

Date: 2013-11-17

User Guide

CCTEST (V1.5)

TEST PROGRAM FOR LIBFG AND LIBREG

Author

**Quentin King
CERN TE-EPC-CC**

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. 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. A test program called cctest was written to test these libraries and to provide an example for how converter control software could be written using them.

Table of Contents

1. INTRODUCTION 3

1.1 OVERVIEW..... 3

2. RUNNING CCTEST 4

2.1 COMMAND LINE OPTIONS..... 5

2.2 PARAMETER FILES 7

2.3 FUNCTION DATA FILES.....17

2.4 OUTPUT.....18

3. PROGRAM STRUCTURE..... 19

3.1 SOURCE FILES19

1. INTRODUCTION

1.1 OVERVIEW

cctest is a C program written under Linux that will test **libfg** and **libreg** and also demonstrates how the libraries can be combined to create a power converter control program. It is driven by command line options that mostly specify the names of text files containing all the parameters for a run.

It has two principle modes of operation, with a number of sub-modes:

- Function generation only
 - With normal increasing time
 - With reverse decreasing time
- Function generation *and* simulation of the voltage source and load
 - Open-loop (function defines voltage reference)
 - Closed-loop (function defines current or field reference and RST algorithm calculates the voltage reference)

The program writes ASCII output to standard error and standard out:

- Standard error is used to report errors that stop execution. If the verbose option (-v) is given then it will also be used to print all the parameter values before the run begins.
- Standard out is used to report the time sequence for all the signals during the run. Four different output formats are supported; three generate comma separated values and one generates FLOT graphs in html that can be examined with a web browser.

The units of time and frequency are not specified in **cctest**, nor in **libfg** or **libreg**. Normally seconds and hertz will be used, but this is not proscribed and you are free to use any units you like.

2. RUNNING CCTEST

Between function generation, load and voltage source simulation, limits, and regulation, there are a lot of parameters needed to drive **cctest**. These are gathered together in a small number of ASCII files with a very simple syntax:

- Lines can contain a parameter name and value, separated by white space.
- If the parameter is an array then multiple values are separated by commas.
- Comment lines start with "#".
- Blank lines and leading white space are ignored.

Here is a simple example:

```
# CCTEST global control file for function generation tests with normal time

RUN_DELAY      1.0
STOP_DELAY     1.0
ITER_PERIOD    1.0E-3
UNITS          VOLTS

# EOF
```

In this documentation, a parameter will be referred to by combining its group and its name separated by a period. For example, GLOBAL.ITER_PERIOD is 1.0E-3 in the example above.

Here is how **cctest** might be run:

```
cctest -g global -d table -m limits -l load -r reg -s vs > spline.csv
```

In this case six files (global, table, limits, load, reg and vs) containing all the parameters are read by **cctest** and the results are redirected to the file spline.csv.

In total there are only five parameter groups:

1. GLOBAL (-g)
2. LOAD (-l)
3. LIMITS (-m)
4. VS (-s)
5. REG (-r)

And there are six types of function data file (all use -d since only one can be specified per run of cctest):

1. START
2. PLEP
3. PPPL
4. TABLE (used with the TABLE and SPLINE functions)
5. TRIM (used with the LTRIM and CTRIM functions)
6. TEST (used with the SINE, COSINE, SQUARE and STEPS functions)

2.1 COMMAND LINE OPTIONS

cctest is completely controlled by the following command line options:

