REAL-TIME PREDICTION AND AVOIDANCE OF FUSION PLASMAS INSTABILITIES USING FEEDBACK CONTROL

Egemen Kolemen
Princeton University / PPPL

With:

Post-Docs: Aza Jalalvand, Leonel Palacios, Anthony Xing

Graduate students: Joe Abbate, Rory Conlin, Yichen Fu, Oak Nelson, Ricardo Shousha

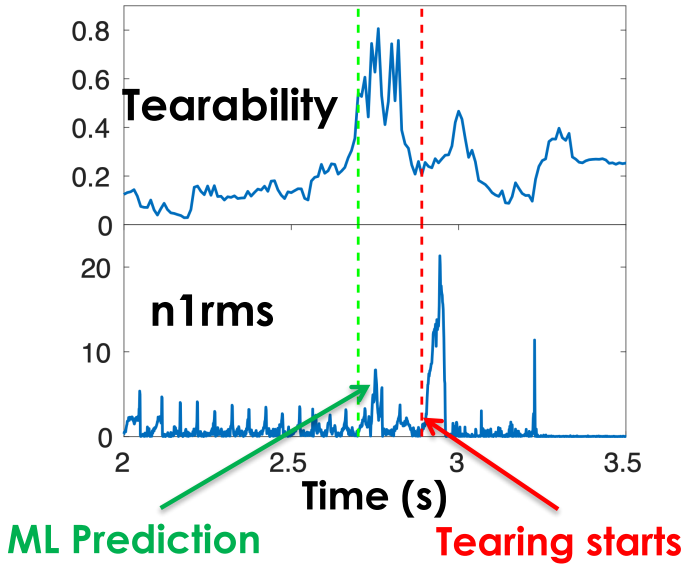
Undergraduate students: Jalal Butt, Aaron Wu, Milan Wolff
Engineer: Keith Erikson

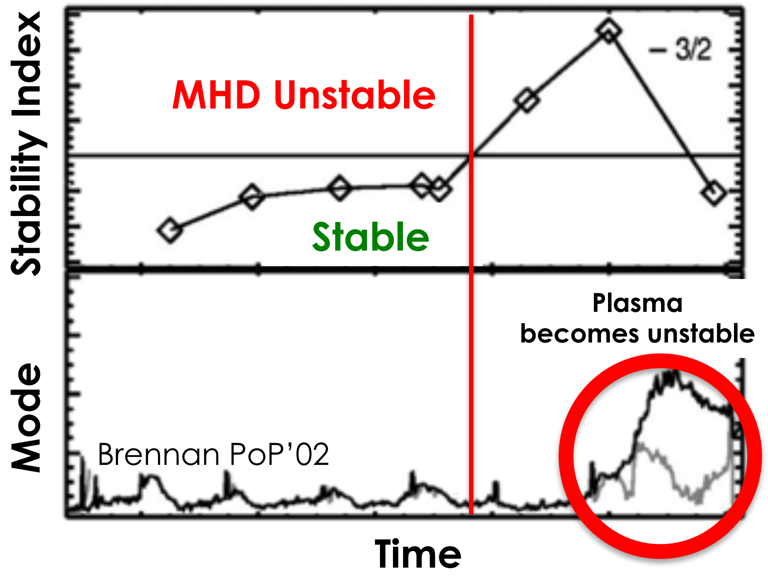
July 2020

Big Updates for 2020: For the people who know our research already

1. ML-based plasma evolution predictor/controller running on DIII-D
2. Real-time kinetic EFIT is functional and being tested on DIII-D
3. Keras2c ➔ Automatic NN to PCS code conversion functional
4. Real-time δW running at ~200 ms at DIII-D: Offline tests projecting to ~20 ms is using GPU, RT-Δ' in development
5. Dynamic Mode Decomposition gives good plasma evolution models
6. Big Highlight: RT-Adaptive ML proof-of-concept shown using reservoir learning. So fast ~20 ms that for ML profile predictor/ controller, we can update the ML online as new data comes in

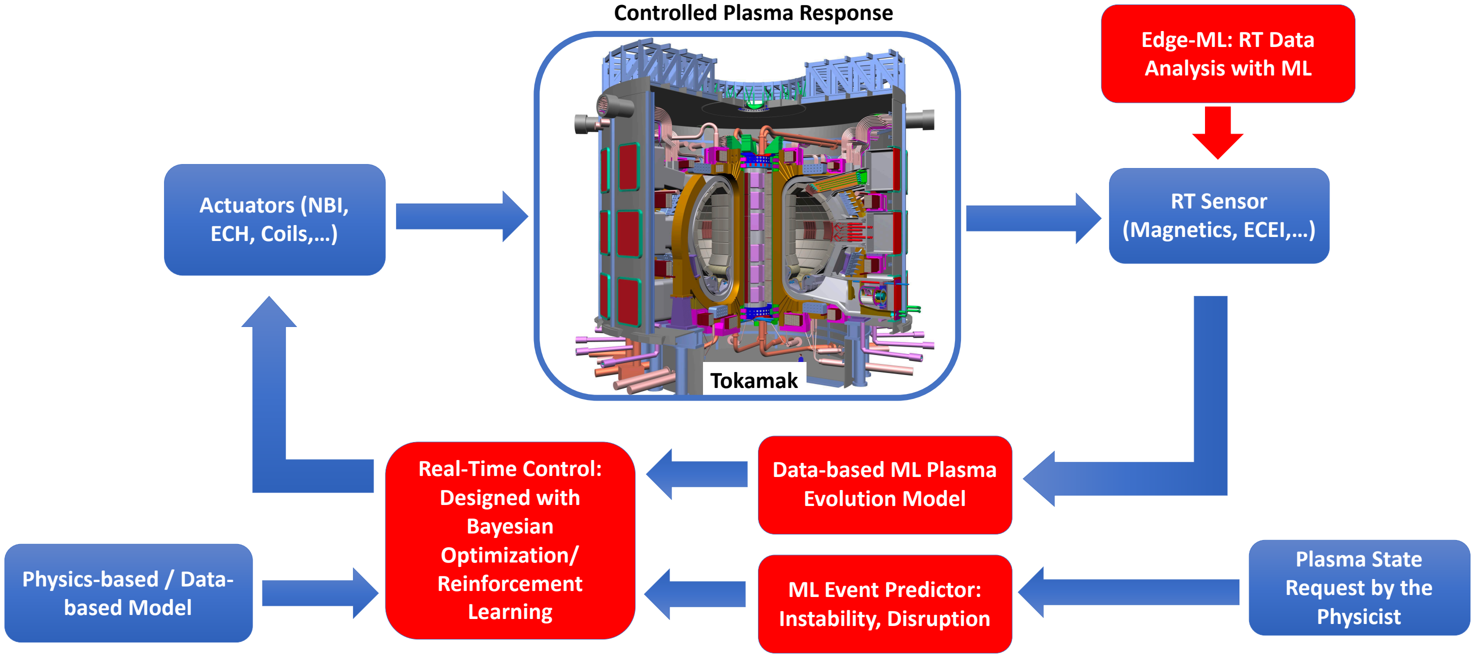
Portfolio Approach to Disruption Avoidance: 1st Physics-Based + Data-Based RT Stability Analysis (Event Prediction)





- **Physics-based approach are useful for predicting instabilities**
  - **Good:** Machine independent; Gives insight and possible solutions
  - **Bad:** May need long analysis (not for real-time application) and may not predict all phenomena yet
- **Data-based based (Machine-Learning {ML})approach:**
  - Good predictions, harder to project and understand the reason
- **We need a solutions portfolio to avoid disruptions at ITER and beyond**

Portfolio Approach to Disruption Avoidance: 2nd RT Feedback Control with ML


Controlled Plasma Response
Edge-ML: RT Data Analysis with ML
Actuators (NBI, ECH, Coils,…)
Tokamak
RT Sensor (Magnetics, ECEI,…)
Real-Time Control: Designed with Bayesian Optimization/ Reinforcement Learning
Data-based ML Plasma Evolution Model
Physics-based / Data-based Model
ML Event Predictor: Instability, Disruption
Plasma State Request by the Physicist

- The only point of RT-prediction is to take action ➔ Control
- DOE Fusion Energy Science report on “ML for Fusion” explains the path for ML for fusion control
- Event prediction + Evolution prediction ➔ ML Control

1. Automated Plasma Equilibrium from Diagnostics

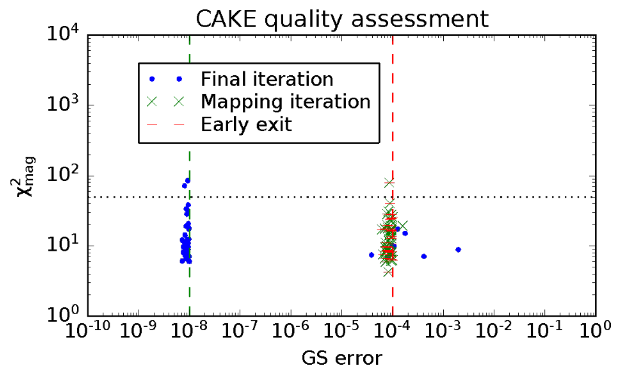
2. Tearing and Disruption Prediction

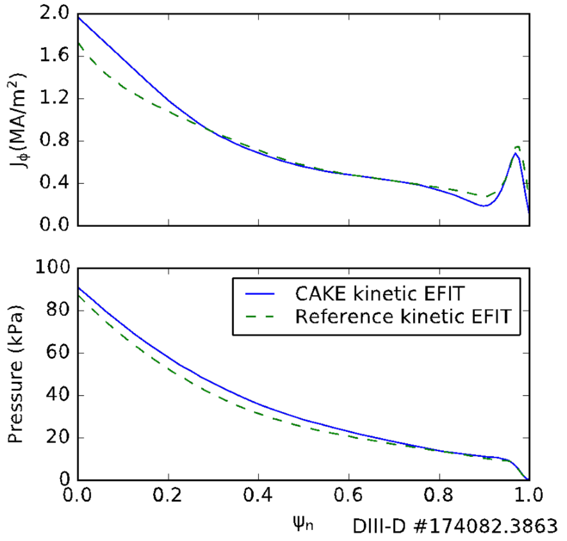
3. Machine Learning Control for Disruption Avoidance

4. RT Adapting ML Prediction and Control

Automatic kinetic EFIT workflow robustly generates quality equilibria

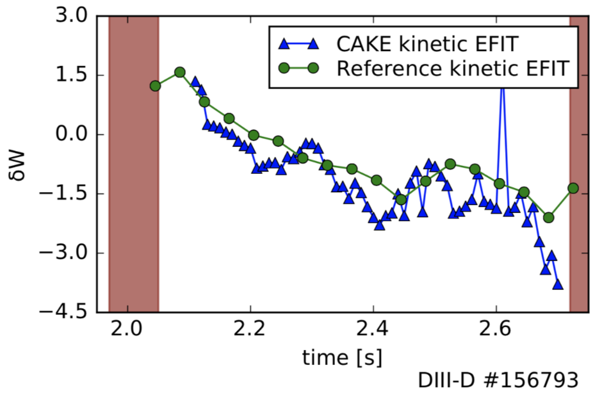
- The Consistent Automatic Kinetic Equilibria(CAKE) tool has been developed to produce kinetically constrained reconstructions with minimal human intervention



