

A Python-based LLRF Algorithm Library

Z. Geng (zheqiao.geng@psi.ch), R. Rybaniec, Paul Scherrer Institut (PSI), CH-5232 Villigen PSI, Switzerland

Abstract

Many common algorithms are used in LLRF applications or testing software. Implementing these algorithms as a library with widely used computer languages is attractive to share knowledge within the LLRF community and avoid duplications in development. This poster reports the progress of implementing an LLRF algorithm library in Python, a popular language used in LLRF high-level applications and beam controls. The following algorithms are implemented: cavity parameters and model identification, RF system calibration, RF signal demodulation, RF controller design and analysis, noise analysis, and RF system simulation. The library can be directly used in Python-based software, such as the Python-EPICS-based soft IOC automating the operation of an RF station. We also demonstrate the library with data from actual or simulated RF systems.

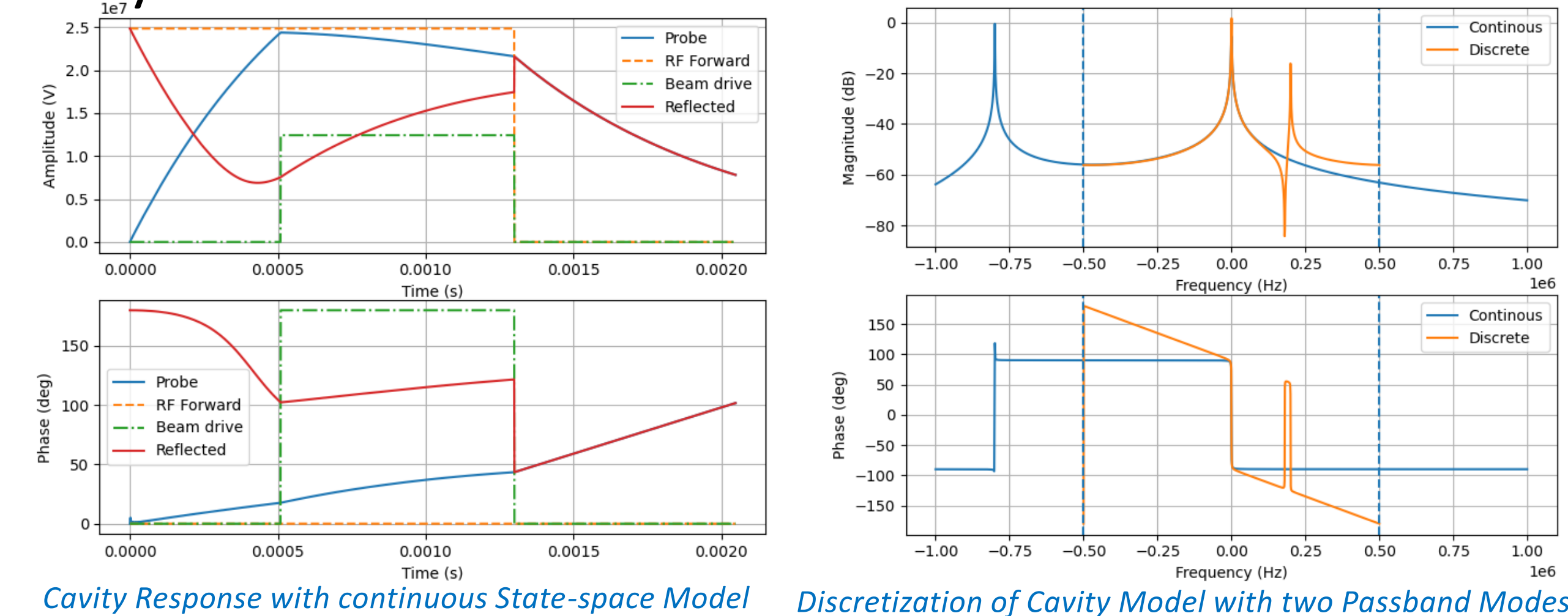
Python Algorithm Library

Module	Explanation
rf_sim	Sim. cavity response; determine operation parameters
rf_control	Design and analyze RF feedback/feedforward controllers
rf_det_act	Measure RF amplitude and phase from ADC samples
rf_noise	Noise analysis, generation, and filtering
rf_calib	RF calibrations like virtual probe, RF actuator offset/imbalance, forward and reflected calibrations
rf_sysid	Identify the system transfer function and characteristic parameters (e.g., cavity QL and detuning)
rf_fit	Fit data to sine/cosine, circle, ellipse or Gaussian functions
rf_misc	Misc. functions like saving data to files and reading files
rf_plot	Plot functions for internal usage

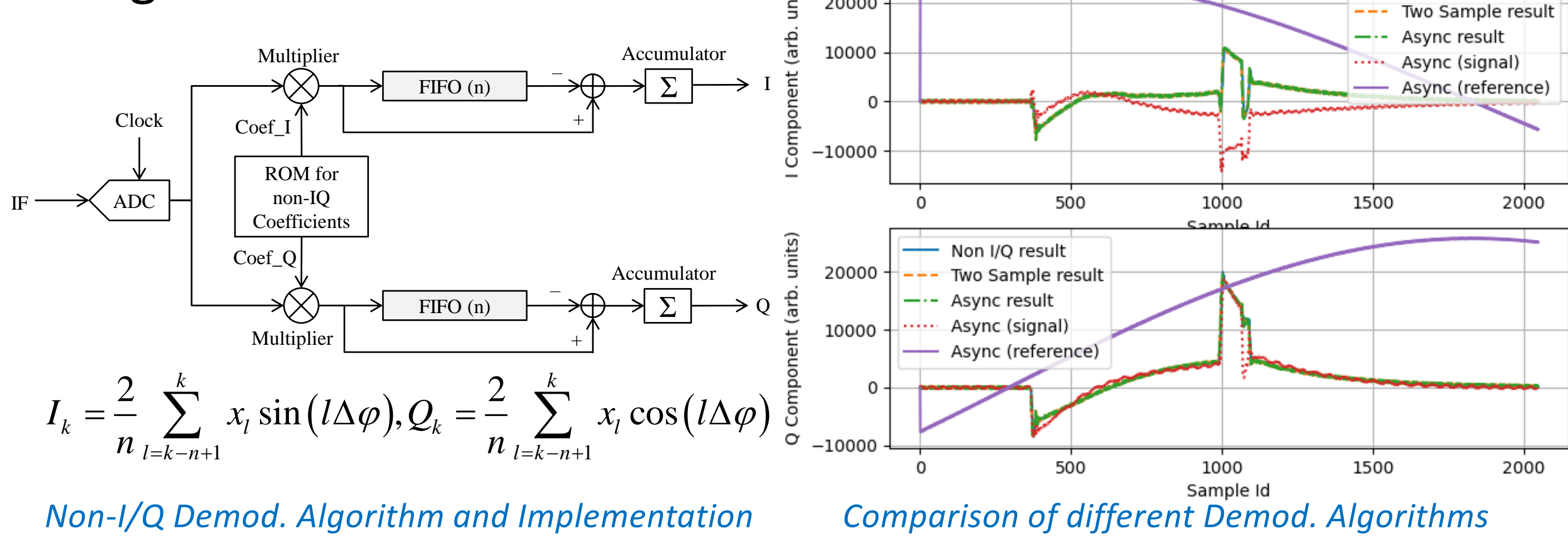
The library relies on "numpy" and "scipy", "matplotlib" is also loaded if any plot function is enabled.

