2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
MUNICH, GERMANY
OCTOBER 14-15, 2025

DMS Verification Environment for Gyroscope

Thierry Nouguier, Ajay G Nayak, Fabrice Boissieres

NXP

accellera SYSTEMS INITIATIVE

Agenda

- **Introduction and Motivation**
- MEMS Gyroscope System
- Modeling Strategy
- Use Case: Safety Mechanism Verification
- Conclusion
- QA

Motivation and Context

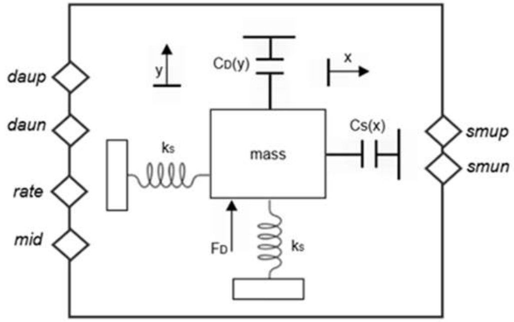
- Digital Mixed-Signal Simulation
  - DMS environments use modeling language (wreal/systemverilog) for fast simulation and improved test coverage
- Need for Analog Mixed-Signal
  - AMS simulations are required for high-accuracy applications like MEMS gyroscopes despite slower performance
- Performance and Scalability Limits
  - AMS simulations face challenges in speed and scalability for complex system-level verification
- Shifting AMS to Digital Domain
  - Enhanced DMS techniques aim to migrate AMS workload digitally for faster yet accurate verification.

Agenda

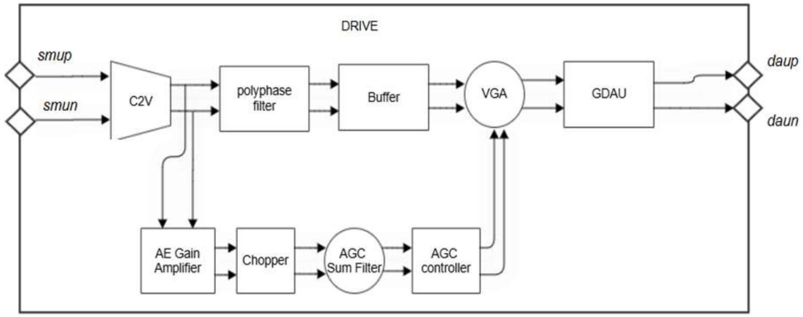
- Introduction and Motivation
- **MEMS Gyroscope System**
- Modeling Strategy
- Use Case: Safety Mechanism Verification
- Conclusion
- QA

Structure and Function

- Vibrating Proof Mass
  - The MEMS gyroscope uses a vibrating proof mass suspended by microfabricated springs to detect angular velocity
- Coriolis Force detection
  - Rotation causes Coriolis forces that deflect the mass along a sense axis, detected by capacitive sensing



- ASIC Signal Processing
  - An ASIC drive vibrating, amplifies signals, demodulates and converts data digitally, managing calibration and communication
- System Integration
  - Precise modeling ensures accurate behavior and this tightly integrated MEMS gyroscope system



Modeling Challenges in DMS

- Analog and Mechanical complexity
  - MEMS devices exhibit resonance, damping and thermal drift, complicating SystemVerilog modeling due to analog and mechanical effects
- Lack of Multi-domain simulation
  - SystemVerilog does not support multi-domain and phase accurate simulation, creating challenges in closed-loop DMS modeling
- Noise and Parasitics Limitation
  - SystemVerilog struggles to simulate noise and parasitics effects, impacting high precision sensing performance in DMS

Agenda

- Introduction and Motivation
- MEMS Gyroscope System
- **Modeling Strategy**
- Use Case: Safety Mechanism Verification
- Conclusion
- QA