Option	Action	Required?	Note
-v	Enables verbose mode	No	All parameters are printed to standard error before starting the run.
-o	Specifies the Output Format	No	When set it will override the GLOBAL.OUTPUT_FORMAT parameter.
-f FUNCTION	Specifies reference function (PLEP, TABLE, PPPL, etc...)	No	The function can be defined in the global parameter file or as a separate command line option. This can reduce the number of global parameters files.
-g global_file	Specifies global parameter file name	Yes	The global parameter file can contain the function type (PLEP, TABLE, PPPL, etc...) or it can be specified using the -f command line option.
-d function_data_file	Specifies the reference function data.	Yes	The function data file must match the function type specified in the global parameter file or using the -f command line option.
-m limits_file	Specifies the limit parameters file name.	No	Field, current and voltage limits can be used by libfg to check that the function being initialised will not exceed any of the limits. It can also be used during the run (in "real-time") to clip the reference and to trip the simulated converter if the measured field or current exceeds the limits.
-l load_file	Specifies the load parameters file name.	No	These parameters are required if simulation is enabled.
-s vs_file	Specifies the voltage source parameters file name.	No	These parameters are required if simulation is enabled.
-r regulation_file	Specifies the regulation parameters file name.	No	These parameters are required if regulation of field or current is enabled.

There are several interdependencies between options and certain global parameters which are summarised in this table:

REVERSE_TIME	UNITS	FG_LIMITS	SIM_LOAD	-m limits	-l load	-s vs	-r reg
ENABLED	VOLTS	DISABLED	DISABLED	EXCLUDED	EXCLUDED	EXCLUDED	EXCLUDED
DISABLED	VOLTS	DISABLED	DISABLED	EXCLUDED	EXCLUDED	EXCLUDED	EXCLUDED
			ENABLED	REQUIRED	REQUIRED	OPTIONAL	EXCLUDED
		ENABLED	DISABLED	REQUIRED	EXCLUDED	EXCLUDED	EXCLUDED
			ENABLED	REQUIRED	REQUIRED	OPTIONAL	EXCLUDED
	AMPS or GAUSS	DISABLED	ENABLED	REQUIRED	REQUIRED	OPTIONAL	REQUIRED
		ENABLED		REQUIRED	REQUIRED	OPTIONAL	REQUIRED

Two other dependencies are checked by cctest:

- The START function can only be specified if GLOBAL.UNITS is set to AMPS or GAUSS and GLOBAL.FG_LIMITS is ENABLED.
- GLOBAL.ABORT_TIME can only be greater than zero if GLOBAL.UNITS is set to AMPS or GAUSS and GLOBAL.FG_LIMITS is ENABLED.

2.2 PARAMETER FILES

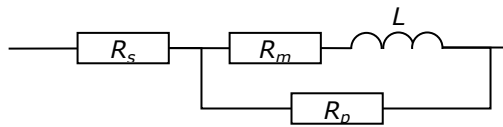
2.2.1 Global parameters file (-g global_file)

The global parameters file must be specified, however, only one parameter is mandatory (ITER_PERIOD). All the rest have default values.

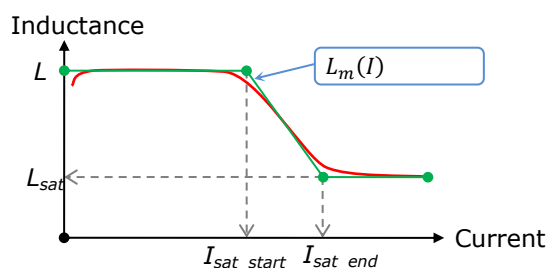
GLOBAL parameter	Type or ENUM	Default	Comments
RUN_DELAY	Float	1.0	Time before start of function
STOP_DELAY	Float	1.0	Time after end of function
ITER_PERIOD	Float	None	Base iteration period – this defines the rate at which the function is sampled and the simulation executed (if enabled). Current or field regulation can run at a multiple of this period.
ABORT_TIME	Float	0.0	Time at which the Current or Field reference function should be aborted. This involves switching to a PLEP function that smoothly takes over from the running reference function and ramps down to the MIN value provided in the limits.
REVERSE_TIME	ENABLED DISABLED	DISABLED	When enabled the function is sampled with decreasing times to test if the function handles time going backwards.
UNITS	VOLTS AMPS GAUSS	VOLTS	Selecting VOLTS disables feedback. Selecting AMPS or GAUSS will regulate this quantity by calculating the voltage reference to be sent to the voltage source.
FUNCTION	START PLEP PPPL SPLINE TABLE STEPS SQUARE SINE COSINE LTRIM CTRIM	None	The function does not have to be specified in the global parameters file. It can be specified instead on the command line using “-f”.
FG_LIMITS	ENABLED DISABLED	DISABLED	This option controls whether limits are checked by libfg when initialising the reference function. The same limits are used during the simulation of the voltage source and load so to test these limits with a function that exceeds them, FG_LIMITS will need to be disabled.
SIM_LOAD	ENABLED DISABLED	DISABLED	When SIM_LOAD is disabled only the function generation is tested. When enabled, the voltage source and load will also be simulated.
OUTPUT_FORMAT	STANDARD FGCSPY LVDV FLOT	FGCSPY	STANDARD targets Excel or equivalent. FGCSPY adds the suffix “_D” to the signal names for signals that need trailing step interpolation. LVDV adds a meta data header line and cursors for the Labview Dataviewer (in development). FLOT generates HTML files that use the FLOT javascript graphing library (based on jQuery).

