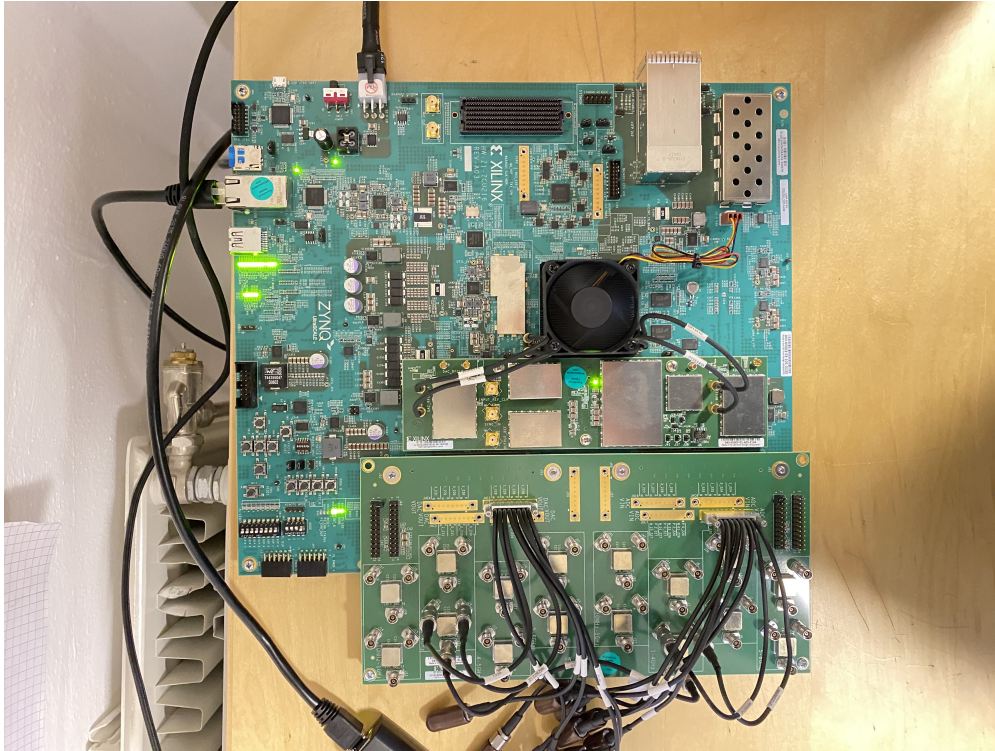




CHALMERS
UNIVERSITY OF TECHNOLOGY



Static Complex-Response Characterization of Selected ZCU216 RFSoc Loop-back Paths at 3 GHz Using Gaussian-Process Regression

Degree project report in Electrical Engineering

Nick Mashayekhi

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

DEGREE PROJECT REPORT IN ELECTRICAL ENGINEERING 2026

Static Complex-Response Characterization of Selected ZCU216 RFSoc Loopback Paths at 3 GHz Using Gaussian-Process Regression

Nick Mashayekhi



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Static Complex-Response Characterization of Selected ZCU216 RFSoc Loopback
Paths at 3 GHz Using Gaussian-Process Regression

Nick Mashayekhi

© Nick Mashayekhi, 2026.

Supervisor: Mustafa Ayebe, Department of Electrical Engineering
Examiner: Rob Maaskant, Department of Electrical Engineering

Bachelor degree project report 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Picture of ZCU216 RFSoc board

Typeset in L^AT_EX
Gothenburg, Sweden 2026

Static Complex-Response Characterization of Selected ZCU216 RFSoc Loopback Paths at 3 GHz Using Gaussian-Process Regression

Nick Mashayekhi

Department of Electrical Engineering

Chalmers University of Technology

Abstract

Modern wireless communication systems, particularly 5G networks and radar applications, demand exceptional hardware performance to meet increasing data-rate requirements and to support complex modulation schemes. Practical implementations inevitably suffer from hardware imperfections, including non-linear characteristics in digital-to-analog converters (DACs) and analog-to-digital converters (ADCs), which degrade signal fidelity. While advanced spatial processing techniques such as digital beamforming provide the overarching motivation for understanding these impairments, the fundamental prerequisite is a rigorous static characterization of the data converters themselves.

This thesis characterizes the static complex-response of selected loopback paths on the AMD Xilinx Zynq UltraScale+ ZCU216 RFSoc development board operating at a single-tone frequency of 3 GHz. The completed work comprises single-tone end-to-end loopback measurements of seven distinct DAC-to-ADC paths, the generation of a 28×16 two-dimensional full LHS map for a single selected path, and the development of a forward GPR interpolation model using MATLAB programming language. It is explicitly noted that the measurements represent the complete uncalibrated DAC-to-ADC loopback cascade, encapsulating analog interfaces, cables, timing, and software processing. The thesis establishes a forward interpolation model and provides quantifiable metrics on amplitude and phase response deviations. No inverse digital pre-distortion was implemented, nor was beamforming or communication-link performance evaluated. This work provides the foundational mathematical forward-modeling framework necessary for future hardware imperfection mitigation.

Keywords: RFSoc, Non-linearity, Gaussian process regression, ADC, DAC

Sammanfattning

Moderna trådlösa kommunikationssystem, särskilt 5G-nätverk, ställer höga krav på prestanda för att möta ökande datataktkrav och komplexa moduleringsscheman. Praktiska implementationer drabbas dock av hårdvarubrister, inklusive icke-linjära egenskaper i digital-till-analog-omvandlare (DAC) och analog-till-digital-omvandlare (ADC), vilket försämrar signaltroheten. Även om avancerade tekniker som strålformning utgör den övergripande motiveringen för att förstå dessa brister, är den grundläggande förutsättningen en noggrann statistisk karaktärisering av själva dataomvandlarna.

Detta examensarbete karaktäriserar den statistiska komplexa responsen hos utvalda loopback-vägar på utvecklingskortet AMD Xilinx Zynq UltraScale+ ZCU216 RF-SoC vid en tonfrekvens på 3 GHz. Det utförda arbetet omfattar loopback-mätningar från ände till ände av sju olika DAC-till-ADC-vägar, genereringen av en 28×16 tvådimensionell fullfaktoriell kartesisk karta för en utvald väg, samt utvecklingen av en framåt-modellerande Gaussisk processregression (GPR) i programmeringsspråket MATLAB. Det noteras uttryckligen att mätningarna representerar den kompletta okalibrerade DAC-till-ADC-kaskaden, vilket inkluderar analoga gränssnitt, kablar, tidsstyrning och mjukvarubehandling. Avhandlingen etablerar en strikt interpolationsmodell framåt och tillhandahåller kvantifierbara mått på amplitud- och fasavvikelser. Ingen invers digital fördistortion (DPD) implementerades, och inte heller utvärderades strålformning eller kommunikationslänkens prestanda. Detta arbete tillhandahåller det grundläggande matematiska ramverket för framtida hantering av hårdvarubrister.

Acknowledgements

I would like to thank Rob Maaskant and Mustafa Ayabe for their patience and guidance throughout this thesis.

Nick Mashayekhi, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DPD	Digital Pre-Distortion
FPGA	Field Programmable Gate Array
GP	Gaussian Process
GPR	Gaussian Process Regression
IoT	Internet of Things
LHS	Latin Hypercube Sampling
LPF	Low Pass Filter
RF	Radio Frequency
RFSoc	Radio Frequency System-On-Chip

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	Index for observed data points in dataset
k	Index for consecutive analog input values or codes
n	Index for discrete signal samples

Sets

$\mathcal{D}$	Dataset of input-output pairs
X	Input matrix of all observed inputs
y	Output vector of all observed outputs

Parameters

ΔV	Voltage step size
σ_n^2	Noise variance
f_s	Sampling frequency
N	Resolution of data converter in bits (or desired sample size)
T	Sample period
V_0	Minimum analog input to an all-zero output
V_{LSB}	Spacing between two adjacent output codes
V_{max}	Maximum voltage level
V_{min}	Minimum voltage level
$ V_{in} $	Input voltage magnitude to the DAC

$\angle V_{in}$	Input phase angle to the DAC
-----------------	------------------------------

Variables

$DNL(k)$	Differential non-linearity at code k
$f(x)$	latent function modelled by a Gaussian process
$INL(D)$	Integral non-linearity for code D
$k(x, x')$	Covariance function (kernel)
$m(x)$	Mean function
t	Time
$x[n]$	Discrete-time signal
$x_{ZOH}(t)$	Continuous signal constructed by Zero-Order Hold
μ_*	Posterior mean (predicted value)
σ_*^2	Posterior variance (predictive uncertainty)
$ V_{out} $	Output voltage magnitude to the ADC
$\angle V_{out}$	Output phase angle to the ADC

Contents

List of Acronyms	ix
Nomenclature	xi
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Background	1
1.2 Purpose and Goal	2
1.3 Limitations and Demarcations	2
2 Theory	3
2.1 RFSoc ZCU216	3
2.1.1 Digital-to-Analog Converter (DAC)	4
2.1.2 Analog-to-Digital Converter (ADC)	6
2.2 Non-linearity & RF Impairments	7
2.3 Beamforming	7
2.4 Gaussian Process Regression	9
2.5 Latin Hypercube sampling	10
3 Methods	13
3.1 MATLAB Hardware Interface	13
3.1.1 Reproducible DSP Parameters	13
3.2 Mapping DAC-to-ADC Behavior	13
3.2.1 Hardware Configuration and Setup	13
3.2.2 Measurement Protocol	14
3.2.2.1 Amplitude Response Characterization	14
3.2.2.2 Phase Response	15
3.2.3 Data Acquisition	15
3.2.4 Visualization	16
3.2.5 Dense Grid Measurement mapping for $T0D1 \rightarrow T0A1$	16
3.3 Regression Model	16
3.3.1 Implementation of GPR	16
3.3.2 Data Scaling and Basis Functions	17
3.3.3 Visualization of the Kriging Field	17

3.3.4	Cross-Validation and Predictive Performance	17
4	Results & Discussion	19
4.1	Mapping DAC-to-ADC Transfer Characteristics	19
4.2	Two-dimensional mapping of DAC1-to-ADC1	24
4.3	GPR model for the two-dimensional DAC1-to-ADC1 mapping	26
5	Conclusion	29
	Bibliography	31
A	Appendix	I

List of Figures

2.1	Illustration of Zynq UltraScale+ RFSoc family taken from [2]	3
2.2	XM655 Block Diagram [14]	4
2.3	Response after zero-order hold taken from [2]	5
2.4	A continuous signal before and after filtering taken from [2]	5
2.5	Analog to digital conversion taken from [2]	6
2.6	Conceptual illustration: (a) Single broad pattern vs (b) Steered highly directive pattern, taken from [2]	8
2.7	Signals from a linear array: (a) Unsteered broadside emission vs (b) Steered emission via progressive phase delays, taken from [2]	9
2.8	1D GPR example. The red line is the predictive mean, and the shaded gray region represents the prediction band (uncertainty/credible interval). Taken from [13]	10
2.9	Comparison of random and Latin hypercube sampling examples in two dimensions (every row and column is sampled with LHS)[7]	11
4.1	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC1	19
4.2	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC1 $\rightarrow$ ADC1	20
4.3	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC3	20
4.4	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC1 $\rightarrow$ ADC3	20
4.5	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC5	21
4.6	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC1 $\rightarrow$ ADC5	21
4.7	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC7	21
4.8	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC1 $\rightarrow$ ADC7	22
4.9	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC3 $\rightarrow$ ADC1	22
4.10	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC3 $\rightarrow$ ADC1	22

4.11	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC5 $\rightarrow$ ADC1	23
4.12	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC5 $\rightarrow$ ADC1	23
4.13	$ V_{\text{out}} $ and $\angle V_{\text{out}}$ as a function of $ V_{\text{in}} $ with a constant phase of $\angle V_{\text{in}} = 0^\circ$ for loopback of DAC7 $\rightarrow$ ADC1	23
4.14	$ V_{\text{out}} $ & $\angle V_{\text{out}}$ as a function of $\angle V_{\text{in}}$ with a constant amplitude of $ V_{\text{in}} = 1.0$ for loopback of DAC7 $\rightarrow$ ADC1	24
4.15	Phase output as a function of input phase and amplitude for DAC1 $\rightarrow$ ADC1	25
4.16	Amplitude output as a function of input phase and amplitude for DAC1 $\rightarrow$ ADC1	25
4.17	The Amplitude field in the plot above and the 95% prediction interval width below, with the z-axis showing the difference between the upper and lower limits of the 95% interval.	26
4.18	The Phase field in the plot above and the 95% prediction interval width below, with the z-axis showing the difference between the upper and lower limits of the 95% interval.	27
4.19	Comparison between heatmap of the phase field and confidence interval of the GPR phase. The lack of even sampling in the amplitude direction can be noticed on the GPR model. At points 0.3 V, 0.4 V and 0.65 V	27

List of Tables

3.1	RFSoc DSP Configuration Parameters	14
3.2	Hardware to MATLAB Channel Alias Mapping	14

1

Introduction

1.1 Background

Electromagnetic waves serve as the backbone of modern technological civilization, enabling wireless communication, radar detection, medical imaging, and countless other applications across diverse frequency bands. From the essential Maxwell equations governing wave propagation to practical implementations in radio frequency systems, the precise control and manipulation of electromagnetic signals remains central to advancing communication technologies.

