Lime's Open Source Adaptive Digital Predistortion for High Power Amplifiers

Table of Contents

1. Introduction

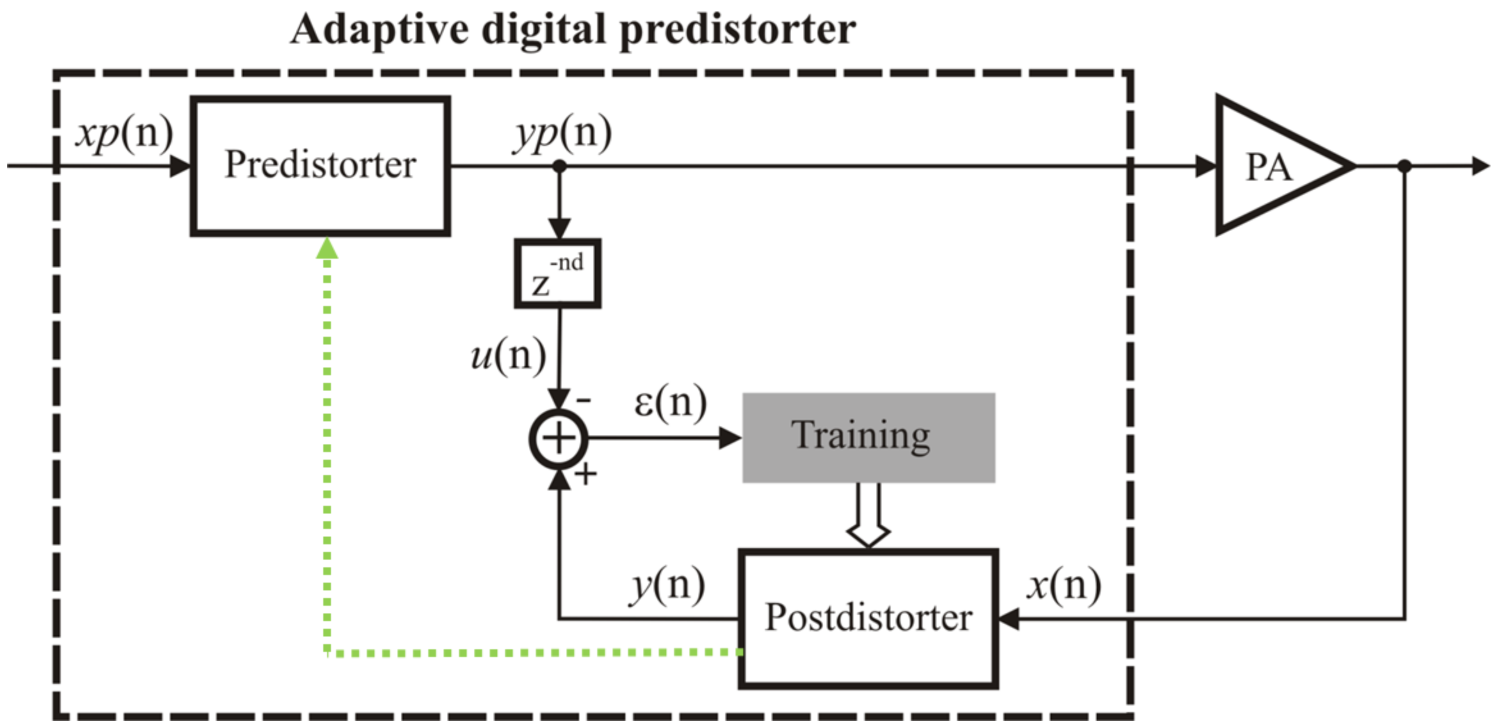
Motivation for PA Linearization

- Linearization improves PA power efficiency and reduces running cost of the wireless infrastructure
- Linear power amplifier is more spectrum efficient
  - ACPR is reduced, EVM is improved,
  - support for multicarrier signals and high bandwidth,
  - allows for sophisticated modulation schemes.
- Lime solution for PA linearization is based on adaptive digital predistortion (ADPD)
- The complete solution is Open Source and is available on GitHub[1] and the LimeNET app store.

1 https://github.com/myriadrf/ADPD_LimeSDR-QPCIe_USB3

2. ADPD Model

Indirect Learning Architecture



- Delay line compensates ADPD loop (*yp(n)* to *x(n)*) delay
- Postdistorter is trained to be inverse of power amplifier
- Predistorter is simple copy of postdistorter
- When converged (ε(*n*)=0) *yp(n)*=*y(n)* => *x(n)*=*xp(n)*

Complex Valued Memory Polynomial

- LimeADPD algorithm is based on modeling nonlinear system (PA and its inverse in this case) by complex valued memory polynomials which are version of Volterra series

Input: $\mathbf{x}(n) = x_I(n) + \mathbf{j}x_Q(n)$

Coefficients: $\mathbf{a}_{ij} = a_{ij} + \mathbf{j}b_{ij}$

Envelop

for Complete Form : $e(n) = \sqrt{x_I(n)^2 + x_Q(n)^2}$

for Even Order Terms Only: $e(n) = x_I(n)^2 + x_Q(n)^2$

- Even Order Terms Only envelop is used since it is simpler to calculate and provides even better results

Output:

$$\mathbf{y}(n) = y_I(n) + \mathbf{j}y_Q(n)$$

$$\mathbf{y}(n) = \sum_{i=0}^{N}\sum_{j=0}^{M}\mathbf{a}_{ij}\,\mathbf{x}(n-i)e(n-i)^j$$

- Complex valued memory polynomial takes into account both system memory effects as well as the system nonlinearity; in equations N is memory length and M is nonlinearity order

ADPD Equations

Predistorter:

$$\mathbf{yp}(n) = \sum_{i=0}^{N}\sum_{j=0}^{M} \mathbf{a}_{ij}\, \mathbf{xp}(n-i) ep(n-i)^{j}$$

$$\mathbf{xp}(n) = xp_I(n) + \mathbf{j}\, xp_Q(n)$$

$$ep(n) = xp_I(n)^2 + xp_Q(n)^2$$

Postdistorter:

$$\mathbf{y}(n) = \sum_{i=0}^{N}\sum_{j=0}^{M} \mathbf{a}_{ij}\, \mathbf{x}(n-i) e(n-i)^{j}$$

$$\mathbf{x}(n) = x_I(n) + \mathbf{j}\, x_Q(n)$$

$$e(n) = x_I(n)^2 + x_Q(n)^2$$

Predistorter and postdistorter share the same set of complex coefficients a$_{ij}$.

Delay Line:

$$\mathbf{u}(n) = \mathbf{yp}(n - nd)$$

Training Algorithm

ADPD training algorithm alters complex valued memory polynomial coefficients $\boldsymbol{a_{ij}}$ in order to minimize the difference between ***yp(n)*** and ***y(n)***, ignoring the delay and gain difference between the two signals.

Instantaneous Error:

$$\varepsilon(n) = \sqrt{\left(u_I(n) - y_I(n)\right)^2 + \left(u_Q(n) - y_Q(n)\right)^2}$$

Training is based on minimising Recursive Least Square (RLS) ***E(n)*** error

RLS Error:

$$E(n) = \frac{1}{2}\sum_{m=0}^{n} \lambda^{n-m}\varepsilon(m)^2, \ \lambda < 1$$

Training Algorithm

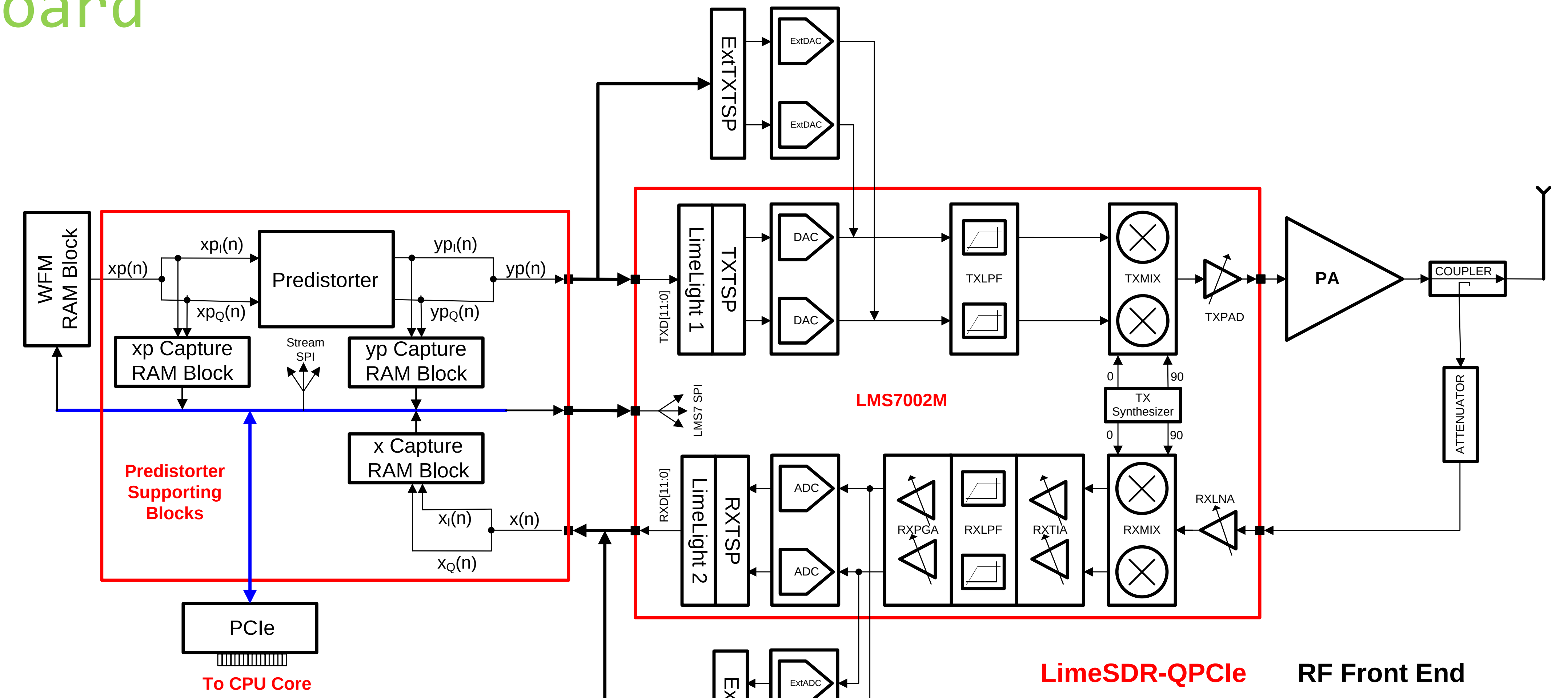
- Predistorter and postdistorter are nonlinear adaptive filters
  - There are two choices. To use
    - Complete form or
    - "Even order terms only" which halves the hardware complexity
      Adopted for implementation to save hardware
- RLS error $E(n)$ is minimized by solving linear system of equations:

$$\frac{\partial E(n)}{\partial a_{kl}} = 0, \quad \frac{\partial E(n)}{\partial b_{kl}} = 0; \; k = 0,1,\ldots,N; \; l = 0,1,\ldots,M$$

- Equation solver candidates
  - **LU Decomposition**
    - Can only be implemented as DSP/software solution
    - It is very fast as compared to other candidates
      Adopted for implementation to get faster adaptation and tracking of the ADPD loop
  - **Gauss – Seidel**
    - Iterative technique
    - Can be implemented in hardware but requires divider module
  - **Gradient Descent**
    - Iterative technique
    - Can be implemented in hardware. There is no division

3. Implementation

ADPD Implementation Based on LimeSDR-QPCIe board



Platform Description

- For the development or demo, test waveform is uploaded and played from WFM RAM Block, implemented using Altera Cyclon V FPGA resources.
- More importantly, the same FPGA also implements predistorter. Initially, predistoter is bypassed i.e. $yp_I = xp_I$, $yp_Q = xp_Q$.
- Predistorter has provision for SPI in order to update the coefficients during the training.
- Signals *xp*, *yp* and *x* are captured using Data Capture RAM Blocks, implemented also using FPGA resources. Captured data is transferred to CPU (Intel Motherboard) Core via PCIe interface.
- FPGA implements PCIe and other glue logic required to interconnect LimeSDR-QPCIe on board components including two LMS7002M ICs to the CPU Core.
- CPU implements postdistorter block, delay line and the rest of training algorithm. After each adaptation step, CPU updates predistorter coefficients via PCIe/ LimeSDR-QPCIe SPI.
- PC/GUI implements graphical display for demo and debugging purposes. GUI is capable to show important ADPD signals in FFT (frequency), time and constellation (I vs Q) domains.