2.2.2 Load parameters file (-l load_file)

The load parameters file must be specified if GLOBAL.SIM_LOAD is ENABLED. The regulation library implements the following load model:



Furthermore, non-superconducting magnets may be affected by saturation of their iron yokes, resulting in a reduction in their inductance as the current increases. This can be by as much as 60% which can destabilise the current regulation if it is not compensated. Libreg supports compensation of saturation using a simple linear model as shown in this figure:



The saturation is summarised by three parameters: L_{sat} , I_{sat_start} and I_{sat_end} . Despite being rather crude, this model works very well and hides the non-linearity of the load from the RST regulation algorithm.

Note that magnet saturation is not a problem when regulating the magnetic field. In this case it is a second-order effect and can be neglected.

Here is the list of LOAD parameters.

LOAD parameter	Type or ENUM	Default	Comments
OHMS_SER	Float	1.0	Series resistance (R_s) corresponding to the resistance of the cables from the power converter to the magnet(s).
OHMS_PAR	Float	1.0E+9	Parallel damping resistance (R_p).
OHMS_MAG	Float	1.0	Magnet resistance (R_m).
HENRYS	Float	1.0	Magnet inductance without saturation effects (L).
HENRYS_SAT	Float	1.0	Magnet inductance when fully saturated (L_{sat}).
I_SAT_START	Float	0.0	Current at start of magnet saturation (I_{sat_start}).
I_SAT_END	Float	0.0	Current at which the magnet is fully saturated (I_{sat_end}). This is set to 0.0 if the magnet does not saturate.
GAUSS_PER_AMP	Float	1.0	Field to current ratio without saturation effects.
I_MEAS_DELAY	Float	0.0	Current measurement delay. This models the delay of any analogue anti-aliasing filter, ADC sampling and digital filtering as a simple delay.

LOAD parameter	Type or ENUM	Default	Comments
B_MEAS_DELAY	Float	0.0	Field measurement delay. This models the delay of filtering in the field measurement system. Generally this can be almost zero because of the techniques used to measure the field.
PERTURB_VOLTS	Float	0.0	Voltage perturbation level. This voltage will appear as a disturbance on the simulated load at the time given by LOAD.PERTURB_TIME and will remain until the end of the run. This is useful to see the rejection of perturbations by the regulation loop.
PERTURB_TIME	Float	0.0	Voltage perturbation time. This is time at which LOAD.PERTURB_VOLTS will appear as a perturbation on the simulated load. This is useful to see the rejection of perturbations by the regulation loop.
SIM_TC_ERROR	Float	0.0	<p>Simulation time constant error factor. In order to evaluate the robustness of the regulation, the simulated load parameters can be distorted compared to the load parameters given in this file and which may (optionally) be used to calculate the RST coefficients. The distortion results in a shift in the time constant by a factor related to LOAD.SIM_TC_ERROR.</p> <p>The simulated load parameters will be adjusted as follows:</p> $L = \text{HENRYS} * (1 + \text{SIM_TC_FACTOR})$ $L_{\text{sat}} = \text{HENRYS_SAT} * (1 + \text{SIM_TC_FACTOR})$ $R_s = \text{OHMS_SER} * (1 - \text{SIM_TC_FACTOR})$ $R_p = \text{OHMS_PAR} * (1 - \text{SIM_TC_FACTOR})$ $R_m = \text{OHMS_MAG} * (1 - \text{SIM_TC_FACTOR})$ <p>Where:</p> $\text{SIM_TC_FACTOR} = \frac{\text{SIM_TC_ERROR}}{\text{SIM_TC_ERROR} + 2}$ <p>If SIM_TC_ERROR is 0.1 the time constant of the simulated load will be 10% greater than expected by the RST regulation if the RST coefficients are calculated based on the load parameters.</p>

