A DIGITAL SYSTEM FOR LONGITUDINAL EMITTANCE BLOW-UP IN THE LHC

M. Jaussi*, M. E. Angoletta, P. Baudrenghien, A. Butterworth,
J. Sanchez-Quesada, E. Shaposhnikova, J. Tuckmantel,
CERN, Geneva, Switzerland

Abstract

In order to preserve beam stability with nominal bunch intensity in the LHC, longitudinal emittance blow-up is performed during the energy ramp by injecting phase noise in the main accelerating cavities. The noise spectrum spans a small frequency band around the synchrotron frequency. It is generated continuously in software and streamed digitally into the Digital Signal Processor (DSP) of the Beam Control system where it is added to the pick-up signal of the beam phase loop, resulting in a phase modulation of the accelerating RF. In order to achieve reproducible results, a feedback system, using as input the measured bunch lengths averaged over each ring, controls the strength of the excitation, allowing the operator to simply set a target bunch length. The spectrum of the noise is adjusted to excite the core of the bunch only, extending to the desired bunch length. As it must follow the evolution of the synchrotron frequency through the ramp, it is automatically calculated by the LHC settings management software from the momentum ramp and RF voltage. The system is routinely used in LHC operation since June 2010. We present here the details of the implementation in software, FPGA firmware and DSP code, as well as some results with beam.

INTRODUCTION

Effective control of the bunch length in the LHC is essential to avoid loss of Landau damping leading to longitudinal instability during the energy ramp [1]. A longitudinal emittance blow-up system has been developed, inspired by the system that has been used in the SPS for several years [2]. The beam is excited with RF phase noise acting on the fundamental RF system (400.8 MHz). The frequency spectrum of the noise is tailored to the synchrotron frequency spread of the particles in the bunch so as to selectively excite a chosen portion of the particles centered around the core of the bunch [3]. The frequency of a single-particle synchrotron oscillation depends on the peak amplitude of its trajectory ϕ_{pk}

$$\Omega_s(\phi_{pk}) \approx \Omega_{s0}\left[1-\left(\frac{\phi_{pk}}{4}\right)^2\right] \qquad (1)$$

with Ω_{s0} the synchrotron frequency of the zero-amplitude oscillation. For $\phi_{pk} = \frac{\pi}{2}$ the synchrotron frequency is $\sim \frac{6}{7}\Omega_{s0}$. A rectangular frequency spectrum extending from $\frac{6}{7}\Omega_{s0}$ to Ω_{s0} will excite particles in a 1.25 ns window of the 400 MHz bucket. We extend the spectrum to $1.1\Omega_{s0}$ to guarantee that we do not miss the core. The noise is injected into the RF via the beam phase loop, whose main purpose is to keep the accelerating RF in phase with the beam [4]. The noise signal is added digitally into the phase loop error signal, "shaking" the phase of the accelerating cavity voltage with respect to the beam to produce the desired excitation.

* michael.jaussi@cern.ch

REQUIREMENTS

The system needs to generate band-limited noise that must follow the synchrotron frequency and with an amplitude that eventually achieves the desired bunch length. It must be able to run for a long period of time, since the ramping process in LHC lasts for 11 minutes (it was 40 minutes in 2010). The system must allow great flexibility in the noise generated as it was not well known at the commissioning of the blow-up process what would be the most effective settings for the beam, and how to achieve the targeted bunch length.

IMPLEMENTATION

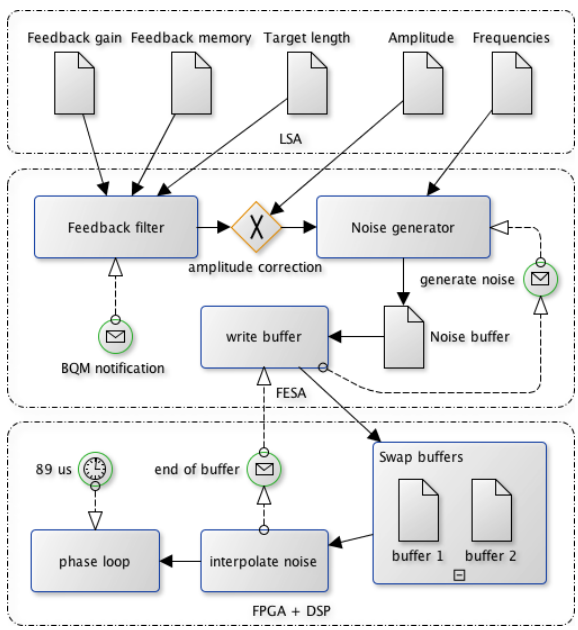

Figure 1: Block diagram showing the 3 layers.