Modern RF systems rely heavily on sophisticated hardware components, including digital-to-analog converters (DACs) and analog-to-digital converters (ADCs), which serve as critical interfaces between digital signal processing and the analog electromagnetic domain. These components, particularly in high-performance platforms such as Radio Frequency System on Chip (RFSoc) architectures, must maintain exceptional linearity and phase coherence to preserve signal integrity across the entire signal chain. The increasing demand for higher data rates and more complex modulation schemes in contemporary wireless systems places strict requirements on analog components.

To keep up with these demands, techniques like beamforming are utilized to enable high data rates, reliable connectivity, and efficient utilization of the radio-frequency spectrum. However, practical implementations of such technologies are subject to various hardware imperfections, including non-linear amplifier characteristics, phase noise, and manufacturing tolerances. These imperfections degrade antenna system performance, leading to increased system output variance and reduced signal fidelity.

Understanding and characterizing these effects is essential because real-world systems often deviate from the ideally linear relationship between input and output. This deviation is particularly relevant in high-frequency communication systems where precision and stability are critical. A critical first step in mitigating these errors is characterizing the static non-linearities of the data converters themselves. While true Digital Pre-Distortion (DPD) requires dynamic, wideband characterization, foundational loopback measurements provide initial insights into the coupled amplitude and phase deviations inherent to the hardware.

Beyond pure technical optimization, a comprehensive societal and ethical assess-

ment is relevant to contextualize the deployment of such hardware. Key considerations include the energy consumption and overall lifecycle environmental impact of complex RF hardware, alongside challenges in spectrum coexistence and radio-frequency pollution. Finally, technical evaluations must maintain methodological rigor; it is vital to distinguish between basic calibration or characterization methods and fully demonstrated real-world performance improvements to avoid overstating system capabilities.

1.2 Purpose and Goal

The primary objective of this thesis is to systematically analyze the static, complex-response non-linear effects of the RFSoc ZCU216 development board's DAC and ADC loopback paths at a continuous-wave frequency of 3 GHz. The outputs of interest are the measured amplitude magnitude $|V_{out}|$ and the measured phase angle $\angle V_{out}$ extracted from the ADC after traversing the digital baseband, digital mixers, data converters, analog cables, and measurement software. Subsequently, the project aims to model this complex transfer behavior using a forward GPR model implemented in MATLAB. The GPR serves as a non-parametric probabilistic interpolation tool to map the input parameter space (commanded amplitude and phase) to the measured output space. The primary goal is to establish a mathematical framework that characterizes the distortion of two-dimensional static amplitude and phase response surfaces. Then generating a smooth predictive field and corresponding uncertainty metrics.

1.3 Limitations and Demarcations

Due to resource and time constraints, the scope of this evaluation is strictly bounded by several technical demarcations. The first limitation is that the characterization utilizes a single static 3 GHz tone, meaning the wideband frequency spectrum typical of dynamic 5G networks is not captured within this study. Furthermore, the developed GPR model serves exclusively as a forward interpolation model ($f(V_{in}) \rightarrow V_{out}$). Consequently, this thesis neither implements an inverse digital pre-distorter (DPD) nor evaluates real-time hardware error mitigation on the FPGA. The measurement setup is also restricted to a complete, uncalibrated loopback cascade. Because no vector network analyzer (VNA) de-embedding was performed to isolate the non-linearities of the DAC or ADC chips themselves, all systemic artifacts, including analog interfaces, baluns, cables, timing skew, FFT windowing, and software-induced processing effects—remain lumped together. In terms of data collection, the full two-dimensional full-factorial LHS grid mapping and its GPR modeling were executed exclusively on a single selected loopback path out of the seven tested. Finally, while digital beamforming and the optimization of a 5×1 Vivaldi antenna array provided the initial motivation for this research, practical beamforming performance falls outside the evaluated scope and is not presented in the final results.

2

Theory

2.1 RFSoc ZCU216

The Xilinx ZCU216 is an RFSoc development board designed for advanced RF, wireless, and mixed-signal applications. It is built around the Zynq UltraScale+ RFSoc family. It features ultrafast ADCs and DACs; this gives it the ability to sample the radio signal directly on the board without the need for an external analog upconversion mixer or down-sampling stages. This is especially important in systems that use multiple channels as it requires fewer components. This also leads to more power-efficient systems, which reduce noise and increase bandwidth [3].

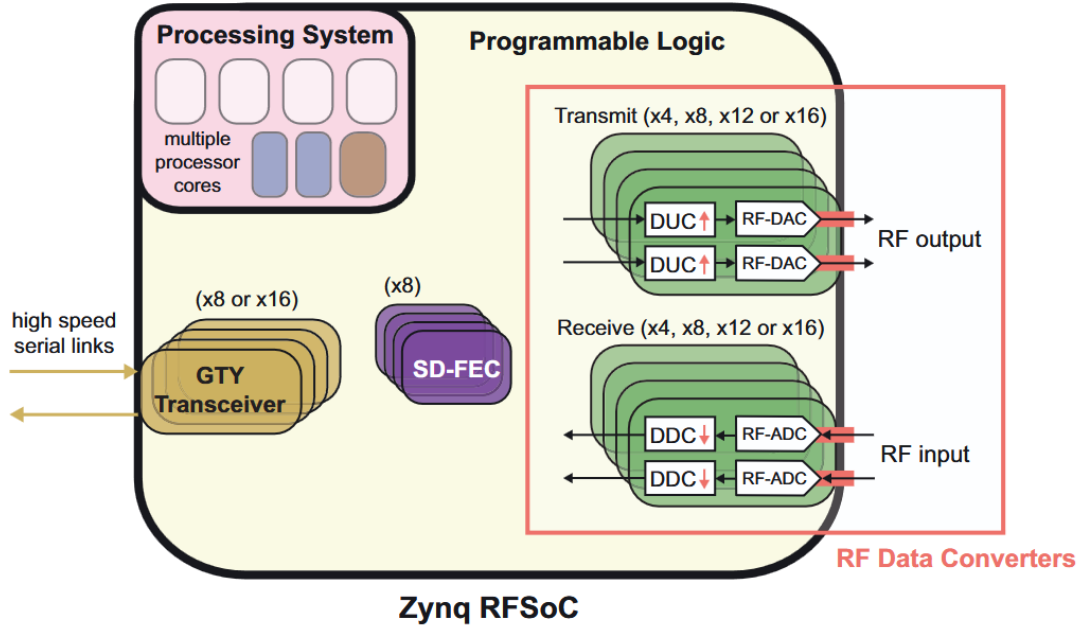


Figure 2.1: Illustration of Zynq UltraScale+ RFSoc family taken from[2]

As shown in Figure 2.1, the RFSoc platform contains the data converters (green), the soft-decision forward error correction (purple), the gigabit transceivers (yellow), as well as the processing system (pink). The RF Data converters in Figure 2.1 are accessible through the XM655 add-on board, which is specifically used for loop-back testing. Loopback testing routes the transmitted RF signal back to the receiver. The layout of the breakout board can be seen in Figure 2.2.

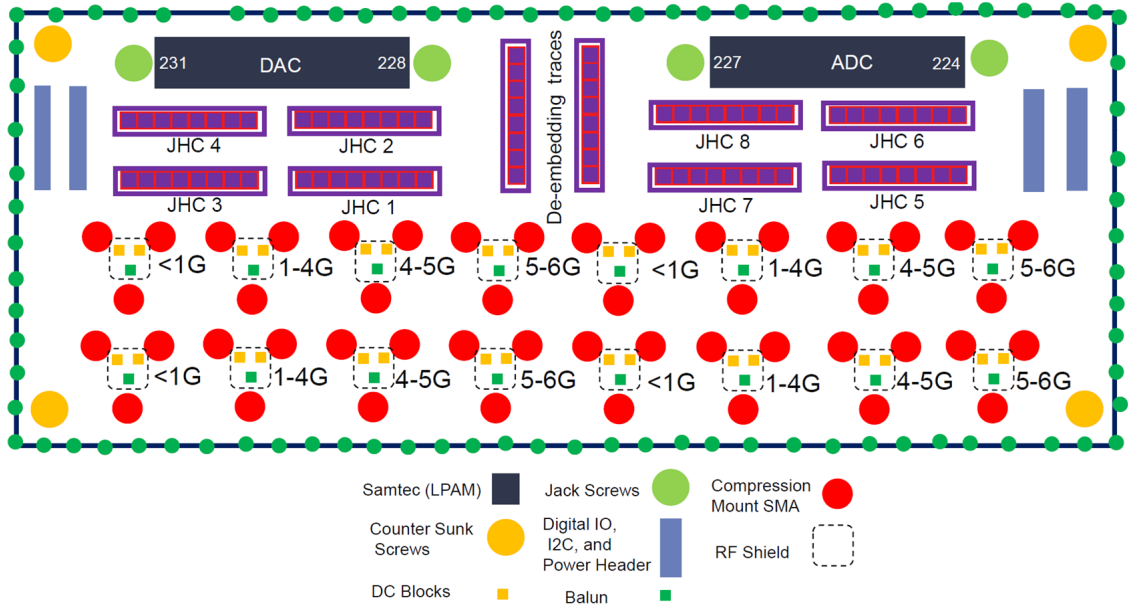


Figure 2.2: XM655 Block Diagram [14]

The DACs and ADCs are accessible through the Carlisle Core HC2 8-channel connectors (JHC connectors in Figure 2.2). The parallel connectors (JHC3, JHC4, etc.) logically group into “tiles.” These tiles form the connection from the software to the board hardware. There are four tiles in total for both the DACs and ADCs, labeled Tile0 through Tile3. Each DAC and ADC block contains four individual converters, resulting in a total of 16 DACs and 16 ADCs on the XM655 add-on board. An individual converter is labeled as $TXDX - TXAX$, where T refers to the tile number, D to the DAC, and A to the ADC. For loopback testing, the differential converter interface connects through an appropriate balun/transformer network, performing balanced-to-unbalanced conversion and impedance transformation. A balun is also connected on the receiver side to complete a 50Ω match [9].

2.1.1 Digital-to-Analog Converter (DAC)

In a world where only analog signals propagate through space, DACs are crucial in antenna systems. The role of a DAC is to convert a discrete-time sequence into an analog waveform; the analog output network then suppresses unwanted spectral images as required.

The DAC in the ZCU216 uses a zero-order hold, which holds each sample for one sampling period $T = 1/f_s$ [2]. This results in a staircase-like waveform as shown in Figure 2.3.

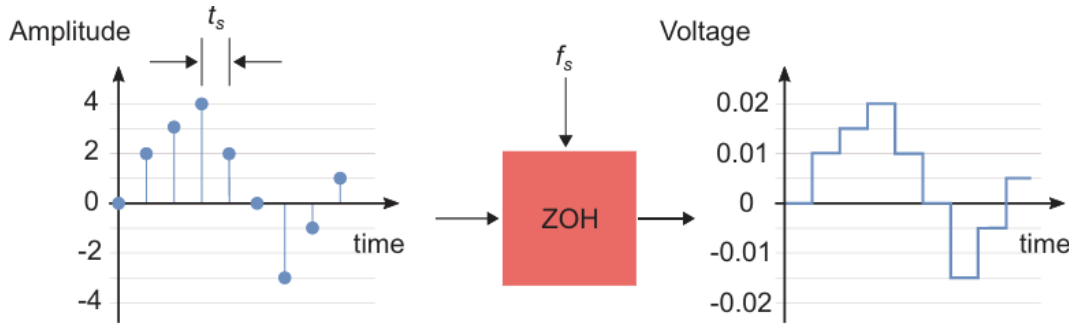


Figure 2.3: Response after zero-order hold taken from [2]

A zero-order hold constructs this shape from a discrete sequence $x[n]$ mathematically as:

$$x_{\text{ZOH}}(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \text{rect}\left(\frac{t - \frac{T}{2} - nT}{T}\right) \quad (2.1)$$

where $\text{rect}(\cdot)$ is the rectangular function defined to be 1 over $[-0.5, 0.5]$ and 0 elsewhere, t is continuous time, and T is the sampling period. The $T/2$ shift ensures that each held pulse properly spans the continuous time interval from nT to $(n+1)T$, making the system causal.