Enhancing Simulation speed & Accuracy with Custom UDNs

```
// user-defined resolution function VIR
function automatic elec VIR  (input elec driver[]);
  real G=1e-18;    // Total conductance for node
  real Idiff=0;// Total of net drivers current contributions for node

  foreach (driver[i]) begin

          Idiff += driver[i].V/(driver[i].R + 1e-18); //added 1e-18 ohm minimum resistance
          G +=1/(driver[i].R + 1e-18) + 1e-18; //for convergence issue avoidance
          VIR.R = 1/G;
          VIR.I += driver[i].I;
  end

  VIR.V = Idiff * VIR.R;

endfunction
```

enet resolution function

- Resolution function simplification
- Focus exclusively on Kirchhoff's Current law to modelize our analog blocks

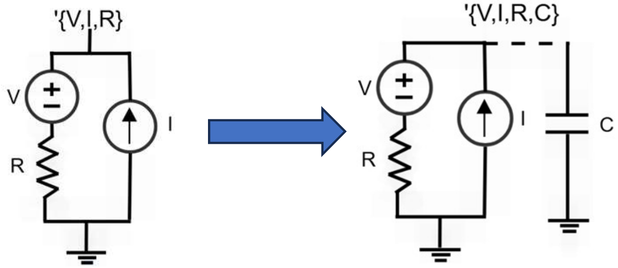
SIMULATION METHOD	SIMULATION TIME (S)	SPEED-UP
Spectre (VAMS)	256	x1
EEnet	129	x2
enet	24	x10

Performance on RC filter

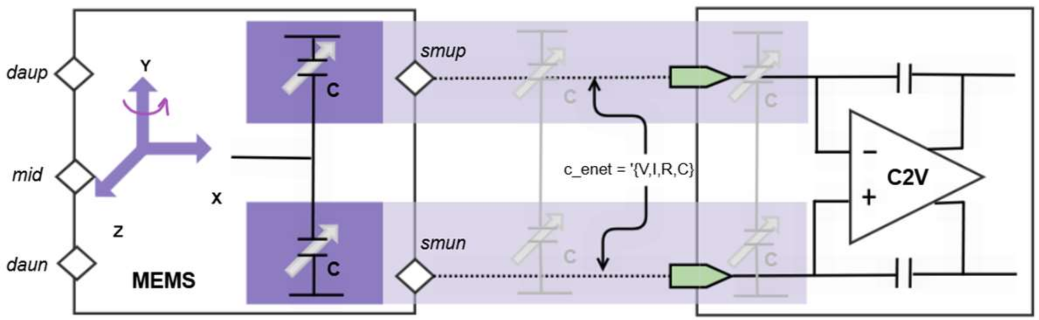
- Maintains accuracy (0.1%)
- Improves performance

Capacitance modeling with c_enet

- Introduction of c_enet
  - c_enet extends enet to include voltage, current, impedance, and capacitance for better signal representation



- Improved MEMS-ASIC Modeling
  - c_enet enables accurate modeling at the MEMS-to-ASIC interface, preserving mechanical signal integrity



Multi-Rate Sampling strategy

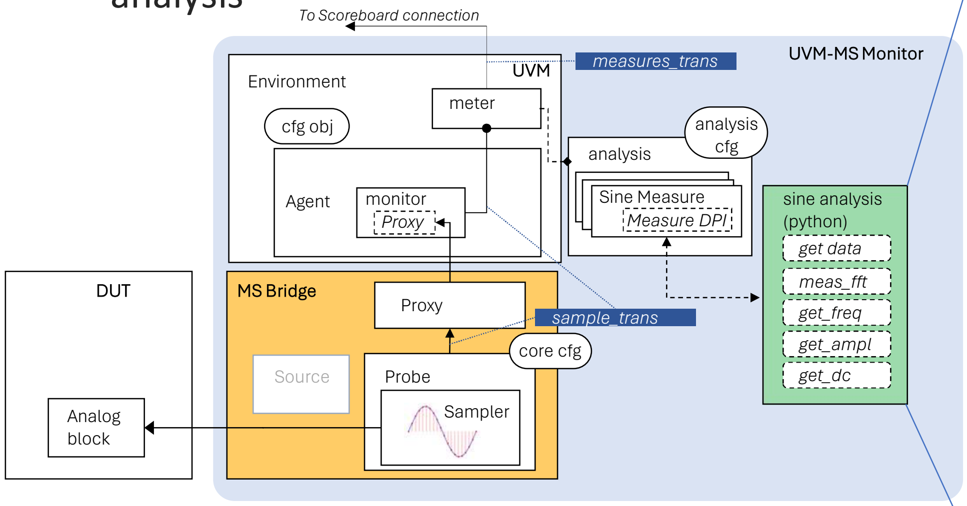
- High-Resolution MEMS Simulation
  - MEMS systems require high-frequency simulation at 0.1ns for accurate phase tracking and performance
- Coarser ASIC Sampling Rate
  - ASIC subsystems use a coarser sampling rate at 1ns to reduce computational demands while maintaining adequate accuracy
- Unified Testbench Validation
  - A unified testbench validates the multi-rate sampling methodology ensuring coherent and reliable simulation results