The system is subdivided into 3 layers (see Fig. 1). From the bottom up, we have the hardware layer, implemented

in a VME board, containing a Field Programmable Gate Array (FPGA) and an Analog Devices TigerSHARC DSP, responsible for the digital processing of the beam control loops. In the CPU of the VME crate we have the device driver and the front-end software, written in C++ under CERN's Front-End Software Architecture (FESA) [5] framework. Finally we have the control system application software based on LSA [6], and written in Java, which handles settings management for the whole accelerator.

FPGA AND DSP

The FPGA and DSP are in the phase loop module of the LHC beam control. The FPGA implements a double-buffering scheme to allow continuous operation for the DSP. It provides 2 buffers of 4096 float values which are alternately filled via the VME bus by the front-end software. The FPGA presents the data one point at a time to the DSP which injects it into the phase error correction. Once a buffer is empty, the FPGA swaps the buffers and sends an interrupt to the CPU asking for the empty buffer to be filled.

The start of the noise generation is triggered from the software by writing a bit in the appropriate register. The processing rate of the phase loop is once per machine turn, so the DSP has to calculate a new value about once every 89 μs. With a synchrotron frequency varying between 60 Hz and 28 Hz during the acceleration ramp, the spectrum of the excitation data does never extend beyond 70 Hz. The DSP code specifies the rate at which the buffer should be read and intermediate samples are interpolated to achieve the 89 μs period. In the buffer, the noise samples are at the 2 kHz rate (one value every 500 μs.) so with a buffer size of 4096, we need to fill a new buffer every 2 seconds. The FPGA has a few registers for buffer handling. This allows us to set the interpolation rate, specify the buffer size (up to 4096), enable the interrupt and flag a buffer as ready. A flag is also raised when a buffer is empty and a new buffer is required.

FESA CLASS

To control the FPGA and DSP, the FESA class accepts settings from the control system and then generates the noise. The settings are :

- 2 functions for programming the upper and lower frequency during the ramp.
- 1 function for programming the maximum amplitude allowed.
- 2 functions programming for the feedback system.
- 1 scalar or function for programming the target bunch length.
- A way to start the system, either manually or through a trigger.

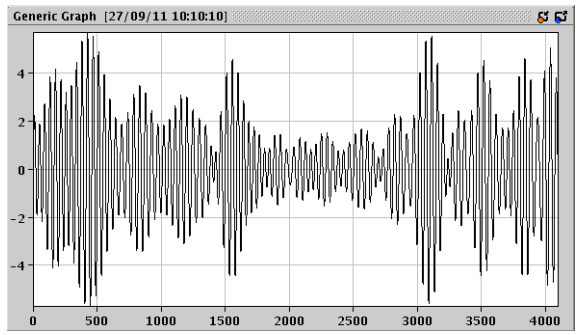

Figure 2: Noise buffer calculated by the FESA class.

As the noise generated cannot fit entirely in memory, it is generated in partial records (example Fig. 2), and we keep track of the position in time within the functions by counting the number of noise samples generated. This gives us a time precision equal to 500 μs with the current settings, which is more than good enough as the frequencies variy slowly (see Fig. 4). In order to react quickly to the interrupt sent by the FPGA when a buffer is empty, one buffer of noise is always generated in advance. When the interrupt is sent, it triggers a FESA real-time task which writes the noise data into the buffer and then acknowledges it. This task then fires a user-event which triggers another real-time task with a lower priority in charge of generating the next noise buffer. If any of the settings was updated, either by the feedback or by human intervention (such as changing a function), then the new values are taken into account at this point of time. If for some reason, the noise cannot be generated before the next interrupt, an alarm is raised and the system set the noise output to zero.

Noise Generation Algorithm

To allow the generation of precisely band-limited noise, a dedicated noise generation algorithm was developed [7]. It uses sine and cosine generators which are initialized to a random phase, scaled with a random amplitude and that generate noise in a normalized frequency range, with the generators frequencies equally spaced between 0 and 1. The noise is then translated numerically to the correct frequency range, defined by two parameters, named f_{up} and f_{low} in [7], which reflect the upper and lower bound of the range. To allow different spectral shapes, there is a weighting of all generators amplitudes, allowing for the generation of any spectrum, with a resolution depending on the number of generators. Another parameter specifies the RMS amplitude of the noise. To modulate the amplitude, each value is simply multiplied by the desired scaling.

