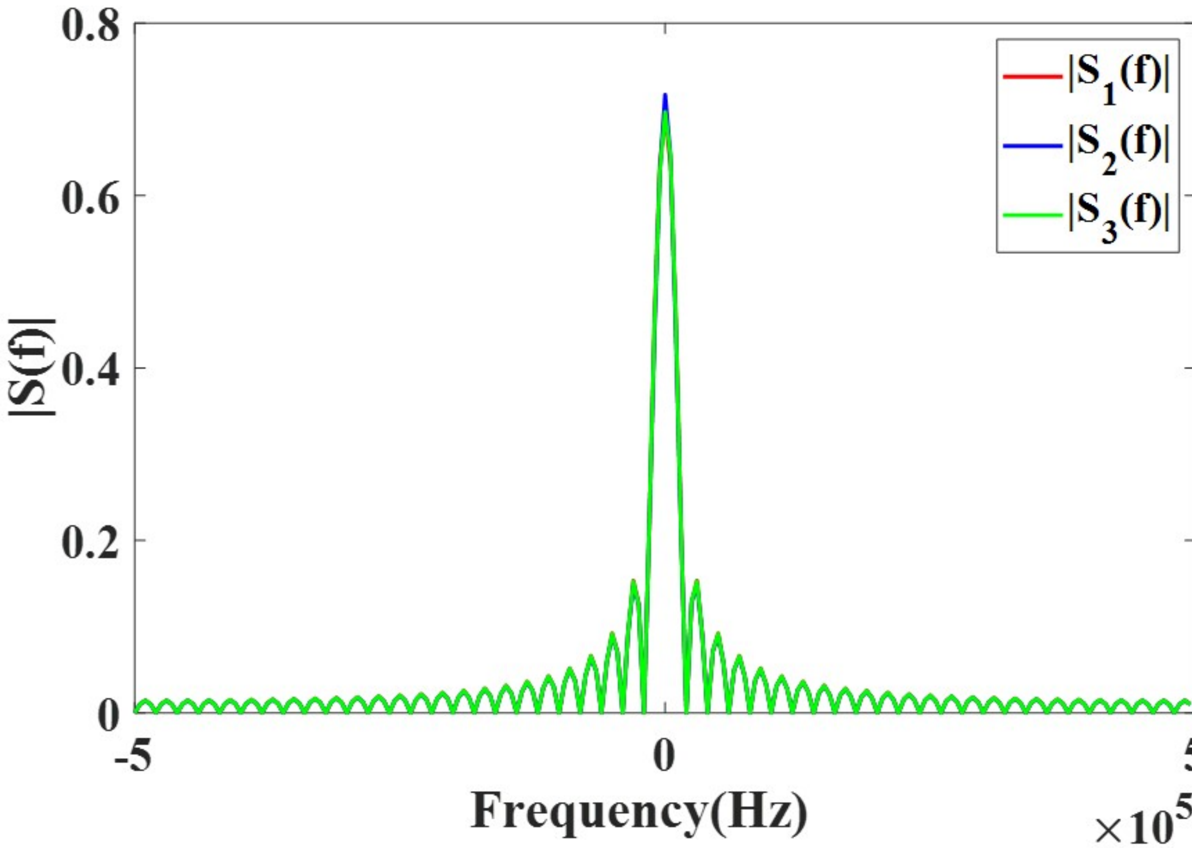


Fig. 11: Frequency spectrum of 8-PSK modulated signal at step A in Figs. 2 and 3

Fig. 12 shows the spectrum at the output of the up-sampler at step B which contains $N = 5$ number of input signal spectrum images. After up-sampling, a synthesis filter bank is used to combine the set of up-sampled signals from each user into a single-sample stream signal . Each of these filters in the filter bank is chosen to be a BPF and designed using the MATLAB with a filter order 39 and having central frequency f_k, as obtained from (16). The subscript k denotes the user index.

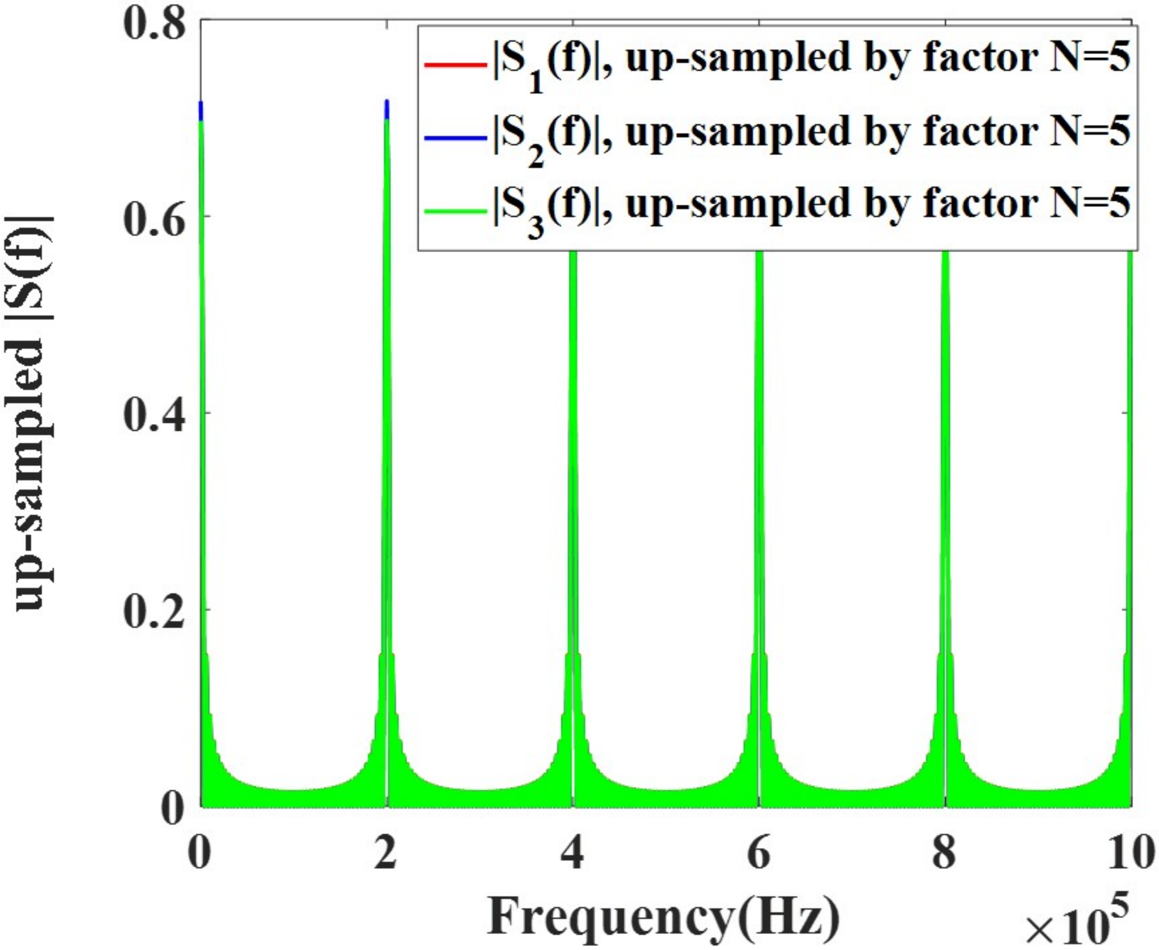


Fig. 12: Frequency spectrum at step B in Figs. 2 and 3

Fig. 13 shows the output spectrum of each of these synthesis filters at step C in Figs. 2 and 3. Compare the clear spectrum in Fig. 13 (generated by the proposed algorithm) with the unclear spectrum caused by the analog HPA in Fig. 1.

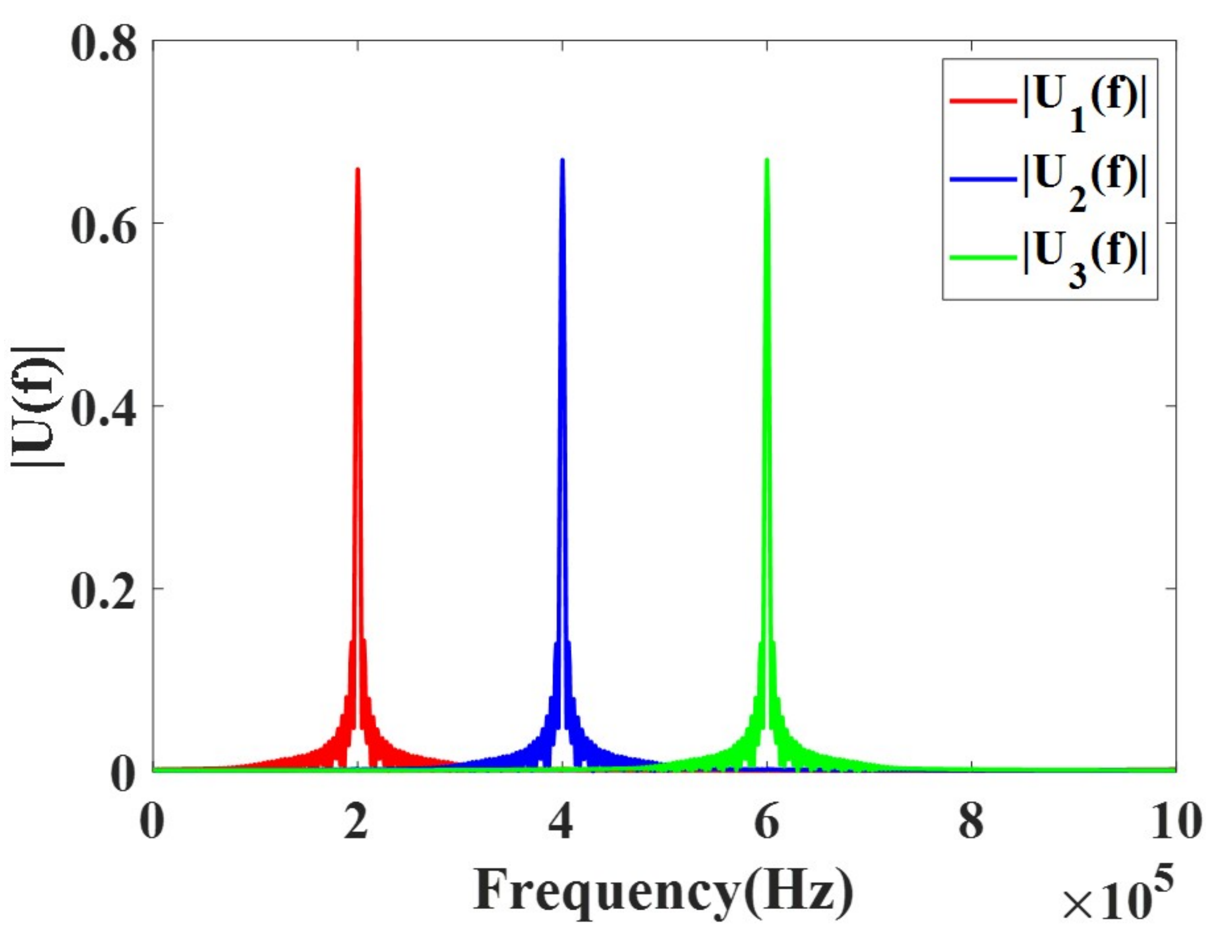


Fig. 13: Frequency spectrum at step C in Figs. 2 and 3

Fig. 14 shows that the synthesis filter combines the up-sampled and filtered signals $\boldsymbol{u}_k$ into one single stream signal $x(n)$, and displays the spectrum at step D in Figs. 2 and 3. Due to distortion in phase and amplitude caused by the synthesis filter bank at the transmitter and due to the analysis filter bank at the receiver, a simple phase and amplitude compensation for the deterministic filter distortions were included. The combined, i.e., polyphased, signal $x(n)$ is passed through the PD and HPA, and then transmitted through the AWGN channel. The receiver side has a set of analysis filter banks that demultiplex the predistorted output signal from the HPA output $\boldsymbol{v}$ into M components for each user k receiver processing using the M polyphase channels.

