IMPLEMENTATION OF A NEW RAMP COMPUTATION SCHEME FOR THE MAGNET POWER SUPPLIES AT ELSA

D. Proft* and W. Hillert, ELSA, Bonn, Germany

Abstract

At the ELSA electron stretcher facility new power supply control units have been commissioned. These require a new software interface for set-point calculation based on the accelerator and timing model.

Goal of the new scheme is a strict separation of the bidirectional ramp computation into an accelerator model dependent, a magnet dependent and a power supply dependent part. This introduces possible calibration/correction factors on each layer, thus allowing easy component replacement of the power supplies, the control units or even the magnets without the need for recalibration of the whole chain.

In this contribution we will provide insights into the implementation of the new modeling scheme.

ELSA

The electron stretcher facility ELSA (see Fig. 1) is a three stage accelerator consisting of two linear accelerators and a fast ramping booster synchrotron which are utilized as injector for the main storage ring. In the storage ring electrons can be accumulated by multiple injections of the booster, post-accelerated to a maximum energy of 3.2 GeV and then be extracted to one of the hadron physics experiments. The typical cycle length is 6 s consisting of 400 ms injection time and 2×300 ms acceleration/deceleration time (see Fig. 2). The extraction takes place during the remaining 5 s, using a third integer resonance, resulting in a constant electron current of typically up to 2 nA at the experiments.

The new timing system [1] allows for flexible acceleration cycles with variable injection time[1]. As well the extraction time can either be statically configured or lasts until the internally stored current falls below a given threshold resulting in a variable extraction time.

To enable this mode of operation an upgrade of the existing in-house developed interface to the power supplies as well as an upgrade of their software components being responsible for the computation of the set point curves for the power supplies is required.

Hardware

During the last year new micro controller based power supply control units have been developed [2] and tested successfully. These control units provide slow control capabilities and an analog set point voltage for the connected power supply. In total there are six power supplies for main magnets and 54 in house developed power supplies for steerer magnets. Four of them can be served by one control unit.

* proft@physik.uni-bonn.de

[1] currently only a static, but configurable amount of injection shots is possible

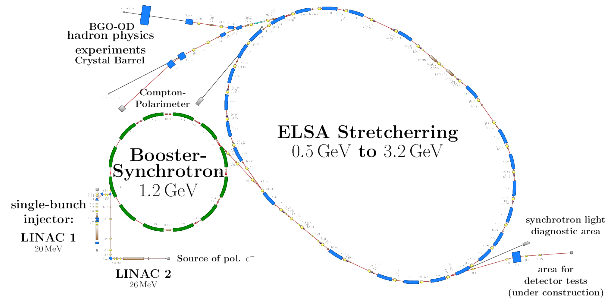

Figure 1: Sketch of the electron stretcher facility ELSA.

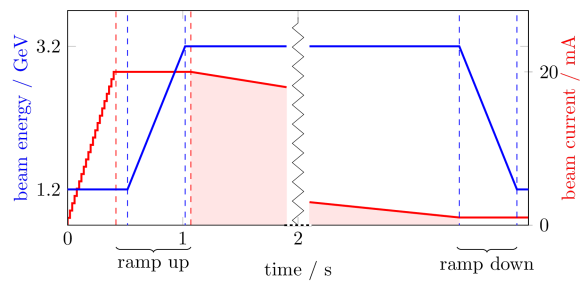

Figure 2: ELSA operation cycle consisting of accumulation, ramp up, extraction and ramp down phase.

The design of the steerer power supplies follows a modular principle [3]. The micro controller board is connected to a controller board with set point and actual value comparison which itself is connected to an H-bridge and the magnet[2]. Calibrations have to be done for each module including the instrument shunt of the bridge, offset and slope of the individual DAC, as well as saturation effects of the magnet.

COMPUTATION SCHEME

Goal of the new computation scheme is to derive steering curves for the power supplies based on the operation cycle of the accelerator shown in Fig. 2. As input parameters for the model a set of accelerator physics parameters is used:

- the energy E,
- the desired orbit at BPM locations x and z,
- the tune Q_x and Q_z,
- and the chromaticity ξ_x and ξ_z.

The operation cycle is divided into two parts: one for ramping the energy to the desired extraction energy, and one for ramping down to injection energy. On each part the energy is changed linearly, so the energy as a function of time $E(t)$ can be expressed in terms of a piecewise linear function which is called an *energy ramp* $\mathcal{R}_E(t)$. Accordingly, the

[2] either a dedicated vertical steerer magnet, or the main dipole magnet's correction coils for horizontal correction

ISBN 978-3-95450-147-2

evolution over time of the other parameters $\chi \in \{x, Q, \xi\}$ can also be expressed by *ramps*: $\mathcal{R}_\chi(t)$.

The parameters χ have to be transformed to the desired set point value for the power supply as shown in Fig. 3. On the first layer this is done by utilizing the accelerator model: The desired orbit can be expressed in terms of corrector kicks α_i of each steerer magnet, the tune by the quadrupole strengths k_f and k_d, and the chromaticity ξ by the sextupole strengths m_f and m_d. In case of the sextupoles, a third parameter m_x is introduced to allow for driving the third integer resonance during the extraction phase.

