Applications of Neural Networks to Modeling and Control of Particle Accelerators

Auralee Edelen

Fermilab Accelerator Physics and Technology Seminar

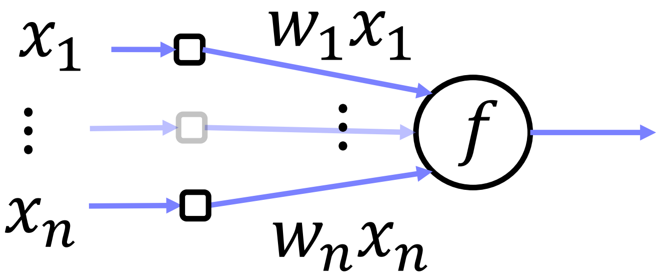
8 June 2017

Overview

- Background on Neural Networks
- Control Challenges in Particle Accelerators
- Overview of Applications *(with some examples)*
  - Online Modeling
  - Model Predictive Control
  - Virtual Diagnostics
  - Failure Prediction, Anomaly Detection, and Machine Protection
  - Reinforcement Learning / Neural Network Control Policies
  - Incorporating Image-based Diagnostics into Control Policies
- Final Notes
  - Practical Challenges
  - Funding Climate
  - Conclusions

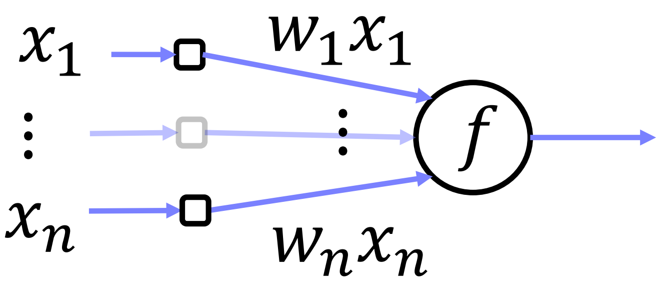
What are neural networks?

Artificial Neural Networks

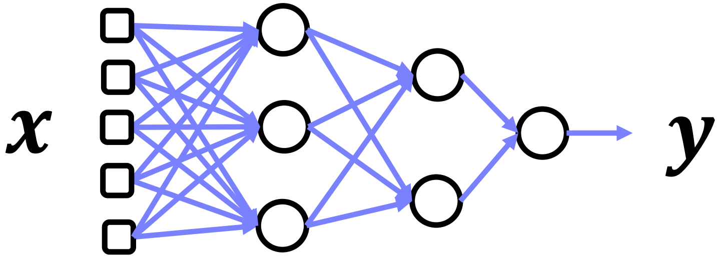


a neuron or node

Artificial Neural Networks

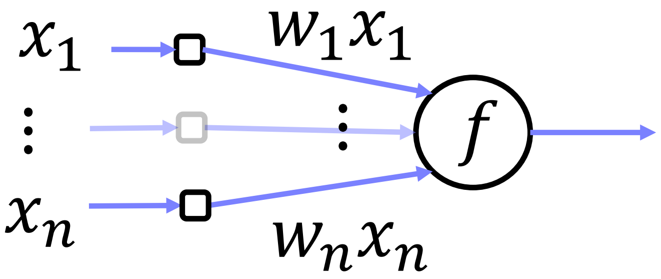


a neuron or node

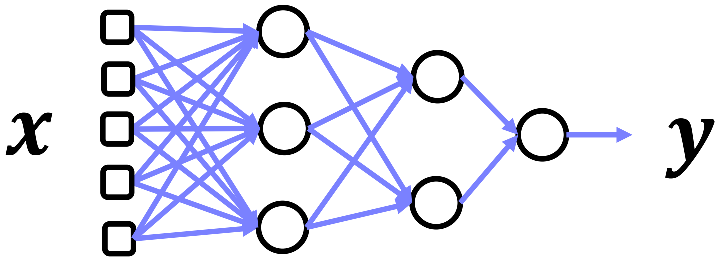


a feed-forward network

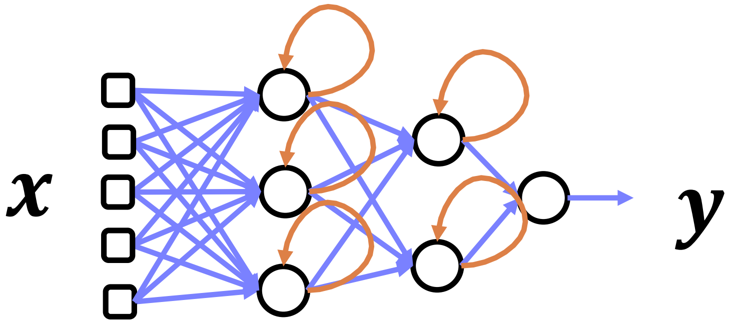
Artificial Neural Networks



a neuron or node

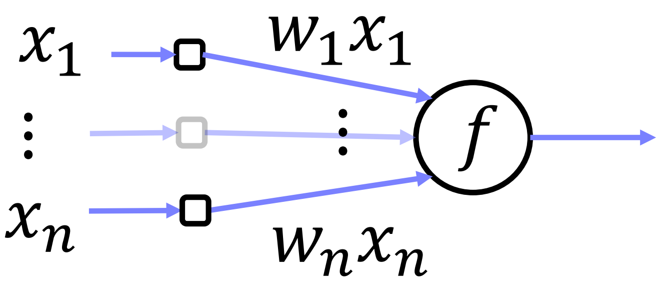


a feed-forward network

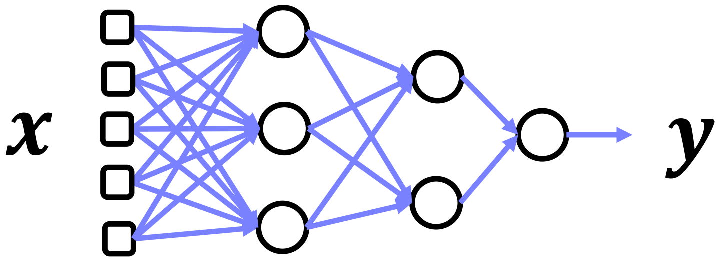


a recurrent network

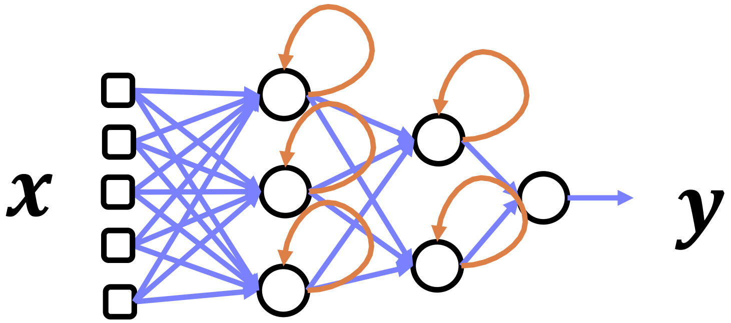
Artificial Neural Networks



a neuron or node



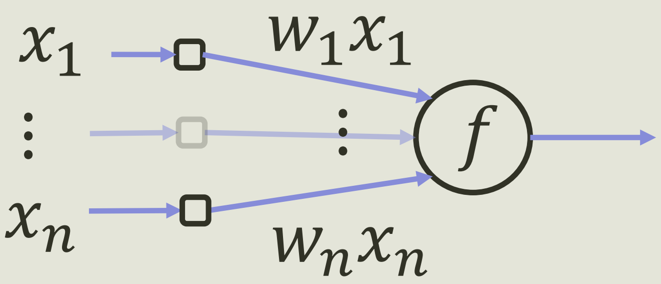
a feed-forward network



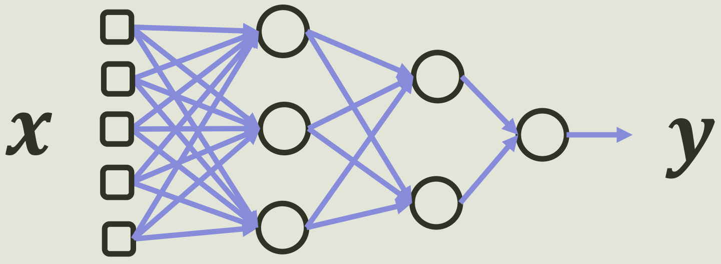
a recurrent network

… many more architectures!

Artificial Neural Networks

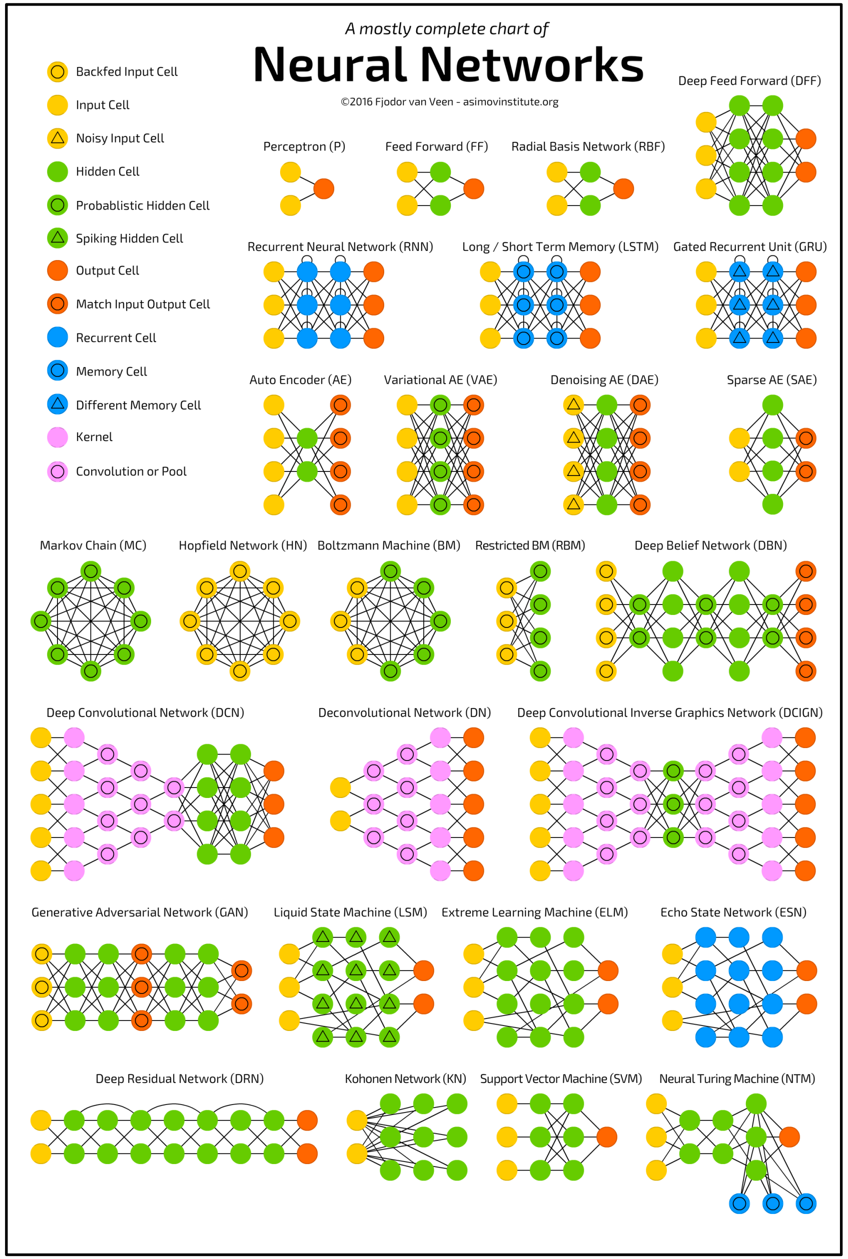


a neuron or node



a feed-forward network

… many more architectures!


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

See, for example, the Neural Network Zoo website.

How does this relate to "machine learning," "artificial intelligence," and "deep learning"?

…what do these terms mean anyway?

Field Taxonomy (as of now…)

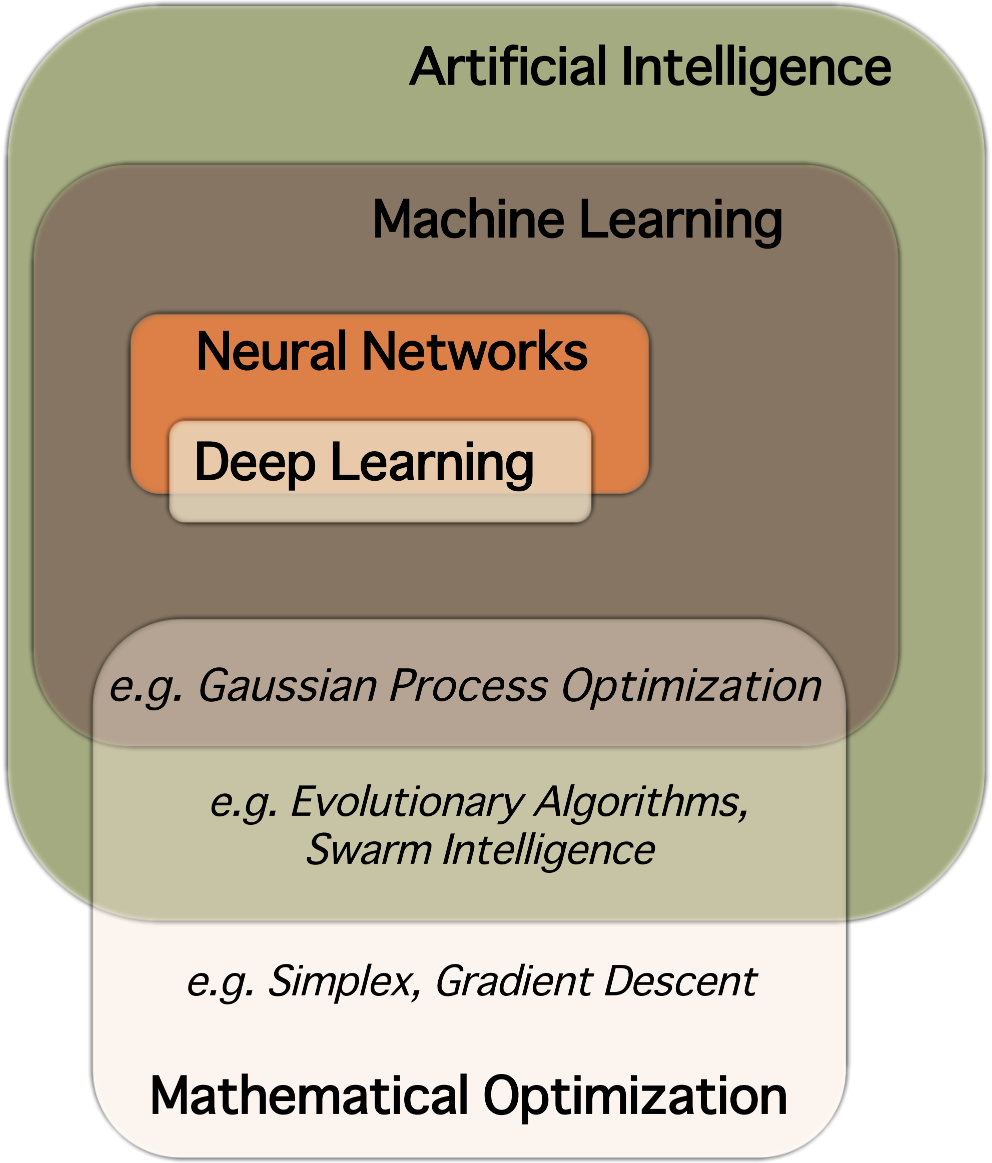
- Artificial Intelligence (AI)
  - *Concerned with enabling machines to exhibit aspects of human intelligence: knowledge, learning, planning, reasoning, perception*
  - Narrow AI: focused on a task or similar set of tasks
  - General AI: human-equivalent or greater performance on any task

- Machine Learning (ML)
  - *Enabling machines to complete tasks without being explicitly programmed*
  - Common tasks: Regression, Classification, Clustering, Dimensionality Reduction

- Neural Networks (NNs)
  - *An approach within ML that uses many connected processing units*
  - Many different architectures and training techniques

- Deep Learning (DL)
  - *Learning hierarchical representations*
  - Right now, largely synonymous with deep (many-layered) NN approaches

Note that these definitions are not rigid: there is a lot of fluidity in the field



Artificial Intelligence

Machine Learning

Neural Networks

Deep Learning

e.g. Gaussian Process Optimization

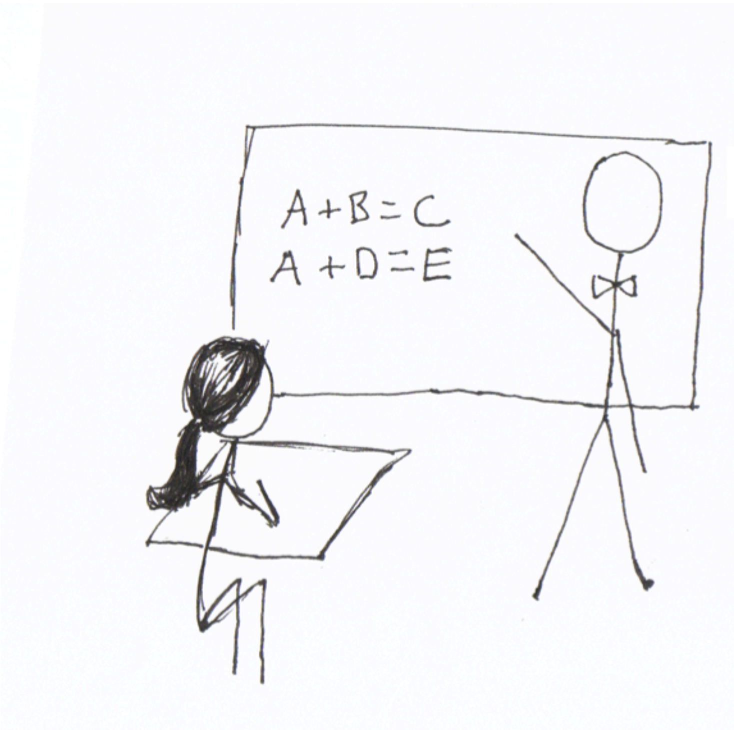
e.g. Evolutionary Algorithms, Swarm Intelligence

e.g. Simplex, Gradient Descent

Mathematical Optimization

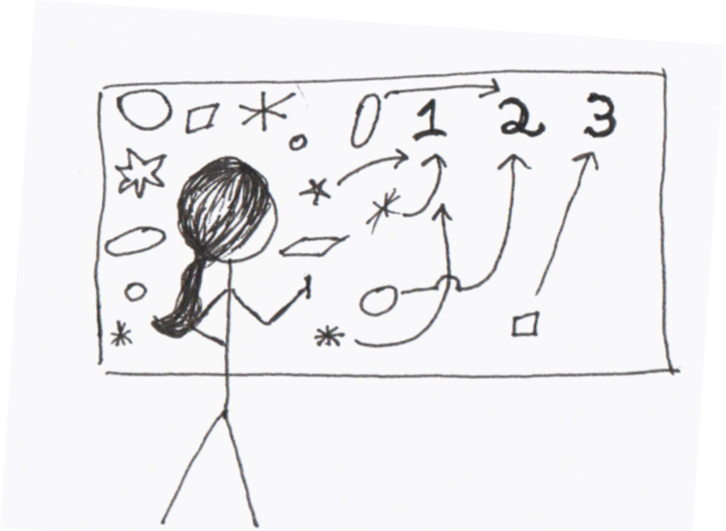
How do neural networks "learn"?

