

J.M. López, J. Vega, D. Alves, Member, S. Dormido-Canto, A. Murari,
J.M. Ramírez, R. Felton, M. Ruiz, G. de Arcas
and JET EFDA contributors

Implementation of the Disruption Predictor APODIS in JET's Real-Time Network using the MARTe Framework

“This document is intended for publication in the open literature. It is made available on the understanding that it may not be further circulated and extracts or references may not be published prior to publication of the original when applicable, or without the consent of the Publications Officer, EFDA, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK.”

“Enquiries about Copyright and reproduction should be addressed to the Publications Officer, EFDA, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK.”

The contents of this preprint and all other JET EFDA Preprints and Conference Papers are available to view online free at www.iop.org/Jet. This site has full search facilities and e-mail alert options. The diagrams contained within the PDFs on this site are hyperlinked from the year 1996 onwards.

Implementation of the Disruption Predictor APODIS in JET's Real-Time Network using the MARTe Framework

J.M. López¹, J. Vega², D. Alves³, S. Dormido-Canto⁴, A. Murari⁵,
J.M. Ramírez⁴, R. Felton⁶, M. Ruiz¹, G. de Arcas¹
and JET EFDA contributors*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

¹*CAEND CSIC-UPM. Universidad Politécnica de Madrid, Spain*

²*Asociación EURATOM CIEMAT para Fusión Madrid, Spain*

³*Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal*

⁴*Dpto. Informática y Automática. UNED. Madrid, Spain*

⁵*Consorzio RFX-Associazione EURATOM ENEA per la Fusione, Padova, Italy*

⁶*EURATOM-CCFE Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK*

* *See annex of F. Romanelli et al, "Overview of JET Results", (23rd IAEA Fusion Energy Conference, Daejeon, Republic of Korea (2010)).*

ABSTRACT

This paper describes the implementation of a real-time disruption predictor that is based on Support Vector Machine (SVM) classifiers. The implementation was performed under the MARTe framework on a six-core x86 architecture. The system is connected via JET's Real-time Data Network (RTDN). The online results show a high degree of successful predictions and a low rate of false alarms, thus confirming the usefulness of this approach in a disruption mitigation scheme. The implementation shows a low computational load, which will be exploited in the immediate future to increase the prediction's temporal resolution.

1. INTRODUCTION

THE evolution of machine learning techniques and the technological evolution of computer architectures and operating systems in recent years are enabling new approaches to complex problems in different areas of industry and research in which a classical approach is nonviable due to lack of knowledge about the problem's nature. A typical example of this situation is the prediction of plasma disruptions in Tokamak devices. Disruptions in Tokamak devices are unavoidable and can have a significant impact on machine integrity [1]. JET's new metal wall, which is made of solid beryllium, tungsten, and tungsten-coated, is much less robust than the previous one, so developing mechanisms to predict this phenomenon is crucial. Disruption prediction is a very complex task, not only because it is a multidimensional problem but also because to be effective, it has to detect the actual disruptive event early in advance to be able to deploy successful mitigation strategies. With these requirements in mind, a real-time disruption predictor, APODIS (Advanced Predictor Of DISruptions), has been developed for use in the JET Tokamak. The predictor has been designed to run in the Multithreaded Application Real-Time executor (MARTe) framework [2]. Currently, the alarm output is stored as a signal in JET's database but it is ready to be sent to the Real-Time Protection Sequencer (RTPS) [3] system, which in turn acts by coordinating a plasma termination. APODIS is based on a combination of Support Vector Machine (SVM) classifiers [4].

This paper is structured as follows. Section II shows the predictor background and covers aspects related to the architecture, training process and signal description. Section III describes the implementation aspects of the predictor in JET using the MARTe framework. Section IV introduces the hardware basics of the implementation, the core distribution, and the IRQ routing. Finally, in section V, the results obtained during the C28 campaign are presented.

2. PREDICTOR ARCHITECTURE

APODIS [5] is based on a two-layer system of SVM classifiers. The current implementation uses seven signals to characterize the disruptive/non-disruptive plasma state: plasma current, mode lock amplitude, plasma inductance, plasma density, diamagnetic energy time derivative, radiated power and total input power. These signals are processed using 32ms time windows with a sampling frequency of 1kHz. Two features per signal are computed during the 32ms time windows: mean

value and standard deviation of the FFT, (excluding the first component). Fig.1 shows the two-layer architecture of APODIS and a time diagram showing the 32-ms windows' overlay. The first layer of APODIS is made up of three SVM classifiers with Radial Basis Function (RBF) kernels. The second layer is also a SVM classifier but based on a linear kernel [6].