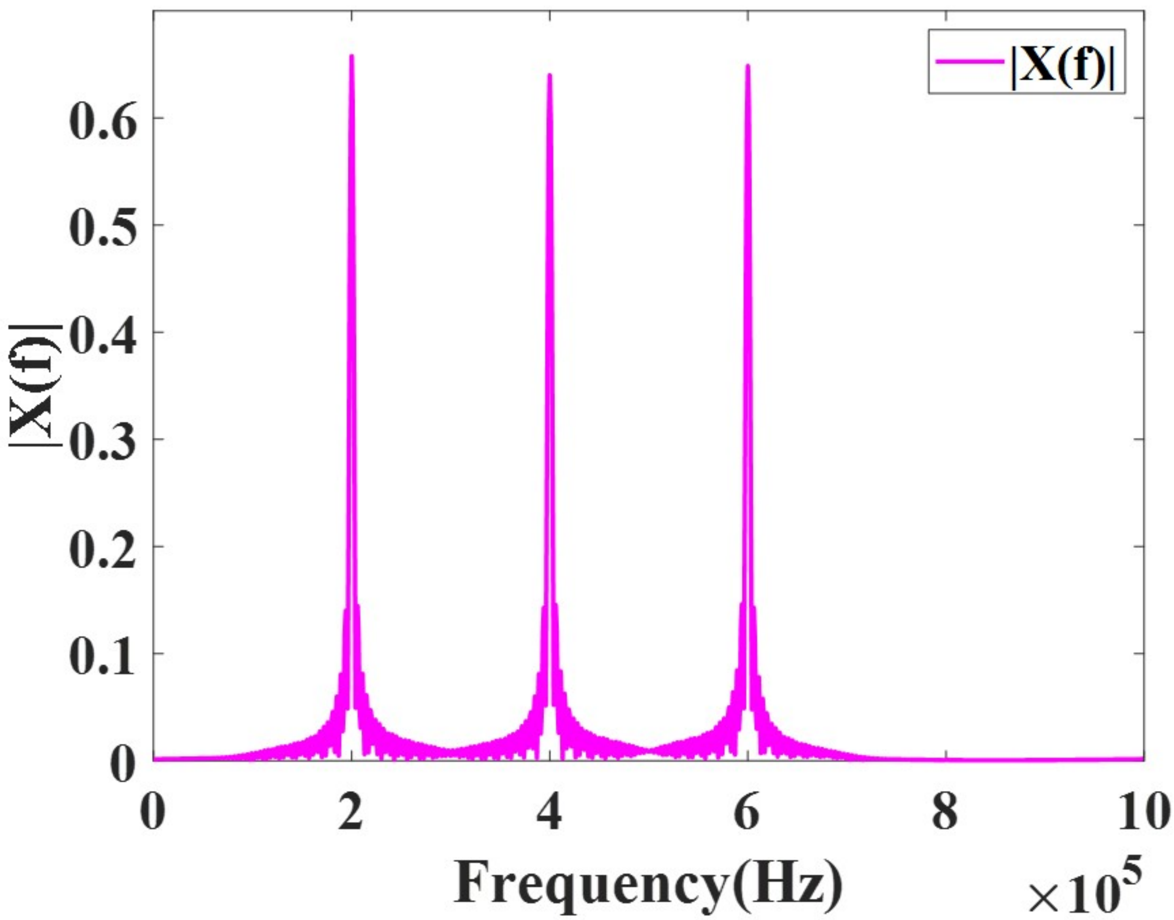


Fig. 14: Frequency spectrum at step D in Figs. 2 and 3

Fig. 15 shows the spectrum at the output of these analysis filters at step E in Figs. 2 and 3. Each of the filters in the analysis filter bank is designed using MATLAB Remez BPFs of order 39, which behaves closely to the polyphase filter design in [2], which employs a much higher filter order, such as 200.

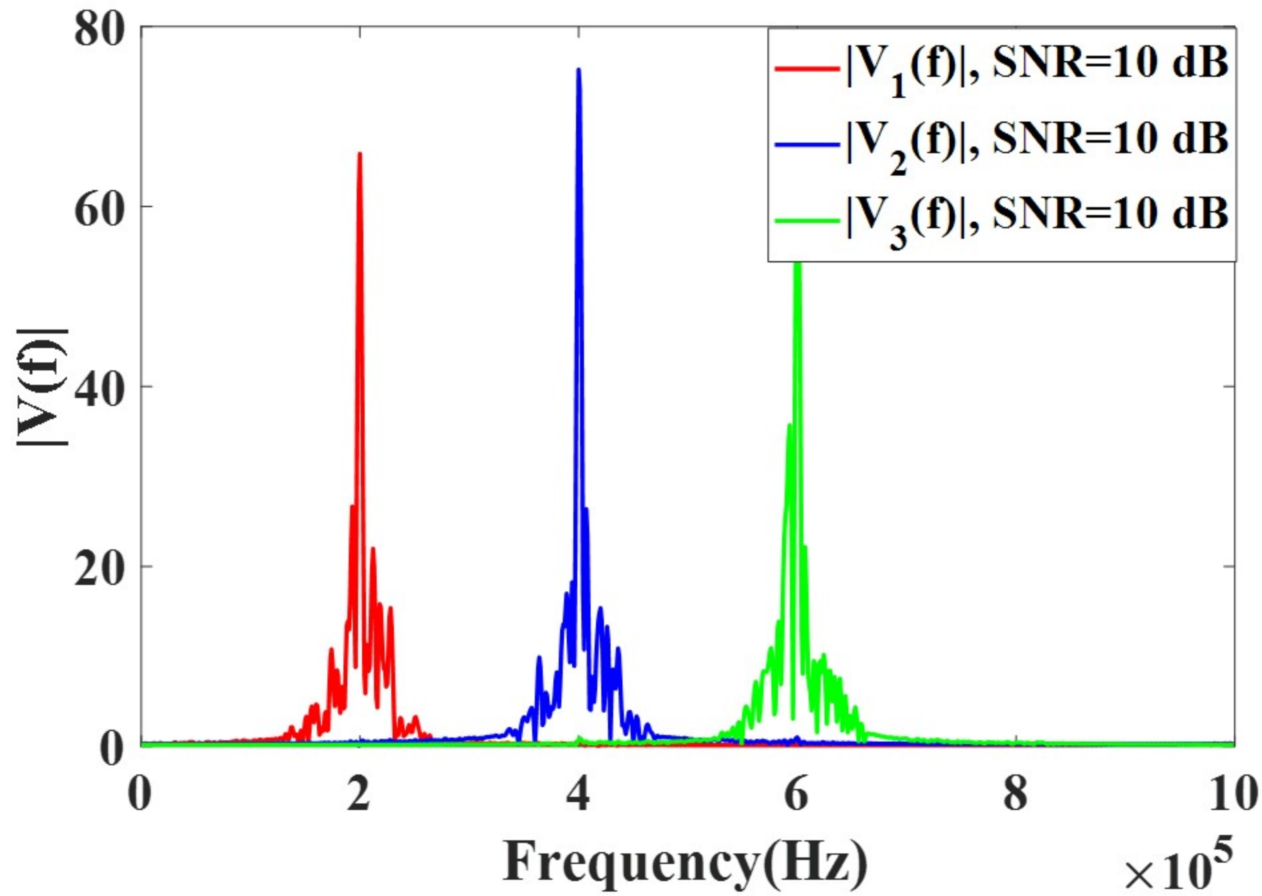


Fig. 15: Frequency spectrum at step E in Figs. 2 and 3 under AWGN noise, for SNR = 10 dB

Finally, Fig. 16 shows the spectrum of the down-sampled signals at step F in Fig. 2 and Fig. 3. These down-sampled signals are demodulated to obtain the received symbol vectors $\hat{s}_k$.

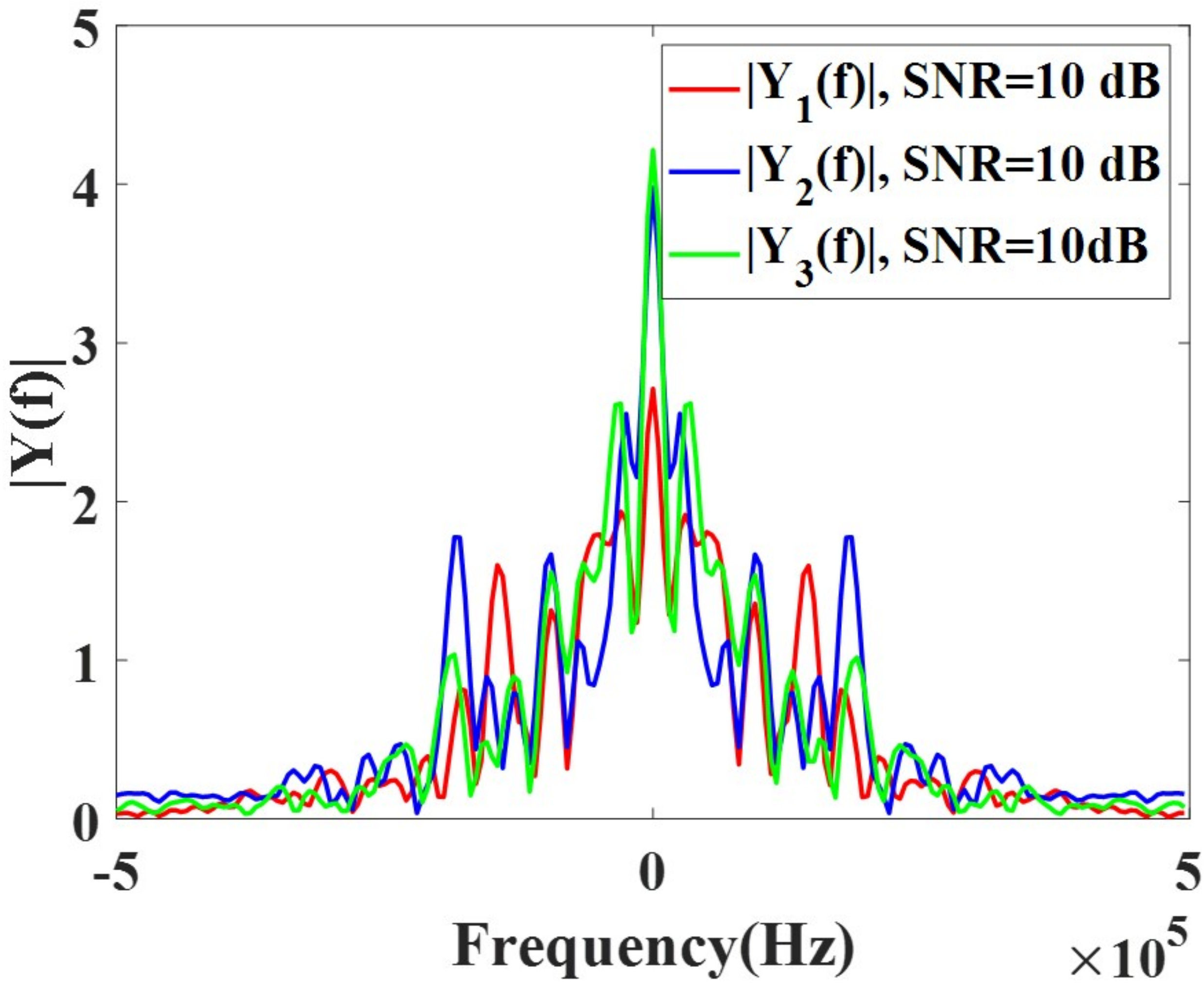


Fig. 16: Frequency spectrum at step F in Figs. 2 and 3 under AWGN noise, for SNR = 10 dB

Fig. 17 shows the bit error rate by comparing the transmitted and demodulated symbols at the receiver for 10^5 number of symbols, i.e., $n_{total} = 5\times10^6$ samples. Since we use $m = 50$ samples per symbol, the signal-to-noise-ratio (SNR) is *m* times higher than the SNR of the one-sample-per-symbol case, if each noise sample in the multi-sample per symbol has the same variance as the one-sample-per-symbol case. Therefore, we need to increase each noise sample magnitude by multiplying it with $\sqrt{m}$. This is because the variance of the sample average noise with *m* samples per symbol becomes *m* times smaller as $\sigma_N{}^2 = N_0/m$, where N_0 denotes the one-sided AWGN power spectral density, which is equal to the noise variance of the one-sample-per-symbol case. In order to obtain the same BER results as that of the theory under an ideal AWGN channel, the noise variance of the one-sample-per-symbol case must be multiplied by factor *m* for the multiple samples per symbol case. As shown in Fig. 17, the simulated BER curve under the AWGN channel for each user agrees almost perfectly with that of the theoretical BER under AWGN. Thus, we can state, from our simulation results and theoretical analysis, that the proposed PD method removes the nonlinear impairments due to HPA almost perfectly, and degradation by using the digital channelizer is negligible.