Basic Learning Paradigms



Supervised Learning

learn known input/output pairs



Unsupervised Learning

no labeled data → infer structure

Reinforcement Learning

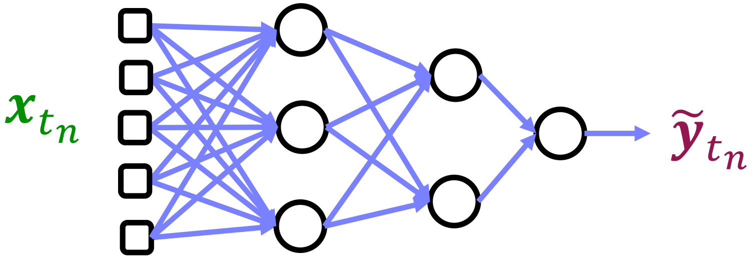
interact with the environment → adjust behavior based on reaction

Auralee Edelen May 2017

Example: multiple-input, single-output process model

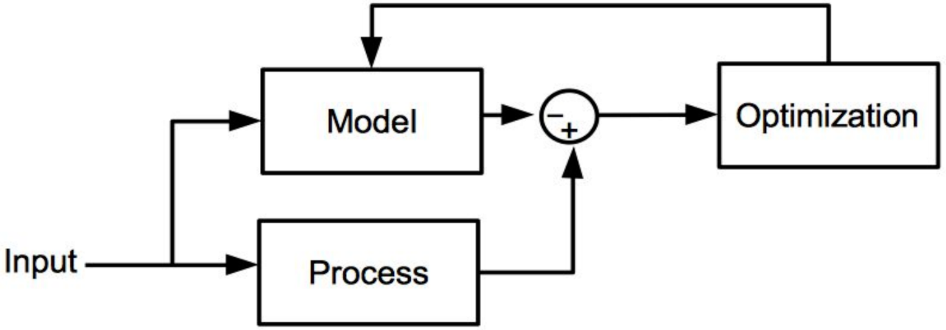
Data set of **input** and **output** pairings:

$$\left.\begin{matrix} x_1 \\ \vdots \\ x_n \end{matrix}\right\} \quad \begin{matrix} \boldsymbol{x}_{t_1} \\ \vdots \\ \boldsymbol{x}_{t_n} \end{matrix} \qquad \begin{matrix} \boldsymbol{y}_{t_1} \\ \vdots \\ \boldsymbol{y}_{t_n} \end{matrix}$$



Want to find approximate map: $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{y}$

Model Learning



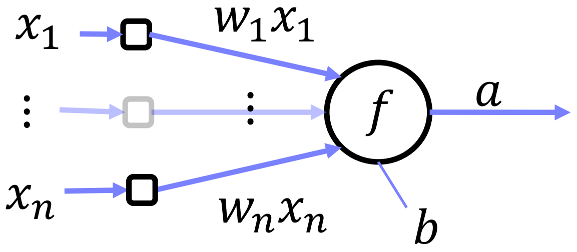
Basic Update Example

$$C(w,b) = \frac{1}{2t_n}\left[\sum_{t_n}(y_{t_n} - \tilde{y}_{t_n})^2\right]$$

$$w_k \longrightarrow w'_k = w_k - \alpha\frac{\partial C}{\partial w_k}$$

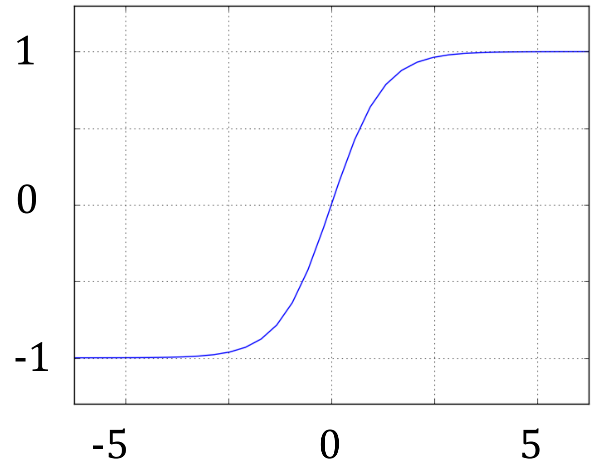
$$b_k \longrightarrow b'_k = b_k - \alpha\frac{\partial C}{\partial b_k}$$

Basic Structures



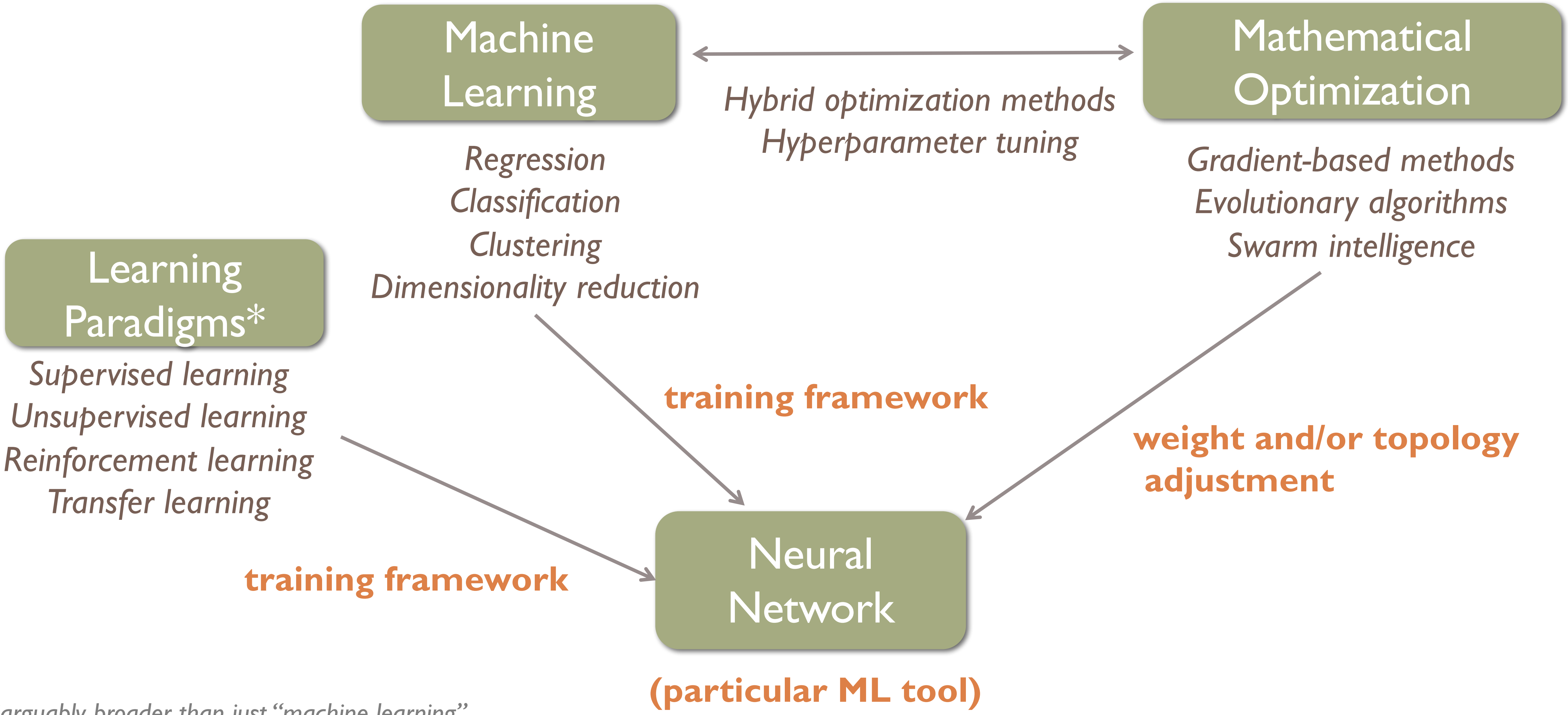
$$f\left(\sum_n w_nx_n + b\right) = a$$

e.g. $f(z) = \frac{2}{(1+e^{-2z})} - 1$



Auralee Edelen May 2017

How this all fits together for NNs



Machine Learning

Regression
Classification
Clustering
Dimensionality reduction

Mathematical Optimization

Hybrid optimization methods
Hyperparameter tuning

Gradient-based methods
Evolutionary algorithms
Swarm intelligence

Learning Paradigms*

Supervised learning
Unsupervised learning
Reinforcement learning
Transfer learning

training framework

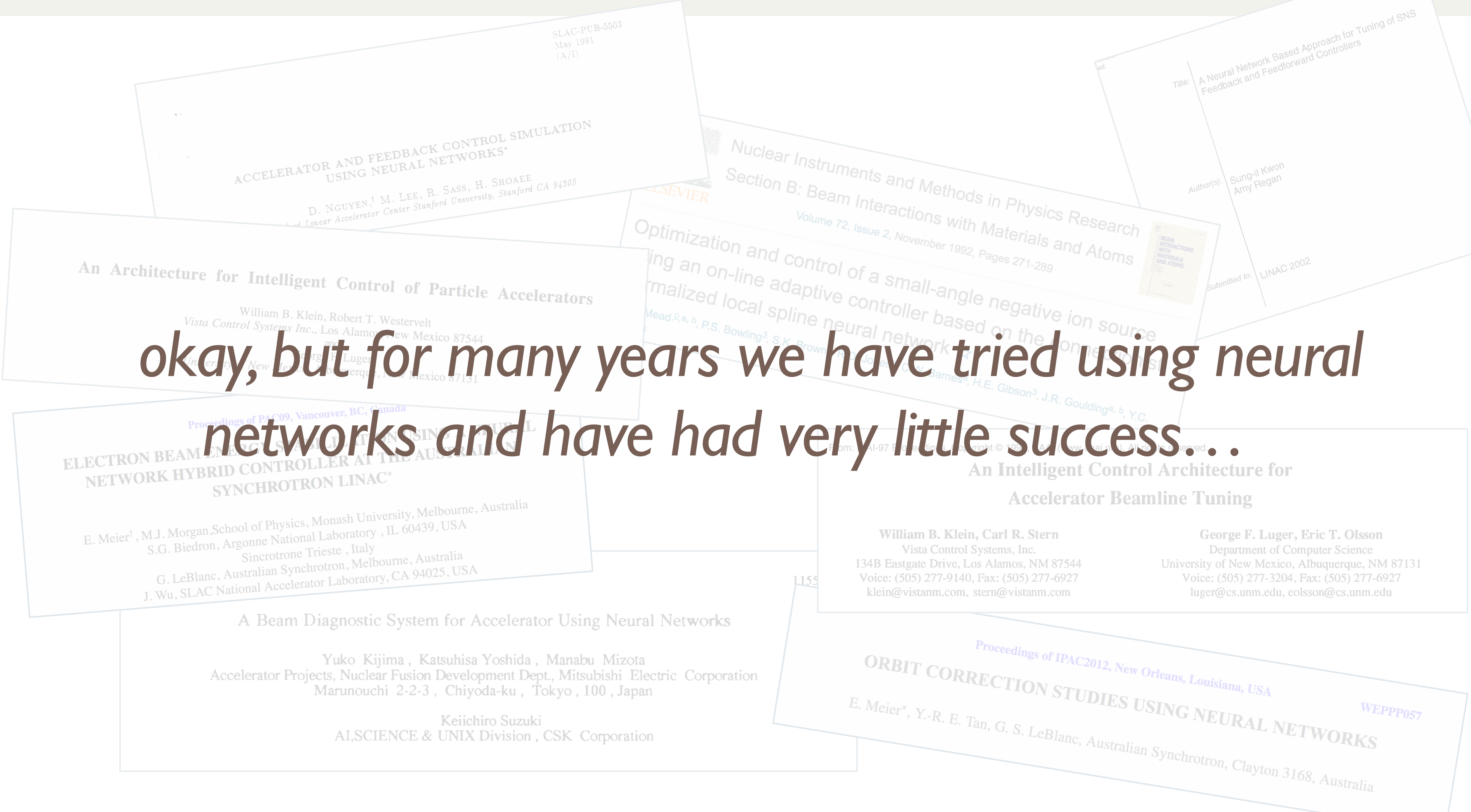
training framework

weight and/or topology adjustment

Neural Network

(particular ML tool)

**arguably broader than just "machine learning"* *okay, but for many years we have tried using neural networks and have had very little success…*



SLAC-PUB-5503
May 1991
(A/I)

ACCELERATOR AND FEEDBACK CONTROL SIMULATION USING NEURAL NETWORKS*

D. NGUYEN,† M. LEE, R. SASS, H. SHOAEE
Stanford Linear Accelerator Center Stanford University, Stanford CA 94305

An Architecture for Intelligent Control of Particle Accelerators

William B. Klein, Robert T. Westervelt
Vista Control Systems Inc., Los Alamos, New Mexico 87544
and
George F. Luger
University of New Mexico, Albuquerque, New Mexico 87131

ELSEVIER

Nuclear Instruments and Methods in Physics Research
Section B: Beam Interactions with Materials and Atoms
Volume 72, Issue 2, November 1992, Pages 271-289

Optimization and control of a small-angle negative ion source
ing an on-line adaptive controller based on the connectionist
rmalized local spline neural network

Mead[a, b], P.S. Bowling[3], S.K. Brown, R. Jones, C. Barnes[a], H.E. Gibson[3], J.R. Goulding[a, b], Y.C.

Title: A Neural Network Based Approach for Tuning of SNS Feedback and Feedforward Controllers

Author(s): Sung-Il Kwon
Amy Regan

Submitted to: LINAC 2002

Proceedings of PAC09, Vancouver, BC, Canada

ELECTRON BEAM ENERGY STABILIZATION USING A NEURAL NETWORK HYBRID CONTROLLER AT THE AUSTRALIAN SYNCHROTRON LINAC*

E. Meier†, M.J. Morgan, School of Physics, Monash University, Melbourne, Australia
S.G. Biedron, Argonne National Laboratory, IL 60439, USA
Sincrotrone Trieste, Italy
G. LeBlanc, Australian Synchrotron, Melbourne, Australia
J. Wu, SLAC National Accelerator Laboratory, CA 94025, USA

From: PAC-97 Proceedings. Copyright © 1997, AAAI (www.aaai.org). All rights reserved.

An Intelligent Control Architecture for Accelerator Beamline Tuning

William B. Klein, Carl R. Stern
Vista Control Systems, Inc.
134B Eastgate Drive, Los Alamos, NM 87544
Voice: (505) 277-9140, Fax: (505) 277-6927
klein@vistanm.com, stern@vistanm.com

George F. Luger, Eric T. Olsson
Department of Computer Science
University of New Mexico, Albuquerque, NM 87131
Voice: (505) 277-3204, Fax: (505) 277-6927
luger@cs.unm.edu, eolsson@cs.unm.edu

1155

A Beam Diagnostic System for Accelerator Using Neural Networks

Yuko Kijima, Katsuhisa Yoshida, Manabu Mizota
Accelerator Projects, Nuclear Fusion Development Dept., Mitsubishi Electric Corporation
Marunouchi 2-2-3, Chiyoda-ku, Tokyo, 100, Japan

Keiichiro Suzuki
AI,SCIENCE & UNIX Division, CSK Corporation

Proceedings of IPAC2012, New Orleans, Louisiana, USA

WEPPP057

ORBIT CORRECTION STUDIES USING NEURAL NETWORKS

E. Meier*, Y.-R. E. Tan, G. S. LeBlanc, Australian Synchrotron, Clayton 3168, Australia

. . . so, what is different now?

Increased computational capability enables more complicated NN architectures and faster training + larger data sets

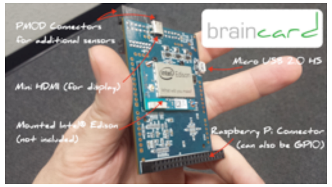
GPUs

IBM, ANL

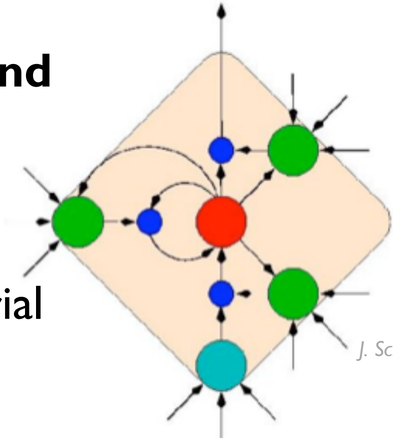
Accessibility of HPC clusters

Can **easily share** large data sets, code, and computing setups (e.g. via cloud computing services)

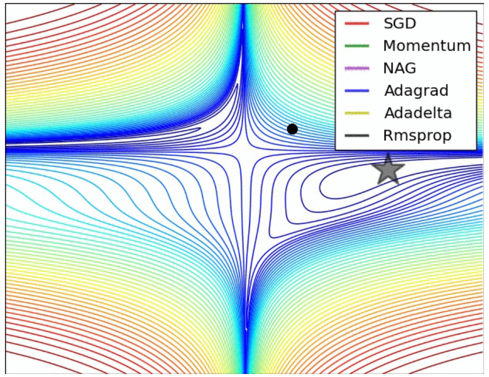
Up-and-coming advancements: neuromorphic hardware


PMOD Connectors for additional sensors
braincard
Micro USB 2.0 HS
Mini HDMI (for display)
Mounted Intel® Edison (not included)
Raspberry Pi Connector (can also be GPIO)

New network architectures and training paradigms, such as long short term memory (LSTM) networks, neural turing machines, and generative adversarial networks (GANs)



J. Schmidhuber



A. Radford

Better **theoretical understanding** of NNs and improved **optimization methods**

Applications have driven a lot of advancement (both algorithmic and practical/heuristic)

Google

. . . so, what is different now?

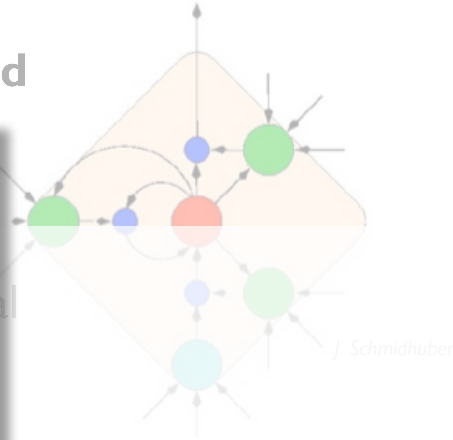
New network architectures and

Increased com
enables more co
and faster trainin

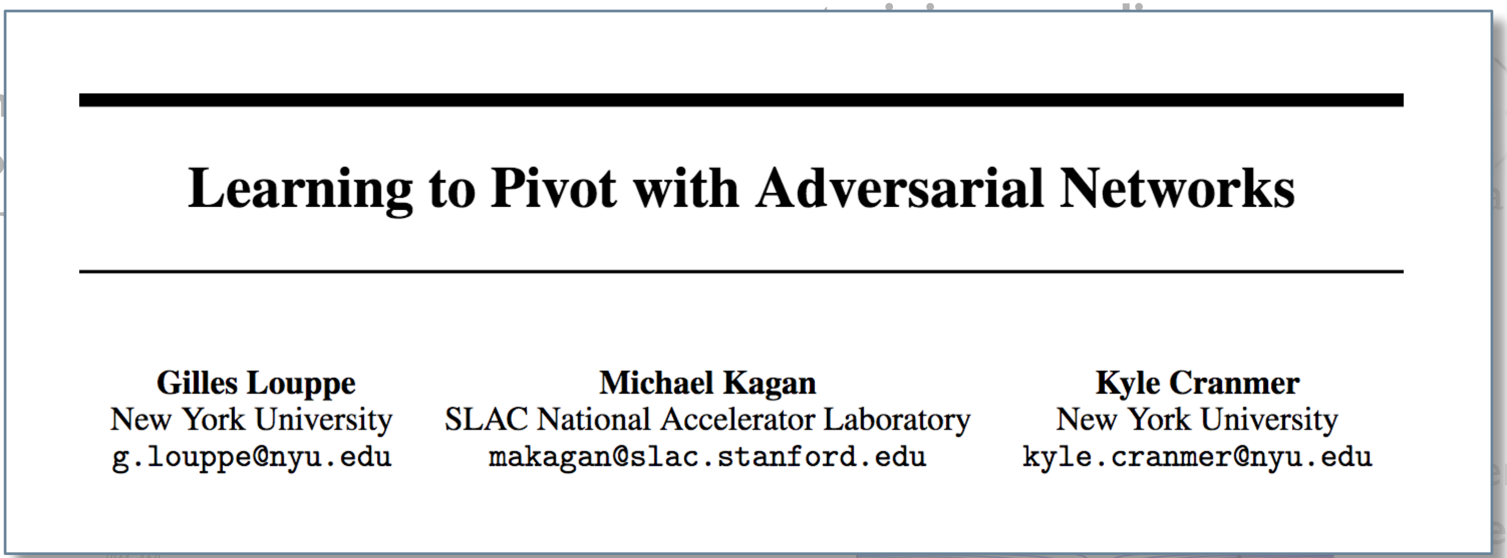
GPUs

IBM, ANL

Shutterstock



J. Schmidhuber



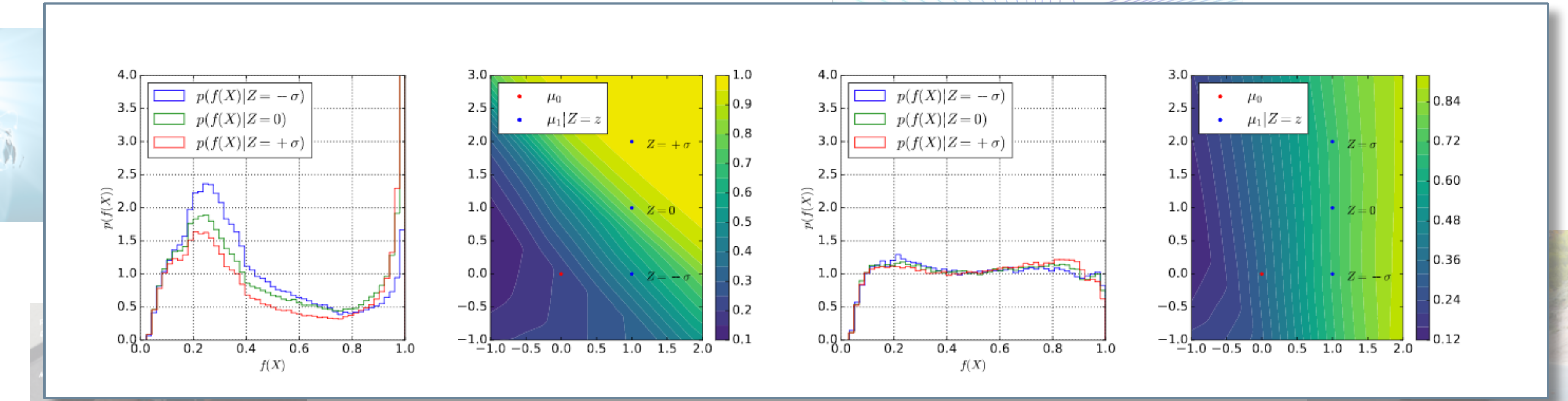
Learning to Pivot with Adversarial Networks

Gilles Louppe
New York University
g.louppe@nyu.edu

Michael Kagan
SLAC National Accelerator Laboratory
makagan@slac.stanford.edu

Kyle Cranmer
New York University
kyle.cranmer@nyu.edu

...r theoretical
...rstanding of
NNs and improved



Up-and-coming
advancements:
neuromorphic
hardware

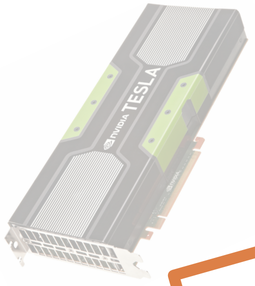
Google

e.g. physics application (HEP) → algorithmic development

. . . so, what is different now?

