DARPA ACCEL Phase 1.5 LLRF Control System Specifications

14 Dec. 2022, Dr Chao Liu, TID-ID, chaoliu@slac.stanford.edu

With inputs from Ryan Herbst, Bo Hong, Zenghai Li and Emilio Nanni

1. System Requirements

The system requirements of DAEPA ACCEL phase 1.5 LLRF control system are summarized in Table 1. The duration of the burst at the threshold beam energy and threshold beam current is only 8s, so the temperature change introduced by the pulse is as low as 0.1 degree. The change in the resonant frequency of cavities due to the temperature change is acceptable for the klystron. However, the operating temperature range of the ACCEL system can change significantly, which can result in large cavity resonant frequency drifts. The control system needs to be capable of compensating for the frequency drifts, which can be a high-level control implemented in software.

ACCEL Specifications	**Our Target**	**Comments**
Pulse Rep Rate [Hz]	60/120	
Pulse Flat Top [s]	1/5.00E-06	Data rate and memory size check
Cavity Frequency [MHz]	5712	
Phase stability <[deg]	<1° (486fs)	
Amplitude stability <[%]	<1% variation RMS	
Duration of the burst (s)	8 (target)/ 2 (threshold)	The cavities are operated at cold temperature as the duty burst is so low.
Accelerator system weight (kg)	75 (target)/ 200 (threshold)	LLRF electronics need to be as compact as possible.
Accelerator system size	Fit within a cylinder that is 1 m long and 0.4 m in diameter	LLRF electronics need to be as compact as possible.
Non-operating temperature (°C)	-40 to +125 (target) -40 to +85 (threshold)	
Operating temperature (°C)	-40 to +85 (target) -20 to +85 (threshold)	LLRF control is required to cope with temperature change.

Tabel 1: System Requirements for DAEPA ACCEL Phase 1.5 LLRF Control System Requirements

2. Feedback Control System Hardware Architecture

Figure 1 shows the full ACCEL circuit for two pairs of cavities and phase 1.5 prototype system only have a pair of cavities. For each pair of cavities, there are two RF signals available from the system – the cavity reflection from the magic T and klystron forward from the coupler at the output of the klystron. The prototype may have two probes for the klystron forward signal to improve the signal quality, so another RF signal will be added. The two signals from the klystron forward port will be digitized and down converted with independent ADC channels, and then averaged between them. Therefore, the rest of the firmware processing modules will remain the same as when there is just a single probe on klystron forward port. As there is no cavity probe, the LLRF control algorithm needs to be designed differently with other accelerators.

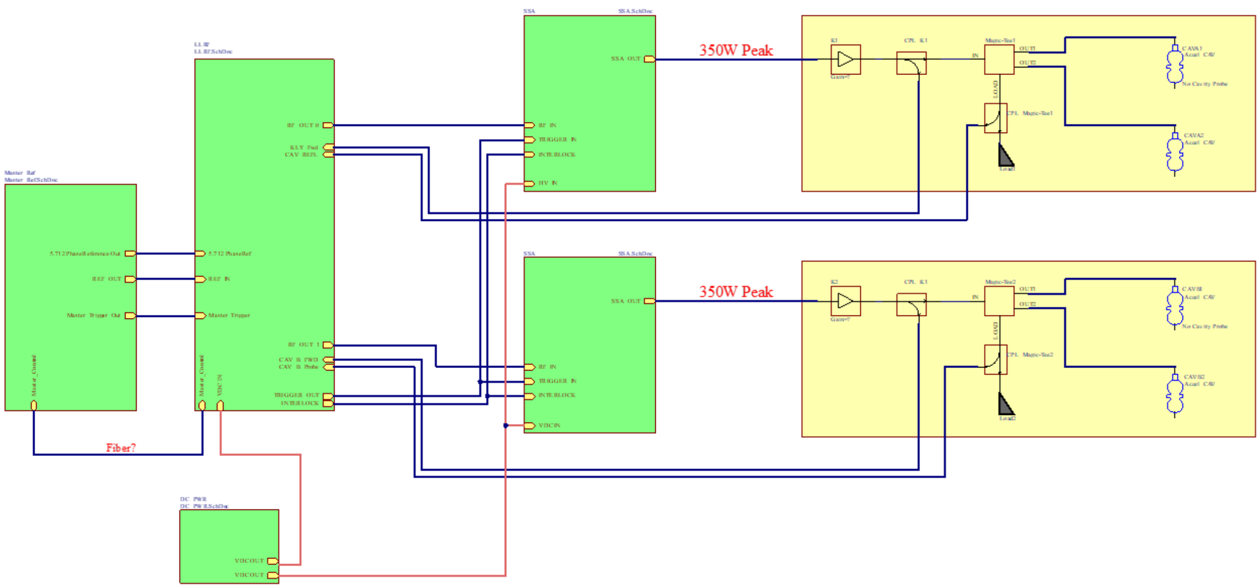

Figure 1. System architecture of the full ACCEL circuit (figure credit: Hong Bo)

Due to the extreme desired SWaP requirements, we propose to direct sample the RF signals by using the integrated data converters integrated in RFSoC. The up conversion and down conversion are both performed digitally by using the hardened DUC and DDC data paths, which are parts of the integrated RF data converters in RFSoC.

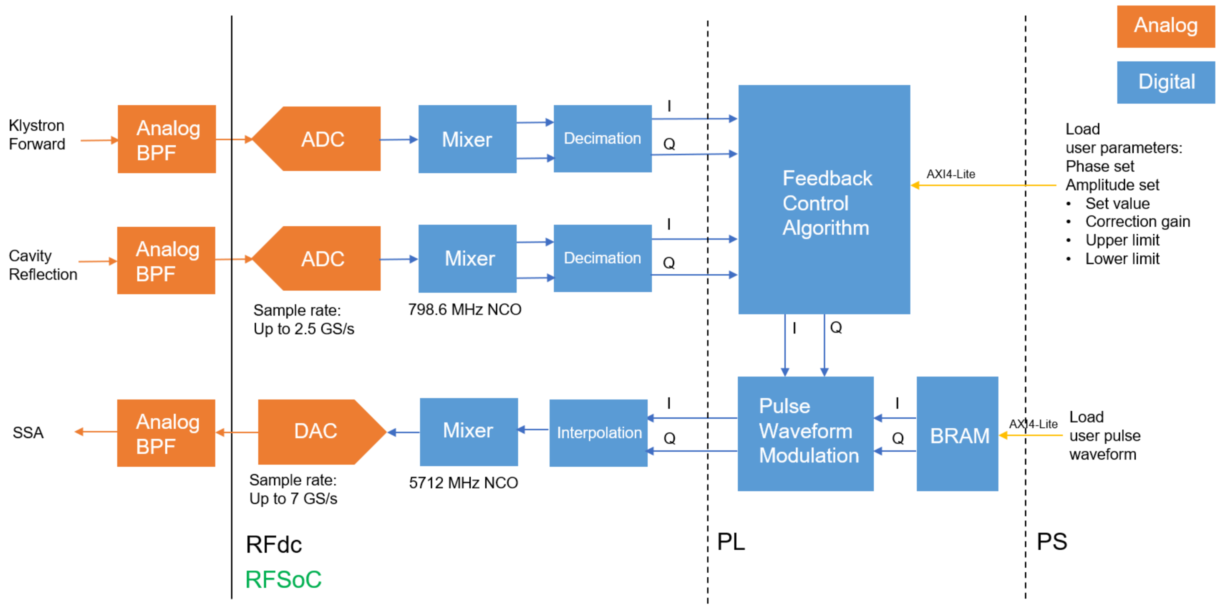

Figure 2. The block diagram for the LLRF control circuit

Figure 2 shows the block diagram of the LLRF control algorithm emphasized on the RF data conversion part. The requirements for the feedback control algorithm will be expanded in Section 3 and the flow charts for the control algorithm will be described in Section 4.

As shown in Figure 2, the RF signal processing chain can be divided into the following steps:

1. User pulse waveform loaded into BRAM in I and Q format
2. Pulse waveform complex multiplied with new set values I and Q calculated by Feedback Control Algorithm module
3. New pulse waveform interpolated (from 245.76 MS/s to DAC sampling speed), up converted (NCO at 5712 MHz) and played back by DAC
4. RF signal from DAC fed into SSA and then the rest of accelerator circuit
5. The klystron forward and cavity reflection RF signals are directly sampled by ADCs at 2.4576 GS/s then down converted (NCO at 798.6 MHz: 5712 MHz aliased back to the first Nyquist zone) and then decimated by 10
6. The I and Q streaming data fed into the Feedback Control Algorithm block to calculate the new set value for I and Q based on use defined operation state

3. Feedback Control System Functional and Performance Requirements

Requirement 1: RF Frequency Control

Motivation: the target operation temperature of ACCEL is from - 40 to 85 °C and the resonant frequency changes by 0.1 MHz per °C. As there is no temperature stabilization mechanism for ACCEL, the RF frequency needs to be tuned according to the ambient temperature when operating.

Objective: tune the RF frequency to resonant frequency of cavity based on ambient temperature.

Signal source: cavity reflection probe (1.1 to 1.5 µs in Figure 4).

Control mechanism: The **first step of the RF frequency control** will be implemented by using the reading of the temperature sensor with a look-up-table of temperature versus cavity resonant frequency defined by RFAR team. The RF frequency can be coarsely set according to the ambient temperature, so the RF signal can be injected to the cavity. The NCO frequency in the hardened digital up conversion block can be set to the corresponding RF frequency by **SOFTWARE,** which can be implemented with API provided by Xilinx for RFdc IP.

After the field is injected to the cavity, the frequency error can be measured more accurately and then the RF frequency can be fine-tuned as the **second step of RF frequency control**. The frequency error can be derived from the **phase of cavity reflection signal.** There is a simple equation to convert the slop of phase to the frequency different between the operating RF frequency and the resonant frequency. RFAR team will provide the equation for conversion. The phase of the cavity reflection signal will be calculated in **FIRMWARE** and then propagated to **SOFTWARE** for slop calculation and frequency error conversion. The second step frequency tuning will be applied by resetting the NCO frequency by the same procedure as the first step frequency control.

NOTE: the NCO frequency on down-conversion side also needs to be tuned to the corresponding first Nyquist zone frequency on the up-conversion side via RFdc API on **SOFTWARE** side.

Performance requirement: required accuracy of frequency control to be confirmed by RFAR team.

Software and firmware partition summary: signal windowing and phase calculations in **FIRMWARE**; frequency error calculation and frequency setting in **SOFTWARE**.

Requirement 2: Pulse-top Flatness Control

Motivation: The flatness of the pulse top needs to be controlled as there are bunches over the entire RF pulse duration.

Objective: tune the pulse waveform to achieve a plat pulse top at cavity with defined tolerance.

Signal source: cavity reflection probe (0.25 to 1 µs in Figure 3)

Control mechanism: Cavity reflection signal measured to infer the amplitude of cavity signals. Adaptive feedback control to tune to the pulse shape to minimize the fluctuations. The pulse top can be divided into a few segments for flatness control purposes. As there is an integration effect on the pulse waveform, the flat pulse top can be achieved by adaptively tuning the offsets in segments of pulse waveform. At the beginning of each run, a customer pulse shape is loaded to cavity and the average amplitude over the entire pulse and the average amplitude per segment can be calculated. Then the offset required for each of the segments can be determined by comparing the overall average and the average for each of the individual segments. The average values can also be calculated after the flatness control for monitoring purposes. If the flatness is out of tolerance, the pulse top flatness control routine can be applied again.

Open questions for RFAR team: 1) the number of segments needed for flatness control to be confirmed from RFAR's simulation model for cavity 2) confirm the control strategy is acceptable 3) confirm if the phase is necessary for flatness control as there is change of sign in the phase between under and overloading.

Requirement: tolerance to be confirmed from simulation model of RFAR team.

Software and firmware partition summary: overall average amplitude and segment average value can be calculated in **FIRMWARE**; the average values can be propagated to **SOFTWARE** then the pulse waveform can be offset per segment to achieve the flatness requirement.

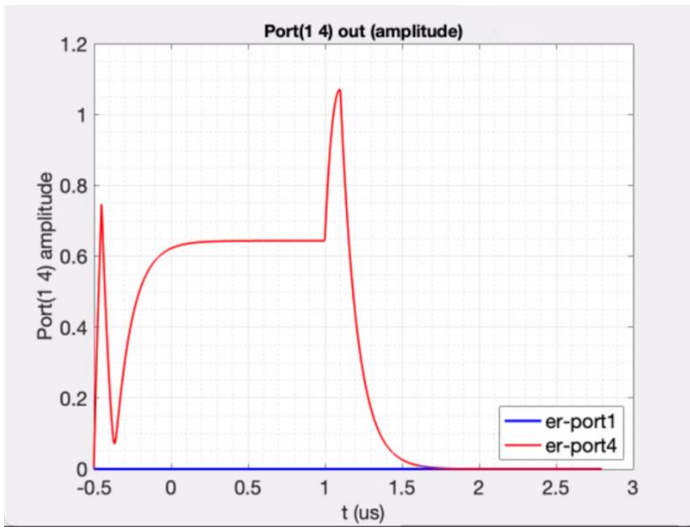

Figure 3. Amplitude of reflected signals (credit to Zenghai)

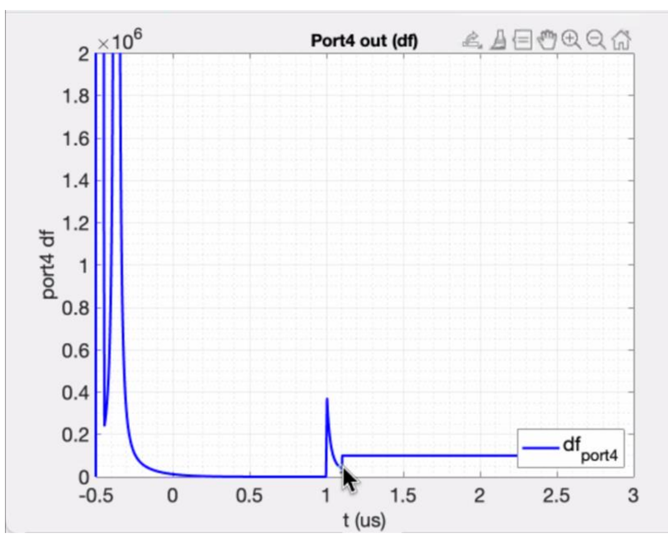

Figure 4. Time derivative of the phase of cavity reflection. (credit to Zenghai)

Requirement 3: Amplitude and phase control

Signal source: klystron forward probe(s)

Control mechanism: Klystron forward signal measured to infer the amplitude and phase of cavity signals. User will define desired phase and amplitude values in SOFTWARE and then the values are propagated to firmware as a target for an adaptive feedback control loop. The windowing of samples on during the pulse can be flexible and windowing function can also be customizable. In FIRMWARE, moving average will be applied to the digital down-converted data and then windowing averaged to a single IQ value. Then the single values for IQ components will be converted to phase and amplitude then used in the correction feedback loop in Figure 2.

Requirement: ±1° for phase and ±1% for amplitude

Software and firmware partition summary: User defined phase and amplitude values are set in **SOFTWARE** propagated to **FRIMWARE;** the feedback control loop for phase and amplitude control will be implemented in **FIRMWAR**.

4. Time Sequence of Control Flows

The ACCEL LLRF control has been divided into three main parts, which correspond to the first three requirements stated in Section 3. The control functions are performed by the following time sequence in general: 1) RF frequency control, 2) pulse-top flatness control and 3) phase and amplitude control. Flow charts have been produced to explain the procedure of the control flows. In the flow charts, the orange blocks will be implemented in software and blue blocks will be implemented in firmware. The interface between the software and firmware is defined in section 5.

Figure 5 shows the flow chart of RF frequency control flow. When the system is first switched on, the RF frequency control flow will be executed. The RF frequency will be coarsely set based on the reading of the temperature sensor. The RF pulse should be able to be injected into the cavity and the frequency error can be calculated based on the phase of the cavity reflection signal on the pulse tail. Then the RF frequency can be fine-tuned based on the frequency error. The frequency error calculation and fine-tuning loop can be executed at lower frequency after the system is first on to make sure the operating frequency is not too far off the actual resonance of the cavity.

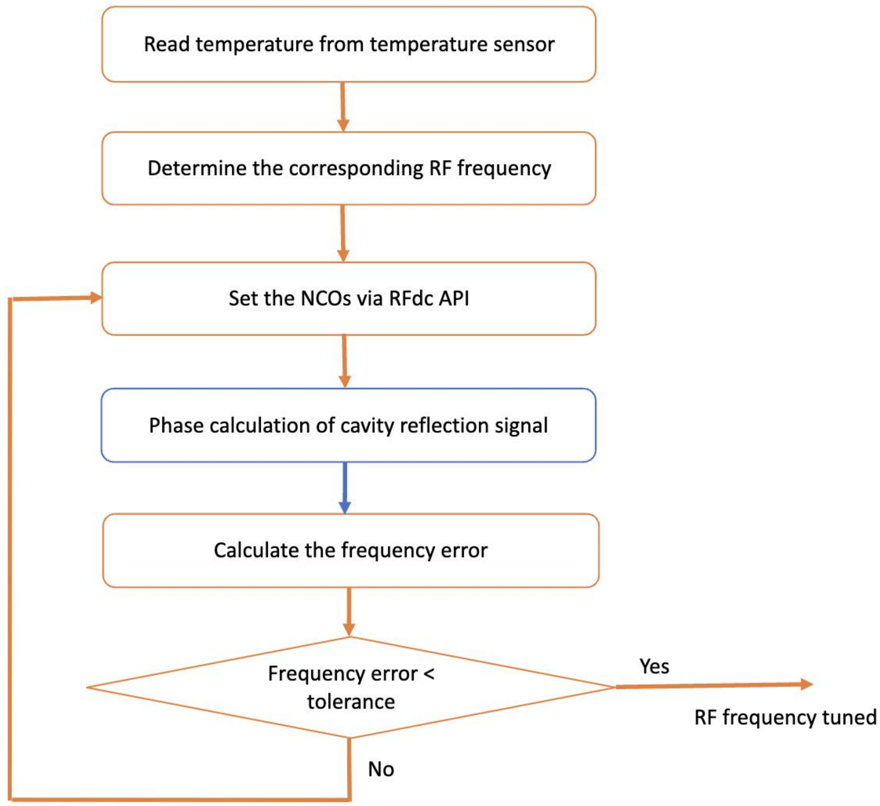

Figure 5. Flow chart for RF frequency control

Figure 6 shows the flow chart of pulse-top flatness control. After the RF frequency is set to the resonance of cavity, the pulse-top flatness control flow will be performed. The amplitude values of cavity reflections signal are calculated from the IQ signals from the down-conversion. Then the average of the amplitude values for the entire pulse-top and the segments defined in Requirement 2 mechanism. The average values are then passed to the software to calculate the offset required for each of the segments. The pulse waveform is then offset by the required values per segment until the amplitude fluctuation satisfies the pulse-top flatness requirement.

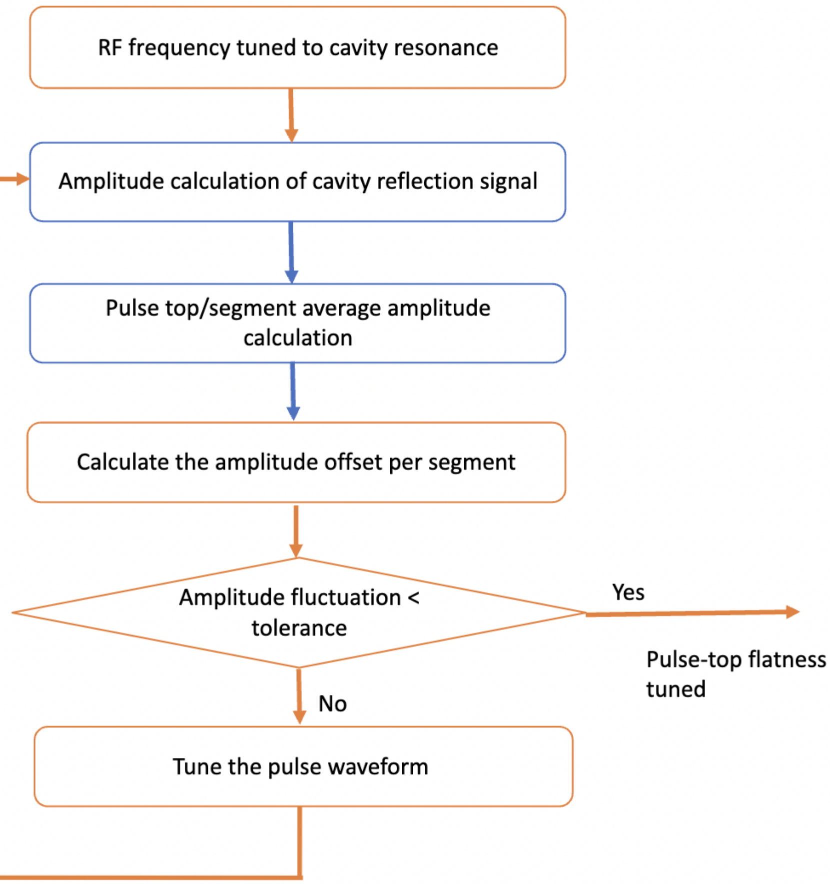

Figure 6. Flow chart for pulse-top flatness control

Figure 7 shows the flow chart of phase and amplitude control. After the RF frequency and pulse-top flatness are tuned to desire requirements, the phase and amplitude feedback control loop will start operating. The moving average with certain number of samples will be calculated for the down converted IQ klystron forward signals to improve the signal stability. Then the average value for the entire pulse duration will be calculated and then converted to phase and amplitude values. The current phase and values will be compared to the user defined values and then calculating for a new set of phase and amplitude to approach the user defined values in controlled rate. The new set of phase and amplitude values are converted back to IQ values and modulated with the latest pulse waveform. The up converted RF signal samples are finally clocked into the DAC to generate the updated RF pulse. The feedback loop will run continuously during the operation time to perform real-time pulse to pulse phase and amplitude correction. Most of the control loop will be implemented in firmware, except for some of the control parameters and the pulse waveform will be loaded from the software.

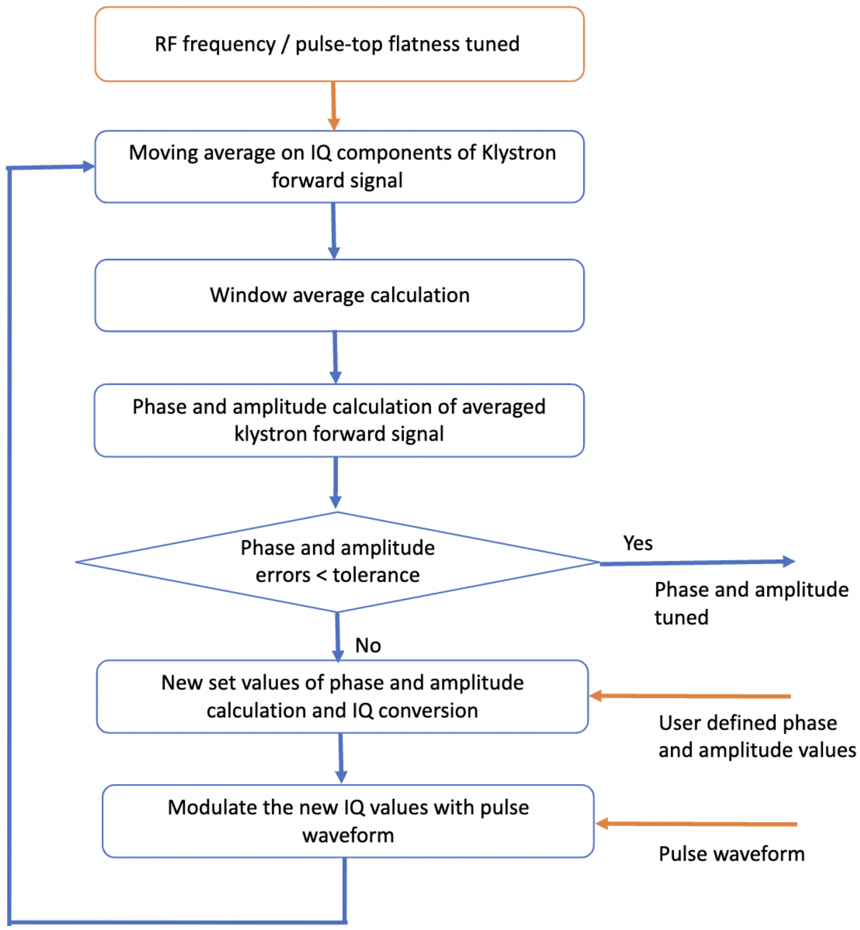

Figure 7. Flow chart for phase and amplitude control

5. Software and Firmware Interface

The DARPA LLRF system is implemented using the RFSoC platform, which is a system on chip technology, coupling a high-power FPGA with high bandwidth ADC's and DACs integrated with a multicore ARM processor. The firmware resident on this device is managed by a piece of software based upon SLAC's Rogue architecture which provides the hardware abstraction layer as well as interface to higher level software. The low-level interface between the firmware and software is facilitated by a custom kernel drive which provides an interface to streaming data through a firmware resident DMA engine as well as providing an API for reading and writing registers contained in the FPGA.