Advanced Signal Analysis with Python DPI

- UVM-MS testbench with passive-mode UVCs
  - The testbench uses passive mode UVCs for accurate signal monitoring without driving signals
- Python DPI for Signal Processing
  - Analog signals are sampled and analyzed in Python using DPI with FFT, phase extraction, and amplitude analysis
- Python DPI based on extended pyhdl-if Package
  - Pyhdl-if package extended to support float and double types for seamless data type integration.

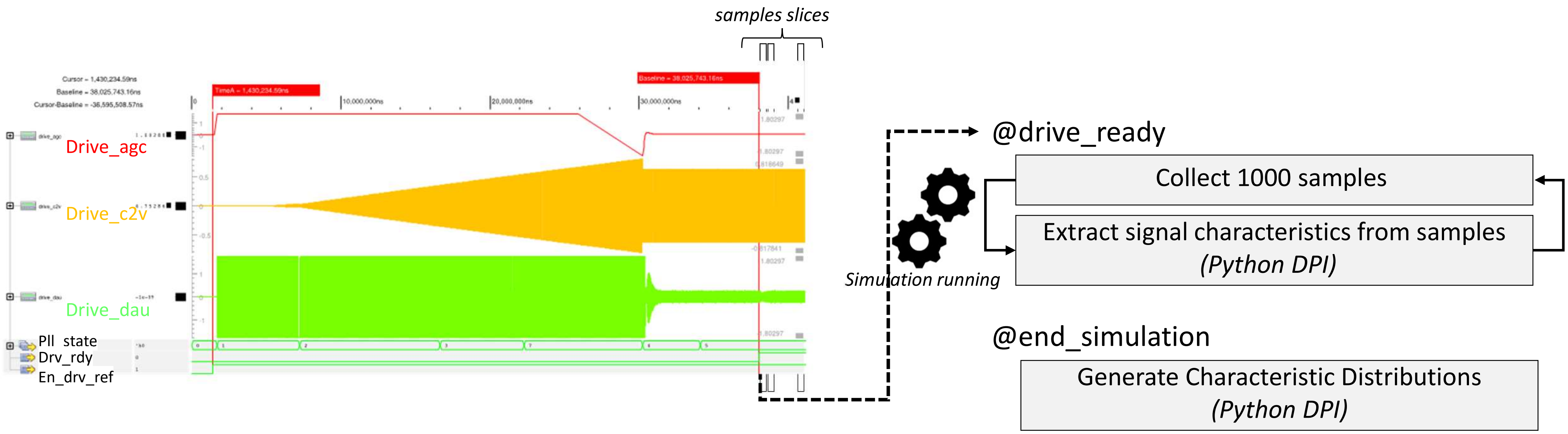
Python DPI-Based Signal Analysis – UVC Architecture Overview