Increased computational capability enables more complicated NN architectures and faster training + larger data sets

GPUs

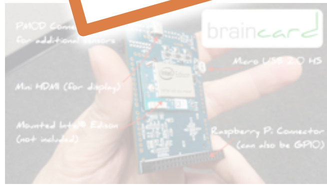


Accessibility

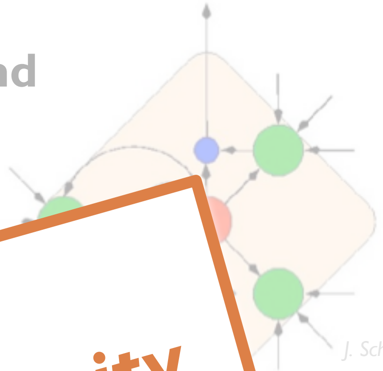
Shutterstock

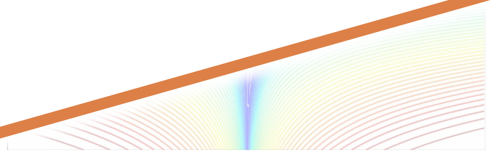
es)

Up-and-coming advancements: neuromorphic hardware



New network architectures and training paradigms, such as long short term memory (LSTM) networks, neural turing machines, and genera networks (G


J. Schmidhuber


A. Radford

ng of s and improved **optimization methods**

Applications have driven a lot of advancement (both algorithmic and practical/heuristic)

Google

→ much greater overall technological maturity

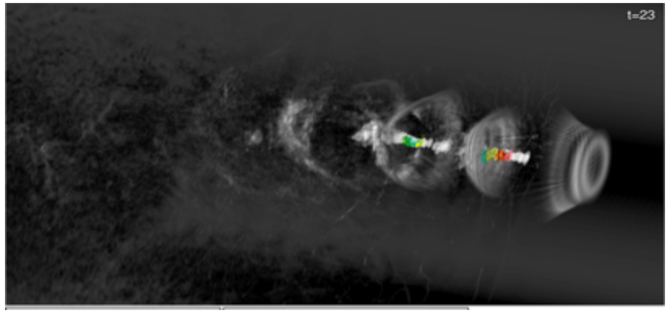
→ many advances in the last 3-5 years

Let's talk about accelerators…

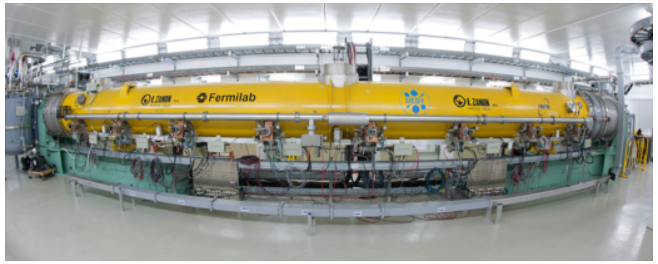
http://fast.fnal.gov/gallery.html

Interesting Technical Challenges

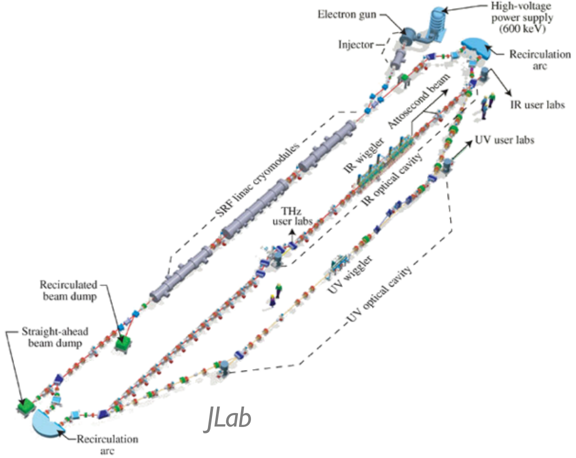
- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior



LBNL Visualization Group



Fermilab

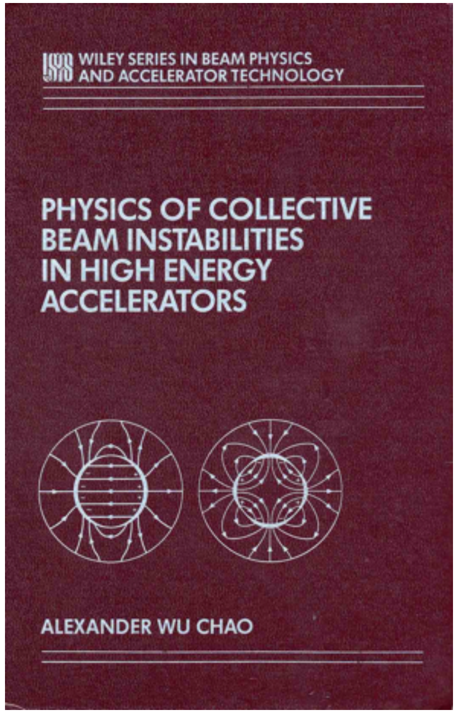


JLab

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:
→ of great interest for research in control and machine learning
→ lots of opportunity to both gain from and contribute to this area

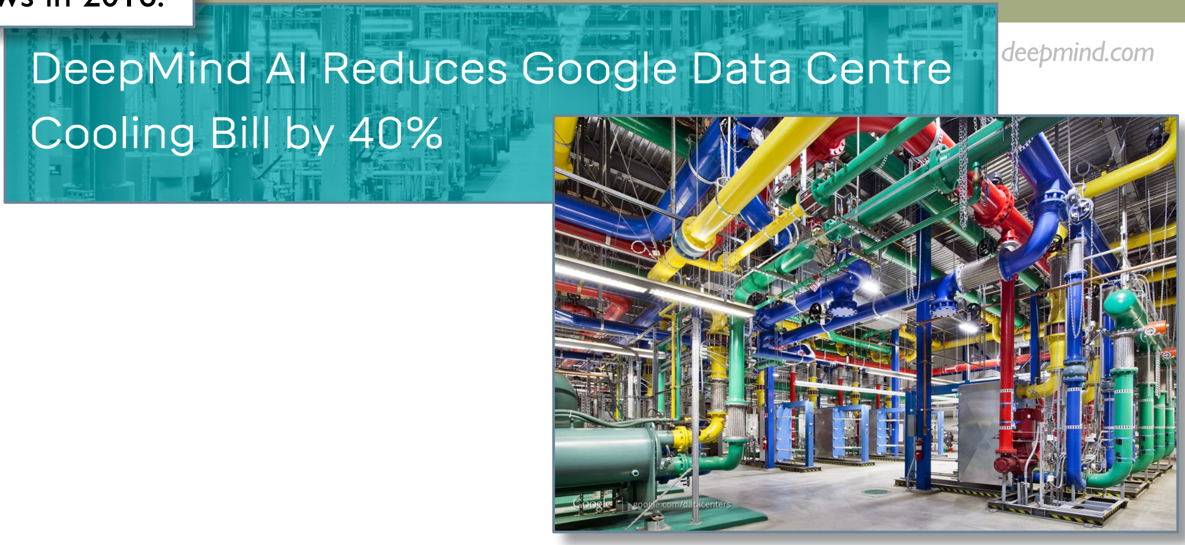

WILEY SERIES IN BEAM PHYSICS AND ACCELERATOR TECHNOLOGY

PHYSICS OF COLLECTIVE BEAM INSTABILITIES IN HIGH ENERGY ACCELERATORS

ALEXANDER WU CHAO

Interesting Technical Challenges

Global News in 2016:



DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

deepmind.com

https://googleblog.blogspot.com

- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:

→ of great interest for research in control and machine learning

→ lots of opportunity to both gain from and contribute to this area

Interesting Technical Challenges

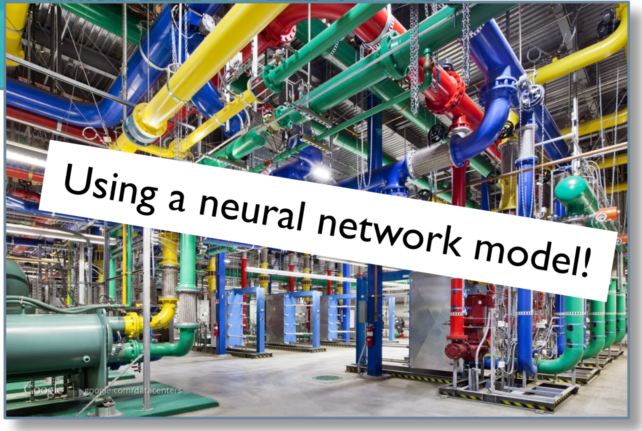
- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior

Global News in 2016:

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

deepmind.com

Transport delays, variable heat load
Efficient servers alone not enough



Using a neural network model!

https://googleblog.blogspot.com

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:
→ of great interest for research in control and machine learning
→ lots of opportunity to both gain from and contribute to this area

Interesting Technical Challenges

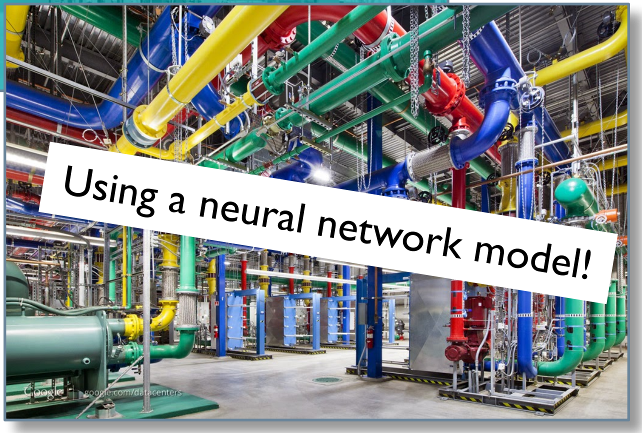
- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior

Global News in 2016:

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

deepmind.com

Transport delays, variable heat load
Efficient servers alone not enough


Using a neural network model!

https://googleblog.blogspot.com

Work at FNAL during 2014:

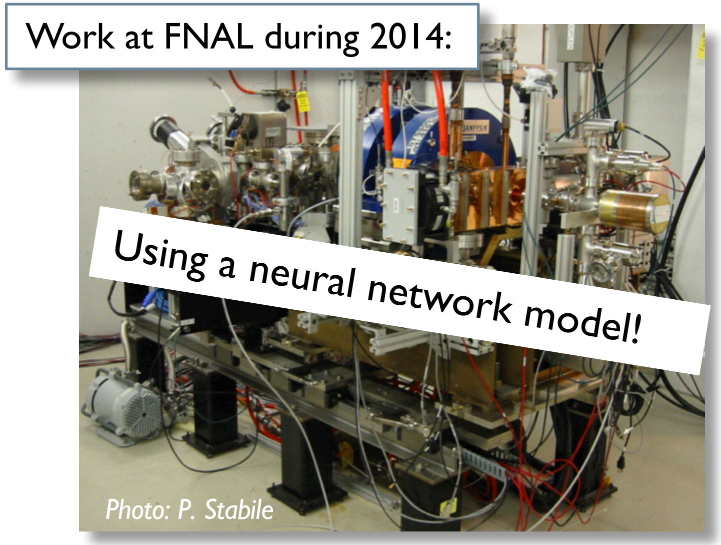

Using a neural network model!

Photo: P. Stabile

A. L. Edelen, et al. IPAC15 ,TUPOA51

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:

→ of great interest for research in control and machine learning

→ lots of opportunity to both gain from and contribute to this area

Interesting Technical Challenges

- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:

→ of great interest for research in control and machine learning

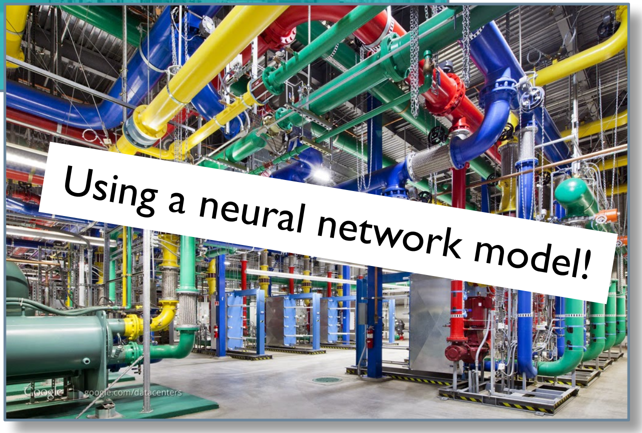
→ lots of opportunity to both gain from and contribute to this area

Global News in 2016:

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

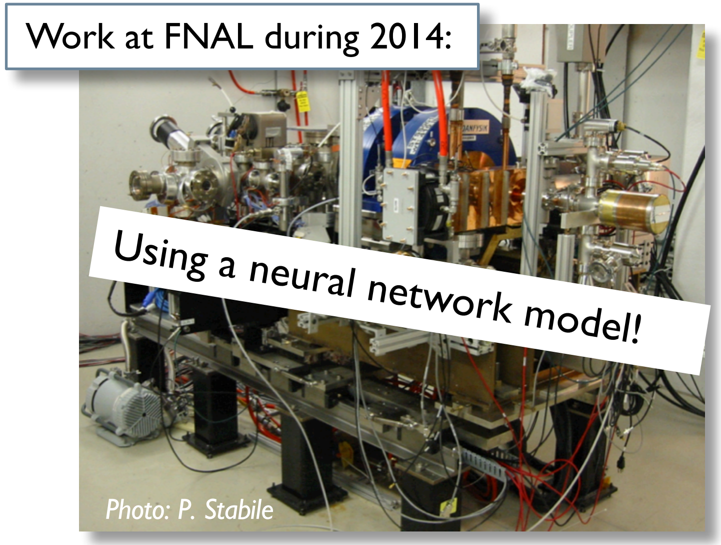
deepmind.com

Transport delays, variable heat load
Efficient servers alone not enough



https://googleblog.blogspot.com

Work at FNAL during 2014:



A. L. Edelen, et al. IPAC15 ,TUPOA51

Looks vaguely familiar…

Transport delays, variable heat load, complex dynamics

Interesting Technical Challenges

- Complex/nonlinear dynamics
- Many small, compounding errors
- Many parameters to monitor and control
- Interacting sub-systems
- On-demand changes in operational state
- Diagnostics sometimes limited or not put to full use in control (e.g. images)
- Time-varying/ non-stationary behavior

Strong Incentives for Better Control

- Cost of running → Time/energy efficiency of control
- Cost of unintended down-time → Personnel cost, user time, bulk scientific output
- Achieving performance needed for science goals and other applications
  - *improving accelerator components and control both play a role*

Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:

→ of great interest for research in control and machine learning

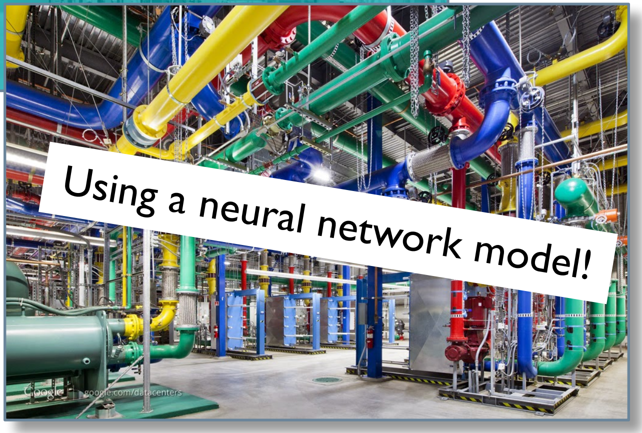
→ lots of opportunity to both gain from and contribute to this area

Global News in 2016:

DeepMind AI Reduces Google Data Centre Cooling Bill by 40%

deepmind.com

Transport delays, variable heat load
Efficient servers alone not enough



https://googleblog.blogspot.com

Work at FNAL during 2014:

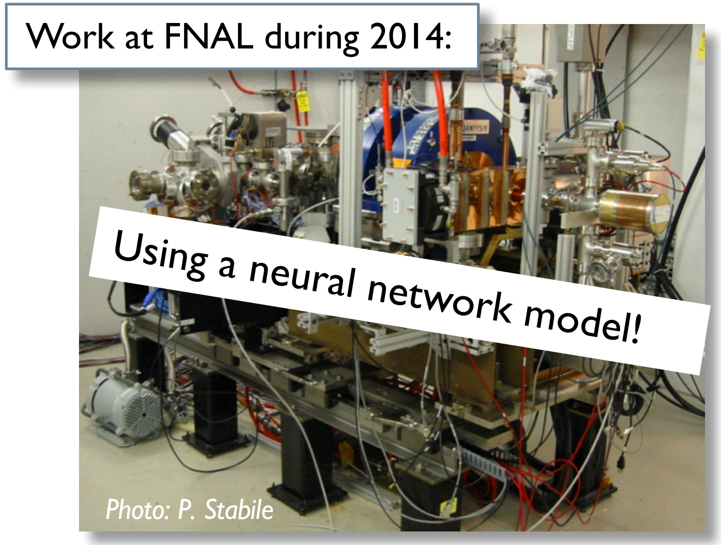


Photo: P. Stabile

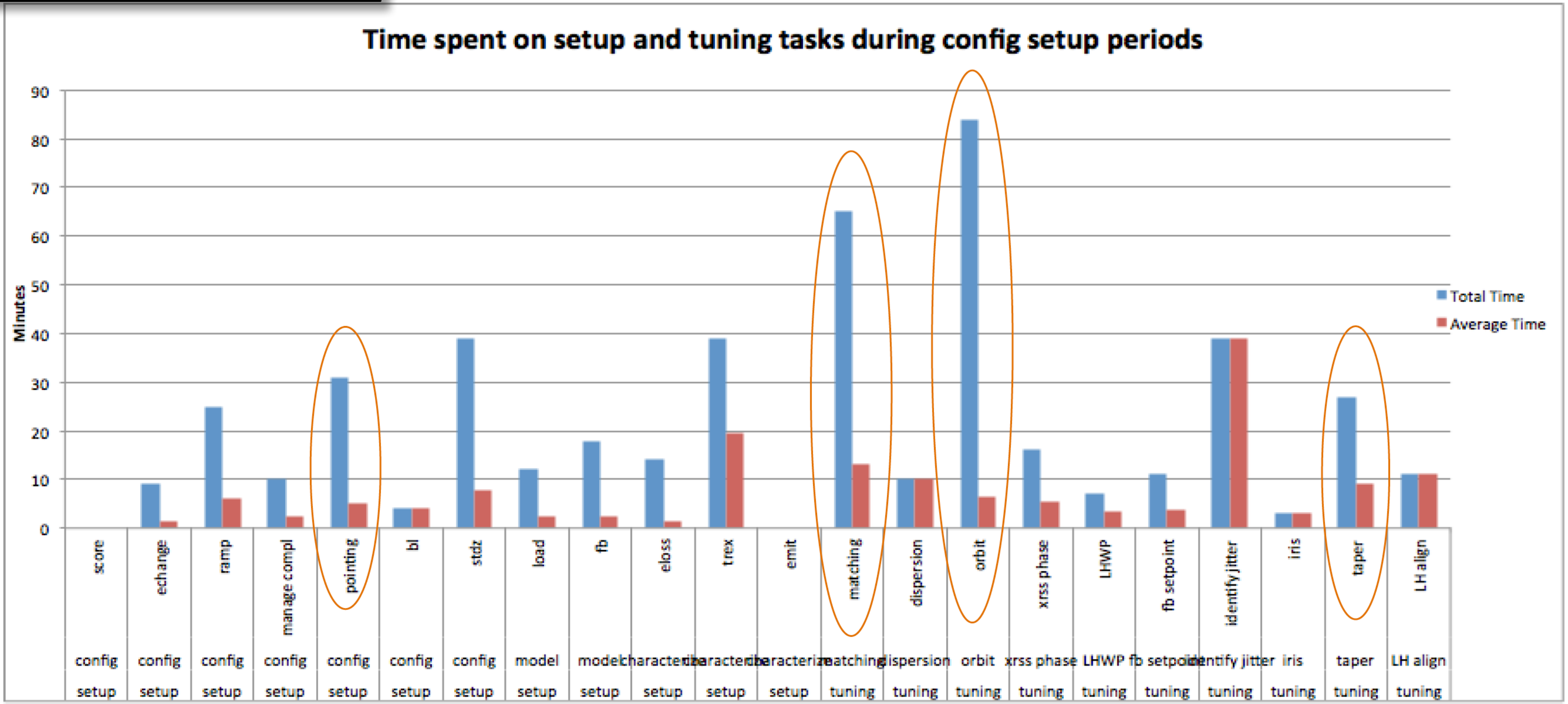
A. L. Edelen, et al. IPAC15 ,TUPOA51

Looks vaguely familiar…

Transport delays, variable heat load, complex dynamics

Cryo plant photo: A. Grassellino talk at IPAC '17, (THPPA2)

Example from LCLS



2015: 450 hand tuning hours, 250 dedicated!

⇒ Lots of opportunity to speed operations and relieve operator load

We rely heavily on operators for day-to-day control tasks . . .

Fermilab Control Room Photo: Reidar Hahn, FNAL

. . . so what can we learn from them,

and what analogous techniques can we use?

Inspiration from Operators

Model Learning

Prediction

Planning

Diagnostic Analysis

Optimization

Learning Control

Fermilab Control Room Photo: Reidar Hahn, FNAL

Application Areas for Accelerators

- Online modeling → *NN model*
- Time delays → *model predictive control + NN models*
- Image-based diagnostics → *convolutional or locally-connected NNs*
- Frequent switching between operating conditions → *NN policy*
- Virtual diagnostics → *NN model trained from intercepting diagnostics or simulation*
- Encode an existing policy and/or adapt upon it → *NN policy*
- High-level assessment of machine or device states → *NN process model, classifier*
- Failure prediction / Anomaly detection → *NN process model, classifier*

Online Modeling

- Operators maintain a learned mental machine model: *let's supplement it*

This can be very hard!

- Ideally:
  - Fast-executing, but accurate enough to be useful
  - Use measured inputs directly from machine
  - Combine *a priori* knowledge + learned parameters

- Applications
  - A tool for operators + virtual diagnostics
  - Predictive control
  - Help flag aberrant behavior

One approach: faster modeling codes

- Simpler models (tradeoff with accuracy)
- Parallelization and GPU-acceleration of existing codes

PARMILA	*X. Pang, PAC13, MOPMA13*
elegant	*I.V. Pogorelov, et al., IPAC15, MOPMA035*

- Improvements in underlying modeling algorithms

(fractions of a second)

Another approach: machine learning model

- Once trained, neural networks can execute quickly
- Train on slow, high-fidelity simulation results
- Also train on measured results

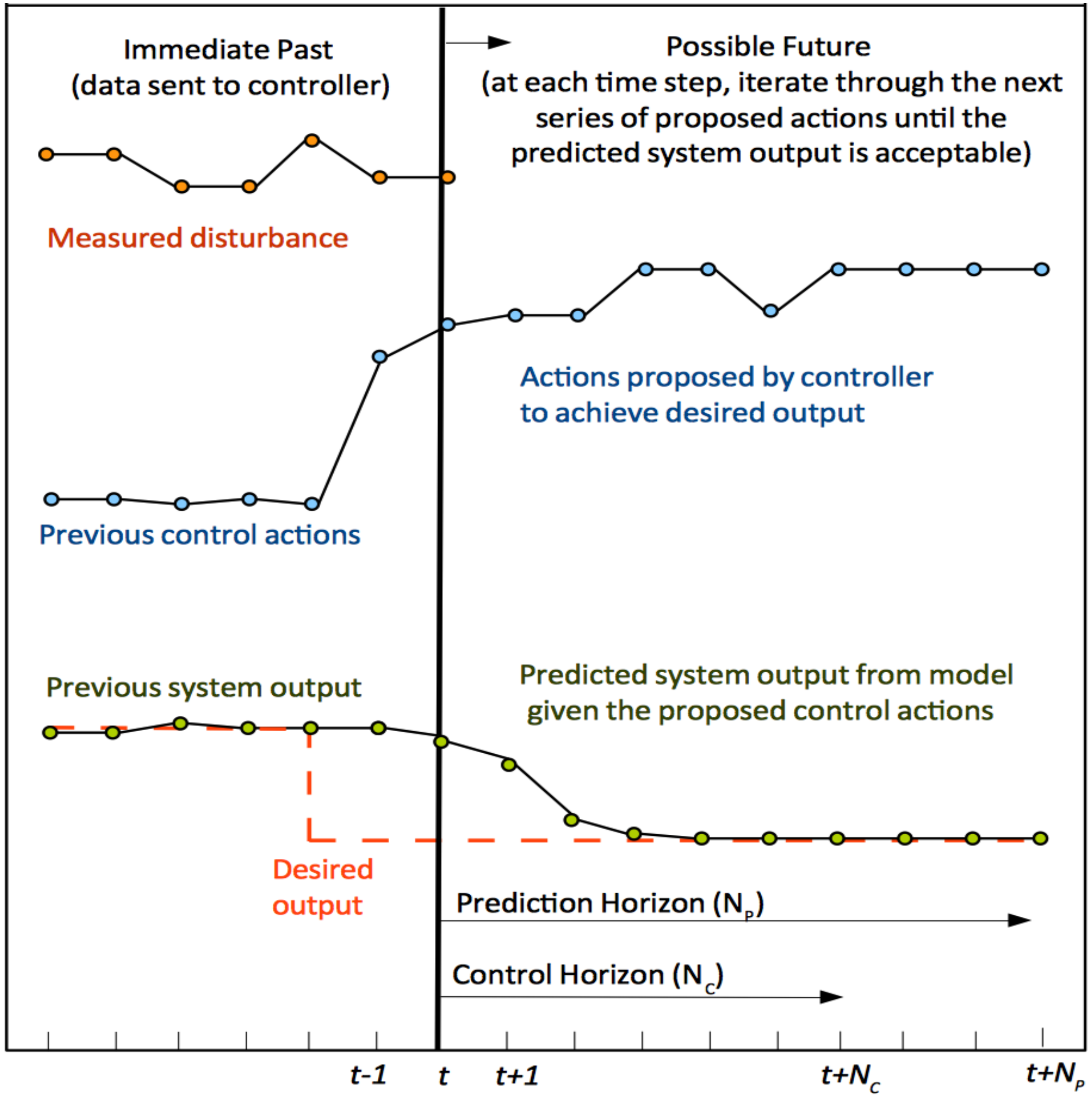
Yields a fast-executing model that can be used operationally, but approximates behavior from high-fidelity simulations (e.g. PIC codes, LPA)

An initial study involving this at FAST:

A. L. Edelen, et al. NAPAC16, TUPOA51

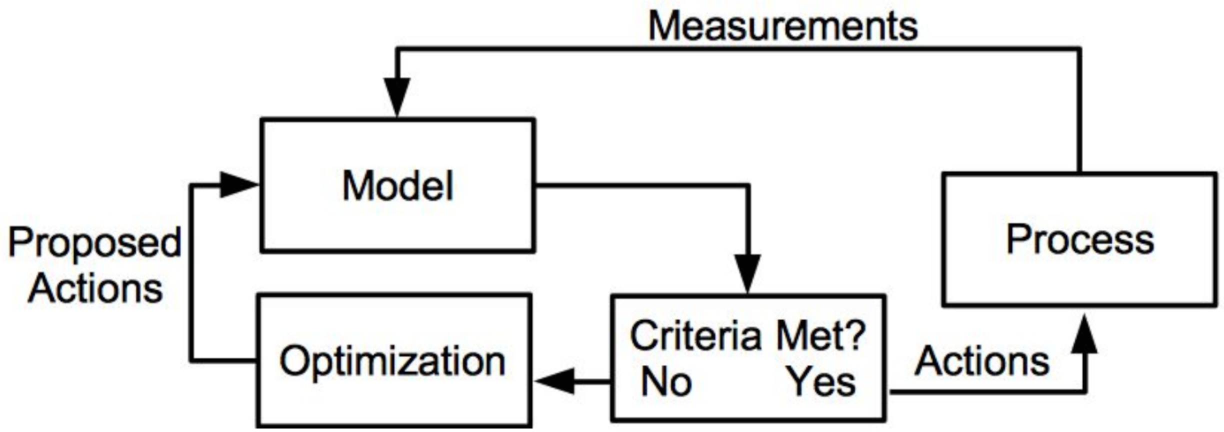
one PARMELA run: ~20 min

Model Predictive Control (Prediction + Planning)



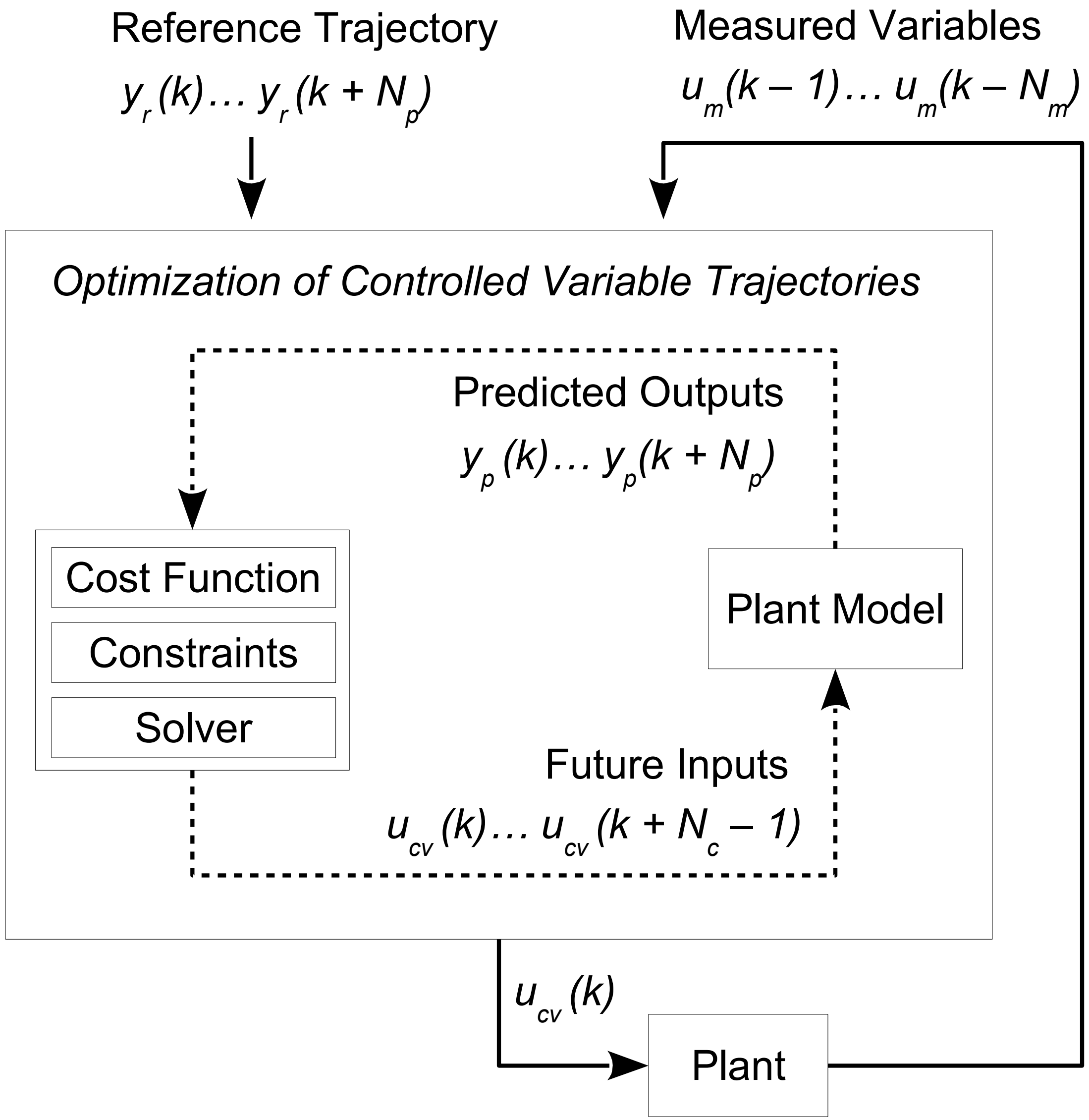
Basic concept:

1. Use a predictive model to assess the outcome of possible future actions
2. Choose the best series of actions
3. Execute the first action
4. Gather next time step of data
5. Repeat



Auralee Edelen May 2017

Model Predictive Control (Prediction + Planning)



Reference Trajectory
$y_r(k) \ldots y_r(k+N_p)$
Measured Variables
$u_m(k-1) \ldots u_m(k-N_m)$
Optimization of Controlled Variable Trajectories
Predicted Outputs
$y_p(k) \ldots y_p(k+N_p)$
Cost Function
Constraints
Solver
Plant Model
Future Inputs
$u_{cv}(k) \ldots u_{cv}(k+N_c-1)$
$u_{cv}(k)$
Plant

N_m previous measurements

N_p future time steps predicted

N_c future time steps controlled

$$\sum_{i=1}^{N_p} \{w_y\,[y_r(k+i) - y_p(k+i)]\}^2$$

(output variable targets)

$$\sum_{j=1}^{ncv} \sum_{i=0}^{N_p-1} \{w_{u,j}[u_j(k+i) - u_{j,ref}(k+i)]\}^2$$

(controllable variable targets)

$$\sum_{j=1}^{ncv} \sum_{i=0}^{N_p-1} \{w_{\Delta u,j}\,[u_j(k+i) - u_j(k+i-1)]\}^2$$

(movement size)

Temperature Control for the RF Photoinjector at FAST

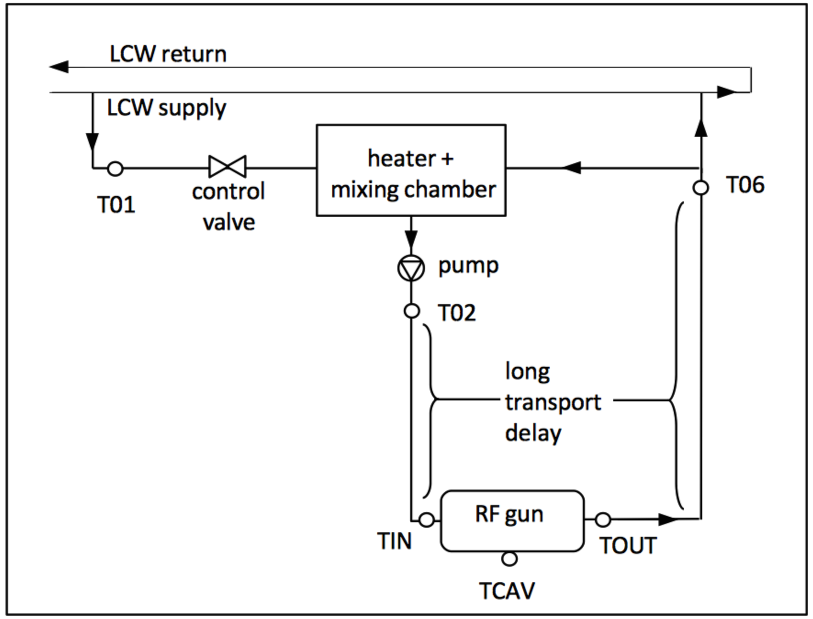
Resonant frequency controlled via temperature

PID control is undesirable in this case:

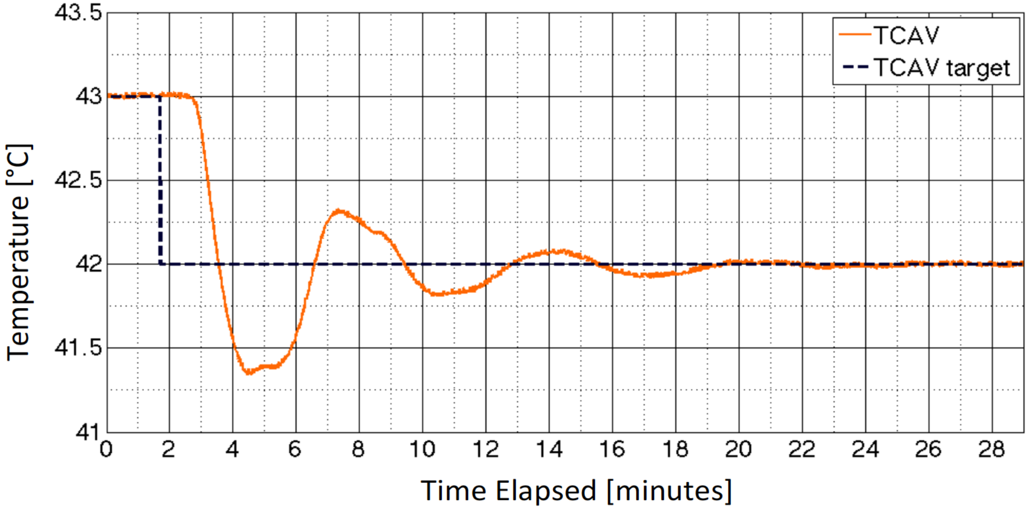
- Long transport delays and thermal responses
- Recirculation leads to secondary impact of disturbances
- Two controllable variables: heater power + valve aperture

Applied model predictive control (MPC) with a neural network model trained on measured data: ~ 5x faster settling time + no large overshoot