2.2.3 Limit parameters file (-m limits_file)

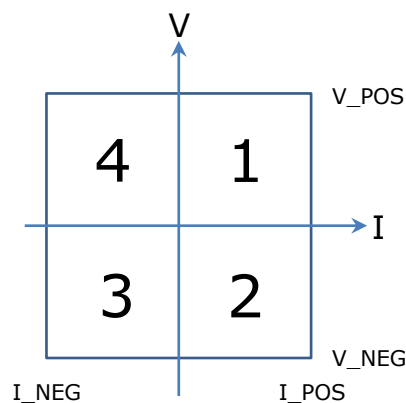
These limits can be used during the arming of the reference function of GLOBAL.FG_LIMITS is ENABLED. They will also be used if GLOBAL.SIM_LOAD is ENABLED.

LIMITS Parameter	Type or ENUM	Default	Comments
B_POS	Float	1.0E+9	Max positive field.
B_MIN	Float	0.0	Minimum field reference. This can be zero for a 4-quadrant converter, otherwise it should indicate the minimum field that can be regulated by the converter.
B_NEG	Float	-1.0E+9	Maximum negative field. This should be zero for a 1 or 2-quadrant converter.
B_RATE	Float	1.0E+9	Maximum absolute rate of change of field. If this is set to zero then the rate check is disabled.
B_ACCELERATION	Float	1.0E+9	Maximum absolute field acceleration. If this is set to zero the acceleration check are disabled.
B_ERR_WARNING	Float	0.0	Field regulation error warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than 3 times the tracking delay. If the value is zero the warning check is disabled.
B_ERR_FAULT	Float	0.0	Field regulation error warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than 3 times the tracking delay. If the value is zero the fault check is disabled.
I_POS	Float	1.0E+9	Max positive current.
I_MIN	Float	0.0	Minimum current reference. This can be zero for a 4-quadrant converter, otherwise it should indicate the minimum current that can be regulated by the converter.
I_NEG	Float	-1.0E+9	Maximum negative current. This should be zero for a 1 or 2-quadrant converter.
I_RATE	Float	1.0E+9	Maximum absolute rate of change of current. If set to zero then the rate check is disabled.
I_ACCELERATION	Float	1.0E+9	Maximum absolute field acceleration. If set to zero the acceleration check are disabled.
I_ERR_WARNING	Float	0.0	Current regulation error warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than 3 time the track delay. If the value is zero the fault check is disabled.
I_ERR_FAULT	Float	0.0	Current regulation error warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than 3 time the track delay. If the value is zero the fault check is disabled.
I_QUADRANTS41	Float	0.0,0.0	Together with V_QUADRANTS41 this allows the voltage to be limited as a function of the current (see below).
V_POS	Float	1.0E+9	Max positive voltage
V_MIN	Float	0.0	Minimum field voltage