UVM-MS Monitor (UVC Sine Monitor) architecture with DPI and Python-based signal analysis




```python
import ctypes
import numpy as np
from nfft import nfft
import matplotlib.pyplot as plt
import hdl_if as hif
from scipy.fft import fft,fftfreq
from scipy.signal import hilbert

@hif.api
class sine_analysis(object):

    @hif.exp
    def init_proc(self):
        self.list_of_values=[]
        self.measure = Measure()
    pass

    @hif.exp
    def get_data(self, t: ctypes.c_double, v : ctypes.c_double ) :
        self.list_of_values.append((t,v))
    pass

    @hif.exp
    def meas_fft(self, sample_rate : ctypes.c_double, plot : ctypes.c_int):
        list_x,list_y=zip(*self.list_of_values)
        self.measure.measure_fft(list_x,list_y, sample_rate, plot=plot)

    @hif.exp
    def get_freq(self) -> ctypes.c_double:
        return self.measure.Freq


    @hif.exp
    def get_amplitude(self) -> ctypes.c_double:
        return self.measure.Amplitude

    @hif.exp
    def get_dc(self) -> ctypes.c_double:
        return self.measure.DC

    @hif.exp
    def get_phase(self) -> ctypes.c_double :
        list_x,list_y,list_z=zip(*self.list_of_values)
        phase=self.measure.coupling_angle_hilbert(list_x[10:],list_y[10:],list_z[10:],pad=True)
        return phase
```

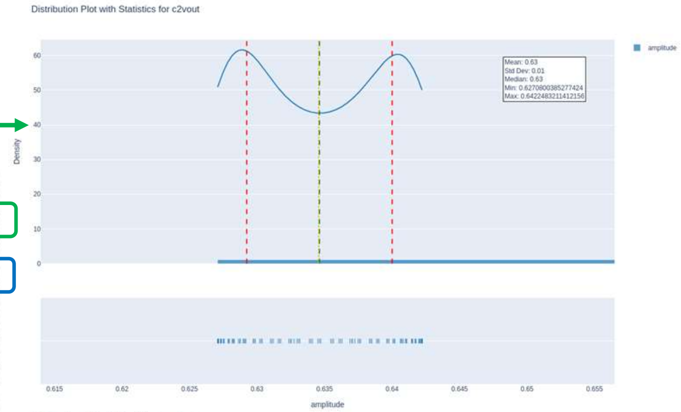
Signal Characteristics extraction : principle(1)

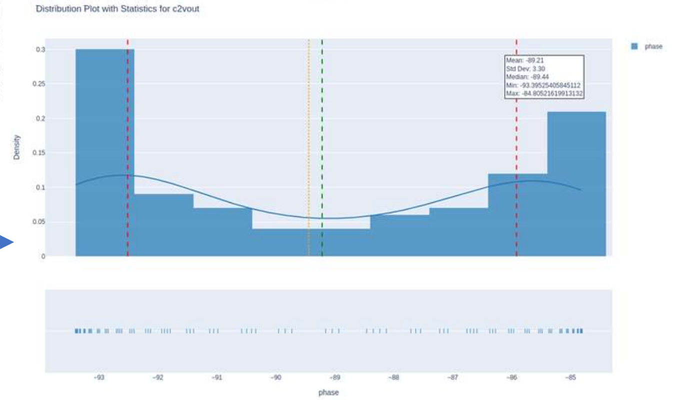


Signal Characteristics extraction: results (2)

Signals Profile Distribution Report (CSV format)

