GENERAL AUTOMATION OF LLRF CONTROL FOR SUPERCONDUCTING ACCELERATORS

A. Brandt (alexander.brandt@desy.de), V. Ayvazyan, S. Simrock, DESY, Hamburg, Germany
W. Cichalewski, B. Koseda, TUL-DMCS, Lodz, Poland

Abstract

Future and present superconducting linear accelerator projects based on superconducting resonators have tight requirements on field stability that vary with their application. The Vacuum Ultra Violet Free Electron Laser (VUV-FEL) that is currently commissioned at DESY, Hamburg, requires a stability of 10^{-3} in amplitude and 0.1 degree in phase. The upcoming X-ray Free Electron Laser (XFEL) is even more demanding by one order of magnitude. Additionally, these machines need to provide a high reliability and availability, since light sources serve as user facilities. Facing the large number of RF stations as for the case of an international linear collider, this is even more challenging. Therefore, a high degree of automation is mandatory for the Low Level RF (LLRF) control in order to accomplish for these demands. At the VUV-FEL, an automation framework based on the techniques of finite state machines has been developed and tested. It provides already a number of automated procedures that improve the operation of the VUV-FEL. It is a design goal to develop a framework that is general enough to be applied to future accelerator projects.

REQUIREMENTS FOR AUTOMATION

The demand for automation increases since electronic devices in every field have become more complex.

The requirements for automation are closely related to the requirements towards the LLRF control system of the superconducting cavities. The first and most obvious requirement is the field quality, which for the XFEL is of the order of 0.01% in amplitude and 0.01° in phase. Second, the manageability of large systems as the ILC has to be assured. Standard procedures cannot be done manually as it is common for current accelerator facilities. The third point is the availability of the machines, which is very critical in machines that serve as a user-facilities where short time-slots are foreseen for each experiment. Finally, there is a demand for more complicated RF-pulse structures and shapes in user machines, such as gradient or phase profiles within the pulse or varying shapes in consecutive pulses, which needs to be accommodated by the LLRF.

From this, we conclude the requirements for the automation system. The demand for a high machine-availability implies that the front-end presented to the operator has to be simple and the automation should protect the machine from faulty operation.

The LLRF control is a system that involves a large number of subsystems and therefore the knowledge of many people will be included in an automation system. It is therefore desirable, that actions triggered by the automation system remain transparent to the subsystem experts. Even more, it is eligible that subsystem experts can actively contribute to the automation system by adding routines to new components (modularity) and can adapt existing modules to changes in subsystems.

The automation system itself has to be designed in a way that it can be applied on top of existing infrastructures.

It has to accommodate the fact that a particle accelerator, whether it is used as a user facility or not, is a system that is permanently being optimized in the sense that hardware is changed or added. Matching the automation to new situations has to be easy and furthermore should the automation system detect potential changes in the hardware.

In order to grant maximum flexibility, the automation system needs to be switched off easily as a whole or partially in order to allow expert users perform special machine operation. Even more, it is desirable to have a mode in the automation that detects user intervention by itself and switches off parts of the automation. Therefore, operation of the LLRF system bypassing the automation has to be handled by the automation system. In this case, our automation system differs from usual industrial automation systems.

FINITE STATE MACHINES

Finite State Machines (FSMs) play a central role in our approach and shall therefore be introduced here [1]. Informally speaking the FSM is a representation of a reactive (event driven system) which can make a transition from one state (mode) to another prescribed state provided that the condition defining the change is true.

A formal definition can be given: a Finite State Machine $M = (I, O, S, \delta, \lambda)$ is a five-tuple where I is a finite nonempty set of inputs, O is a finite nonempty set of outputs, S is a finite nonempty set of states, $\delta : I \times S \rightarrow S$ is the transition function mapping

