# INTERNAL POST OPERATION CHECK SYSTEM FOR KICKER MAGNET CURRENT WAVEFORMS SURVEILLANCE

Nicolas Magnin, Etienne Carlier, Brennan Goddard, Volker Mertens, Jan Uythoven
CERN, Geneva, Switzerland

*Abstract*

A software framework, called Internal Post Operation Check (IPOC), was developed to acquire and analyse kicker magnet current waveforms. It was initially aimed at performing the surveillance of LHC Beam Dumping System (LBDS) extraction and dilution kicker current waveforms. It was subsequently also deployed on various other kicker systems at CERN. It was implemented using the Front-End Software Architecture (FESA) framework, and uses many CERN control services. It provides a common interface to various off-the-shelf digitiser cards, allowing a transparent integration of new digitiser types into the system. The waveform analysis algorithms are provided as external plug-in libraries, leaving their specific implementation to the kicker system experts. The general architecture of the IPOC system is presented in this paper, along with its integration within the control environment at CERN. Some application examples are provided, including the surveillance of the LBDS kicker currents and trigger synchronisation, and a closed-loop configuration to guarantee constant switching characteristics of high voltage thyratron switches.

## INTRODUCTION

The first IPOC applications have been developed in 2008 for the surveillance of the current in the extraction and dilution kicker magnets (MKD&MKB) of the LBDS [1]. They were deployed as three independent FESA applications, each being able to interface with a unique digitiser type (National Instrument PXI-5122 or ACQIRIS DC-270), and to execute a unique waveform analysis algorithm. Due to the similarity of the three applications, code was duplicated and the maintenance was difficult. Later a new deployment of an IPOC system was requested for the surveillance of the current in the injection kicker magnets (MKI) of LHC, and it was decided to change the architecture of the IPOC system to make it a more generic, configurable system dedicated to the acquisition and analysis of analogue signal waveforms.

The new system supports various off-the-shelf digitiser types and can exchange data with any device that interfaces with the CERN Controls Middleware (CMW) using the Remote Device Access (RDA) protocol [2], allowing the realisation of complex heterogeneous macro-systems involving various industrial control sub-systems.

To deploy a new IPOC system one only has to provide the analysis algorithms as external library and then to configure the system using XML files.

The implementation of the IPOC system is presented in this paper, along with application examples.

## ARCHITECTURE & IMPLEMENTATION

An IPOC system is composed of a computer running the IPOC server application, fitted with one CERN Timing Receiver (CTR) card and one or more digitiser card(s). The overview of the IPOC system is shown in Fig. 1.

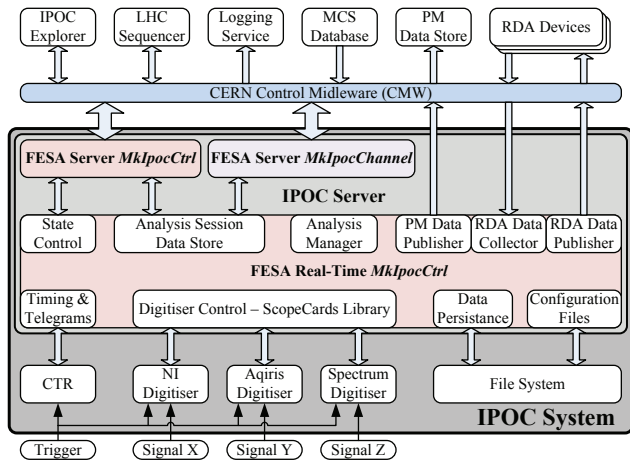


Figure 1: Block diagram of the IPOC system.

The IPOC server application was implemented using the Front-End Software Architecture (FESA) framework [3]. It is a composition of two FESA classes: The *Controller* class MkIpocCtrl, and the *Channel* class MkIpocChannel**.** It is deployed as a single multi-threaded process, with one instance of *Controller* per IPOC system, each containing a collection of *Channel* instances, one per digitiser channel to acquire and analyse.