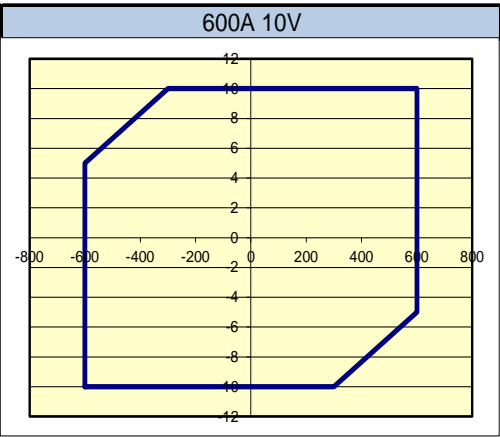
LIMITS Parameter	Type or ENUM	Default	Comments
V_NEG	Float	-1.0E+9	Maximum negative voltage
V_RATE	Float	1.0E+9	Maximum absolute rate of change of voltage.
V_ACCELERATION	Float	1.0E+9	Maximum absolute voltage acceleration. If set to zero the acceleration checks are disabled.
V_ERR_WARNING	Float	0.0	Libreg does not support the voltage regulation loop (though it could in required). However, if the voltage is measured, the error of the external voltage regulation loop can be calculated and monitored. This parameter defines the warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than 3 times the tracking delay. If the value is zero the fault check is disabled.
V_ERR_FAULT	Float		Field regulation error warning threshold. The warning flag will be set if the regulation error exceeds the threshold for more than
V_QUADRANTS41	Float	0.0,0.0	Together with I_QUADRANTS41 this allows the voltage to be limited as a function of the current (see below).

The I_QUADRANTS41 and V_QUADRANTS41 parameters are used to protect 2 or 4-quadrant converters that cannot send energy back onto the mains. This means that when the voltage has the opposite sign to the current the magnet energy is dissipated in the output stage of the converter. In some cases, the rating of the cooling of the output stage will be less than required to handle the extreme cases of maximum positive current and maximum negative voltage or vice versa.

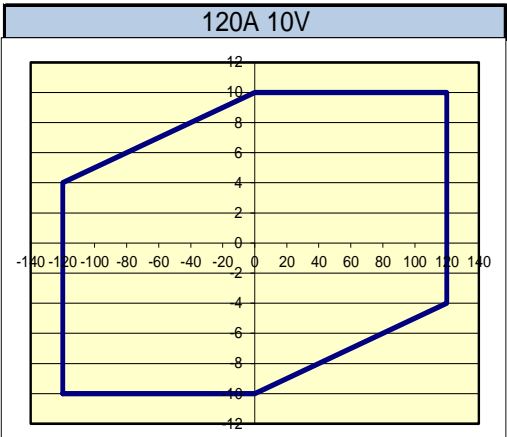
Converter quadrants are numbered as shown below:



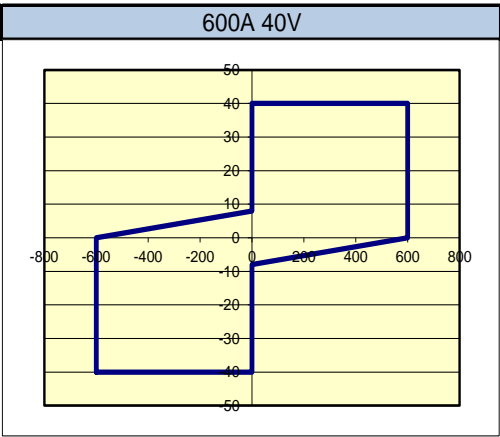
The I_QUADRANTS41 and V_QUADRANTS41 parameters define the current dependent voltage limits for quadrants 4 and 1. The software automatically rotates these limits and applies them to quadrants 2 and 3. Examples from the LHC 4-quadrant power converters are shown in the figures below.



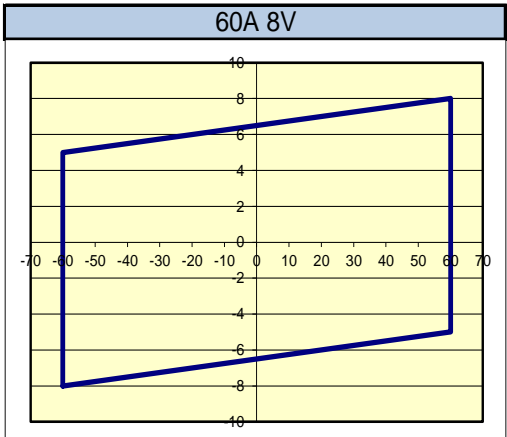
I_QUADRANT41 -600, -300
V_QUADRANT41 5, 10



