ONLINE BUNCH BY BUNCH TRANSVERSE INSTABILITY DETECTION IN LHC

M.E. Söderén*, G. Kotzian, M. Ojeda-Sandonís, D. Valuch, CERN, Geneva, Switzerland

Abstract

Reliable detection of developing transverse instabilities in the Large Hadron Collider is one of the main operational challenges of the LHC's high intensity proton run. A full machine snapshot provided from the moment of instability is a crucial input to develop and fine tune instability models. The transverse feedback system (ADT) is the only instrument in LHC, where a full rate bunch by bunch transverse position information is available. Together with a sub-micron resolution it makes it a perfect place to detect transverse beam motion. Very large amounts of data, at very high data rates (8 Gb/s) need to be processed on the fly to detect onset of transverse instability. A very powerful computer system (so called ADTObsBox) was developed and put into operation by the CERN RF group, which is capable of processing the full rate data streams from ADT and perform an on the fly instability detection. The output of this system is a timing event with a list of all bunches developing instability, which is then sent to the LHC-wide instability trigger network to freeze other observation instruments. The device also provides buffers with raw position data for offline analysis.

INTRODUCTION

Since the ADTObsBox observation system had been commissioned in 2015, it is used for injection oscillation transient analysis, with online and offline beam parameter and transverse feedback parameter extraction. The system is also an invaluable resource for the machine development studies, as it is the only place in the LHC machine where a bunch-by-bunch transverse position data is available for unlimited number of turns. The positional data are available to all users or other systems in a form of data arrays through a subscription to the ObsBoxBuffer FESA class [1].

An online transverse instability detection system has been designed to detect an onset of transverse instability with a detailed information about the activity already at the very time when the activity is detected. By sending a trigger through the LIST network [2] to all relevant observation instruments in the LHC, a snapshot of the critical LHC machine parameters at the time of the instability can be captured, what greatly simplifies the later analysis of the cause of the instability. There is no need anymore to blindly record large amounts of data, with a high probability of missing the interesting bit. Once the trigger is sent, the other instruments know where and when to look already.

* martin.soderen@cern.ch

DETECTION

The system has to be able to detect a large variety of instability signatures. These include very different rise times (typically from 100's turns, up to 100k's turns) or a very different activity behavior. At the same time, the system should produce as few false triggers as possible. Various detection methods have been already tested in the LHC using the BBQ tune measurement system [3]. The BBQ system measures an average of transverse activity of all bunches, therefore it is not completely optimal to detect an instability of individual bunches. On the other hand, the ADTObsBox has access to bunch by bunch data, and some of the discussed techniques [4], adopted for very high throughput pickup data can be employed. The first online detection system, presented in this paper uses the moving average algorithm.

Algorithm

The algorithm uses three moving averages, each with subsequently longer integration window. Currently, the window lengths W of 256, 1024 and 4096 turns are defined. Use of multiple windows with different lengths allows for reliable detection of activities with very different rise times. The algorithm compares the value of the current window (the average of the last W samples) against a threshold, which is calculated as an accumulated sum of the previous windows (see algorithm 1). The detection sensitivity, efficiency and immunity to perturbations can be adjusted by careful selection of the β and ω parameters. Detection of fast instabilities happening shortly after injection calls for a higher threshold and usage of only very limited amount of historic data, while reliable detection of very slowly rising instabilities requires low relative change and long data records. In the implementation, each of the three running sums allows for individual β and ω values.

Analytic Signal Reconstruction

The normalized bunch transverse position is received by the ADTObsBox in a form of 1 GBit/s stream of 16-bit integer data (per pickup). The data is sliced turn-by-turn, converted to float and memory-aligned with 32 bit for AVX2 compatibility [5]. The closed orbit offset is removed by a two tap notch filter and the bunch oscillation amplitude is calculated from the position data using the Hilbert transformer [6]. This method is already used in the ADT for analyzing the damper performance [7]. The oscillation amplitude is calculated from the I and Q components and passed to the instability detection algorithm. An overview of the calculation pipeline can be seen in Fig. 1.

ISBN 978-3-95450-182-3

Data: Oscillation Amplitude($A[n]$)
Result: instability trigger
/* Prefill avg and avg_{old} with samples */
$avg = \sum_{n=0}^{W} \frac{A[n]}{W}$;
$thres = avg$;
/* Do this for every amplitude sample */
while *forever* **do**
  for *W times* **do**
    /* update average and remove the oldest sample */
    $avg+ = new \frac{A}{W}$;
    $avg- = oldest \frac{A}{W}$;
  end
  /* ω adjusts the sensitivity */
  if $thres < \omega \times avg$ **then**
    send trigger;
  end
  /* β can be set to limit the growth rate of the threshold */
  $thres = \beta \times avg + (1 - \beta) \times thres$;
end

Algorithm 1: Simplified algorithm.

