20th Real Time Conference

5-10 June 2016
Padova, Italy

IEEE NPSS NUCLEAR & PLASMA SCIENCES SOCIETY

EUROfusion

Real-time implementation in JET of the SPAD disruption predictor using MARTe

S. Esquembri[1], J. Vega[2], A. Murari[3], M. Ruiz[1], E. Barrera[1], S. Dormido-Canto[4], R. Felton[5], M. Tsalas[5], D. Valcarcel[6], and JET Contributors*

[1]Instrumentation and Applied Acoustic Research Group (I2A2), Universidad Politécnica de Madrid, Madrid, Spain
[2]Laboratorio Nacional de Fusión, CIEMAT, Madrid, Spain
[3]Consorzio RFX (CNR, ENEA, INFN, Universitá di Padova, Acciaierie Venete SpA), Padova, Italy
[4]Dpto. Informática y Automática - UNED, Madrid, Spain
[5]EUROfusion Consortium, JET, Culham Science Centre, Abingdon, UK
*See the Appendix of F. Romanelli et al., Proceedings of the 25th IAEA Fusion Energy Conference 2014, Saint Petersburg, Russia.

Acknowledgements
This work was partially funded by the Spanish Ministry of Economy and Competitiveness under the Projects No ENE2012-38970-C04-01, ENE2012-38970-C04-03, ENE2012-38970-C04-04, predoctoral fellowship BES-2013-064875, and the grant for predoctoral short-term stays in R&D centers (2014)

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Outline

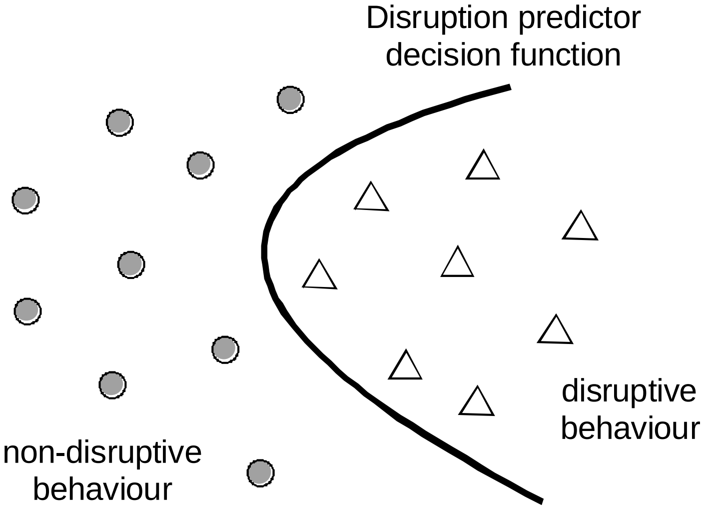
- Introduction
  - Disruption Predictors
  - MARTe & JET
- SPAD Disruption predictor
  - Algorithm
  - Real-time Implementation
- Results
- Conclusions
- Future Work
- Q&A

Introduction

- Context: Current (and future) Tokamak devices
- Plasma disruptions are unavoidable and damage the machine
  - Electro-Magnetic Forces
  - Thermal Loads
  - Runaway Electrons
- Mitigation techniques must be applied before the disruption occurs -> Disruption Predictors

Introduction: Disruption Predictors

- Several approaches already exist:
  - Threshold based predictor -> Limit set manually based on the risk
    - Using Locked Mode, Plasma Current, or Restraint Ring Loop
    - Optimal/Universal threshold?
  - APODIS -> Machine Learning (SVM).
    - Using **seven** signals (including Locked Mode and Plasma Current)
    - Need training based on disruptive and non-disruptive discharges
    - Machine dependant?
    - New/Upgraded machines?
    - Big changes in plasma parameters?
    - Difficult to find physics explanation justifying the detector classification



POLITÉCNICA | GOBIERNO DE ESPAÑA | MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD | Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas | JET | UNED | CONSORZIO RFX

