MODEL-BASED ANALYSIS OF DIGITAL SIGNAL PROCESSING BLOCKS IN A BEAM PHASE CONTROL SYSTEM*

C. Spies†, H. Klingbeil, M. Glesner, TU Darmstadt, Darmstadt, Germany

Abstract

A beam phase control system comprises digital phase detectors and band pass filters to detect coherent longitudinal dipole and higher order bunch oscillations. These digital signal processing functions can be implemented in several ways, e. g. in software or on a programmable logic device. In this paper, we consider different possible implementations and compare them in terms of their real-time performance and their system resource consumption. For the phase detectors, a software implementation is compared against different hardware implementations. For the band pass filters, different possible architectures are considered.

BEAM PHASE CONTROL

Longitudinal oscillations in synchrotrons may occur due to an initial energy mismatch or may be excited by discontinuities in the input ramps, by wake fields, etc. These oscillations can be characterized by two mode numbers m and n [1]. In order to eliminate undesired dipole oscillations ($m = 1$, $n = 0$), a beam phase control (BPC) system has been proposed for the SIS100 synchrotron under construction at GSI [1].

The inputs to the BPC system are the signal of a beam position monitor, which is proportional to the beam current, and a low-level signal proportional to the RF voltage across the gap of a reference cavity. An analog preprocessing stage [2] that includes low-pass filters reduce the wideband beam signal to a narrow-band signal at the harmonic frequency. Two RF phase detectors are used to determine the phase of each of those signals with respect to a common, arbitrary reference signal. The difference between both phases is the phase of the bunch center of gravity with respect to the RF voltage. A frequency-variable comb filter [3] is then used to extract the component at the characteristic synchrotron frequency from the phase difference; this component corresponds to longitudinal dipole oscillations. In the future, the magnitude of the beam signal will also be detected and a filter tuned to twice the synchrotron frequency will be used to identify quadrupole oscillations [4].

A system-level overview is shown in Fig. 1, and a block diagram of the digital signal processing blocks involved is shown in Fig. 2. In the following, we will focus on the phase detector and band-pass filter blocks.

* Work supported by the German Ministry of Education and Research, Grant No. 06DA9028I

† christopher.spies@mes.tu-darmstadt.de

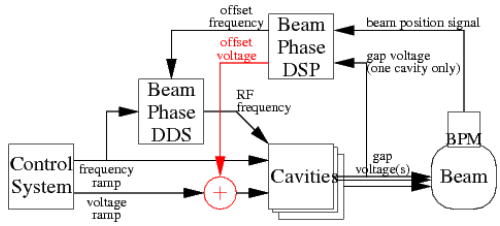

Figure 1: System level block diagram.

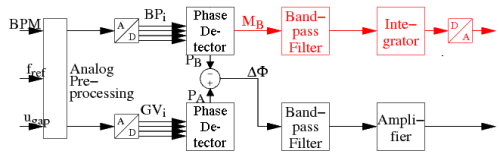

Figure 2: Signal processing block diagram.

PHASE DETECTORS

The analog preprocessing mentioned previously mixes the (variable-frequency) input signals to a fixed intermediate frequency (IF) and makes sure that the phase detector receives exactly 4 samples per IF period [2]. Assuming the phase φ of the input signals with respect to the reference is constant (or changes very little) during one IF period, 4 subsequent samples $s_{1,2,3,4}$ correspond to $+\hat{x}\sin\varphi$, $+\hat{x}\cos\varphi$, $-\hat{x}\sin\varphi$, $-\hat{x}\cos\varphi$, where $\hat{x}$ is the amplitude of the input signal. An inphase/quadrature (IQ) phase detector can be used to compute

$$i = \frac{s_1 - s_3}{2}, \quad (1)$$

$$q = \frac{s_2 - s_4}{2}, \quad (2)$$

$$\varphi = \arctan\frac{q}{i}. \quad (3)$$

The arctan function is not available on many simple digital signal processors (DSPs). Different possible approximations exist. In order to fairly compare them to each other, we subjected each implementation to the same inputs. The desired result is shown in Fig. 3 and corresponds to the expected beam phase in the SIS100 synchrotron during an $^{238}U^{28+}$ acceleration cycle. The individual approximations investigated are described in the remainder of this section. Table 1 compares them in terms of their maximum absolute error, mean absolute error and standard deviation vs. the expected result (computed using MATLAB), and Table 2 compares the hardware implementations in terms of their resource consumption.