I_QUADRANT41 -120 0
V_QUADRANT41 4, 10



I_QUADRANT41 -600, 0
V_QUADRANT41 0, 8



I_QUADRANT41 -60, 60
V_QUADRANT41 5, 8

2.2.4 Voltage source parameters file (-s vs_file)

The voltage source parameters file is optional when GLOBAL.SIM_LOAD is ENABLED. If it is not supplied then the default values are used for all parameters which corresponds to a voltage source with infinite bandwidth.

VS Parameter	Type or ENUM	Default	Comments
V_REF_DELAY	Float	1.0	Delay between the start of an iteration in which V_REF is calculated and the time that it enters the simulation of the voltage source. This models the delay that might be due to a DAC settling, or a digital link between a current controller and the voltage source electronics. <i>This must be at least equal to GLOBAL.ITER_PERIOD.</i>
V_MEAS_DELAY	Float	0.0	Delay in the measurement of the voltage. This models the analogue anti-alias filter and any digital filtering as a simple delay.
TRACK_DELAY	Float	GLOBAL.ITER_PERIOD	Anticipated delay between the setting of the voltage reference and the moment when the measured voltage should equal the reference. This should equal the V_REF_DELAY + V_MEAS_DELAY + voltage source response modelled as a simple delay. This is used to calculate the error in the response of the voltage source.
NUM	Float	1.0,0,0,0	Voltage source transfer function numerator
DEN	Float	1.0,0,0,0	Voltage source transfer function denominator

The voltage source model is implemented as a third order digital filter using the equation:

$$V_{LOAD_0} = \frac{\sum_{i=0}^{i=3} V_{REF_i} \cdot NUM[i] - \sum_{i=1}^{i=3} V_{LOAD_i} \cdot DEN[i]}{V_{REF_0}}$$

2.2.5 Regulation parameters file (-r reg_file)

The regulation parameters file must be provided if GLOBAL.UNITS equals AMPS or GAUSS.

REG Parameter	Type or ENUM	Default	Comments
PERIOD_ITERS	<i>Unsigned</i>	<i>None</i>	Regulation period in units of GLOBAL.ITER_PERIOD. This is an integer because the regulation can only run on iteration boundaries.
TRACK_DELAY	<i>Float</i>	0.0	Anticipated delay between the setting of the current or field reference and the moment when the measured current or field should equal the reference. This is used to calculate the error in the response of the regulation loop. The track delay will depend upon the type of regulator being used. If the deadbeat PII algorithm is used then the track delay should be 1 regulation period.
CLBW	<i>Float</i>	0.0	Closed loop bandwidth for real pole.
CLBW2	<i>Float</i>	0.0	Close loop bandwidth for conjugate poles.
Z	<i>Float</i>	0.0	Damping factor.
PURE_DELAY	<i>Float</i>	0.0	Pure delay in the regulation loop. This should include VS.V_REF_DELAY + measurement delay + voltage source response modelled as a simple delay. For stability, this cannot be more than 40% of the regulation period. This drastically limits the acceptable measurement delay and means that the anti-aliasing filter must be carefully judged.
R	<i>Float</i>	<i>None</i>	Array of coefficients for measurement history
S	<i>Float</i>	<i>None</i>	Array of coefficients for actuation history
T	<i>Float</i>	<i>None</i>	Array of coefficients for reference history
OL_TIME	<i>Float</i>	0.0	Open loop time. Set this to the time when the simulation should open the regulation loop for the time defined in OL_DURATION. This allows the testing of the libreg function that can open and close the regulation loop.
OL_DURATION	<i>Float</i>	0.0	Open loop duration used with OL_TIME.

When GLOBAL.UNITS equals AMPS the measurement delay is LOAD.I_MEAS_DELAY. When GLOBAL.UNITS equals GAUSS the measurement delay is always zero because the delay in the field measurement is negligible.