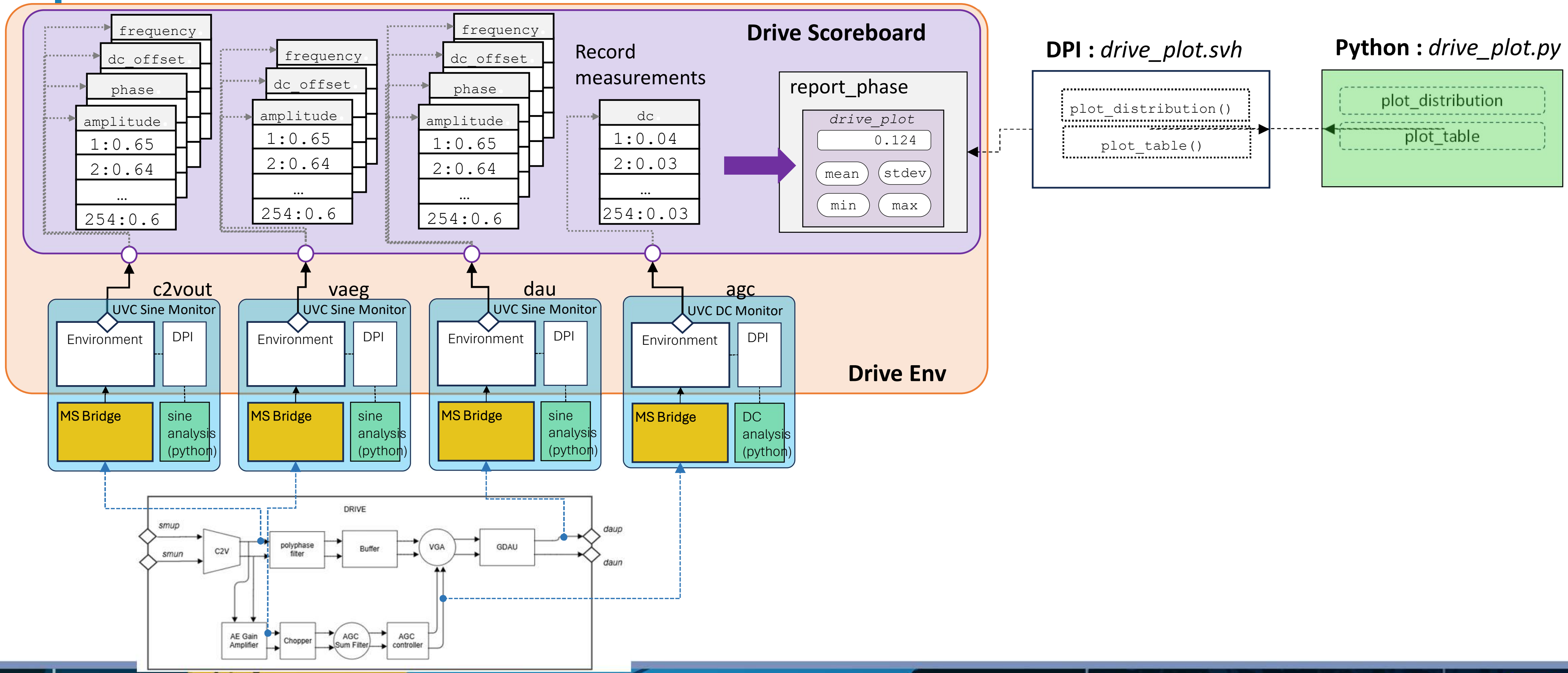
Signals	Signal Characteristics	Mean	Std_dev	Min	Max
c2vout	frequency	20000	0	20000	20000
	amplitude	0.634618533325387	0.0053888491620766	0.627080038527742	0.642248321141216
	dc_offset	0.00946686371997927	0.00465370981272827	0.000302502945467375	0.0149807315799656
	phase	-89.2141920615173	3.29749248065595	-93.3952540584511	-84.8052161991313
dau	frequency	20000	0	20000	20000
	amplitude	0.256884100434424	0.0124355926808513	0.200556345006466	0.28564480710703
	dc_offset	0.00416630939162828	0.00210803492420717	0.000267260228577477	0.0121642054848046
	phase	-0.00273559325168435	0.182534081402091	-0.318260159503401	0.766937506837987
vaeg	frequency	20000	0	20000	20000
	amplitude	0.67088244951541	0.00569678339990954	0.662913183586471	0.678948225206428
	dc_offset	0.0100078273611209	0.00491963608774131	0.000319788828065512	0.0158367733845351
agc	dc	0.0459645342194556	0.00203134263043896	0.034375903990341	0.0490415261670375