Platform Description

- In the real applications, WFM and xp Capture RAM blocks are not required. The algorithm needs only *yp* and *x*. CPU Core performs both ADPD adaptation, as explained above, and base band (BB) digital modem functions which are application specific, LTE for example.
- Frequency conversion from BB to RF is performed by LMS7002M transmitter chains. Frequency down conversion from RF to BB is implemented by only one LMS7002M receive chain dedicated to ADPD, i.e. one receiver of the available RF RX chains is allocated as ADPD monitoring path.
- In case of MIMO applications, the same ADPD monitoring path is used as time sharing resource to linearize multiple PAs which saves power consumption as well as on board RF resources.
- LimeSDR-QPCIe board offers two options: to use LMS7002M on chip DACs/ADCs or external ones. On chip data converters are 12-bit devices while external ones are 14-bit.
- 14-bit data converters are more suitable for ADPD if one wants to cancel IMD3 and IMD5 as well as to support wide band modulation schemes. If external data converters are used, LMS7002M on chip Transceiver Signal Processing blocks (TXTSP and RXTSP) are bypassed. In this case, minimum functionality required for ADPD to function is implemented by external counterparts, shown as ExtTXTSP and ExtRXTSP which are implemented by FPGA.

4. Measured Results

Measured Results

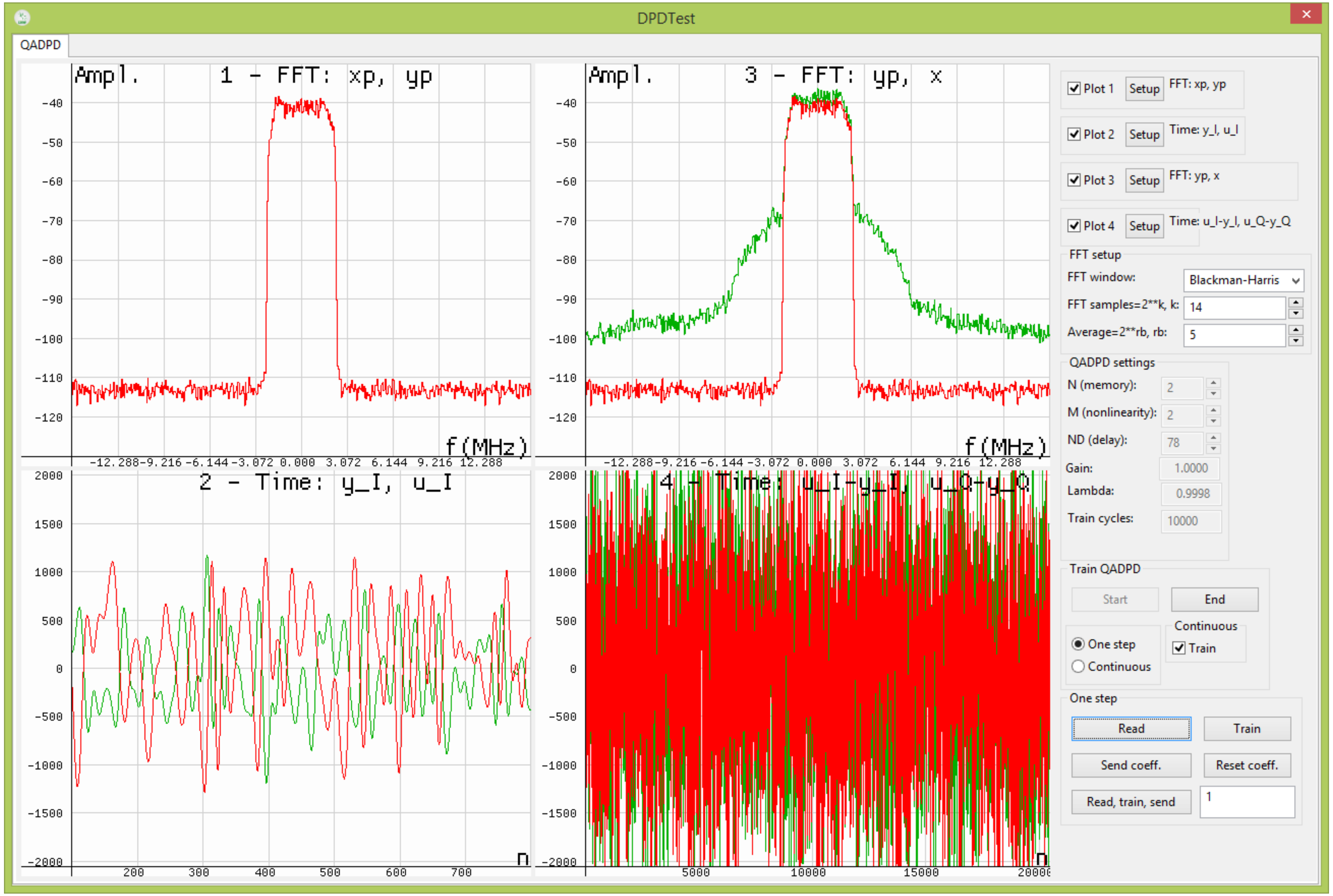
- Before implementation and measurements, ADPD algorithm has been thoroughly simulated.
- ADPD performance has been measured and the results for three cases
- Test Case 1 - Power amplifier: Maxim Integrated MAX2612
  - Moderate output power amplifier device Maxim Integrated MAX2612. Psat ~ 19dBm.
  - single carrier WCDMA Test Model 1
  - RF centre frequency 2.14GHz.
- Test Case 2: Power amplifier Class-J GaN HEMPT
  - MAX2612 is used as preamplifier stage followed by Class-J GaN HEMPT amplifier. GaN amplifier drain efficiency ~ 70%. Psat ~ 40dBm.
  - single carrier WCDMA Test Model 1
  - RF centre frequency 1.5GHz.
- Test Case 3: LMSP10C 2.1–2.6 GHz 10 W Power Amplifier Module
  - 20MHz LTE signal
  - RF centre frequency 2.675GHz.

Test Case 1: Maxim Integrated MAX2612 PA

- Power amplifier: Maxim Integrated MAX2612
  - Moderate output power amplifier device Maxim Integrated MAX2612.
  - Psat ~ 19dBm.
  - Test signal: single carrier WCDMA Test Model 1

- ADPD Parameters
  - Nonlinearity order: M=3
  - Memory order: N=3

ADPD Signals Before Training

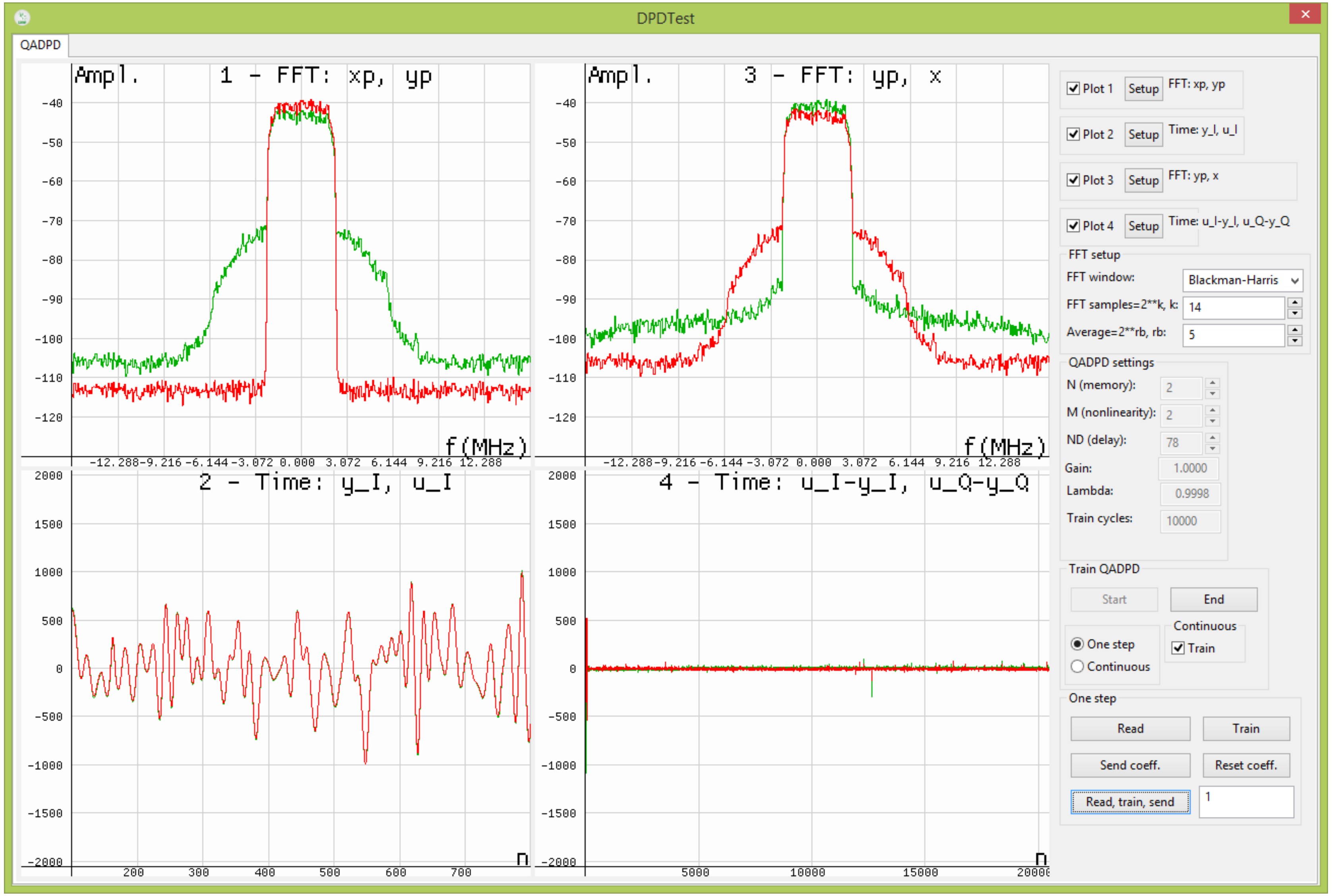


Predistorter signals *yp* and *xp* are equal (plot 1).

x as a measure of PA output is distorted (plot 3).