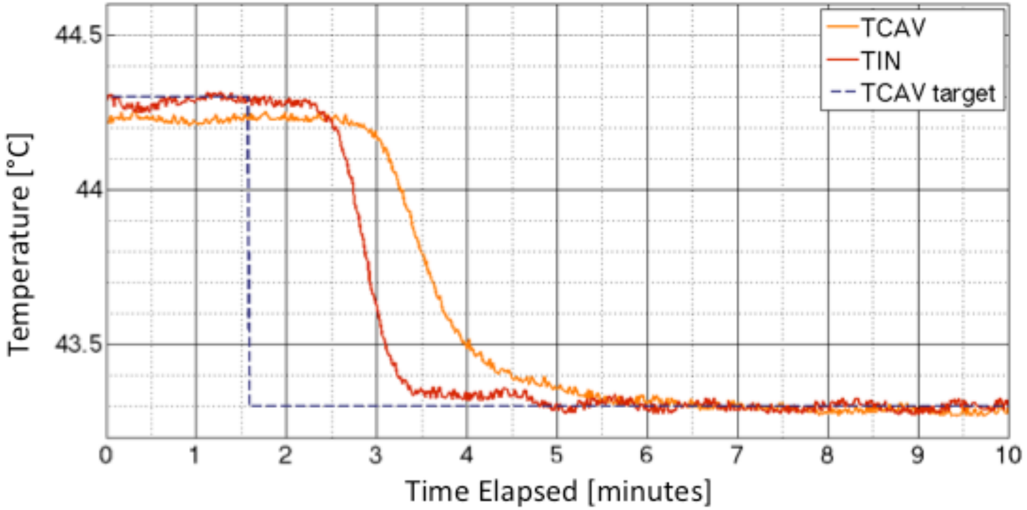
Gun Water System Layout



Existing Feedforward/PID Controller



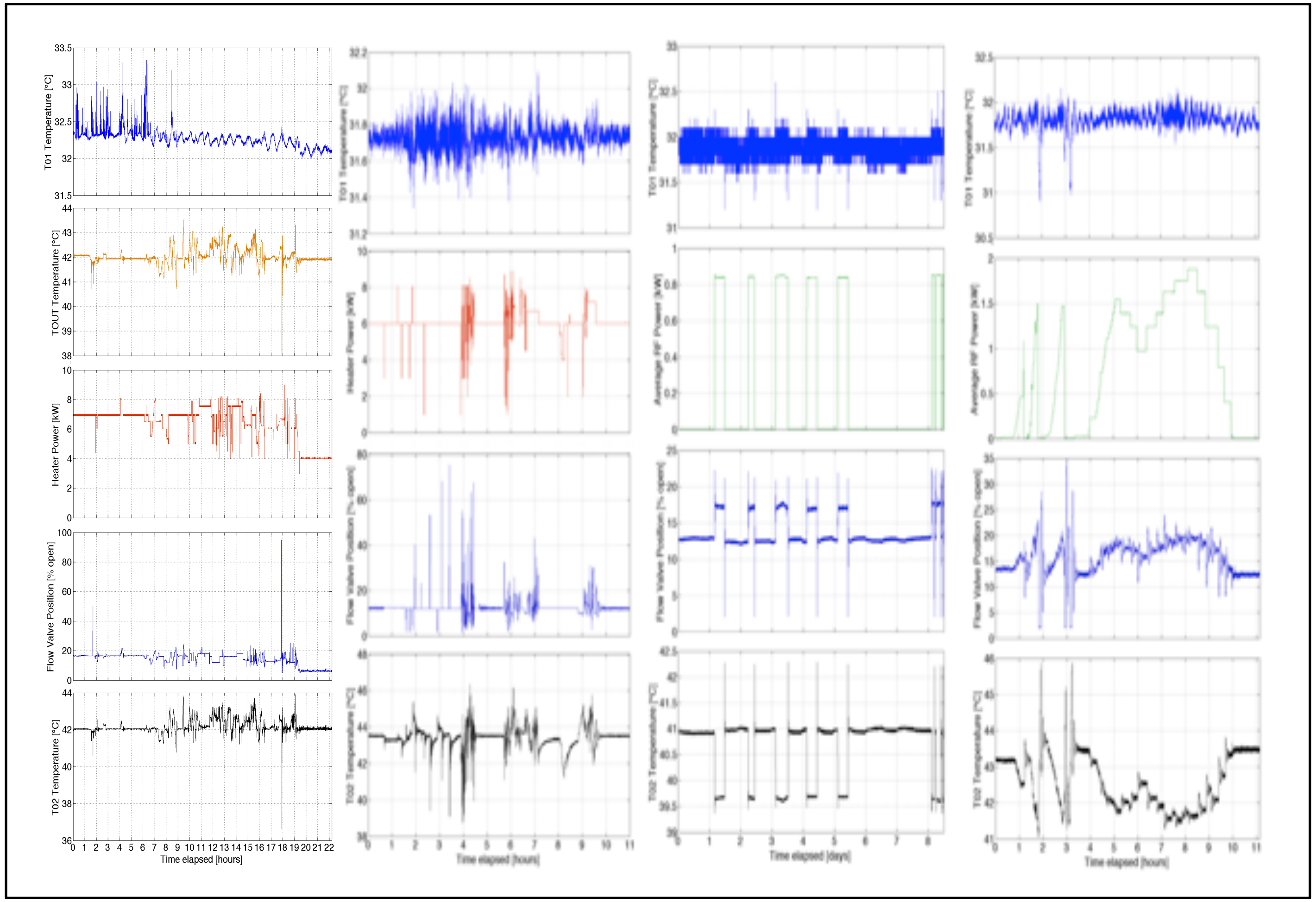
Model Predictive Controller



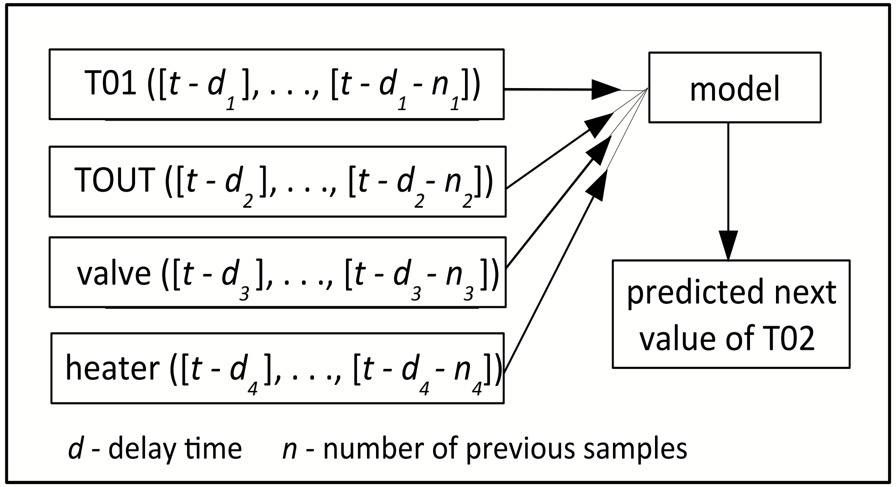
Note that the oscillations are largely due to the transport delays and water recirculation, rather than PID gains

More info: A. L. Edelen. IEEE TNS, vol. 63, no. 2, 2016

Neural Network Model



Training data from machine



T01 ([$t - d_1$], . . ., [$t - d_1 - n_1$])

TOUT ([$t - d_2$], . . ., [$t - d_2 - n_2$])

valve ([$t - d_3$], . . ., [$t - d_3 - n_3$])

heater ([$t - d_4$], . . ., [$t - d_4 - n_4$])

model

predicted next value of T02

d - delay time *n* - number of previous samples

NN model

Why does this matter (for resonant frequency control in general)?

LLRF system will compensate for detuning by increasing forward power

But…

- Ability to do this bounded by the amplifier specs
- RF overhead adds to initial machine cost and footprint
- Using additional RF power → *increasing operational cost*
- Increased waste heat into cooling system → *increasing operational cost*
- If detuned beyond overhead → *interrupt normal operations (beam not properly accelerated or LLRF in frequency-tracking mode)*

PIP-II Injector Test RFQ

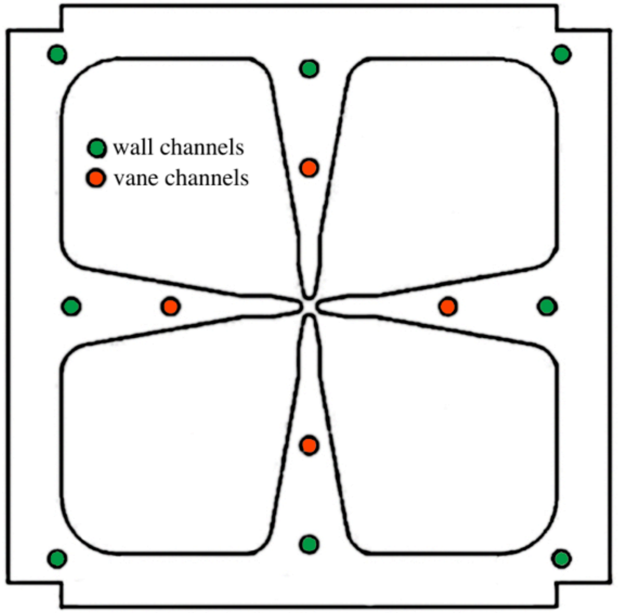
Variable heating

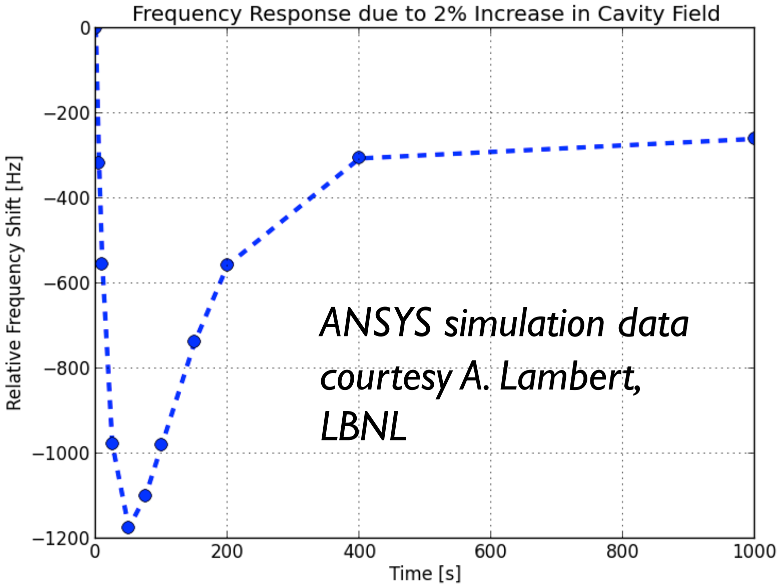
Specification for GDR: 3-kHz maximum frequency shift

Range of RF duty factors and pulse patterns (including up to CW)

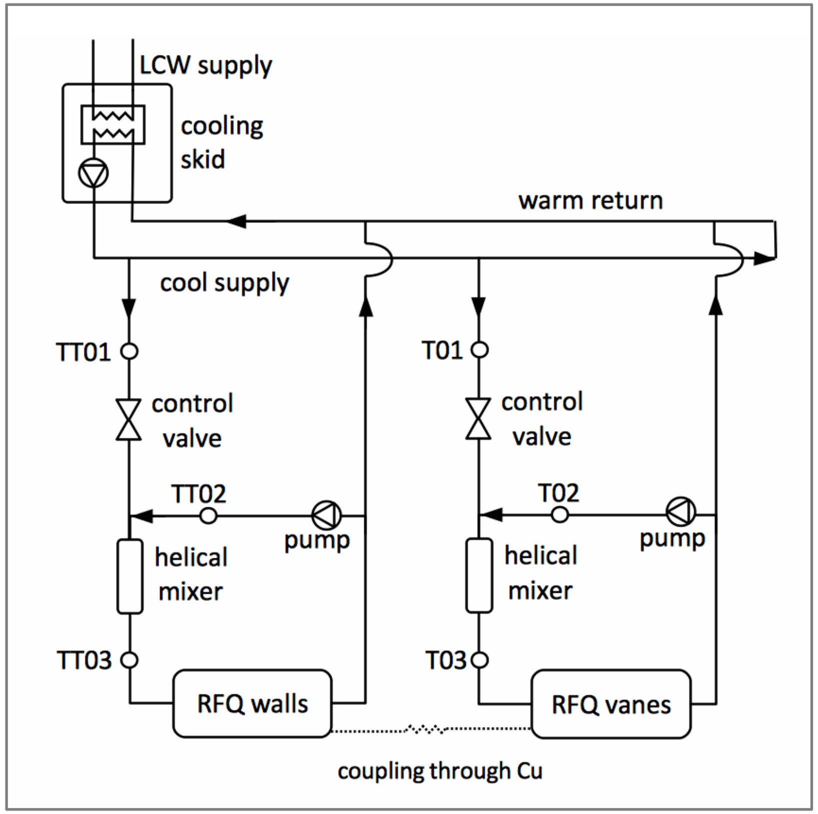
-16.7 kHz/°C in the vanes and 13.9 kHz/°C in the walls*

** A. R. Lambert et al., IPAC'15, WEPTY045*

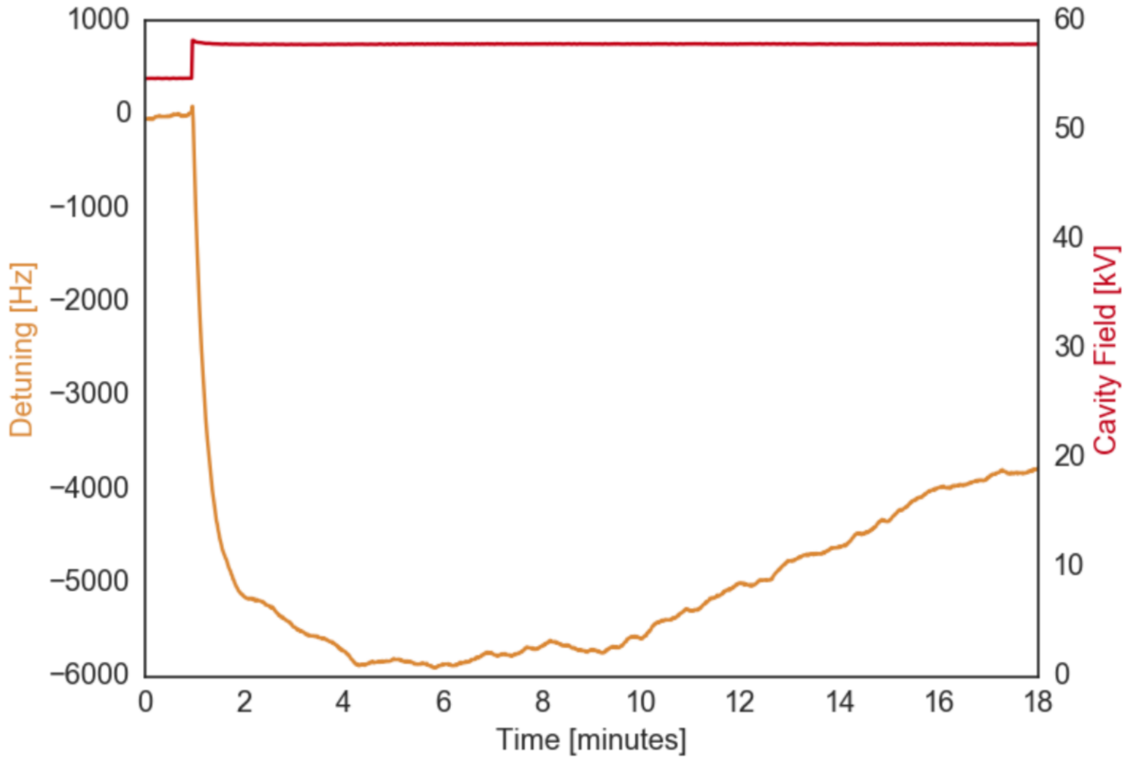




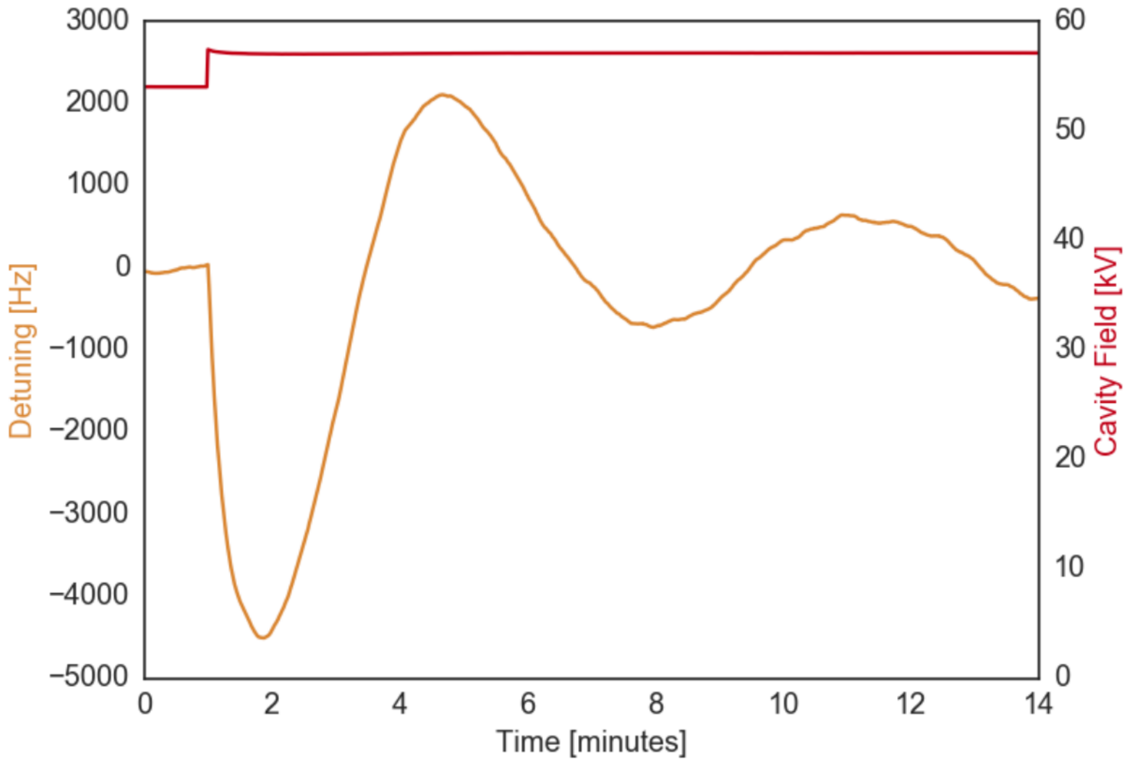
ANSYS simulation data courtesy A. Lambert, LBNL



RFQ Detuning in CW Mode



Example of uncontrolled detuning in CW mode under a small change in cavity field (55 kV to 58 kV)



PI frequency control in CW operation under a small change in cavity field (55 kV to 58 kV)

What about a simple first-principles model, or a learned linear model?