The problem with this rectangular shape is the presence of spectral images and the sinc-shaped frequency response introduced by the sample-and-hold operation. These high-frequency artifacts are undesirable in radio applications, as they could interfere with adjacent channels. To mitigate this, the DAC applies a low-pass filter (LPF), often referred to as a reconstruction filter [4].

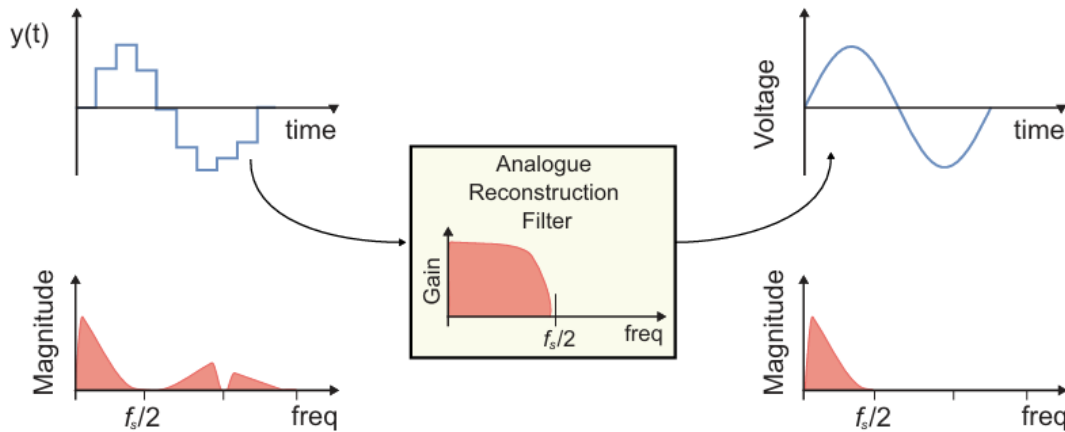


Figure 2.4: A continuous signal before and after filtering taken from [2]

A DAC has two other fundamental characteristics: the sampling rate and the resolution. Resolution is expressed in bits; for instance, a 12-bit DAC can represent $2^{12} = 4096$ different voltage points. The sampling rate determines the reproducible frequency range. According to the Nyquist theorem, the simplest low-pass criterion requires $f_{\text{max}} < f_s/2$. For example, a 200 kHz sampling rate theoretically reproduces

signals up to 100 kHz. However, direct-RF systems like the ZCU216 deliberately utilize higher Nyquist zones and external band-pass networks, meaning the analog filter passband is not universally set exactly at $f_s/2$.

For the ZCU216 board configured for these measurements, the DAC resolution is 14 bits and the maximum capable sampling rate is 9.86 Gsps [2]. However, generating a 3 GHz RF tone relies on the configured sampling rate, digital interpolation, digital mixer/NCO frequencies, and the selected output mode (e.g., Mix-Mode to optimize higher Nyquist zone power). Assuming a generic differential full-scale range (FSR) of 1 V, the endpoint step size is given by:

$$\Delta V = \frac{\text{FSR}}{2^N - 1} \approx \frac{1 \text{ V}}{16383} \approx 61 \mu\text{V}$$

This internal architecture allows the device to directly synthesize RF frequencies without an external analog upconversion mixer.

2.1.2 Analog-to-Digital Converter (ADC)

Once the analog signal is captured, it must be converted to the digital domain. One of the fundamental aspects of converting an analog signal to a digital one is sampling. Sampling discretizes time into a set of instances. The second fundamental part of an analog-to-digital converter (ADC) is its ability to handle quantization, which maps continuous amplitude to discrete levels. Finally, digital coding maps these quantized levels into digital words.

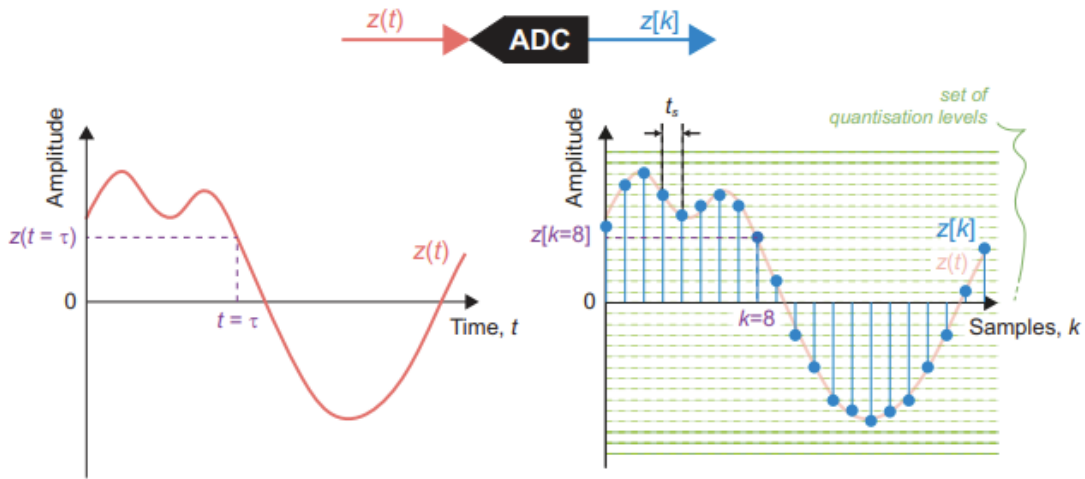


Figure 2.5: Analog to digital conversion taken from [2]

The sample spacing is $T_s = 1/f_s$. The appropriate sampling rate is governed by the Nyquist sampling theorem, which dictates $f_s > 2f_{max}$ for strict low-pass signals. If this condition is not met, aliasing occurs. Aliasing folds frequency components into the sampled spectrum according to the Nyquist-zone mapping. To receive a 3 GHz signal, the RF-ADC employs band-pass sampling (operating in a higher Nyquist zone) coupled with internal digital down-conversion (DDC), a numerically

controlled oscillator (NCO), and anti-alias filtering to properly translate the alias to baseband [4].

The quantization resolution is given in bits, providing 2^N discrete values corresponding to voltage steps. Following the endpoint convention matching the DAC, the gap between values is:

$$\Delta V = \frac{V_{max} - V_{min}}{2^N - 1} \quad (2.2)$$

where V_{max} and V_{min} define the full-scale range (FSR).

2.2 Non-linearity & RF Impairments

A linear system is one that obeys two principles: superposition and homogeneity [6]. If an input signal $x_1(t)$ produces an output $y_1(t)$ and $x_2(t)$ produces $y_2(t)$, additivity requires $x_1 + x_2$ to produce $y_1 + y_2$. Homogeneity means that if an input $x(t)$ results in the output $y(t)$, an input $a \cdot x(t)$ will produce $a \cdot y(t)$ for any scalar a . A nonlinear system violates at least one of these properties.

While traditional data converters such as DAC and ADC are often evaluated using static metrics like differential nonlinearity and Integral nonlinearity, RF systems operating at microwave frequencies are much more sensitive to dynamic and complex impairments. differential nonlinearity and Integral nonlinearity evaluate code transition levels and static transfer curves; however, characterizing a complete uncalibrated loopback chain at a 3 GHz continuous-wave frequency requires a different approach. Active RF components and cascaded data converters introduce complex gain errors. Instead of static code deviations, the system's non-linearity is characterized by its large-signal sinusoidal steady-state complex response. By varying the amplitude and phase of the input tone, the resulting complex fundamental at the output can be modeled as:

$$|V_{out}| = F(|V_{in}|, \angle V_{in}) \quad (2.3)$$

$$\angle V_{out} = G(|V_{in}|, \angle V_{in}) \quad (2.4)$$

2.3 Beamforming

Electromagnetic waves radiated by real antennas depend heavily on their geometry and environment, resulting in specific element patterns and array factors, rather than radiating uniformly in all directions as a theoretical isotropic source would. For advanced tasks such as 5G, RADAR, and Wi-Fi, transmitters and receivers need to steer the signal in desired directions to overcome high path loss at higher frequencies. This spatial processing improves array gain, interference suppression, and spatial reuse, which in return increases achievable link throughput [10].

One method is mechanical steering, such as physically reorienting a dish antenna. However, a more versatile and rapid method is electronic beamforming.

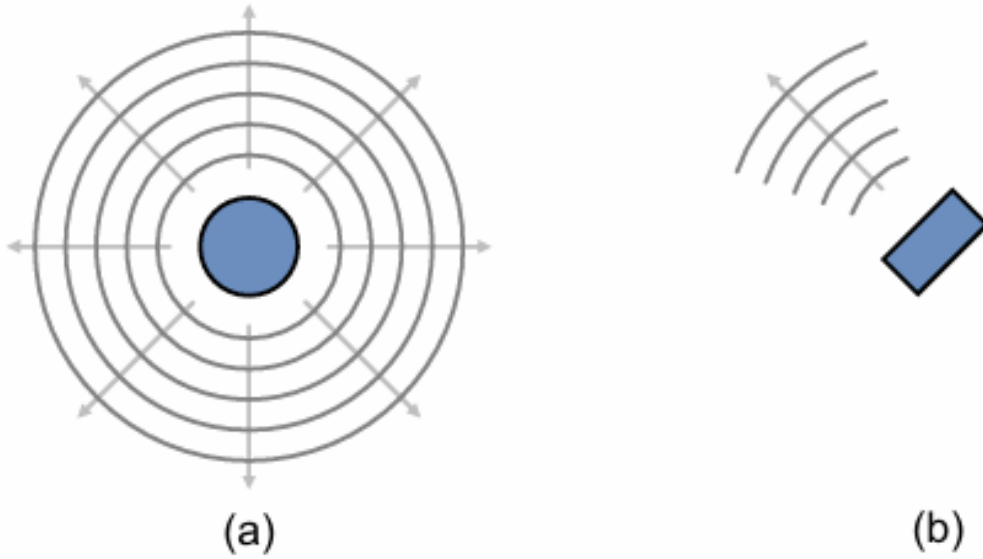


Figure 2.6: Conceptual illustration: (a) Single broad pattern vs (b) Steered highly directive pattern, taken from [2]

Beamforming is a widely used technique in antenna engineering, signal processing, and biomedical applications. It is defined as the weighted combination of multiple spatial channels to control the overall array response. When elements radiate, their fields superpose, producing an angular radiation pattern characterized by main lobes and sidelobes. The main lobe contains the direction of maximum radiation intensity and gain. To effectively focus a signal, elements must be spaced relative to the wavelength (often near $\lambda/2$) to prevent unwanted grating lobes.

Constructive and destructive field superposition dictates the far-field angular response. For a uniform linear array (ULA) of N elements with spacing d , the array factor is given by:

$$AF(\theta) = \sum_{n=0}^{N-1} w_n e^{jnk d \sin(\theta)} \quad (2.5)$$

where w_n represents the complex weight (amplitude and phase) applied to element n , $k = 2\pi/\lambda$ is the wavenumber, and θ is the observation angle. If all w_n are equal and in-phase, the main beam is broadside ($\theta = 0^\circ$).

A phased array commonly applies phase shifts to steer the beam; however, this is a narrowband approximation of a true-time-delay array and introduces beam-squint over wide bandwidths. Digital beamforming applies these complex weights w_n in the digital domain, prior to the DACs in the transmit path. This allows the beam to be steered rapidly and electronically, limited only by hardware update rates and latency, without mechanically reorienting the array [1].