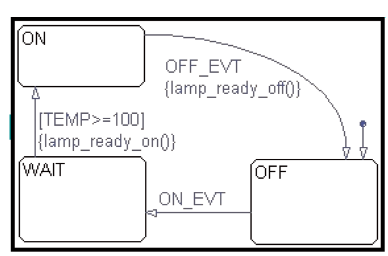

Figure 1: FSM representing a simple water heater

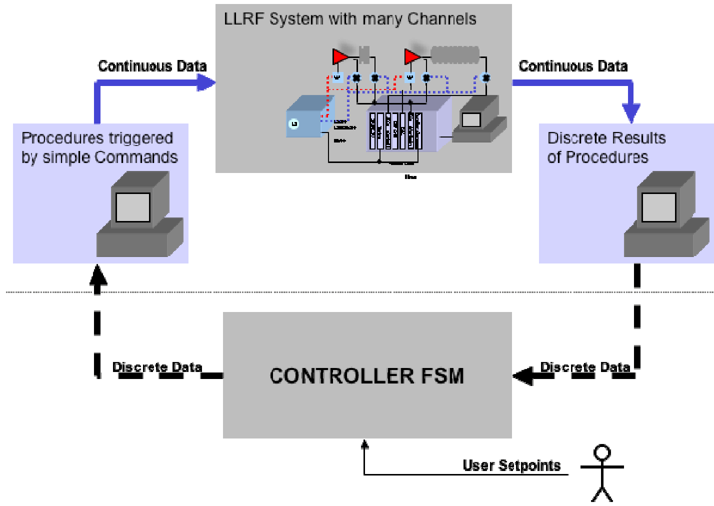

Figure 2: Quantization of LLRF Automation. Seen by the controller FSM, the complex LLRF-System appears as a system with discrete in- and output signals.

$\lambda: S \to O$ in a Moor FSM and $\lambda: I \times S \to O$ in a Mealy FSM [2].

State diagrams are used to graphically represent FSMs. There are many forms of state diagrams that differ slightly and have different semantics. A classic form of a state diagram for an FSM is a directed graph where each edge (represented by an arrow) is a transition between two nodes interpreted as states (depicted as circles, ellipses or rectangles). Harel statecharts [3] allow to model superstates, concurrent state diagrams e.g. to model activities as part of a state. Classic state diagrams are so called "or"-diagrams, because the machine can only be in one state or the other. With Harel statecharts it is possible to model "and"-machines, where a machine is in two or more states at the same time due to the introduction of superstates. Figure 1 shows a simple Harel diagram.

AUTOMATION SCHEME

FSMs model time-discrete systems with quantized input and output data flows. The complex LLRF system surely is not time-discrete and quantized, but an appropriate set of procedures can make it look discrete to the outside world. Figure 2 depicts the major elements of an automation scheme. The continuous LLRF system is accessible only via procedures that execute defined tasks like 'measure data quality', 'apply loop phase correction' or 'ramp-up the feedback gain'. Every procedure returns a quantized result like 'data quality is good', 'loop phase corrected' or 'feedback gain successfully ramped'. Based on this discretization, the controller FSM (see figure 3) can implement a strategy that meets the operator's specifications.

Procedure Attachment

The attachment of procedures that leads to an appropriate quantization has to be flexible in order to meet the requirements of automation. Subsystem experts provide procedures without awareness of the automation scheme. Therefore we propose a three-layer approach to attach procedures to the automation scheme by introducing procedure servers[1]. The procedure server (middle layer, we call it a satellite) presents itself towards the controller FSM (top layer) as whatever is the preferred communication medium of the FSM implementation. It can, for example, be the accelerators control systems communication protocol. For the test implementation at VUV-FEL we chose the DOOCS remote procedure call (RPC) protocol for communication between the FSM and its satellites. The procedures (bottom layer) are compiled programs that are executed by the satellite server on request. The subsystem expert is free to choose any toolbox for the implementation of his procedure, he only has to comply to a convention that transports results of the procedure back to the satellite server. This can easily be done via a local file system, if the satellite resides on the same physical server as its procedures.

In the presented scheme, one can imagine more sophisticated procedures (or sequences of procedures) to be implemented by another FSM. At VUV-FEL, the automation of the RF high-power amplifier is implemented as a FSM rather than a procedure, [6].

FSM Top Level View

Our design of the FSM for LLRF control is guided by operational experience. The basic idea is to provide maximum transparency to the operator by identifying tasks that can clearly be separated. These are

1. a sequence of parallel procedures needed for operation,
2. a distributed exception handling block,
3. a global observer
4. and a number of local and gobal applications.