Examples

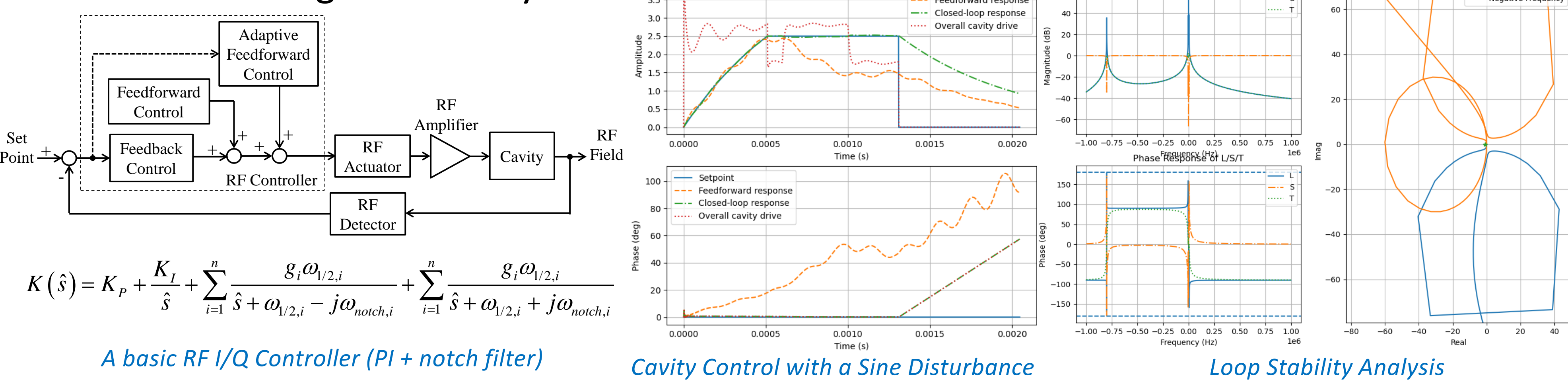
RF System Simulation