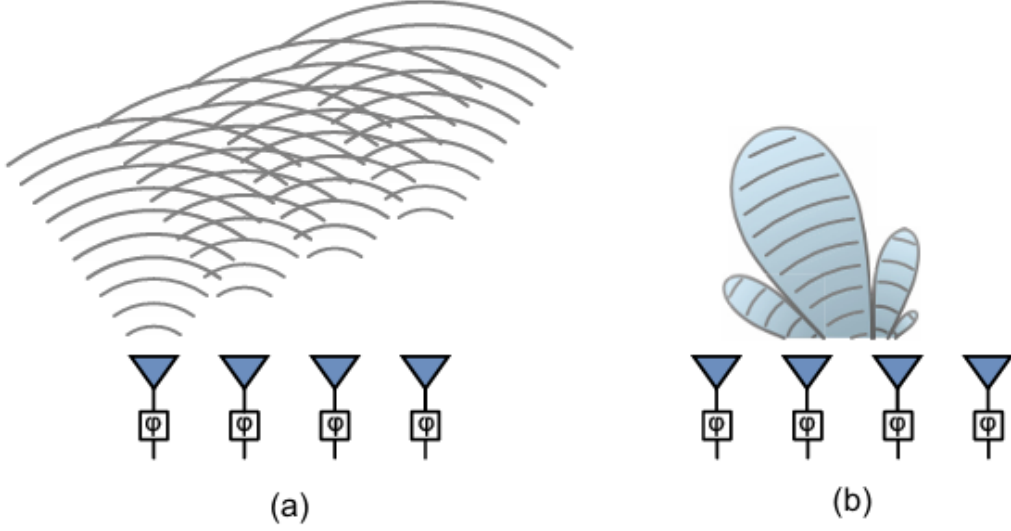


Figure 2.7: Signals from a linear array: (a) Unsteered broadside emission vs (b) Steered emission via progressive phase delays, taken from [2]

2.4 Gaussian Process Regression

Gaussian process regression (GPR) is a key model in probabilistic supervised machine learning, widely applied in regression tasks [8]. Regression involves inferring an unknown function $f(\mathbf{x})$ and predicting its value at new inputs. GPR is defined by a mean function and a covariance function (kernel). The covariance kernel specifies how strongly function values at two input locations are correlated spatially.

A Gaussian process (GP) describes a distribution over functions, defined by its mean $m(\mathbf{x})$ and kernel $k(\mathbf{x}, \mathbf{x}')$:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \quad (2.6)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \quad (2.7)$$

We write the GP as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (2.8)$$

The expected value $\mathbb{E}$ is taken over different realizations of the GP. The covariance function determines how the values of f at $\mathbf{x}$ and $\mathbf{x}'$ covary.

In GPR, we have a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where the observations contain noise: $y_i = f(\mathbf{x}_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$. Using observed inputs $\mathbf{X}$ and outputs $\mathbf{y}$, the joint distribution of observed values and a latent predicted value f_* at new inputs $\mathbf{x}_*$ is [12]:

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & K(\mathbf{X}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbf{X}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right), \quad (2.9)$$

Note that the lower-right covariance block represents the latent function f_* ; predicting a new noisy observation y_* would require adding $\sigma_n^2 \mathbf{I}$ to that block.

Conditioning on the data yields the posterior predictive distribution $p(f_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(\mu_*, \sigma_*^2)$, where:

$$\mu_* = m(\mathbf{x}_*) + K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1}(\mathbf{y} - m(\mathbf{X})) \quad (2.10)$$

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{x}_*) \quad (2.11)$$

While the theoretical conditional variance σ_*^2 does not depend directly on the specific values in $\mathbf{y}$, in practice, optimizing the kernel hyperparameters based on $\mathbf{y}$ causes the fitted variance to indirectly depend on the observations. GPR is highly capable but involves $O(N^3)$ training complexity due to the matrix inversion, requiring careful dataset sizing.

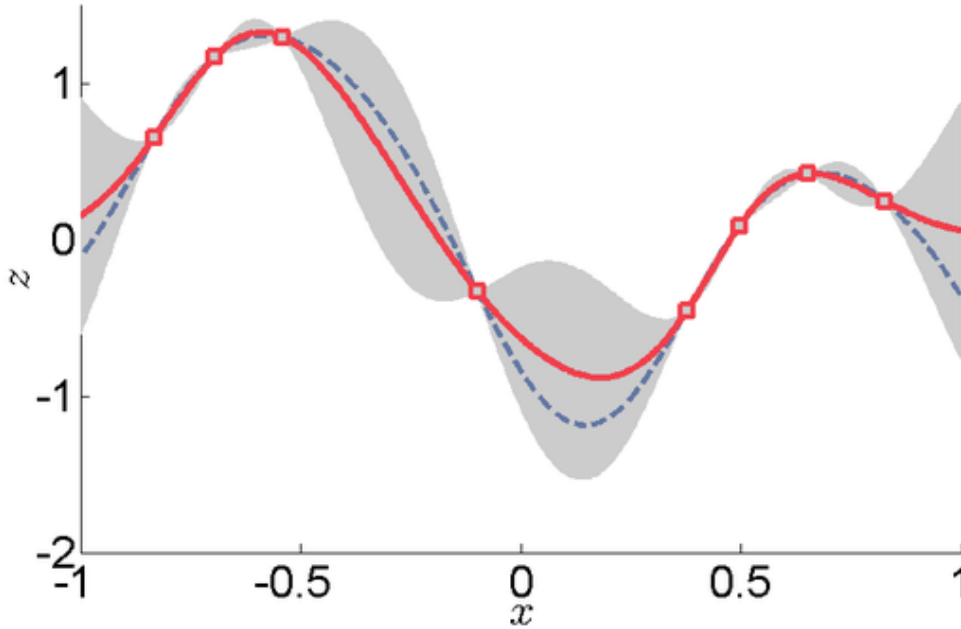


Figure 2.8: 1D GPR example. The red line is the predictive mean, and the shaded gray region represents the prediction band (uncertainty/credible interval). Taken from [13]

2.5 Latin Hypercube sampling

Latin Hypercube sampling or from now on LHS is an near statistical method for generating near random samples values from a multidimensional distribution [5]. LHS is an extension of the Latin square sampling to higher dimensions and provides better coverage of the sample space compared to simple random sampling [11].

In LHS, the sampling procedure divides each parameter dimension into N equally probable intervals, where N is the desired sample size. For a k -dimensional parameter space, this creates an N^k grid of hypercubes. One sample is then placed in each of the N intervals for each dimension, with the constraint that when projected onto any axis, each sample falls into a distinct interval

This could be explained as having chess rooks on a board without threatening each other

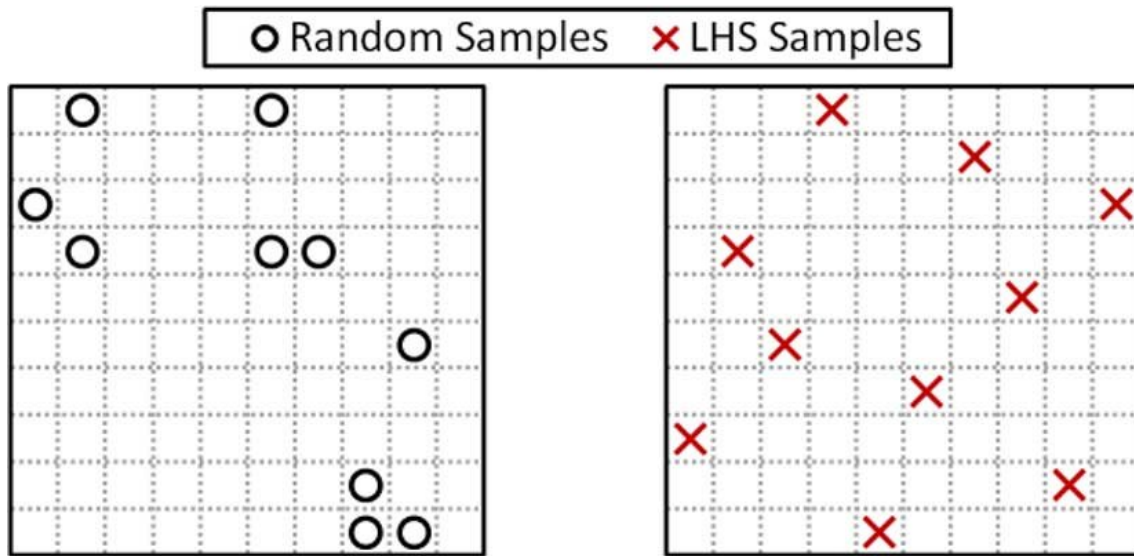


Figure 2.9: Comparison of random and Latin hypercube sampling examples in two dimensions (every row and column is sampled with LHS)[7]

3

Methods

3.1 MATLAB Hardware Interface

To characterize the data converters on the ZCU216 RFSoc board, the measurement campaign was automated using a custom MATLAB framework. The primary execution and modeling were managed through a MATLAB script, which controlled the life-cycle of the hardware interaction from initialization to data processing. The primary execution script, `Example0_DACandADCModeling.m` written by Mustafa Ayabe, synchronizes the software to the hardware using the underlying library for the ZCU216.

Most of the board constants and inputs can be configured in the `Load_Board_Inputs` script. Within this script, the specific ADC and DAC can be chosen, alongside their amplitude, phase, and frequency parameters.

3.1.1 Reproducible DSP Parameters

To ensure strict reproducibility, the internal digital signal processing parameters of the RFSoc were explicitly defined in the Matlab program. Table 3.1 gives the exact configuration state of the uncalibrated cascade during the 3 GHz single-tone characterization.

3.2 Mapping DAC-to-ADC Behavior

3.2.1 Hardware Configuration and Setup

The experimental setup employed four DAC channels and four ADC channels interfaced through the Carlisle Core HC2 8-channel connector. A total of seven distinct DAC-to-ADC channel combinations were evaluated to characterize the system performance across multiple signal paths. The MATLAB script and hardware setup utilize a specific notation to access a given ADC and DAC pairing, denoted as $\text{DACX} \rightarrow \text{ADCX}$.

Table 3.2 defines the mapping between the physical hardware designations and the MATLAB notation.

The DAC channels were interfaced to the RF front-end through baluns optimized for the 1-4 GHz frequency band. Each DAC output was connected to its corresponding balun to ensure proper impedance matching and signal conditioning within the

Table 3.1: RFSoc DSP Configuration Parameters

Parameter	Configuration Value
Clock Source	CLK104 Add-on (LMK04828 / LMX2594)
Reference Clock	300.00 MHz
DAC Sample Rate (f_s)	9.6 GSPS
ADC Sample Rate (f_s)	2.4 GSPS
DAC Data Path	Digital up-Conversion Bypassed
DAC Interpolation	1x
ADC Decimation	1x
Mixer Mode	Bypassed / 0 Hz NCO
DAC Nyquist Zone	Zone 1
ADC Nyquist Zone	Zone 4
Signal Frequency	3.000 GHz (Continuous Wave)
MTS (Multi-Tile Sync)	Enabled (MMCM $F_{in} = 300$ MHz)
DAC Buffer Length	16,384 samples
ADC Capture Length	16,384 samples

Table 3.2: Hardware to MATLAB Channel Alias Mapping

Hardware Path	MATLAB Alias
$T0D1 \rightarrow T0A1$	DAC1 $\rightarrow$ ADC1
$T0D1 \rightarrow T0A3$	DAC1 $\rightarrow$ ADC3
$T0D1 \rightarrow T1A1$	DAC1 $\rightarrow$ ADC5
$T0D1 \rightarrow T1A3$	DAC1 $\rightarrow$ ADC7
$T0D3 \rightarrow T0A1$	DAC3 $\rightarrow$ ADC1
$T1D1 \rightarrow T0A1$	DAC5 $\rightarrow$ ADC1
$T1D3 \rightarrow T0A1$	DAC7 $\rightarrow$ ADC1

specified operational bandwidth. The balun configuration eases the conversion between the differential outputs and the signal requirements of the RF measurement system.

3.2.2 Measurement Protocol

The characterization of DAC-to-ADC behavior in the loop-back configuration on the RFSoc ZCU216 development board was conducted through systematic amplitude and phase response measurements. Two distinct test protocols were implemented to evaluate the system performance.

3.2.2.1 Amplitude Response Characterization

The amplitude response characterization employed variable input voltage magnitudes while maintaining a constant phase. Input voltage magnitudes were systematically varied from a $|V_{in}| = 0.02$ to 1.0 in increments of 0.02, with the input phase

angle maintained at a constant $\angle V_{in} = 0^\circ$. This approach provided 50 distinct amplitude points across the full range. For each input magnitude, the generated waveform was transmitted through the loop-back path and captured by the ADC. Ten measurements were made at each amplitude to estimate repeatability and measurement noise.