Every 32ms, the classifier makes a prediction. Classifier M1 receives the most recent block of 32 samples, M2 receives the previous 32 samples, and so on. The approach uses a schema of 32ms sliding windows without overlapping. Therefore, every 32ms, only 32 samples are new. With the data from these windows, a feature vector X_N is obtained (1) every 32 samples.

$$X_N = \{A_{s0}, A_{s1}, \sigma_{s0}, \dots, A_{s6}, \sigma_{s6}\} \quad (1)$$

is the average value of a 32-sample window of signal x, and is the spectrum estimation according to equation (2).

$$\sigma_{sx} = \sqrt{\sum_{k=1}^{j=15} \frac{|s_i[k] - \bar{S}|^2}{15}} \quad (2)$$

are the values of the Discrete Fourier Transform (DFT) performed using a Fast Fourier Transform (FFT) algorithm, where k is the index of the DFT coefficients. \bar{S} refers to the mean value of s_i for $k = 1$ to 15. The window size is chosen to be a power of 2 to improve the FFT execution time [7]. The first-layer classifiers implement the following equation:

$$D_M = (\sum_{i=0} \alpha_i \cdot e^{-\gamma_{\parallel} \text{vec}_i - X_{N\parallel}^2}) - \text{Bias} \quad (3)$$

D_M is the distance for models M_1 , M_2 and M_3 , with N equal to 210, 166 and 60 for SVM(M3), SVM(M2) and SVM(M1), respectively. This expression requires a higher computational time in comparison with the FFT and is the most critical part of the APODIS predictor implementation. These RBF kernels determine whether the current discharge can be disruptive or not. Their outputs are fed to the second-layer SVM, which is a linear kernel that makes the decision about to trigger an alarm or not using the decision function (4).

$$R = \omega_3 \cdot D_3 + \omega_2 \cdot D_2 + \omega_1 \cdot D_1 + B \quad (4)$$

where ω_x are coefficients and B is an offset.

Before the predictor can be used, a training phase must be performed. The initial number of discharges used for the training process is 10,845. A manual selection has been performed to remove discharges with no signals present or signals containing only noise. In the second phase, signals outside the ranges in Table I have been removed.

After this selection process, 8407 discharges are retained. Table II shows the discharge distribution for the campaigns processed.

Two sets of disruptive discharges must be handled: intentional and non-intentional disruptions. In discharges with intentional disruptions, the plasma does not show any disruptive behavior until the disruption is provoked. Therefore, these types of discharges are removed from the training process. However, these discharges will be used for the test phase. The training datasets are composed of 100 non-disruptive discharges, which are randomly selected from a group of 2312, as well as 125 unintentional disruptive discharges. These data were taken during campaigns C15, C16, C17, C18, C19, C20 C21 and C22. The test datasets consist of 3578 non-disruptive discharges and 228 unintentional disruptive discharges from campaigns C23, C24, C25, C26, C27a and C27b. The significant amount of data involved in the training phase made it necessary to use High-Performance Computing (HPC). This HPC environment is formed by 240 nodes of 2 Quad-Core Xeon processors (X5450 and X5570) at 3 GHz and with 16 GB of RAM memory.

3. SYSTEM ARCHITECTURE USING MARTE

The Multithreaded Application Real-time executor (MARTE) framework is used for building applications optimized for multicore architectures. This framework has been used in JET for the past several years, and it has proven to be a good platform for developing highly configurable real-time applications [8], [9]. The framework is based on a real-time oriented C++ library called BaseLib2. The developer has to write the software code that implements its own application. MARTE is basically a sequential executor of generic application modules (GAMs) in a real-time context. Another important feature of MARTE is that it facilitates development, testing and debugging off line and later, rapid switching to the deployment phase due to its hardware abstraction. MARTE is optimized to develop real-time applications, and it provides many interesting high-level functionality such as messaging, http services, and execution time measurements, as well as low-level functionalities, such as semaphores, threads, timers and messaging. It provides a configurable internal state machine, which is driven remotely by JET's global state machine. The GAMs are executed according to the state transitions. The communication between GAMs must be performed by writing and reading signals from the Dynamic Data Buffer (DDB).