The top level view of the state machine pictured in figure 3 incorporates these ideas. It is a top level view, therefore the depicted states may actually be superstates containing more complicated substate and flow structures.

[1] In this case, a server is a (UNIX-) process that is accessible via some network. It provides functionality to its clients, e.g. the ability to execute procedures.

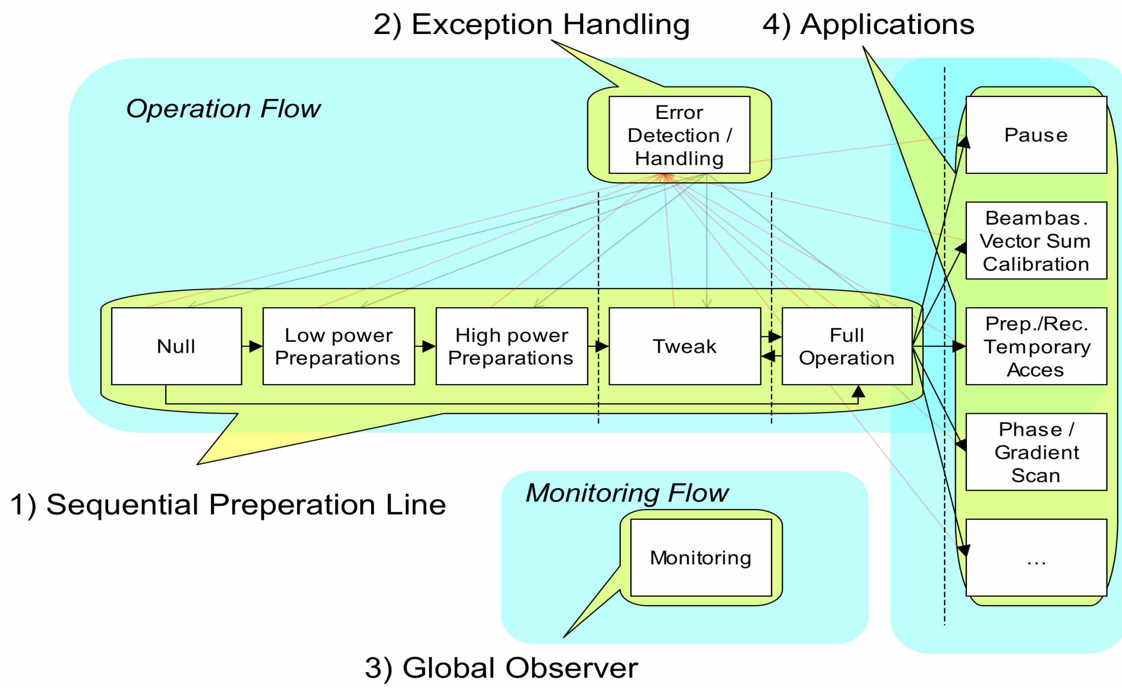

Figure 2: Top level view of the LLRF FSM

Two simultaneously active flows are used in this scheme. In the operation flow, procedures that act on the machine are triggered. Usually, only one procedure at a time is active, therefore it is reasonable to have all procedures in one flow. Another flow, labeled the Monitoring Flow, is permanently observing the machine status and eventually sending signals to the Operation Flow.

The proposed FSM state design is not the only solution to the automation problem. The FSM that is implemented for automation is not necessarily modeling the underlying complex LLRF control system. It is modeling the operator or rather a strategy used by an operator in order to reach the target state of the LLRF control system.

Heart of the strategy proposed is the sequential preparation line. Here, preparations to full operation are executed in a well-defined order. Depending on the state of the machine, a certain stage in this sequence can be activated.

PROCEDURES

Two of the procedures that contribute to an automatic start-up of an LLRF system shall be introduced here. Both are related to pulsed RF systems with one klystron controlling several cavities (vector sum).