3.2.2.2 Phase Response

Phase response measurements utilized a similar approach, but with constant amplitude and variable phase conditions. The input voltage magnitude was fixed at $|V_{in}| = 1.0 \text{ V}$, while the input phase angle was varied from $\angle V_{in} = 7.2^\circ$ to 352.8° in increments of 7.2° . This approach provided 50 distinct phase points, avoiding physical duplication of the 0° and 360° states. Much like the amplitude tests, ten measurements were made at each phase configuration to quantify measurement repeatability and noise effects.

3.2.3 Data Acquisition

The ZCU216 board captures the ADC output, whereas the input parameters (V_{in} magnitude and phase) are commanded digital settings that set the input voltage and phase. The acquired discrete-time samples are sent to the MATLAB program, which subsequently transforms them into the frequency domain using Matlab's own Fast Fourier Transform (FFT) function. For the commanded input, the variables are:

- input voltage magnitude ($|V_{in}|$)
- input phase angle ($\angle V_{in}$)

For the measured output, MATLAB extracts:

- output voltage magnitude ($|V_{out}|$)
- output phase angle ($\angle V_{out}$)

Before the FFT is applied, the raw ADC samples are multiplied by a Hann window to reduce spectral leakage. To compensate for the amplitude attenuation introduced by the windowing function, an amplitude correction factor is applied (defined as the window length divided by the sum of the Hann window coefficients). Notably, there is no explicit hardware-to-volts conversion (e.g., LSB-to-voltage scaling based on V_{ref}) applied in the data extraction scripts; the normalized digital amplitude obtained from the processed FFT is implicitly treated as the absolute voltage ($|V_{out}|$). The windowed data is transformed to the frequency domain and converted into a single-sided amplitude spectrum. Normalization is achieved by multiplying the absolute value of the FFT output by 2 and dividing by the total number of samples N (with the DC bin scaled separately by $1/N$). The fundamental tone is identified via peak detection. The single-sided amplitude spectrum is converted to decibels, and the frequency bin corresponding to the maximum amplitude is selected. Prior to extraction, the script accounts for the active Nyquist zone; for even Nyquist zones (e.g., Zone 2, Zone 4), the amplitude and phase arrays are flipped and the frequency axis is inverted relative to the sampling rate to accurately map the aliased frequency back to its true physical frequency. The phase angle ($\angle V_{out}$) is extracted from the complex FFT output at the selected peak frequency bin. It

is calculated using a standard four-quadrant arctangent function and converted to degrees. The phase strictly remains wrapped within the standard boundaries of -180° and 180° (no unwrapping algorithm is applied). But this is later corrected to a $0^\circ - 360^\circ$ in the plotting.

3.2.4 Visualization

The measurement results were visualized with a separate MATLAB script. A total of four plots were generated to illustrate the non-linearity in the DAC-to-ADC system:

- Magnitude transfer ($|V_{in}|$ to $|V_{out}|$) with $\angle V_{in} = 0^\circ$
- Phase-versus-amplitude ($|V_{in}|$ to $\angle V_{out}$) with $\angle V_{in} = 0^\circ$
- Magnitude-versus-input-phase ($\angle V_{in}$ to $|V_{out}|$) with $|V_{in}| = 1.0$
- Phase transfer ($\angle V_{in}$ to $\angle V_{out}$) with $|V_{in}| = 1.0$

Each measurement configuration was repeated ten times to visualize the measurement noise across the loop-back system.

3.2.5 Dense Grid Measurement mapping for $T0D1 \rightarrow T0A1$

In the previous sections, we evaluated how one input ($|V_{in}|$ or $\angle V_{in}$) changes one output ($|V_{out}|$ or $\angle V_{out}$). To fully characterize the transfer mapping, we must analyze the joint effect of both inputs on the outputs. This was achieved by mapping the inputs over a two-dimensional LHS grid, where phase values form the columns and amplitude values form the rows, ensuring an even distribution of test coordinates across the parameter space.

Two separate 3D surfaces visualize the two-output mapping, where one of the outputs is the z -axis and the two inputs form the x - and y -axes. Three measurements were made at each coordinate to estimate repeatability and measurement noise; these replicates were subsequently averaged before implementing the Kriging model. Specifically, 28 distinct phase sample points and 16 amplitude points were measured taken using LHS, resulting in a 28×16 grid yielding 448 unique input combinations and 1,344 total captures.

3.3 Regression Model

3.3.1 Implementation of GPR

To define, create, and fit the regression model in this thesis, MATLAB was used. MATLAB requires the Statistics and Machine Learning Toolbox to perform Gaussian Process Regression (GPR). The `fitrgp` function was the principal tool used.

While `fitrgp` accepts an N -by- p predictor matrix, modeling our transfer behavior required mapping the two predictors (input voltage magnitude $|V_{in}|$ and input phase angle $\angle V_{in}$) to scalar responses. Therefore, two separate models were fitted independently for $|V_{out}|$ and $\angle V_{out}$.

3.3.2 Data Scaling and Basis Functions

A critical challenge in modeling the multi-dimensional hardware response is the disparity in numerical scales between the input parameters. The amplitude inputs are bounded between 0 and 1.0 V, while the phase input coordinate spans from 0° to 360° . Because the phase coordinate spans a much larger numerical range, appropriate predictor scaling is necessary so one dimension does not artificially dominate the distance metric computation within the kernel. A specific basis function was selected for each response. When modeling the amplitude output $|V_{\text{out}}|$, a pure quadratic basis function was implemented to capture non-linear, curvature-dependent trends across the input amplitude range:

```
gprMdl_amp = fitrgp(X_train, Z_train_amp, 'KernelFunction', ...
'squaredexponential', ...
'Standardize', true, 'BasisFunction', 'pureQuadratic');
```

On the other hand, when mapping the phase output $\angle V_{\text{out}}$, a linear basis function was used to capture the parameter space:

```
gprMdl_phase = fitrgp(X_train, Z_train_amp, 'KernelFunction', ...
'squaredexponential', ...
'Standardize', true, 'BasisFunction', 'linear');
```

3.3.3 Visualization of the Kriging Field

To effectively present the multidimensional regression results without the visual clutter of overlapping boundaries, the Kriging outputs were separated into two distinct spatial subplots. The visualization structure was rendered using a continuous color map and consisted of two primary surfaces: The first surface is the Kriging Mean Field, which displays the expected output response given the two variable inputs. The second surface represents the 95% prediction interval width, which displays a graphical map of the model's uncertainty.

By isolating the interval width as an independent surface, regions of high uncertainty—such as gaps between data points or outer boundaries—appear as peaks. This approach provides a clear, comprehensive mapping of both the DAC-to-ADC system behavior and the predictive model's uncertainty.

3.3.4 Cross-Validation and Predictive Performance

To evaluate the performance of the GPR models and ensure the metrics reflect out of sample accuracy rather than in sample training fit, k fold cross validation was implemented. The dataset was partitioned using 10-fold cross-validation ($k = 10$). In this approach, the data is randomly divided into ten equal-sized subsamples. The model is trained on nine of the subsamples and tested on the remaining single hold-out subsample. This process is repeated ten times, with each subsample used exactly once as the validation data. In MATLAB, this was executed using the `crossval` function to create the partitioned models, followed by the `kfoldPredict` function to generate the out-of-sample predictions. The cross-validated predictions were then

used to calculate two primary performance metrics: the coefficient of determination (R^2) and the Root Mean Square Error (RMSE). The R^2 metric quantifies the proportion of the variance in the dependent variable that is predictable from the independent variables. To calculate the cross-validated R^2 , the following equation was used:

$$R^2 = 1 - \frac{\sum_{i=1}^N (z_i - \hat{z}_i)^2}{\sum_{i=1}^N (z_i - \bar{z})^2} \quad (3.1)$$

where N is the total number of observations, z_i is the actual measured response (representing either $|V_{out}|$ or $\angle V_{out}$ depending on the model), $\hat{z}_i$ is the out-of-sample predicted response from the cross-validated model, and $\bar{z}$ is the mean of the observed data. The RMSE provides a measure of the average magnitude of the prediction errors, reported in the same units as the response variable. It is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (z_i - \hat{z}_i)^2} \quad (3.2)$$

By utilizing the cross-validated predictions ($\hat{z}_i$) rather than the training-fit predictions in these formulas, the resulting metrics provide a robust assessment of how well the hardware behavioral mapping generalizes to unseen parameter configurations without the optimistic bias of overfitting to measurement noise.

4

Results & Discussion

4.1 Mapping DAC-to-ADC Transfer Characteristics

The results of the DAC-to-ADC loopback characterization across the seven evaluated channel combinations are presented in Figures 4.1a through 4.14b. For each hardware configuration, the data is separated into amplitude transfer characteristics (capturing $|V_{in}|$ to $|V_{out}|$ and $|V_{in}|$ to $\angle V_{out}$) and phase transfer characteristics (capturing $\angle V_{in}$ to $|V_{out}|$ and $\angle V_{in}$ to $\angle V_{out}$).

The plotted data includes the ten repeated measurements at each configuration step. Across the measured combinations, the amplitude-to-amplitude and phase-to-phase responses demonstrate the baseline behavior of the data-converter system, while the amplitude-to-phase and phase-to-amplitude cross-plots highlight the inherent phase-amplitude dependencies present at 3 GHz through paths using the 1-4 GHz balun ports.