Feedback

The feedback system is divided into two blocks. The first one is the measurement and the second one is the correction of the excitation by continuously adjusting the amplitude of the phase noise. It must be noted that, because of the next

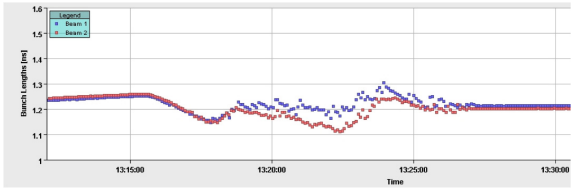

Figure 3: Four-sigma bunch length evolution during a ramp with the blow-up system set for a 1.2 ns target. Start of ramp at ∼13:16, end at ∼13:27

buffer already generated in memory, the change induced by the feedback is only taken into account for the next buffer, which might be used only after 2 to 4 seconds. This could be reduced by using smaller buffer size, but then a new noise buffer must be provided more often, inducing a bit more overhead on the CPU as more interrupts need to be handled.

Bunch length measurement: In order to see the result of the noise injected, we need to measure the length of each bunch (or at least have a mean length over all bunches). To do so, we use an existing diagnostic system called the Beam Quality Measurement (BQM) [8], which uses high-speed digitizers to acquire the signal from a wideband longitudinal beam pickup over a complete turn of the machine and then calculates from the signal a length for each bunch (see Fig. 3). The calculation extracts the Full Width at Half Maximum (FWHM) for each bunch, and estimates the 4σ equivalent length assuming a Gaussian profile. As this system is also implemented using a FESA class, it allows other systems to subscribe to the measurement, thus allowing our system to be notified each time a new measurement is available. The measurement rate of this system is about 5 seconds at the time of writing, which is rather slow, and is in fact one of the major limitations on performance of the blow-up. A upgrade is underway to improve the data rate of this system.

Filter: The feedback system is a simple low pass filter, with two variable parameters, defined as functions of time to allow fine tuning of the system. One parameter is the gain, which is increased during the ramp to compensate for the reduced sensitivity to RF noise with energy. The second one is the time constant of the filter, intended to smooth the bunch length measurements.

CONTROL SYSTEM

The control system provides a way to store, generate and send the settings to the FESA class. Most of the settings are defined as functions which vary along the ramp. Some settings are declared as being dependent on other settings. This is the case for the frequencies as they are calculated as a band around the synchrotron frequency (Fig. 4). This frequency is in turn dependent on the momentum and the RF voltage. The target bunch length (Fig. 5) is typically set

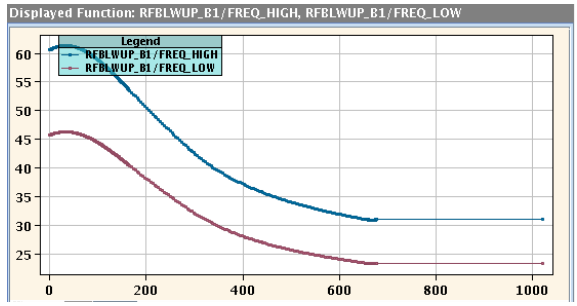

Figure 4: Upper and lower frequency bound functions in LSA. Horizontal axis: time in ramp (sec), vertical axis: frequency (Hz).

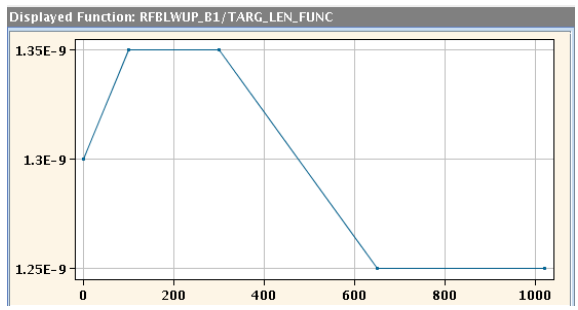

Figure 5: Target bunch length function in LSA.

to a higher value at the beginning of the ramp to mitigate the shrinking which occurs due to the increase of the RF voltage. The feedback gain (Fig. 6) is increased during the ramp as the beam becomes less responsive to the excitation as the energy increases. The control system is in charge of updating the frequency values if any of the dependent parameters changes. It is also responsible for arming the system before the ramp, allowing it to start on reception of an event from the accelerator timing system. The arming process is needed as the event used as a trigger is far from unique and could happen also when the machine stays at injection energy. Instead of having a dedicated timing, the sequencer software that prepares the entire ramp process takes care of arming the trigger. The control system also monitors any alarm that could be raised by the FESA class.