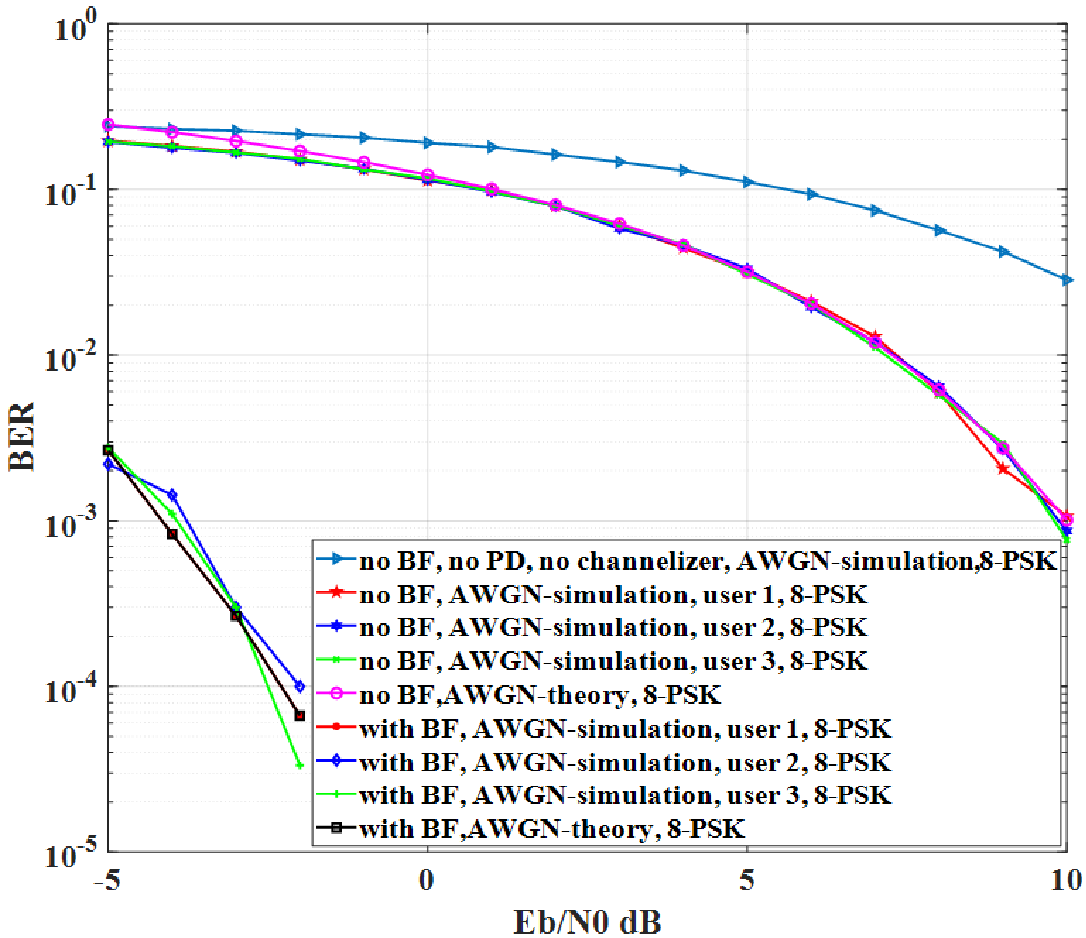


Fig. 17: BER under AWGN channel, 8-PSK modulation for proposed system of digital channelizer, digital PD, and digital HPA

6.03 HPA and Proposed PD Demonstration

As shown previously in Figs. 2 and 3, the digital predistorter restores the phase distortion due to the digital HPA. To demonstrate this, we used 200 samples, 4 symbols, and hence 50 samples per symbol in Figs. 18 and 19. Fig. 18 shows the phase snapshot plots for the input signal $x(n)$ to the digital HPA, the distorted phase at the output of the digital HPA when PD is included and not included, and finally the restored phase through the proposed digital PD, which exactly matches with the phases of the no-HPA phase distortion case.

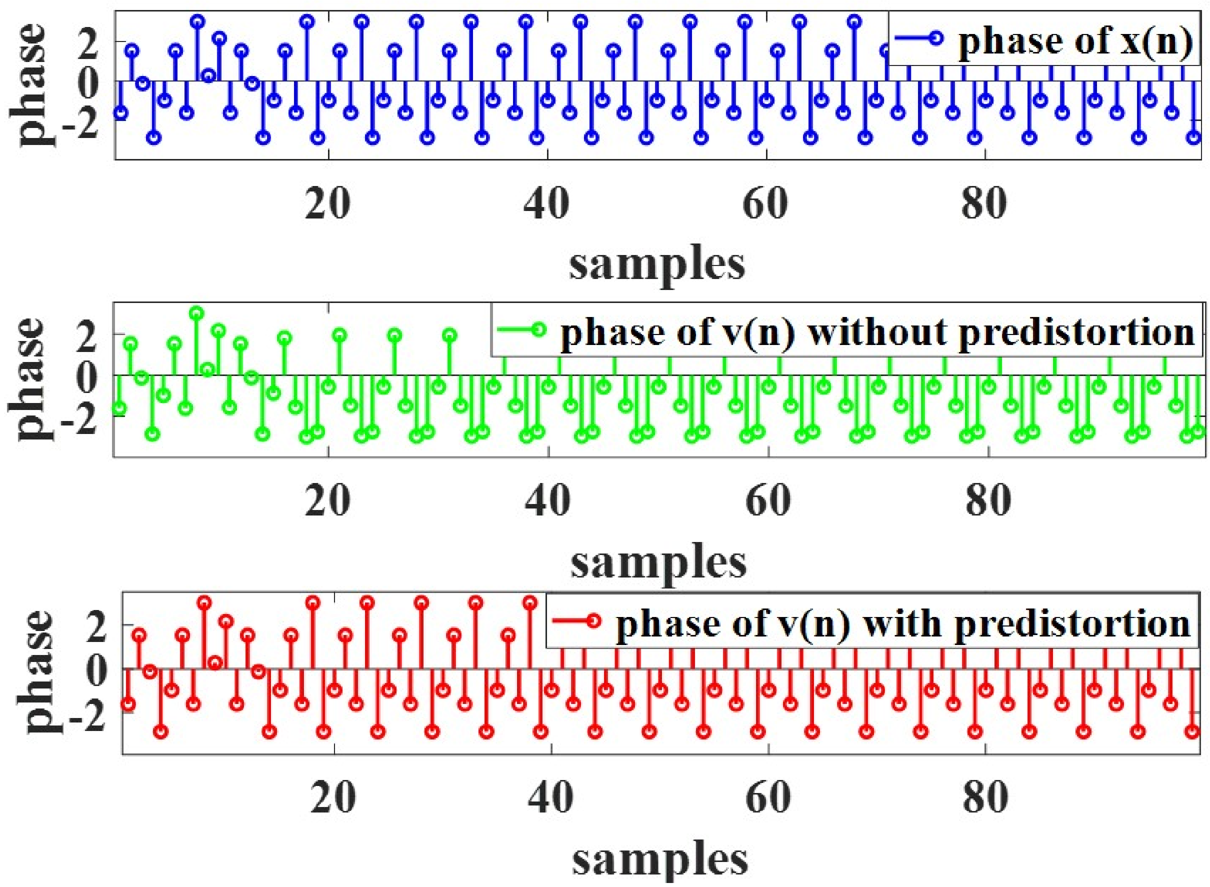


Fig. 18: Phase distortion due to HPA and restoration through proposed phase pre-distortion method

Similarly, it can be observed that the proposed PD restores almost perfectly the distorted amplitude at the HPA output to the amplitude of the input signal amplitude $x(n)$ times c. The only phase restoration has been included in this report. The slope c between point (h_1, g_1) and the origin can be found using HPA's AM/AM characteristics shown in Fig. 19 as

$$c = \frac{f_a\left(|x(n)|\right)}{|x(n)|}. \tag{19}$$

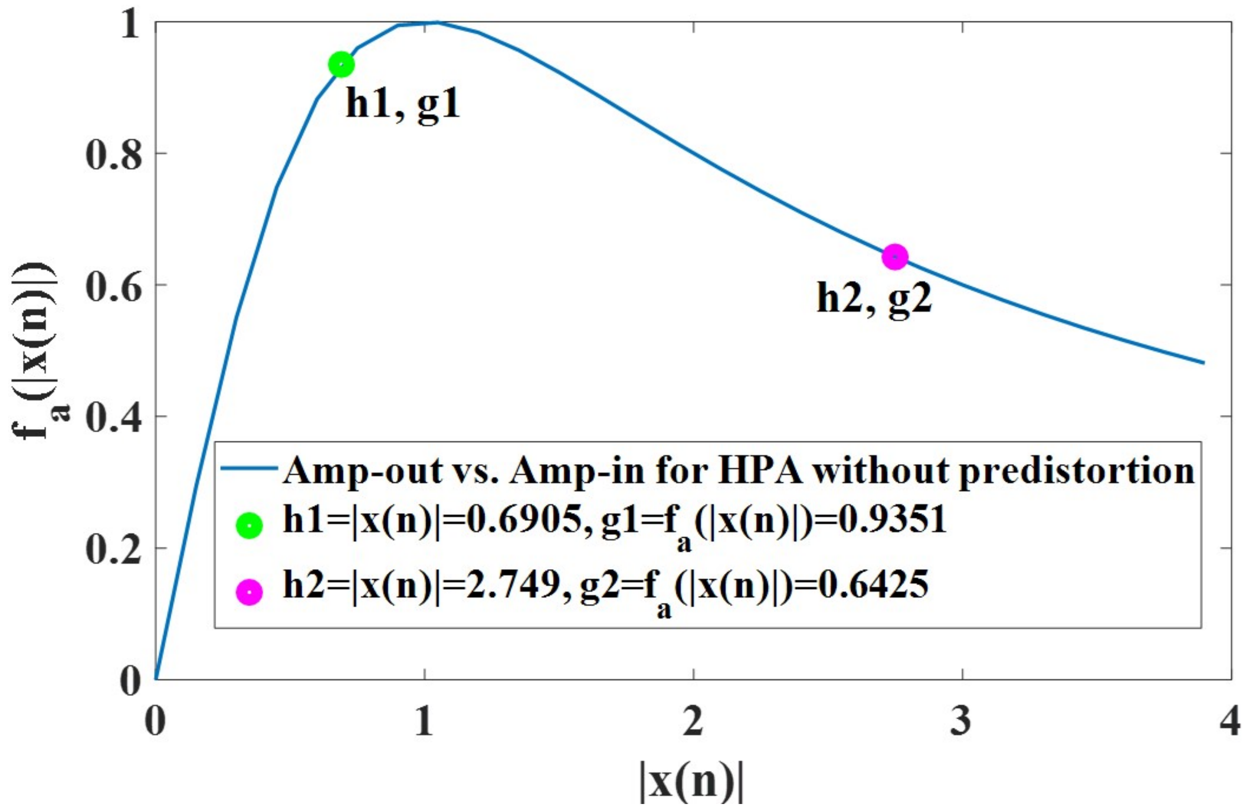


Fig. 19: Output amplitude vs input amplitude for Saleh HPA model with no predistortion

From Fig. 19, the HPA back-off, which is the output power difference, calculated as $\left(1-g_1^{\,2}\right)$ between the saturation point and a test point $\left(h_1, g_1\right)$, is only 0.5829 dB, i.e., the HPA back-off is small. Therefore, the system is under a nonlinear condition. Observe in Fig. 19 that the test point $\left(h_1, g_1\right)$ is before the saturation point $\left(1,1\right)$. Even if the input amplitude is larger than the saturation point, we can also verify that the proposed PD works almost perfectly. For example, we considered a test point at $\left(h_2, g_2\right)$= (2.749, 0.6425) with $h_2 = 2.749$, which is greater than the saturation point input amplitude 1, and we could still observe that the proposed PD restores both amplitude and phase almost perfectly. Both the amplitude and phase restoration snapshot plots have been omitted.