**FESA Real-Time:** The *MkIpocCtrl* class handles the real-time scheduling of the core IPOC functions shown in Fig. 1, such as rearming the system, controlling the digitisers, executing the analysis modules or pushing the result data to the Post-Mortem Data Store [4]. These core functional blocks are detailed in the following sub-section.

**FESA Server:** The *MkIpocCtrl* and *MkIpocChannel* classes provide the CMW interfaces needed to control the IPOC system remotely using the *IPOC-Explorer* Java application (screenshot shown in Fig. 5) or the *LHC Sequencer* application. They allow as well accessing the configuration of the system to be checked against the *Machine Critical Settings (MCS)* database, or the saving of the latest analysis session results by the *LHC Logging service* [5].

ISBN 978-3-95450-139-7

The IPOC system makes use of three C++ libraries, *IpocFesa*, *IpocAnalyser* and *ScopeCards*, deployed as Linux shared object files.

## *The IpocFesa library*

The *IpocFesa* library implements all the logic for the IPOC server as shown in Fig.1. The user code in the FESA classes just calls this library.

**State Control:** The IPOC system can have different operational states as described in Fig. 2.

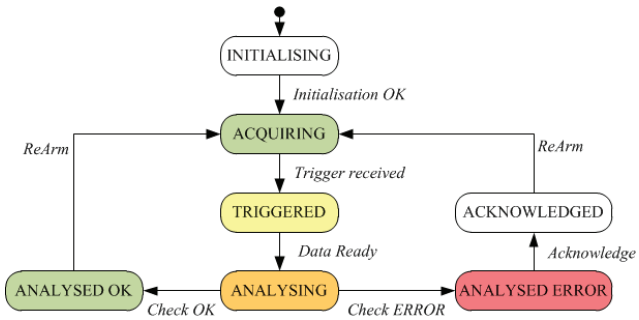


Figure 2: IPOC system states.

The system starts in the *Initialising* state, in which the hardware is configured and all the analysis modules are loaded. After a correct initialisation the system is automatically armed in the *Acquiring* state, where all digitisers are in acquisition, waiting for a trigger. When the system receives a trigger, it moves to the *Triggered* state where it retrieves the waveform data from all the digitisers. The system then enters the *Analysing* state where it executes the analysis module of all channels, one after the other. In case the overall result is OK, the system goes into the *Analysed_OK* state and waits for a rearming command. In case a problem is detected, the overall analysis result is not OK, and the system falls into the *Analysed_ERROR* state. In this state the system cannot be rearmed, and the intervention of an expert is required to evaluate the situation and acknowledge the error, after which the system can be rearmed normally.

**RDA Data Collector:** For each analysis module one can define a list of RDA 'Device/Property' to be used as input data, in addition to the waveform from the digitiser. The Data Collector will establish a subscription to the device property and take the first data received within a configured time window w.r.t the trigger timestamp.

**Analysis Session Data Store:** Each time an analysis session is started a new data store is created to contain all data regarding the analysis session, including the collections of analysis module input data (digitiser waveforms, collected data and persistent data), the digitiser status at the time of trigger (horizontal and vertical settings, temperatures), and the collections of analysis module result data.

**Analysis Manager:** This block is responsible for the initialisation, configuration and execution of the analysis modules defined for every *Channel* and *Controller* instance. At the reception of a trigger on the CTR card, the IPOC server starts a new analysis session. All the *Channel* analysis modules are executed one after the other, and then the *Global* analysis module is called for the *Controller* to determine the overall check result.

**RDA Data Publisher:** After the execution of an analysis session, all the analysis module output data that has been configured for this can be published, i.e. used as a setting data for any RDA Device/Property.

**Data Persistence:** Any output data can be configured as 'persistent', i.e. it will be saved to an XML file at the end of an analysis session, and will be re-injected as input data into the next analysis session.

**PM Data Publisher:** Any data at the output of the analysis module can be configured to be pushed onto the PM Data Store after every analysis session, for the data to be further used in Post-Mortem analyses like XPOC [4].

