

FUNCTION GENERATION AND REGULATION LIBRARIES AND THEIR APPLICATION TO THE CONTROL OF THE NEW MAIN POWER CONVERTER (POPS) AT THE CERN CPS

Q. King, S. T. Page, H. Thiesen, CERN, Geneva, Switzerland
M. Veenstra, EBG MedAustron, Wr. Neustadt, Austria

Abstract

Power converter control for the LHC is based on an embedded control computer called a Function Generator/Controller (FGC). Every converter includes an FGC with responsibility for the generation of the reference current as a function of time and the regulation of the circuit current, as well as control of the converter state [1]. With many new converter controls software classes in development it was decided to generalise several key components of the FGC software in the form of C libraries: function generation in libfg, regulation, limits and simulation in libreg and DCCT, ADC and DAC calibration in libcal. These libraries were first used in the software class dedicated to controlling the new 60MW main power converter (POPS) at the CERN Proton Synchrotron (CPS) where regulation of both magnetic field and circuit current is supported. This paper reports on the functionality provided by each library and in particular libfg and libreg. The libraries are already being used by software classes in development for the next generation FGC for Linac4 converters, as well as the CERN SPS converter controls (MUGEF) and MedAustron converter regulation board.

FUNCTION GENERATION LIBRARY

Calculating a reference value as a function of time is a key requirement for most regulated systems. Libfg is a C library that can generate ten different types of reference function, as listed in Table 1. For each type there are two C structures; one for the configuration of the reference function (config) and the other containing the parameters (pars) needed to generate the reference at a given time. Figure 1 shows how these structures relate to the associated C functions *Init()* and *Gen()*.

To summarise, the *Init()* function processes the config structure and calculates the contents of the pars structure. This is then used by the *Gen()* function to calculate the reference at the specified time.

Two other structures are concerned: limits and meta. Limits are provided to the *Init()* function and allow the

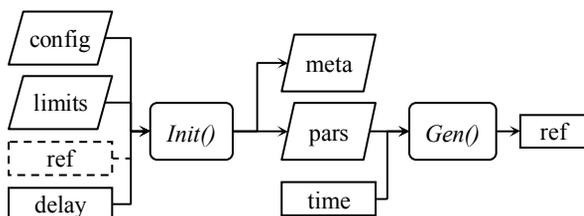


Figure 1: libfg data flow.

Table 1: Libfg Function Types

Name	Function type
PLEP	Single parabola-linear-exponential-parabola segment
PPPL	Multiple parabola-parabola-parabola-linear segments
TABLE	Table interpolated with linear segments
SPLINE	Table interpolated with parabolic splines
LINEAR	Single linear segment in a given time
CUBIC	Single cubic segment in a given time
SINE	Sine with optional window for smooth start/end
COSINE	Cosine with optional window for smooth start/end
SQUARE	Offset square wave
STEPS	Staircase of one or more steps

absolute, rate of change and acceleration limits to be checked. Meta is produced by the *Init* function with summary information (duration, min/max and start/end reference values). The *Init* function also needs to know the initial time delay and in most cases the initial reference value since most reference functions are relative to an initial reference. Only TABLE and SPLINE define absolutely the entire function and do not need it.

Reference and Time Units

Libfg does not dictate the interpretation of the reference. For magnet circuit control it might be voltage, current or field, depending upon the regulation mode. Similarly, libfg does not dictate the units of time, though at CERN the units are always seconds.

PLEP

The PLEP function provides a standard way to change the magnet circuit current. The initial and final parabolas provide a smooth start and end, which is vital for superconducting circuits. The linear section takes over if the rate limit is reached and the exponential section may be needed when ramping down with a 1-quadrant converter. These are unipolar in current and voltage and so the current cannot be ramped down faster than the natural decay rate for the circuit ($T_c = L/R$). Thus when a PLEP is initialised, the result might be a P-P, a P-L-P, a P-E-P or a P-L-E-P depending upon the parameters and the difference between the initial and final reference values.

The PLEP function is special amongst all the functions in libfg because it can be initialised with a non-zero initial rate of change. This can be important if a running

function needs to be aborted smoothly, which is the case in the LHC if cryogenics report an impending problem.

PPPL

The PPPL reference was created for the CPS main magnet controls. The field is ramped up in stages with a series of linear plateaus defined parametrically using seven values. These specify a fast parabolic acceleration followed by a slow parabolic deceleration, then a fast parabolic deceleration and finally a linear section that is not necessarily constant.