In summary, first, the proposed PD tests whether input amplitude $|x(n)|$ is less than or equal to 1. If yes, then c is computed using (19). Otherwise, the proposed PD sets the output amplitude multiplied to $c = 1$, i.e., no amplification. This is because the HPA reduces the amplitude when $h > 1$. Still the PD applies the PD phase restoration in (10) for this case.

7.0 CONCLUSIONS

Through our theoretical analysis and numerical results, we can conclude that the proposed predistortion method can restore almost perfectly both the amplitude and phase distortion due to the HPA if the HPA is memoryless. Digital beamforming, digital channelizer, and high-power amplifier have been studied extensively for years as can be seen from some of the references cited in this report.

Here we proposed for the first time a combined system of digital channelizer, digital PD, digital memoryless HPA, and PAA BF, and observed that the digital polyphase channelizer can be useful for a future digital satellite design. This is because the digital memoryless HPA takes only one input sample coming from one single sub-band MAU signal sample at a given sample time n; hence, no overlapping happens among the MAU sub-band user samples, and also, the intermodulation products that appeared in the analog HPA can be avoided completely.

In addition, we could compensate for the phase and amplitude distortion due to the transmit and receive filters almost perfectly because these filters and distortions are deterministic. Furthermore, both filter and HPA PD compensation can restore the overall distortion almost perfectly and can achieve almost the same BER as the theoretical BER under the AWGN channel. Finally, it could be seen that the BER performance improves significantly with the PAA BF, e.g., by 14 dB gain in SNR over the no-BF case when $N_u = 5$ number of antenna array elements per user is used.

REFERENCES

[1] B. S. Watson, "Digital channelizer unit for processing multiple digital input signals," Mar. 9 2010, US Patent 7,675,985.

[2] F. J. Harris, C. Dick, and M. Rice, "Digital receivers and transmitters using polyphase filter banks for wireless communications," IEEE Transactions on Microwave Theory and Techniques, vol. 51, no. 4, pp. 1395–1412, 2003.

[3] Z. Xiang, M. Tao, and X. Wang, "Massive MIMO multicasting in noncooperative cellular networks," IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1180–1193, 2014.

[4] W. Xiong, G. Wang, X. Tian, K. Pham, E. Blasch, and G. Chen, "Hybrid onboard and ground based digital channelizer beam-forming for satcom interference mitigation and protection," in Sensors and Systems for Space Applications IX, vol. 9838, International Society for Optics and Photonics, 2016, p. 98380R.

[5] T. M. Nguyen, A. T. Guillen, and S. S. Matsunaga, "Practical achievable capacity for advanced satcom on-the-move," in MILCOM 2016-2016 IEEE Military Communications Conference, IEEE, 2016, pp. 150–155.

[6] A. A. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," IEEE Transactions on Communications, vol. 29, no. 11, pp. 1715–1720, 1981.

[7] F. Antonio, W. Hamdy, P. Heidmann, J. Heizer, N. Kasturi, D. P. Os´es, and C. Riddle, "A novel adaptive predistortion technique for power amplifiers," in 1999 IEEE 49th Vehicular Technology Conference (Cat. No. 99CH36363), vol. 2. IEEE, 1999, pp. 1505–1509.

[8] B. F. Beidas, "Adaptive digital signal predistortion for nonlinear communication systems using successive methods," IEEE Transactions on Communications, vol. 64, no. 5, pp. 2166–2175, 2016.

[9] A. Katz, R. Gray, and R. Dorval, "Wide/multiband linearization of TWTAs using predistortion," IEEE Transactions on Electronic Devices, vol. 56, no. 5, pp. 959–964, 2009.

[10] B. Ai, Z.-x. Yang, C.-y. Pan, S.-g. Tang, and T.-t. Zhang, "Analysis on LUT based predistortion method for HPA with memory," IEEE Transactions on Broadcasting, vol. 53, no. 1, pp. 127–131, 2007.

[11] Y.Lin, S.Phoong, and P.P.Vaidyanathan, Filter Bank Transceivers for OFDM and DMT Systems, Cambridge, 2011.

[12] P. Corn, E. Ross, and J. Khim. "Discrete Fourier transform," [Online], available: https://brilliant.org/wiki/discrete-fourier-transform/.

APPENDIX - MATLAB CODES

A. Additional MATLAB Functions Used

MPSK Modulator

```
function[s, ref]=mpsk_modulator(M,nsymb)
%Function to MPSK modulate the vector of data symbols - d
%[s,ref]=mpsk_modulator(M,d) modulates the symbols defined by the
%vector d using MPSK modulation, where M specifies the order of
%M-PSK modulation and the vector d contains symbols whose values
%in the range 1:M. The output s is the modulated output and ref
%represents the reference constellation that can be used in demod
ref_i= 1/sqrt(2)*cos(((1:1:M)-1)/M*2*pi);
ref_q= 1/sqrt(2)*sin(((1:1:M)-1)/M*2*pi);
ref = ref_i+1i*ref_q;
%ref=exp(1j*(((1:1:M)-1)/M*2*pi));
s = ref(nsymb); %M-PSK Mapping
end
```

MPSK Demodulator

```
function [dCap]= mpsk_detector(M,r)
%Function to detect MPSK modulated symbols
%[dCap]= mpsk_detector(M,r) detects the received MPSK signal points
%points - 'r'. M is the modulation level of MPSK
ref_i= 1/sqrt(2)*cos(((1:1:M)-1)/M*2*pi);
ref_q= 1/sqrt(2)*sin(((1:1:M)-1)/M*2*pi);
ref = ref_i+1i*ref_q; %reference constellation for MPSK
%ref=exp(1j*(((1:1:M)-1)/(M*2*pi)));
%dCap = ref(abs(r); %M-PSK Mapping
[~,dCap]= iqOptDetector(r,ref); %IQ detection
end
function [idealPoints,indices]= iqOptDetector(received,ref)
%Optimum Detector for 2-dim. signals (MQAM,MPSK,MPAM) in IQ Plane
%received - vector of form I+jQ
%ref - reference constellation of form I+jQ
%Note: MPAM/BPSK are one dim. modulations. The same function can be
%applied for these modulations since quadrature is zero (Q=0).
x=[real(received); imag(received)]';%received vec. in cartesian form
y=[real(ref); imag(ref)]';%reference vec. in cartesian form
[idealPoints,indices]= minEuclideanDistance(x,y);
end
function [idealPoints,indices]= minEuclideanDistance(x,y)
%function to compute the pairwise minimum Distance between two
%vectors x and y in p-dimensional signal space and select the
%vectors in y that provides the minimum distances.
% x - a matrix of size mxp
% y - a matrix of size nxp. This acts as a reference against
% which each point in x is compared.
% idealPoints - contain the decoded vector
% indices - indices of the ideal points in reference matrix y
[m,p1] = size(x);
[n,p2] = size(y);
if p1~=p2
error('Dimension Mismatch: x and y must have same dimension')
end
X = sum(x.*x,2);
```

```
Y = sum(y.*y,2)';
d = X(:,ones(1,n)) + Y(ones(1,m),:) - 2*x*y';%Squared Euclidean Dist.
[~,indices]=min(d,[],2); %Find the minimum value along DIM=2
idealPoints=y(indices,:);
indices=indices.';
end
```

B. MATLAB Code for Bit Error Rate Plot for Digital Channelizer without Digital Beamforming

```
clc;
clear all
n=1:5*10^5;
nSym=10^4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Rs=1/50 i.e. 50 samples per symbol%%%%%%
n1=(reshape(n,[50,nSym]))';
n2=(reshape(n,[50,nSym]))';
n3=(reshape(n,[50,nSym]))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fs=2*10^6; % sampling frequency
M=5;%up-sampling/down-sampling factor
f1=1/M;%user 1 carrier freq
f2=2/M;%user 2 carrier freq
f3=3/M;%user 3 carrier freq
Mod=8;
EbN0dB =1:1:10; % Eb/N0 range in dB for simulation
k=log2(Mod);
gammab=10.^(EbN0dB./10);
gammas=k.*gammab;
EsN0dB=10.*log10(k)+EbN0dB;
d1=ceil(Mod.*rand(1,nSym));
d2=ceil(Mod.*rand(1,nSym));
d3=ceil(Mod.*rand(1,nSym));
%%%%%%%%%%%%%%%%%%%%%%%%%% 8-PSK Modulated Symbols %%%%%%%%%%%%%%%%%%%%%%%%
s1=mpsk_modulator(Mod,d1);
s2=mpsk_modulator(Mod,d2);
s3=mpsk_modulator(Mod,d3);
%%%%%%%%%%%%%%%%%%%%%%%%% TX filter %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h0= fir1(39-1,1/5);
n_h0=1:length(h0);
h1=h0.*exp(1j.*(2.*pi.*(n_h0-(length(n_h0)/2)).*0.2)); %(1/N)
h2=h0.*exp(1j.*(2.*pi.*((n_h0+1)-(length(n_h0)/2)).*0.4)); %(2/N)
h3=h0.*exp(1j.*(2.*pi.*((n_h0+2)-(length(n_h0)/2)).*0.6)); %(3/N)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX filter %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hh1=remez(39,[1 2 3 5]/5,[1 1 0 0],[1 100]); hh=hh1;
gg1=hh.*exp(1j*2*pi*(0:39)*1/5);
gg2=hh.*exp(1j*2*pi*(0:39)*2/5);
gg3=hh.*exp(1j*2*pi*(0:39)*3/5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP A Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%
x1=[]; x2=[]; x3=[];
for ii=1:nSym
x1=[x1 (exp((1j*2*pi*1).*n1(ii,:)))*s1(ii)];
x2=[x2 (exp((1j*2*pi*1).*n2(ii,:)))*s2(ii)];
x3=[x3 (exp((1j*2*pi*1).*n3(ii,:)))*s3(ii)];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
SER_sym1=zeros(1,length(EsN0dB));
SER_sym2=zeros(1,length(EsN0dB));
SER_sym3=zeros(1,length(EsN0dB));
SER_theory=zeros(1,length(EbN0dB));
%%%%%%%%%%%%%%%%%%%%%%%%% STEP B in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%
x1_up=upsample(x1,5);
x2_up=upsample(x2,5);
x3_up=upsample(x3,5);
%%%%%%%%%%%%%%%%%%%%%%%%% STEP C in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%
u1=filter(h1,1,x1_up);
u2=filter(h2,1,x2_up);
u3=filter(h3,1,x3_up);
%%%%%%%%%%%%%%%%%%%%%%%%% STEP D in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%
uu=u1+u2+u3;
%%%%%%%%%%%%%%%% %%%%%%%%% HPA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
backoff=10;
a_1=2; b_1=1; % model parameters
a_2=4; b_2=9; % model parameters
m=10^(backoff/20);%normalization
vv = zeros(1,length(uu))*(1.0+1i*1.0); % initialize output array
for j=1:length(uu)
ain = m*abs(uu(j));
a_in(j)=ain;
end
h=(mean((a_in)));
g=((2.*mean(a_in))./(1+ (mean(a_in))^2));
c=(g/h); % slope
%%%%%%%%%%%%%%%%%%%%%%% HPA Predistorter Design %%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:length(uu)
B=(a_1+sqrt(a_1^2-(4*b_1*c^2*(abs(uu(j)))^2)))/((2*(abs(uu(j)))^2)*b_1*c);
A=-(a_2*((abs(uu(j)*B))^2))/(1+b_2*((abs(uu(j)*B))^2));
phase_dist=(A+((a_2*(abs(uu(j)*B))^2)/(1+b_2*((abs(uu(j)*B))^2))));
amp_dist=B*(a_1/(1+b_1*((abs(uu(j)*B))^2)));
uu_PD(j)=abs(uu(j))*exp(1j*angle(uu(j)))*B*exp(1j*A);
uu_HPA(j)=uu_PD(j)*(a_1/(1+(b_1*((abs(uu_PD(j)))^2))))*exp(1j*((a_2*((abs(uu_
PD(j)))^2))/(1+(b_2*((abs(uu_PD(j)))^2)))));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
y1=zeros(1,nSym); y2=zeros(1,nSym); y3=zeros(1,nSym);
for i=1:length(EsN0dB)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP E in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%
r1=filter(gg1,1,uu_HPA);
r2=filter(gg2,1,uu_HPA);
r3=filter(gg3,1,uu_HPA);
r1_d=downsample(r1,5);
r2_d=downsample(r2,5);
r3_d=downsample(r3,5);
%%%%%%%%%%%%% precompensation to remove distortion due to filters%%%%%%%%%%
pd_a1=angle(r1_d)-angle(x1);  pd_a2=angle(r2_d)-angle(x2);  pd_a3=angle(r3_d)-
angle(x3);
ad_a1=abs(r1_d)-abs(x1); ad_a2=abs(r2_d)-abs(x2); ad_a3=abs(r3_d)-abs(x3);
x1_d=((abs(r1_d)-ad_a1).*exp(1j.*(angle(r1_d)-pd_a1)));%+noise;
x2_d=((abs(r2_d)-ad_a2).*exp(1j.*(angle(r2_d)-pd_a2)));%+noise;
x3_d=((abs(r3_d)-ad_a3).*exp(1j.*(angle(r3_d)-pd_a3)));%+noise;
Es=(sum(abs(((abs(r1_d)-ad_a1).*exp(1j.*(angle(r1_d)-
pd_a1)))).^2))/length(x1_d);
```

```
noise                                                                          =
(sqrt((50*Es)/(2*(10^(EsN0dB(i)/10))))).*(randn(size(x1))+1i*randn(size(x1)))
;%computed AWGN noise
%%%%%%%%%%%%%%%%%%%%%%% STEP F in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:nSym
y1(ii)=mean((x1_d(n1(ii,:))./(exp((1j*2*pi*1).*n1(ii,:))))+noise(n1(ii,:))./(
exp((1j*2*pi*1).*n1(ii,:))));
y2(ii)=mean((x2_d(n2(ii,:))./(exp((1j*2*pi*1).*n2(ii,:))))+noise(n2(ii,:))./(
exp((1j*2*pi*1).*n2(ii,:))));
y3(ii)=mean((x3_d(n3(ii,:))./(exp((1j*2*pi*1).*n3(ii,:))))+noise(n3(ii,:))./(
exp((1j*2*pi*1).*n3(ii,:))));
end
%%%%%%%%%%%%%%%%%%% Demodulation and Symbol detection %%%%%%%%%%%%%%%%%%%%%
dcap1=mpsk_detector(Mod,y1);
dcap2=mpsk_detector(Mod,y2);
dcap3=mpsk_detector(Mod,y3);
%%%%%%%%%%%%%%%%%%%%%%%%% Symbol error rate calculation %%%%%%%%%%%%%%%%%%
SER_sym1(i)=(sum(d1~=dcap1))/nSym;
SER_sym2(i)=(sum(d2~=dcap2))/nSym;
SER_sym3(i)=(sum(d3~=dcap3))/nSym;
SER_theory(i)=erfc((sqrt(10^(EsN0dB(i)/10)))*sin(pi/M));
end
%%%%%%%%%%%%%%%%%%%%%%%%%% Bit error rate plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%
theoreticalBER=(berawgn(EbN0dB,'psk',8,'nondiff'));
figure(1)
semilogy(EbN0dB, SER_sym1./k, '-*')
hold on
semilogy(EbN0dB, SER_sym2./k, '-*')
hold on
semilogy(EbN0dB, SER_sym3./k, '-*')
hold on
semilogy(EbN0dB,theoreticalBER,'--')
hold on
xlabel('Eb_N0 (dB)')
ylabel('BER')
title('Bit error rate plot for 8PSK digital channelizer with HPA predistortion')
legend('BER user 1','BER user 2','BER user 3','Theoretical BER')
```