**Digitiser Control:** In many applications one has to adapt the range of the digitisers to some external conditions on which the digitiser input signal amplitude depends indirectly. Each time a timing telegram data (such as beam energy) or a RDA collected data (such as kicker generator settings) changes, the *Digitiser Control* re-evaluates the expected input signal amplitude, and determines the best range to be used, based on the digitiser channel vertical characteristics.

**Configuration files:** Every module described in this section is initialised from an XML configuration file. These files can be stored in the file system, but usually they are committed to an SVN repository, delivered and then accessed through a web server. This guarantees a better follow-up of the IPOC systems parameters, and allows an easy roll-back of the configuration if needed.

## *The ScopeCards Library*

The various applications of the IPOC system have different requirements for digitiser performances, for instance in terms of bandwidth and resolution. This results in an inhomogeneous operational configuration based on different types of off-the-shelf digitisers from various manufacturers. The user libraries provided by the various manufacturers are proprietary and their Application Programming Interfaces (APIs) are not compatible with each other. So in order to integrate them all in the IPOC system, the *ScopeCards* library has been introduced to define an abstraction layer, thanks to which it is relatively easy to support a new digitiser type without having to recompile the applications which will use it. The library is based on a simple model of the digitiser functionalities, offering only the features needed for the IPOC applications, as shown in Fig. 3.

According to this model each digitiser card has one *Horizontal control* block allowing the setting of the sample rate, the acquisition length, and the trigger source.



ISBN 978-3-95450-139-7

Up to now only an external trigger source is supported, with as parameter the coupling, the level and the edge. Each digitiser card contains a collection of input channels with configurable input impedance and coupling. Each channel provides a collection of ranges, every range having its own offset adjustment limits.

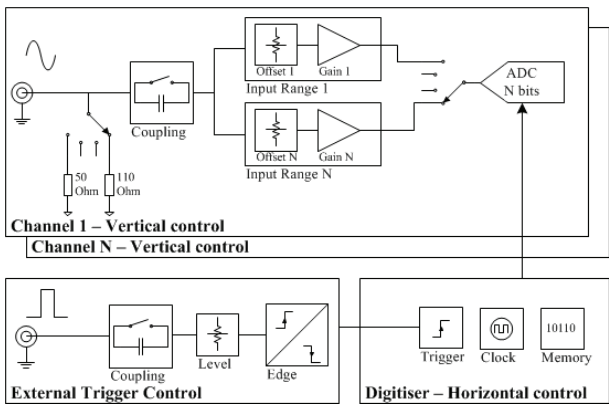


Figure 3: Simple digitiser model of ScopeCard library.

The library also provides the horizontal characteristics for each digitiser card, as well as the vertical characteristics for each of its channels, describing the possible values for all the parameters of a given digitiser.

### *The IpocAnalysis Library*

This library provides the API for developing an analysis module and the tools to create a new reusable analysis plugin library. The programmer basically has to provide an *IpocAnalyser* class and implement its *analyse* method, which takes a collection of *IpocData* as input, and must return another collection of *IpocData* as output, as presented in Fig. 4.

The input data collection always contains the predefined *IpocContext* data, which provides information on the dump event such as timestamp, cycle name, beam energy and intensity, and the RAW *IpocWaveform* data as captured by the digitiser channel. It can contain also *Collected* and *Persistent* data, depending on the configuration.

The output data collection contains the *IpocResult* data, which provides the IPOC analysis check result. It can also contain a *Filtered* waveform data, or any other data that can later be used for *Publishing* or *Persistence* depending on the configuration.

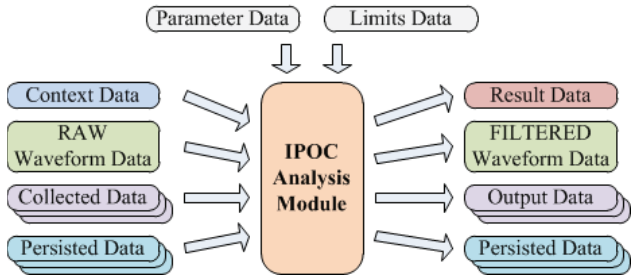


Figure 4: Simplified data flow of an analysis module.

The *IpocAnalyser* object also receives at start-up a *Parameter* data and a *Limits* data to configure the analysis module.