Waveforms *y* and *u* are very different (plot 2) which results in huge error (plot 4) which ADPD has to minimize.

ADPD Signals After Training



Signal *yp* (plot 1) is predistorted in order to cancel PA distortion components.

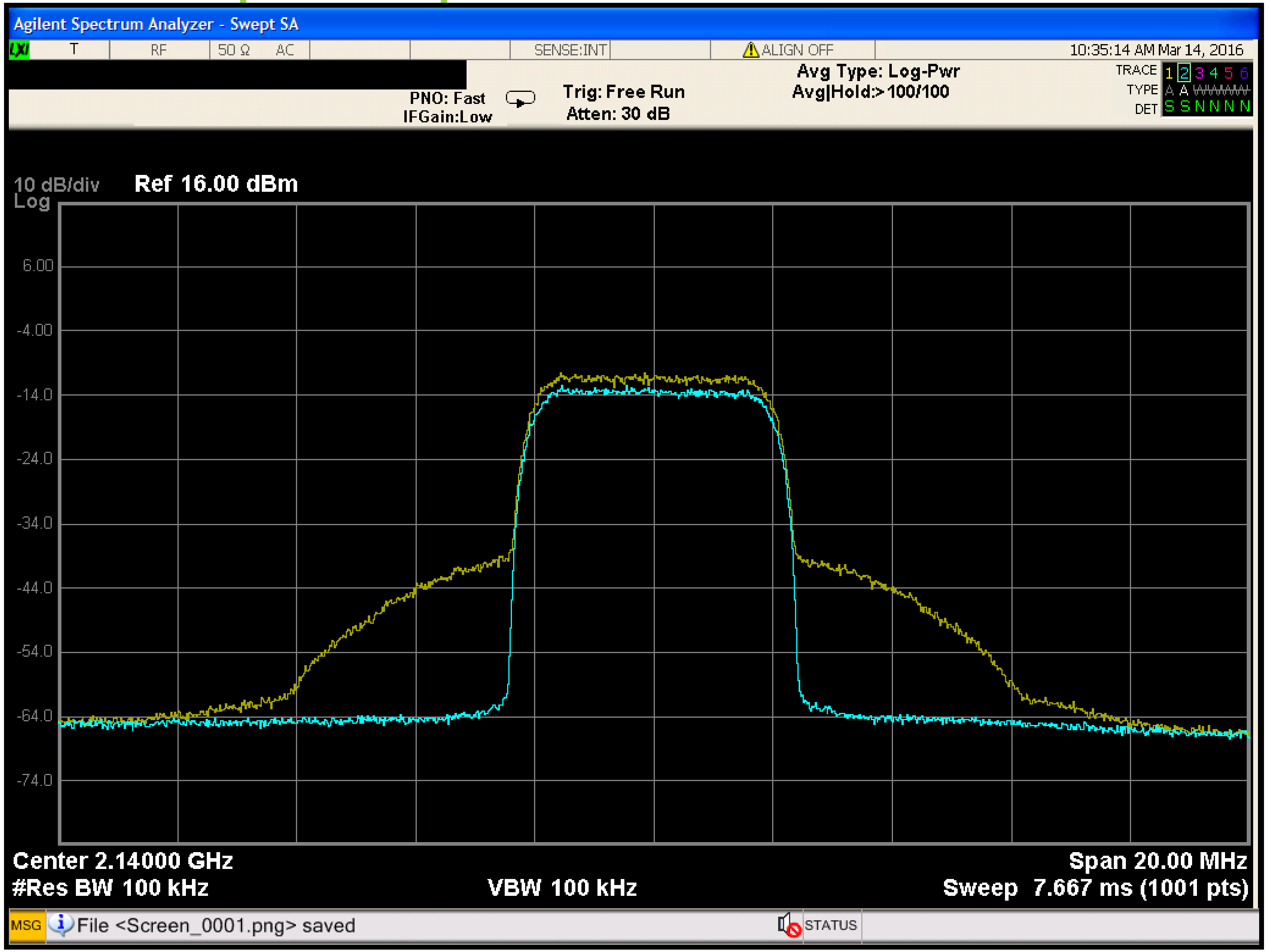
x as a measure of PA output is now linearized (plot 3).

Excellent match between *y* and *u* waveforms in both time and amplitude scale (plot 2).

ADPD error (plot 4) is minimized.

Improvement in PA linearization can be seen by comparing *yp* and *x* spectra of plot 3.

PA Output Spectrum

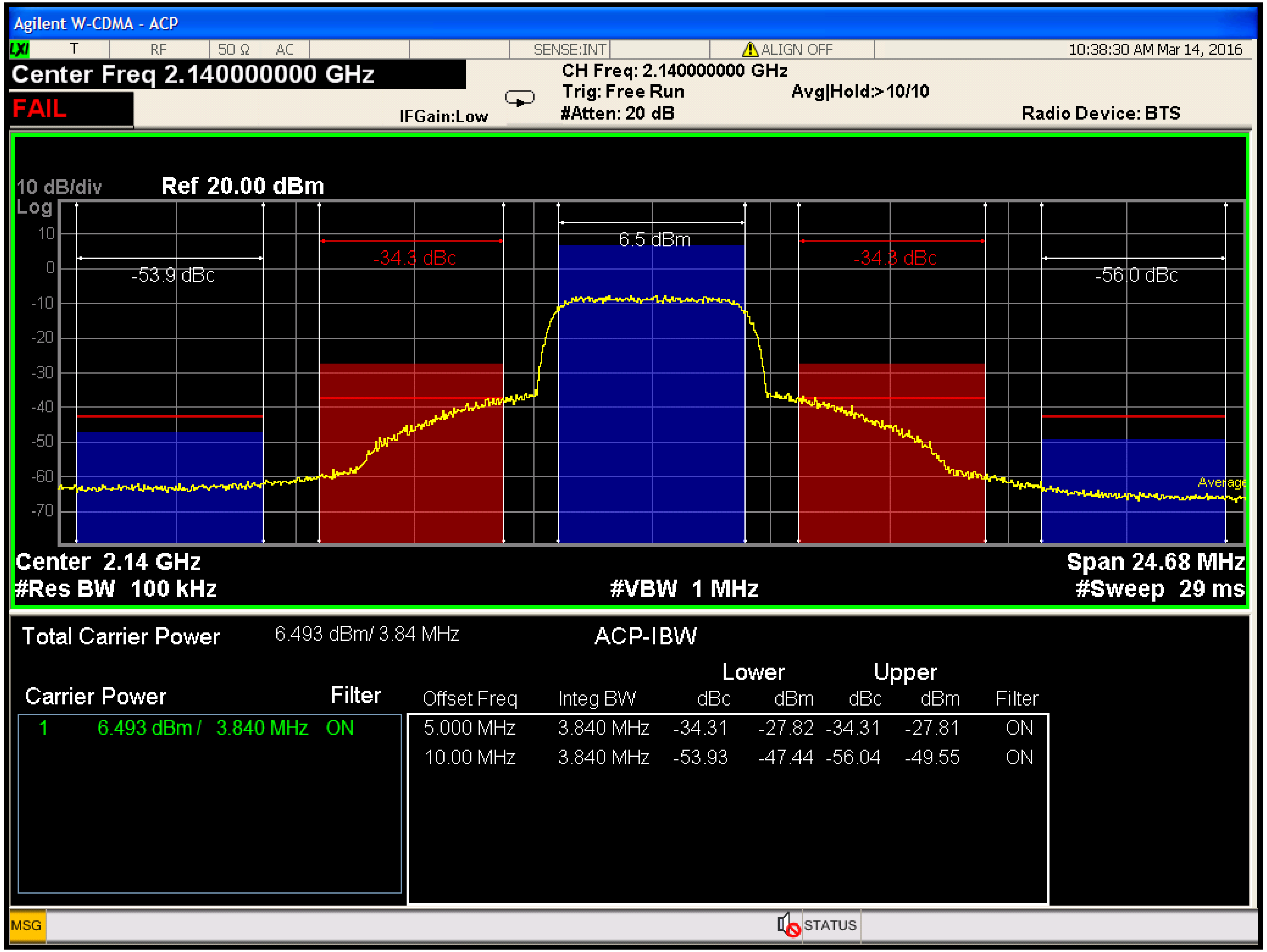


No ADPD – yellow trace

Width ADPD – blue trace

IMD3 and IMD5 distortion components at the PA output reduced by almost 20 dB.

ACPR Without ADPD

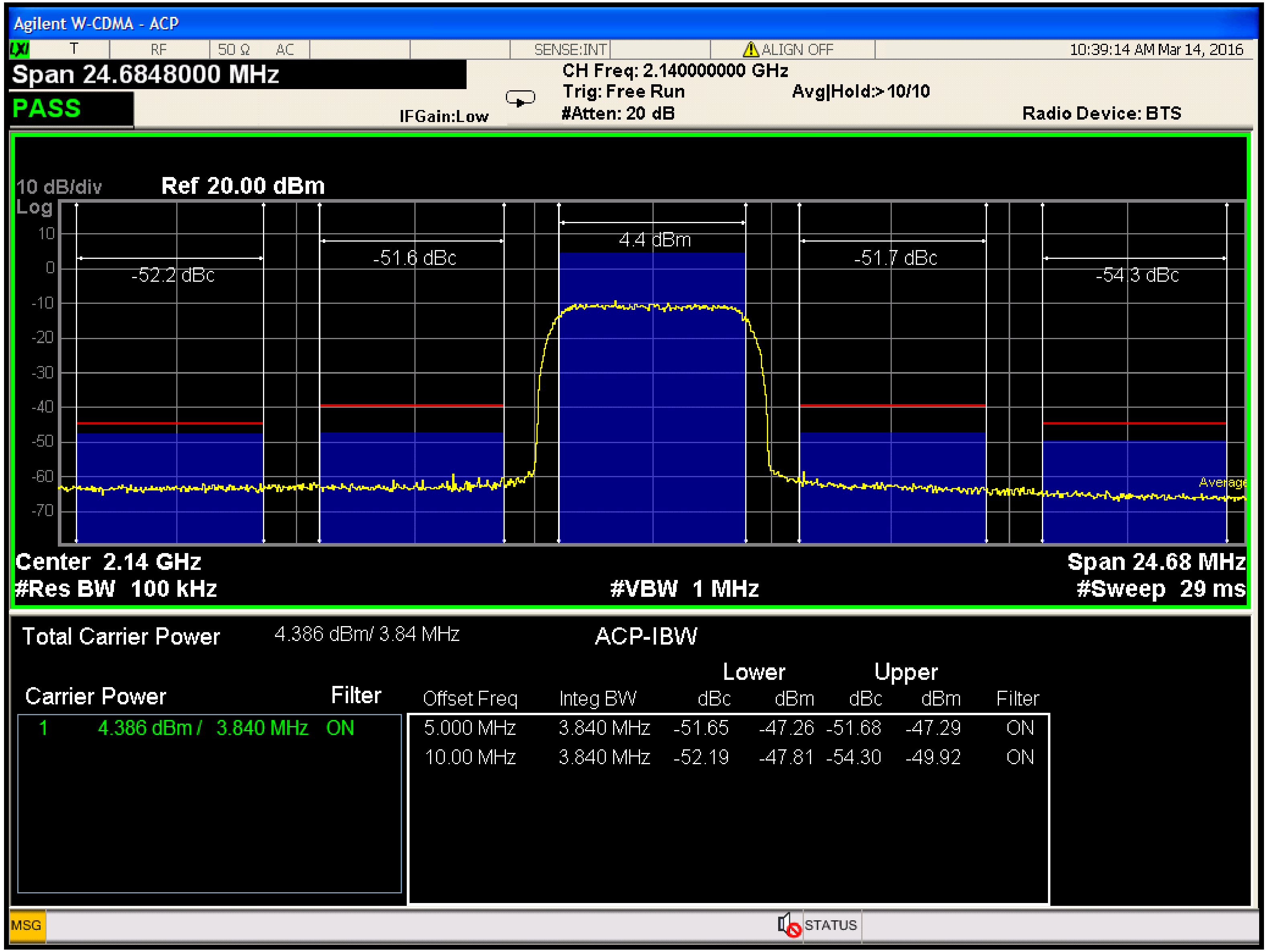


Offset Freq	Integ BW	Lower dBc	Lower dBm	Upper dBc	Upper dBm	Filter
5.000 MHz	3.840 MHz	-34.31	-27.82	-34.31	-27.81	ON
10.00 MHz	3.840 MHz	-53.93	-47.44	-56.04	-49.55	ON