C. MATLAB Code for Bit Error Rate Plot for Digital Channelizer with Digital Beamforming

```
clc
clear all
%%%%%%%%%%%%%% beamforming vector generation %%%%%%%%%%%%%%%%%%%%%%%%%%
Nt=15; %Total number of antenna elements
Nu=3; %total no. of users
Ntu=Nt/Nu; % antenna per user
r_1=0;
 r_n_x=1:Nt;
d=r_n_x-r_1; % distance between antenna elements
d_1=1:Ntu; % set of antennas for user 1
d_2=Ntu+1:Ntu+5; % set of antennas for user 2
d_3=Ntu+6:Ntu+10; % set of antennas for user 3
theta_1=45; % Angle of arrival for user 1
theta_2=60; % Angle of arrival for user 2
theta_3=120; % Angle of arrival for user 3
```

```
%%%%%%%% since we consider linear array in the elevation plane,
%d in equation (14) is only function of cos(theta_k(n)) %%%%%%%%%%%
sai1= cos(theta_1*pi/180);
sai2= cos(theta_2*pi/180);
sai3= cos(theta_3*pi/180);
%%%%%%%%%%%%% weight vector calculation using equation (14) %%%%%%%%
w1 =exp(1j*pi*(d_1')*sai1);
w2 = exp(1j*pi*(d_2')*sai2);
w3 = exp(1j*pi*(d_3')*sai3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=1:5*10^5;
nSym=10^4;
%%%%%%%%%%%%%%%%%%%% Rs=1/50 i.e. 50 samples per symbol
n1=(reshape(n,[50,nSym]))';
n2=(reshape(n,[50,nSym]))';
n3=(reshape(n,[50,nSym]))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fs=2*10^6;%sampling frequency
M=5;%up-sampling/down-sampling factor
f1=1/M; % user 1 carrier freq
f2=2/M; %user 2 carrier freq
f3=3/M; %user 3 carrier freq
Mod=8;  %8-PSK modulation is used
EbN0dB =-5:2:12; % Eb/N0 range in dB for simulation
k=log2(Mod);
gammab=10.^(EbN0dB./10);
gammas=k.*gammab;
EsN0dB=10.*log10(k)+EbN0dB;
d1=ceil(Mod.*rand(1,nSym));
d2=ceil(Mod.*rand(1,nSym));
d3=ceil(Mod.*rand(1,nSym));
%%%%%%%%%%%%%%%%%%% 8-PSK modulation of the input symbols %%%%%%%
s1=mpsk_modulator(Mod,d1);
s2=mpsk_modulator(Mod,d2);
s3=mpsk_modulator(Mod,d3);
%%%%%%%%%%%%%%%%%%%%%%%%% TX filter %%%%%%%%%%%%%%%%%%%%%%%%%%%
h0= fir1(39-1,1/5);
n_h0=1:length(h0);
h1=h0.*exp(1j.*(2.*pi.*(n_h0-(length(n_h0)/2)).*0.2)); %(1/N)
h2=h0.*exp(1j.*(2.*pi.*((n_h0+1)-(length(n_h0)/2)).*0.4)); %(2/N)
h3=h0.*exp(1j.*(2.*pi.*((n_h0+2)-(length(n_h0)/2)).*0.6)); %(3/N)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX filter %%%%%%%%%%%%%%%%%%%%%%%%
hh1=remez(39,[1 2 3 5]/5,[1 1 0 0],[1 100]); hh=hh1;
 gg1=hh.*exp(1j*2*pi*(0:39)*1/5);
gg2=hh.*exp(1j*2*pi*(0:39)*2/5);
gg3=hh.*exp(1j*2*pi*(0:39)*3/5);
%%%%%%%%%%%%%%%%%%%%%%%%% STEP A in Fig. 2 and 3 %%%%%%%%%%%%%%%%
x1=[]; x2=[]; x3=[];
for ii=1:nSym
x1=[x1 (exp((1j*2*pi*1).*n1(ii,:)))*s1(ii)];
x2=[x2 (exp((1j*2*pi*1).*n2(ii,:)))*s2(ii)];
x3=[x3 (exp((1j*2*pi*1).*n3(ii,:)))*s3(ii)];
end
%%%%%%%%%%%%%%%%%%%%%%%% STEP B in Fig 2. and 3 %%%%%%%%%%%%%%%%
x1_up=upsample(x1,5);
x2_up=upsample(x2,5);
x3_up=upsample(x3,5);
```

```
%%%%%%%%%%%%%%%%%%%%%%% STEP C in Fig 2. and 3 %%%%%%%%%%%%%%%%
u1=filter(h1,1,x1_up);
u2=filter(h2,1,x2_up);
u3=filter(h3,1,x3_up);
%%%%%%%time domain separated signals of each user are fed%%%%%%%to the HPA
uu1=u1;
uu2=u2;
uu3=u3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% HPA Model %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
backoff=10;
a_1=2; b_1=1; % model parameters
a_2=4; b_2=9; % model parameters
m=10^(backoff/20);%normalization
vv = zeros(1,length(uu1))*(1.0+1i*1.0); % initialize output array
for j=1:length(uu1)
ain1 = m*abs(uu1(j));
ain2 = m*abs(uu2(j));
ain3 = m*abs(uu3(j));
a_in1(j)=ain1;
a_in2(j)=ain2;
a_in3(j)=ain3;
end
h1=(mean((a_in1)));
g1=((2.*mean(a_in1))./(1+ (mean(a_in1))^2));
c1=g1/h1; % slope
h2=(mean((a_in2)));
g2=((2.*mean(a_in2))./(1+ (mean(a_in2))^2));
c2=g2/h2; % slope
h3=(mean((a_in3)));
g3=((2.*mean(a_in3))./(1+ (mean(a_in3))^2));
c3=g3/h3; % slope
%%%%%%%%%%%%%%%%%%%%%%%%% HPA predistorter Modelling %%%%%%%%%%%%%%%%%%%%
for j=1:length(uu1)
B1=(a_1+sqrt(a_1^2-
(4*b_1*c1^2*(abs(uu1(j)))^2)))/((2*(abs(uu1(j)))^2)*b_1*c1);
A1=-(a_2*((abs(uu1(j)*B1))^2))/(1+b_2*((abs(uu1(j)*B1))^2));
phase_dist1=(A1+((a_2*(abs(uu1(j)*B1))^2)/(1+b_2*((abs(uu1(j)*B1))^2))));
amp_dist1=B1*(a_1/(1+b_1*((abs(uu1(j)*B1))^2)));
uu_PD1(j)=abs(uu1(j))*exp(1j*angle(uu1(j)))*B1*exp(1j*A1);
uu_HPA1(j)=uu_PD1(j)*(a_1/(1+(b_1*((abs(uu_PD1(j)))^2))))*exp(1j*((a_2*((abs(
uu_PD1(j)))^2))/(1+(b_2*((abs(uu_PD1(j)))^2)))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B2=(a_1+sqrt(a_1^2-
(4*b_1*c2^2*(abs(uu2(j)))^2)))/((2*(abs(uu2(j)))^2)*b_1*c2);
A2=-(a_2*((abs(uu2(j)*B2))^2))/(1+b_2*((abs(uu2(j)*B2))^2));
phase_dist2=(A2+((a_2*(abs(uu2(j)*B2))^2)/(1+b_2*((abs(uu2(j)*B2))^2))));
amp_dist2=B2*(a_1/(1+b_1*((abs(uu2(j)*B2))^2)));
uu_PD2(j)=abs(uu2(j))*exp(1j*angle(uu2(j)))*B2*exp(1j*A2);
uu_HPA2(j)=uu_PD2(j)*(a_1/(1+(b_1*((abs(uu_PD2(j)))^2))))*exp(1j*((a_2*((abs(
uu_PD2(j)))^2))/(1+(b_2*((abs(uu_PD2(j)))^2)))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B3=(a_1+sqrt(a_1^2-
(4*b_1*c3^2*(abs(uu3(j)))^2)))/((2*(abs(uu3(j)))^2)*b_1*c3);
A3=-(a_2*((abs(uu3(j)*B3))^2))/(1+b_2*((abs(uu3(j)*B3))^2));
phase_dist3=(A1+((a_2*(abs(uu3(j)*B3))^2)/(1+b_2*((abs(uu3(j)*B3))^2))));
amp_dist3=B3*(a_1/(1+b_1*((abs(uu3(j)*B3))^2)));
uu_PD3(j)=abs(uu3(j))*exp(1j*angle(uu3(j)))*B3*exp(1j*A3);
```

```
uu_HPA3(j)=uu_PD3(j)*(a_1/(1+(b_1*((abs(uu_PD3(j)))^2))))*exp(1j*((a_2*((abs(
uu_PD3(j)))^2))/(1+(b_2*((abs(uu_PD3(j)))^2)))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y1=zeros(1,nSym);
y2=zeros(1,nSym);
y3=zeros(1,nSym);
y1_bf=zeros(1,nSym);
y2_bf=zeros(1,nSym);
y3_bf=zeros(1,nSym);
y1_th=zeros(1,nSym);
y2_th=zeros(1,nSym);
y3_th=zeros(1,nSym);
SER_sym1=zeros(1,length(EsN0dB));
SER_sym1_bf=zeros(1,length(EsN0dB));
SER_sym1_th=zeros(1,length(EsN0dB));
SER_sym2=zeros(1,length(EsN0dB));
SER_sym2_bf=zeros(1,length(EsN0dB));
SER_sym2_th=zeros(1,length(EsN0dB));
SER_sym3=zeros(1,length(EsN0dB));
SER_sym3_bf=zeros(1,length(EsN0dB));
SER_sym3_th=zeros(1,length(EsN0dB));
SER_theory=zeros(1,length(EbN0dB));
SER_BFtheory=zeros(1,length(EbN0dB));
for i=1:length(EsN0dB)
%%%%%%%%%%%%%%%% STEP E in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r1_fil=filter(gg1,1,uu_HPA1);
r2_fil=filter(gg2,1,uu_HPA2);
r3_fil=filter(gg3,1,uu_HPA3);
uu_HPA_d1=downsample(r1_fil,5);
uu_HPA_d2=downsample(r2_fil,5);
uu_HPA_d3=downsample(r3_fil,5);
%%%%%%%% precompensation to remove distortion due to filters %%%%%%%%%%%
pd_a1=angle(uu_HPA_d1)-angle(x1);           pd_a2=angle(uu_HPA_d2)-angle(x2);
pd_a3=angle(uu_HPA_d3)-angle(x3);
ad_a1=abs(uu_HPA_d1)-abs(x1);                   ad_a2=abs(uu_HPA_d2)-abs(x2);
ad_a3=abs(uu_HPA_d3)-abs(x3);
x1_d=((abs(uu_HPA_d1)-ad_a1).*exp(1j.*(angle(uu_HPA_d1)-pd_a1)));%+noise;
x2_d=((abs(uu_HPA_d2)-ad_a2).*exp(1j.*(angle(uu_HPA_d2)-pd_a2)));%+noise;
x3_d=((abs(uu_HPA_d3)-ad_a3).*exp(1j.*(angle(uu_HPA_d3)-pd_a3)));%+noise;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r1=(w1'*w1)*x1_d;
r2=(w2'*w2)*x2_d;
r3=(w3'*w3)*x3_d;
r1_th=(w1'*w1)*x1;
r2_th=(w1'*w1)*x2;
r3_th=(w1'*w1)*x3;
%%%%%%%%%%%%%%%%%%%%%%% signal energy calculation %%%%%%%%%%%%%%
Es1=((sum((abs(x1)).^2))/length(x1));
Es2=((sum((abs(x2)).^2))/length(x2));
Es3=((sum((abs(x3)).^2))/length(x3));
%%%%%%%%%%%%%%%%%%%%%%%%%%% AWGN noise calculation %%%%%%%%%%%%%%%
noise1=(sqrt((50*Es1)/(2*(10^(EsN0dB(i)/10))))).*(randn(size(uu_HPA_d1))+1i*r
andn(size(uu_HPA_d1)));
noise2=(sqrt((50*Es2)/(2*(10^(EsN0dB(i)/10))))).*(randn(size(uu_HPA_d2))+1i*r
andn(size(uu_HPA_d2)));
```

```
noise3=(sqrt((50*Es3)/(2*(10^(EsN0dB(i)/10))))).*(randn(size(uu_HPA_d3))+1i*r
andn(size(uu_HPA_d3)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    STEP    F    in    Fig.    2    and    3
%%%%%%%%%%%%%%%%%%%%%
for ii=1:nSym
y1(ii)=mean((x1(n1(ii,:))./(exp((1j*2*pi*1).*n1(ii,:))))+noise1(n1(ii,:))./(e
xp((1j*2*pi*1).*n1(ii,:))));
y1_bf(ii)=mean((r1(n1(ii,:))./(exp((1j*2*pi*1).*n1(ii,:))))+noise1(n1(ii,:)).
/(exp((1j*2*pi*1).*n1(ii,:))));
y1_th(ii)=mean((r1_th(n1(ii,:))./(exp((1j*2*pi*1).*n1(ii,:))))+noise1(n1(ii,:
))./(exp((1j*2*pi*1).*n1(ii,:))));
y2(ii)=mean((x2(n2(ii,:))./(exp((1j*2*pi*1).*n2(ii,:))))+noise2(n2(ii,:))./(e
xp((1j*2*pi*1).*n2(ii,:))));
y2_bf(ii)=mean((r2(n2(ii,:))./(exp((1j*2*pi*1).*n2(ii,:))))+noise2(n2(ii,:)).
/(exp((1j*2*pi*1).*n2(ii,:))));
y2_th(ii)=mean((r2_th(n2(ii,:))./(exp((1j*2*pi*1).*n2(ii,:))))+noise2(n2(ii,:
))./(exp((1j*2*pi*1).*n2(ii,:))));
y3(ii)=mean((x3(n3(ii,:))./(exp((1j*2*pi*1).*n3(ii,:))))+noise3(n3(ii,:))./(e
xp((1j*2*pi*1).*n3(ii,:))));
 y3_bf(ii)=mean((r3(n3(ii,:))./(exp((1j*2*pi*1).*n3(ii,:))))+noise3(n3(ii,:)).
 /(exp((1j*2*pi*1).*n3(ii,:))));
y3_th(ii)=mean((r3_th(n3(ii,:))./(exp((1j*2*pi*1).*n3(ii,:))))+noise3(n3(ii,:
))./(exp((1j*2*pi*1).*n3(ii,:))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
end
%%%%%%%%% demodulation and received symbol detection %%%%%%%%%%%
dcap1=mpsk_detector(Mod,y1);
dcap1_bf=mpsk_detector(Mod,y1_bf);
dcap1_th=mpsk_detector(Mod,y1_th);
dcap2=mpsk_detector(Mod,y2);
dcap2_bf=mpsk_detector(Mod,y2_bf);
dcap2_th=mpsk_detector(Mod,y2_th);
dcap3=mpsk_detector(Mod,y3);
dcap3_bf=mpsk_detector(Mod,y3_bf);
dcap3_th=mpsk_detector(Mod,y3_th);
%%%%%%%%%%%%%%%%%%%%%%%%%% symbol error rate calculation %%%%%%%%%%%%%%%
SER_sym1(i)=(sum(d1~=dcap1))/nSym;
SER_sym1_bf(i)=(sum(d1~=dcap1_bf))/nSym;
SER_sym1_th(i)=(sum(d1~=dcap1_th))/nSym;
SER_sym2(i)=(sum(d2~=dcap2))/nSym;
SER_sym2_bf(i)=(sum(d2~=dcap2_bf))/nSym;
SER_sym2_th(i)=(sum(d2~=dcap2_th))/nSym;
SER_sym3(i)=(sum(d3~=dcap3))/nSym;
SER_sym3_bf(i)=(sum(d3~=dcap3_bf))/nSym;
SER_sym3_th(i)=(sum(d3~=dcap3_th))/nSym;
SER_theory(i)=erfc((sqrt(10^(EsN0dB(i)/10)))*sin(pi/M));
end
%
%%%%%%%%%%%% Bit error rate calculation %%%%%%%%%%%%%%%%%%%%%%%
theoreticalBER=(berawgn(EbN0dB,'psk',8,'nondiff'));
figure(1)
semilogy(EbN0dB,SER_sym1./k,'r-p',EbN0dB,SER_sym2./k,'b-
h',EbN0dB,SER_sym3./k,'g-x','MarkerSize',6,'LineWidth',2)
hold on
semilogy(EbN0dB,theoreticalBER,'m-d','MarkerSize',6,'LineWidth',2)
hold on
```

```
semilogy(EbN0dB, SER_sym1_bf./k, 'r-*',EbN0dB, SER_sym2_bf./k, 'b-o',EbN0dB,
SER_sym3_bf./k, 'g-+','MarkerSize',6,'LineWidth',2)
hold on
semilogy(EbN0dB, SER_sym1_th./k, 'k-s','MarkerSize',6,'LineWidth',2)
hold on
xlabel('Eb/N0 dB')
ylabel('BER')
legend('no BF, AWGN-simulation, user 1, 8-PSK','no BF, AWGN-simulation, user 2,
8-PSK','no BF, AWGN-simulation, user 3, 8-PSK','no BF,AWGN-theory, 8-PSK','with
BF, AWGN-simulation, user 1, 8-PSK','with BF, AWGN-simulation, user 2, 8-
PSK','with BF, AWGN-simulation, user 3, 8-PSK','with BF,AWGN-theory, 8-PSK')
 grid on
```