Introduction: MARTe

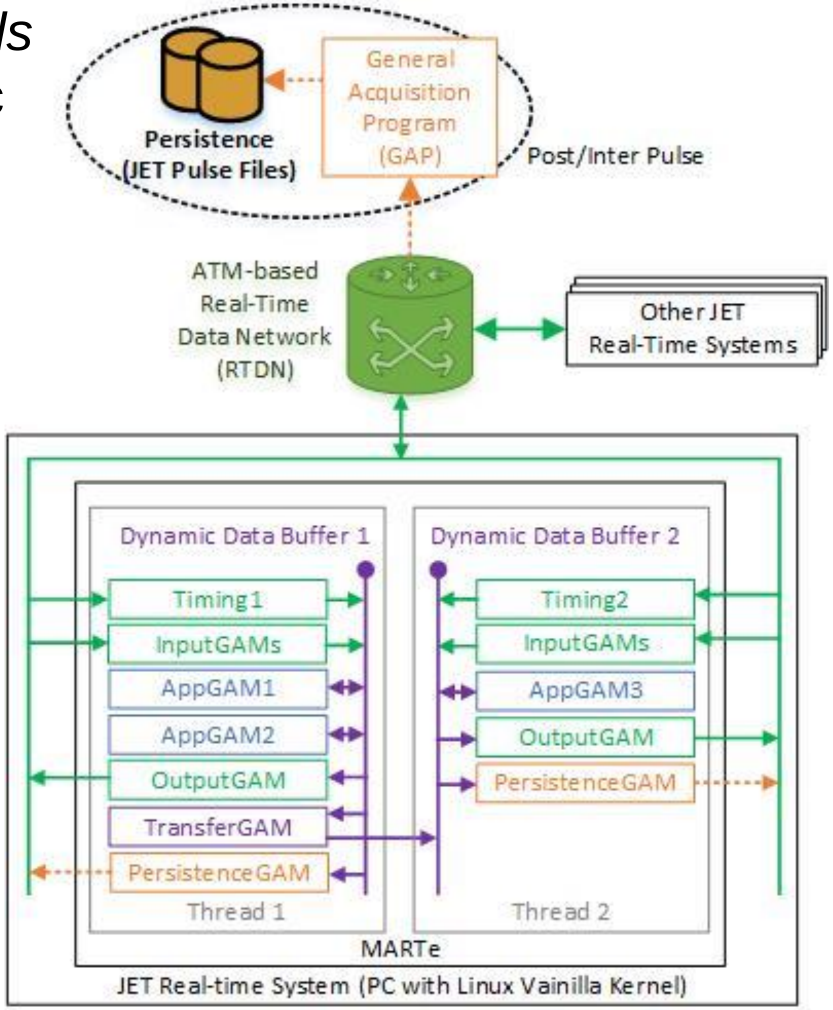
- MARTe : Multi-threaded Application Real-Time executor
  - Multiplatform framework for implementing real-time applications
  - Isolation
    - From PC hardware & OS
    - From system I/O software
  - Modularity

Code reuse + maintainability

- Already used in several tokamaks (RFX,COMPASS, ISTTOk, FTU)
  - Also in JET: Interface with JPF/RTDN already implemented

Introduction: MARTe

- MARTe applications consist of *Threads* periodically executing *GAMs* (Generic Application Modules)
- GAMs are interconnected using Dynamic Data Buffer (DDB)
- Typical GAMs:
  - Timing (Thread clock)
  - System I/O
    (JETs: Read/Write from/to RTDN/JPFs)
  - Process GAMs
  - Thread-to-Thread communication (DDB-to-DDB)



SPAD Disruption Predictor

- **SPAD**: **S**ingle signal **P**redictor based on **A**nomaly **D**etection

- Objectives:
  - No use of data from past discharges -> No training needed

- Method
  - Use Locked Mode signal -> Highly related with disruptions
  - Detect abnormal behavior in the signal
    - Normal behavior is learnt from the beginning of the discharge
  - Predict disruptions using only one signal
  - Easy to understand explanation for detection

SPAD Disruption Predictor

- Already presented
  - J. Vega *et al.* “Advanced disruption predictor based on the locked mode signal: application to JET”. 1st EPS Conference on Plasma Diagnostics. April 14-17, 2015. Book of abstracts. Frascati, Italy.
  - J. Vega *et al.* “Disruption Precursor Detection: Combining the Time and Frequency Domains”. 26th Symposium on Fusion Engineering (SOFE 2015). May 31st-June 4th, 2015.Austin (TX), USA.
  - J. Vega *et al.* “Real-time anomaly detection for disruption prediction: the JET case“. Nuclear Fusion (submitted).

- Implemented in MATLAB and not suitable for real-time.