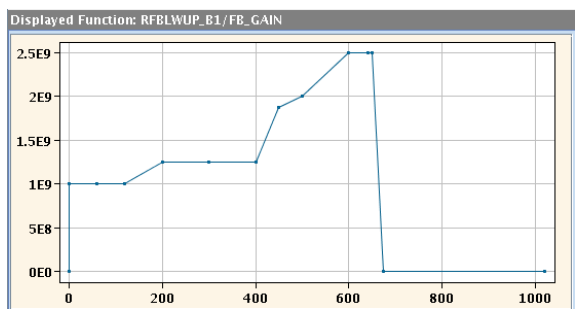

Figure 6: Gain function in LS.

RESULTS AND CONCLUSIONS

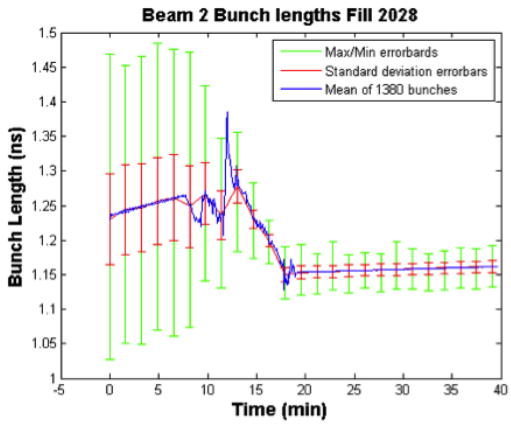

Figure 7: Statistics on bunch length (mean, min, max and standard deviation error bars) measured by the BQM during the ramp.

The first results without feedback, for a few bunches, were encouraging. It demonstrated that the system was working, but also that a feedback system was required to ensure reproducibility of the results. Once the feedback system was in place, the results were seen to be highly reproducible, and the system has been in use operationally since June 2010. A good feature of the blow-up is that it reduces the spread of bunch lengths. As can be seen in Fig. 7, the spread of bunch length at injection is large, with a standard deviation of about 60 ps, which after the ramp is reduced to around 15 ps. We often observe periods of very fast change in the measured bunch length, and this limits the final accuracy of bunch length control. We believe that these transients come from a change in bunch profile that has a significant impact on the FWHM measurement without much increase in longitudinal emittance. This has an impact on the stability of the feedback and sometimes results in missing the target bunch length at the end of the ramp. To mitigate this type of problem, the BQM system is being updated to provide measurements at a higher rate, which will allow higher feedback gain settings without becoming unstable.

REFERENCES

[1] E. Shaposhnikova *et al.*, "Loss of Landau damping in the LHC", IPAC'11, San Sebastian, Spain, 4–9 September 2011,

[2] J. Tuckmantel *et al.*, "Study of Controlled Longitudinal Emittance Blow-up for High Intensity LHC beams in the CERN SPS", EPAC08, Genoa, Italy, June 2008,

[3] P. Baudrenghien *et al.*, "Longitudinal emittance blow-up in the LHC", IPAC'11, San Sebastian, Spain, 4–9 September 2011,

[4] P. Baudrenghien *et al.*, "The LHC Low Level RF", EPAC06, Edinburgh, UK, June 2006,

[5] M. Arruat *et al.*, "Front-End Software Architecture", ICALEPCS07, Knoxville, USA, October 2007,

[6] G. Kruk *et al.*, "LHC Software Architecture [LSA]: Evolution toward LHC Beam Commissioning", ICALEPCS07, Knoxville, USA, October 2007,

[7] J. Tuckmantel, "Digital Generation of Noise-Signals with Arbitrary Constant or Time-Varying Spectra", LHC Project Report 1055, February 2008,

[8] G. Papotti *et al.*, "Longitudinal Beam Measurements at the LHC: the LHC Beam Quality Monitor", IPAC'11, San Sebastian, Spain, 4–9 September 2011,