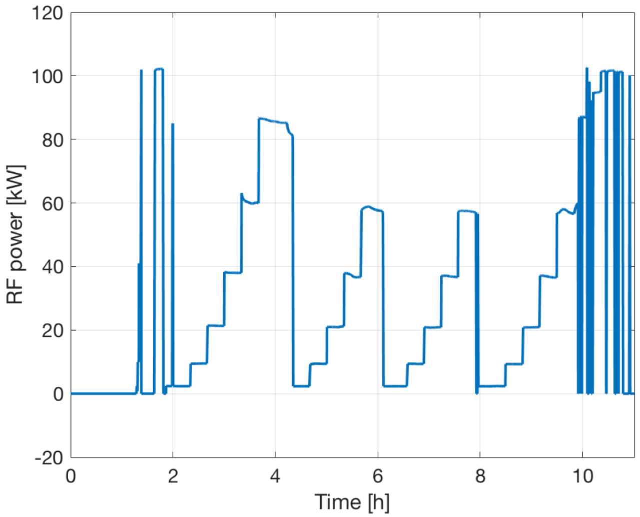
measured input data → first-principles model   4 ms pulse duration, 10 Hz rep rate   variety of valve and power settings

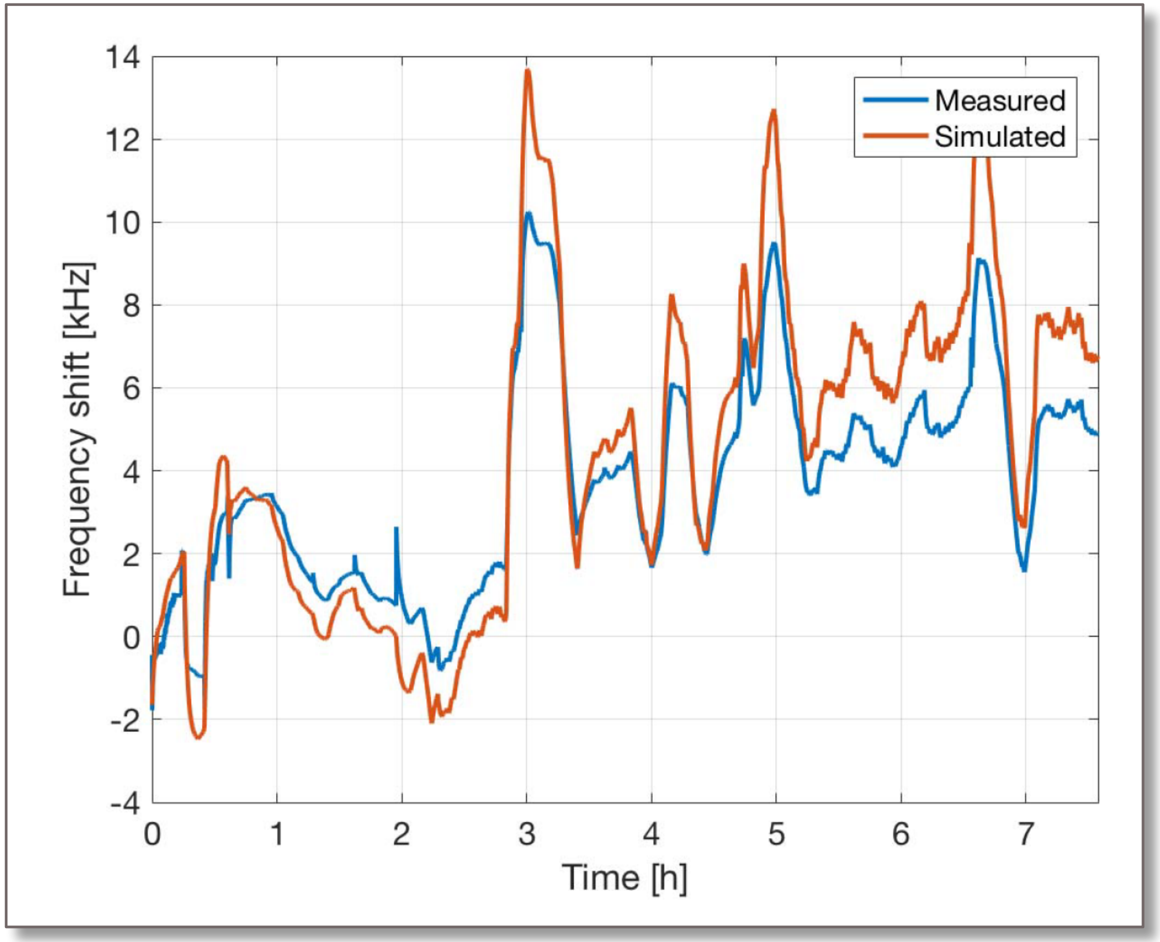
1.67 kHz RMS error

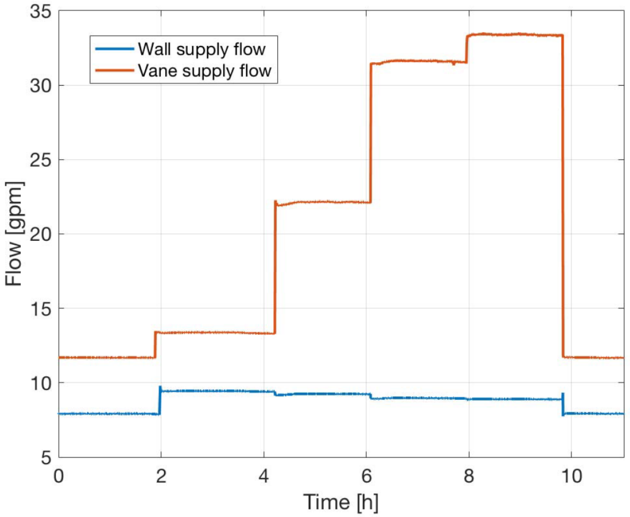
4.01 kHz max error

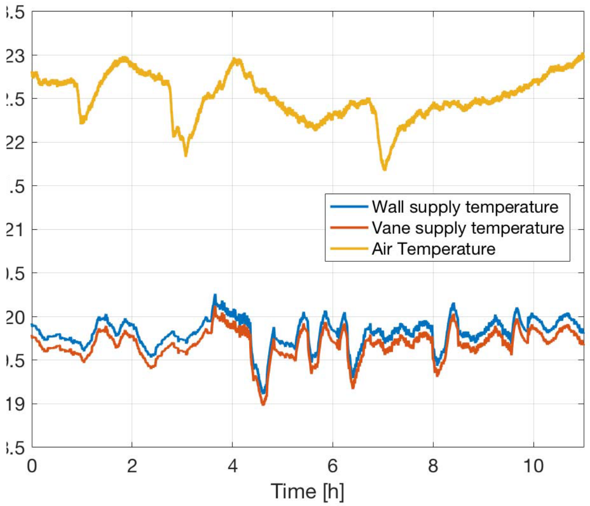
↓

not good enough!



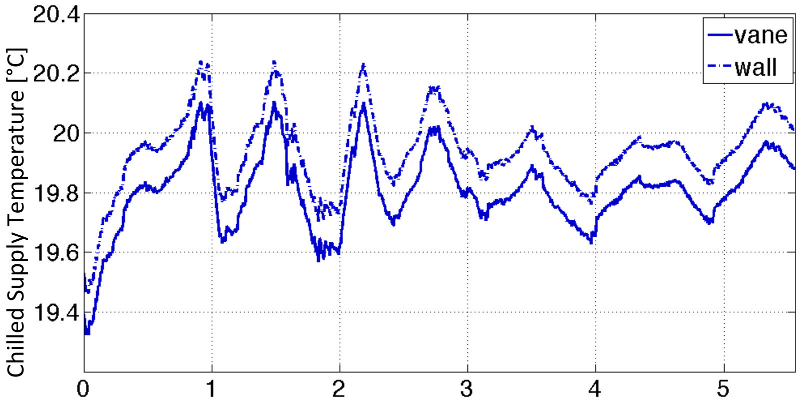


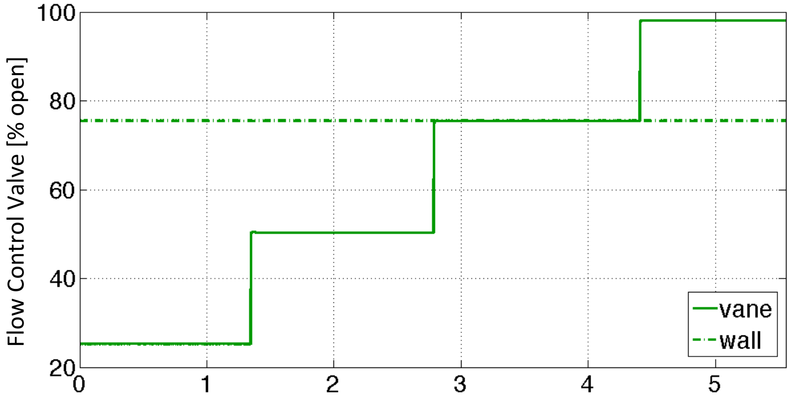


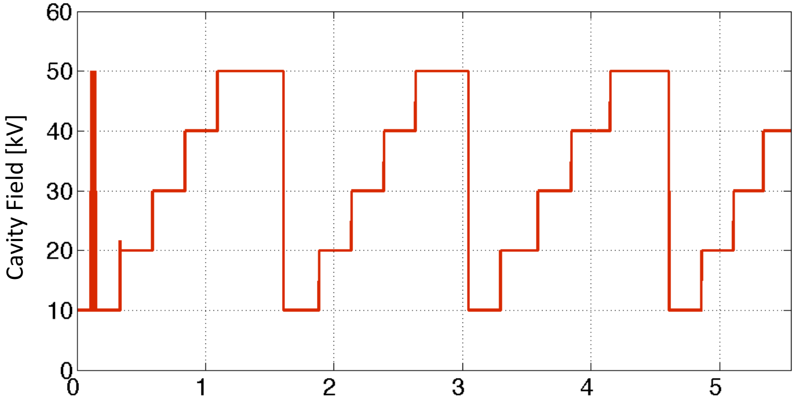


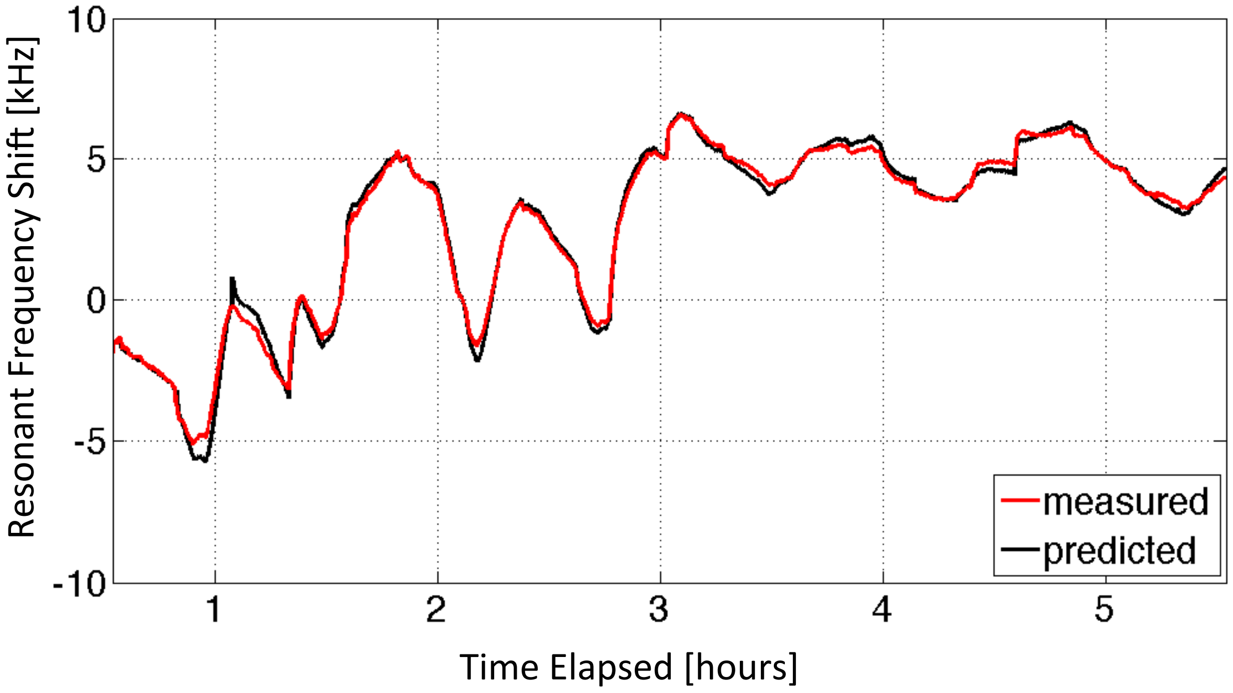
J. Edelen, A. Edelen, et al. TNS 64, vol. 2, (2017)

Neural Network Model









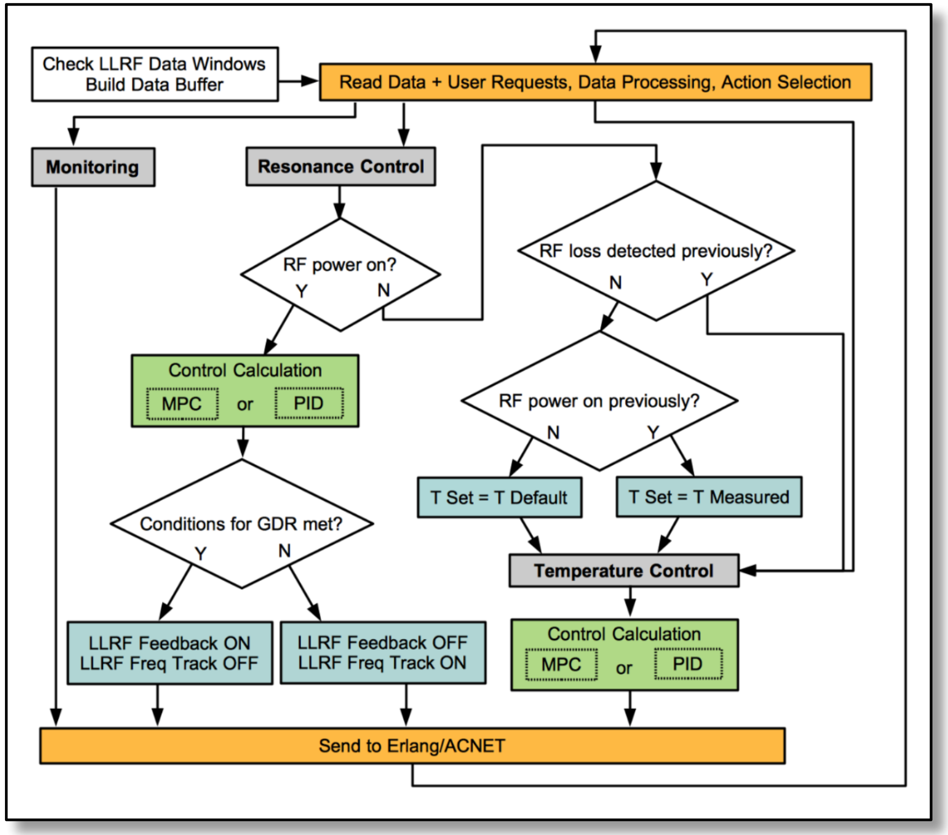
Inputs

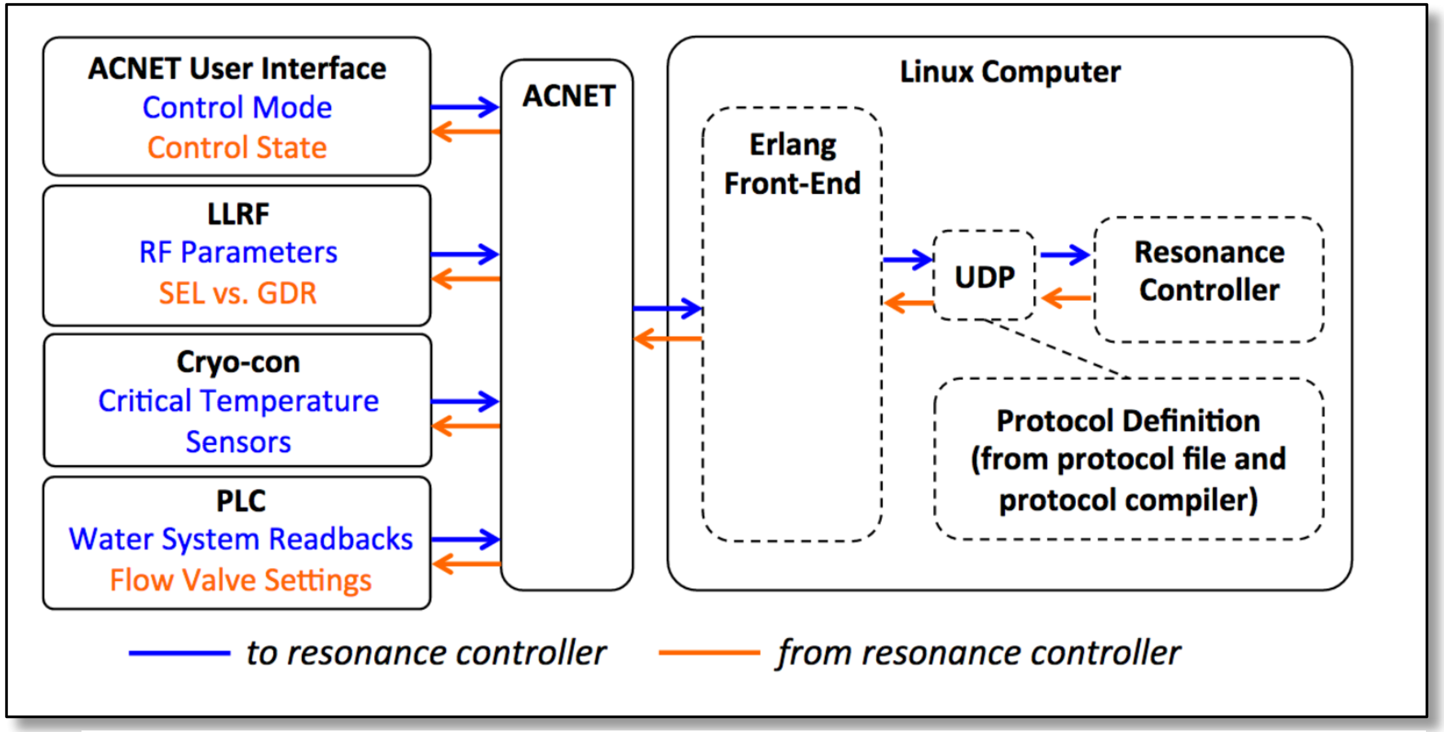
Vane and wall valve settings
Average RF power
Water temperatures
Ambient temperature and humidity

Mean Absolute Prediction Error

346 Hz on the test set
98 Hz on the validation set
115 Hz across all sets

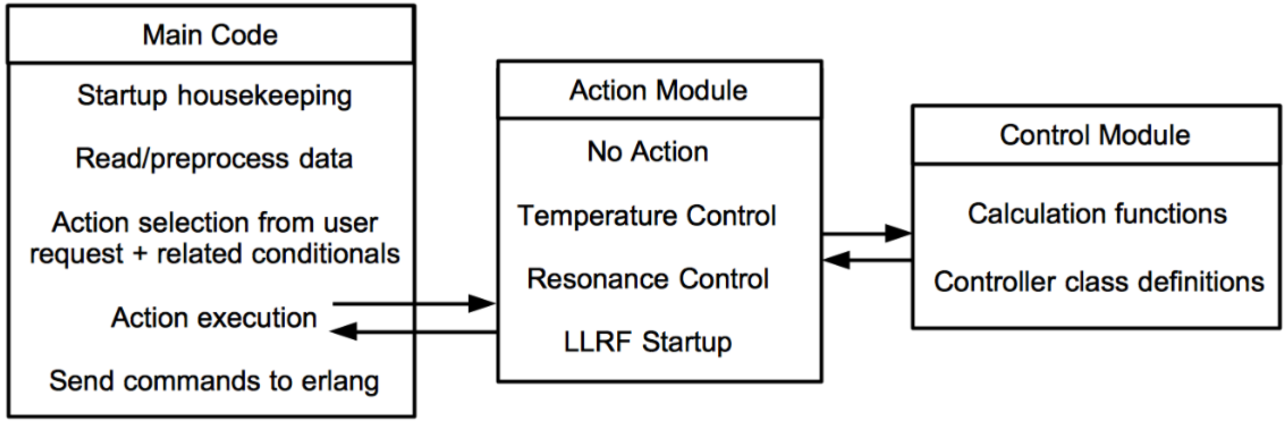
A. Edelen, et al., IPAC '16, THPOY020





Built a *python-based control framework*

- *Executes on controls network linux computer*
- *PI control in regular operational use*
- *Preparing for test of MPC*
- *Designed to be portable + modular*



Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don't exist

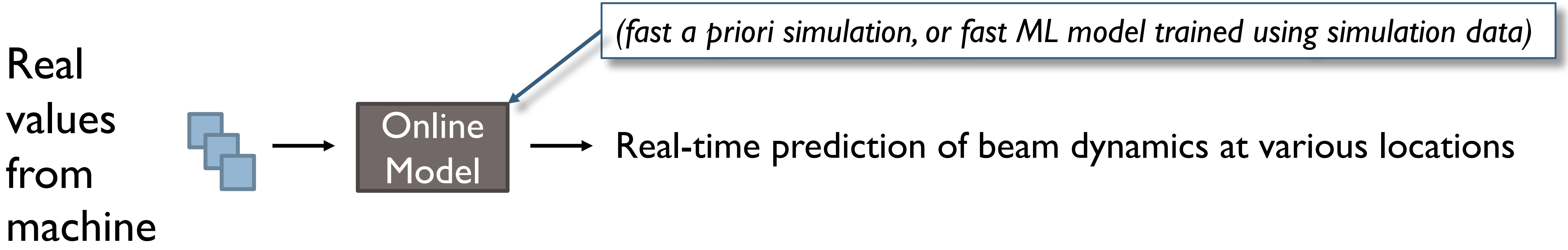
Real values from machine

Online Model

Real-time prediction of beam dynamics at various locations

Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don't exist



Real values from machine

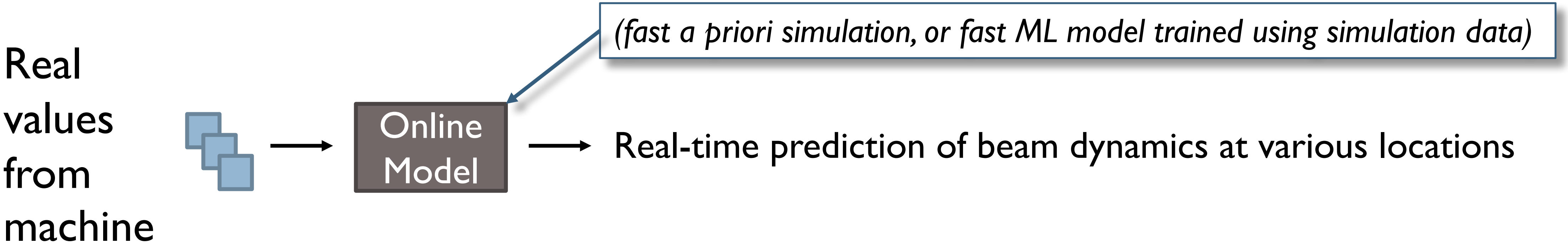
Online Model

(fast a priori simulation, or fast ML model trained using simulation data)

Real-time prediction of beam dynamics at various locations

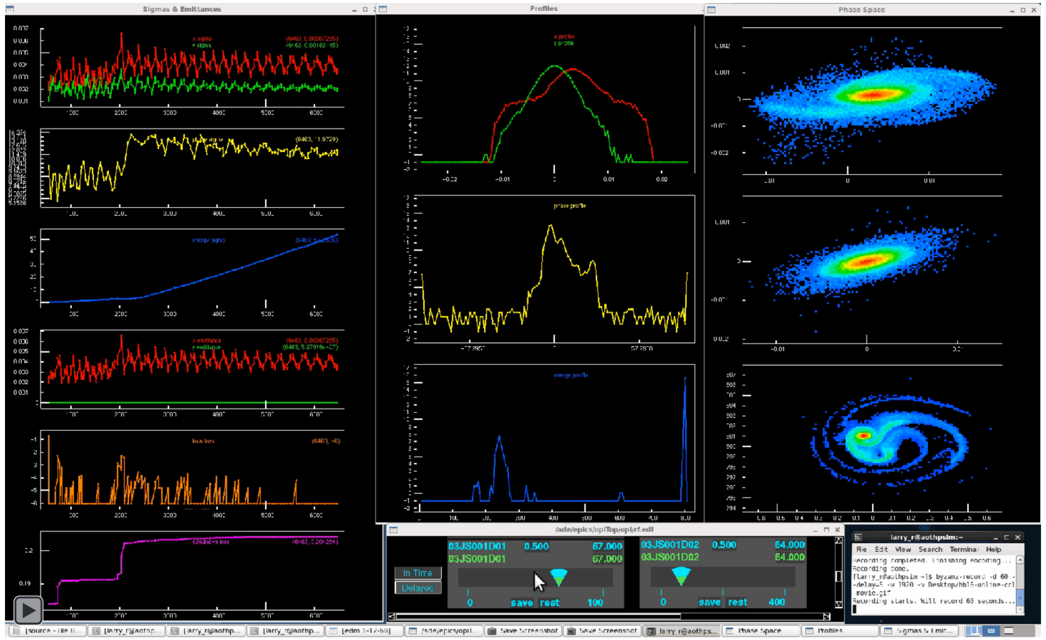
Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don't exist



Real values from machine → Online Model → Real-time prediction of beam dynamics at various locations

(fast a priori simulation, or fast ML model trained using simulation data)



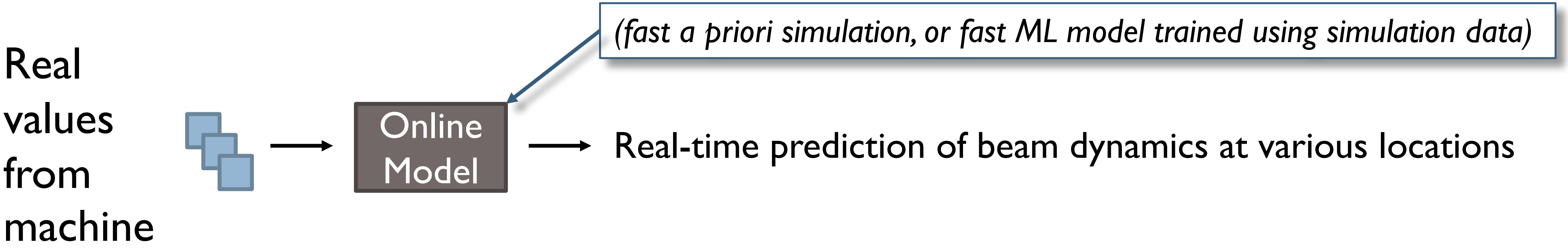
```
Recording completed. Finishing encoding...
Recording done.
[larry_r@aothpsim ~]$ byzanz-record -d 60 -
-delay=5 -w 1920 -v Desktop/hb16-online-crl
movie.gif
Recording starts. Will record 60 seconds...
```

e.g. GPU-accelerated PARMILA at LANSCE

X. Pang, et al., PAC13, MOPMA13
X. Pang, IPAC15, WEXC2
X. Pang and L. Rybarcyk, CPC185, is. 3 (2014)
L. Rybarcyk, et al., IPAC15, MOPWI033
L. Rybarcyk, HB2016, WEPM4Y01

Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don't exist



Real values from machine

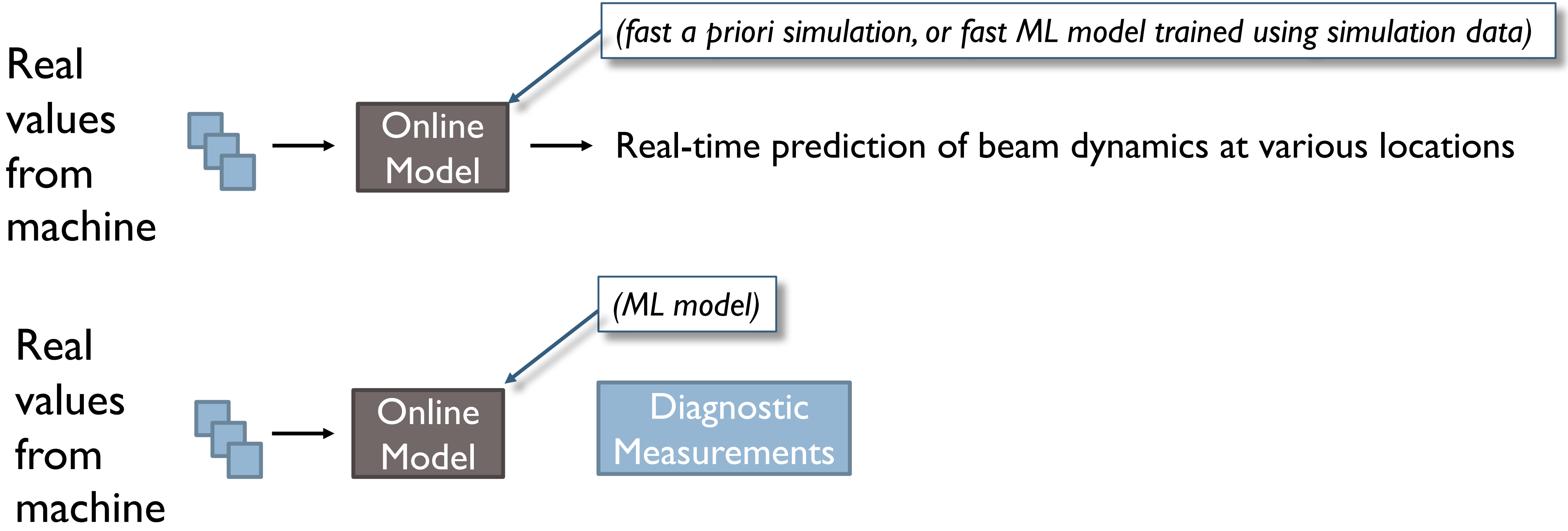
Online Model

Real-time prediction of beam dynamics at various locations

(fast a priori simulation, or fast ML model trained using simulation data)

Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don't exist



Real values from machine → Online Model → Real-time prediction of beam dynamics at various locations

(fast a priori simulation, or fast ML model trained using simulation data)

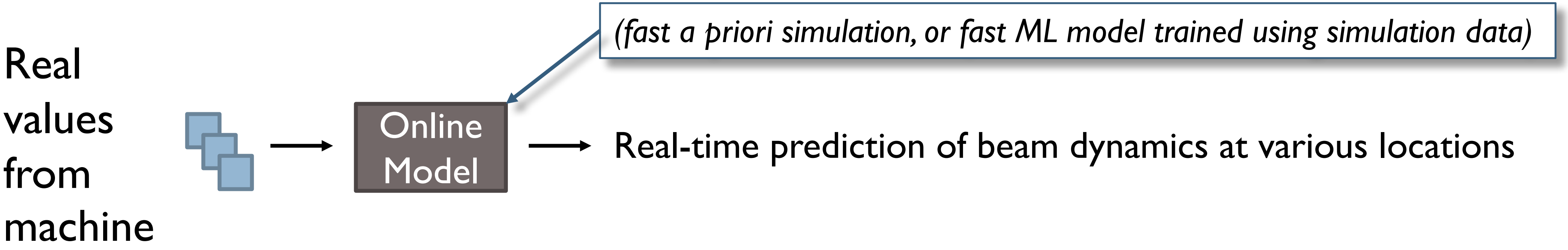
Real values from machine → Online Model

(ML model)

Diagnostic Measurements

Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don’t exist

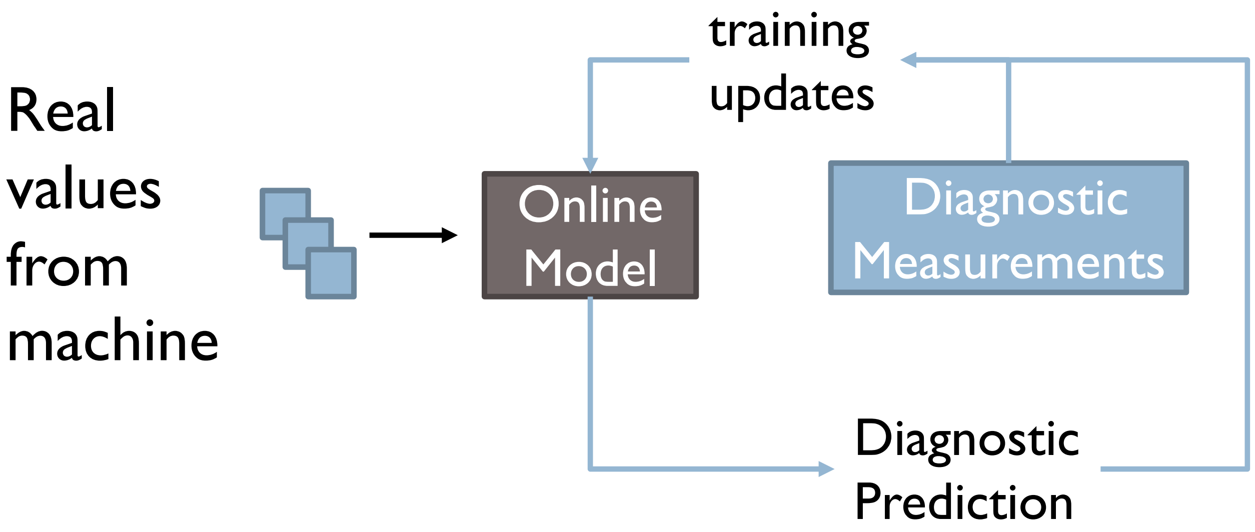


Real values from machine

Online Model

(fast a priori simulation, or fast ML model trained using simulation data)

Real-time prediction of beam dynamics at various locations



Real values from machine

Online Model

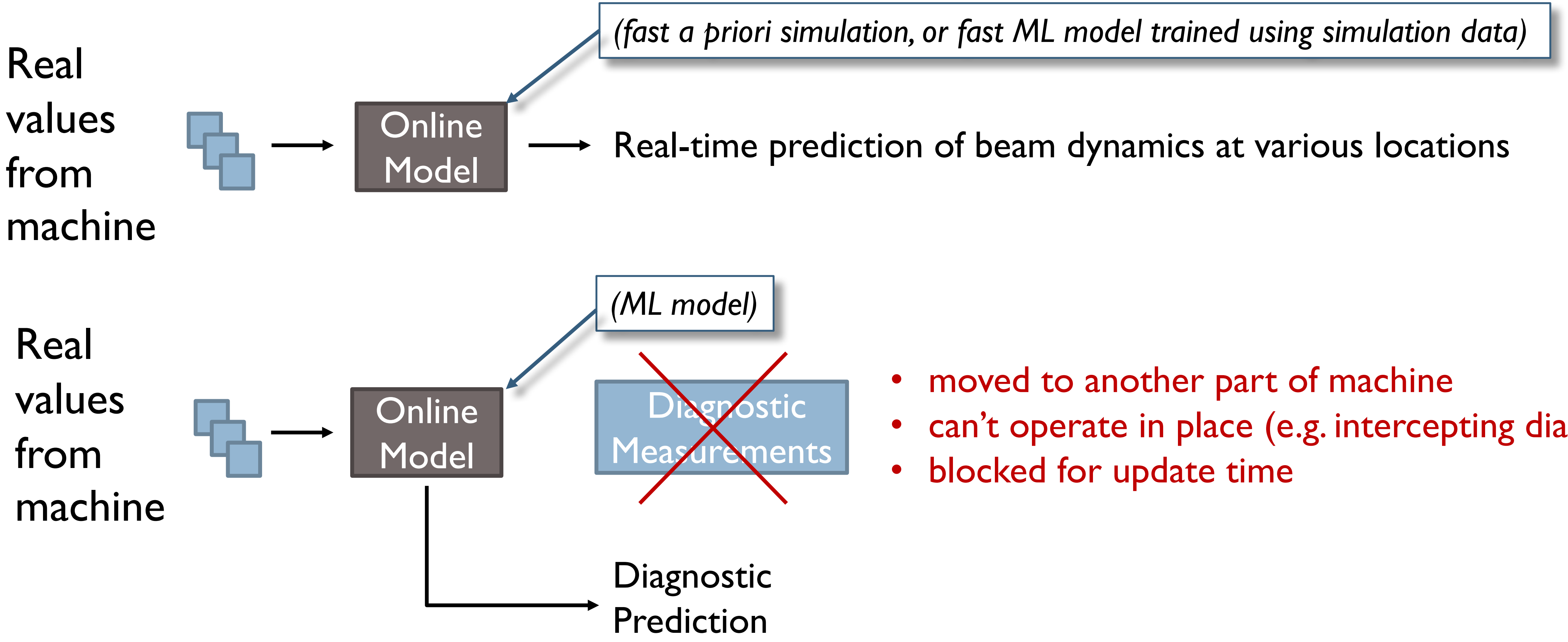
training updates

Diagnostic Measurements

Diagnostic Prediction

Virtual Diagnostics

Predict what diagnostics might look like when they are unavailable or don’t exist



Real values from machine → Online Model → Real-time prediction of beam dynamics at various locations

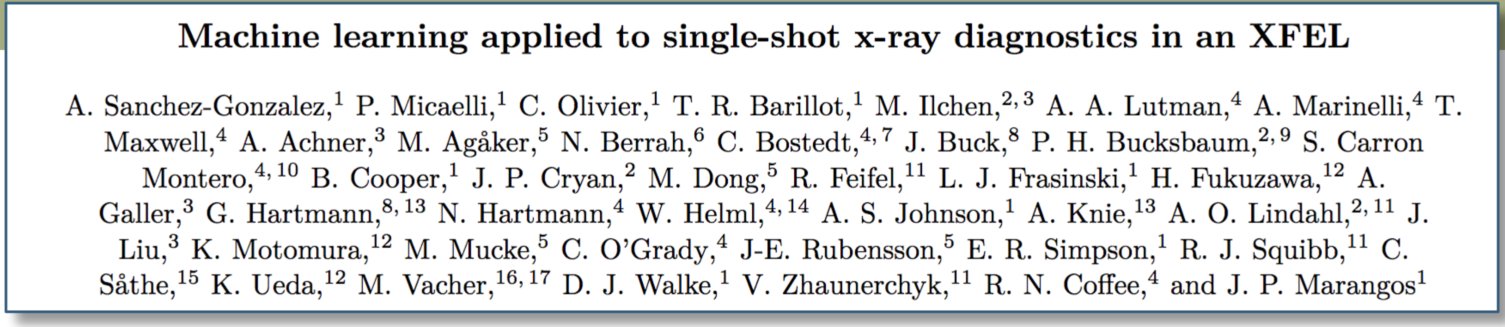
(fast a priori simulation, or fast ML model trained using simulation data)

Real values from machine → Online Model → Diagnostic Prediction

(ML model)

~~Diagnostic Measurements~~

- moved to another part of machine
- can’t operate in place (e.g. intercepting diagnostics)
- blocked for update time

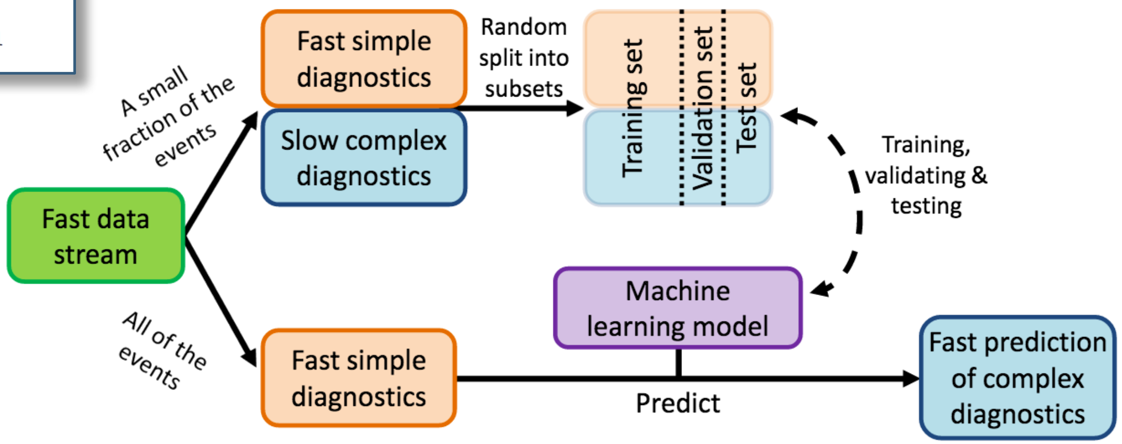


Machine learning applied to single-shot x-ray diagnostics in an XFEL

A. Sanchez-Gonzalez,[1] P. Micaelli,[1] C. Olivier,[1] T. R. Barillot,[1] M. Ilchen,[2,3] A. A. Lutman,[4] A. Marinelli,[4] T. Maxwell,[4] A. Achner,[3] M. Agåker,[5] N. Berrah,[6] C. Bostedt,[4,7] J. Buck,[8] P. H. Bucksbaum,[2,9] S. Carron Montero,[4,10] B. Cooper,[1] J. P. Cryan,[2] M. Dong,[5] R. Feifel,[11] L. J. Frasinski,[1] H. Fukuzawa,[12] A. Galler,[3] G. Hartmann,[8,13] N. Hartmann,[4] W. Helml,[4,14] A. S. Johnson,[1] A. Knie,[13] A. O. Lindahl,[2,11] J. Liu,[3] K. Motomura,[12] M. Mucke,[5] C. O'Grady,[4] J-E. Rubensson,[5] E. R. Simpson,[1] R. J. Squibb,[11] C. Såthe,[15] K. Ueda,[12] M. Vacher,[16,17] D. J. Walke,[1] V. Zhaunerchyk,[11] R. N. Coffee,[4] and J. P. Marangos[1]