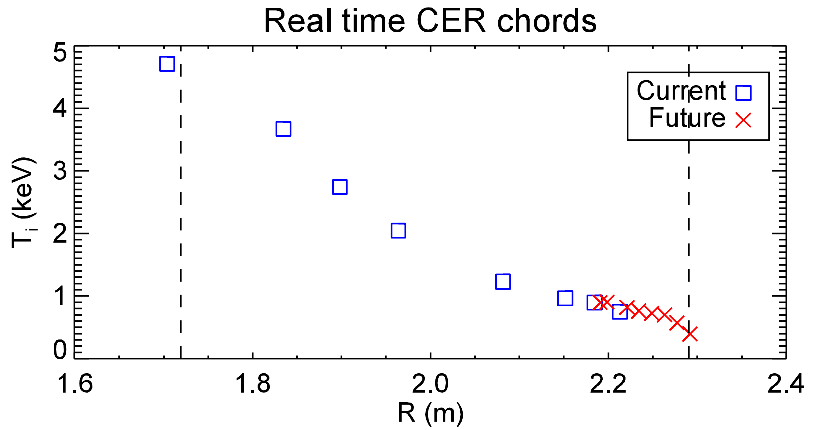


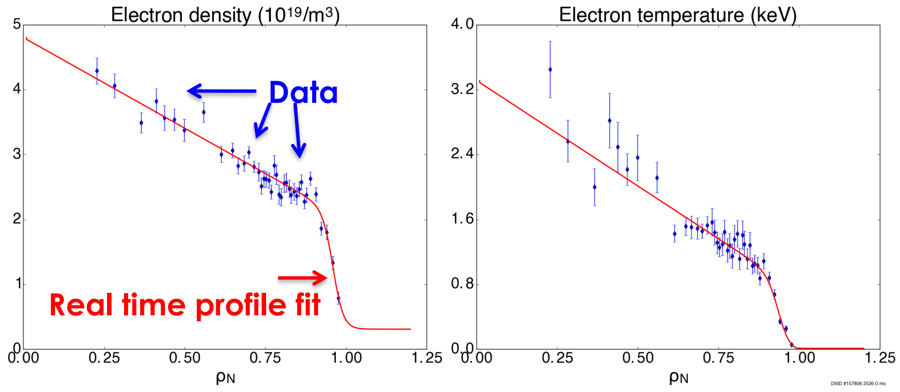
- CAKE has been able to robustly generate low force balance error equilibria that compares well with manual kinetic equilibrium.

- Preliminary work with DCON and STRIDE show CAKE outputs compares well in ideal MHD stability calculations.



Electron and Ions: RT-Thomson and RT-CER constraints on the Current and Pressure (R. Shousa, D. Kaplan, D. Piglowski)





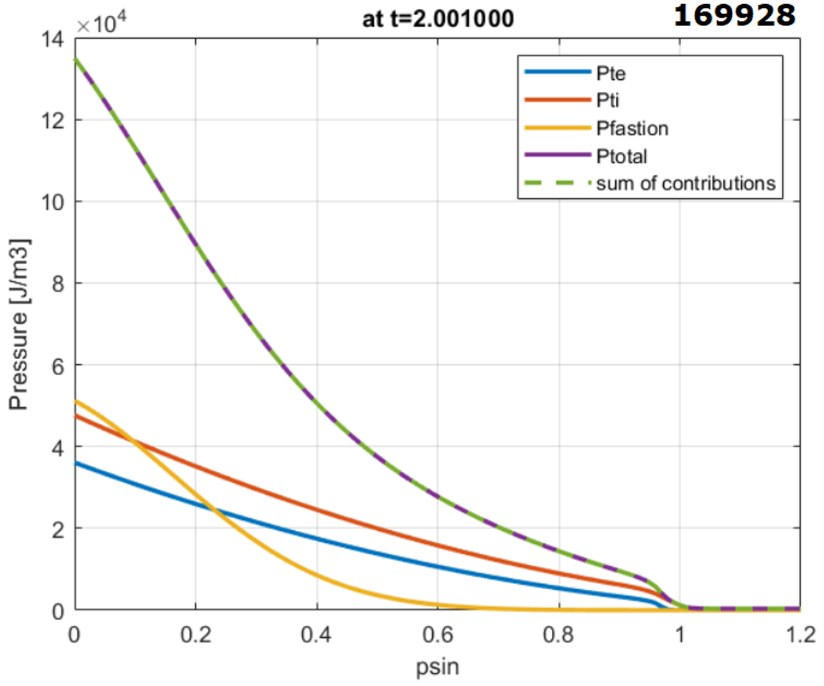
□ **Core CER channels already acquired**

× **Edge CER chords for pedestal are added**

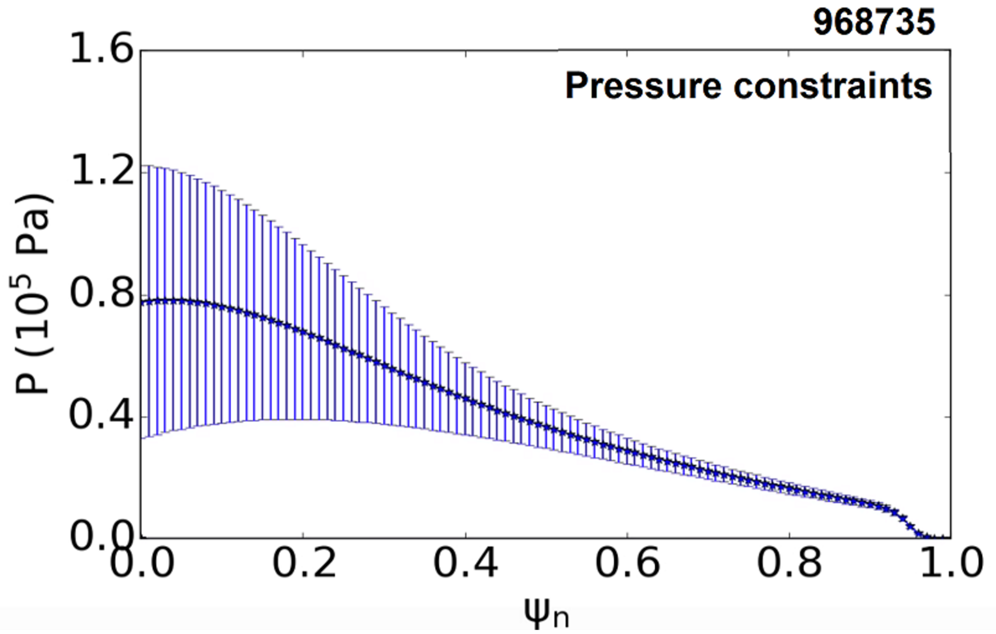
- **RT-CER server/system system working. RT analysis running on DIII-D**
- **Thomson analysis + fitting**
- **Fast Ions – NN based prediction**

Real-time Kinetic EFIT with new CER, Thomson (+MSE) constraints is running on DIII-D

Example of pressure constraint contributions: Electron, Ion, Fast Ion



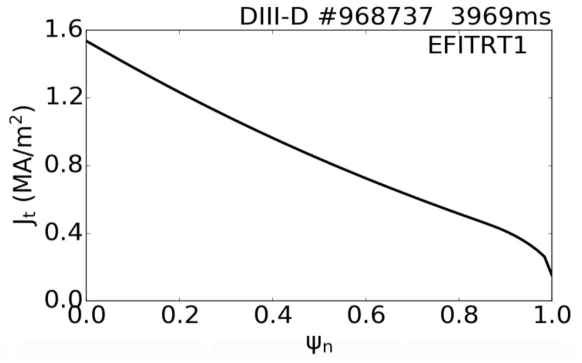
Example of pressure constraints used in equilibrium reconstruction

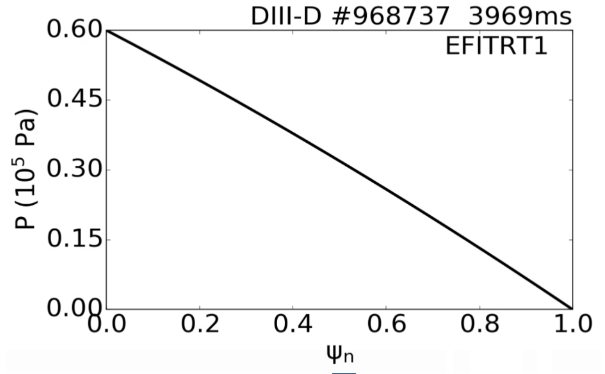


R. Shousha with K. Erickson
F. Laggner, Z. Xing, J. Ferron

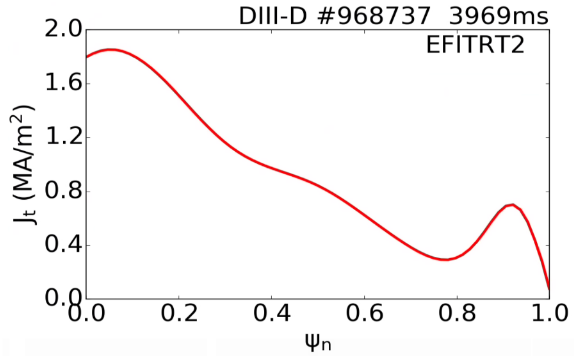
Real-time Kinetic EFIT with new CER, Thomson (+MSE) constraints is running on DIII-D