ACPR1 = -34 dBc

ACPR2 = -54 dBc

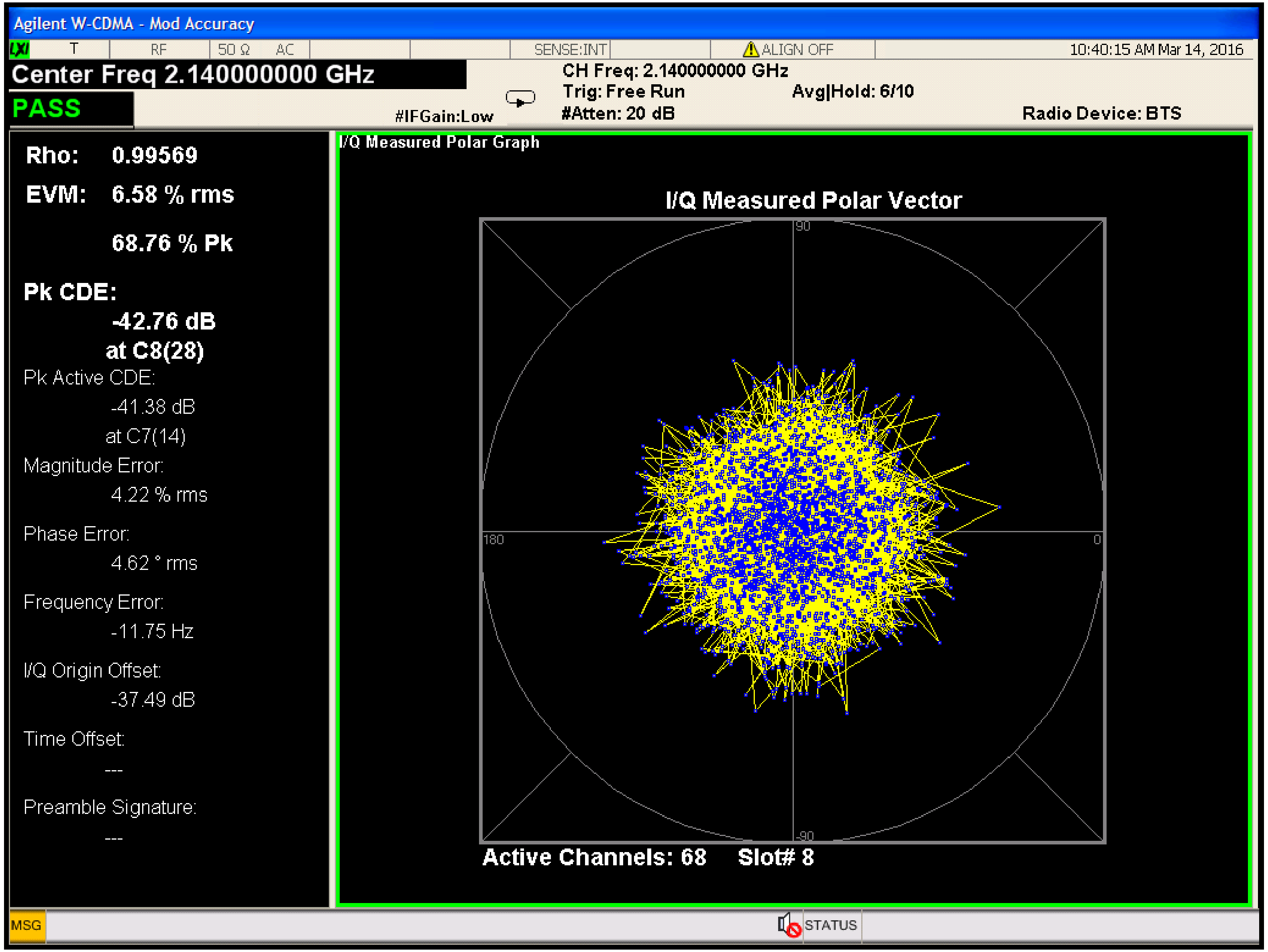
ACPR With ADPD



ACPR1 = -51 dBc

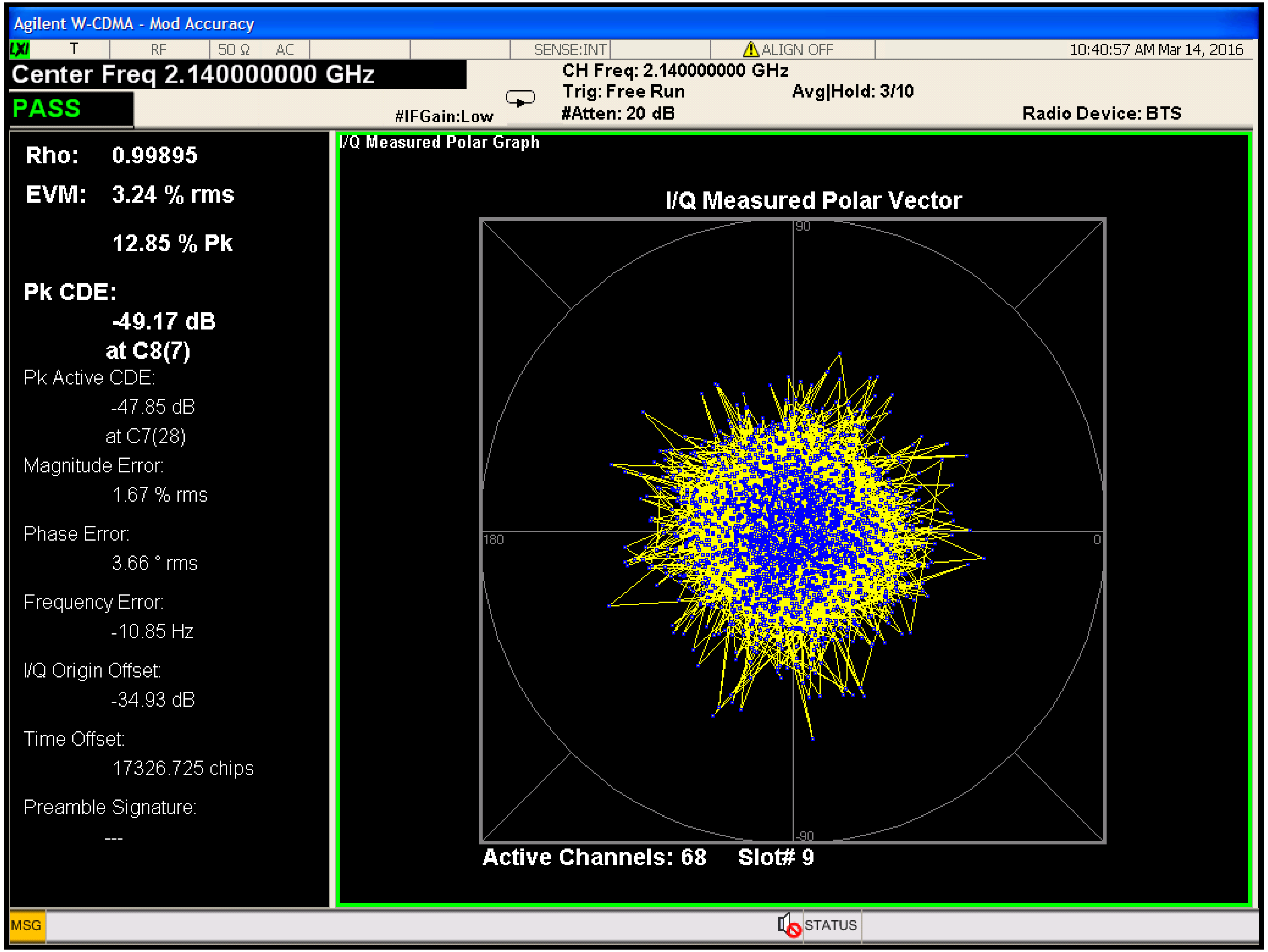
ACPR2 = -52 dBc

EVM Without ADPD



EVM = 6.58%

EVM With ADPD



EVM = 3.24%

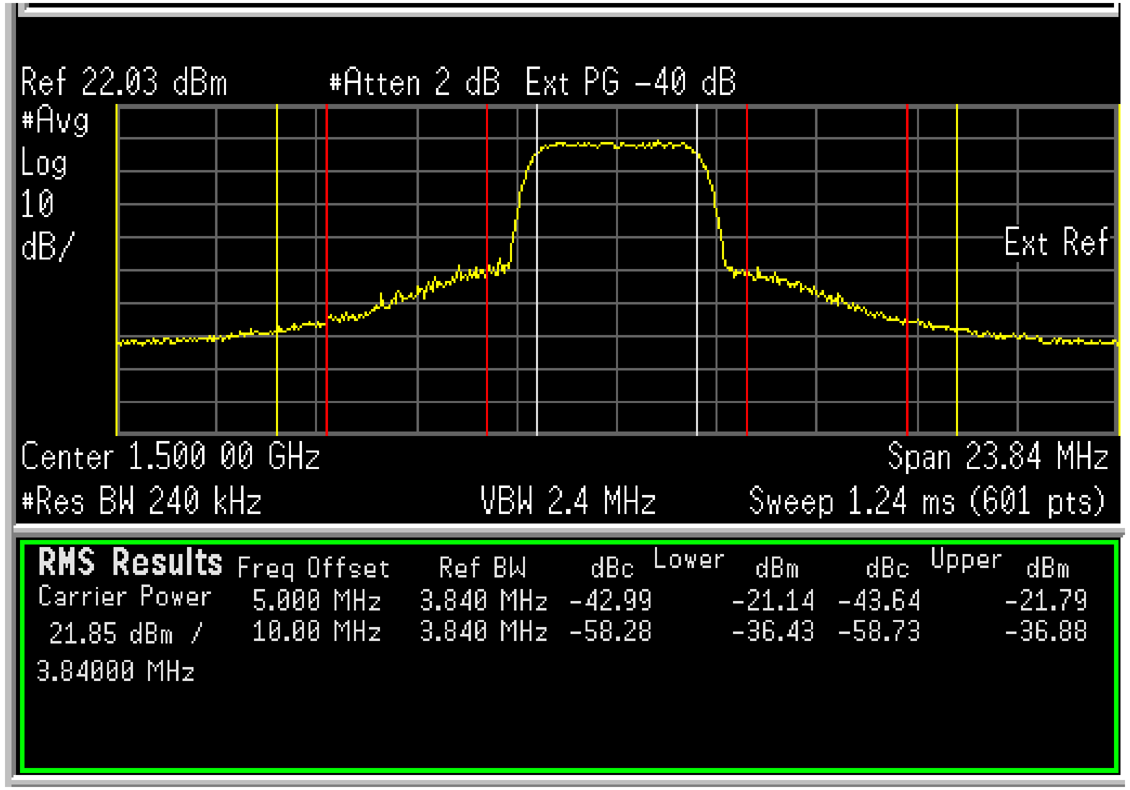
Test Case 2: Class-J GaN HEMPT Amplifier

- Power amplifier: Maxim Integrated MAX2612 +Class-J GaN HEMPT Amplifier
  - MAX2612 is used as preamplifier stage followed by Class-J GaN HEMPT amplifier.
  - GaN amplifier drain efficiency ~ 70%.
  - Psat ~ 40dBm.