Signals Profile spread – illustrated with C2V amplitude and Phase plots

Drive Environment with UVC/Python DPI implementation



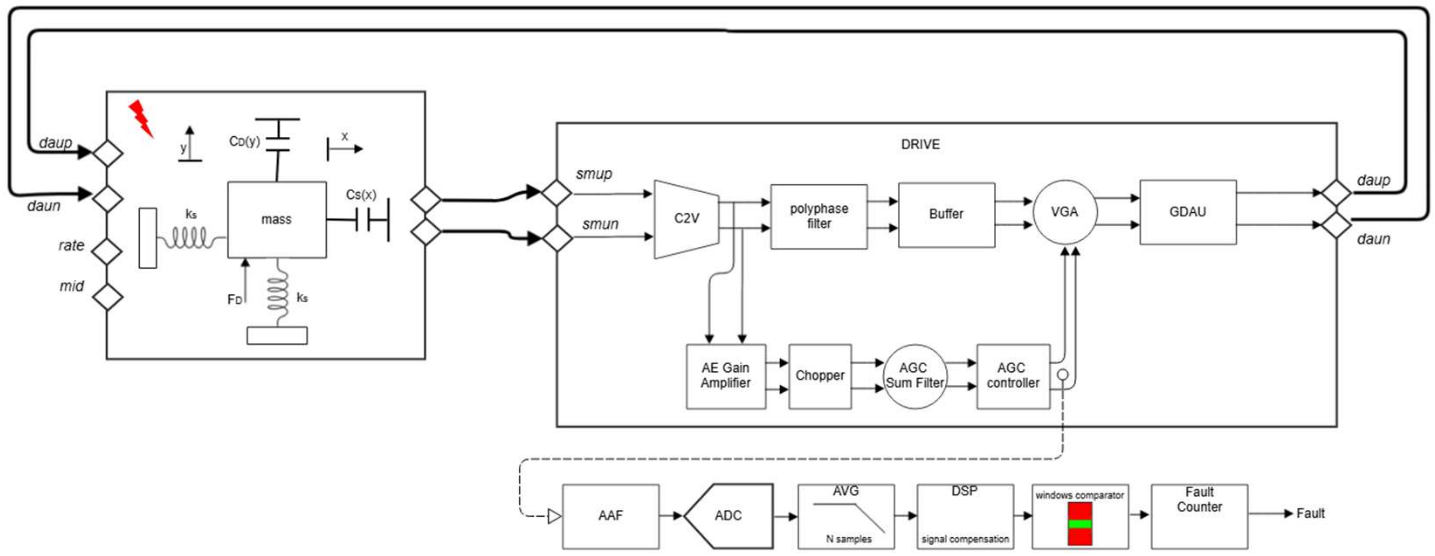
Agenda

- Introduction and Motivation
- MEMS Gyroscope System
- Modeling Strategy
- **Use Case: Safety Mechanism Verification**
- Conclusion
- QA

Safety Mechanism

- ISO26262 product require Safety Mechanism to detect fault on critical function (65 Safety Mechanism in our product)
- As Safety Mechanism will involve different analog blocks and complex digital function; an efficient verification is required
- Amount different Safety Mechanism on the Gyroscope function, DAU monitor will be detailed
- This function could detect for example
  - MEMS to ASIC bond wire/pad Opens and Shorts
  - Broken Drive Fingers
  - Drive Suspension Crack

DAU Monitoring



A rupture in MEMS displacement will create an amplitude change in drive loop and so in AGC loop that will be no more in define safety limit

Simulation and model refinement

- Initial Model inaccuracy
  - Original MEMS models did not accurately reflect DAU response to changes in Quality Factor (Qd), requiring refinement
- Refined Simulation Results
  - Post-refinement simulations demonstrated expected behavior with decreased Qd increasing DAU output
- Reliability Quantification
  - Simulations across various Qd values quantified impact on safety mechanism reliability and validated model robustness