JET's Asynchronous Transfer Mode-based (ATM) Real-Time Data Network (RTDN) is the communication medium for all real-time systems in JET. All of the signals used by APODIS are available in the RTDN. Signals come from different real-time systems. These systems are very heterogeneous; thus, each signal has a different sampling rate. However, APODIS works at a 1-kHz sampling rate. If the sample is not available, then the RTDN provides the most recent sample. This situation is equivalent to a hold effect, which is shown in Fig.2.

APODIS has been implemented using two application threads. The first thread has two functions: a) collecting samples for the input sources, such as the JET database for debugging purposes or packets from the RTDN in the deployment phase, and b) calculating the feature vectors for the input signals. The second thread collects the data from thread one and organizes them to fit in the three-window architecture. Figure 3 shows the GAMs implemented according to the MARTE

framework philosophy: (a) *JPF/ATM*: this GAM makes the ATM packets or file signals available as DDB signals; (b) *Normalize*: this GAM normalizes the input signals according to the training range used; (c) *Mean*: calculates the mean of 32 samples; (d) *DesvFFT*: makes a spectrum estimation according to equation (2); (e) *Enable*: this GAM analysis of the plasma current signal achieves a trigger threshold to start the predictor; (f) *Data Transfer*: used to transfer blocks of 32 samples to the second thread; (g) *3 Windows*: this GAM is the FIFO stack that stores samples to form the sliding windows; (h) *First Layer*: implements the first layer of the classifier according to equation (3); (i) *Second Layer*: implements the second layer of the classifier according to equation (4); and (j) *Persistence*: these GAMS form part of the MARTe framework and are used to store the intermediate values and results generated by the application in the JET database.

4. EXPERIMENTAL IMPLEMENTATION

The system has been implemented on a six-core x86 architecture with an ethernet Network Interface Card (NIC) for remote administration and introspection and an Asynchronous Transfer Mode (ATM) NIC handling all real-time I/O within JET's RTDN. The predictor receives six ATM datagrams with the data of the required signals. These datagrams must be configured by the system administrator before they can be used. The application (a user-space) is running on a mainstream Linux vanilla kernel (Linux 2.6.35.9) and is implemented using MARTe. Real-time performance has been achieved by combining available Central Processing Unit (CPU) isolation and Interrupt ReQuests (IRQ) routing mechanisms. Table III shows the core distribution, its thread and the IRQ affinity distribution.

In terms of execution times, the second thread is the most critical one because this is where the SVM is implemented. Special care has been taken in the coding of GAMS in this thread to minimize execution time. The MARTe framework has tools to measure the execution time of GAMS, and http clients can be used to monitor the application status and signal/variable values in real-time without affecting the application's cycle time. Using these tools, the execution time of thread two has been measured. The results are quite satisfactory because the average execution time of thread two is 250us for Pulse No's: 82429 to 82905. Figure 4 shows the mean value of the execution time of thread two and its maximum value for Pulse No's: from 82429 to 82905.

RESULTS AND CONCLUSIONS

Since APODIS was installed in JET, Pulse No's: 82429 to 82905 of campaign C28 have been analyzed. Figure 5 shows the accumulative fraction of detected disruptions vs. disruption time. The system is able to predict 97% of the disruptions 30ms in advance (point A, in Fig.5). It is estimated that 30ms is a sufficient amount of time to take mitigation actions. Note that 50% of the disruptions are predicted 280 ms in advance (point B, in Fig.5).

Note, the system was trained using data from campaigns C15 to C27b. In these campaigns, JET's wall is made of carbon, and in the C28 campaign, the wall is made of metal (ILW). This situation

shows that the design procedure is very robust, and the predictor obtained is accurate and reliable (high success prediction and low rate of false alarms). The disruptions are predicted sufficiently in advance (approx. 300ms mean time). The MARTe framework allows for an implementation model based on data structures. The code is unique, and the data structures define the software behavior using configuration files. The architecture used has a reduced execution time and validates this implementation for use in real time. The results obtained show that this implementation is valid, even when compared with other architectures that use hardware implementations using FPGAs [7]