TABLE

TABLE provides linear interpolation between points and is the main function type for LHC current references. Linear interpolation is sufficient provided points are close together when the reference is changing rapidly. If the reference should follow a parabola then equation (1) is a useful formula that gives the point spacing t as a function of the maximum acceptable interpolation error ε and the parabolic acceleration a :

$$t = 2\sqrt{(2\varepsilon/a)} \tag{1}$$

SPLINE

SPLINE offers a way to use fewer points than TABLE by using parabolic splines rather than linear interpolation.

TRIM Functions

Trim functions are useful for small changes in the reference, especially when many circuits must change synchronously, since unlike the PLEP function, the duration for the change is an input parameter. CUBIC trims are essential for super-conducting circuits because they avoid discontinuities in the rate of change which would generate voltage spikes. LINEAR trims can be used for warm circuits but they suffer from undershoot and overshoot and are not used in the LHC.

TEST Functions

The SINE, COSINE, SQUARE and STEPS functions are very useful for converter testing and in some cases for accelerator measurements (e.g. K-modulation).

Definition of Time

Libfg uses single precision floats for time (and all other reference related parameters). The resolution of IEEE 32-bit floating point values is limited by the 23-bit mantissa and will allow 1 μ s granularity for up to 10 s, 10 μ s granularity up to 100 s and so on. If finer granularity is required then the library can be modified and recompiled to use double precision floats.

REGULATION LIBRARY

Libreg provides all the key components needed to write a program to regulate magnetic field or circuit current.

Magnet Load Model

Figure 2 shows the load model used by libreg. It is based on a magnet with inductance L and resistance R_m (zero for superconducting magnets). This is associated with a parallel damping resistor R_p and a series resistor R_s which corresponds to the cables or bus bars leading to the magnet. In many case multiple magnets are connected in series but for low frequencies their individual impedances can be combined into the four values given in figure 2.

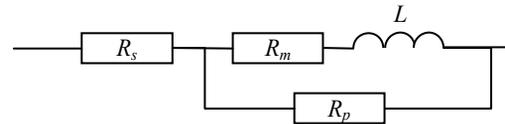


Figure 2: libreg load model.

The gain response of this model is first order and is shown in figure 3.

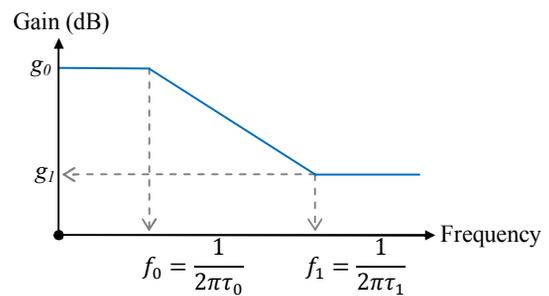


Figure 3: Load model Bode plot.

The pole and zero gains and time constants associated with the break frequencies in the Bode plot are given by:

$$g_0 = \frac{1}{R_s + \frac{R_p R_m}{R_p + R_m}} \quad g_1 = \frac{1}{R_p + R_s} \tag{2}$$

$$\tau_0 = \frac{L}{R_m + \frac{R_p R_s}{R_p + R_s}} \quad \tau_1 = \frac{L}{R_p + R_m} \tag{3}$$

The gains g_0 and g_1 correspond to the effective load resistance at zero and infinite frequency, while the time constants τ_0 and τ_1 correspond to the effective circuit resistance as seen by the inductance for shorted and open circuits.

Reference and Measurement Limits

Libreg provides functions that implement limits on the current or field and voltage references and the current or field measurement. These have different objectives. The measurement limit is used for protection of the circuit; the converter will be tripped if the limit is exceeded. The reference limits are associated with constraints in the converter and circuit and may cause the reference to be clipped. Normally limits are applied by libfg when initialising a reference function, however if a real-time reference is included then real-time limits are required.