SPAD: Algorithm

$$LockedMode_{signal}[k] = LM[k]$$

Sliding Windows

$$W[k] = (LM[k-31], \dots, LM[k])$$

Haar Wavelet Transform

$$H_{L2}(W[k]) = (\; H_{App}[k] \;,\; H_{Det}[k] \;) = ((H_{App\,0}[k], \dots, H_{App\,7}[k]), (H_{Det\,0}[k], \dots, H_{Det\,23}[k]))$$

Mahalanobis Distance

$$D_M(H_{App}[k], H_{App}[0..k-1]) = D_M[k]$$

Outlier Factor Function

$$O_F(D_M[k], D_M[0..k-1]) = O_F[k]$$

Alarm Criteria

$$SPAD_{Alarm}(O_F[k], LM[k]) = SPAD_{Alarm}[k] \in \{0,1\}$$

SPAD: Real-time Implementation

- Each step is performed in a different GAM
  - + GAM to signal start based on Plasma current threshold
- One single thread executing all the GAMS
- Thread execution period = 1ms
- Problems
  - Covariance Matrix
  - Standard Deviation
  - Mean

Of a set growing with each iteration

~~Execution time not bounded~~

"Update" the values, no recalculate

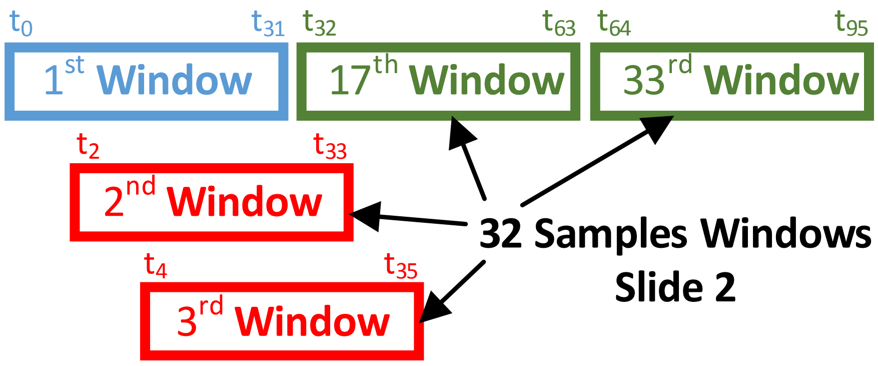
SPAD: Real-time Implementation
SlidingWindowGAM

- Provide the latest 32 values (window) of Locked Mode signal
- Update the window every 2 thread cycles (2 ms) (slide)
- Signals when the window has been updated
- Parameters:
  - Input signal
  - Window size
  - Window slide

$$LockedMode_{signal}\,[k] = LM[k]$$

Sliding Windows

$$W[k] = (LM[k-31], \dots, LM[k])$$
$$W[k+1] = W[k]$$
$$W[k+2] = (LM[k-29], \dots, LM[k+2])$$



SPAD: Real-time Implementation
HaarApp1DCoefGAM

- Calculate the Haar Wavelet Transform Approximation Coefficients (for SPAD Level 2 $H_{L2}: \mathbb{R}^{32} \rightarrow \mathbb{R}^{8}$)
- Optimizations
  - Coefficients not calculated by iteration
- Parameters:
  - Input signal
  - Level applied

$$W[k] = (LM[k-31], \dots, LM[k])$$

Haar Wavelet Transform

$$H_{L2}(W[k]) = (H_{App}[k], H_{Det}[k]) = ((H_{App\,0}[k], \dots, H_{App\,7}[k]), (H_{Det\,0}[k], \dots, H_{Det\,23}[k]))$$

$$(x_0, \dots, x_{31})$$

$$H_{App\,1,i} = \sqrt{2}(x_{2i} + x_{2i+1})/2$$

Level 1 $(Happ_{1,0}, Happ_{1,1}, \dots, Happ_{1,15})$

$$H_{App\,2,i} = \sqrt{2}(H_{App\,1,2i} + H_{App\,1,2i+1})/2$$

Level 2 $(Happ_{2,0}, Happ_{2,1}, \dots, Happ_{2,7})$

$$H_{App\,X,Y} = (\sqrt{2})^{X\%2} \sum_{i=Y2^X}^{(Y+1)2^X} x_i \Big/ 2^{\left\lfloor \frac{X}{2} \right\rfloor}$$