Loop Phase and System Gain Measurement

The accelerating mode of a superconducting cavity is usually described as

$$\dot{x} + (\omega_{1/2} + i\Delta\omega)x = Z\omega_{1/2}u ,$$

where u denotes the envelope of the drive signal (the dc signal that is later upconverted to the cavity-frequency and amplified by a high power amplifier), x the envelope of the cavity field (usually the downconverted probe pick up), $\omega_{1/2}$ and $\Delta\omega$ the half-bandwidth and the detuning of the cavity. The complex quantity Z shall indicate that an arbitrary (but time-invariant) phase- and amplitude relation between the cavity-input (u) and –output (x) exists.

If the vector sum is properly tuned and the cavities have comparable properties in terms of detuning the formula is valid for vector sums, [7].

For the calculation of Z (system gain and loop phase), knowledge of

$$\dot{x}\,,\ x\,,\ \omega_{1/2}\,,\ \Delta\omega \text{ and } u$$

at a certain point in the RF pulse is sufficient.

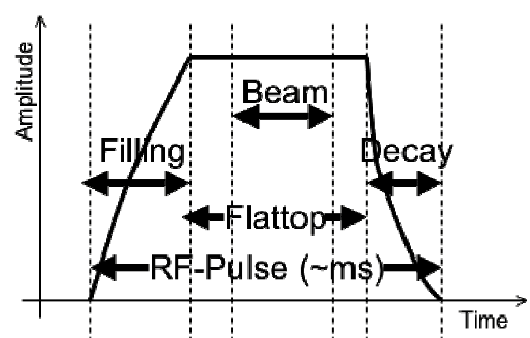

Figure 3: Pulse structure of the accelerating field in a superconducting cavitiy.

In reality, loop phase and system gain do change during the pulse, for example due to the change of the high voltage of a klystron. The calculation of especially $\omega_{1/2}$ and $\Delta\omega$ at an arbitrary point in the pulse is not trivial. The end of the flattop (see figure 4) is a good point to calculate $\omega_{1/2}$ and $\Delta\omega$. It is at a comparable power level to the rest of the flattop. Additionally, transient behaviour of the klystron high-voltage should be damped at this time. And, most important, as an approximation, $\dot{x}$, x, and u can easily be calculated shortly before the

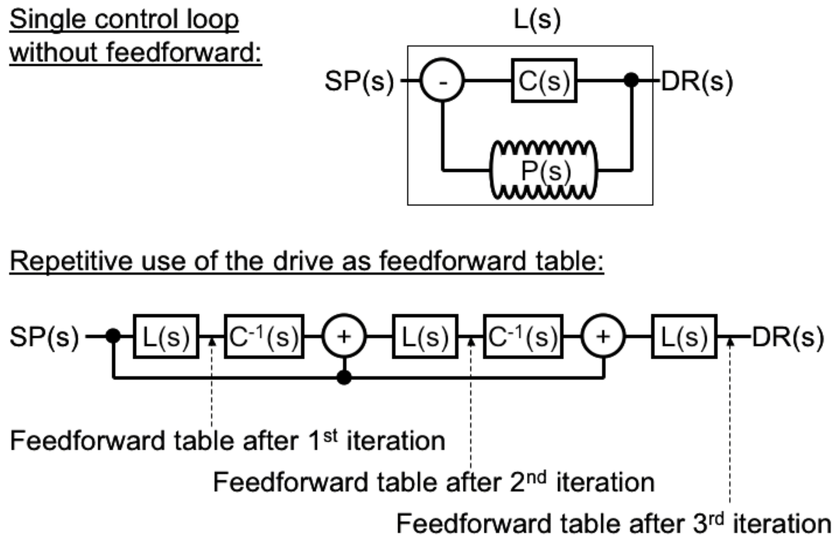

Figure 4: Control theoretical view of a very simple scheme for adaptive feedforward. SP(s) *and* DR(s) *shall be the Laplace transforms of the setpoint table of one pulse and the drive that is applied to the cavity (or vector sum).* P(s) *is the transfer function of the plant.* C(s) *is the transfer function of the controller, which e.g. can be a proportional controller.* L(s) *is the transfer function of a closed loop system.*

end of the flattop (by estimating an average value and a slope). $\omega_{1/2}$ and $\Delta\omega$ are easily obtained shortly after the pulse. The decay of the amplitude of the cavity field envelope yields the half-bandwidth $\omega_{1/2}$ while the change of the phase of the cavity field yields the detuning $\Delta\omega$ at the pulse end.