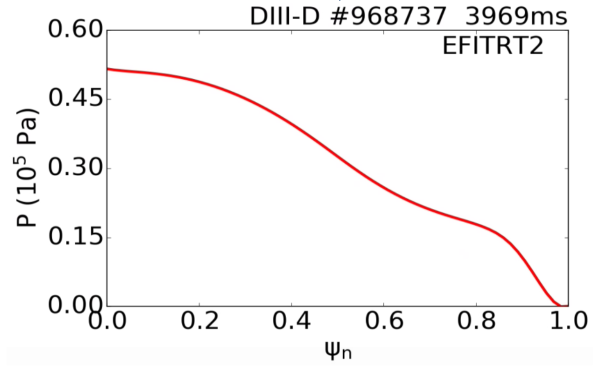
EFITRT1





New Real-time MSE+CER+ Thomson EFITRT2





R. Shousha with K. Erickson
F. Laggner, Z. Xing, J. Ferron

1. Automated Plasma Equilibrium from Diagnostics

2. Tearing and Disruption Prediction

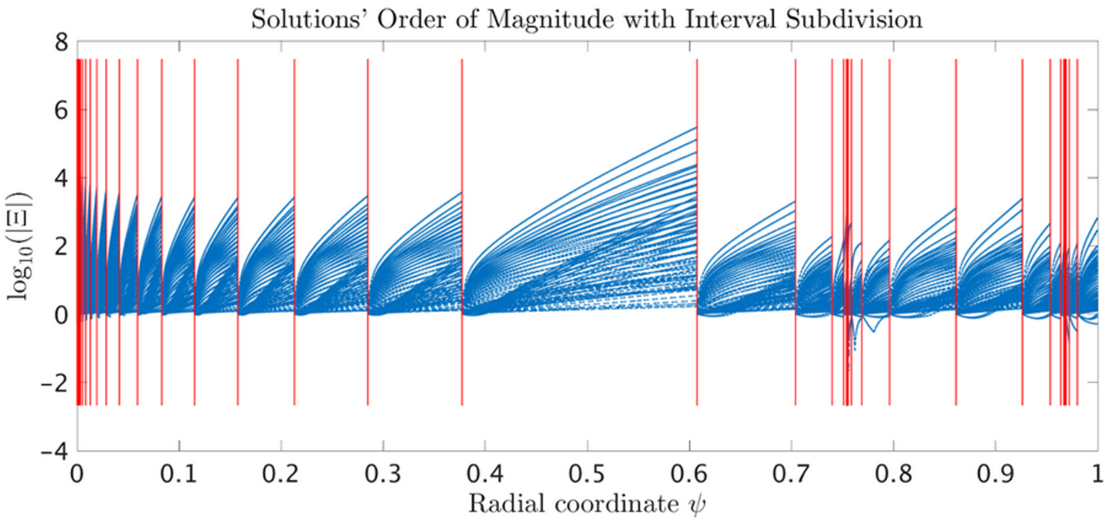
3. Machine Learning Control for Disruption Avoidance

4. RT Adapting ML Prediction and Control

STRIDE: Real Time δW Calculations (A. Glasser, R. Conlin)

$$\delta W = \frac{1}{2}\int_\Omega \mathrm{d}\mathbf{x}\left[Q^2 + \mathbf{J}\cdot\boldsymbol{\xi}\times\mathbf{Q} + (\boldsymbol{\xi}\cdot\boldsymbol{\nabla}P)(\boldsymbol{\nabla}\cdot\boldsymbol{\xi}) + \gamma P(\boldsymbol{\nabla}\cdot\boldsymbol{\xi})^2\right]$$

- Quadratic Lagrangian gives Linear Euler-Lagrange equation
- Linear E-L can be domain decomposed using state transition matrices

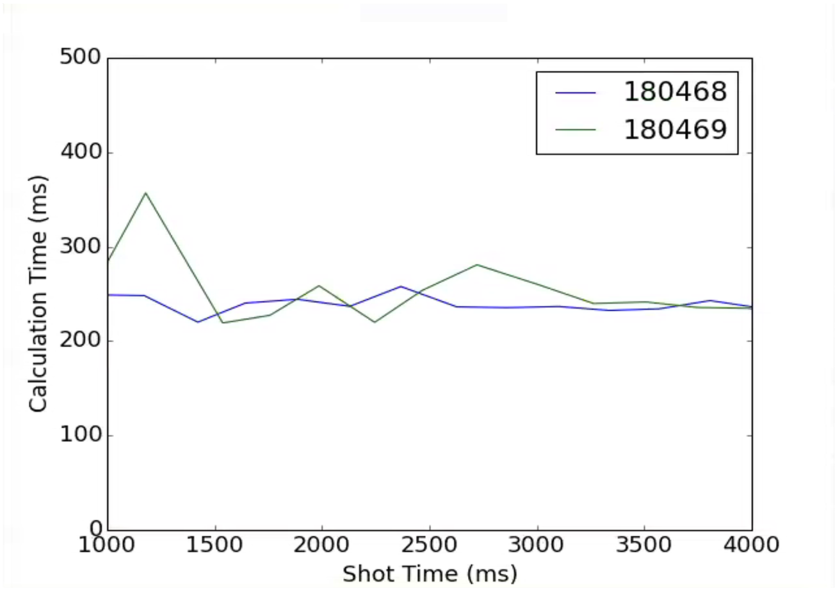


$$\mathbf{x}'(\psi) = \mathbf{L}(\psi)\mathbf{x}(\psi)$$
$$\boldsymbol{\Phi}'(\psi) = \mathbf{L}(\psi)\boldsymbol{\Phi}(\psi)$$
$$\mathbf{x}(\psi_2) = \boldsymbol{\Phi}(\psi_2,\psi_0)\mathbf{x}(\psi_0) = \boldsymbol{\Phi}(\psi_2,\psi_1)\boldsymbol{\Phi}(\psi_1,\psi_0)\mathbf{x}(\psi_0)$$

Easy parallelization → fast stability calculations

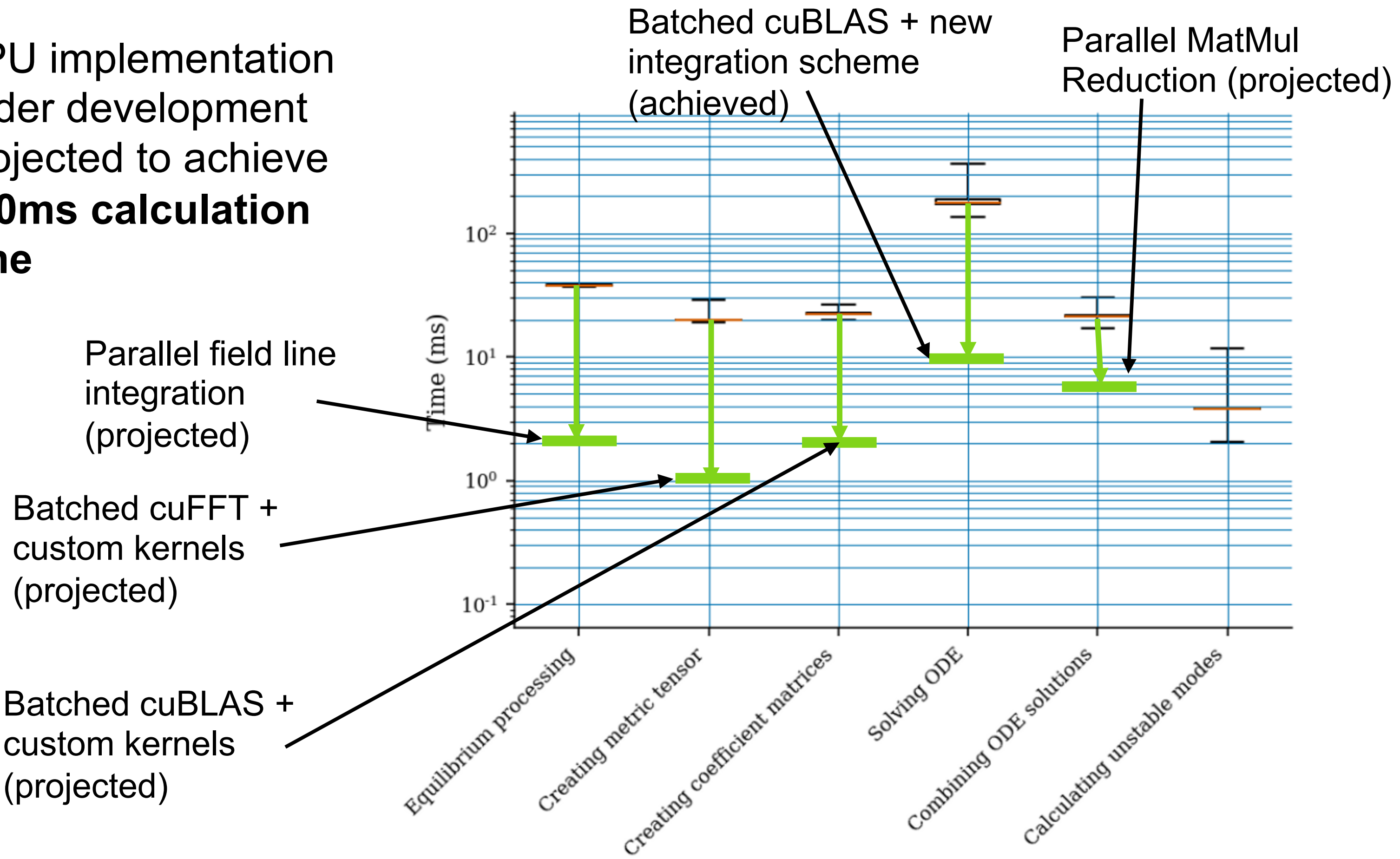
STRIDE: Real Time δW Calculations (A. Glasser, R. Conlin)

- Parallel calculation integrated into plasma control system (PCS) on 72 core CPU
- Tested on DIII-D, Sept 2019 & June 2020
- Achieved calculation times **~250ms** during steady state
- Further optimization to achieve **<200ms**
- Will be used for control Summer 2020



STRIDE GPU (R. Conlin)

- GPU implementation under development
- Projected to achieve **~20ms calculation time**

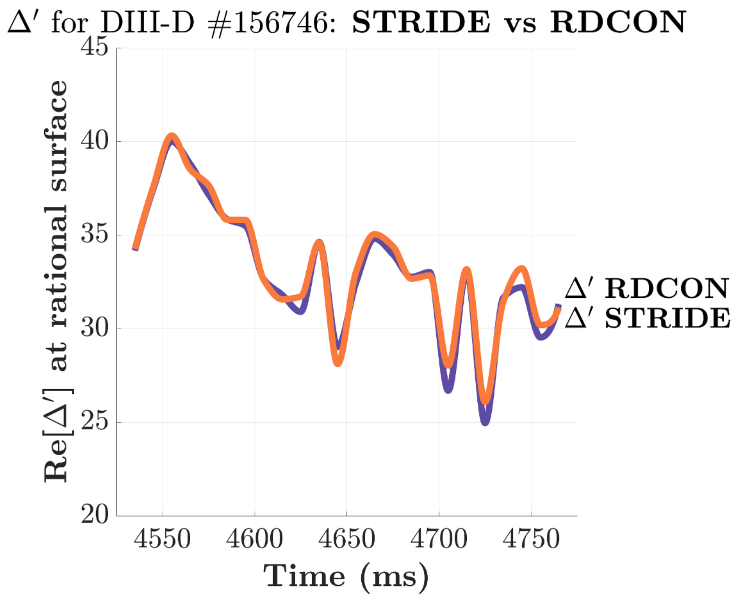


STRIDE: Delta-Prime Calculation (A. Glasser, R. Conlin)