The implementation of all these *IpocData* container objects must be provided along with the analysis module code. The newly created analysis library is deployed as a Linux shared object file, and can then be dynamically loaded by the various IPOC systems as needed.

## APPLICATION EXAMPLES

### *LBDS Post-Operation Checks*

The surveillance of the LBDS extraction and dilution kicker magnet currents is performed by the three MKD and MKB IPOC systems in place [1]. LBDS kicker magnets are powered by high-voltage generators which are composed of internal redundant paths, and are triggered by four redundant Power Trigger Units (PTUs). The LBDS also relies on a complex Trigger Synchronisation and Distribution System (TSDS) [6], partially fault-tolerant thanks to a high level of system redundancy.

To guarantee that the LBDS is 'as good as new' for the next operational cycle, one has to make sure that all the fault-tolerant redundant parts operated properly for every beam dump. For this purpose 30 identical IPOC systems composed of a *Kontron KISS 4U* computer fitted with 3 digitiser cards *Spectrum M2i.3013* (4 ch, 12 bit, 40 MS/s) are installed. They acquire and analyse the 10 internal

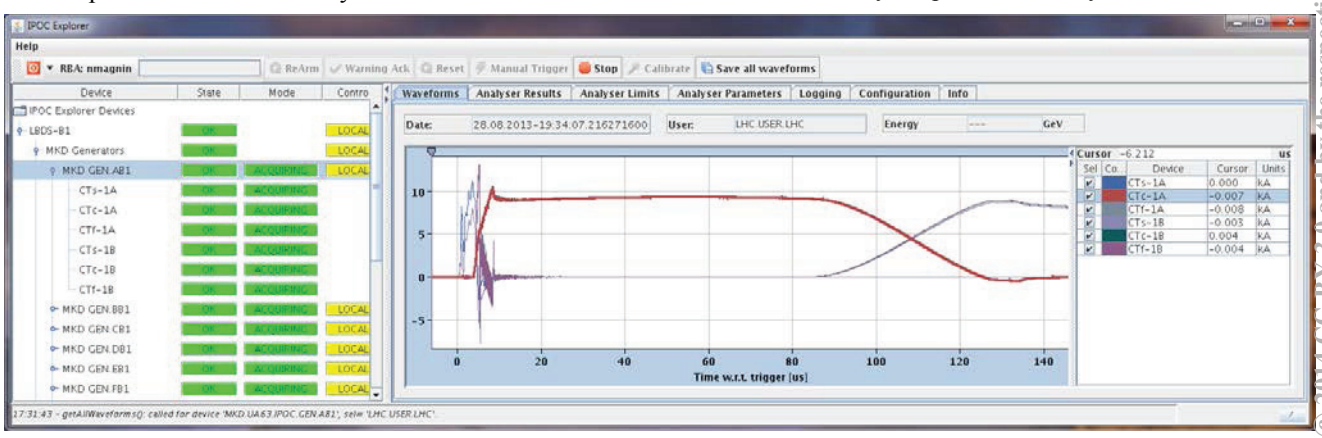


Figure 5: IPOC-Explorer application displaying the MKD generator internal currents.

ISBN 978-3-95450-139-7

currents of the generators and their PTUs, to ensure that their strengths are correct w.r.t. the beam energy and that they are properly shared among the redundant paths. Another two *Kontron KISS 4U* computers, fitted with 3 fast digital I/O cards *Spectrum M2i.7020* (32 ch, 1 bit, 125MS/s), are deployed to capture and analyse up to 192 signals of the TSDS to verify their presence and correct synchronisation.

The results of the *MKD-Generator* and the *TSDS* IPOC analyses are also checked by the XPOC system [4] which will guarantee that no beam can be injected in the LHC in case a problem has been detected in the LBDS.

Figure 5 shows a screenshot of the IPOC-Explorer Java expert application displaying the internal currents of one MKD generator.

After every beam dump, more than **500** waveforms are acquired and analysed by the various IPOC systems deployed at the LBDS.