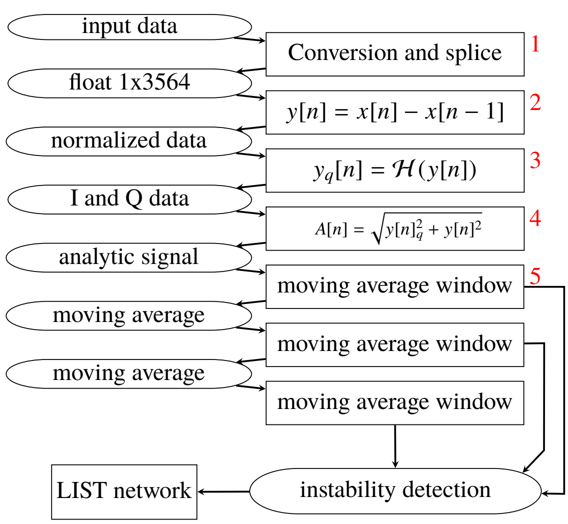

Figure 1: The LHC online instability detection pipeline.

Algorithm Tuning and Testing

Data from the ADTObsBox with known transverse instabilities has been stored from the LHC operations during 2016. Algorithm tuning was automated by first manually extracting interesting data sets and adding meta data about the instability. A test environment was set up and the pipeline was tested with this data and many combinations of configurations to find the maximum coverage of instabilities and as few false triggers as possible. Example output from the algorithm can be seen in Fig. 2.

IMPLEMENTATION

The ObsBoxes are four off-the-shelf SuperMicro 6028U-TR4+ servers which are capable of concurrent online data analysis [8]. On the servers there are several instances of the ObsBoxBuffer FESA class running, which allows for data acquisition of different pickups and for different number of turns. Specific buffers, providing the full throughput of 1 Gbit/s per pickup are reserved for online analysis. A FESA class called ALLADTCopra, subscribes to these buffers and performs the online instability analysis.

Handling the High Throughput

The biggest challenge for the online transverse instability detection is to handle the high data throughput. The application needs to process a data-stream at 1 Gbit/s. The first implementation was a proof of concept where the implementation was non-multithreaded and used no special instructions or matrix libraries. It just used standard C++ operations which were applied to a complete data package which in this case is a 4094×3564 matrix. In that implementation the analysis of 4096 turns took 6 seconds. A simple multi threaded library called ParallelFir was implemented which did the analysis in the same way but the computation was split up, so each thread worked on a sub-set of bunches. This reduced the computation time to 1.8 seconds.
By optimizing ParallelFir using SSE4.1 instructions the computation time was further reduced to 0.6 seconds. To greater improve the performance, a pipeline design was introduced. Each stage in the pipeline uses extensively the Advanced Vector Extension (AVX2) instruction set introduced by Intel in their Haswell micro-architecture. It uses 256 bit registers specifically for SIMD type instructions. The splicing of the data into separate turns reduced the number of cache misses since the data was always read consecutive. All these combined measures dramatically reduced the computation time down to the final 80 ms which is approximately 5 times faster than the input data rate.

Performance Testing

The performance of the implementation was compared with a native C++ implementation. Both were compiled with G++-4.4.7 which is the standard compiler for front-end computers at CERN. Maximum compiler optimization was enabled so the native implementation used auto vectorization. For the conversion stage data for 131072 turns and 3564 bunches were simulated. For the other stages data for one turn and 128000000 bunches were simulated. The result can be seen in Table 1.