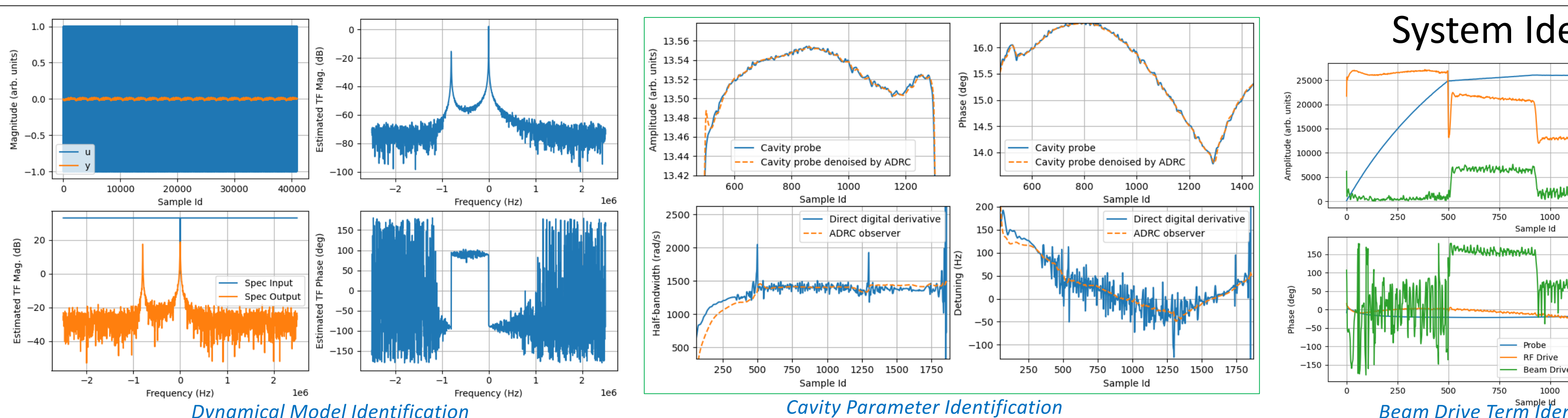
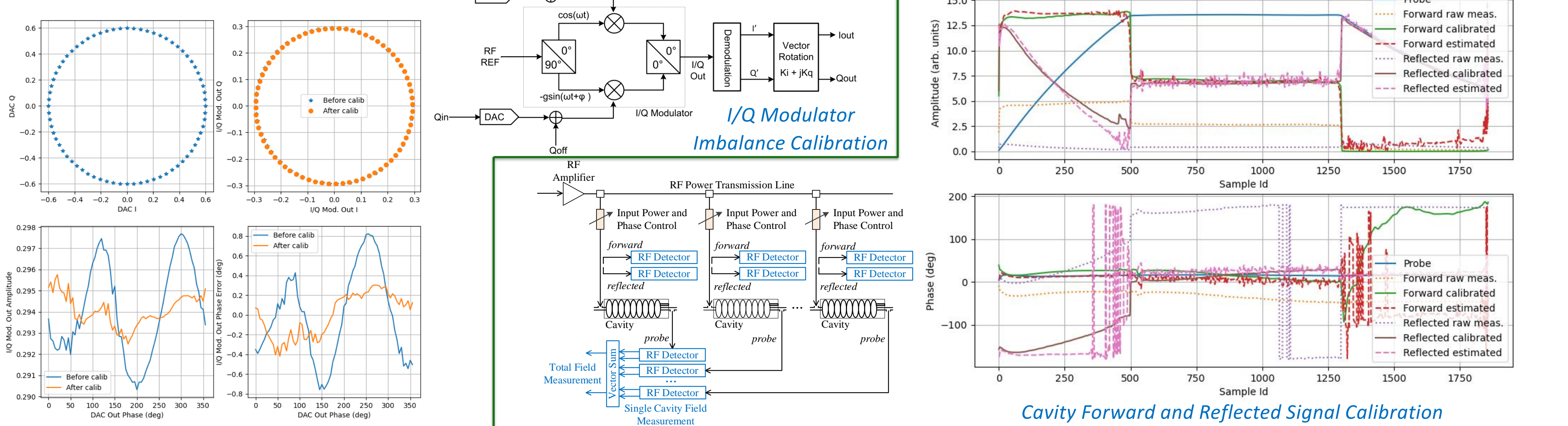
RF Signal Detection