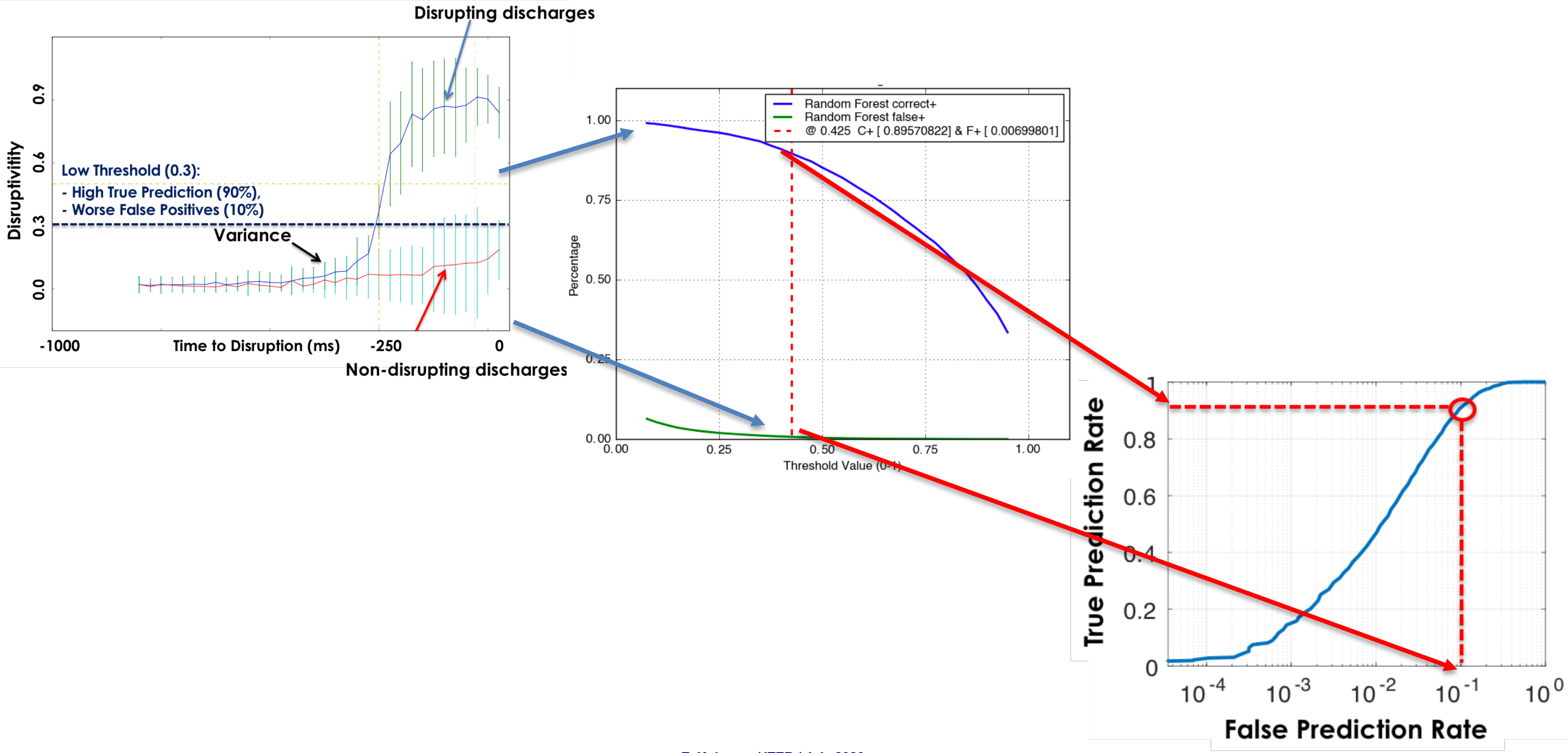
- E-L solution for δW (ideal stability) allow easy calculation of **Δ**' (resistive stability)
- Directly solves BVP, no Galerkin projection / FEM
  - Robust, converges for all equilibria
- Extremely fast: <100ms (CPU only)
- Planned to incorporate into real time code summer 2020

$\Delta' = (\psi'_+ - \psi'_-)/\psi$ ➔ **Find** (ψ, ψ') *or* (p, q) **on left/right singular surface**

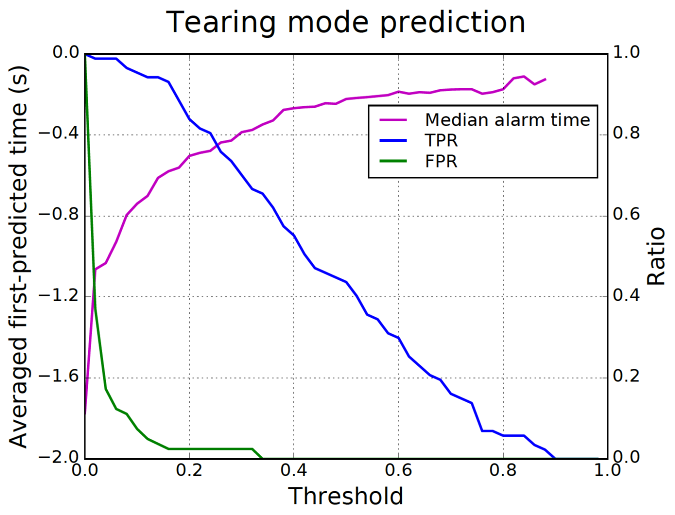
$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \mathbf{A}(\psi) \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix} \text{ with } \mathbf{q}(0) = 0,\ \mathbf{p}(1) = \mathbf{W}_{\text{vacuum}}\,\mathbf{q}(1)$$

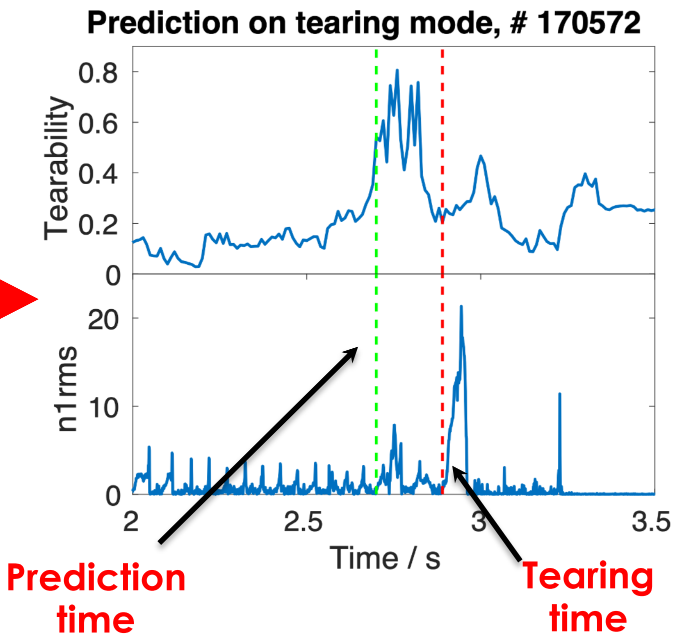


Disruptivity Threshold Gives a Choice Between Prediction Rate and False Positives



Tearibility: Sufficiently Early Tearing Modes Prediction May Allow Prevention





- Shot based prediction of instabilities and prediction time.
- Bagging:~%90/3% for 600 ms. %65/0 360 ms
- Tested Various ML and NNs

Tearability, predict tearing mode 250ms before it happen. Enough time to control and avoid tearing

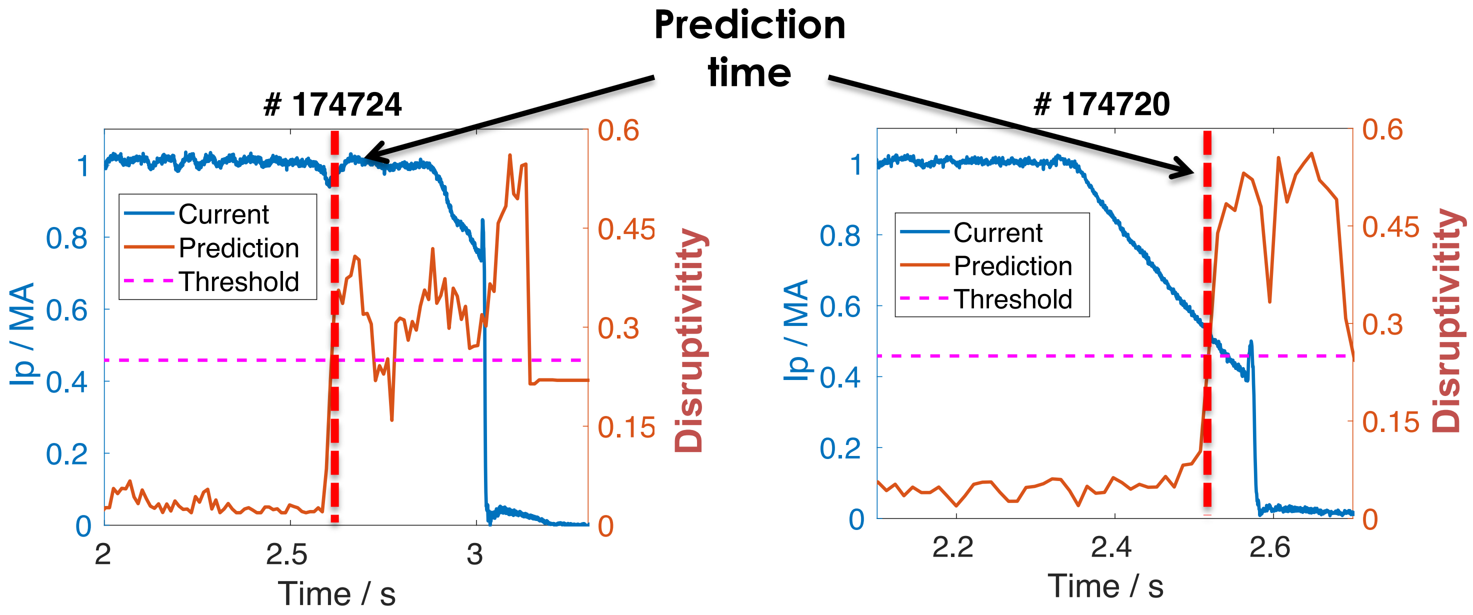
1. Automated Plasma Equilibrium from Diagnostics