D. MATLAB Code for Frequency Domain Representation at Steps A, B, C, D, E, and F in Figs. 2 and 3

```
clc;
clear all
n=1:200;% number of samples for simulation purpose
nSym=4; % number of symbols for simulation purpose
%%%%%%%%% Rs=1/50 i.e. 50 samples per symbols %%%%%
n1=(reshape(n,[50,nSym]))';
n2=(reshape(n,[50,nSym]))';
n3=(reshape(n,[50,nSym]))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fs=2*10^6;%sampling frequency
M=5;%up-sampling/down-sampling factor
f1=1/M; % user 1 carrier freq
f2=2/M; % user 2 carrier freq
f3=3/M; % user 3 carrier freq
Mod=8; % 8-PSK modulation used
d1=ceil(Mod.*rand(1,nSym));
d2=ceil(Mod.*rand(1,nSym));
d3=ceil(Mod.*rand(1,nSym));
%%%%%%%%% 8-PSK modulation of the input symbols %%%%
s1=mpsk_modulator(Mod,d1);
s2=mpsk_modulator(Mod,d2);
s3=mpsk_modulator(Mod,d3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1=[];
x2=[];
x3=[];
h1=[];
h2=[];
h3=[];
%%%%%%%%%%%%%%%%%%% STEP A in Fig. 2 and Fig. 3
for ii=1:nSym
x1=(1/length(n1(ii,:))).*[x1 (exp((1j*2*pi*1).*n1(ii,:)))*s1(ii)];
x2=(1/length(n1(ii,:))).*[x2 (exp((1j*2*pi*1).*n2(ii,:)))*s2(ii)];
x3=(1/length(n1(ii,:))).*[x3 (exp((1j*2*pi*1).*n3(ii,:)))*s3(ii)];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%% TX filters %%%%%%%%%%%%%%%%%%%%
h0= fir1(39-1,1/5);
n_h0=1:length(h0);
h1=h0.*exp(1j.*(2.*pi.*(n_h0-(length(n_h0)/2)).*0.2)); %(1/N)
h2=h0.*exp(1j.*(2.*pi.*((n_h0+1)-(length(n_h0)/2)).*0.4)); %(2/N)
```

```
h3=h0.*exp(1j.*(2.*pi.*((n_h0+2)-(length(n_h0)/2)).*0.6)); %(3/N)
%%%%%%%%%%%%%%%%%%%%%%%% STEP B in Fig. 2 and 3 %%%%%%%%%%%%%
x1_up=upsample(x1,M);
x2_up=upsample(x2,M);
x3_up=upsample(x3,M);
%%%%%%%%%%%%%%%%%%%%%%% STEP C in Fig. 2 and 3 %%%%%%%%%%%%%%
n_up=1:length(x1_up);
u1=filter(h1,1,x1_up);
u2=filter(h2,1,x2_up);
u3=filter(h3,1,x3_up);
%%%%%%%%%%%%%%%%%%%%%%% STEP D in Fig. 2 and 3 %%%%%%%%%%%%%%
uu=u1+u2+u3;
%%%%%%%%%%%%%%%%%%%%%%%%%%% HPA model %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
backoff=10;
%backoff=18;
a_1=2; b_1=1; % model parameters
a_2=4; b_2=2; % model parameters
m=10^(backoff/20);%normalization
vv = zeros(1,length(uu))*(1.0+1i*1.0); % initialize output array
for k=1:length(uu)
ain = m*abs(uu(k));
a_in(k)=ain;
thetain(k) = angle(uu(k));
aout = a_1*ain/(1+b_1*ain^2);
a_out(k)=aout;
thetapm(k) = a_2*ain^2/(1+b_2*ain^2);
thetaout(k) = thetain(k)+thetapm(k);
vv(k) = aout*exp(1i*thetaout(k));
end
h=(mean((a_in).^2));
g=((2.*mean(a_in))./(1+ (mean(a_in)).^2));
c=g/h; % slope
%%%%%%%%%%%%%%%%%% HPA Predistorter design %%%%%%%%%%%%%%%%%
for k=1:length(uu)
B=(a_1+sqrt(a_1^2-(4*b_1*c^2*(abs(uu(k)))^2)))/((2*(abs(uu(k)))^2)*b_1*c);
A=-(a_2*((abs(uu(k)*B))^2))/(1+b_2*((abs(uu(k)*B))^2));
phase_dist=(A+((a_2*(abs(uu(k)*B))^2)/(1+b_2*((abs(uu(k)*B))^2))));
amp_dist=B*(a_1/(1+b_1*((abs(uu(k)*B))^2)));
uu_PD(k)=abs(uu(k))*exp(1j*angle(uu(k)))*B*exp(1j*A);
%x_PD(k)=x(k)*B*exp(1j*A);
uu_HPA(k)=uu_PD(k)*(a_1/(1+(b_1*((abs(uu_PD(k)))^2))))*exp(1j*((a_2*((abs(uu_
PD(k)))^2))/(1+(b_2*((abs(uu_PD(k)))^2)))));
end
%%%%%%%%%%%%%%%%%%%%%%% RX Filter %%%%%%%%%%%%%%%%%%%%%%%%%%%
hh1=remez(39,[1 2 3 5]/5,[1 1 0 0],[1 100]); hh=hh1;
gg1=hh.*exp(1j*2*pi*(0:39)*1/5);
gg2=hh.*exp(1j*2*pi*(0:39)*2/5);
gg3=hh.*exp(1j*2*pi*(0:39)*3/5);
%%%%%%%%%%%%%%%%%%%%%%% STEP E in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%
y1_out=filter(gg1,1,uu_HPA);
y2_out=filter(gg2,1,uu_HPA);
y3_out=filter(gg3,1,uu_HPA);
%%%%%%%%%%%%%%%%%%%%%%%%%% STEP F in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%
y1_out_d=downsample(y1_out,M);
y2_out_d=downsample(y2_out,M);
y3_out_d=downsample(y3_out,M);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%DFT transform to see frequency domain plots at STEP A, B, C, D, E and F
X1=zeros(1,length(x1));
X2=zeros(1,length(x2));
X3=zeros(1,length(x3));
%XX=zeros(1,length(xx));
for k = 0:(length(x1)-1)
    for len = 0:(length(x1)-1)
        X1(k+1) = X1(k+1) + (x1(len+1)*exp(-1j*2*(pi/length(x1))*len*k));
        X2(k+1) = X2(k+1) + (x2(len+1)*exp(-1j*2*(pi/length(x2))*len*k));
        X3(k+1) = X3(k+1) + (x3(len+1)*exp(-1j*2*(pi/length(x3))*len*k));
        %XX(k+1) = XX(k+1) + (xx(len+1)*exp(-1j*2*(pi/length(xx))*len*k));
    end
end

X1_up=zeros(1,length(x1_up));
X2_up=zeros(1,length(x2_up));
X3_up=zeros(1,length(x3_up));
for k = 0:(length(x1_up)-1)
    for len = 0:(length(x1_up)-1)

X1_up(k+1)=X1_up(k+1)+(x1_up(len+1)*exp(1j*2*(pi/length(x1_up))*len*k));
X2_up(k+1) = X2_up(k+1) + (x2_up(len+1)*exp(-1j*2*(pi/length(x2_up))*len*k));
X3_up(k+1) = X3_up(k+1) + (x3_up(len+1)*exp(-1j*2*(pi/length(x3_up))*len*k));
    end
end
U1=zeros(1,length(u1));
U2=zeros(1,length(u2));
U3=zeros(1,length(u3));
UU=zeros(1,length(uu));
UU_HPA=zeros(1,length(uu_HPA));
for k = 0:(length(u1)-1)
    for len = 0:(length(u1)-1)
        U1(k+1) = U1(k+1) + (u1(len+1)*exp(-1j*2*(pi/length(u1))*len*k));
        U2(k+1) = U2(k+1) + (u2(len+1)*exp(-1j*2*(pi/length(u2))*len*k));
        U3(k+1) = U3(k+1) + (u3(len+1)*exp(-1j*2*(pi/length(u3))*len*k));
        UU(k+1) = UU(k+1) + (uu(len+1)*exp(-1j*2*(pi/length(uu))*len*k));

UU_HPA(k+1)=UU_HPA(k+1)+(uu_HPA(len+1)*exp(1j*2*(pi/length(uu))*len*k));
    end
end
Y1_out=zeros(1,length(y1_out));
Y2_out=zeros(1,length(y2_out));
Y3_out=zeros(1,length(y3_out));
for k = 0:(length(y1_out)-1)
    for len = 0:(length(y1_out)-1)

Y1_out(k+1)=Y1_out(k+1)+(y1_out(len+1)*exp(1j*2*(pi/length(y1_out))*len*k));
Y2_out(k+1)=Y2_out(k+1)+(y2_out(len+1)*exp(1j*2*(pi/length(y2_out))*len*k));
Y3_out(k+1)=Y3_out(k+1)+(y3_out(len+1)*exp(1j*2*(pi/length(y3_out))*len*k));
    end
end
Y1_out_d=zeros(1,length(y1_out_d));
Y2_out_d=zeros(1,length(y2_out_d));
Y3_out_d=zeros(1,length(y3_out_d));
for k = 0:(length(y1_out_d)-1)
    for len = 0:(length(y1_out_d)-1)
```

```
Y1_out_d(k+1)=Y1_out_d(k+1)+(y1_out_d(len+1)*exp(1j*2*(pi/length(y1_out_d))*l
en*k));
Y2_out_d(k+1)=Y2_out_d(k+1)+(y2_out_d(len+1)*exp(1j*2*(pi/length(y2_out_d))*l
en*k));
Y3_out_d(k+1)=Y3_out_d(k+1)+(y3_out_d(len+1)*exp(1j*2*(pi/length(y3_out_d))*l
en*k));
    end
end
w = pi.*(0:(length(x1)-1))./length(x1);
w_0=pi.*(-length(x1)/2:(length(x1)/2)-1)./length(x1);
w_1 = pi.*(0:(length(u1)-1))./length(u1);
w4=(w*fs)./(2*pi);
w4_1=(w_1*fs)./(2*pi);
w4_0=(w_0*fs)./(2*pi);
%frequency domain plot at STEP A %%%%%%%
figure(1)
plot(w4_0,fftshift(abs(X1)),'r')
hold on
plot(w4_0,fftshift(abs(X2)),'b')
hold on
plot(w4_0,fftshift(abs(X3)),'g')
hold on
%plot(w4,abs(XX))
hold on
xlabel('Frequency(MHz)');
ylabel('|X(k)|');
title('Frequency domain - Modulated Input signal')
%Frequency domain plot at STEP B in Fig. 2 and 3 %%%%%%%%%%%%
figure(2)
plot(w4_1,abs(X1_up),'r')
hold on
plot(w4_1,abs(X2_up),'b')
hold on
plot(w4_1,abs(X3_up),'g')
hold on
xlabel('Frequency(MHz)');
ylabel('|X_up(k)|');
title('Frequency domain - upsampled input signals')
%%%%%%%%%%%%%%%%%%%%% Frequency domain plot at STEP C in Fig. 2 and 3 %%%
figure(3)
%subplot(1,1,1)
plot(w4_1,abs(U1),'r')
hold on
plot(w4_1,abs(U2),'b')
hold on
plot(w4_1,abs(U3),'g')
hold on
xlabel('Frequency(MHz)');
ylabel('|U(n)|');
figure(4)
plot(w4_1,abs(UU_HPA),'m')
hold on
xlabel('Frequency(MHz)');
ylabel('|UU(k)|');
title('Frequency domain - sum of transmit filter output after HPA and HPA
predistortion')
```

```
a1_d_c=[];
a2_d_c=[];
a3_d_c=[];
a11_out=[];
a22_out=[];
a33_out=[];

y1=[]; y2=[]; y3=[];
for jj=1:nSym
a1= (exp((1j*2*pi*1).*n1(jj,:)))*s1(jj);
a2= (exp((1j*2*pi*1).*n2(jj,:)))*s2(jj);
a3= (exp((1j*2*pi*1).*n3(jj,:)))*s3(jj);
a1_up=upsample(a1,M);
a2_up=upsample(a2,M);
a3_up=upsample(a3,M);
a1_filt=filter(h1,1,a1_up);
a2_filt=filter(h2,1,a2_up);
a3_filt=filter(h3,1,a3_up);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aa=a1_filt+a2_filt+a3_filt;
for k=1:length(aa)
B=(a_1+sqrt(a_1^2-(4*b_1*c^2*(abs(aa(k)))^2)))/((2*(abs(aa(k)))^2)*b_1*c);
A=-(a_2*((abs(aa(k)*B))^2))/(1+b_2*((abs(aa(k)*B))^2));
phase_dist=(A+((a_2*(abs(aa(k)*B))^2)/(1+b_2*((abs(aa(k)*B))^2))));
amp_dist=B*(a_1/(1+b_1*((abs(aa(k)*B))^2)));
aa_PD(k)=abs(aa(k))*exp(1j*angle(aa(k)))*B*exp(1j*A);
aa_HPA(k)=aa_PD(k)*(a_1/(1+(b_1*((abs(aa_PD(k)))^2))))*exp(1j*((a_2*((abs(aa_
PD(k)))^2))/(1+(b_2*((abs(aa_PD(k)))^2)))));
end
Es=(sum((abs(aa_HPA)).^2))/length(aa_HPA);
noise                                                                          =
(sqrt((Es)/(2*(10^(10/10))))).*(randn(size(aa_HPA))+1i*randn(size(aa_HPA)));%
computed noise
rr=aa_HPA+noise;
a1_out=filter(gg1,1,aa_HPA);
a2_out=filter(gg2,1,aa_HPA);
a3_out=filter(gg3,1,aa_HPA);
a1_out_n=filter(gg1,1,rr);
a2_out_n=filter(gg2,1,rr);
a3_out_n=filter(gg3,1,rr);
a11_out=1/50.*[a11_out a1_out];
a22_out=1/50.*[a22_out a2_out];
a33_out=1/50.*[a33_out a3_out];
a1_d=downsample(a1_out,M);
a2_d=downsample(a2_out,M);
a3_d=downsample(a3_out,M);
a1_d_n=downsample(a1_out,M);
a2_d_n=downsample(a2_out,M);
a3_d_n=downsample(a3_out,M);
%%%%%%%%%%%%%%% precompensation to avoid distortions due to filter %%%%%%%
pd_a1=angle(a1)-angle(a1_d);   pd_a2=angle(a2)-angle(a2_d);   pd_a3=angle(a3)-
angle(a3_d);
ad_a1=abs(a1)-abs(a1_d); ad_a2=abs(a2)-abs(a2_d); ad_a3=abs(a3)-abs(a3_d);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a1_d=(abs(a1_d_n)+ad_a1).*exp(1j.*(angle(a1_d_n)+pd_a1));
a2_d=(abs(a2_d_n)+ad_a2).*exp(1j.*(angle(a2_d_n)+pd_a2));
a3_d=(abs(a3_d_n)+ad_a3).*exp(1j.*(angle(a3_d_n)+pd_a3));
```

```
%
a1_d_c=1/50.*[a1_d_c a1_d];
a2_d_c=1/50.*[a2_d_c a2_d];
a3_d_c=1/50.*[a3_d_c a3_d];

end


A11_out=zeros(1,length(a11_out));
A22_out=zeros(1,length(a22_out));
A33_out=zeros(1,length(a33_out));

for k = 0:(length(a11_out)-1)
    for len = 0:(length(a11_out)-1)

A11_out(k+1)=A11_out(k+1)+(a11_out(len+1)*exp(1j*2*(pi/length(a11_out))*len*k
));
A22_out(k+1)=A22_out(k+1)+(a22_out(len+1)*exp(1j*2*(pi/length(a22_out))*len*k
));
A33_out(k+1)=A33_out(k+1)+(a33_out(len+1)*exp(1j*2*(pi/length(a33_out))*len*k
));
    end
end
%%%%%%%%%%% frequency domain plot at STEP E in Fig. 2 and 3 %%%%%%%%%%%%%
figure(5)
plot(w4_1,abs(A11_out))
hold on
plot(w4_1,abs(A22_out))
hold on
plot(w4_1,abs(A33_out))
hold on
xlabel('Frequency(MHz)');
ylabel('|V(k)|');
title('Frequency domain - Receiver filter output (with noise)')

A1_d_c=zeros(1,length(a1_d_c));
A2_d_c=zeros(1,length(a2_d_c));
A3_d_c=zeros(1,length(a3_d_c));

for k = 0:(length(a1_d_c)-1)
    for len = 0:(length(a1_d_c)-1)

A1_d_c(k+1)=A1_d_c(k+1)+(a1_d_c(len+1)*exp(1j*2*(pi/length(a1_d_c))*len*k));
A2_d_c(k+1)=A2_d_c(k+1)+(a2_d_c(len+1)*exp(1j*2*(pi/length(a2_d_c))*len*k));
A3_d_c(k+1)=A3_d_c(k+1)+ (a3_d_c(len+1)*exp(-1j*2*(pi/length(a3_d_c))*len*k));
    end
end
%%%%%%%%%%% frequency domain plot at STEP F in Fig. 2 and 3 %%%%%%%%%%%%%
figure(6)
plot(w4_0,fftshift(abs(A1_d_c)),'r')
hold on
plot(w4_0,fftshift(abs(A2_d_c)),'b')
hold on
plot(w4_0,fftshift(abs(A3_d_c)),'g')
hold on
xlabel('Frequency(MHz)');
```

```
ylabel('|Y_{k}(n)|');
title('Frequency domain - demodulated received signal (with noise)')
```

