Real-time control and estimation in tokamaks with machine learning accelerated predictive models

Dan Boyer¹

S. Kaye¹, K. Erickson¹, V. Gajaraj², J. Kunimune³, M. Zarnstorff¹
H.-S. Kim⁴, S.-H. Hahn⁴, S. Morosohk⁵, E. Schuster⁵, S. Sabbagh⁶, J. Ahn⁶

¹PPPL, New York U.²Olin College³NFRI⁴Lehigh U.⁵Columbia U.⁶

3rd IAEA TM on Fusion Data Processing Validation and Analysis
27-31 May, 2019 Vienna, Austria
Understanding plasma at reactor relevant conditions requires multi-billion dollar devices - how do we make the most of this investment?

• **Expert operators?:** Shot development, control commissioning, parameter scans, between shot decisions and analysis are *expensive, limited, and error prone*

• **Predict first?:** use integrated models (e.g., TRANSP) to develop experiments
  • *Can take hours/days to run predictive models for entire discharge*
  • *Plus, careful planning can only get us so far...*
    • e.g., do we shut down a shot if a beam fails to come on right away?
    • Do we try alternative plans to recover the scenario?
    • Do we switch to another scenario altogether?

• **Advanced model-based real-time control and optimization** can help ensure performance is safely maximized
Advanced control capabilities for online shot planning will require reduced model based control and optimization techniques.

**Real-time Diagnostics**

**Estimate**
plasma state from limited measurements

Where we think we are

*Real-time prediction*
Advanced control capabilities for online shot planning will require reduced model based control and optimization techniques.

**Estimate**
plasma state from limited measurements

Where we think we are

**Forecast**
future behavior of the shot

Where we think will be

Faster-than-real-time prediction

**Actuator plan**

Real-time Diagnostics

Real-time prediction
Advanced control capabilities for online shot planning will require reduced model based control and optimization techniques.

Real-time Diagnostics

**Estimate**
plasma state from limited measurements

Where we think we are

**Forecast**
future behavior of the shot

Where we think will be

**Faster-than-real-time prediction**

Actuator plan

**Supervisory control**
shut down the shot or change mission requirements

Real-time prediction
Advanced control capabilities for online shot planning will require reduced model based control and optimization techniques.

- **Estimate**: Plasma state from limited measurements
- **Forecast**: Future behavior of the shot
- **Actuator planning**: To optimize performance + avoid machine limits
- **Supervisory control**: Shut down the shot or change mission requirements

**Real-time Diagnostics**

- **Where we think we are**
- **Where we want to go**

**Faster-than-real-time prediction**

- **Where we think will be**
- **Where we could go**

**Real-time prediction**

- **Much-faster-than-real-time prediction**
Advanced control capabilities for online shot planning will require reduced model based control and optimization techniques.

- Estimate plasma state from limited measurements
- Forecast future behavior of the shot
- Supervisory control shut down the shot or change mission requirements
- Actuator plan to optimize performance + avoid machine limits
- Forecast future behavior of the shot
- Much-faster-than-real-time prediction
- Faster-than-real-time prediction

Real-time Diagnostics

Where we think we are

Where we want to go

Where we think will be

Where we could go

Can we make our integrated plasma models (e.g., TRANSP) fast enough while maintaining high enough fidelity?
Machine learning approaches can enable faster versions of modules in integrated modeling code for real-time applications

- Predictive TRANS(p) simulations can take hours per simulation second
- **NUBEAM** is a Monte Carlo code that calculates the effect of neutral beams on the plasma (heating, current drive, torque)
  - Often takes >30% of calculation time
- Basic machine learning approaches enable the development of **NubeamNet**


**Boyer et al., Nuclear Fusion 2019**

Calculation of beam effects (at 5ms intervals) took less than 50ms for the entire shot, compared to minutes to hours for NUBEAM
Orders of magnitude speed increase enabled by neural networks trained on database of NUBEAM results

Neural network model development

1. **Database generation**: NSTX-U TRANSP runs (~2000, ~100 samples per run), including scans of important parameters
2. **Data reduction**: Make the data manageable. e.g., reduce profile data, time history

- **Dataset**
- **Input layer**
- **Hidden layer**
- **Output layer**
- **Neuron** (w/ nonlinear activation function)

**Connection weights** (tuned based on training data)

**Neural network** - a universal nonlinear approximator

3. **Training**
   - Tune connection weights

4. **Validation**
   - Select model topology

5. **Testing**
   - Does the model generalize?

Averaging the results of an ensemble of trained models can give a sense of the confidence of predictions
Two of the challenges to machine learning for NUBEAM: spatially distributed data and time history dependence

- Deep learning would address with convolutional and/or recurrent neural networks
- Much simpler approach used here:
  - **Principal component analysis** compresses spatial data

Average q profile in dataset

Most significant q profile modes

q profile projected onto modes, coefficients used for training
Two of the challenges to machine learning for NUBEAM: spatially distributed data and time history dependence

- Blackbox machine learning would address with convolutional and/or recurrent neural networks
- Much simpler approach used here:
  - **Principal component analysis** compresses spatial data and **low-pass filtering** encodes time-history

### Diagrams

1. **Average q profile in dataset**
   - Graph showing q profile as a function of normalized toroidal flux.