In the following, a chain of transformations is needed to calculate an integer value U for the digital analog converter installed on the microcontroller board:

- calculate the magnetic field gradient $B^{(n)}$ ($B^n = \frac{\partial^n B}{\partial x^n}$) with respect to the appropriate energy E from χ
- ↳ calculate the current I_s using the calibration curve for the installed magnet (including saturation effects)
- ↳ correct the current I_s and derive I_i by taking into account the calibration of the power supply's instrument shunt
- ↳ calculate the DAC value U using the DAC calibration (linearity and offsets)

The complete transformation chain is illustrated in Fig. 3.

In analogy the computation can be carried out using *ramps*. The transformation function $\mathcal{T}_i$, in detail, is achieved by transforming every single support point of the ramp $\mathcal{R}_i$ using the same chain (i.e. $\mathcal{T}_k : \mathcal{R}_k(t, E) \mapsto \mathcal{R}_{B'}(t)$).

In total the U-ramp $\mathcal{R}_\mathrm{U}(t)$ is calculated i.e. from the desired tune by concatenating the $\mathcal{T}_i$:

$$\mathcal{R}_\mathrm{U}(t) = \mathcal{T}_{I_i}\left(\mathcal{T}_{I_s}\left(\mathcal{T}_{B'}\left(\mathcal{T}_k\left(\mathcal{R}_k(t, E)\right)\right)\right)\right) \quad .$$

IMPLEMENTATION

The previously mentioned ramp computation scheme is implemented in *C++11*. The key component of the developed library is a *LinearRamp* class containing the piecewise linear function $\mathcal{R}_i$. It is made up of a vector of time-value pairs representing the support points of the function. It is specialized using *C++ template programming* techniques to support and distinguish the ramps of different physical quantities. The set of available quantities is pre-defined[3], but easily extensible. Accessor functions to retrieve interpolated values at any given time t and methods for manipulation (inserting, deleting and replacing time-value pairs) are provided by the implementation as well.

The conjunction of a power supply and it's connected magnet is represented in the software as *device*. Transformations of single values or entire *LinearRamps* to the subsequent one (see Fig. 3) are performed having regard to a given *device*'s configuration and calibration factors (called *device configuration*). Calibrations and configurations are imported from

[3] currently supported quantities are: energy, tunes, corrector kicks, chromaticity, magnet strengths, magnet field strength, coil currents and DAC values.

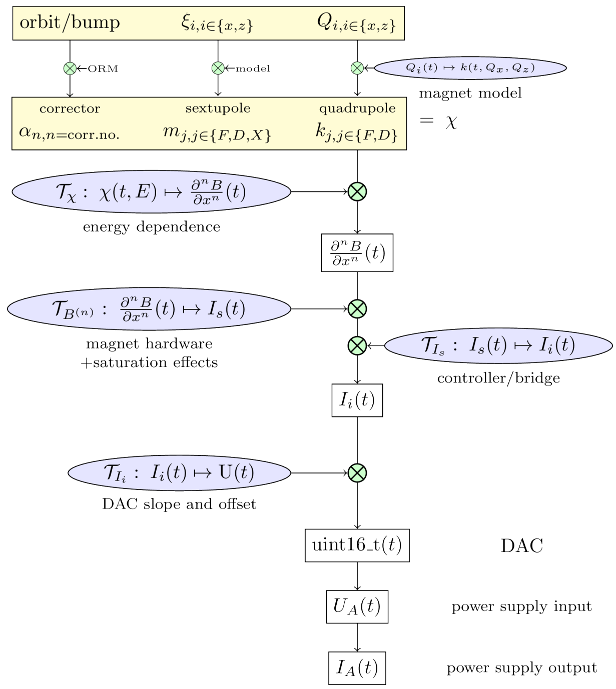

Figure 3: Sketch of the new ramp computation scheme.

XML files during runtime (see Fig. 5). Each file can be referred to by an arbitrary number of devices. Beside common polynomials up to the degree of five, special transformations using, for example, lookup tables can be used. In either case only the coefficients need to be given in the file.

In order to compensate magnet dependent saturation effects, either these lookup tables or fitted curves can be used. The lookup tables are generated by measuring pairs of $B(I)$ of the magnet and the corresponding current I. For intermediate points a linear interpolation between the measured values is used. E.g. for the quadrupole magnets the best fit was achieved by using a linear approximation below $I = 500$ A and an additional square root contribution afterwards.

In case of non-linear transformations, the linear approximation with the given set of supporting points is automatically checked for deviations (as visualized in Fig. 4). The threshold is one significant bit of the used 16 bit DAC used in the units, which is the maximum achievable accuracy anyway. If the threshold is exceeded, new intermediate supporting points are automatically inserted.

For better documentation each *XML* file includes a comment-section in which LaTeXmarkup language commands can be used to make the files "self-documenting".