ACKNOWLEDGEMENTS

This work was partially funded by the Spanish Ministry of Science and Innovation under the Projects No. ENE2008-02894/FTN and ENE2009-10280 and carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

- [1]. A.H. Boozer, "Theory of tokamak disruptions." *Physics of Plasmas*, Vol **19**, pp. 25, Mar. 2012.
- [2]. A.C. Neto, F. Sartori, F. Piccolo, R. Vitelli, G. de Tommasi, L. Zabeo et al. "MARTe: a multiplatform real-time framework". *IEEE Transactions on Nuclear Science*. vol. **57**, n 2, pp. 479-486. April 2010.
- [3]. A.V. Stephen et al., "Centralised Coordinated Control To Protect The JET ITER-like Wall". At the 13 th International Conference on Accelerator and Large Experimental Physics control Systems (2011).
- [4]. V. Cherkassky and F. Mulier. "*Learning from data*" New York, Wiley 1998.
- [5]. G.A. Ratta, J. Vega, A. Murari, G. Vagliasindi, M.F. Johnson, P.C. Vries et al. "An Advanced Disruption Predictor for JET tested in simulated Real time Environment". *Nuclear Fusion*. vol. **50**, pp 10. 2010
- [6]. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". *Data mining and knowledge discovery*, 2. Pp 121-167 (1998).
- [7]. Ruiz, M, Vega, J, Arcas, G. Ratta, G.; Barrera, E. Murari, A. Lopez, J.M. Melendez, R."Real-time plasma disruptions detection in JET implemented with the ITMS platform using FPGA based IDAQ. *IEEE Transactions on Nuclear Science*. Vol. **58**, Issue: 4, Part:1. pp: 1576-1581. 2011
- [8]. D. Alves, R. Vitelli, L. Zaccarian, L. Zabeo, A. Neto, F. Sartori, et al., "The new Error Field Correction Coil controller system in the Joint European Torus tokamak". *Fusion Engineering and Design*, **86**. Pp 1024-1038. 2011
- [9]. D. Alves, et al., "Vessel thermal map real-time system for the JET tokamak". *Physical review special topics- accelerators and beams*, 15, 2012.

Signal	Range
Plasma density	$0 \leq \text{to} \leq 10^{19} \text{ m}^3$
Mode lock amplitude	$0 \leq \text{to} \leq 10^{-4} \text{ T}$
Plasma internal inductance	$-1 \leq \text{to} \leq 10^{-6} \text{ H}$
Poloidal beta	$-1 \leq \text{to} \leq 30$

Table I. Signal range

Campaign	Non-Intentional	Intentional	Non-Disruptive	Total
C15a	21	6	256	283
C15b	10	0	158	168
C16	3	0	10	13
C17	73	0	1032	1105
C18	14	13	302	329
C19	47	41	585	673
C20	28	12	703	743
C21	16	3	573	592
C22	34	3	451	488
C23	24	8	490	522
C24	14	12	362	388
C25	19	22	570	611
C26	58	49	1323	1430
C27a	43	10	320	373
C27b	70	59	513	642
Total	521	238	7648	8407

Table II. Campaign distribution

Element	Function and description
Core 1	Linux, MARTe services and IRQs
Core 2	ATM IRQ
Core 3	MARTe threads for receiving datagrams
Core 4	MARTe local timer thread
Core 5	MARTe real-time thread 1
Core 6	MARTe real-time thread 2