ISBN 978-3-95450-116-8

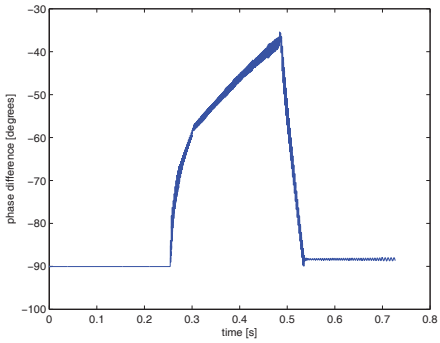

Figure 3: Phase detector test input.

Table 1: Error of Different Phase Detectors

Implementation	Max.	Mean	Std. Dev.
DSP software	0.280°	0.094°	0.081°
1024 × 12 bit LUT	0.0654°	0.0302°	0.0150°
16-stage CORDIC	0.0274°	0.0051°	0.0038°

Software Implementation

In early versions of the phase detector [2], the following approximation formula was used:

$$\arctan x \approx \frac{x}{1 + 0.28 \cdot x^2} \tag{4}$$

for $|x| \leq 1$. The DSP (a TI TMS320C6713 [2] at 225 MHz) requires 230 cycles (about 1 μs) to compute the phase difference between both inputs. A maximum sampling rate of 3.91 $\frac{\text{MS}}{\text{s}}$ could be sustained.

Hardware Implementation

Running the filter on the same DSP as the phase detector would not have been possible due to its limited computing power. It was therefore decided to offload the phase detector into a field-programmable gate array (FPGA). All hardware implementations use 16 bit integer arithmetic.

Look-Up-Table The initial implementation [5] used a look-up table (LUT). The quotient $\frac{q}{i}$ is computed using a pipelined divider and used as an index into a LUT of precomputed values of arctan. This architecture can sustain a maximum sampling rate of about 150 $\frac{\text{MS}}{\text{s}}$ [5].

Coordinate Rotation Digital Computer The LUT-based implementation was replaced by the well-known CORDIC algorithm [6] in order to improve accuracy and reduce resource consumption [7]. This architecture can sustain a maximum sampling rate of about 169 $\frac{\text{MS}}{\text{s}}$. The CORDIC algorithm has the added advantage of simultaneously computing the magnitude of each input signal. This enables the detection of longitudinal quadrupole oscillations of the beam [4].

Table 2: FPGA Footprint of Different Phase Detectors

Implementation	Slices	Flipflops	LUTs
1024 × 12 bit LUT	878	910	1,026
16 bit, 16-stage CORDIC	478	788	792

BAND PASS FILTER

The computed phase difference is fed into a band-pass filter in order to detect oscillations at the synchrotron frequency (dipole oscillations) or integer harmonics thereof (quadrupole, sextupole, ... oscillations). The synchrotron frequency can vary from a few Hz to a few kHz, i. e. it is highly variable but much smaller than the sampling frequency. This necessitates a very large sample memory (16,384 samples) and it was decided to use a sparse coefficient vector (at most 64 nonzero taps) [3, 8]. The output y is computed from the current input x_0 and past inputs $x_1, \ldots, x_{N-1}$ as

$$y_k = a \cdot y_{k-1} + g \cdot \sum_{j=1}^{64} b_j \cdot x_{d_j}, \tag{5}$$

where a is a feedback factor selecting between FIR ($a = 0$) and IIR ($a \neq 0$) operation, g is the filter gain, b_j is the j-th nonzero coefficient and d_j is the position of the j-th nonzero coefficient in the coefficient vector.

Typically, only 3 nonzero coefficients are used ($b_0 = 1$, $b_1 = -2$, $b_2 = 1$). They are equally spaced ($d_0 = 0$, $d_1 = d$, $d_2 = 2 \cdot d$) and the tap distance is

$$d = (2 \cdot f_C \cdot T_S)^{-1}, \tag{6}$$

where f_C is the desired center frequency and T_S is the sampling interval.

An iterative implementation of the filter consumes 3,874 FPGA slices and can sustain a data rate of 1.56 $\frac{\text{MS}}{\text{s}}$ [8]. Pipelined implementations can sustain higher data rates (up to 100 $\frac{\text{MS}}{\text{s}}$) at the expense of FPGA resources. Note that the phase detectors consume 4 successive input samples to compute one phase value, so the input data rate of the filter is only $\frac{1}{4}$ of the phase detector's input data rate.

Table 3 (taken from [8]) compares different possible filter implementations in terms of their FPGA footprint.