2. Tearing and Disruption Prediction

3. Machine Learning Control for Disruption Avoidance

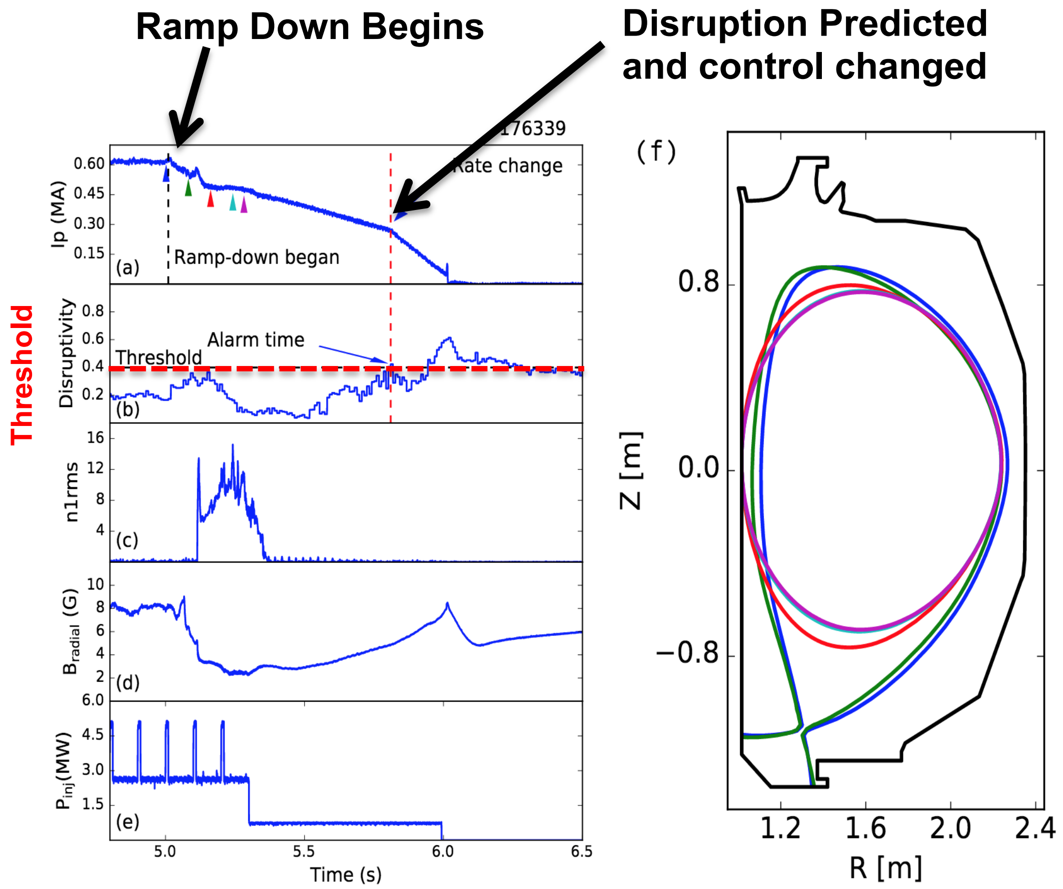
4. RT Adapting ML Prediction and Control

Real-time Disruption Prediction using Machine Learning (Kliejwegt, Fu)



- **Developed and tested Machine Learning Algorithms to find the disruption time**
- **First tests are shown above. Correctly predicts flattop disruption with ~300 ms warning and the ramp down disruption with >50 ms warning**

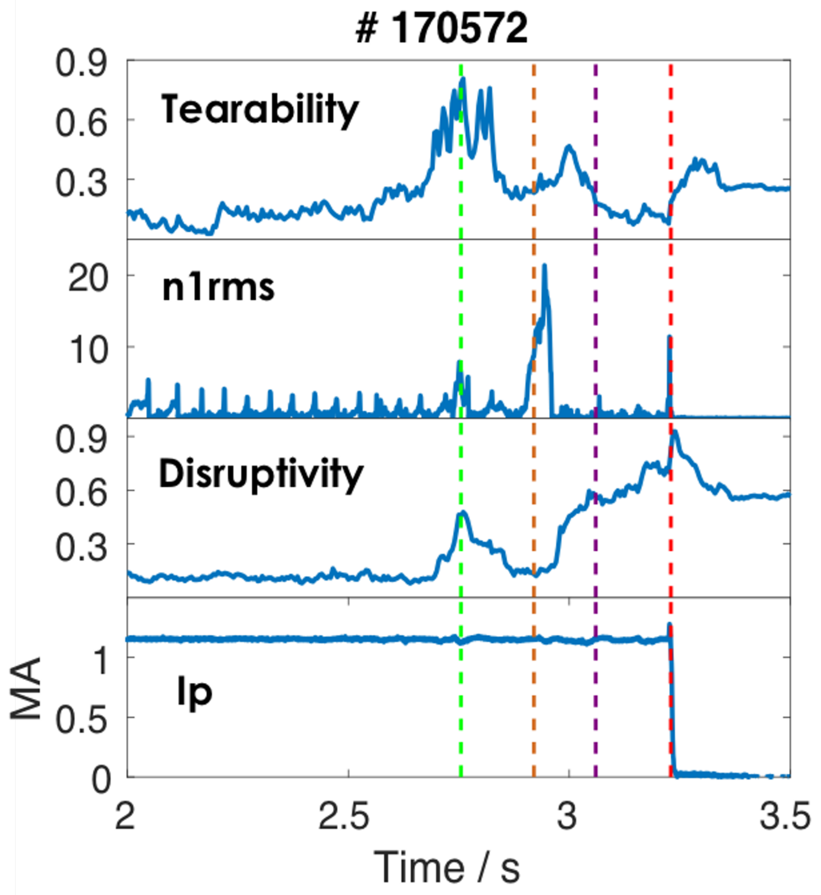
First Feedforward Control (Rampdown Scenario Change), Based on RT Disruption Prediction Tested on DIII-D (Fu, Barr)



- Developed and tested ML algorithm to find the disruption time.
- **Use it to change the off-normal response:**

➔ New ramp down sequence of the plasma (feedforward) when disruption is expected

Disruption Avoidance Can be Achieved by Predicting Instabilities with ML



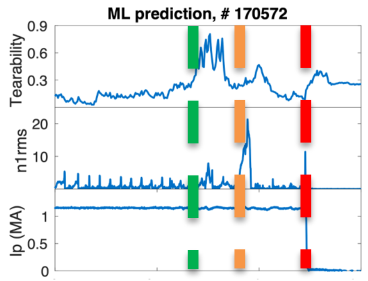
1. Predict instability (tearing mode) with ML

2. Instability (tearing) occurs

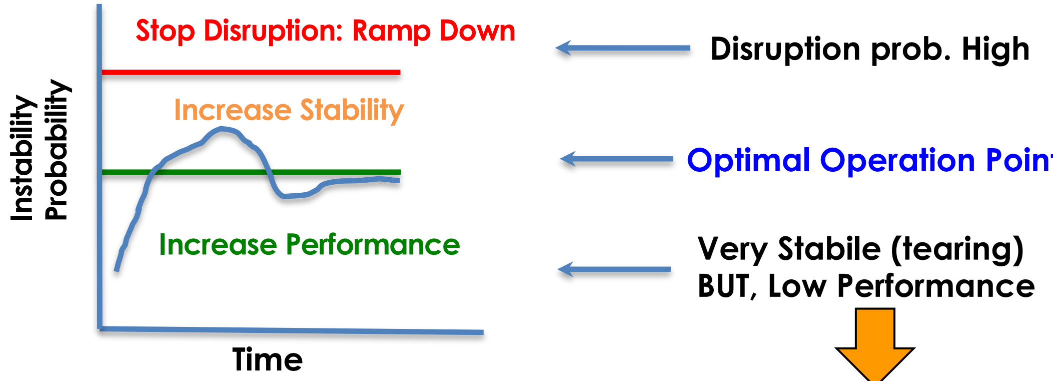
3. Predict disruption with ML

4. Disruption occurs

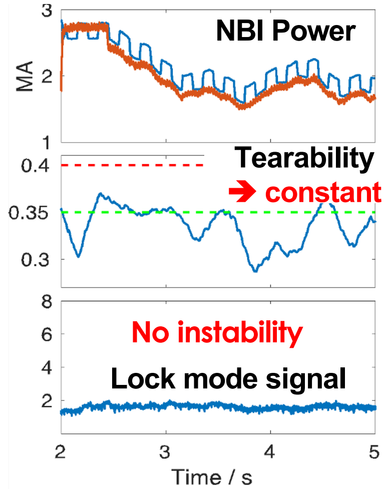
Machine Learning Control at DIII-D: ML Predict Stability (Tearing), Optimize Performance



1. Predict Instability using ML
2. Instability Starts
3. Disruption Occurs



Beam Power Controlled with ML



- Instability prediction gives lots more time than disruption prediction
- Enough time to control the plasma and avoid shutdown
- Choose a stability level to operate at (say %1)
- Then Machine Learning Controls the NBI/ECH for highest performance at that stability level
- PCS Algorithm (Kolemen, Fu, Boyer, Erickson)
- Fu et al. PoP 2020 (Scilight - Highlighted paper)

Neural Network to predict plasma transport (J. Abbate, R. Conlin)

- Given state (temperature, density, rotation, etc) at time t, and actuator inputs (NBI, Ip, gas) predict state an energy confinement time into future

Model Architecture

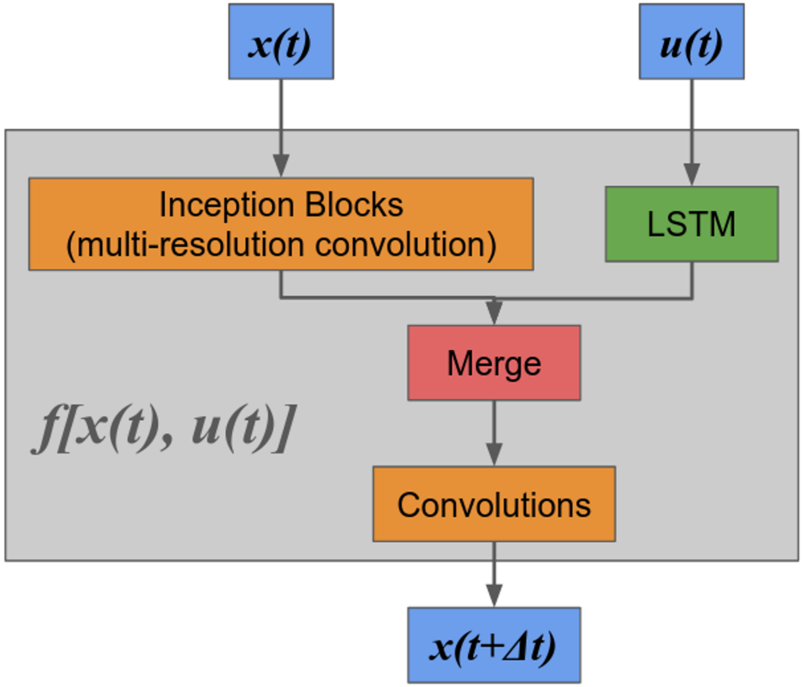
- Convolutional layers to capture gradients of profiles for transport (cf. natural response) *(Szegedy, 2015)*
- Recurrent layers to capture time history of actuators (cf. forced response) *(Gers, 1999)*