### *Thyratron Switches Closed-loop Control*

The injection kicker system of the SPS makes use of high voltage thyratron switches to achieve the required fast rise time of the high current pulse in the kicker magnets. These switches contain gas and use heater modules to maintain the gas pressure to a nominal value, to obtain the nominal rise time and turn-on delay. The thyratron switch control elements are the following:

- The Thyratron Heater Controller (THC) is based on a Siemens S7-400 industrial PLC system, controlling the heater power supplies of the thyratron switches.
- The Kicker Timing System (KiTS) is composed of a VME crate, fitted with various off-the-shelf fine delay boards, and a *MEN A20* CPU board running the FESA application that generates the precisely synchronised triggers for the thyratron switches.

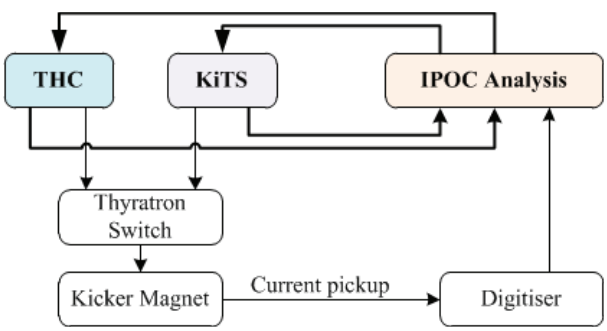


Figure 6: Closed-loop IPOC system planned for the control of thyratron switches.

With the aging of the switches, their characteristics evolve; basically the rise time will diminish and the turn-on delay will increase. When the switch characteristics exceed acceptable limits and generate interlocks, a manual intervention is required to adjust the THC voltage and the KiTS fine delay, to compensate for the degradation of the switch characteristics.

To automatically compensate for these deviations, an IPOC system will be installed to perform a double feedback loop to adjust the THC and the KiTS settings, as shown in Fig. 6. The THC and KiTS setting values will be collected and sent, along with the magnet current waveform, to the IPOC analysis module which will determinate the waveform characteristics (rise time, delay, strength), and will check that these values are correct w.r.t. the setting values. It will perform a long term pulse-to-pulse averaging of these measurements, using the *Data Persistence* mechanism, compute new setting values and send them back to the THC and the KiTS systems using the *Data Publishing* mechanism. This application is under development and it has only been tested in the laboratory to validate the IPOC functionalities involved. It will be deployed in operation for the SPS start-up in 2014.

## CONCLUSION

The IPOC system has evolved over the past five years to adapt to the different requirements of the various systems into which it has been integrated. It has now become a fully-configurable pulse-to-pulse waveform acquisition and analysis system that allows complex macro-system configurations involving the control of any industrial system that interfaces with CMW. It provides a plugin interface for analysis algorithms and supports a wide range of off-the-shelf industrial digitisers.

It is integrated into the LBDS post-operation checks and is therefore part of the LHC Machine Protection System. In this context it has proven its reliability over the past years of LHC operation.

The IPOC system will also be deployed in a more complex configuration to realise a double feed-back closed-loop system for the automatic regulation of thyratron switch characteristics.

References

[1] J. Uythoven et al., "Experience with the LHC beam dump post-operational checks system," in *Particle Accelerator Conference*, Vancouver, Canada, 2009.

[2] N. Trofimov et al., "Remote Device Access in the New CERN Accelerator Controls Middleware," in *ICALEPCS*, San Jose, CA. U.S.A., 2001.

[3] A. Guerrero et al., "CERN Front-End Software Architecture for Accelerator Controls," in *ICALEPCS*, Gyeongju, Korea, 2003.

[4] N.Magnin et al., "External Post-Operational Checks for the LHC Beam Dumping System," in *ICALEPCS*, Grenoble, France, 2011.

[5] C. Roderick et al., "The LHC Logging Service: Handling Terabytes of On-line Data," in *ICALEPCS*, Kobe, Japan, 2009.

[6] A.Antoine et al., "The LHC Beam Dumping System Trigger Synchronisation and Distribution System," in *ICALEPCS*, Geneva, Switzerland, 2005.



ISBN 978-3-95450-139-7