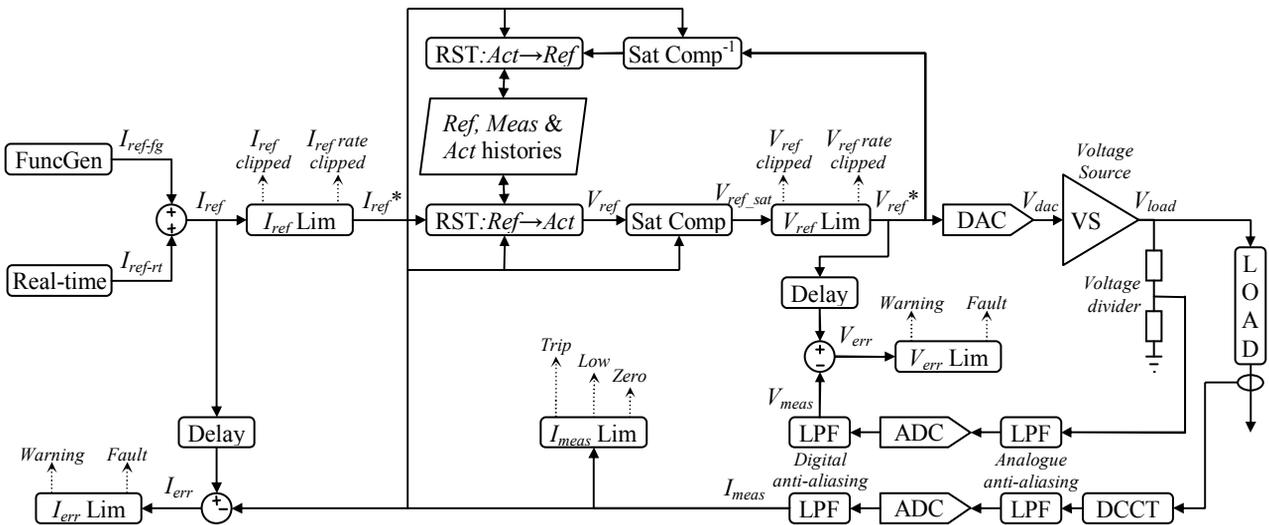


Figure 4: Overview of current regulation of an analogue voltage source.

RST Regulation

Figure 4 presents an overview of the function generation and regulation that can be implemented using libfg and libreg. One of the most flexible ways to implement a linear regulator is with the RST equation:

$$\sum_0^N \{Act_i \cdot S_i\} = \sum_0^N \{Ref_i \cdot T_i\} - \sum_0^N \{Meas_i \cdot R_i\} \quad (4)$$

Where $i=0$ corresponds to the current sample, $i=1$ is the previous sample and so on. This notation was proposed by Landau [2], however in many text books the R and S polynomials are exchanged. By keeping the history of the previous N samples of the reference, measurement and actuation it is easy to calculate the new actuation if you know the new reference and measurement:

$$Act_0 = \frac{\sum_0^N \{Ref_i \cdot T_i\} - \sum_0^N \{Meas_i \cdot R_i\} - \sum_1^N \{Act_i \cdot S_i\}}{S_0} \quad (5)$$

Equally, when the actuation is limited or is being driven in open-loop then it is equally easy to back-calculate the reference which, when combined with the new measurement will result in this actuation:

$$Ref_0 = \frac{\sum_0^N \{Act_i \cdot S_i\} + \sum_0^N \{Meas_i \cdot R_i\} - \sum_1^N \{Ref_i \cdot T_i\}}{T_0} \quad (6)$$

In this way the reference history can be kept coherent with the measurement and actuation histories. This is equivalent to the anti-windup feature of a traditional regulation algorithm.

The benefit of the RST equation is that any linear regulator up to order N can be implemented by choosing the appropriate RST polynomial coefficients. Simple PI, PID, or PII controllers can be implemented [3] as well as more complex higher order systems, *without changing the software*. Of course, for complex higher-order systems calculating the coefficients is a challenge.

For a magnet circuit the reference and measurement can either be of the circuit current or magnetic field. The actuation defines the circuit voltage which the voltage source must try to follow.

If the required bandwidth of the current or field regulation is much less than the bandwidth of the voltage source and the bandwidth of the reference is lower still then a deadbeat controller can be used. Since this is exactly the case for the circuits in the LHC the regulation library includes a function to calculate the RST coefficients that will implement a deadbeat PII controller. This works extremely well in the LHC with negligible tracking error.

For applications that require as much bandwidth as possible the transfer function of the voltage source must be accurately modelled and a different approach taken to derive the RST coefficients. Matlab has a powerful toolkit for this, however it does require expert knowledge of automatic control theory.

Magnet Saturation Model

If warm magnets are used beyond about 0.9 teslas, their iron yokes will enter saturation and the inductance drops by as much as 60%. The corresponding change in time constant can destabilise the regulation loop if it is not compensated. Libreg supports compensation of this effect based on the simple linear model of the magnet inductance $L_m(I)$ shown in figure 5. Three parameters are used to characterise the model: L_{sat} , I_{sat_start} and I_{sat_end} .

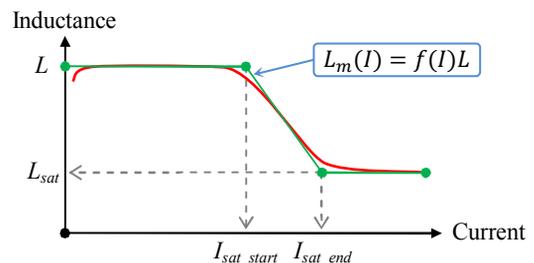


Figure 5: Magnet saturation model.