RF Controller Design and Analysis



RF Calibration



Motivation and Overview

Software Libraries Motivation

Software Framework Library

- Define a common software architecture
- Implement infrastructures for multi-threading, hardware access, and network access

LLRF Algorithm Library (C/C++)

- Implement RF control domain algorithms
- Support soft-real-time applications

LLRF Algorithm Library (Python)

- Implement same algorithms as the C/C++ library
- Also implement features like noise analysis, feedback controller design, and system simulation
- For high-level applications (online or offline) and automation

Firmware Libraries Motivation

Firmware Framework Library

- Construct the firmware project structure
- Simulate the firmware code
- Automate the firmware synthesis
- Access common components (e.g., ADC, DAC, etc.)

Basic Firmware Library

- Collect firmware build components common for different applications
- Implement high-level LLRF control modules
- Can be assembled and configured for fast prototyping of LLRF controllers

RF Control Firmware Library

- Implement high-level LLRF control modules
- Can be assembled and configured for fast prototyping of LLRF controllers

Functional Architecture of LLRF Firmware/Software and possible Libraries

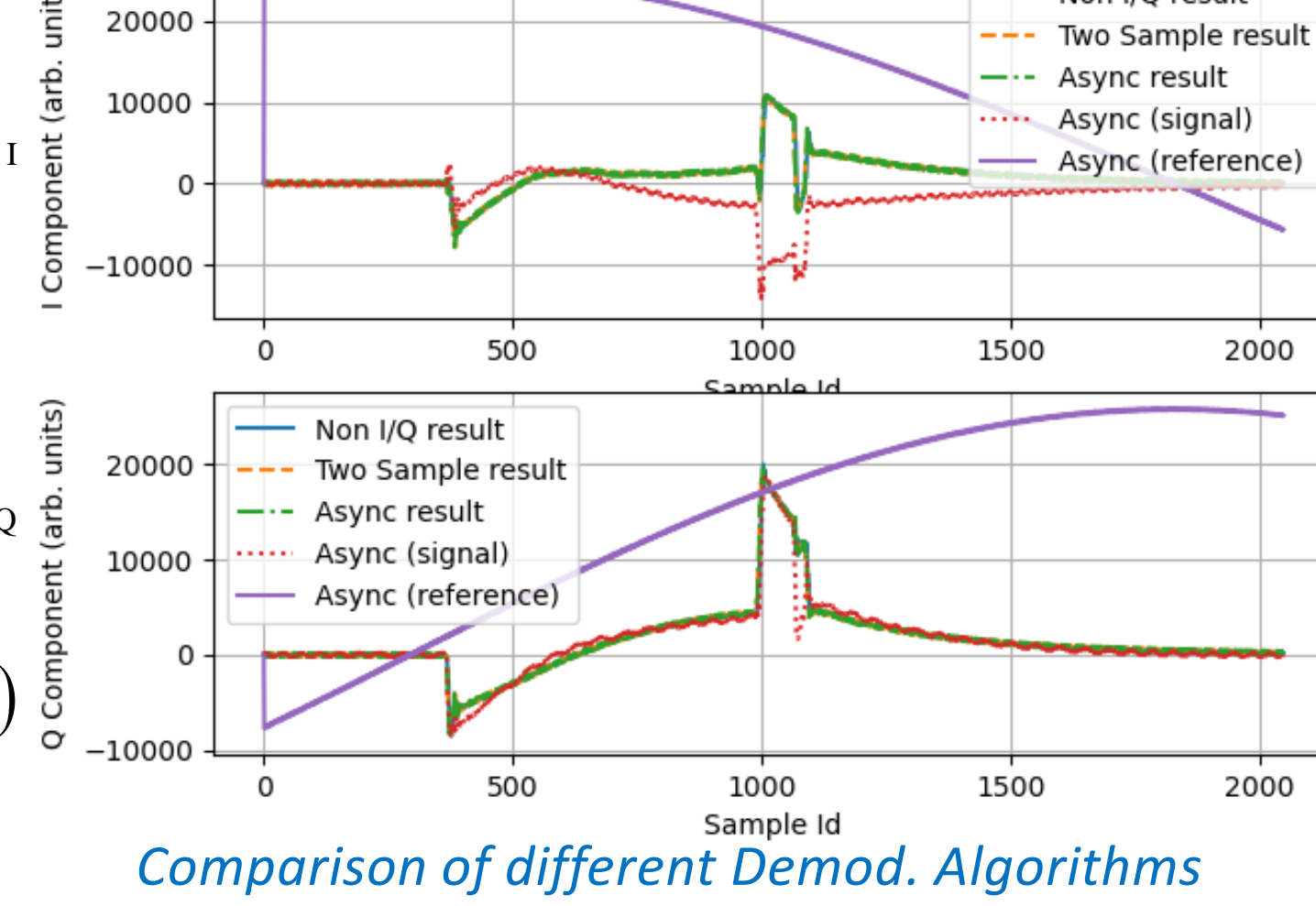
Software Framework Library (e.g., ooEpics, ooPyE, DESY's ChimeraTK)

- General software architecture implementation
- Framework for software development
- General infrastructure for hardware/network access
- Automatic code generation
- ...