2. **Most significant q profile modes**
   - Graph showing the modes of the q profile and coefficients used for training.

3. **Beam power augmented with low pass filtered versions representing slowing down**
   - Graph showing the power profile with filtered versions.

**Legend**
- Unfiltered
- 20ms
- 50ms
- 100ms

**Beamline 1A power, Run: 204118S26**
Overview of NUBEAM Neural Network

- **Inputs**
  - Injected powers
  - Input scalars
  - Input profiles

- **Neural network inputs**
  - Neural network ensemble

- **Low-pass filters**

- **Filtered powers**

- **PCA projection**

- **Mode coefficients**

- **Standardization**

- **Neural network outputs**
  - Ensemble averaging
  - Inverse standardization

- **Mode coefficients**

- **PCA projection**

- **Scalar outputs**

- **Profiles**

- **Outputs**
Validation: The topology of the model must be selected to optimize quality of fit and evaluation time

- Neural net code implemented on NSTX-U real-time computer
- Calculation times well within the 200 microsecond control system cycle time, much faster than time scales of interest

- Significant improvement with more than one layer, but not much benefit in going deeper
- Adding nodes improves fit, but improvement slows or rolls over around 100-125 nodes per layer
Designing model topology for new models will become a challenge as approach expands to more problems, scenarios, and machines.

• But we can exploit cluster computing capability to **automate the design**
  • A workflow has been developed to use **genetic algorithms** to find the optimal trade off between model accuracy and complexity
  • **Generations of models** are trained and evaluated
  • Tournament **selection**, random **mutation** applied to create next generation
  • Individuals in each generation can be **evaluated in parallel on the cluster**
Trained neural network is able to accurately reproduce time history and profiles in testing dataset

- Accuracy and timing indicate the model is well-suited for real-time applications
- Able to capture changes in profile shape, maintain smooth profiles due to use of basis functions
- Ensemble standard deviation gives indication of neural network confidence
Ensemble standard deviation can help determine when model predictions should be considered reliable

- Models **less likely to disagree** on predictions made from inputs inside training region.
- **Large ensemble variance** can be used to declare predictions unreliable
  - Can guide expansion of training data set

Fraction of unreliable predictions increases toward 1.0 when input elongation moves outside of the interval used in training.
NubeamNet enables real-time current profile observer with estimation of uncertain Zeff and fast ion diffusivity

**Extended Kalman Filter**

**Predict**

\[ x_k = f(x_{k-1}, u_k) + w_k \]
\[ z_k = h(x_{k-1}, u_k) + v_k \]

**Update**

compare prediction to measurements

**Measurements**

Neutron rate, plasma current, and in-domain poloidal flux gradient (opt.)

**States and parameters**

Poloidal flux + Zeff, and fast ion diffusivity (assumed flat)

**Reduced magnetic diffusion equation + NubeamNet**

\[
\frac{\partial \psi}{\partial t} = \frac{\eta(T_e)}{\mu_0\rho_b^2\hat{F}^2} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \left( \hat{\rho} \hat{F} \hat{G} \hat{H} \frac{\partial \psi}{\partial \hat{\rho}} \right) + R_0 \hat{H} \eta(T_e) \frac{\langle j_{NI} \cdot \hat{B} \rangle}{B_{\phi,0}}
\]

\[
\frac{\partial \psi}{\partial \hat{\rho}} \bigg|_{\hat{\rho}=0} = 0,
\frac{\partial \psi}{\partial \hat{\rho}} \bigg|_{\hat{\rho}=1} = -\frac{\mu_0}{2\pi} \frac{R_0}{\hat{G} \bigg|_{\hat{\rho}=1}} \hat{H} \bigg|_{\hat{\rho}=1} I(t),
\]
NubeamNet-based real-time current profile observer: estimated Zeff and fast ion diffusivity converge to actual values

- NubeamNet enables real-time calculation of sensitivity of measurements to Zeff, fast ion diffusivity
- Mismatch between predictions and measurements drive updates to estimated states
NubeamNet-based real-time current profile observer: Poloidal flux profile converges to actual, faster with in-domain meas.

Poloidal flux profile at 4 locations
No in-domain measurements used by observer

Poloidal flux profile at 4 locations
4 in-domain measurements used by observer
High-fidelity real-time integrated modeling capabilities through machine learning surrogate models could help optimize future reactor performance.

**INPUT:** Experimental data and model/calculation assumptions

- NB source (NUBEAM) Fully validated For D-T ops
- Equilibrium solver
- Neutral transport
- Plasma transport solver

**OUTPUT:** "Plasma State"

- MHD, *AE stability codes, ELITE, IPS, ScIDACs, etc.
- Gyrokinetic codes (GTC-NEO, GYRO, GS2, GTS...)

Output of TRANSP (Plasma State File) is standardized for simplifying input to other computationally intensive codes

Figure from S. Kaye, TRANSP User Group Meeting 2015

**EFIT01 Neural network**

**Bootstrap current**

**Resistivity**

**Para/diamagnetism**

**Transport fluxes**
Meneghini NF 2017, 2014 (TGLF), Citrin NF 2015 (QuaLiKiz)

**Boundary conditions**
Meneghini NF 2017 (EPED)

D. Boyer

V. Gajaraj
Thank you!

This research was supported by the U.S. Department of Energy under contract number DE-AC02-09CH11466