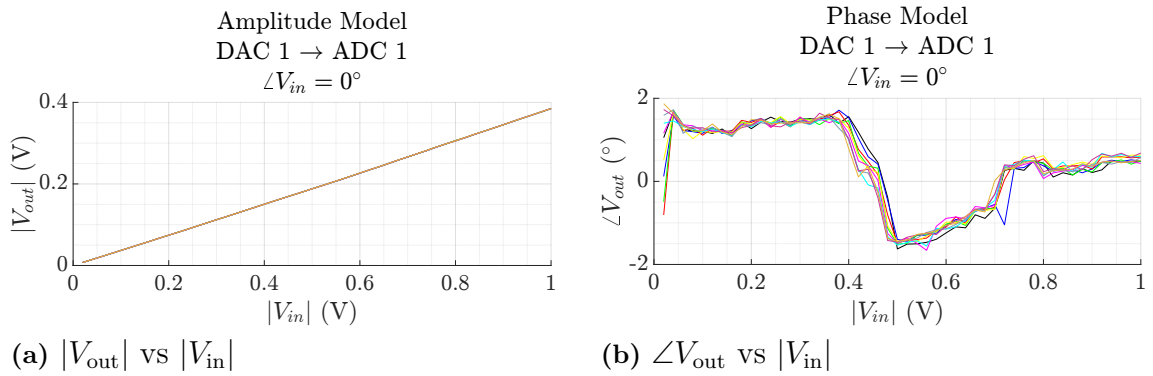


Figure 4.1: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC1

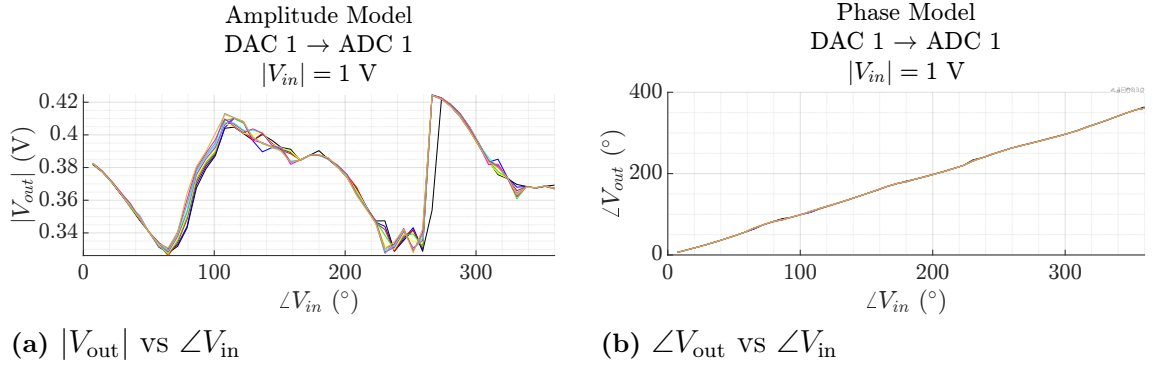


Figure 4.2: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC1 → ADC1

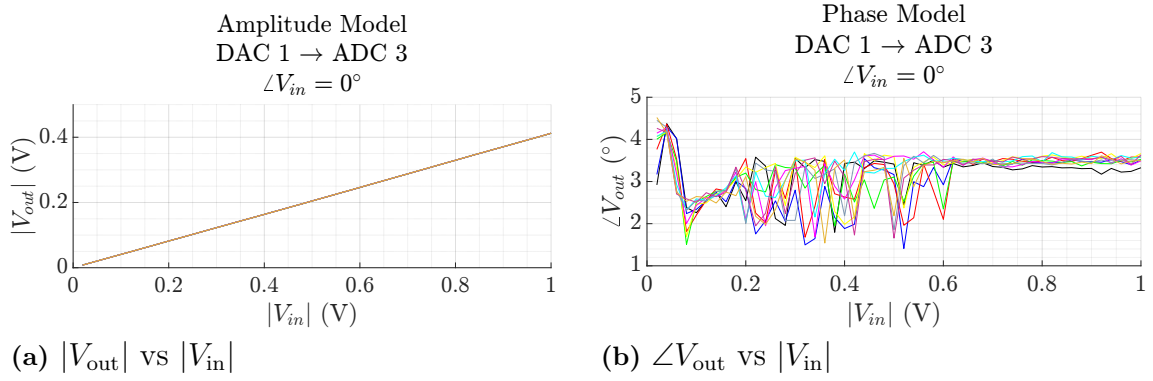


Figure 4.3: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC1 → ADC3

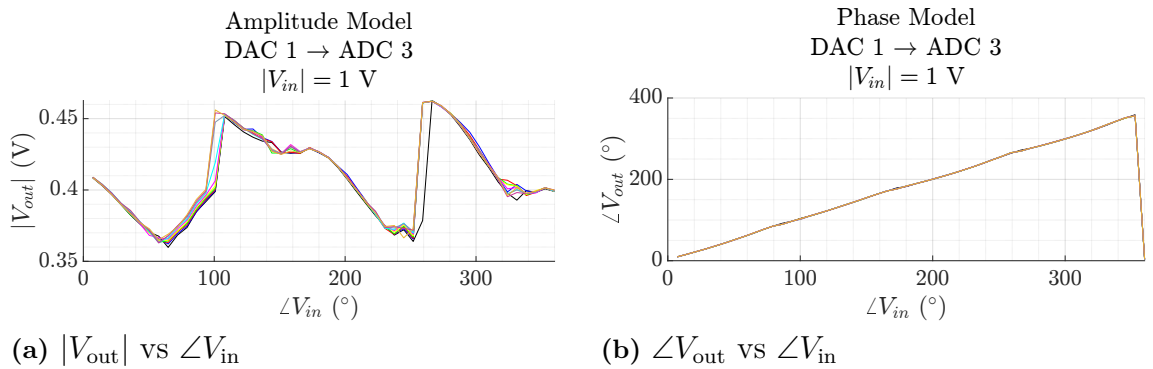


Figure 4.4: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC1 → ADC3

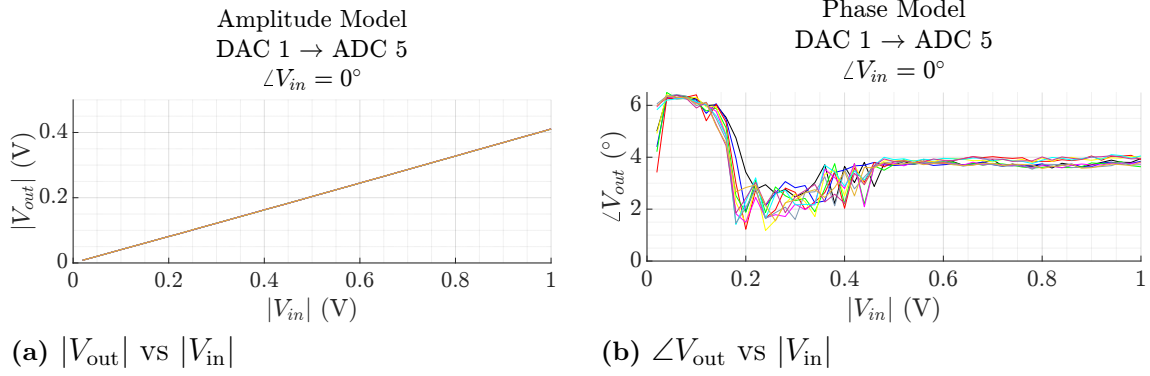


Figure 4.5: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC5

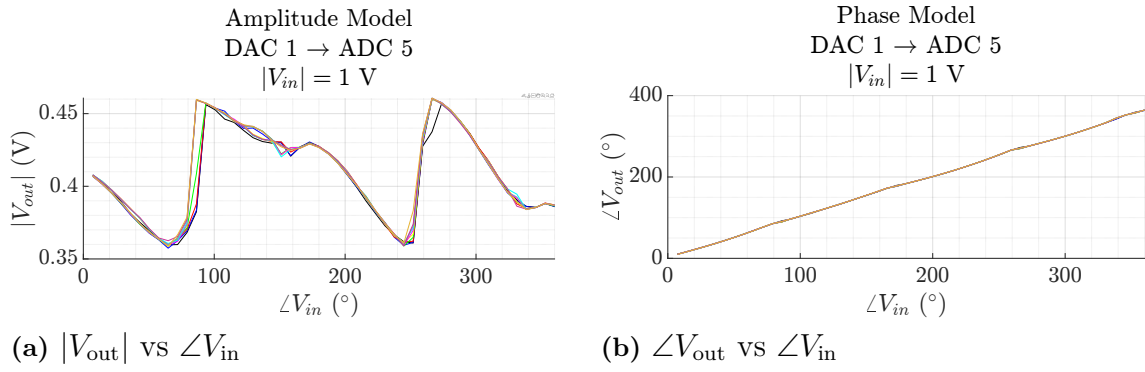


Figure 4.6: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC1 $\rightarrow$ ADC5

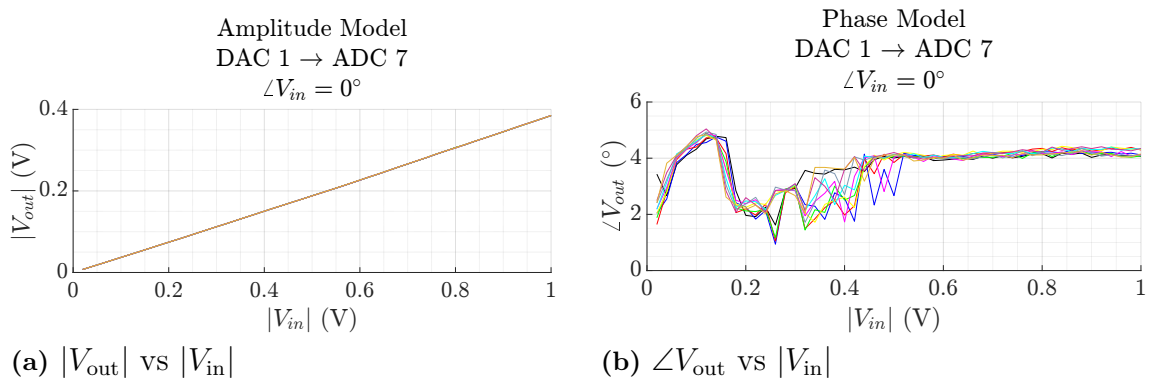


Figure 4.7: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC1 $\rightarrow$ ADC7