Libreg uses the RST algorithm which says:

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

Where i=0 corresponds to the current sample, i=1 is the previous sample and so on. This notation was proposed by Landau, however in many text books the R and S polynomials are exchanged.

For a magnet circuit the reference and measurement can either be of the circuit current or the magnetic field (as indicated in GLOCAL.UNITS). The actuation defines the circuit voltage which the voltage source must try to follow.

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:

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

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:

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

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 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.

Libreg allows you to specify the RST coefficients directly in the file, for example:

```
# CCTEST regulation file: B POPS

PERIOD_ITERS 3
TRACK_DELAY 7.91232E-3
R 8.93192E+01,-7.40453E+01,-9.05475E-01,-1.68384E-01
S 1.00000E+00,-4.50999E-01,-1.57806E-01,-3.74426E-01,-1.67695E-02
T 1.30115E+02,-1.92672E+02,7.67562E+01

# EOF
```

In this case, an expert used MATLAB to model the voltage source and the load and then calculated the RST coefficients that implement a PI regulator that runs at 333.333 Hz (i.e. every third iteration of a 1 kHz controller) and which has a track delay of 7.91232 ms.

In the following conditions are met then it is possible to use the algorithm included in **libreg** that can calculate the RST coefficients:

- The required bandwidth of the current or field regulation is much less than the bandwidth of the voltage source
- The bandwidth of the reference is less than the bandwidth of the regulation

The algorithm implements a deadbeat PII regulator by ignoring the voltage response and modeling the load as defined in section 2.2.2. Being a deadbeat controller, this will have a track delay of exactly 1 regulation period, for constant rates of change of the reference. Note that the regulation bandwidth defines the bandwidth for the rejection of perturbations, not the bandwidth for tracking.

Here is an example based on a voltage source with a second order response at about 200 Hz that is modeled as a simple delay of 2.8 ms.

```
# CCTEST regulation file: RPTG D1

PERIOD_ITERS      20
TRACK_DELAY       20E-3
CLBW              20
CLBW2             20
Z                 0.5
PURE_DELAY        5.1E-3

# EOF
```

In this example, GLOBAL.ITER_PERIOD is 1 ms so the regulation will run every 20 ms and the track delay is also 20 ms (because it uses the deadbeat controller). The bandwidth is 20 Hz and the damping factor is 0.5 (this is the best value). The PURE_DELAY parameter is critical and is 5.1 ms = 1.0 ms (V_REF_DELAY) + 1.3 ms (I_MEAS_DELAY) + 2.8 ms (Voltage source response modeled as a simple delay). This is 25.5% of 20 ms which is under the 40% limit needed for stability.

2.3 FUNCTION DATA FILES

2.3.1 start

Most 1 and 2-quadrant power converters have a minimum operating current below which the voltage response is non-linear. This means that regulation can only start once the current (or field) has risen above the threshold for linear operation. The start function in **fgest** is provided as an example of how an openloop feedforward voltage can be used to safely start a 1 or 2-quadrant converter.