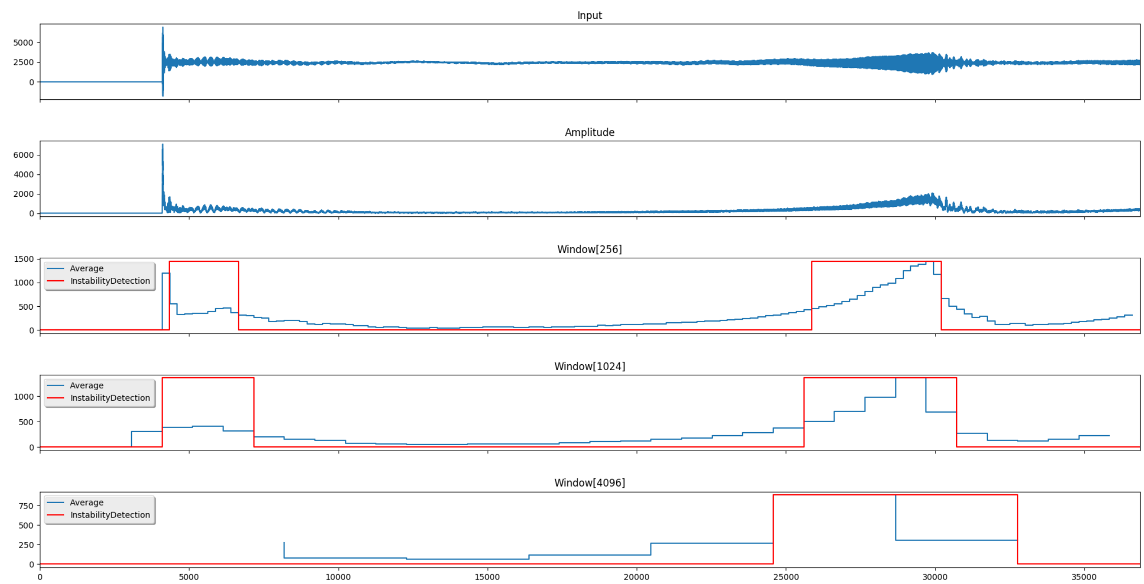

Figure 2: Output from instability detection with data from observations.

Table 1: AVX vs Native C++ Comparison

Conversion	Instructions	Cycles	CPU time[ms]
Native	1534437270	714719743	285
AVX	793964717	279384376	151
Reduction	48.3%	60.9%	47.0%

Notch	Instructions	Cycles	CPU time[ms]
Native	1430236599	493511370	268
AVX	145416604	143719902	100
Reduction	89.8%	70.9%	62.6%

IQ	Instructions	Cycles	CPU time[ms]
Native	4158509240	1361599075	559
AVX	2003146566	1233847315	515
Reduction	51.8%	9.4%	7.9%

Amplitude	Instructions	Cycles	CPU time[ms]
Native	2845177110	4487822502	1692
AVX	225351692	185194294	128
Reduction	92.1%	95.9%	92.4%

The conversion, notch and amplitude stages show a significant reduction of computation time. The improvement of the IQ stage is less significant. A probable cause is that the I component is obtained by a delay, implemented with a memcpy.

RESULTS AND CONCLUSION

It was demonstrated, that the implementation can handle the full data throughput from the ADTObsBoxes 5 times fast than required, providing a true online, bunch by bunch transverse instability detection system for the LHC accelerator. The detection algorithm was tuned using real data samples from the machine and will be put into operation after the winter technical stop 2016/2017. The detection system sends a trigger to all observation systems connected to the LIST [2] along the LHC machine in case a growing transverse activity is detected, together with detailed information about which bunches are unstable, at the moment the instability is detected. This greatly improves the quality and turnaround of instability analysis, for example during scrubbing, or other special runs.

REFERENCES

[1] M. Arruat *et al.*, "Front-end Software Architecture", in *Proc. ICALEPCS'07*, Knoxville, Tennessee, USA, Oct. 2007, paper WOPA04, pp. 301–312.

[2] T. Włostowski, G. Daniluk, M. Lipinski, J. Serrano, F. Vaga, "Trigger and RF distribution using White Rabbit", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, paper WEC3O01, pp. 619–623.

[3] M. Gasior, "Faraday cup award: high sensitivity tune measurement using Direct Diode Detection", in *Proc. BIW'12*, Newport News, VA USA, April 2012, paper MOAP02.

[4] T. Levens, K. Łasocha, T. Lefevre, "Recent developments for instability monitoring at the LHC", in *IBIC'16*, Barcelona, Spain, Sept. 2016, paper THAL02, pp. 852–855.

[5] "How Intel® AVX2 improves performance on server applications", `https://software.intel.com/en-us/articles/how-intel-avx2-improves-performance-on-server-applications`

[6] A.V. Oppenheim, R. W. Schafer, and R. John, "Discrete Hilbert Transforms", in *Discrete-Time Signal Processing – SE*. New Jersey, USA: Prentice Hall, 1999, pp. 775–801.

[7] G. Kotzian, "Transverse feedback parameter extraction from excitation data", presented at IPAC'17, Copenhagen, Denmark, May 2017, paper TUPIK094, this conference.

[8] M. Ojeda *et al.*, "Processing high-bandwidth bunch-by-bunch observation data from the RF and transverse damper systemsd of the LHC", in *ICALEPCS'15*, Melbourne, Australia, Oct. 2015, paper WEPGF062, pp. 841–844.

ISBN 978-3-95450-182-3