E. MATLAB Code for Time Domain Representation at Steps A, B, C, D, E, and F in Figs. 2 and 3

```
clc;
clear all
n=1:200; % for simulation purpose we assumed 200 samples
nSym=4; % for simulation purpose considered 4 symbols
%%%%%%%%%%% Rs=1/50 i.e. 50 samples per symbol
n1=(reshape(n,[50,nSym]))';
n2=(reshape(n,[50,nSym]))';
n3=(reshape(n,[50,nSym]))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fs=2*10^6;
M=5;
f1=1/M;
f2=2/M;f3=3/M;
Mod=8;
EbN0dB =1:1:10; % Eb/N0 range in dB for simulation
k=log2(Mod);
gammab=10.^(EbN0dB./10);
gammas=k.*gammab;
EsN0dB=10.*log10(k)+EbN0dB;
d1=ceil(Mod.*rand(1,nSym));
d2=ceil(Mod.*rand(1,nSym));
d3=ceil(Mod.*rand(1,nSym));
s1=mpsk_modulator(Mod,d1);
s2=mpsk_modulator(Mod,d2);
s3=mpsk_modulator(Mod,d3);
%%%%%%%%%%%%%%%%%%%%%%%% TX filter %%%%%%%%%%%%%%%%%%%%%%%%
h0= fir1(39-1,1/5);
n_h0=1:length(h0);
h1=h0.*exp(1j.*(2.*pi.*(n_h0-(length(n_h0)/2)).*0.2)); %(1/N)
h2=h0.*exp(1j.*(2.*pi.*((n_h0+1)-(length(n_h0)/2)).*0.4)); %(2/N)
h3=h0.*exp(1j.*(2.*pi.*((n_h0+2)-(length(n_h0)/2)).*0.6)); %(3/N)
%%%%%%%%%%%%%%%% Filter representation in time domain %%%%%%%%
figure (1)
subplot(3,1,1)
stem(n_h0,abs(h1),'r')
hold on
xlabel('n time samples (sec)');
ylabel('|F(n)|');
legend('|F_1(n)|');
subplot(3,1,2)
stem(n_h0+1,abs(h2),'b')
hold on
xlabel('n time samples (sec)');
ylabel('|F(n)|');
legend('|F_2(n)|');
subplot(3,1,3)
stem(n_h0+2,abs(h3),'g')
hold on
xlabel('n time samples (sec)');
ylabel('|F(n)|');
```

```
legend('|F_3(n)|');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX filter %%%%%%%%%%%%%%%%%%%%%%%%%%%%
hh1=remez(39,[1 2 3 5]/5,[1 1 0 0],[1 100]); hh=hh1;
gg1=hh.*exp(1j*2*pi*(0:39)*1/5);
gg2=hh.*exp(1j*2*pi*(0:39)*2/5);
gg3=hh.*exp(1j*2*pi*(0:39)*3/5);
%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP A in Fig. 2 and 3 %%%%%%%%%%%%%%%%%
x1=[]; x2=[]; x3=[];
for ii=1:nSym
x1=[x1 (exp((1j*2*pi*1).*n1(ii,:)))*s1(ii)];
x2=[x2 (exp((1j*2*pi*1).*n2(ii,:)))*s2(ii)];
x3=[x3 (exp((1j*2*pi*1).*n3(ii,:)))*s3(ii)];
end
figure(2)
subplot(3,1,1)
stem(n,abs(x1),'r')
hold on
xlabel('n time samples (sec)');
ylabel('|s(n)|');
subplot(3,1,2)
stem(n+1,abs(x2),'b')
hold on
xlabel('n time samples (sec)');
ylabel('|s(n)|');
subplot(3,1,3)
stem(n+2,abs(x3),'g')
hold on
xlabel('n time samples (sec)');
ylabel('|s(n)|');
%%%%%%%%%%%%%%%%%%%%% STEP B in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%
x1_up=upsample(x1,5);
x2_up=upsample(x2,5);
x3_up=upsample(x3,5);
n_up=1:length(x1_up);
figure(3)
stem(n_up,abs(x1_up),'r')
hold on
stem(n_up+1,abs(x2_up),'b')
hold on
stem(n_up+2,abs(x3_up),'g')
hold on
xlabel('n time samples (sec)');
ylabel('up-sampled |s(n)|');
legend('up-sampled |s_1(n)|','up-sampled |s_2(n)|','up-sampled |s_3(n)|');
%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP C in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%
u1=filter(h1,1,x1_up);
u2=filter(h2,1,x2_up);
u3=filter(h3,1,x3_up);
figure(4)
subplot(3,1,1)
stem(n_up,abs(u1),'r')
hold on
xlabel('n time samples (sec)');
ylabel('u(n)|');
legend('|u_1(n)|');
subplot(3,1,2)
stem(n_up+1,abs(u2),'b')
```

```
hold on
xlabel('n time samples (sec)');
ylabel('u(n)|');
legend('|u_2(n)|');
subplot(3,1,3)
stem(n_up+2,abs(u3),'g')
hold on
xlabel('n time samples (sec)');
ylabel('u(n)|');
legend('|u_3(n)|');
%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP D in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%
uu=u1+u2+u3;
figure(5)
stem(n_up,abs(uu),'b')
hold on
xlabel('n time samples (sec)');
ylabel('|x(n)|');
legend('|x(n)|=(|u_1(n)|,|u_2(n)|,|u_3(n)|)');
%%%%%%%%%%%%%%%%%%%%% HPA Model %%%%%%%%%%%%%%%%%%%%%%%%%%%%
backoff=10;
a_1=2; b_1=1; % model parameters
a_2=4; b_2=9; % model parameters
m=10^(backoff/20);%normalization
vv = zeros(1,length(uu))*(1.0+1i*1.0); % initialize output array
for j=1:length(uu)
ain = m*abs(uu(j));
a_in(j)=ain;
end
h=(mean((a_in)));
g=((2.*mean(a_in))./(1+ (mean(a_in))))^2;
c=(g/h); % slope
%%%%%%%%%%%%%%%%%% HPA predistortion model %%%%%%%%%%%%%%%%%%%%%%%
for j=1:length(uu)
B=(a_1+sqrt(a_1^2-(4*b_1*c^2*(abs(uu(j)))^2)))/((2*(abs(uu(j)))^2)*b_1*c);
A=-(a_2*((abs(uu(j)*B))^2))/(1+b_2*((abs(uu(j)*B))^2));
phase_dist=(A+((a_2*(abs(uu(j)*B))^2)/(1+b_2*((abs(uu(j)*B))^2))));
amp_dist=B*(a_1/(1+b_1*((abs(uu(j)*B))^2)));
uu_PD(j)=abs(uu(j))*exp(1j*angle(uu(j)))*B*exp(1j*A);
uu_HPA(j)=uu_PD(j)*(a_1/(1+(b_1*((abs(uu_PD(j)))^2))))*exp(1j*((a_2*((abs(uu_
PD(j)))^2))/(1+(b_2*((abs(uu_PD(j)))^2)))));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
y1=zeros(1,nSym); y2=zeros(1,nSym); y3=zeros(1,nSym);
for i=1:length(EsN0dB)
%%%%%%%%%%%%%%%%%%%%% STEP E in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%
r1=filter(gg1,1,uu_HPA);
r2=filter(gg2,1,uu_HPA);
r3=filter(gg3,1,uu_HPA);
r1_d=downsample(r1,5);
r2_d=downsample(r2,5);
r3_d=downsample(r3,5);
%%%%%%%%%%%precompensation to remove distortion due to filter %%%%%%%%%%%%
pd_a1=angle(r1_d)-angle(x1);  pd_a2=angle(r2_d)-angle(x2);  pd_a3=angle(r3_d)-
angle(x3);
 ad_a1=abs(r1_d)-abs(x1); ad_a2=abs(r2_d)-abs(x2); ad_a3=abs(r3_d)-abs(x3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% STEP F in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%
```

```
x1_d=((abs(r1_d)-ad_a1).*exp(1j.*(angle(r1_d)-pd_a1)));
x2_d=((abs(r2_d)-ad_a2).*exp(1j.*(angle(r2_d)-pd_a2)));
x3_d=((abs(r3_d)-ad_a3).*exp(1j.*(angle(r3_d)-pd_a3)));
end
%%%%%%%%%%%% Plot for STEP E in Fig. 2 and 3 %%%%%%%%%%%%%%%%
figure(6)
subplot(3,1,1)
stem(n_up,abs(r1),'r')
hold on
xlabel('n time samples (sec)');
ylabel('|v(n)|');
legend('|v_1(n)|');
subplot(3,1,2)
stem(n_up+1,abs(r2),'b')
hold on
xlabel('n time samples (sec)');
ylabel('|v(n)|');
legend('|v_2(n)|');
subplot(3,1,3)
stem(n_up+2,abs(r3),'g')
hold on
xlabel('n time samples (sec)');
ylabel('|v(n)|');
legend('|v_3(n)|');
%%%%%%%%%%%% Plot for STEP F in Fig. 2 and 3 %%%%%%%%%%%%%%%%%%%%%%%
figure(7)
subplot(3,1,1)
stem(n,abs(x1_d),'r')
hold on
xlabel('n time samples (sec)');
ylabel('|y(n)|');
legend('|y_1(n)|');
subplot(3,1,2)
stem(n+1,abs(x2_d),'b')
hold on
xlabel('n time samples (sec)');
ylabel('|y(n)|');
legend('|y_2(n)|');
subplot(3,1,3)
stem(n+2,abs(x3_d),'g')
hold on
xlabel('n time samples (sec)');
ylabel('|y(n)|');
legend('|y_3(n)|');
```