- Test signal: single carrier WCDMA Test Model 1

- ADPD Parameters
  - Nonlinearity order: M=3
  - Memory order: N=3

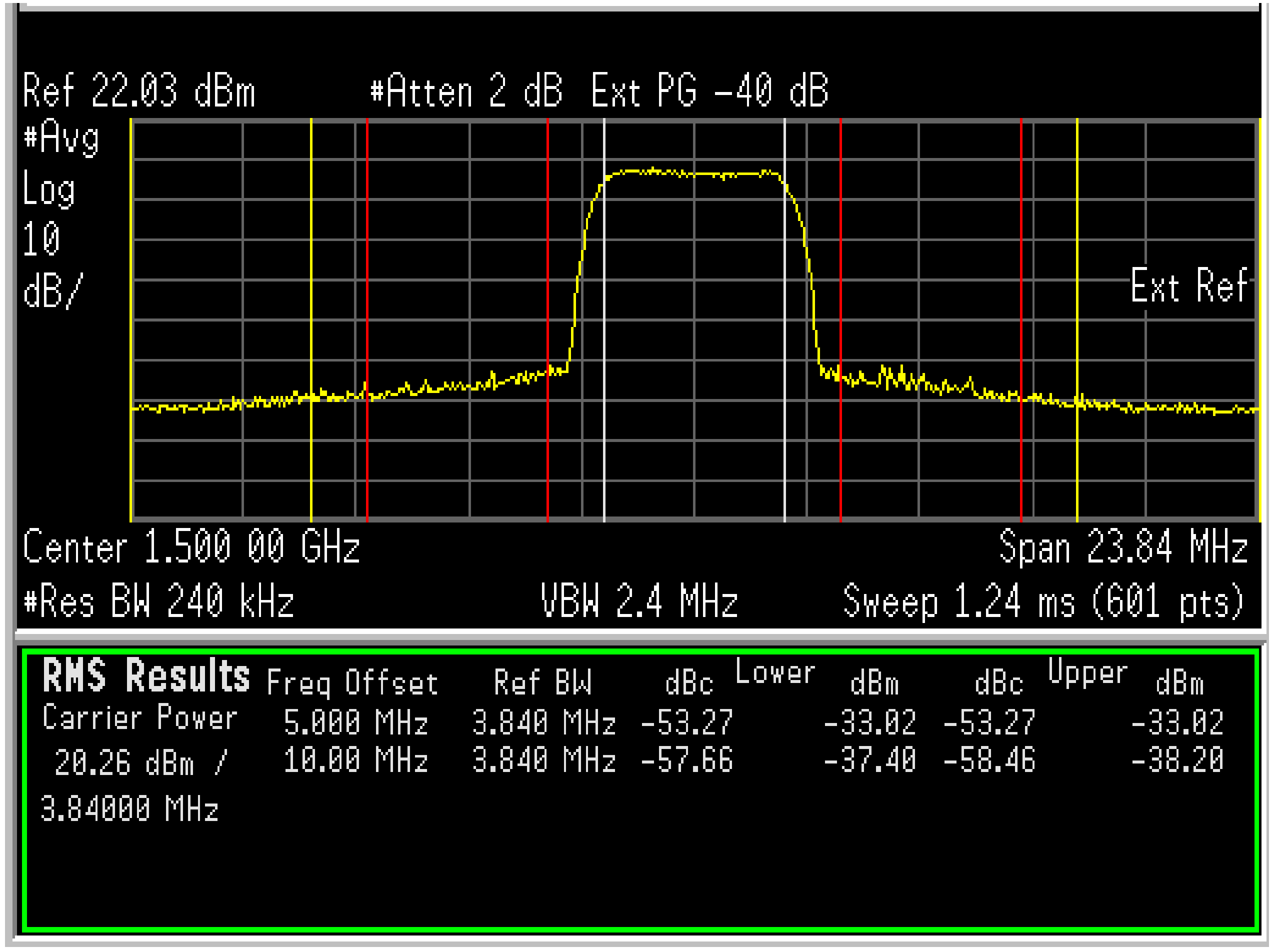
ACPR Without ADPD



No ADPD – yellow trace

ACPR without ADPD.
ACPR1 = -43 dBc, ACPR2 = -58dBc

ACPR With ADPD



With ADPD – yellow trace

ACPR with ADPD.
ACPR1 = -53 dBc, ACPR2 = -58dBc.

Lime microsystems

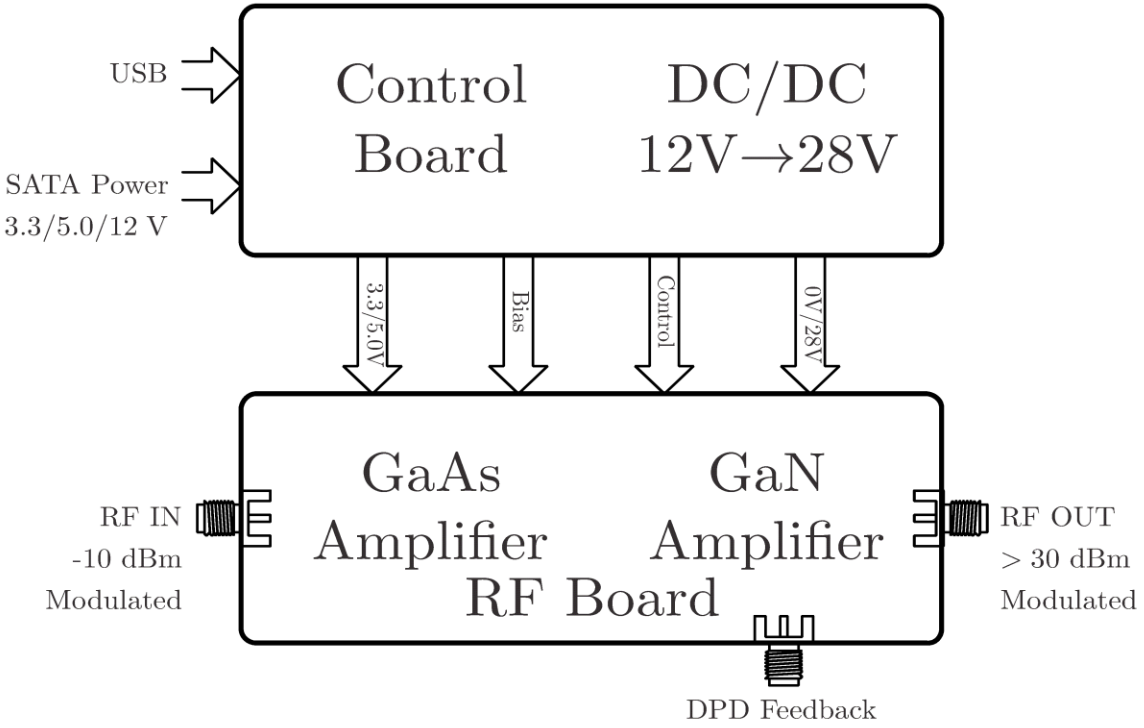
Test Case 3: LMSP10C Amplifier Module

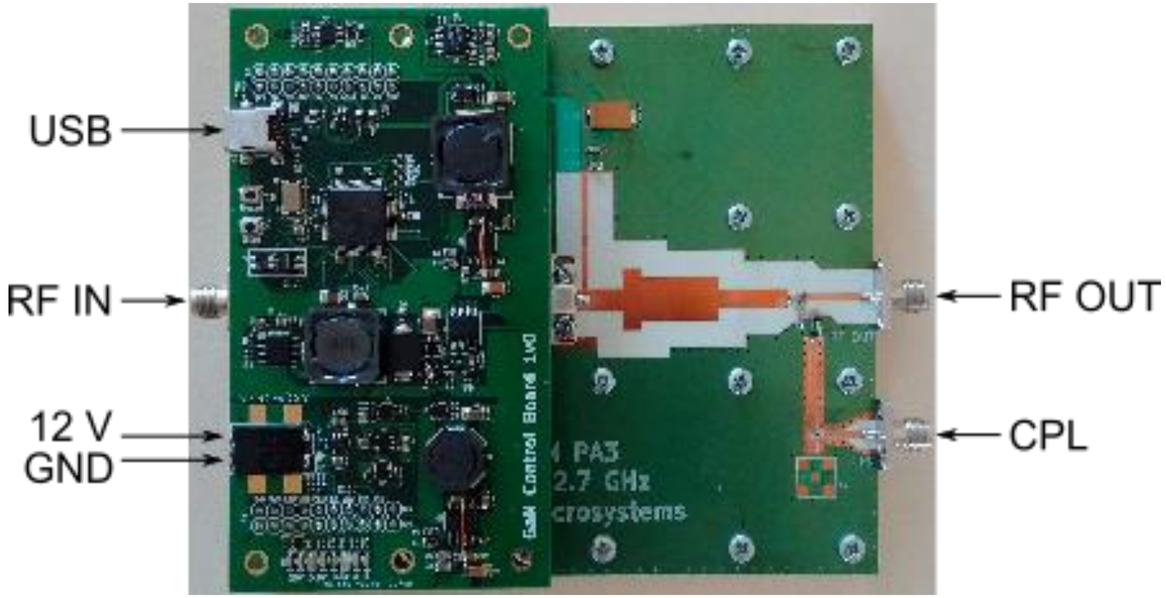
- Power amplifier: 2.1–2.6 GHz 10 W Power Amplifier
  - 2.1–2.6 GHz 10 Watts CW saturated output power
  - 30 dBm output modulated power