Table III. Core distribution

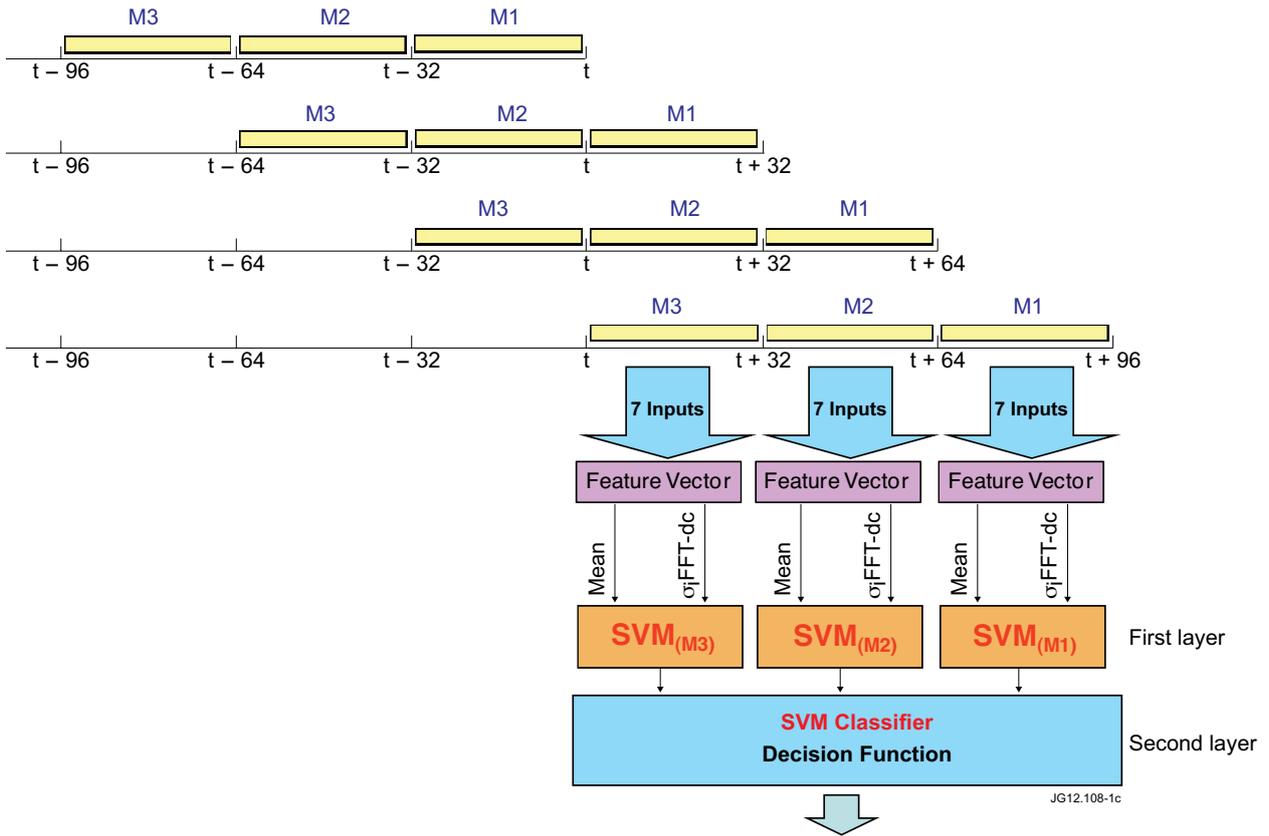


Figure 1: APODIS architecture. The first layer is made up of three SVM-RBF classifiers, M_x , and the second layer is a SVM-linear kernel classifier.