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AFRL	Air Force Research Laboratory
AM/AM	Amplitude-to-amplitude
AM/PM	Amplitude-to-phase
AWGN	Additive white Gaussian noise
BER	Bit error rate
BF	Beamforming
D/A	Digital-to-Analog
dB	Decibel
dBW	Decibel Watt
DF	Decode-and forward
DFT	Discrete Fourier Transform
FD	Frequency-division
FDM	Frequency-division multiplexed
GEO	Geostationary Earth Orbit
HPA	High Power Amplifier
IM	Intermodulation
LUT	Look-up table
MAU	Multiple access user
MIMO	Multiple-input Multiple-output
PAA	Phase array antenna
PD	Predistorter
RF	Radio frequency
RX	Receiver
RHC	Right-hand Circular
SAT-COM	Satellite Communication
SINR	Signal-to-interference-plus-noise Ratio
TX	Transmitter
TWTA	Travelling-wave tube amplifier

DISTRIBUTION LIST

DTIC/OCP 8725 John J. Kingman Rd, Suite 0944 Ft Belvoir, VA 22060-6218	1 cy
AFRL/RVIL Kirtland AFB, NM 87117-5776	1 cy
Official Record Copy AFRL/RVB/Dr. Khanh Pham	1 cy

This page is intentionally left blank.