- Used test signal: 2x2 MIMO 20MHz LTE signal and
  2x2 MIMO 10MHz LTE signal

- ADPD parameters
  - Nonlinearity order: M=4
  - Memory order: N=2

LMSP10C Amplifier Module





Module dimensions are 100 x 115 x 50 mm

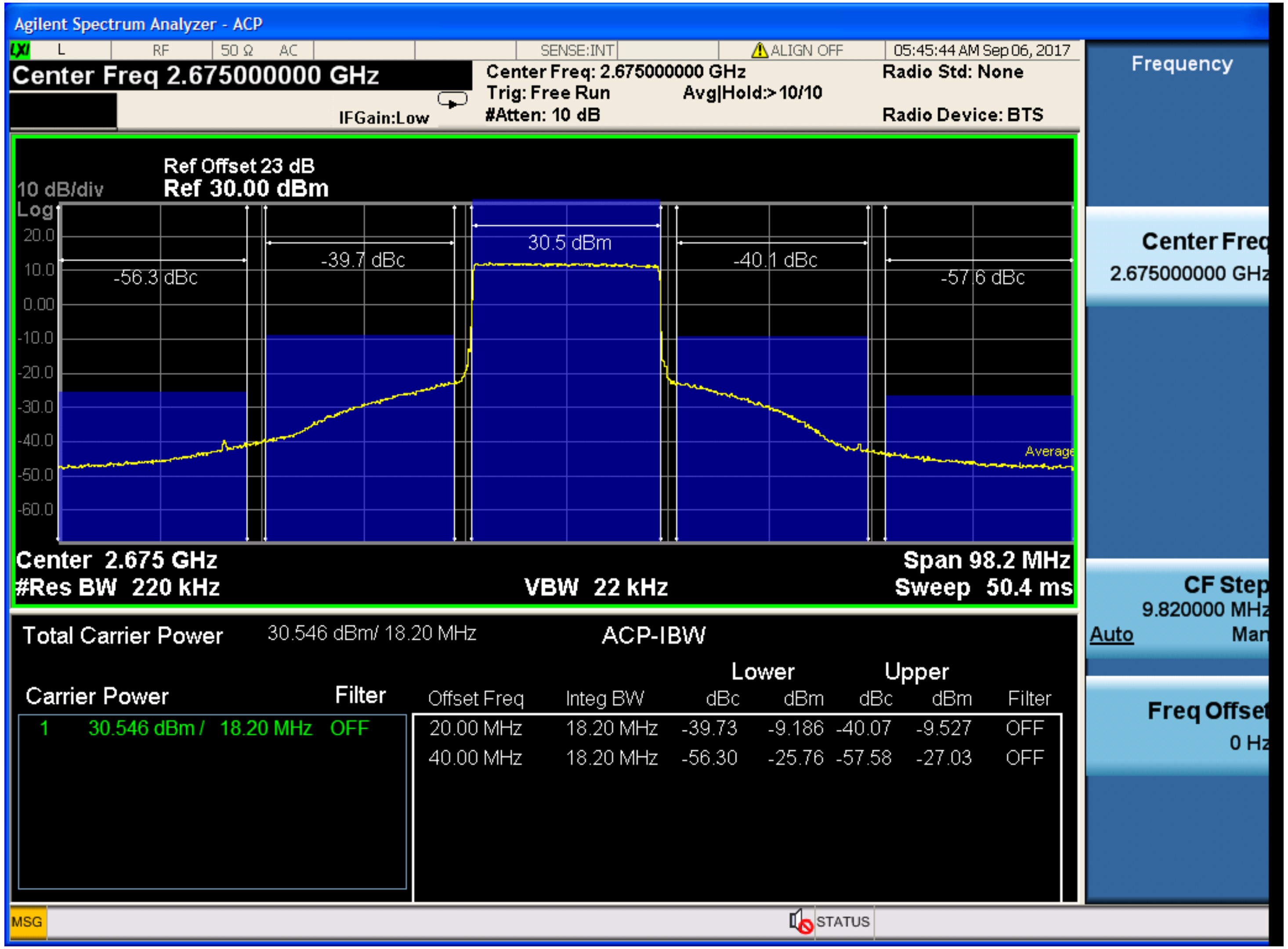
- Power amplifier: 2.1–2.6 GHz 10 W Power Amplifier
  - 2.1–2.6 GHz 10 W CW saturated output power
  - Pre-driver is based on the MGA-22003, GaAs E-pHEMT
  - Driver is based on the CGH40010, 10 W, DC – 6 GHz, RF Power GaN HEMT transistor

LMSP10C Amplifier Module

Parameter	Min.	Typ.	Max.	Unit	Condition/Comment
Frequency Range	2.1		2.6	GHz	3 dB Bandwidth
Gain	45			dB	
Output 1 dB Compression Point	**38**			dBm	
Output IP3	45			dBm	
Return Loss – Input	10			dB	
Return Loss – Output	10			dB	
Saturated Power		40		dBm	
Drain Efficiency			45	%	Maximum value with CW signal
Supply	10	12	20	V	
Supply current	250		500	mA	at 30 dBm 20MHz LTE modulated

LMSP10C Power Amplifier Module General Specifications

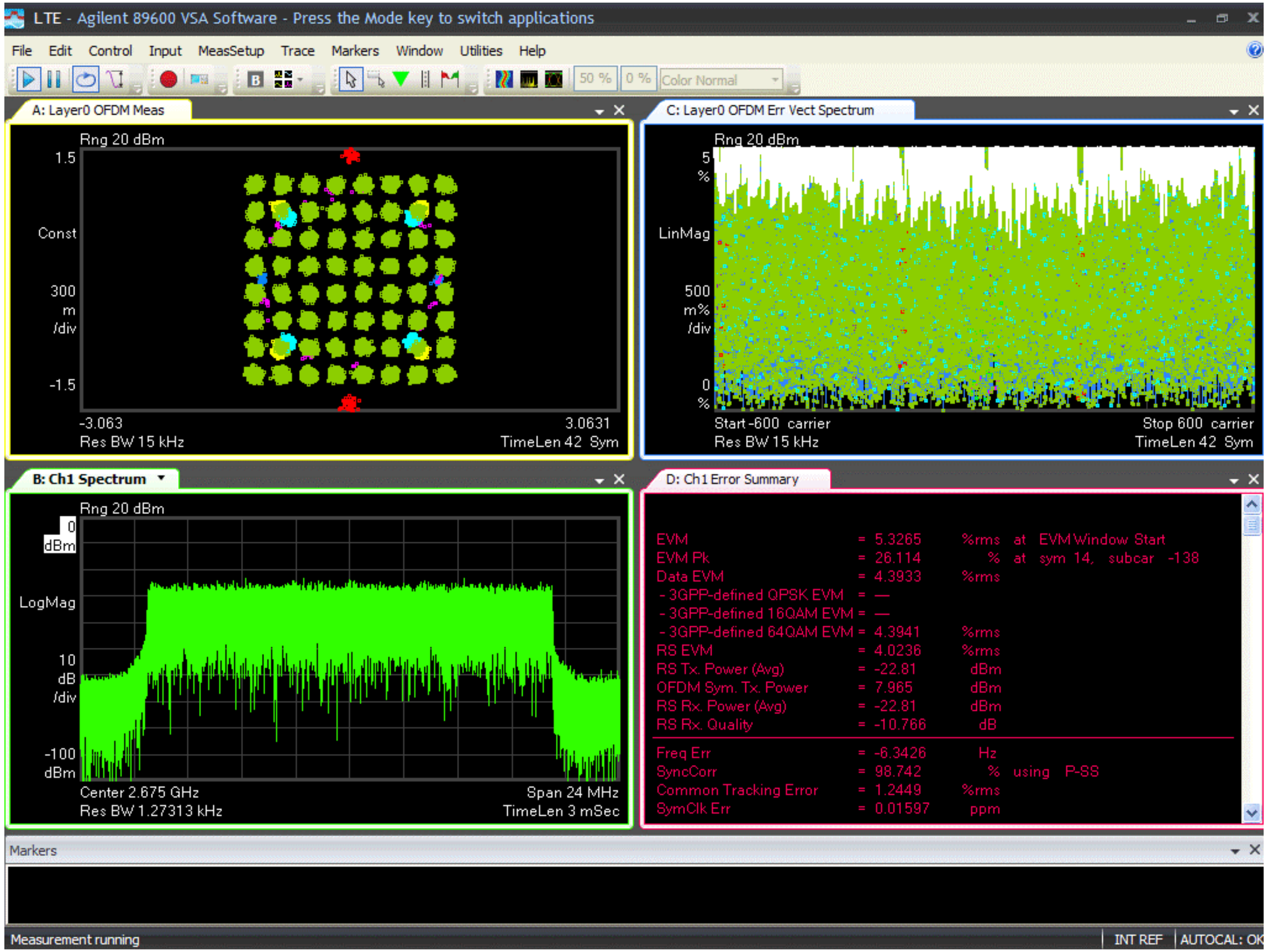
LMSP10C: 2x2 MIMO 20MHz LTE Case



ACPR without ADPD.