SPAD: Real-time Implementation

MahalanobisGAM

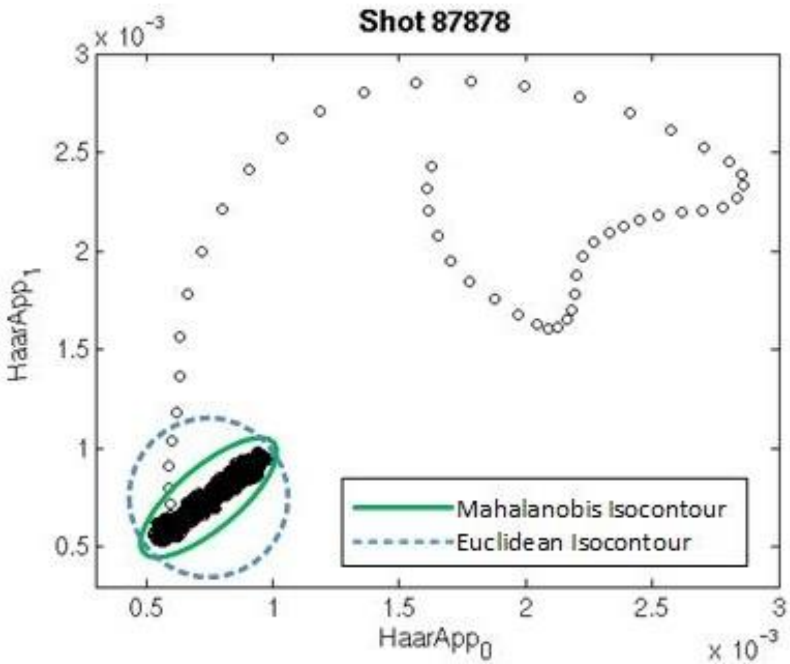
- H_{App} used as feature vector
- Calculate distance between current vector and centroid of the cluster formed by all past vectors in the discharge.
- Optimizations
  - Covariance Matrix (S) and mean vector (μ) "updated"
    - Store partial sums and products
  - LUP method used for inverse product
    - Can be done better!

$$H_{L2App}(W[k]) = (H_{App\,0}[k], \ldots, H_{App\,7}[k])$$

Mahalanobis Distance

$$D_M(H_{App}[k], H_{App}[0..k-1]) = D_M[k]$$

$$D_M(x, X) = \sqrt{(x - \mu_X)^T S_X^{-1} (x - \mu_X)}$$



SPAD: Real-time Implementation
OutlierFactorGAM

- Is the current vector anomalously far from the cluster?
- Factor given by an equation using mean(μ) and standard deviation(σ) from all past distances in the discharge.
- Optimizations
  - Update mean and standard deviation storing partial sums and products

$$D_M(H_{App}[k], H_{App}[0..k-1]) = D_M[k]$$

↓ Outlier Factor Function

$$O_F(D_M[k], D_M[0..k-1]) = O_F[k]$$

$$O_F[k] = \left|\frac{D_M[k] - \mu(D_M[0..k])}{\sigma(D_M[0..k])}\right|$$

SPAD: Real-time Implementation
SPADAlarmGAM

- Trigger the disruption alarm if the conditions are met
  - Outlier factor surpasses a configurable threshold
  - Current value of Locked Mode is a global maxima
- Analysis over all ILW campaigns shows that K=10 works fine
- Parameters
  - Outlier Factor Threshold

$$O_F(D_M[k], D_M[0..k-1]) = O_F[k]$$

↓ Alarm Criteria

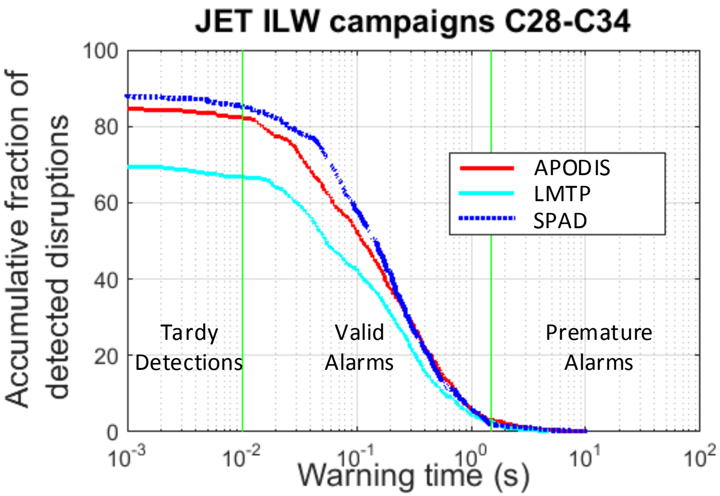
$$SPAD_{Alarm}(O_F[k], LM[k]) = SPAD_{Alarm}[k] \in \{0,1\}$$