The procedure that implements the correction-algorithm for loop phase and system gain carefully checks all its prerequisites and returns a discrete, numerical value indicating the success of the invocation. It is run in the Tweak-state of the FSM-layout shown in figure 3. A derived algorithm that does not correct but just monitor the development of Z is invoked periodically in the Monitoring-flow.

Adaptive Feedforward

A very important part of automatic parameter estimation during start-up is the adaptive feedforward. [7] gives an introduction to the concept of driving a LLRF system with feedback and feedforward.

Besides the algorithm that is used within this project, we will analyse first another method that appears to be an obvious way of acquiring an optimal feedforward table but turned out to be instable.

That is to take the drive of one pulse (feedback plus feedforward) and use it as the feedforward for the next pulse. In order to cope with such an approach in the language of control theory, we formulate this situation in a different way. Instead of applying the drive of one cavity as feedforward of the same cavity in a later pulse, we apply the drive of one cavity as the feedforward of a second identical cavity in the same pulse. Since we assume that we have no time delays, these two situations are identical. Figure 5 depicts this formulation. The top part of the picture shows a single control loop with a plant *P(s)* and a controller *C(s)*. The whole closed loop transfer function *L(s)* then transforms from the setpoint table *SP(s)* to the total drive output *DR(s)* of this loop. In order to apply the drive *DR(s)* as the feedforward table of the next pulse, one would have to add this signal somewhere before the plant *P(s)*, after the controller *C(s)* in the next control loop. Or, one can directly add it to the setpoint table of the next pulse if one compensates for the controller *C(s)* by applying its inverse to the drive. The lower part of figure 5 shows this situation.

The transfer function from the setpoint table to the *n*-th iteration of this adaptive feedforward scheme can easily be calculated from figure 5. From

$$H = \left(\left(\left(L \cdot C^{-1} + 1\right) \cdot L \cdot C^{-1} + 1\right) \ldots\right) \cdot L$$

with *H(s)* being the transfer function after *n* iterations, we get

$$H = \left[\left(L \cdot C^{-1}\right)^{n} + \left(L \cdot C^{-1}\right)^{n-1} + \ldots + 1\right] \cdot L .$$

That is a geometric series, therefore

$$H = \left(\left(L \cdot C^{-1}\right)^{n+1} - 1\right) / \left(\left(L \cdot C^{-1}\right) - 1\right) \cdot L .$$

From the single closed loop transfer function $L = C/(1+CP)$ we get $P = (1 - L \cdot C^{-1})/L$ and then

$$H = \left(1 - \left(L \cdot C^{-1}\right)^{n+1}\right) \cdot P^{-1} .$$

This is not an unexpected result. For $n \rightarrow \infty$, one gets $H \rightarrow P^{-1}$, which is what one would expect from a good feedforward adaption. However, for finite *n*, *H* is a transfer function with many poles and zeros, due to the

‘1’ in the formula for H. This leads to unwanted oscillations and has been observed several times.

Therefore we developed a more heuristic approach that by design provides a good damping of oscillations.

First, we divide the RF-pulse (see figure 4) into sections of certain lengths. The borders of the sections should coincide with events like start and stop of the beam and the flattop. The length of the sections is determined by the noise of the system – the less noise, the shorter the sections. For each section, we determine an average of the vector sum and a slope and subtract it from the setpoint and its slope. Therefore, for each section we get Δx and $\Delta \dot{x}$ (using the same nomenclature as above where x is the complex envelope of the vector sum). From Δx and $\Delta \dot{x}$ one can estimate an “equilibrium error” Δx_{eq}, which is the difference between vector sum x and the setpoint x_{SP}, given that the drive and the setpoint will remain constant all the time:

$$\Delta x_{eq} \approx \Delta x + \Delta \dot{x} / \omega_{1/2}$$

This of course neglects dynamic effects caused by the detuning. After having Δx_{eq} for every section, one scales the drive u for each section with the complex factor $(1 + \Delta x_{eq} / x_{SP})$. Figure 6 shows a small part of the flattop and gives an idea of the partitioning.