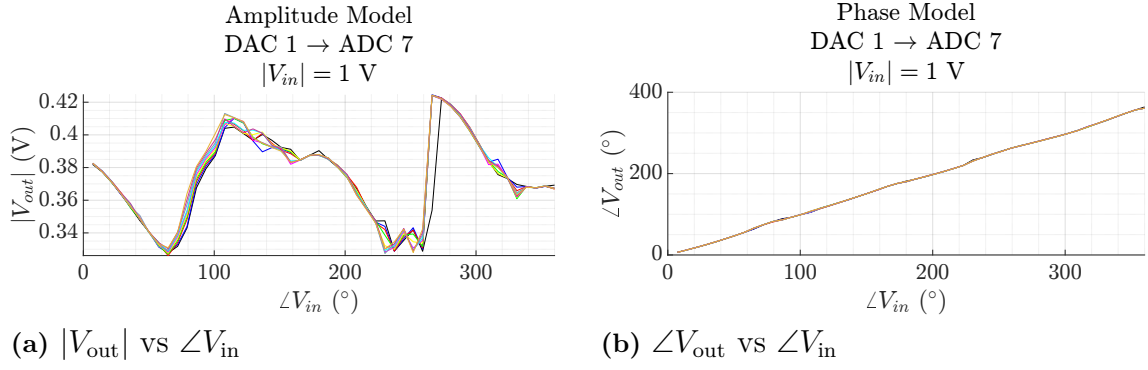


Figure 4.8: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC1 → ADC7

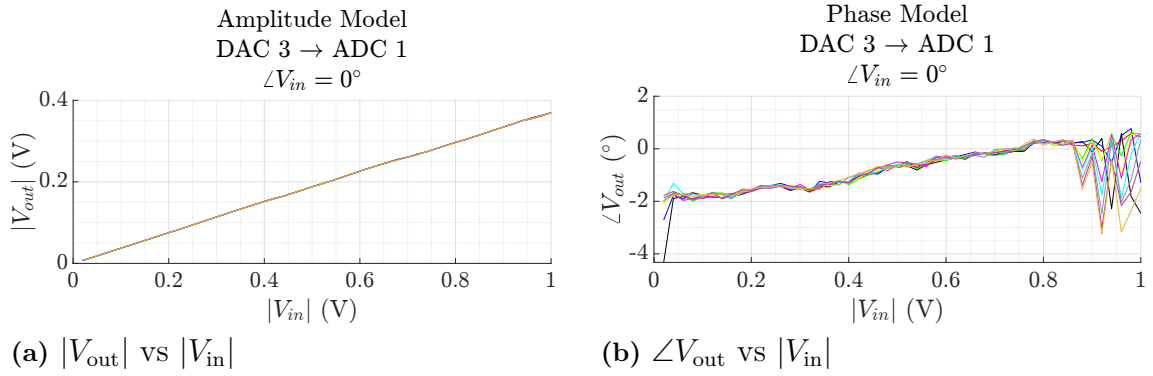


Figure 4.9: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC3 → ADC1

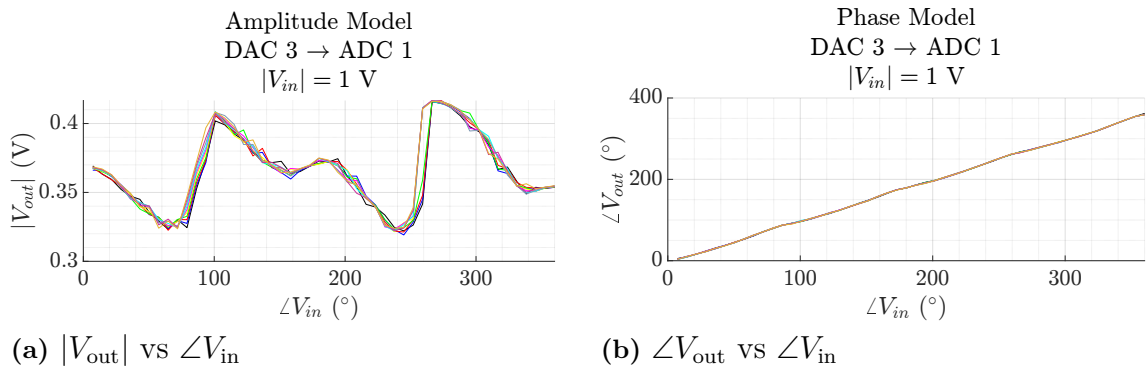


Figure 4.10: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC3 → ADC1

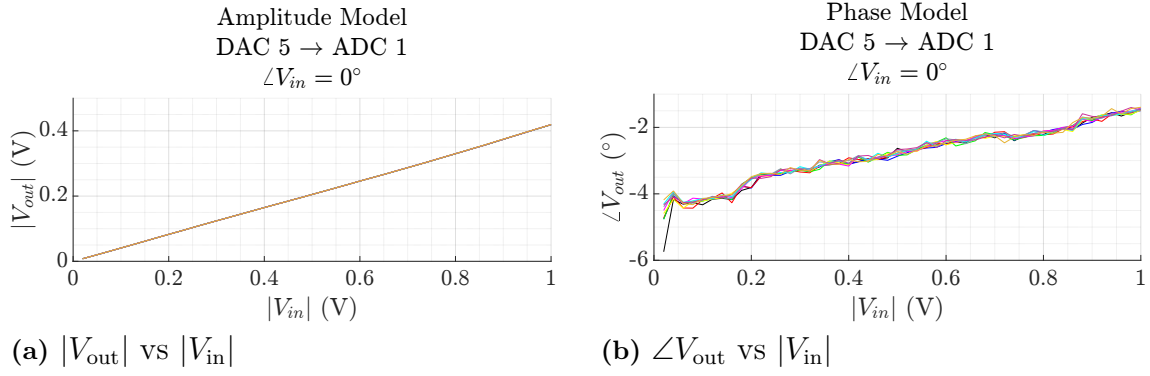


Figure 4.11: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC5 → ADC1

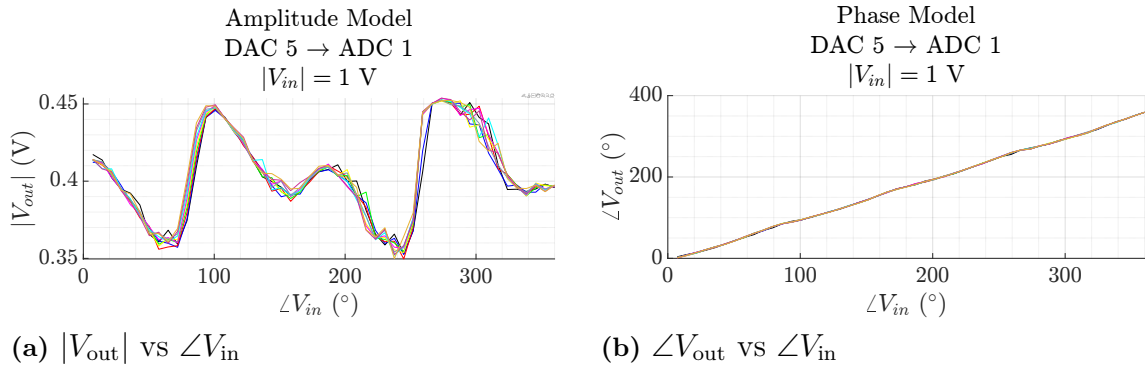


Figure 4.12: $|V_{out}|$ and $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC5 → ADC1

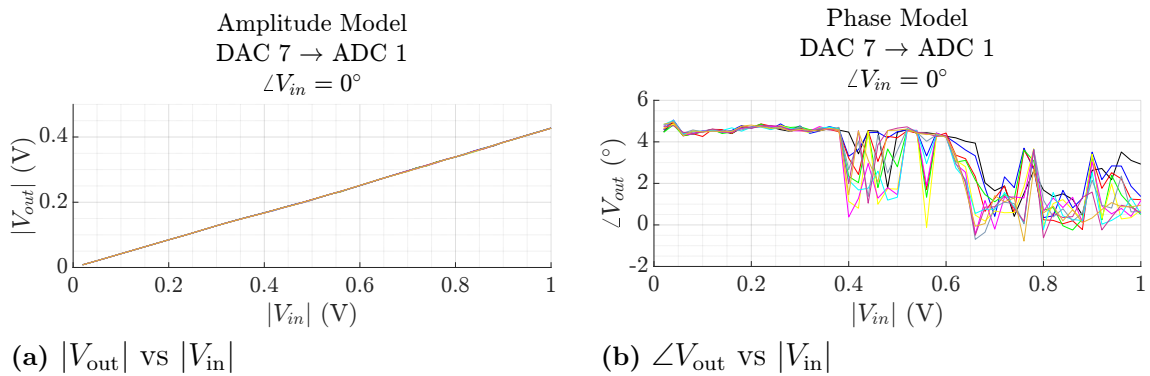


Figure 4.13: $|V_{out}|$ and $\angle V_{out}$ as a function of $|V_{in}|$ with a constant phase of $\angle V_{in} = 0^\circ$ for loopback of DAC7 → ADC1

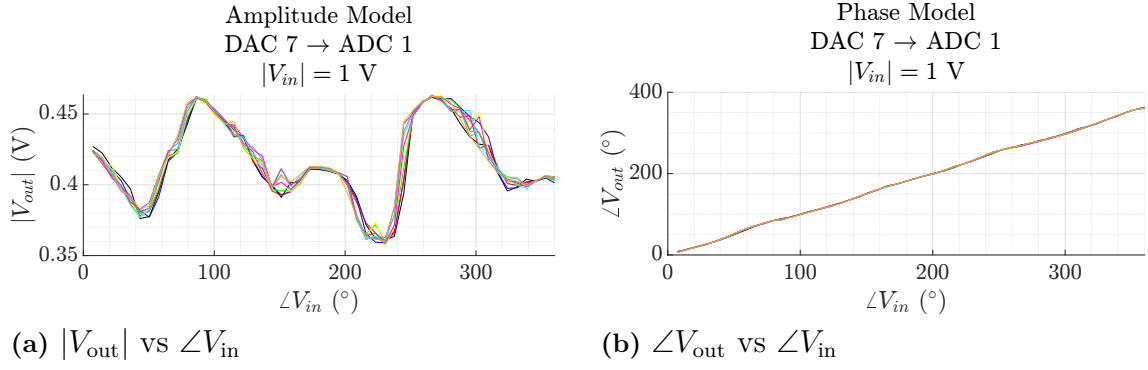


Figure 4.14: $|V_{out}|$ & $\angle V_{out}$ as a function of $\angle V_{in}$ with a constant amplitude of $|V_{in}| = 1.0$ for loopback of DAC7 → ADC1

In the figures depicted above, deviations from an ideal linear response can be evaluated by defining an ideal calibrated response and examining the residuals. This behavior is especially noticeable on the phase response plots given a constant input phase of $\angle V_{in} = 0^\circ$, as well as on the amplitude response plots given a constant amplitude of $|V_{in}| = 1.0$. It can also be observed that the magnitude-versus-input-phase plots across different channels all depict the same periodic shape. This common shape provides evidence for a shared systematic effect; further investigation of the digital tone generator, sample timing, FFT windowing, and shared analog paths is required before attributing this ripple strictly to hardware nonlinearity.

4.2 Two-dimensional mapping of DAC1-to-ADC1

The two-dimensional mapping of DAC1 to ADC1 for the phase response is shown below:

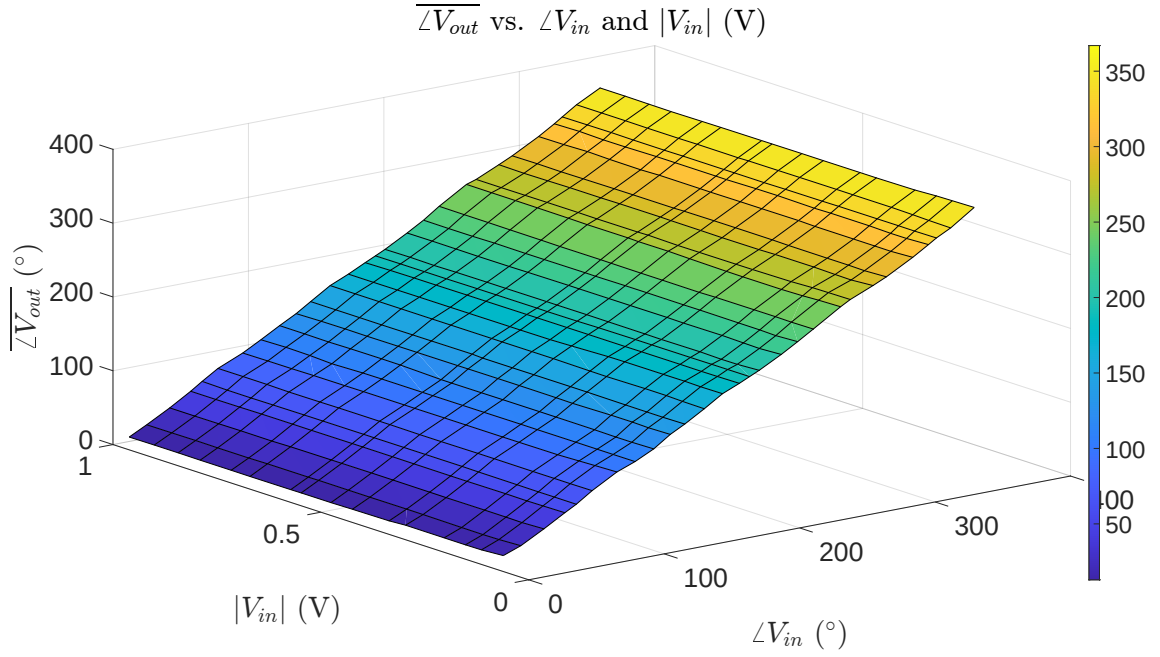


Figure 4.15: Phase output as a function of input phase and amplitude for DAC1 $\rightarrow$ ADC1

The figure above depicts that the output phase tracks the input phase strongly and appears only weakly dependent on amplitude. For the output amplitude, the mapping is as follows:

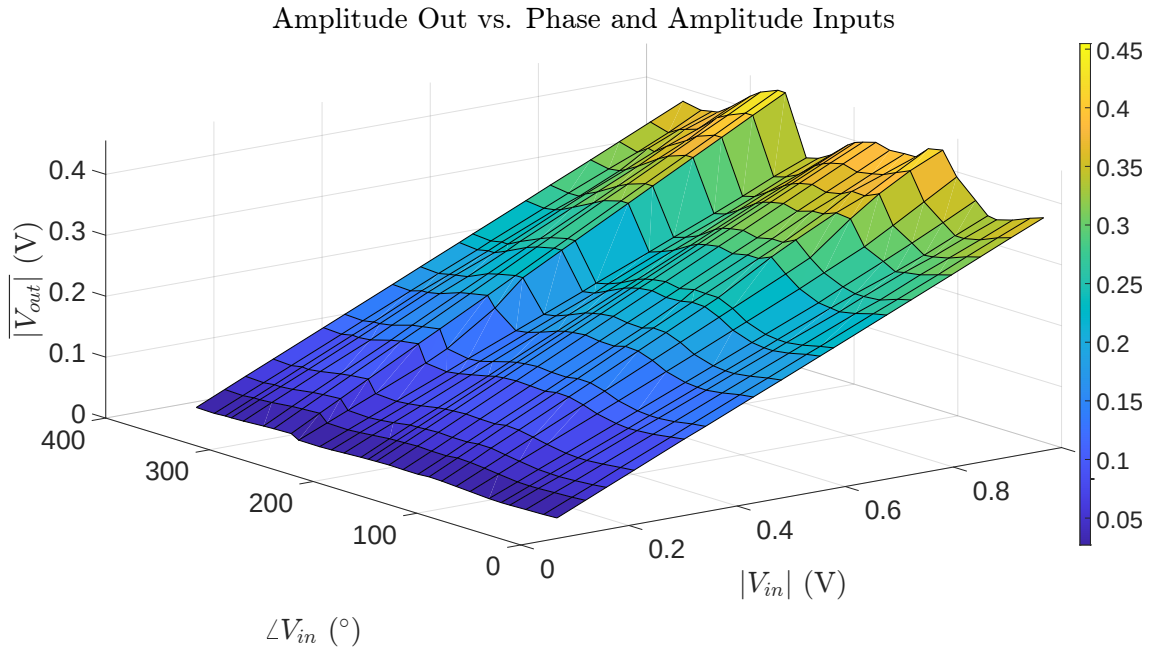


Figure 4.16: Amplitude output as a function of input phase and amplitude for DAC1 $\rightarrow$ ADC1

It can be observed that the output amplitude depends on both the input phase and the input amplitude. At an input magnitude of 1.0, the shape looks similar

to Figure 4.2a. It can also be observed that for lower input voltages, the output amplitude does not fluctuate as much. However, utilizing a lower DAC amplitude to avoid distortion reduces the signal to noise ratio (SNR) and dynamic range, which requires additional downstream gain. Because the system represents a static two-input coordinate mapping, alternative modeling approaches such as look-up tables (LUTs), splines, or low-order complex polynomials could also be considered to invert the mapping. Model selection must ultimately be based on held-out error, robustness, complexity, and implementation cost.