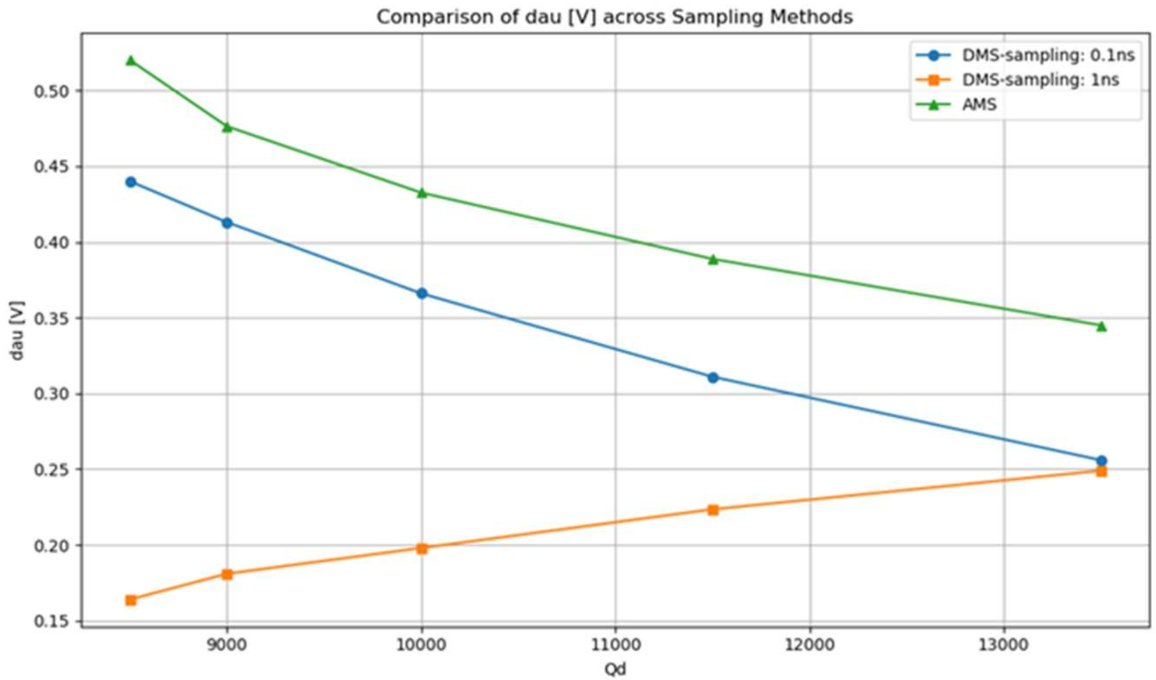
Validation strategy

- Signal-level Characteristics
  - Validation monitors phase and amplitude to ensure accurate signal-level behavior in the system
- System-level Behaviors
  - Checks include startup time and FSM transitions to verify overall system operation
- Analog Property Verification
  - Analog metrics are verified with a scoreboard using DPI for metric plotting and analysis
- Digital Behavior checks
  - SystemVerilog Assertions (SVA) are used to validate digital behaviors for correctness

Comparing DMS vs AMS: accuracy & efficiency

Signals	Drive signal-level characteristics			
	Signal characteristics	DMS SV UDN (sampling:0.1ns)	DMS SV UDN (sampling:1ns)	AMS Verilog AMS (Ref)
c2v	Amplitude [Volt]	0.6346	0.636	0.628
	Frequency [Hz]	20000	20000	20000
	Phase vs (Polyphase) [deg]	-89.21	-89.21	-90
agc	DC level [Volt]	0.04595	-0.0446	0.05
dau	Amplitude [Volt]	0.2567	0.249	0.345
	Frequency [Hz]	20000	20000	20000
	Phase vs (VGA) [deg]	-0.0062	-157.42	0.001
aegain	Amplitude [Volt]	0.67	0.67	0.664
	Frequency [Hz]	20000	20000	20000
Parameters	Drive system characteristics / CPU			
Startup-time	[ms]	36.59	34.71	37.25
CPU time	[h:m:s]	00:36:53	00:17:46	1:30:52

Results



- DMS Sampled with 0.1ns follows the expected behavior as AMS for increase in Quality Factor

Agenda

- Introduction and Motivation
- MEMS Gyroscope System
- Modeling Strategy
- Use Case: Safety Mechanism Verification
- **Conclusion**
- QA

Summary and Future work

- Effective MEMS Modeling
  - Modeling MEMS and analog signal chains with DMS ensures accuracy and robust verification of systems
- Enhanced signals Fidelity
  - Precise sampling configurations and custom enet structures improve signal fidelity, especially for C2V signals
- Python DPI Integration
  - Integration of Python DPI enables support for complex measurement and data analysis in MEMS modeling
- Future Enhancements
  - Future work targets dedicated DMS for MEMS accelerometer extensions, shock detection, AI/ML for tuning and anomaly detection

Acknowledgements

- We would like to formally express our sincere gratitude to **Keith Kraver** and **Margaret Kniffin** for their contributions to this work.
- We are grateful for their insights and support throughout this work.

Agenda

- Introduction and Motivation
- MEMS Gyroscope System
- Modeling Strategy
- Use Case: Safety Mechanism Verification
- Conclusion
- **QA**