***x(t)* :**

- T_e
- n_e
- q
- Ω
- P
- *Shape*

***u(t)* :**

- $P_{injected}$
- $T_{injected}$
- I_p
- $\langle n_e \rangle_{target}$



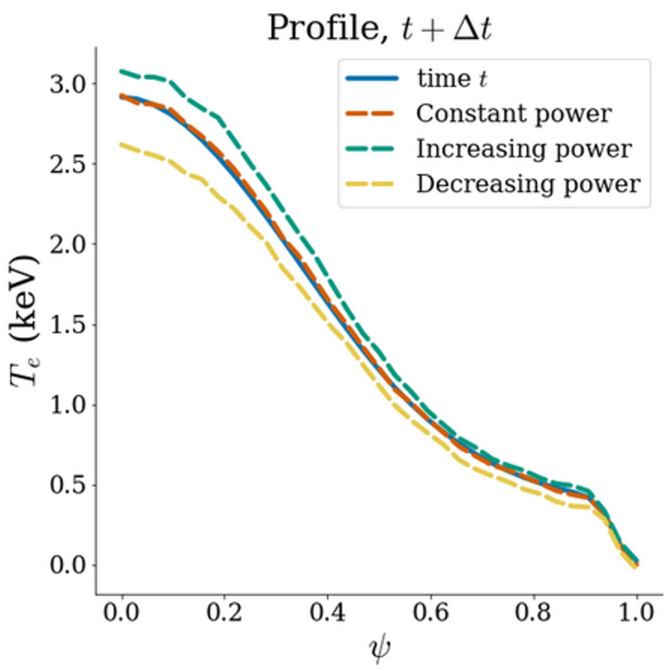
(Abbate, Conlin, NF, submitted)

Using NN Profile Predictor for Control (J. Abbate, R. Conlin)

- Real time predictions allows predictive control
- Simulate different actions in real time
- Take the action to minimize cost function:

$$\mathbf{u}^*(t) = \underset{\mathbf{u}(t)\in U}{\operatorname{argmin}} \left\| \mathbf{w} \cdot \left[\mathbf{x}^{targ}(t+\Delta t) - \mathbf{x}^{pred}\left(t, \mathbf{x}(t), \mathbf{u}(t)\right) \right] \right\|^2$$

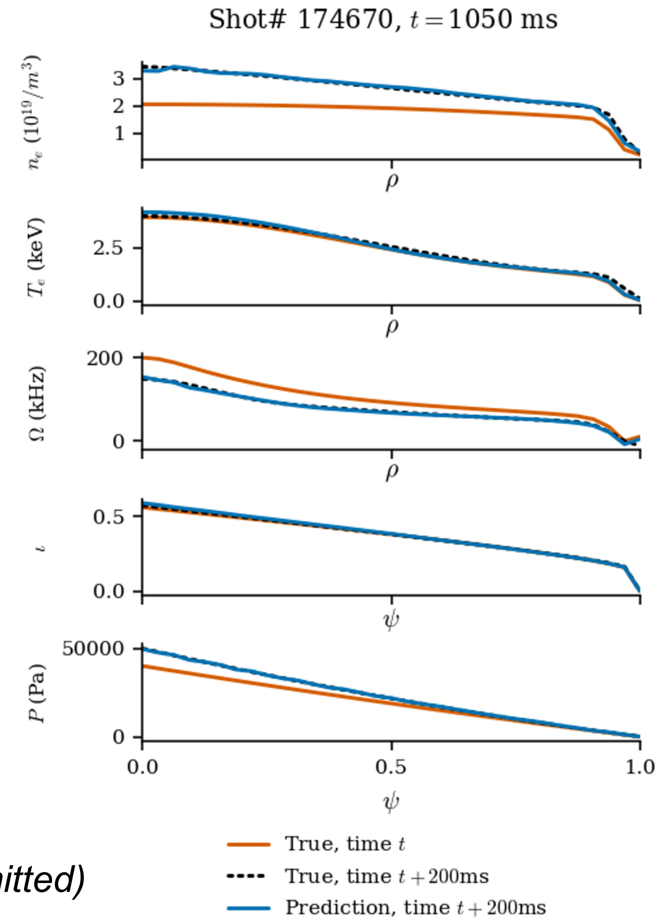
  - **u** : control action
  - **x** : state
  - **w** : weights



(Abbate, Conlin, NF, submitted)

ML Predictions qualitatively accurate (J. Abbate, R. Conlin)

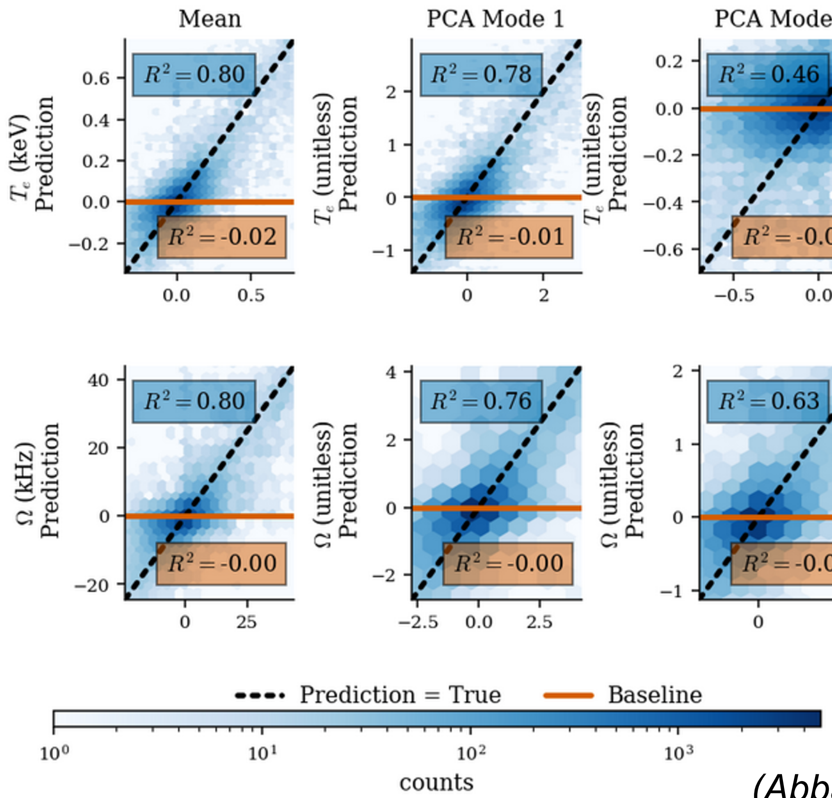
- Trained on ~6,000 shots from DIII-D
  - ~200,000 timesteps
- Predictions generally accurate over wide range of plasma states
- Occasional noise in predictions can be mitigated by smoothing/post processing



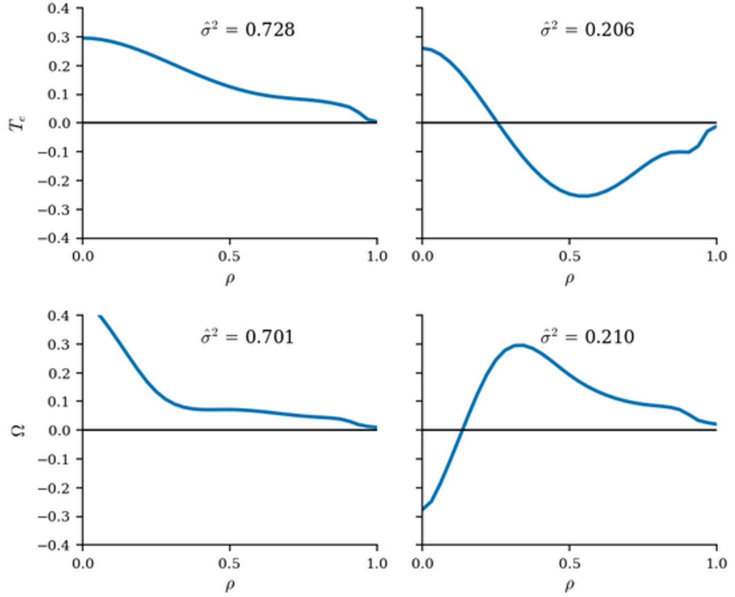
(Abbate, Conlin, NF, submitted)

ML prediction captures dominant modes (J. Abbate, R. Conlin)

Good agreement in mean value and amplitude of first PCA mode



PCA Modes of profiles

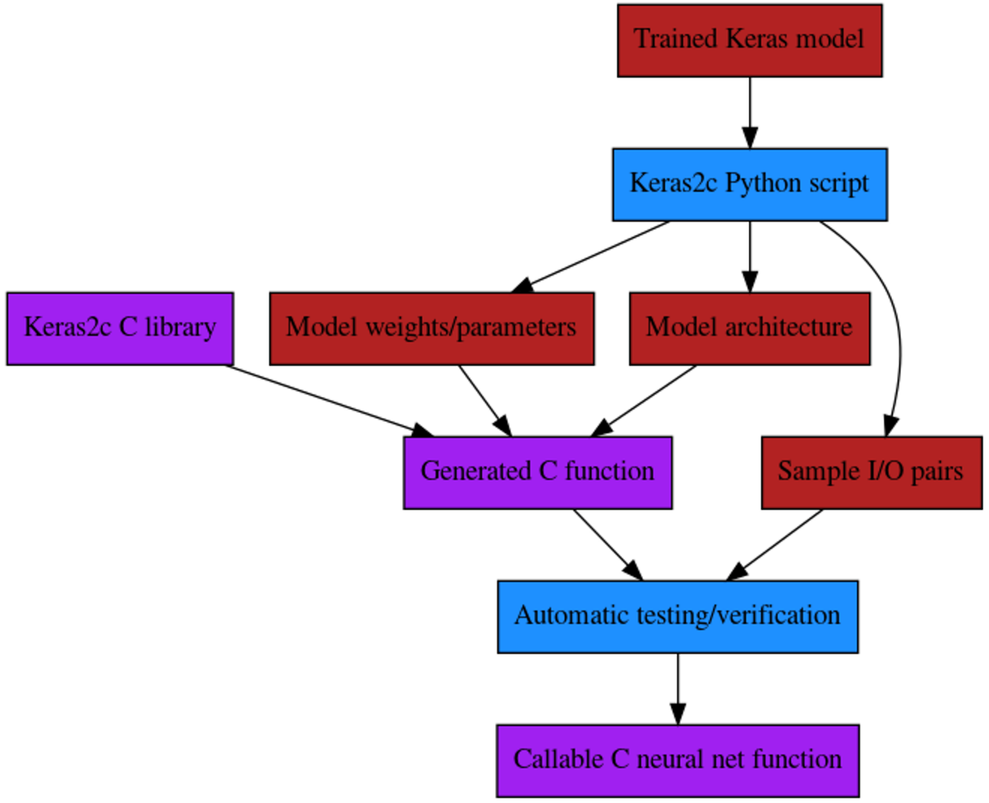


(Abbate, Conlin, NF, submitted)

Developed Keras2c to run NN in real time (R. Conlin)

Script/Library for converting Keras neural nets to C functions

- Designed for simplicity and real time applications
- Core functionality only ~1300 lines
- Generates self-contained C function, no external dependencies
- Supports full range of operations & architectures
- Fully automated conversion & testing
- Tested on DIII-D for profile prediction & disruption prediction w/ FRNN (Tang)