ACPR1 = -39.7 dBc, ACPR2 = -56dBc

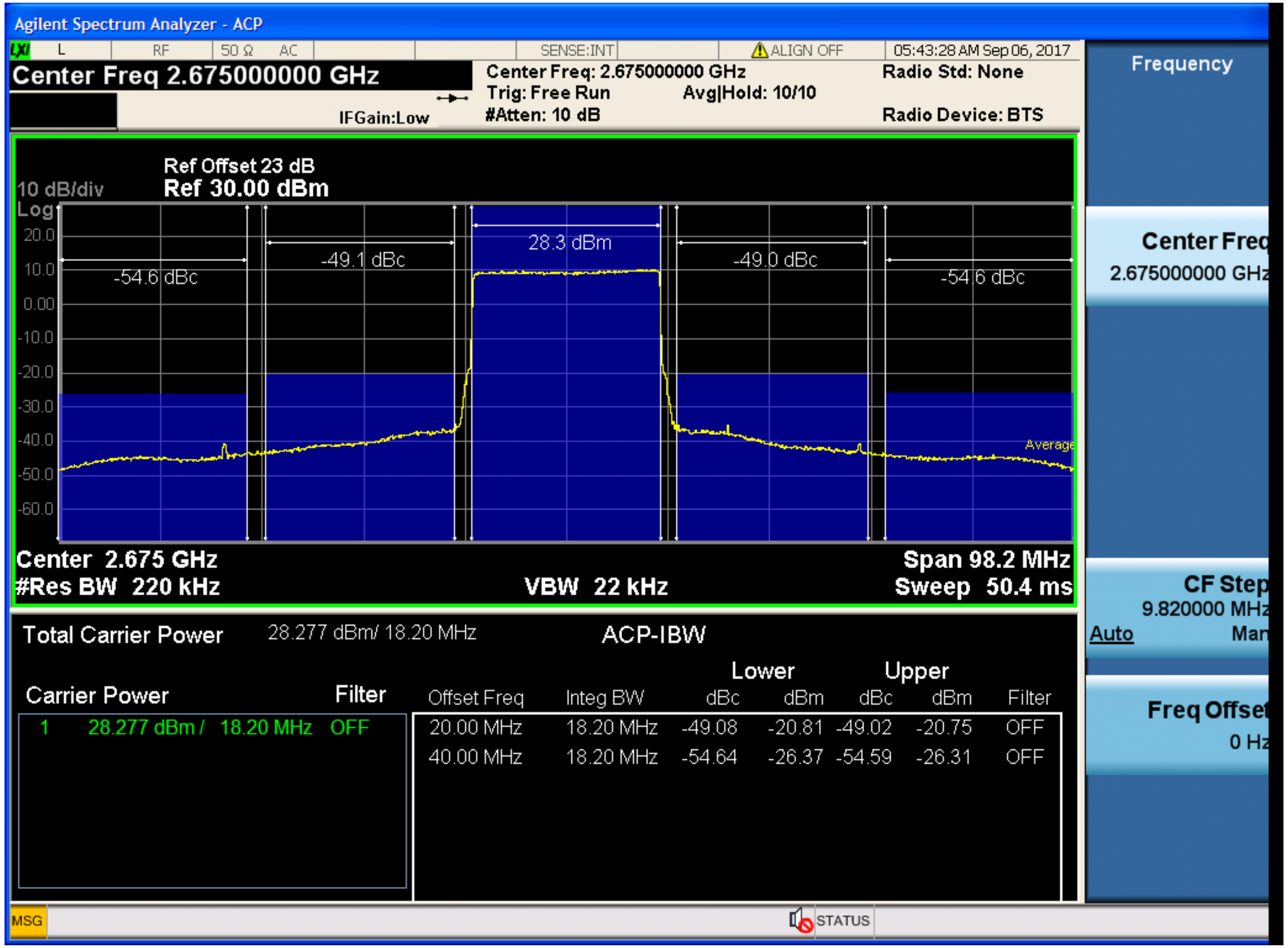
LMSP10C: 2x2 MIMO 20MHz LTE Case



EVM without ADPD

EVM = 5.33%

LMSP10C: 2x2 MIMO 20MHz LTE Case

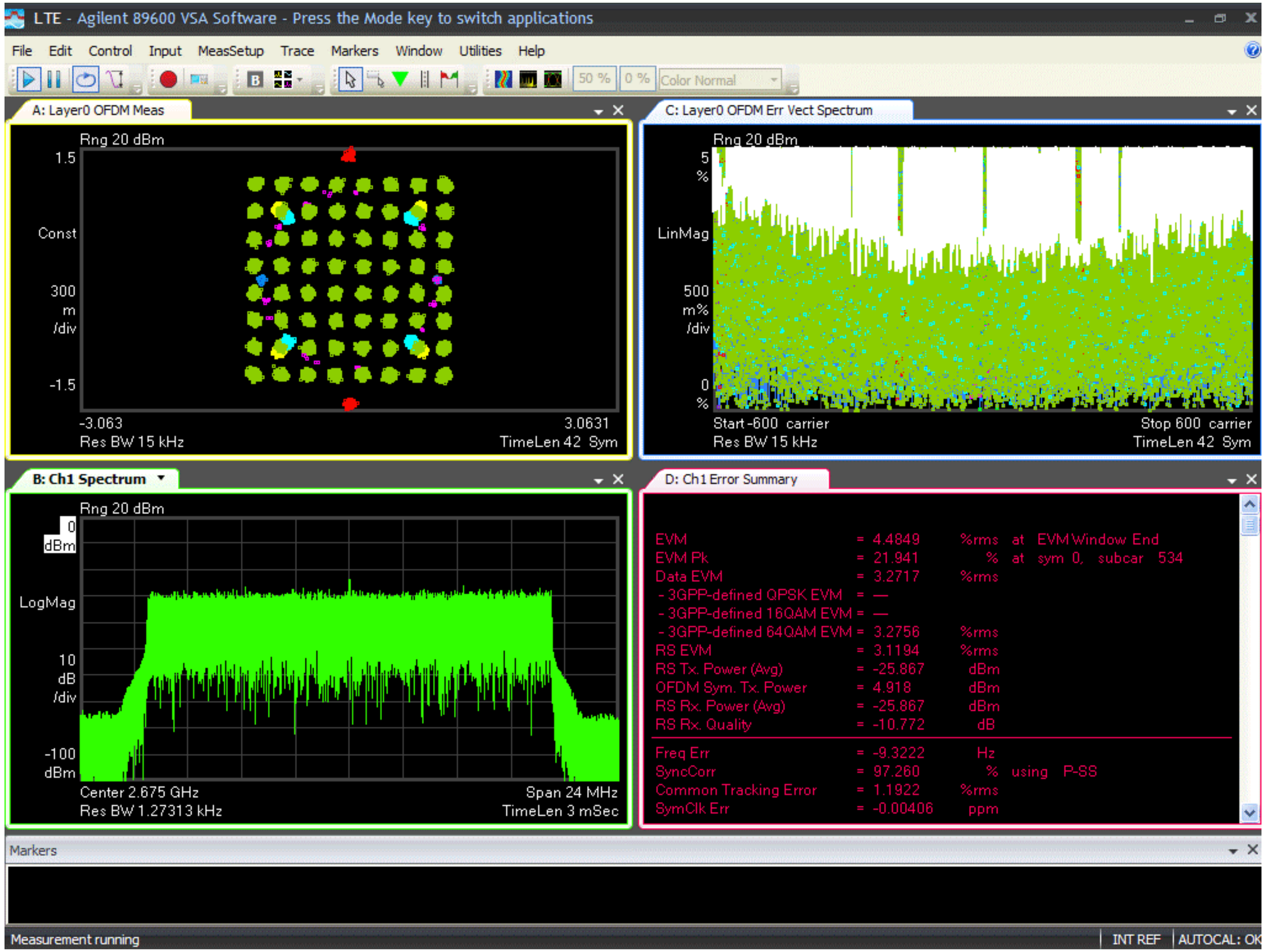


Offset Freq	Integ BW	Lower dBc	Lower dBm	Upper dBc	Upper dBm	Filter
20.00 MHz	18.20 MHz	-49.08	-20.81	-49.02	-20.75	OFF
40.00 MHz	18.20 MHz	-54.64	-26.37	-54.59	-26.31	OFF

ACPR with ADPD.
ACPR1 = -49.1 dBc, ACPR2 = -54.6dBc.