The saturation model is used to transform V_{ref} in order to hide the saturation effect from the RST algorithm:

$$V_{ref_sat} = \{1 - f(I)\}IR + f(I)V_{ref} \quad (7)$$

Where $f(I) = L_m(I)/L$ and R is the load resistance associated with magnet load pole:

$$R = R_s + \frac{R_p R_m}{R_p + R_m} \quad (8)$$

As shown in figure 4, equation (7) is used to derive V_{ref_sat} from V_{ref} which is then limited to arrive at V_{ref}^* . If clipping occurs then equation (7) is inverted to back-calculate V_{ref} from V_{ref}^* and equation (6) is used to back-calculate the reference to keep the RST histories coherent.

Note that when regulating magnetic field, saturation is a second order effect and compensation is not required.

Regulation Error

Figure 4 shows how the regulation error is calculated from the difference between the delayed reference and the measurement. The delay must be set to the tracking delay of the regulator. Libreg includes functions that allow the error to be calculated and limits to be applied.

Simulation of Voltage Source and Load

It is highly desirable to build into converter controls software a simulation mode in which the voltage source, load, and measurements are simulated in software. Libreg includes functions that support a third-order model of the voltage source and simulation of the load model presented above.

FGTEST PROGRAM

A program called fgtest was written under Linux to test libfg and libreg. It is a template for a real-time converter control program, showing how the core elements of the regulation loop presented in figure 4 can be implemented. The program is driven from the command line using parameter files and writes signals to stdout as comma separated values that can be analysed with any suitable tool such as GNUplot, Matlab or Excel.

LIBCAL

Libcal contains functions to support the calibration of voltage dividers, DCCTs, ADCs and DACs, including the option for up to second order temperature compensation for ADCs and DCCTs. More information and the library source files can be downloaded from <http://cern.ch/cclibs>.

APPLICATIONS

Libfg and libreg were created to support multiple platforms and applications. So far they have been compiled on Linux PCs and TMS320C32 and TMS320C6727 DSPs and from 2012 they will be used in the converter controls software for the CERN LHC, SPS,

CPS and PSB accelerators. They will also be used in the new converter controls for the MedAustron accelerator.

For now, only one converter in the CPS accelerator is controlled using the libraries; the new 60 MW POPS converter that drives the main magnets. This is a particularly interesting case since it can regulate magnetic field or circuit current on a cycle by cycle basis. The magnets saturate dramatically above 3.6 kA with a 60% reduction in inductance by the maximum current of 6 kA. This is successfully compensated when regulating current using equation (7).

Figure 6 shows a 1.2 s proton beam cycle with three plateaus. The blue and cyan traces are the field reference and measurement; red shows the current and purple and green show the voltage reference and measurement. The controller runs at 1 kHz, but the regulation acts every 3rd iteration at 333.3 Hz.

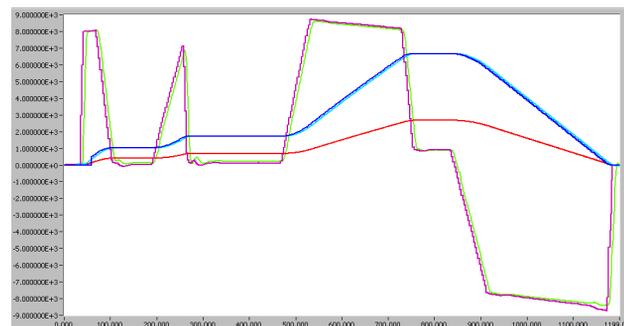


Figure 6: 1.2s cycle with the POPS converter.

AVAILABILITY FOR DOWNLOAD

All the libraries (including libcal) and the fgtest program are freely available under the GNU Lesser General Public License. Visit the website <http://cern.ch/cclibs> for more information and to download the source. Makefiles for Linux are included.

CONCLUSIONS

These well tested libraries are based on years of operational experience [4] and are a valuable resource for the development of converter regulation software. By centralising these vital components into libraries, the maintenance effort has been reduced and new equipment classes have been developed more quickly.

REFERENCES

- [1] Q. King et al, "The All-Digital approach to LHC power converter current control", ICALEPCS'01, THBT004
- [2] I.D. Landau, "System Identification and control design", Prentice-Hall International, 1990
- [3] F. Bordry, H. Thiesen, "RST Digital Algorithm for controlling the LHC magnet current", CERN/LHC Project report 258
- [4] Q. King, "Status of the LHC power converter controls", ICALEPCS'09, MOB003