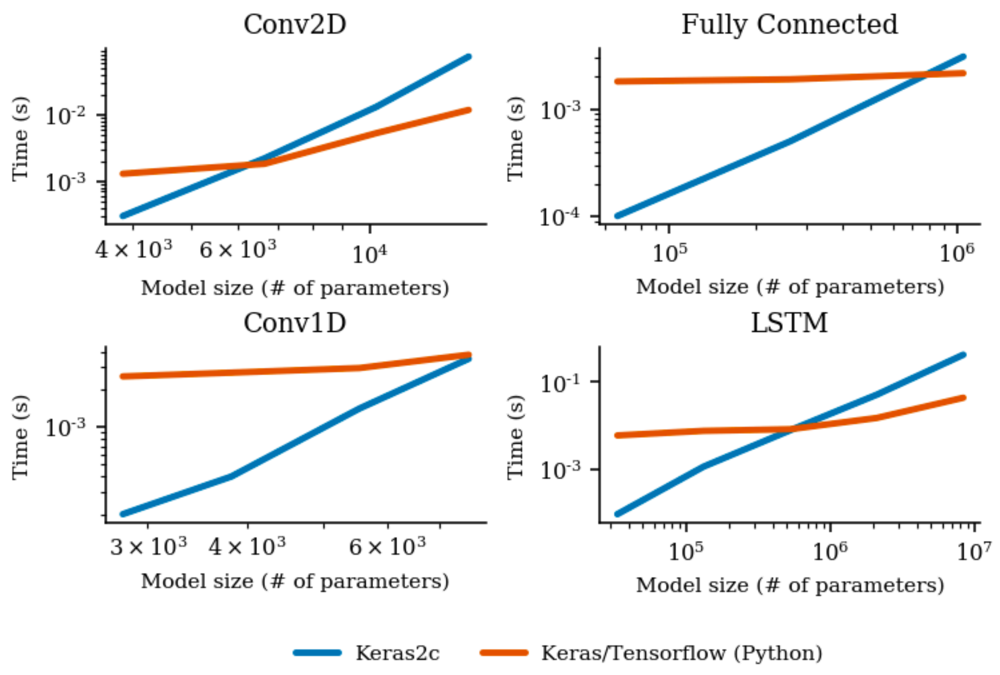


(Conlin, In Review)

Developed Keras2c to run NN in real time (R. Conlin)

Script/Library for converting Keras neural nets to C functions

- Designed for simplicity and real time applications
- Core functionality only ~1300 lines
- Generates self-contained C function, no external dependencies
- Supports full range of operations & architectures
- Fully automated conversion & testing
- Tested on DIII-D for profile prediction & disruption prediction w/ FRNN (Tang)

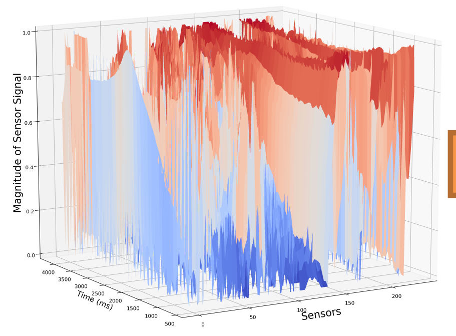


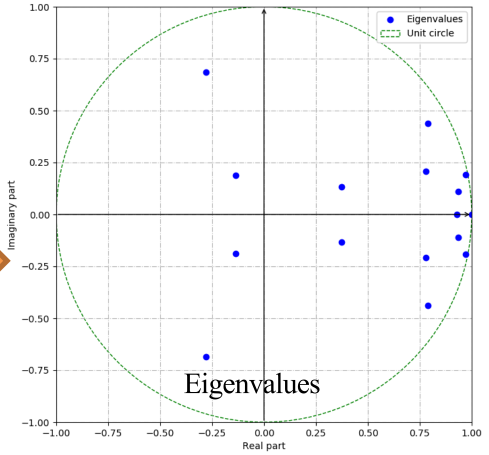
(Conlin, In Review)

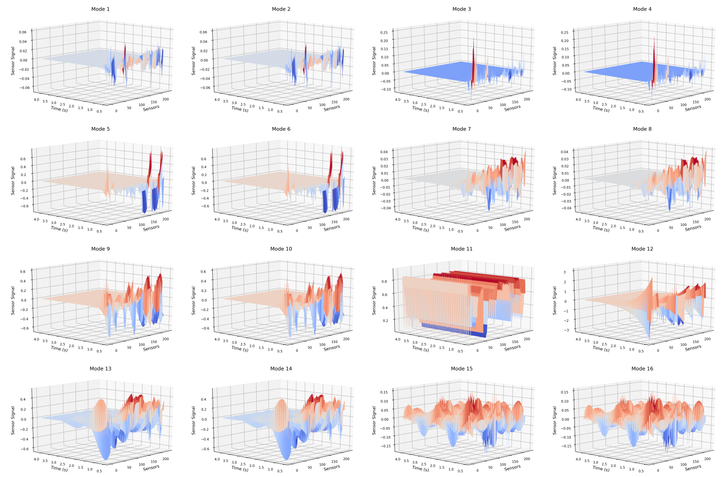
System Identification for Evolution Dynamics: Dynamic Mode Decomposition Applied to Diagnostics (L. Palacios)

Dominant Mode Decomposition and Eigenvalues

ECE #180718

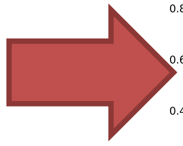


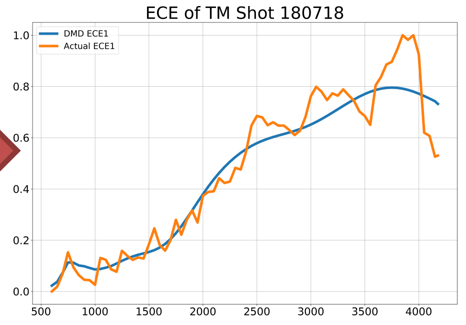




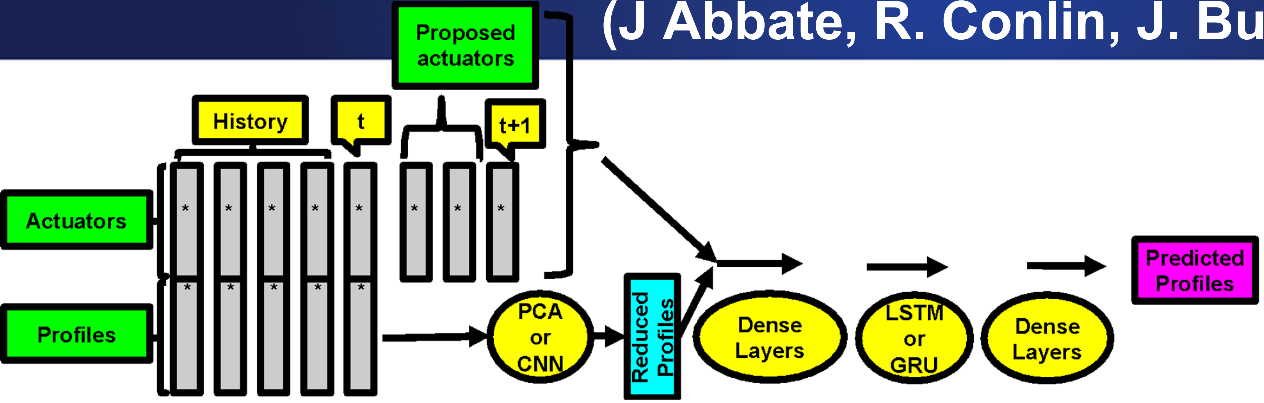
- **Extraction of dominant spatio-temporal modes**
- **Identification of systems (full or reduced) in state-space**
- **➔ Allow model for control**

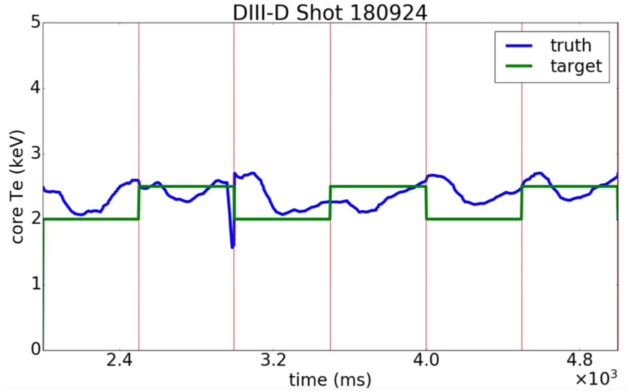
$$x_{k+1} = Ax_k + Bu_k$$

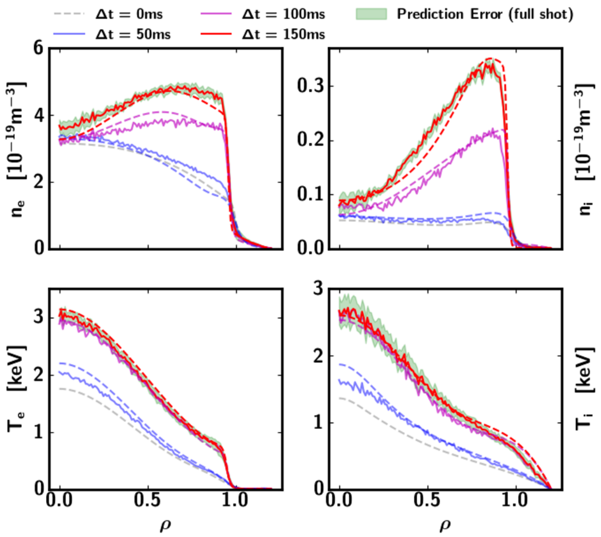




Machine Learning Based Profile Control (J Abbate, R. Conlin, J. Butt)







- ML Algorithms trained on the profiles from the DIII-D database (thousands of shots)
- Initial control tests on DIIID started, calculation times <1ms (Te profile)
- ML Control: Use ML to adjust beams and current to achieve desired profiles

andlinger center for energy+the environment

1. Automated Plasma Equilibrium from Diagnostics

2. Tearing and Disruption Prediction

3. Machine Learning Control for Disruption Avoidance

4. RT Adapting ML Prediction and Control

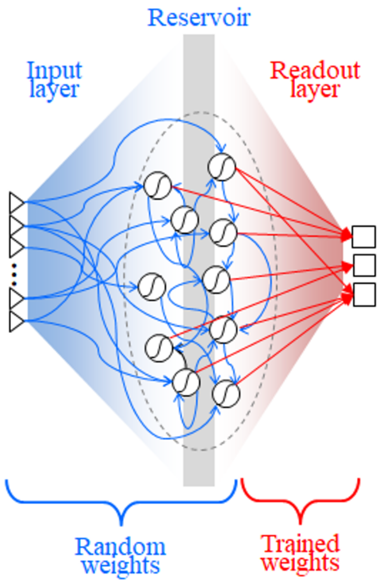
Reservoir Computing Network (A. Jalalvand)

Ghent University - Belgium

A **recurrent** neural network with **random** and **sparsely connected** early layers.
Only the last layer is trained using **linear regression**.