Lime microsystems

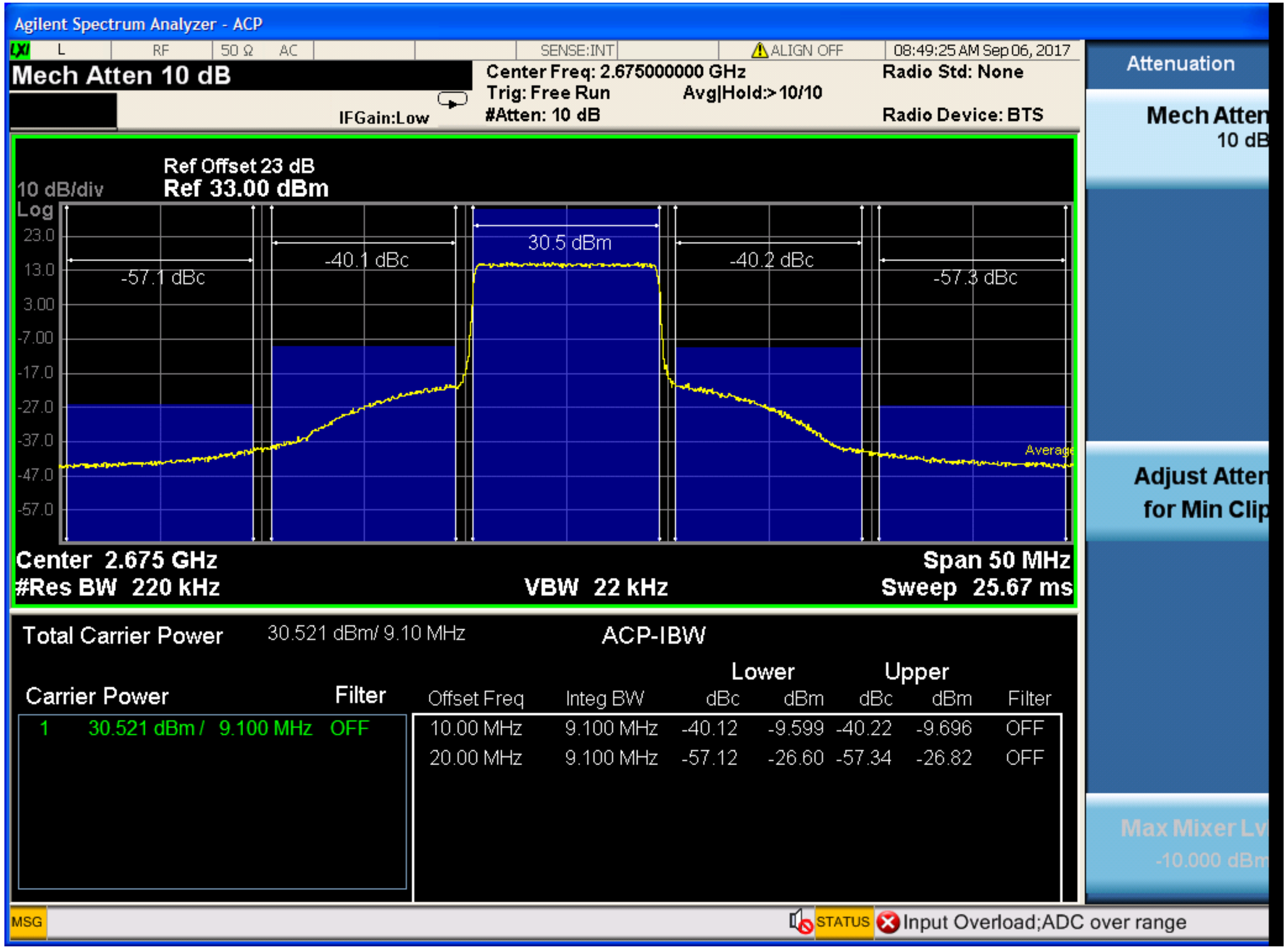
LMSP10C: 2x2 MIMO 20MHz LTE Case



EVM with ADPD

EVM = 4.48%

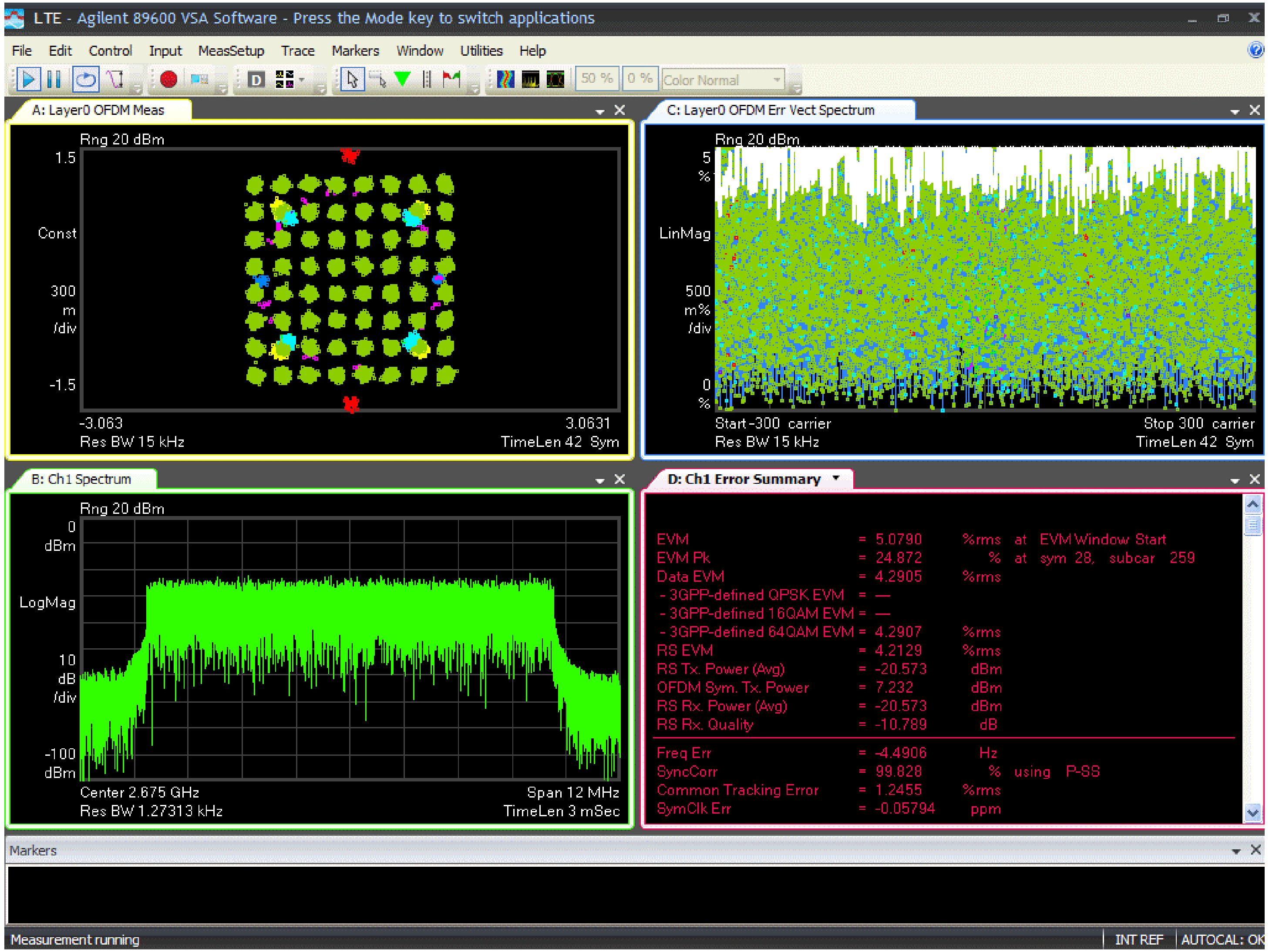
LMSP10C: 2x2 MIMO 10MHz LTE Case



ACPR without ADPD.

ACPR1 = -40.1 dBc, ACPR2 = -57.1dBc

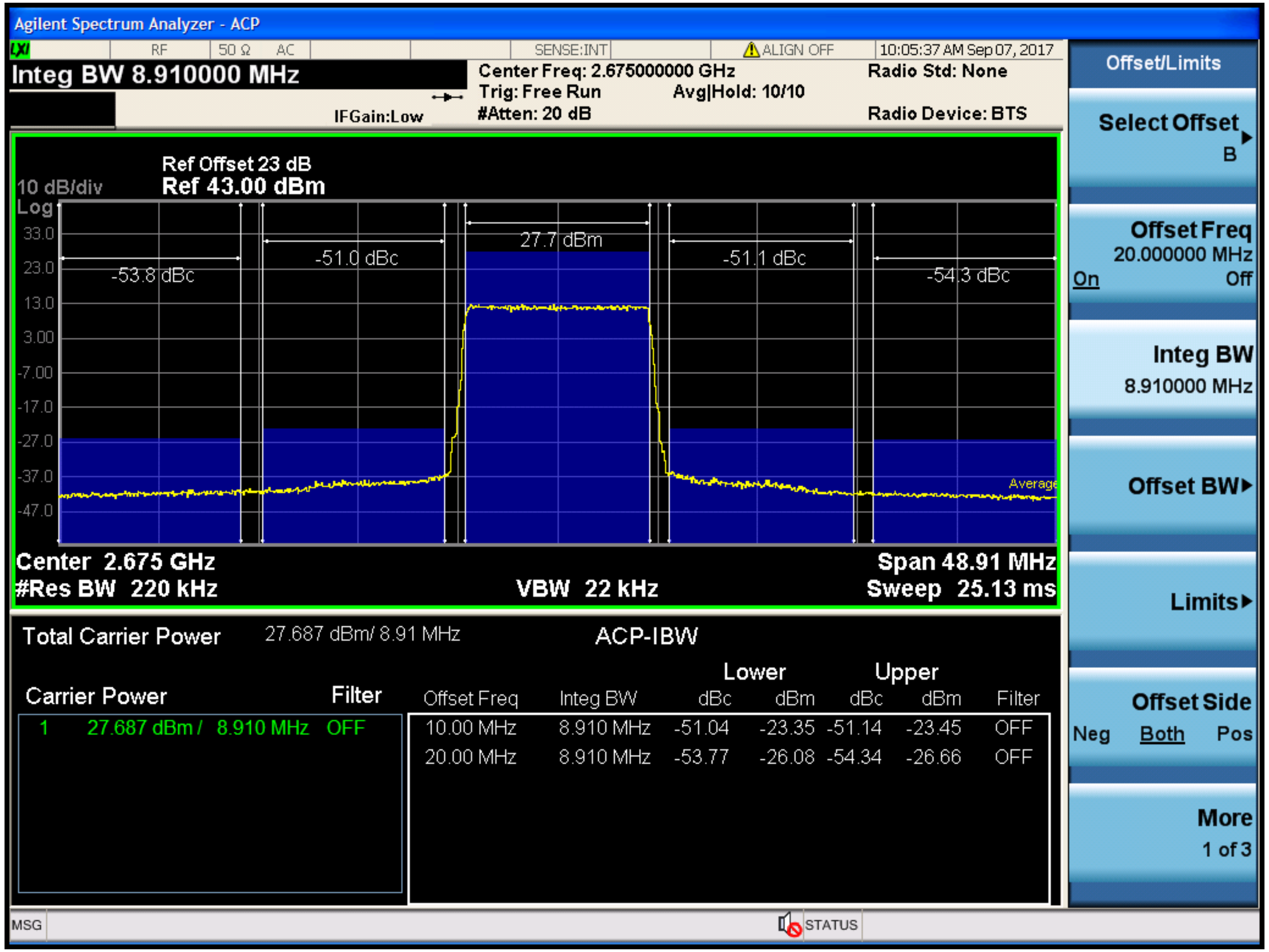
LMSP10C: 2x2 MIMO 10MHz LTE Case



EVM with no ADPD

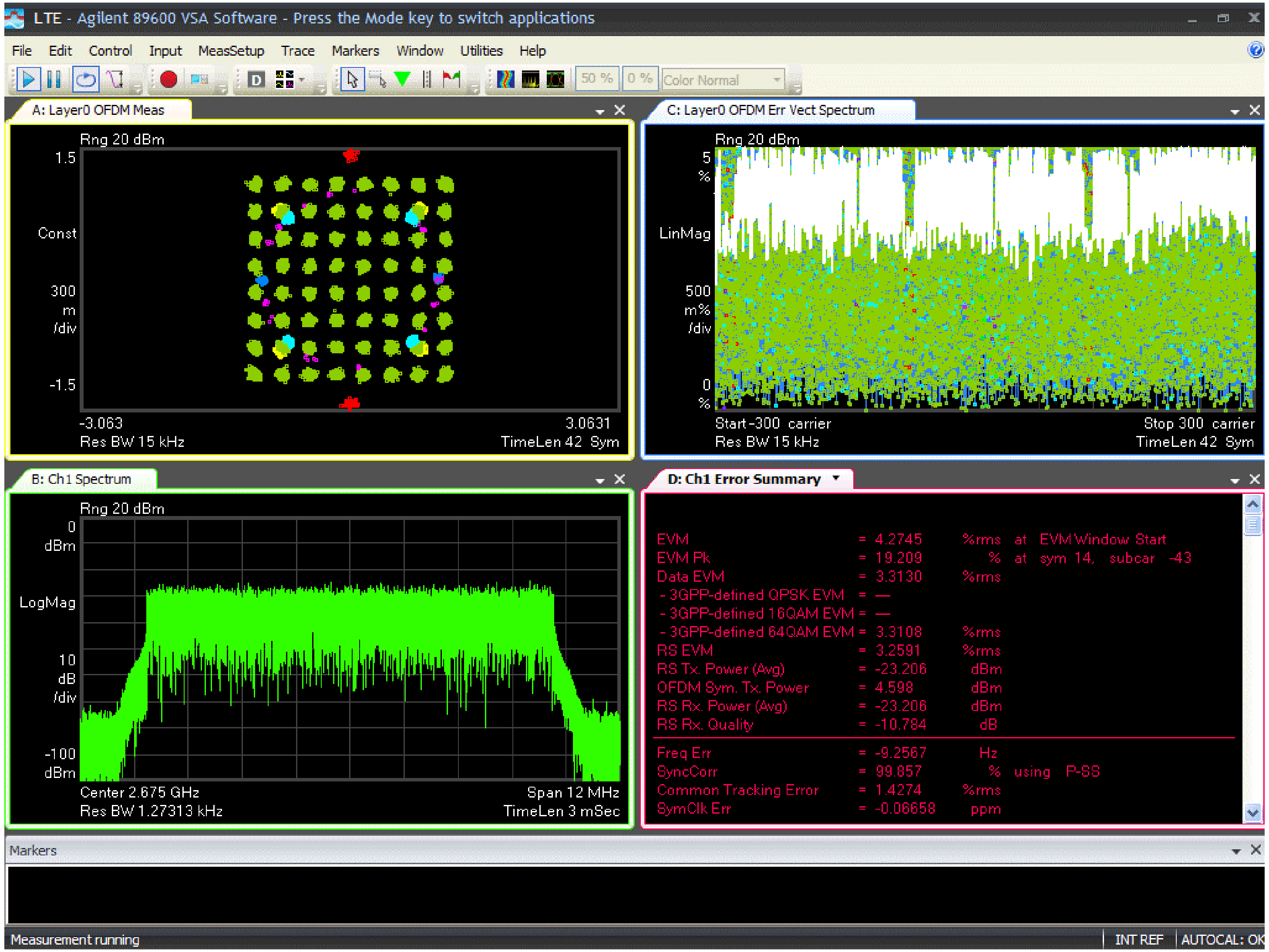
EVM = 5.1%

LMSP10C: 2x2 MIMO 10MHz LTE Case



ACPR with ADPD.
ACPR1 = -51 dBc, ACPR2 = -53.8dBc.

LMSP10C: 2x2 MIMO 10MHz LTE Case



EVM with ADPD

EVM = 4.27%

Conclusion

- LimeADPD algorithm has been implemented and verified by measured results.
- ADPD is capable of cancelling any distortion above system noise floor, DACs -> TX -> PA -> Coupler -> RX -> ADCs.
- Test Case 1 is single stage low power PA.
- Test Case 2 is more challenging to linearize since there are two PA stages (MAX2612 followed by Class-J GaN HEMPT amplifier) and both stages are nonlinear.
- Test Case 3 is most challenging for having a two stage PA, much higher RF frequency and modulation bandwidth (20MHz and 10MHz 2x2 MIMO LTE).
- Improvements in ACPR and EVM have been achieved in all cases as shown in Summary section.

5. Summary

Configuration	Modulation	Psat [dBm]	RF centre frequency [GHz]	ACPR [dBc]		EVM [%]	
				No ADPD	With ADPD	No ADPD	With ADPD
Case 1	WCDMA	19	2.14	-34	-51	6.58	3.24
Case 2	WCDMA	40	1.5	-43	-53	NA	NA
Case 3	2x2 20MHz MIMO LTE	40	2.675	-39.7	-49.1	5.33	4.48
Case 3	2x2 10 MHz MIMO LTE	40	2.675	-40.1	-51	5.1	4.27