Reverse Computation

Each power supply control unit is supplied with an analog digital converter for each of the four channels to read out actual values. The raw ADC data needs to be transformed back to χ's in order to display them in the control system. Due to the variety of different transformation functions, it is rather complicated to give an inverse function of the whole transformation chain.

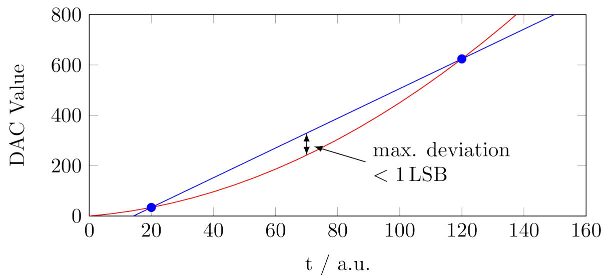

Figure 4: Check for deviation threshold on non-linear transformations. New support points are automatically inserted when necessary.

To circumvent this difficulty, root finding algorithms can be used: Obtaining the original value χ_0 for a given ADC value U_{ADC} is equivalent to solving the equation

$$0 \overset{!}{=} \mathcal{T}^*_{I_i}\left(\mathcal{T}^*_{I_s}\left(\mathcal{T}^*_{B'}\left(\mathcal{T}^*_{\chi}\left(\chi_0\right)\right)\right)\right) - U_{\mathrm{ADC}}$$

for χ_0. Here the *boost* C++ library's implementation of *Algorithm 748* [4] is used.

After finding the original values for all digitized values for one accelerator cycle, the "actual value ramp" is reconstructed and can be displayed in the control system.

Device Configuration

Each microcontroller can be configured with parameters dependent on the connected power supply. This configuration needs to be performed during it's startup sequence and consists of a set of default values:

- the output voltage at device reset/startup in terms of ADC value
- the maximum ADC value slew rate per sampling point as fixed point value (required for machine protection, due to limited capabilities of the main magnets power supplies).
- the sample times for the ADC and the DAC in terms of multiples of 10 µs.

The communication is carried out via a custom protocol on top of *TCP/IP*. It is also used for ramp data upload and status messages. The configuration parameters as well as the network address are also included in the device's configuration XML.

SUMMARY

First in field tests with one magnet family being operated by the new power supply control units as well as the new ramp computation scheme have been successfully carried out. Minor improvements to the micro controller boards have been implemented which are now ready for series production and replacement of the old system.

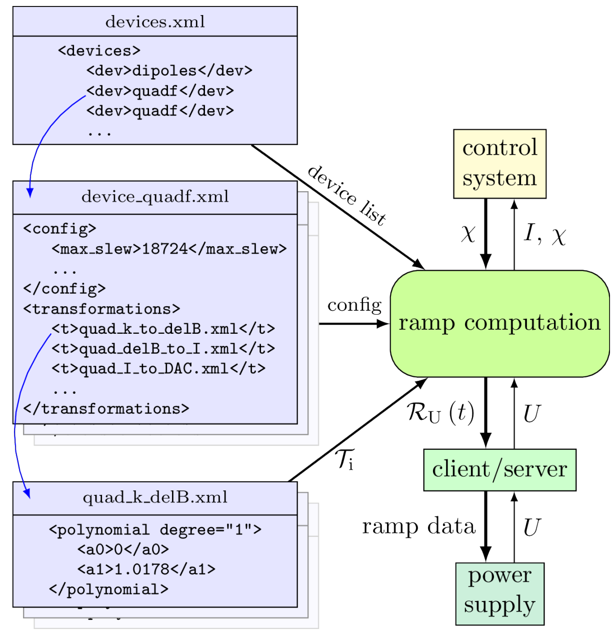

Figure 5: Ramp computation scheme: Settings from control system (yellow) are used to compute ramps $\mathcal{R}_{\mathrm{U}}(t)$ using XML configuration/calibration files (blue). These are send to a server via TCP which communicates with power supplies.

The new scheme has been verified to operate smoothly with the existing infrastructure and behaves like the old algorithms used for ramp computation. In contrast to the old system, the new one offers a much more flexible configuration interface and support for the new power supply control units.

Next, calibration factors for all devices will be determined and the power supply control units will be commissioned.

REFERENCES

[1] D. Proft et al., "A new FPGA based timing system at ELSA", in *Proc.*, IPAC'15, Richmond, VA, USA, May 2015, paper MOPHA012, pp 802-804.

[2] D. Proft et al., "Development of New Microcontroller Based Power Supply Control Units at ELSA", in *Proc.*, IPAC'15, Richmond, VA, USA, May 2015, paper WEPMA014, pp 2770-2779.

[3] A. Dieckmann et al., "Fast Ramping Arbitrary Waveform Power Supplies for Correction Coils in a Circular Electron Accelerator", in *Proc.*, IPAC'12, New Orleans, LA, USA, May 2012, paper THPPD050, pp 3623-3625.

[4] G. E. Alefeld, F. A. Potra and Yixun Shi, "Algorithm 748: Enclosing Zeros of Continuous Functions", in *ACM Transactions on Mathematical Software (TOMS)*, vol. 21, issue 3, pp 327-344, Sept. 1995.