$$SPAD_{Alarm}[k] = (LM[k] == \max(LM[0..k]))\ \&\&\ (O_F[k] > K)$$

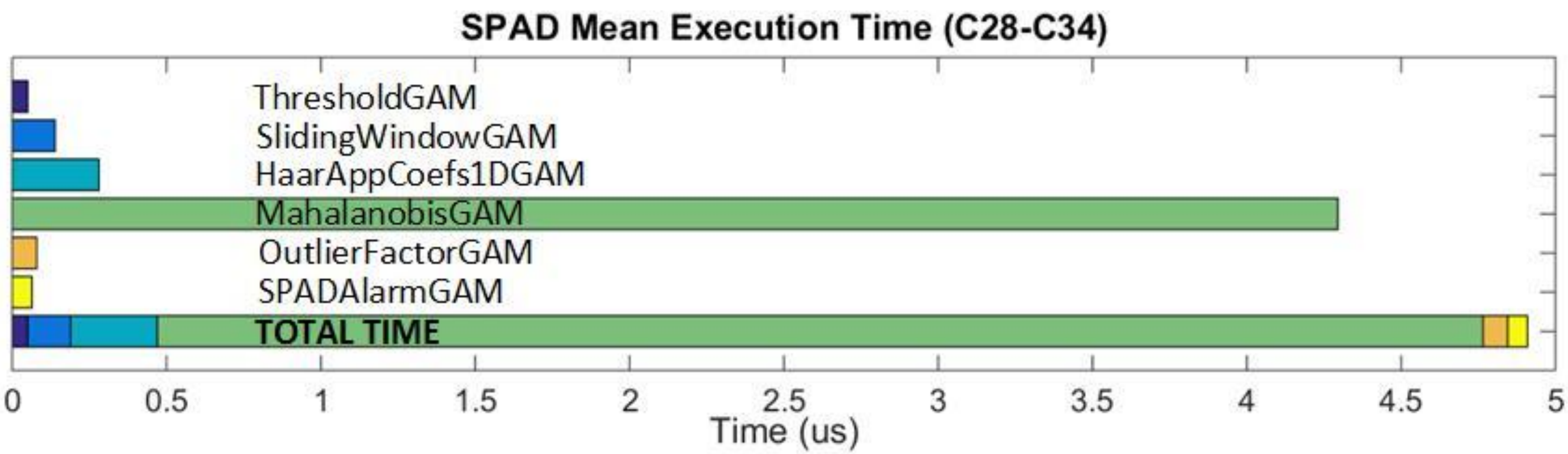
Results

- Same detection results than non-optimized MatLAB reference application.
- Execution time bounded during the whole discharge.
- Tested using JETs JPF data from all safe and unintentional disruptive discharges with ILW from 2011 to 2014 (C28-C34)
  - 1738 safe discharges
  - 566 unintentional disruptive discharges

Predictor	False Alarms	Missed Alarms	Tardy Detections	Valid Alarms	Premature Alarms
SPAD	7.42%	10.60%	3.18%	83.57%	2.65%
APODIS	<5%	15.38%	2.47%	79.15%	3.00%
LMPT	---	30.39%	3.00%	63.69%	2.65%



Results



- Execution time in a **no-Real Time** platform
- 4 cores, 2 threads/core, 3.6 GHz, 16 GB RAM, RHEL 6.5
- Confidence Level 97% (Mean Time + 3·Standard Deviation)
- Max execution time 26.9280 us (OS interruptions?)

Conclusions

- Implemented real-time disruption predictor in MARTe
- Execution time per cycle << 1ms (even in no-RT platforms)
- Optimizations for the calculation of Covariance Matrix, Standard Deviation and mean
  - Execution time bounded for the whole discharge
  - Depends only on the size of the feature vector (not on the duration of the discharge!!)
- Identical detection results for MatLAB reference application and MARTe implemented predictor.

Future Work

- Predictor:
  - Reduce false alarms
  - Improve detection ratio
  - Study behavior with other/more signals
    - Related with root of disruption

- Implementation
  - Real-time implementation of the improvements in the predictor
  - Improve inverse matrix and vector-matrix-vector product performance (Not really important due to current performance)

Q&A

Thanks for your attention