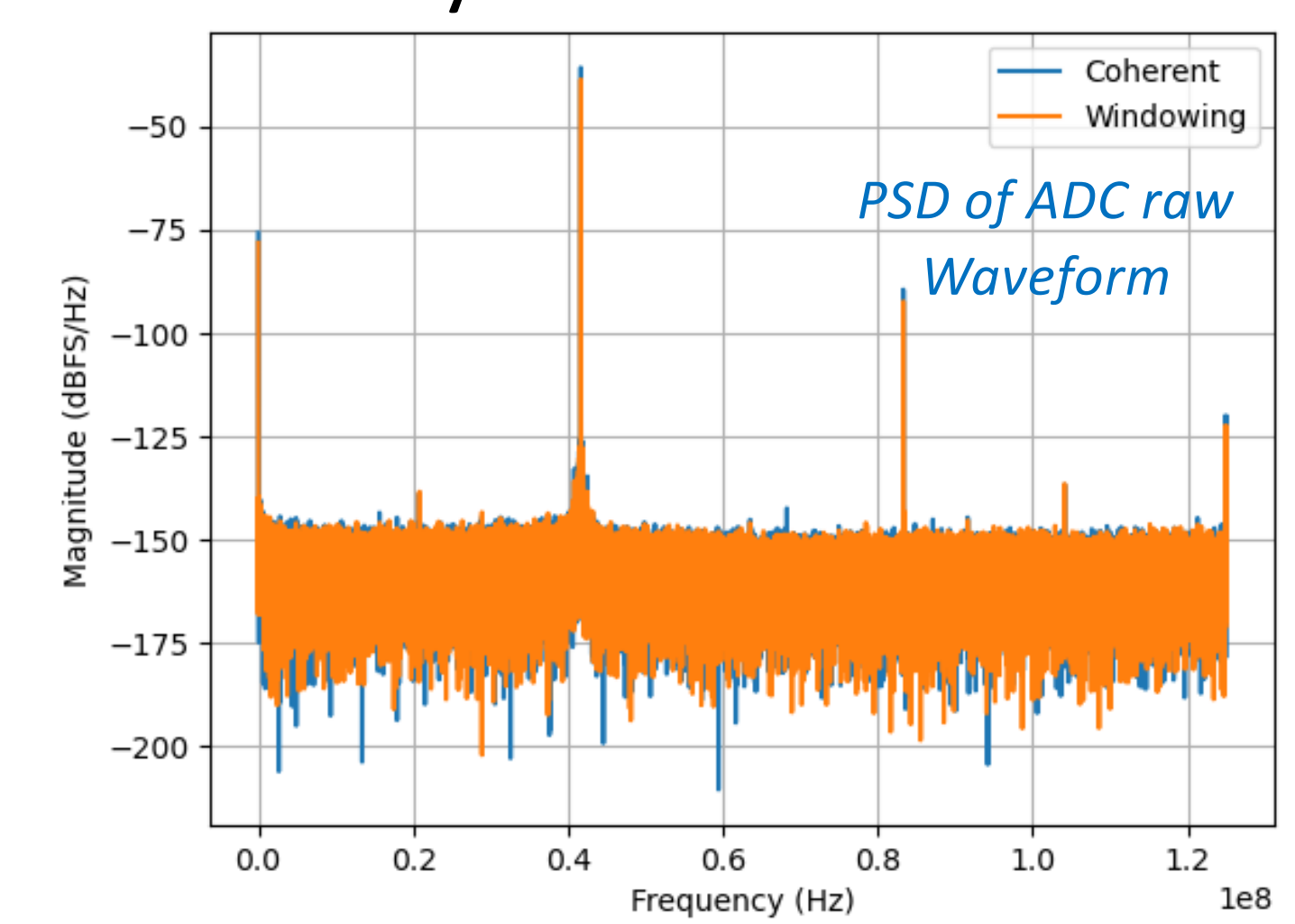
Firmware Framework Library (e.g., PsiSim, psi_tb, DESY's FMK)

- Framework for firmware project
- VHDL code test-bench and simulation
- Automatic firmware synthesizing
- ...

Generate Time-series Noise from given PSD Points



Noise Analysis



Offset Frequency (Hz)	10	100	1e3	10e3
Noise PSD (dBrad ² /Hz)	-82.03	-107.07	-128.22	-135.86
	100e3	1e6	5e6	10e6
	-145.81	-148.15	-148.62	-164

Outlook

Continue the library implementation, develop the documentation, and make the library open source.