CONCLUSIONS AND FUTURE WORK

A frequency offset is computed from the filter output and added to the RF frequency in order to dampen any longitudinal oscillations. This frequency offset is sent from the controller to the synthesizer generating the RF frequency over an optical direct link [9]. The maximum data rate of this link is 40 $\frac{\text{Mbit}}{\text{s}}$ and the packet length is 55 bit. Each packet contains one frequency value. The maximum data rate at the filter output is therefore limited to 728 $\frac{\text{kS}}{\text{s}}$; otherwise, the optical link would be overloaded. This, in turn,

ISBN 978-3-95450-116-8

Table 3: FPGA Footprint of Different Filter Architectures [8]

Implementation	Slices	RAM Blocks	Multipliers
64-stage pipeline	4,826	16	56
32-stage pipeline	4,770	16	32
16-stage pipeline	4,760	16	16
8-stage pipeline	4,769	16	8
4-stage pipeline	4,528	16	4
iterative	3,874	16	1

limits the maximum data rate of the phase detectors to $2.91\,\frac{\mathrm{MS}}{\mathrm{s}}$. Any of the investigated implementations is capable of this data rate; however, the used DSP cannot run the phase detectors and the filter at the same time. Therefore, the limiting factor is the communication link, and contrary to previous publications [8], an improvement of the throughput of the signal processing hardware is uncalled for.

The actual data processing takes little time (well below $2\,\mu$s with any combination of implementations) compared to the total delay including data transmission (about $10\,\mu$s [1]), so the computational latency is also not an issue.

All investigated implementations are sufficiently accurate; the maximum error is well below the accuracy achievable in practice (about 1° due to noise). Again, contrary to previous publications [10, 11], an improvement of the accuracy is uncalled for.

We conclude that future work should concentrate on minimizing the hardware resource consumption in order to be able to offload more functionality onto the existing device.

ACKNOWLEDGMENTS

The authors would like to thank Surapong Pongyupinpanich for his help in preparing this paper.

REFERENCES

[1] H. Klingbeil, B. Zipfel, M. Kumm, P. Moritz, "A Digital Beam-Phase Control System for Heavy-Ion Synchrotrons," IEEE Trans. Nucl. Sci. 54(6), 2007, pp. 2604–2610.

[2] H. Klingbeil, "A Fast DSP-Based Phase-Detector for Closed-Loop RF Control in Synchrotrons," IEEE Trans. Instrum. Meas. 54(3), 2005, pp. 1209–1213.

[3] P. Surapong, C. Spies, M. Glesner, H. Klingbeil, "Design of frequency-variable digital filters for beam phase control," GSI Scientific Report, 2011, p. 340.

[4] D. Lens et al., "Damping of Longitudinal Modes in Heavy-Ion Synchrotrons by RF-Feedback," *Proc. Intl. Conf. Control Applications*, 2010, pp. 1737–1742.

[5] A. Guntoro et al., "Implementation of Realtime and High-speed Phase Detector on FPGA," in K. Bertels, J. M. P. Cardoso, S. Vassiliadis (Eds.), "*Reconfigurable Computing*", Springer, 2006, pp. 1–11.

[6] P. K. Meher et al., "50 Years of CORDIC — Algorithms, Architectures, and Applications," IEEE Trans. Circ. Syst. 56(9), 2009, pp. 1893–1907.

[7] A. Guntoro, M. Glesner, "Resolving longitudinal amplitude and phase information of two continuous data streams for high-speed and real-time processing," Advances in Radio Science 7, 2009, pp. 133–137.

[8] P. Surapong, M. Glesner, H. Klingbeil, "Implementation of Realtime Pipeline-Folding 64-Tap Filters on FPGA," *Proc. Conf. Ph. D. Research in Microelectronics and Electronics*, 2010, pp. 1–4.

[9] H. Klingbeil, U. Laier, K.-P. Ningel, S. Schäfer, C. Thielmann, B. Zipfel, "New digital low-level rf system for heavy-ion synchrotrons," Phys. Rev. ST AB 14(10), 2011, pp. 102802/1–9.

[10] P. Surapong, M. Glesner, "Pipelined Floating-Point Architecture for a Phase and Magnitude Detector based on CORDIC," *Proc. Intl. Conf. Field Programmable Logic and Applications*, 2011, pp. 382–384.

[11] F. A. Samman, P. Surapong, C. Spies, M. Glesner, "Floating-point-based hardware accelerator of a beam phase-magnitude detector and filter for a beam phase control system in a heavy-ion synchrotron application," *Proc. Intl. Conf. Accelerators and Large Experimental Physics Control Systems*, 2011, pp. 683–686.

ISBN 978-3-95450-116-8