Specifications of RCN:

- Projects the inputs to a random very high-dimensional space.
- Ability to process **temporal information** (time-series data analysis)
- Much **faster** and **easier training** procedure compared to DNNs.
  - LSTM: **5 hours** on GPU
  - RCN (with comparable performance): **less than 5 minutes** on CPU
  - Easy & fast training makes **parallel training** and "in-situ" **model adaptation** possible.
- Successful application on complex data such as speech, image, radar.
- **In progress** (since March 2020)**:** Plasma profile and tearing mode prediction

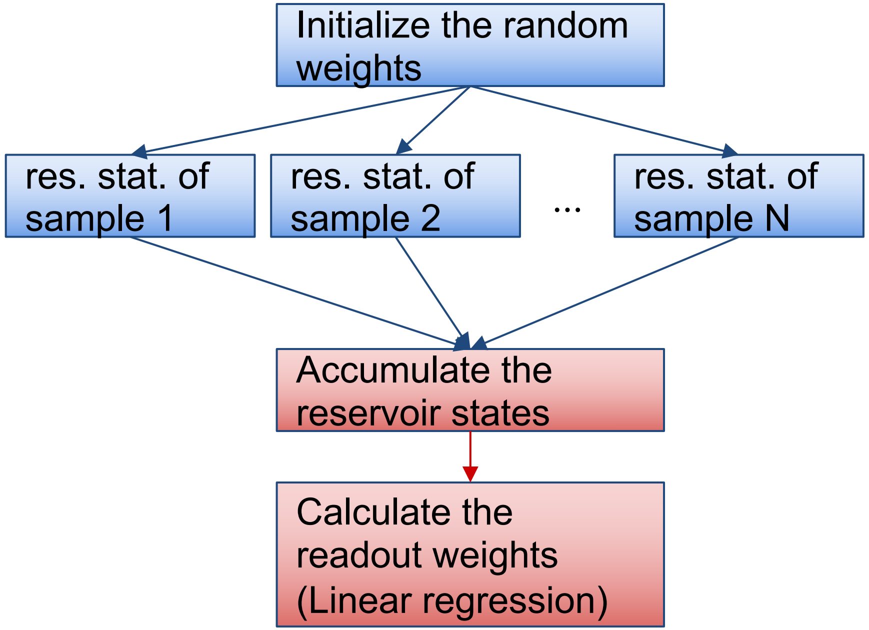


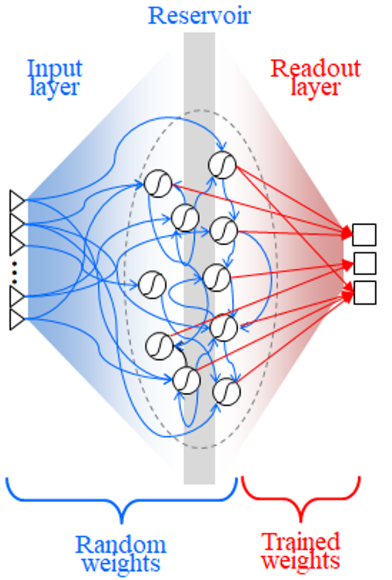
Reservoir Computing Network (A. Jalalvand)

Ghent University - Belgium

Parallel Training: The most expensive part of the training is calculating the reservoir states for the training samples. This part can be run for each sample in parallel.

Calculate the reservoir states for the training samples

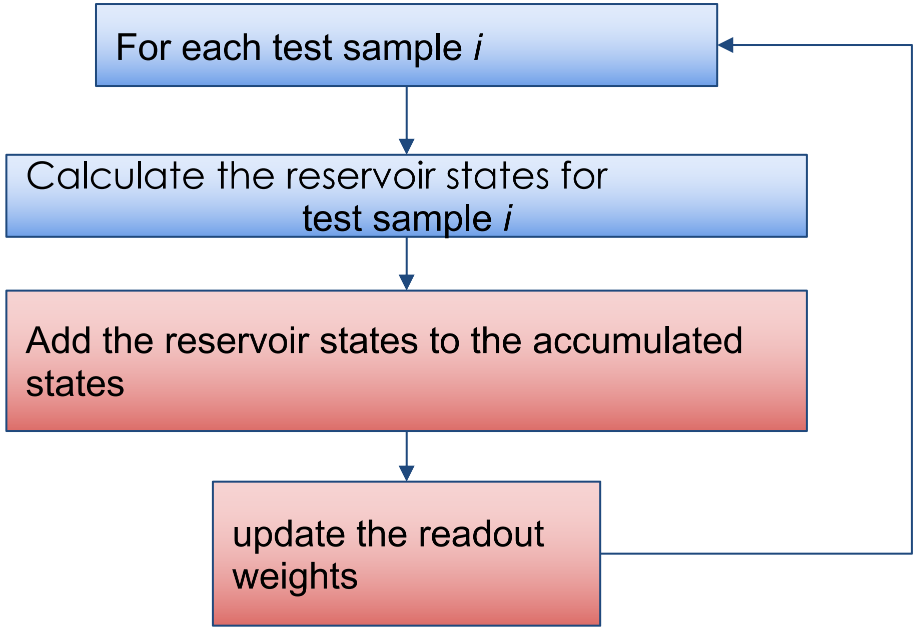

Initialize the random weights
res. stat. of sample 1
res. stat. of sample 2
...
res. stat. of sample N
Accumulate the reservoir states
Calculate the readout weights (Linear regression)



DIII-D NATIONAL FUSION FACILITY PRINCETON UNIVERSITY GHENT UNIVERSITY andlinger center for energy+the environment

Reservoir Computing Network (A. Jalalvand)

Adaptation: By storing the accumulated reservoir states from the training set, we can update it with the validation states and update the readout weights.

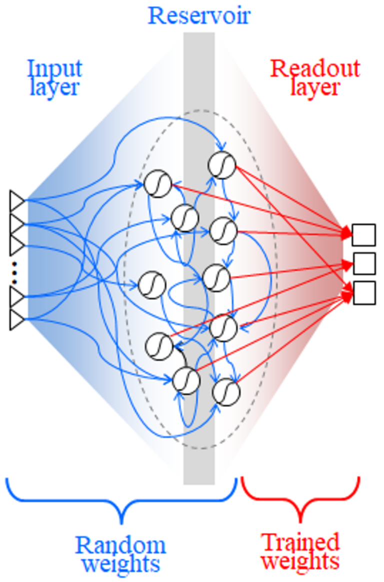


For each test sample *i*

Calculate the reservoir states for test sample *i*

Add the reservoir states to the accumulated states

update the readout weights



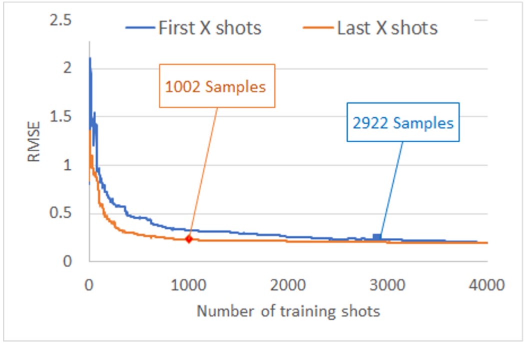
Reservoir Computing Network (A. Jalalvand)

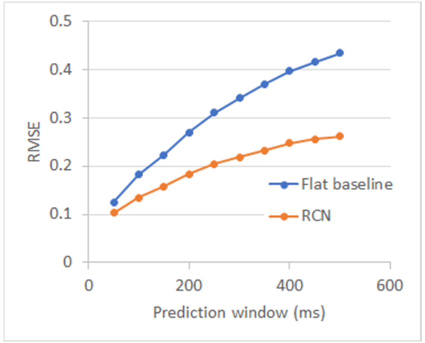
Ghent University - Belgium

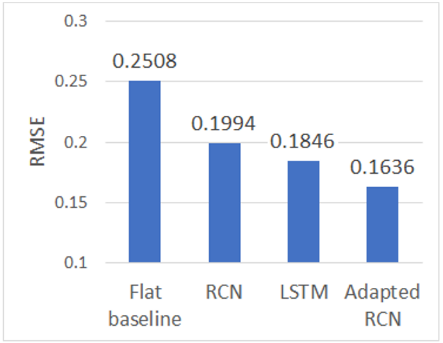
To be submitted:

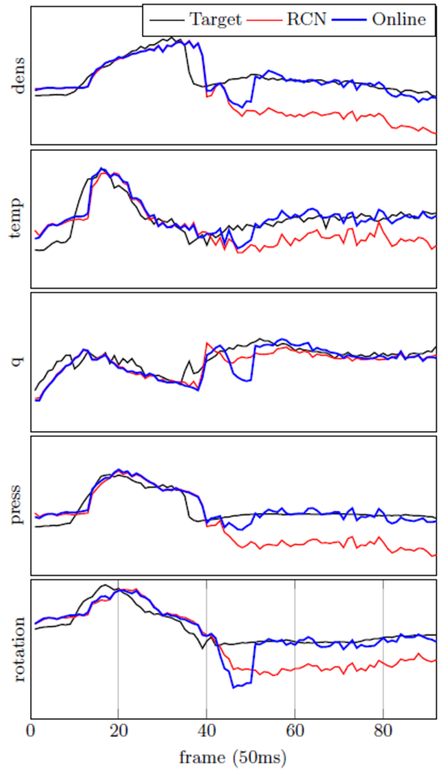
Real-Time and Adaptive Reservoir Computing for Fusion Plasma Applications
A. Jalalvand, J. Abbate, R. Conlin, G. Verdoolaege, and E. Kolemen

- 4000 Training shots (~90 hours) and 500 Test shots
- Training time on full training set 55 sec on Core i7 CPU.
- **Online adaptation** every 500ms: 96ms per adaptation









Backup

Real-Time Tearing Mode Prediction and Avoidance (L. Palacios, E. Kolemen)

- **New signals in dataset**
  - More than 200 signals
  - 5 and 500 kHz ECE signals (40 signals each)
  - Other signals: Thompson, Magnetics, CERREAL and MSE
- **Phased LSTM for sensor fusion and arbitrary sampling rates (Palacios, Kolemen)**
  - Proper integration of signals from diverse range of sensors, each one of them having its own sampling rate
- **Reservoir Computing Networks (RCN) (Jalalvand, Kolemen)**
  - Time-series data analysis using RCN, a variation of recurrent neural networks
- **Dynamic Mode Decomposition and Sparse Identification of Nonlinear Systems (Palacios, Kolemen)**
  - Data-driven system identification solutions based on linear algebra and nonlinear dynamics
  - Also suitable for diagnostics and control