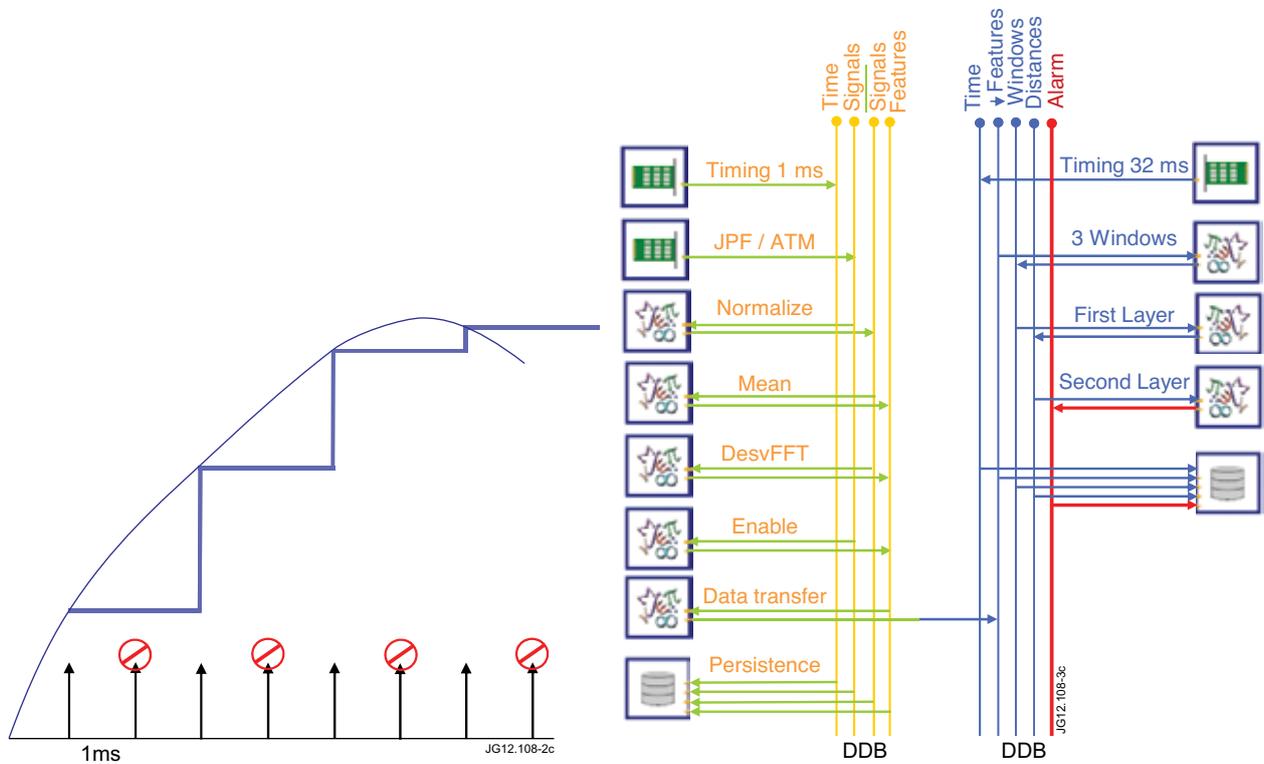


Figure 2: Hold effect in signal reading when no sample is available in the RTDN.

Figure 3: APODIS architecture using the MARTe framework.

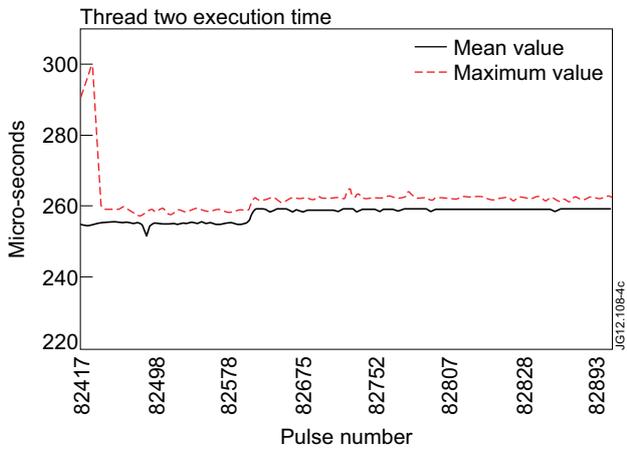


Figure 4: Thread two execution time versus discharge number. The maximum deviation of the maximum value versus the mean value is 46 us.

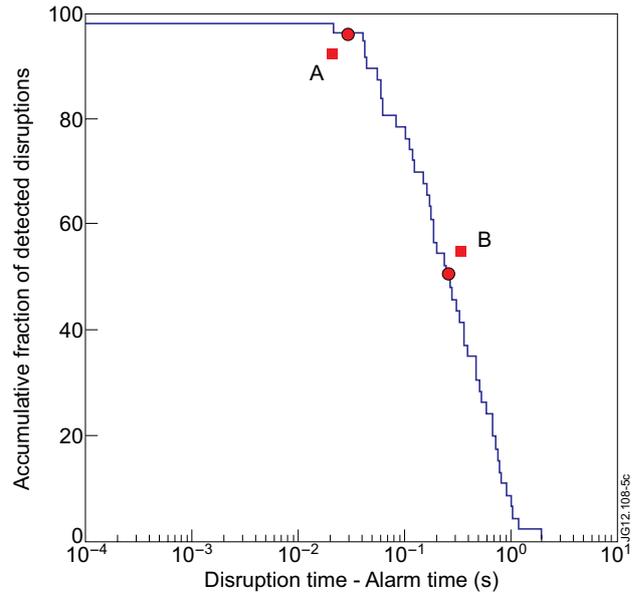


Figure 5: APODIS accumulative detected disruptions versus disruption time for Pulse No's: 82429 to 82905 in the C28 campaign.