4.3 GPR model for the two-dimensional DAC1-to-ADC1 mapping

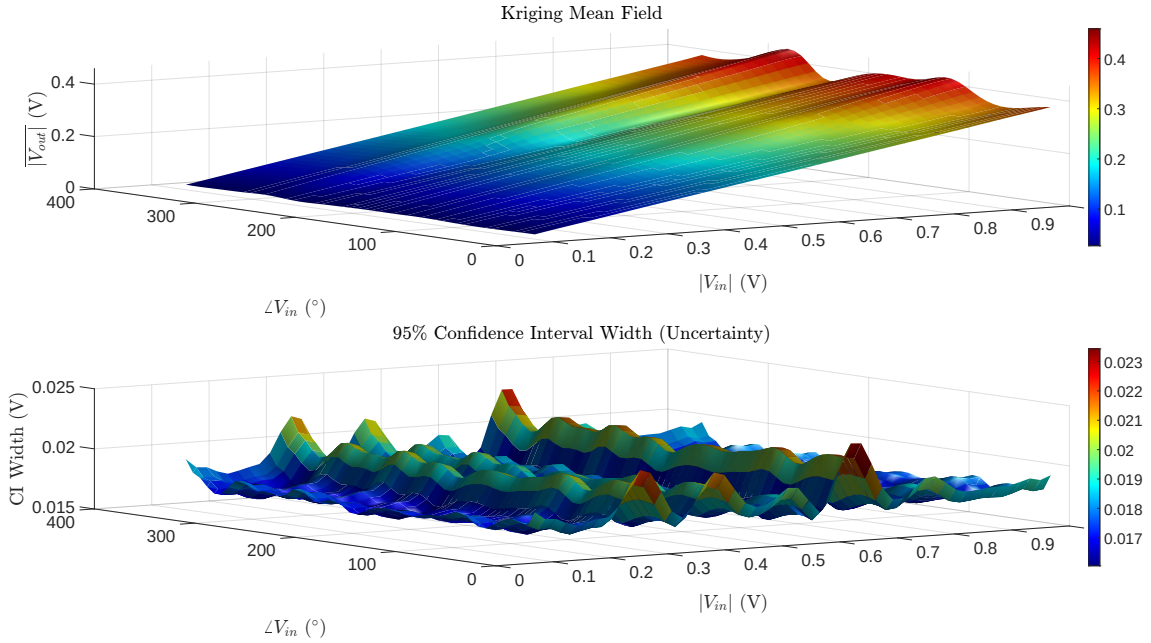


Figure 4.17: The Amplitude field in the plot above and the 95% prediction interval width below, with the z-axis showing the difference between the upper and lower limits of the 95% interval.

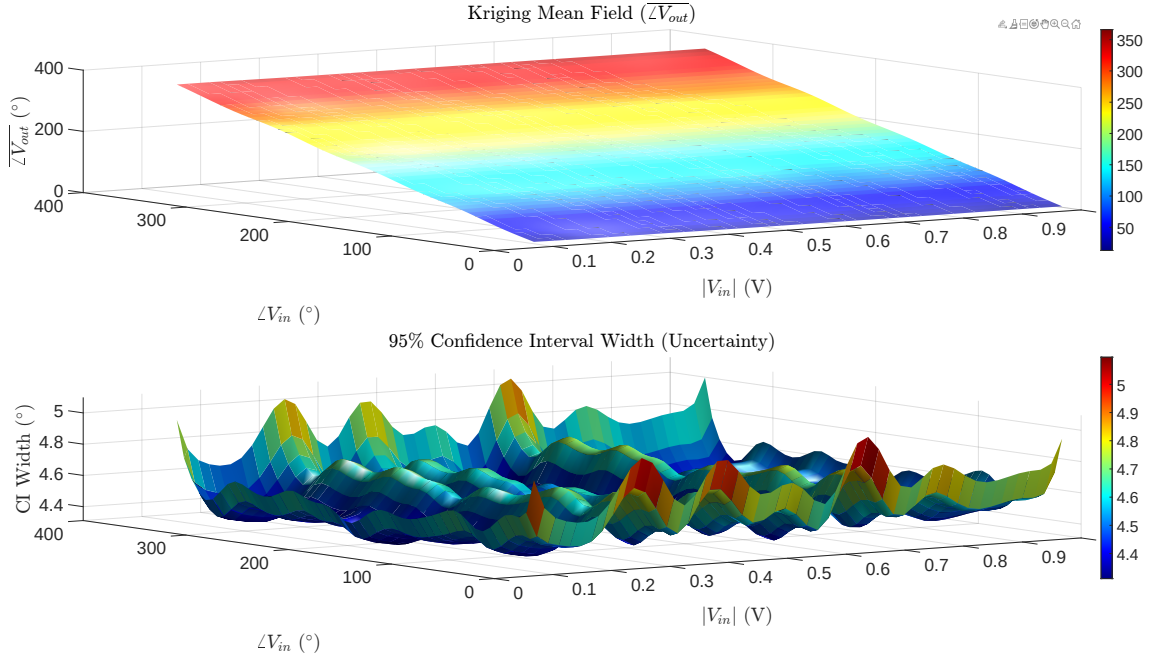


Figure 4.18: The Phase field in the plot above and the 95% prediction interval width below, with the z-axis showing the difference between the upper and lower limits of the 95% interval.

It can be observed that the interval width increases where the sampling density is lower. Which can be backed up by plotting the heatmap of both the prediction interval width as well as the 2 dimensional mapping with all of its individual sampled points to see the density.

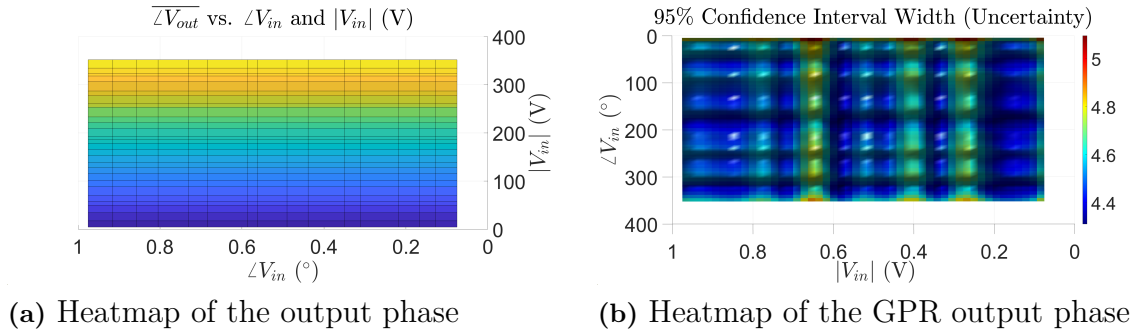


Figure 4.19: Comparison between heatmap of the phase field and confidence interval of the GPR phase. The lack of even sampling in the amplitude direction can be noticed on the GPR model. At points 0.3 V, 0.4 V and 0.65 V

The results of the Kriging application are visualized in Figures 4.17 and 4.18. In each figure, the upper plot shows the posterior mean estimate (μ_*). This represents the expected output amplitude or phase across the continuous input domain. The regression successfully provides interpolated predictions between measurements on a 50×50 query grid. A separate inversion step allows the extraction of denser pre-compensation coordinates, which can then be applied to the baseband digital signal

to correct the hardware’s inherent distortion before it reaches the antenna. The predictive uncertainty is visualized in the lower plots of Figures 4.17 and 4.18. These plots display the 95% prediction interval width, which is directly proportional to the predictive standard deviation (σ_*) of the model at any given coordinate. Because GPR computes the predictive variance based on the complete covariance matrix, the noise model, kernel hyperparameters, and input geometry, the uncertainty fundamentally increases in regions furthest from the measured training data. This trend should be assessed by comparing the interval width with the displayed sampling locations.

To validate the predictive performance of the models, 10-fold cross-validation was performed. The cross-validated coefficient of determination (R^2) and root mean square error (RMSE) were calculated to evaluate out-of-sample accuracy. The amplitude model achieved a cross-validated R^2 of $R_{\text{amp}}^2 = 0.998$ and an RMSE of 0.0050 V. For the phase model, the cross-validated metrics were $R_{\text{phase}}^2 = 0.99$ and an RMSE of 1.15°.

5

Conclusion

The project characterized and mapped the non-linear effects of the Xilinx Zynq UltraScale+ ZCU216 RFSOC development board's data converters at 3 GHz and evaluated a GPR model. A static loop-back configuration was successfully constructed and measured using the XM655 breakout board. Rather than isolating specific amplitude and phase distortions, the work measured the complete uncalibrated loop-back cascade. From the empirical results, the cause of the phase-dependent amplitude variation remains unresolved. These characteristics allowed for the creation of a comprehensive two-dimensional non-linear transfer map utilizing Latin Hypercube Sampling. It was shown that a probabilistic GPR model trained on this dataset was able to successfully interpolate the hardware's two-dimensional static amplitude and phase response surfaces, functioning strictly as a forward interpolation model. It also produced a dynamic 95% prediction interval for predictive uncertainty tracking. Ultimately, neither inverse DPD nor antenna-array performance was demonstrated.

To build upon these findings, future research must address the current hardware implementation constraints through a structured framework. The next steps should focus on establishing controlled measurements alongside independent validation points to verify model accuracy. As well as perform a separate characterization of the observation ADC and transmit DAC paths to isolate component-specific non-linearities. Once these behaviors are uncoupled, efforts can shift toward full inverse-model construction and its subsequent real-time realization via a Look-Up Table or FPGA implementation. Finally, successful deployment on the physical array will necessitate five-channel calibration, dynamic antenna-load testing and beam-pattern validation to confirm operational performance.

Bibliography

- [1] Keith Benson. “Phased Array Beamforming ICs Simplify Antenna Design”. In: *Analog Digital Magazine Vol 53* (2019). URL: <https://www.analog.com/en/resources/analog-dialogue/articles/phased-array-beamforming-ics-simplify-antenna-design.html>.
- [2] L.H. Crockett, D. Northcote, and R. W. Stewart (Editors). *Software Defined Radio with Zynq UltraScale+ RFSoc*. First Edition. Strathclyde Academic Media, 2023. URL: info@RFSocbook.com.
- [3] National Instruments. “Advantages of Direct RF Sampling Architectures”. In: *National Instruments* (Nov. 2024).
- [4] Franco Maloberti. *Data Converters*. en. 2007th ed. New York, NY: Springer, Jan. 2007.
- [5] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (May 1979), p. 239. ISSN: 0040-1706. DOI: 10.2307/1268522. URL: <http://dx.doi.org/10.2307/1268522>.
- [6] Charles L Phillips, John Parr, and Eve Riskin. *Signals, Systems, and Transforms*. 4th ed. Upper Saddle River, NJ: Pearson, July 2007, p. 74.
- [7] Robin Preece and Jovica Milanović. “Efficient Estimation of the Probability of Small-Disturbance Instability of Large Uncertain Power Systems”. In: *IEEE Transactions on Power Systems* 31 (Apr. 2015), pp. 1–10. DOI: 10.1109/TPWRS.2015.2417204.
- [8] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. en. Adaptive Computation and Machine Learning series. London, England: MIT Press, Nov. 2005.
- [9] Bill Schweber. “Understanding the RF Balun and its Transformative Function”. In: *DigiKey* (2015).
- [10] Keith Shaw. *What is beamforming and how does it make wireless better?* URL: <https://www.networkworld.com/article/967954/beamforming-explained-how-it-makes-wireless-communication-faster.html>. (accessed: 05.07.2025).

- [11] William E. Stein, Roger C. Pfaffenberger, and Dan W. French. “SAMPLING ERROR IN FIRST ORDER STOCHASTIC DOMINANCE”. In: *Journal of Financial Research* 10.3 (Sept. 1987), pp. 259–268. ISSN: 1475-6803. DOI: 10.1111/j.1475-6803.1987.tb00496.x. URL: <http://dx.doi.org/10.1111/j.1475-6803.1987.tb00496.x>.
- [12] Jie Wang. “An Intuitive Tutorial to Gaussian Process Regression”. In: *Computing in Science and Engineering* 25.4 (July 2023), pp. 4–11. ISSN: 1558-366X. DOI: 10.1109/mcse.2023.3342149. URL: <http://dx.doi.org/10.1109/MCSE.2023.3342149>.
- [13] Wikipedia. *Kriging* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Kriging&oldid=1277903866>. [Online; accessed 21-May-2025]. 2025.
- [14] *ZCU216 Evaluation Board User Guide (UG1390)*. available at <https://docs.amd.com/r/en-US/ug1390-zcu216-eval-bd/Introduction>. Advanced Micro Devices. 2023.

A

Appendix

Generative AI usage in the project was done with the intent to aid in programming and grammatical errors. Gemini wrote Matlab code for the design of the GPR model as well as plotting of the model. The AI went through the text in order to check for grammatical errors.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY