ARCHITECTURE DESIGN OF THE APPLICATION SOFTWARE FOR THE LOW-LEVEL RF CONTROL SYSTEM OF THE FREE-ELECTRON LASER AT HAMBURG

Z. Geng#, SLAC, Menlo Park, California, U.S.A.
V. Ayvazyan, DESY, Hamburg, Germany
S. Simrock, ITER Organization, St. Paul lez Durance, France

Abstract

The superconducting linear accelerator of the Free-Electron Laser at Hamburg (FLASH) provides high performance electron beams to the lasing system to generate synchrotron radiation to various users. The Low-Level RF (LLRF) system is used to maintain the beam stabilities by stabilizing the RF field in the superconducting cavities with feedback and feed forward algorithms. The LLRF applications are sets of software to perform RF system model identification, control parameters optimization, exception detection and handling, so as to improve the precision, robustness and operability of the LLRF system. In order to implement the LLRF applications in the hardware with multiple distributed processors, an optimized architecture of the software is required for good understandability, maintainability and extendibility. This paper presents the design of the LLRF application software architecture based on the software engineering approach for FLASH.

INTRODUCTION

FLASH is a VUV and soft-X ray free-electron laser machine located at DESY, Hamburg. The layout of FLASH is depicted in Figure 1 [1].

FLASH requires high RF field stabilities up to 0.01 degree in phase (at 1.3 GHz) and 0.01% in amplitude [2]. But various perturbations, such as Lorenz force detuning, microphonics and thermal drift of the cable, will generate amplitude and phase errors on the RF field in the superconducting cavities of FLASH [3].

The LLRF system is introduced to maintain the RF field stabilities with both feedback and feed forward control schemes [4], see Figure 2. The feed forward control is used to control the repetitive perturbations like Lorenz force detuning, beam loading and slow drifts, while the feedback control is used to control the field errors caused by random perturbations like microphonics and charge variations along the bunch train.

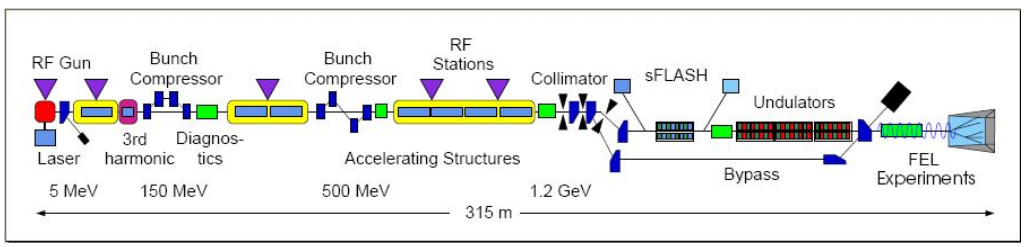

Figure 1: Layout of FLASH.

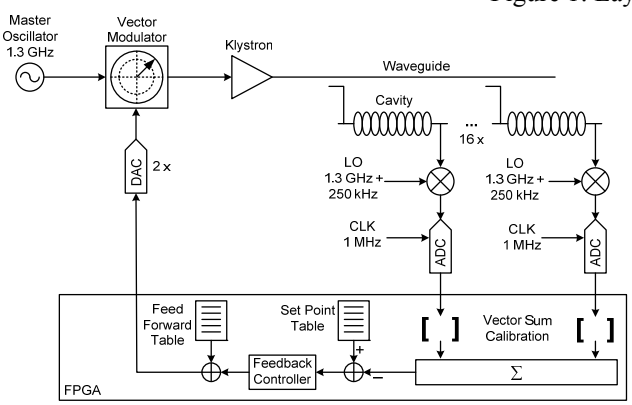

Figure 2: FLASH RF station including LLRF.

There are several great challenges for the LLRF system operation, including precise calibration of the vector sum for multi-cavity operation driven by a single klystron, optimal feedback and feed forward control, exception detection and handling for system robustness and automation for easy operation. So, sophisticated algorithms and application software are required to solve these problems in addition to the basic control loop.

LLRF APPLICATIONS

LLRF applications are sets of software to facilitate the LLRF control loop for better precision, robustness and availability. The main goals of the LLRF applications include

- Improve the RF field stabilities by optimizing the parameters of the RF field controller.
- Improve the robustness and availability of the LLRF system by system diagnostics, exception detection and handling.
- Support automation for easy operation.

The LLRF applications can be divided into several categories and their logical relations are show in Figure 3. Several important applications of each category are listed in Table 1.

#gengzq@slac.stanford.edu

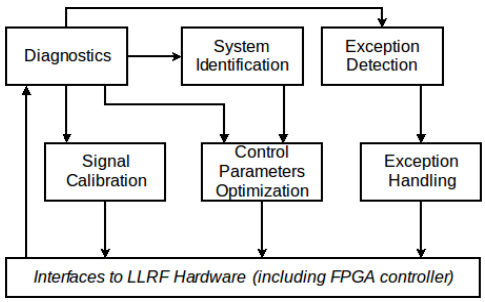

Figure 3: LLRF application categories.

Table 1: LLRF Applications

Category	Applications
Diagnostics	- Beam phase and beam current measurement - Loop phase and loop gain measurement - Cavity incident and reflected waves measurement - Cavity loaded quality factor and detuning measurement
Signal Calibration	- Vector sum calibration - Cavity gradient and phase calibration - Cavity incident and reflected power calibration - RF gun field calibration
System Identification	- RF system dynamic model identification - Cavity model identification - Klystron non-linearity characterization
Control Parameters Optimization	- Adaptive feed forward - Feedback gain scheduling
Exception Detection	- Cavity quench detection - Cavity operational limit exceeded detection - RF system components failure detection
Exception Handling	- Recovery from cavity quench - Adjust cavity gradient

SOFTWARE ARCHITECTURE DESIGN

Software Layers

The architecture of the LLRF applications is designed as several layers, see Figure 4.

The LLRF Algorithm Library will be realized in C language for better performance of the mathematic calculations. It contains most of the LLRF domain knowledge, and is optimized for computation power and memory consumption so that it can be highly reused for both DSP and common CPU programs.

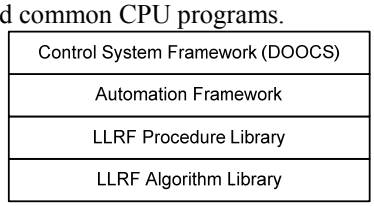

Figure 4: Layers of the LLRF applications architecture.

The LLRF Procedure Library will be realized in C++ language for better extendibility. It realizes the procedures for the tasks (use cases) of the LLRF applications. Sequential or state dependent logics will be used to coordinate the procedures. The platform independent parts of the procedures are designed as abstract classes, which are reusable for different machines with different platforms by derived classes, see Figure 5.

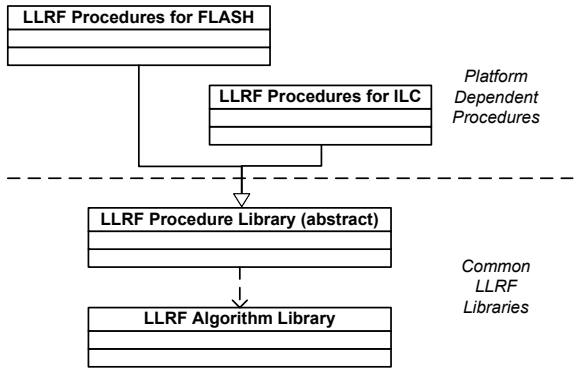

Figure 5: LLRF Libraries.

LLRF Algorithm Library

The aim of this LLRF algorithm library is to share the implementation with FLASH, European XFEL at DESY and even the future ILC.

To be maximum reused by various accelerators, the LLRF algorithm library is designed with C language, which can fit to the popular control systems like DOOCS [5], TINE [6], EPICS [7], and so on. And even can be used for DSP programming.

The library is broken down based on the domains, see Figure 6. The idea is to pack the code for a specified domain (like digital signal processing) so that the whole package can be reused in other applications. Of course, if one domain package has dependencies to other packages, the dependencies should be also considered for reusability.

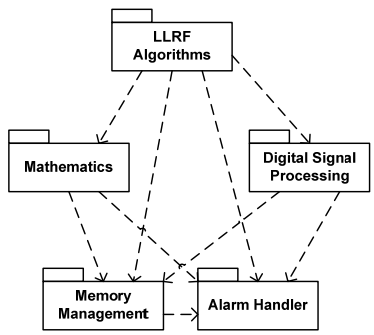

Figure 6: Domain breakdown of the LLRF Algorithm Library.

For each domain package, the codes are divided into three parts: “Available Interface”, “Library Body” and “Required Interface”. The interactions between domain packages are only through the interfaces.

The advantage of this design is that the library body can be upgraded without affecting other domain packages if the interfaces are not changed.

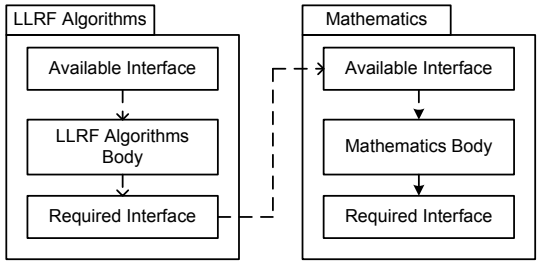

Figure 7: Dependency between packages.

Software Allocations

The LLRF applications at FLASH are realized in the DOOCS control system framework. Different applications are allocated to different DOOCS servers, and they use RPC to communicate with each other through the Ethernet.

The front-end servers are used for front-end hardware control, which will communicate with the LLRF hardware including FPGA controller. Some applications with real-time requirements are also allocated here to avoid the time consumption caused by Ethernet data transfer.

Most other applications for system calibration and optimization are realized in the middle-layer servers for each RF station. The applications that need to adjust more than one RF station are allocated to the middle-layer server for global control. One example is the recovery from quench, from which the cavity gradient of more than one RF stations may be changed.

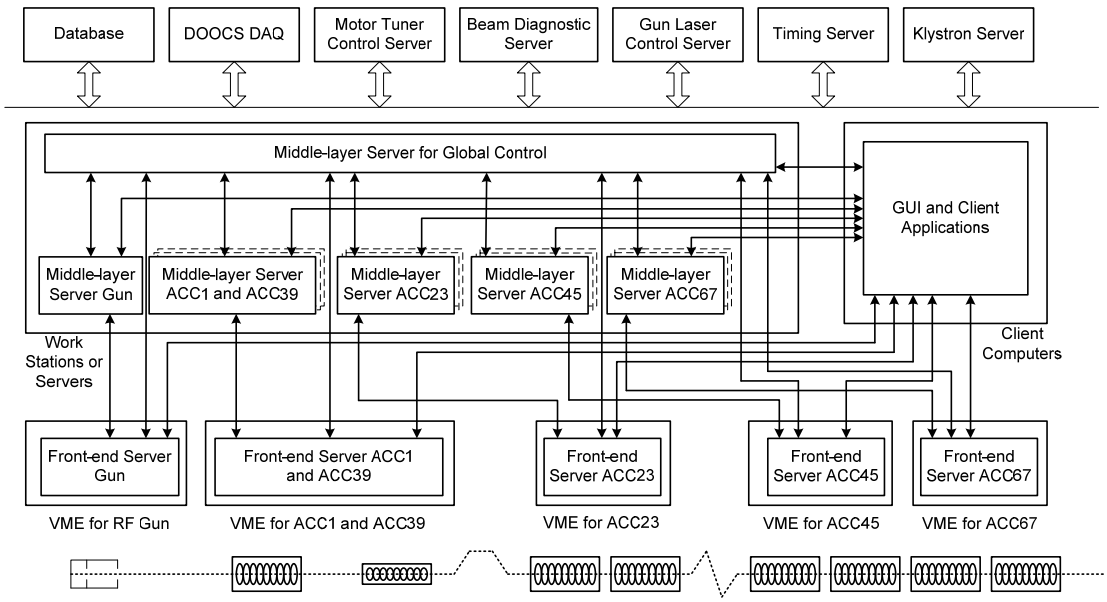

Figure 8: Allocation of the LLRF applications.

CONCLUSION

Software architecture is designed for LLRF applications at FLASH. This is the first time to implement the LLRF applications in a systematic way. The LLRF Algorithm Library has been developed and several applications have been successfully implemented and tested at FLASH [8], including the control table generation, vector sum calibration, RF gun calibration, RF system identification and loop phase and loop gain control. The architecture is proved successful for good understandability, maintainability and extendibility, which will be the reference design for the application software of the LLRF system for the European XFEL project. The experiences gained are also useful for the LLRF system design for other machines like ILC and normal conducting accelerators.

REFERENCES

[1] http://flash.desy.de/
[2] M. Hoffmann. Development of a multichannel RF field detector for the Low-Level RF control of the Free-Electron Laser at Hamburg. PhD thesis. 2008
[3] S. Simrock, Z. Geng. Sources of Field Perturbations, LLRF lecture for the Forth International Accelerator School for Linear Colliders.
[4] S. Simrock, Z. Geng. Cavity Field Control, LLRF lecture for the Forth International Accelerator School for Linear Colliders.
[5] http://tesla.desy.de/doocs/doocs.html
[6] P. K. Bartkiewicz et al. The TINE Control System, Overview and Status, ICALEPCS'07, Knoxville, Tennessee, USA, 2007
[7] http://www.aps.anl.gov/epics/
[8] V. Ayvazyan, K. Czuba, Z. Geng et al., "Low Level RF Control System Upgrade at FLASH", in Proceedings of PCaPAC 2010, Saskatoon, Canada, 2010