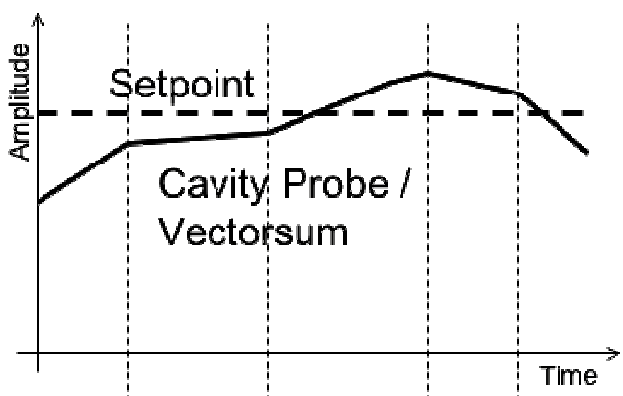

Figure 5: Small part of the flattop of a pulse. The dashed vertical lines define sections, along which an average error Δx and and average slope error $\Delta \dot{x}$ are determined.

Experience with this algorithm show that oscillations do not occur during operation. This algorithm shows convergence with and without feedback.

A refining of this algorithm can be applied in terms of if-statements, like “if the feedback applied in one section is larger than the current feedforward, use the feedback as feedforward in the next pulse”. This assures a faster convergence. Additional refinements are thinkable.

The algorithm has been successfully tested at moderate beam-load and is subject to further tests, especially with high beam-load.

STATUS OF AUTOMATION AT VUV-FEL

Currently, an implementation of the application server is running at the VUV-FEL at DESY. It provides procedures for offset compensation, intelligent coupler interlock reset, adaptive generation of feedforward tables, careful ramping of setpoint values, careful ramping of feedback gain, data quality check, loopphase and systemgain determination, network availability check, detuning measurement, and vectorsum calibration. Additionally, for the complex klystron subsystem, a dedicated FSM has been implemented. All components were carefully tested and it has been shown that all so-called expert parameters in the LLRF system of the VUV-FEL could be set automatically. This includes a startup of the system with unknown parameters and a check for basic inconsistencies like the phase orientation in the control loop. It does not yet cover automatic tuning of cavities that are off-resonance or have wrong phases at their coupler input. Still, wrong phases and wrong tuning is detected by the automation and reported to the operator.

Furthermore, an implementation of the presented FSM is permanently running at VUV-FEL and is available to the operators. It allows experienced operators to change procedures and conditions for state-transitions on the fly via the user interface of the control system. Presently, the automation is undergoing intense tests. It is not yet a part of the normal operation.

ACKNOWLEDGEMENTS

We acknowledge the support of the European Community-Research Infrastructure Activity under the FP6 "Structuring the European Research Area" program (CARE, contract number RII3-CT-2003-506395).

REFERENCES

[1] David Gibson, “Making simple work of complex functions”, © 1998-2000 Splat Controls Pty Ltd.

[2] Sherif M. Yacoub, Hany H. Ammar, “Finite State Machine Patterns”, Third European Conference on Pattern Languages of Programming and Computing 9-11 July, Bad Irsee, Germany.

[3] David Harel, “Statecharts: A Visual Formalism for Complex Systems” Science of Computer Programming 8, 1987, pp. 231-274.

[4] Chonlawit Banphawatthanarak, Ken Butts, “Symbolic Verification of Executable Control Specifications”, Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design Kohala Coast-Island of Hawai'i, USA August 22-27, 1999.

[5] Pushmeet Kohli “A Classifier System based on Evolved Finite State Machines”, 1st Indian International Conference On Artificial Intelligence, 2003.

[6] Boguslaw P. Koseda, Wojchiech Cichalewski, “Design and Implementation of Finite State Machine for RF Power Station”, 12th International Conference Mixed Design of Integrated Circuits and Systems, Kraków, 22-25 June 2005.

[7] Thomas Schilcher, “Vector-sum Control of Pulsed Accelerating Fields in Lorentz Force detuned Superconducting Cavities”, DESY Print TESLA 98-05, 1998.