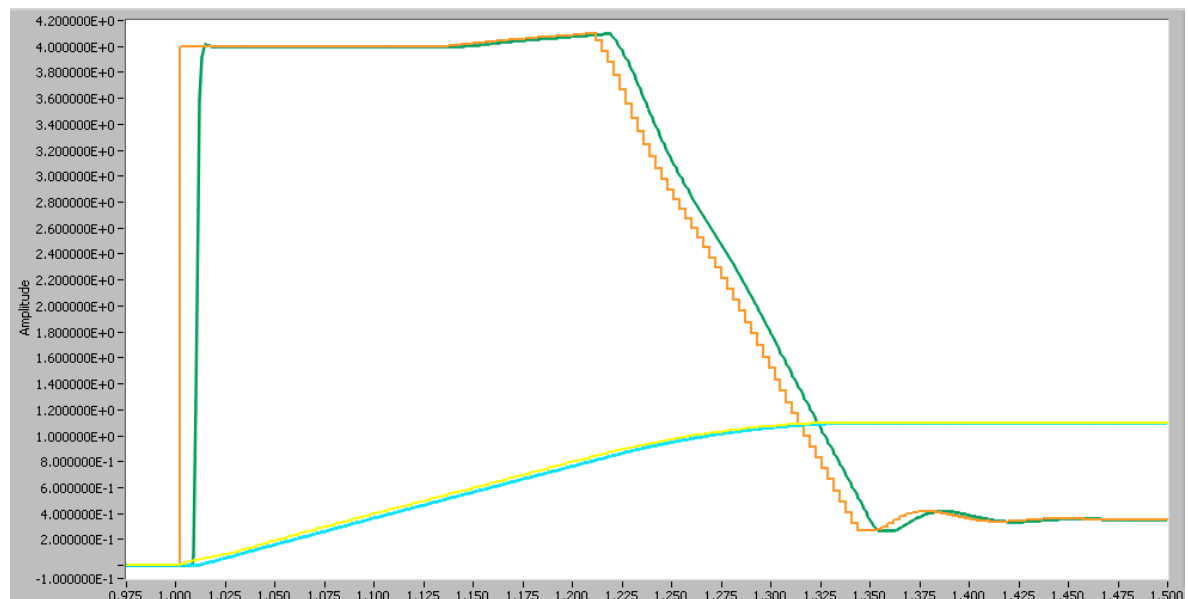
Note that unlike the rest of the reference functions, START can only be specified if GLOBAL.UNITS is set to AMPS or GAUSS *and* GLOBAL.FG_LIMITS is ENABLED.

Parameter	Type or ENUM	Default	Comments
FEEDFORWARD_V_REF	Float	None	Feedforward voltage reference to start openloop ramp up of current and field.
CLOSELOOP_LEVEL	Float	None	Current or field measurement threshold for closing the loop. This must be in the range 20% - 80% of FINAL_REF.
FINAL_REF	Float	None	Reference for plateau at the end of the start function
ACCELERATION	Float	None	This will be used as the deceleration for the parabolic arrival of the reference to the FINAL_REF plateau.

Initially the RST algorithm runs backwards to computer the reference from the actuation (FEEDFORWARD_V_REF) and the measurement. Then once the measurement rises above the CLOSELOOP_LEVEL the RST algorithm is switched to run forwards, calculating the actuation from the reference and the measurement.

The reference function generator is then re-initialised using the PLEP function to continue the same rate of change until it needs to decelerate parabolically to arrive at the FINAL_REF plateau.

Here is an example in which FEEDFORWARD_V_REF is 4, CLOSELOOP_LEVEL is 0.5, FINAL_REF is 1.1 and ACCELERATION is 30:



2.3.2 plep

To be written.

2.3.3 pppl

To be written.

2.3.4 table

To be written.

2.3.5 trim

To be written.

2.3.6 test

To be written.

2.4 OUTPUT

2.4.1 standard error

2.4.2 standard output

2.4.3 output_format

cctest supports four different output formats selectable using the `-o` command line option or the `GLOBAL.OUTPUT_FORMAT` parameter. Three formats are basically CSV files with slightly different headers while FLOT generates an HTML file that exploits the FLOT javascript graphic package to provide an interactive chart in a browser.

2.4.3.1 STANDARD – CSV for EXCEL, MATLAB etc...

STANDARD produces simple CSV files with a header line, making it easy to use with Excel and MATLAB.

2.4.3.2 FGCSKY – CSV for FGCspy program

FGCSKY is the default format as FGCspy is currently the standard data viewer used by the TE-EPC group at CERN.

2.4.3.3 LVDV – CSV for the CERN Labview Dataviewer program

A new and more advanced Labview based data viewer (LVDV) is in development and will take over from FGCspy in 2014.

2.4.3.4 FLOT – HTML using the FLOT javascript graphing library (based on jQuery)

3. PROGRAM STRUCTURE

3.1 SOURCE FILES

3.1.1 cctest.c

3.1.2 ccpars.c

3.1.3 ccref.c

3.1.4 ccrun.c

3.1.5 ccsgs.c