The interface for the communication between hardware abstraction layer and the higher-level software will be EPICS-7 PvAccess variables. Waveforms will be presented as Numpy arrays of the appropriate type (UINT32? FLOAT32?) and scalar variables will be presented as their native type. The state control, state monitoring and other interfaces between higher level control software and the hardware abstraction layer will also occur over EPICS-7 PV access variables. A PvAccess client interface will be provided to provide a debug interface directly to the firmware and associated support hardware. Additionally, the hardware abstraction layer will facilitate streaming data for debugging purposes.

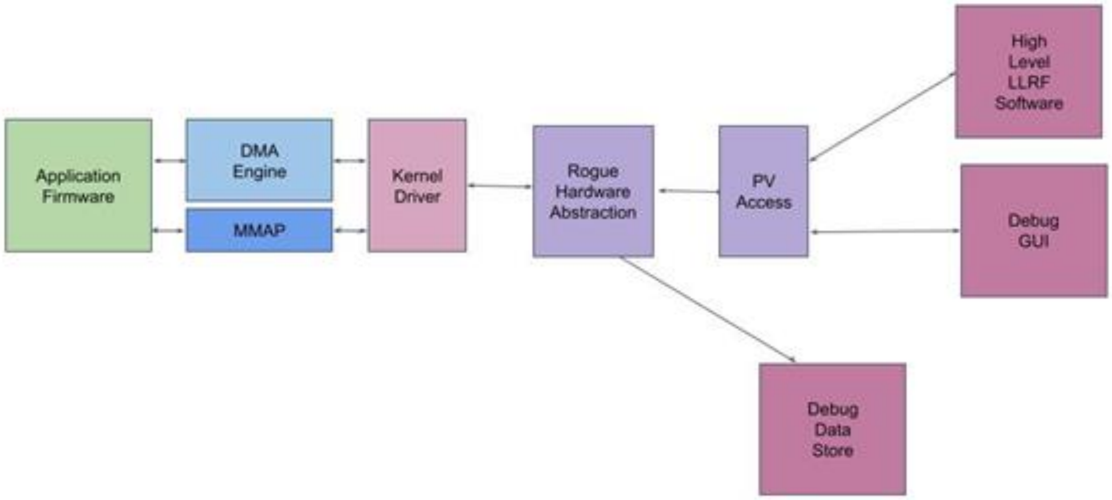

Figure 8. The block diagram of the interface between software and firmware and software layers

The parameters and data streams required to propagate between software and hardware are specified in the following parts. The first list covers the high-level user defined parameters, and the second list consists of data and parameters that need to be transfer between software and firmware to realize all the three control loops mentioned in section 3 and 4. The list is subject to be extended or altered based on further discussions in requirement of ACCEL and the actual implementation of the control flows.

User defined parameters in SOFTWARE:

- Pulse waveform (IQ)
- Phase
  - Desired value, correction gain, upper limit and lower limit
- Amplitude
  - Desired value, correction gain, upper limit and lower limit
- RF Frequency fixed value: to set the NCOs on up and down conversion sides and this maybe only be required for phase 1.5 or testing purposes.

SOFTWARE to FIRMWARE interface:

-> **SOFTWARE to FIRMWARE**

<- FRIMWARE to SOFTWARE

-> Initial pulse waveform (IQ)

-> Phase and amplitude parameters

-> Initial RF frequency value: as input by user or reading from temperature sensor (via GPIO)

<- Phase of cavity reflection signal on pulse tail: selected data points pass to SOFTWARE to be used for calculating the frequency error

-> Tuned RF frequency: set the NCO frequency via SOFTWARE API

<- Average values for amplitude for segments of cavity reflection signal during the pulse (1 amplitude value per segment)

<- Average value for amplitude of cavity reflection signal during the pulse (1 amplitude value per pulse): compare the average values of the segments against the average of entire pulse in SOTFWARE to determine the offset needed per segment

-> Tuned pulse waveform (IQ)

6. Task Division between SLAC and Radiasoft

The plan to divide the tasks of LLRF control is SLAC responsible for the overall system design and firmware development and Radiasoft responsible for the software development. There are a wide range of interaction elements between the software and firmware, which needs to be further clarified before and during development.