A. Sanchez-Gonzalez, et al. https://arxiv.org/pdf/1610.03378.pdf

- Used archived data to learn correlation between fast and slow diagnostics
- Looked at a variety of ML methods and different diagnostics



Fast data stream

A small fraction of the events

Fast simple diagnostics

Slow complex diagnostics

Random split into subsets

Training set

Validation set

Test set

Training, validating & testing

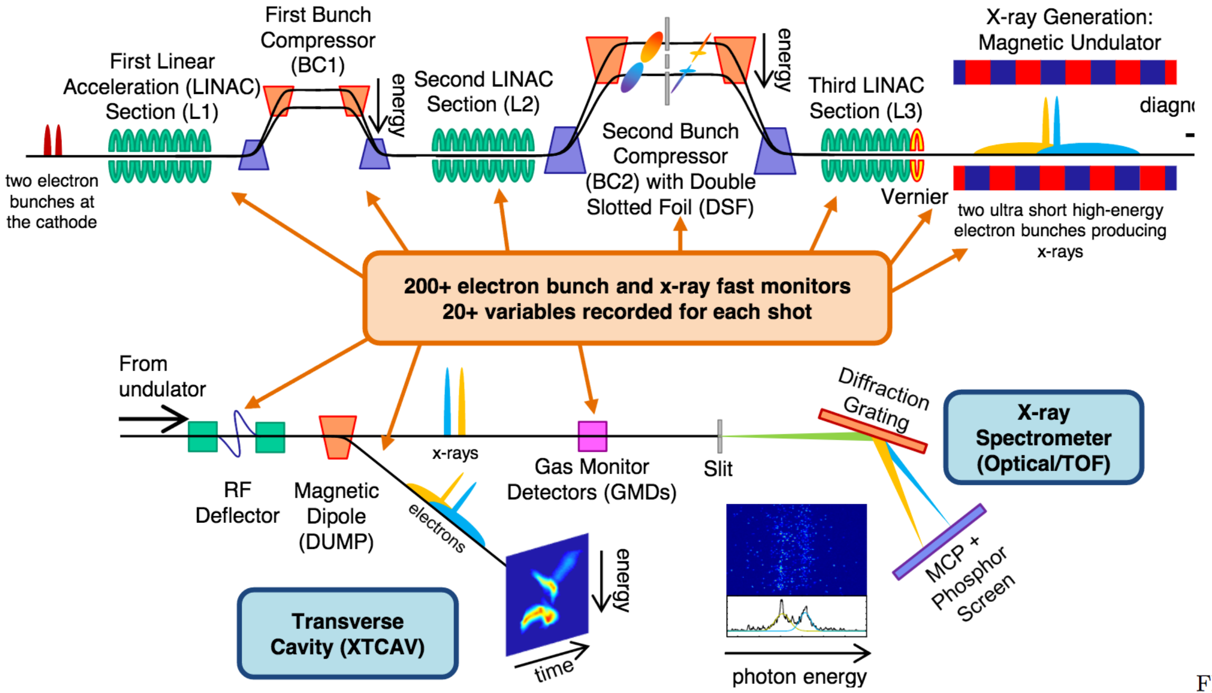
Machine learning model

All of the events

Fast simple diagnostics

Predict

Fast prediction of complex diagnostics


First Bunch Compressor (BC1)
First Linear Acceleration (LINAC) Section (L1)
energy
Second LINAC Section (L2)
energy
Second Bunch Compressor (BC2) with Double Slotted Foil (DSF)
Third LINAC Section (L3)
X-ray Generation: Magnetic Undulator
diagn
Vernier
two electron bunches at the cathode
two ultra short high-energy electron bunches producing x-rays
200+ electron bunch and x-ray fast monitors
20+ variables recorded for each shot
From undulator
RF Deflector
Magnetic Dipole (DUMP)
x-rays
electrons
Gas Monitor Detectors (GMDs)
Slit
Diffraction Grating
X-ray Spectrometer (Optical/TOF)
MCP + Phosphor Screen
Transverse Cavity (XTCAV)
energy
time
photon energy

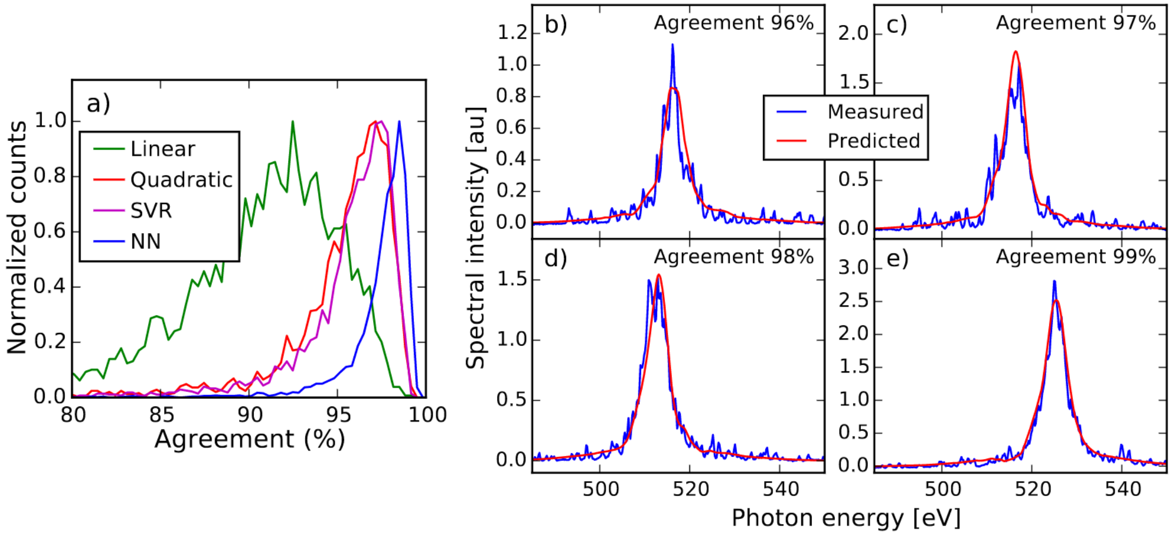


FIG. 4. Spectral shape prediction for a single pulse. (a) Histogram of agreements between the predicted and the measured spectra for the test set using the 4 different models. (b-e) Examples of the measured and the predicted spectra using a neural network to illustrate the accuracy for different agreement values.

Fault Prediction (Prognostics) + Anomaly Detection

Operations:

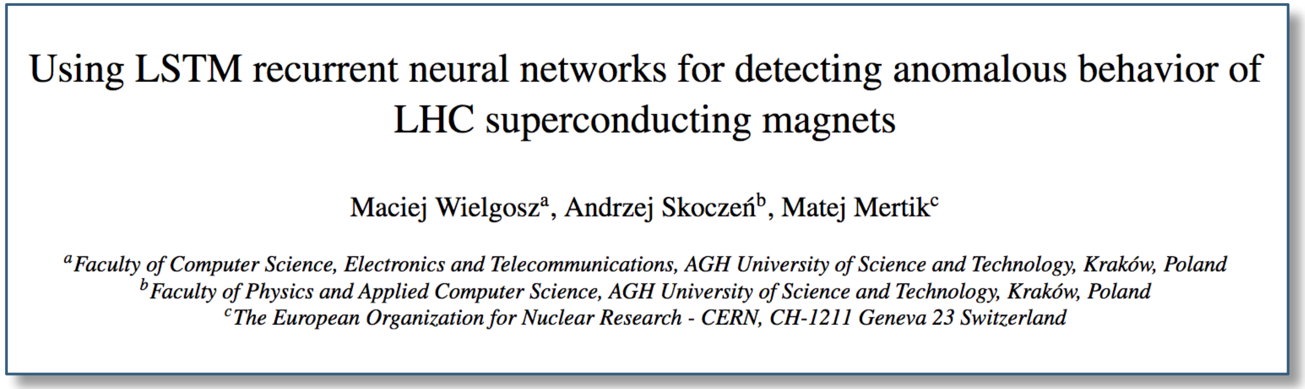
- Identify aberrant behavior that is correlated with faults, failures, or poor machine states
- Detect deviations from normal operating conditions that may otherwise go noticed

Machine Protection:

catastrophic failures and faults sometimes preceded by tell-tale signs

Replacement Cycles:

predict time-to-failure based on real-time and archived data



Using LSTM recurrent neural networks for detecting anomalous behavior of LHC superconducting magnets

Maciej Wielgosz[a], Andrzej Skoczeń[b], Matej Mertik[c]

[a] *Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology, Kraków, Poland*
[b] *Faculty of Physics and Applied Computer Science, AGH University of Science and Technology, Kraków, Poland*
[c] *The European Organization for Nuclear Research - CERN, CH-1211 Geneva 23 Switzerland*

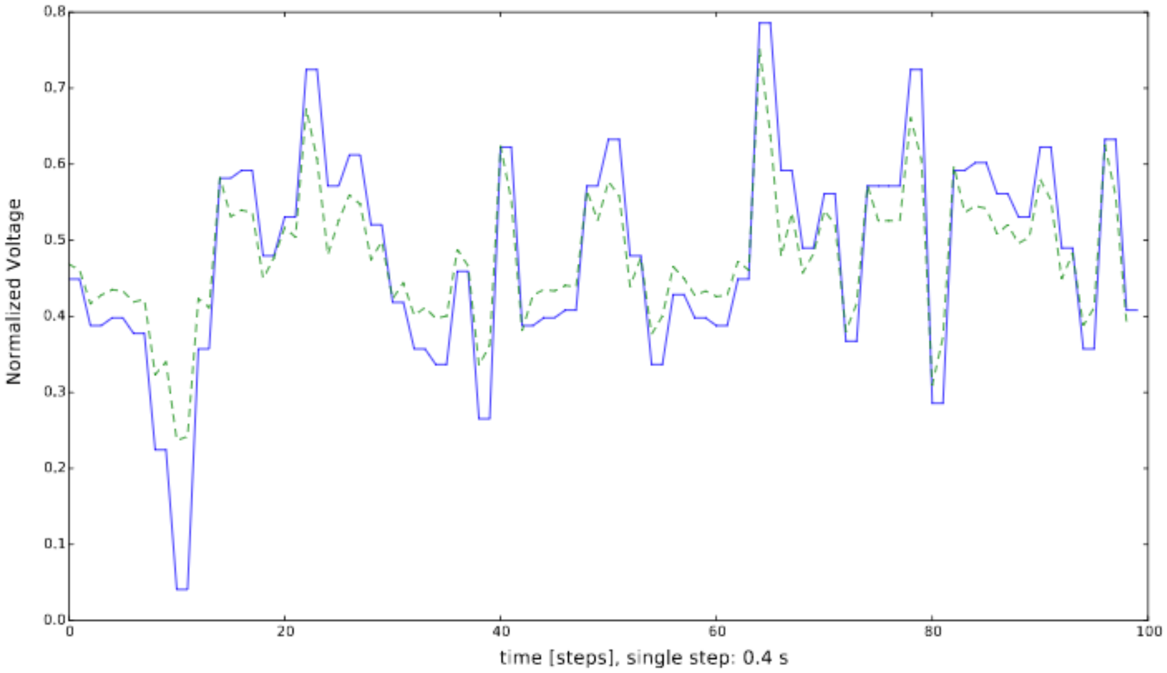
"Some of the most dangerous malfunctions of the magnets are quenches which occur when a part of the superconducting cable becomes normally-conducting."

Aim: use a recurrent NN to identify quench precursors in voltage time series.

→ ***Predict future behavior, then classify it***

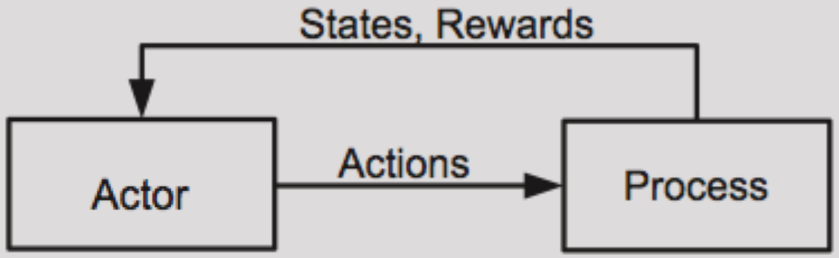
Initial study with small data set:

- 425 quenches for 600 A magnets
- Used archived data from 2008 to 2016
- 16-32 previous values → predict a few time steps ahead



Neural Network Policies and Reinforcement Learning

Actor-only Methods

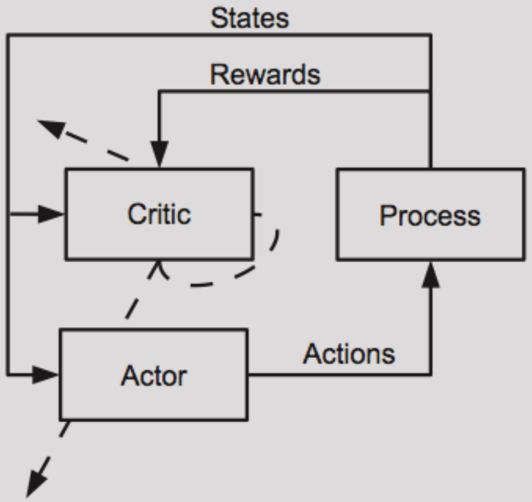


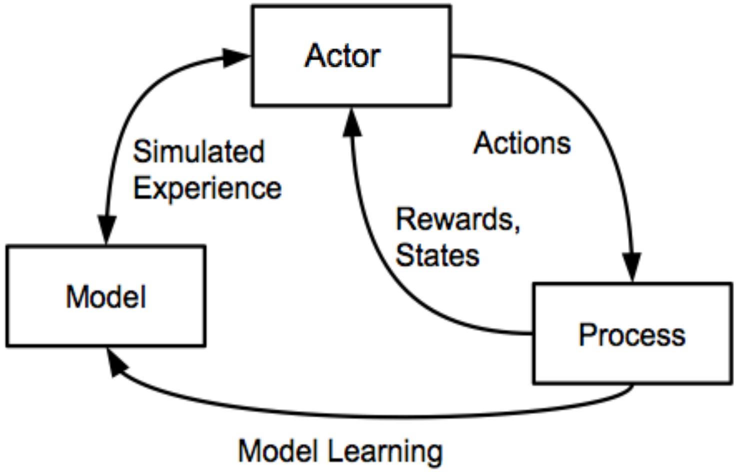
- Actor is a control policy
- Maps states to actions
- Reward provides training signal

Actor-Critic Methods

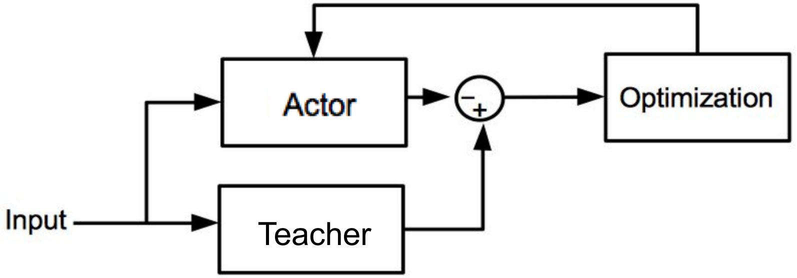
- Critic maps states or state/action pairs to an estimate of long-term reward
- Could be a NN, tabular, etc.
- Critic provides training signal to actor

Without actor: use an optimization algorithm with the critic





Can train on models first to get a good initial solution before deployment



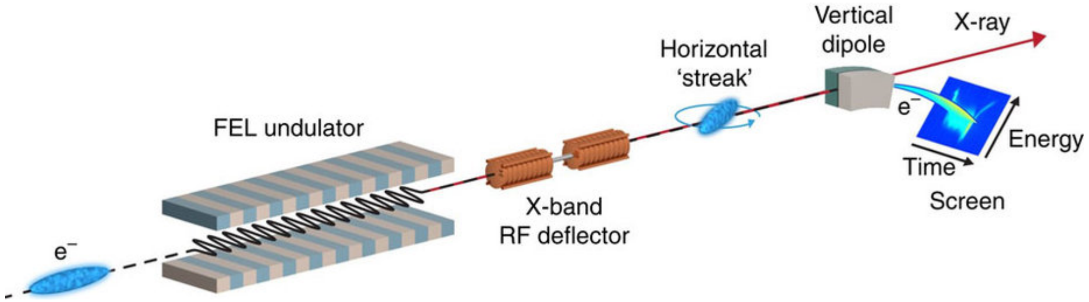
Can use supervised learning to first approximate the behavior of a different control policy

Auralee Edelen May 2017

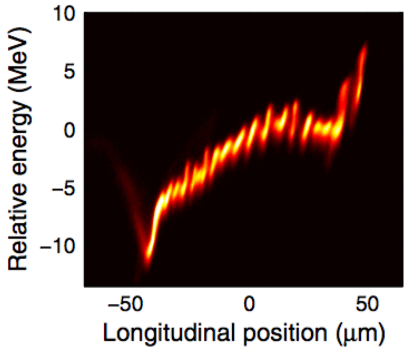
Computer Vision + Neural Network-based RL

- **Image diagnostics** → would be nice to use directly, and some yield relatively complicated information

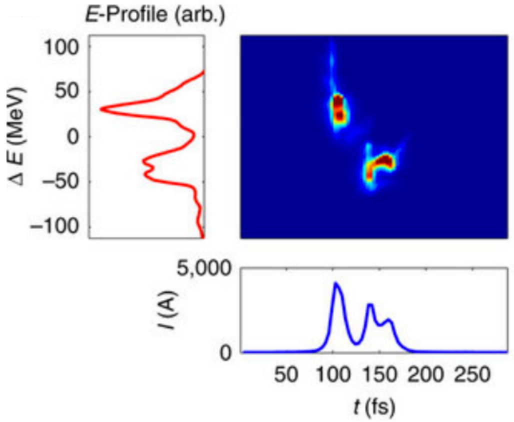
  e.g. XTCAV at SLAC



C. Behrens, et al., Nat. Commun. **5**, 3762 (2014)



D. Ratner, et al., PRSTAB18, 030704 (2015)



A. Marinelli, et al., Nat. Commun. **6**, 6369 (2015)

- **Convolutional Neural Networks (CNNs)** → very good at image processing

- **Reinforcement Learning (RL)** → can learn control policies from data

Why not try using image based diagnostics directly in learned control policies?

What's a relatively simple test case to start with?

Initial Study at FAST/IOTA

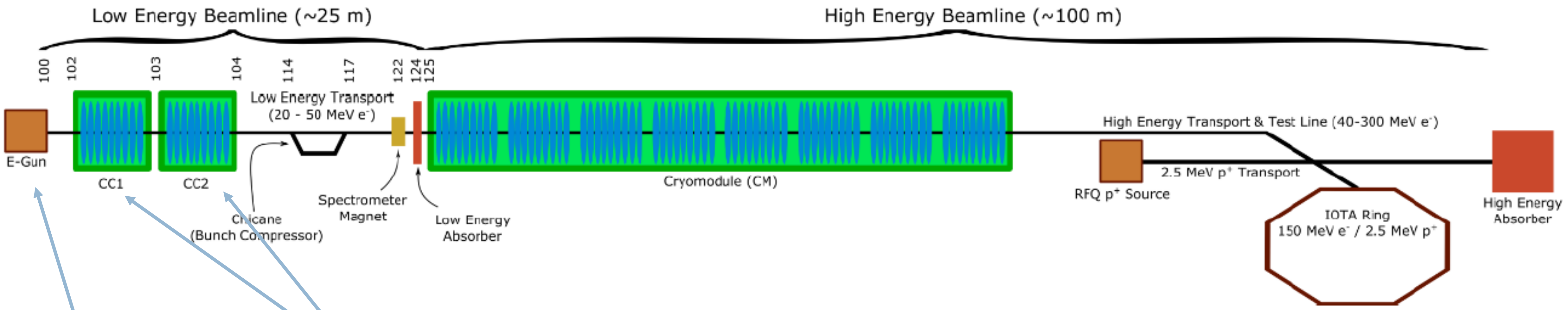


figure from various FAST reports

Photocathode RF Gun

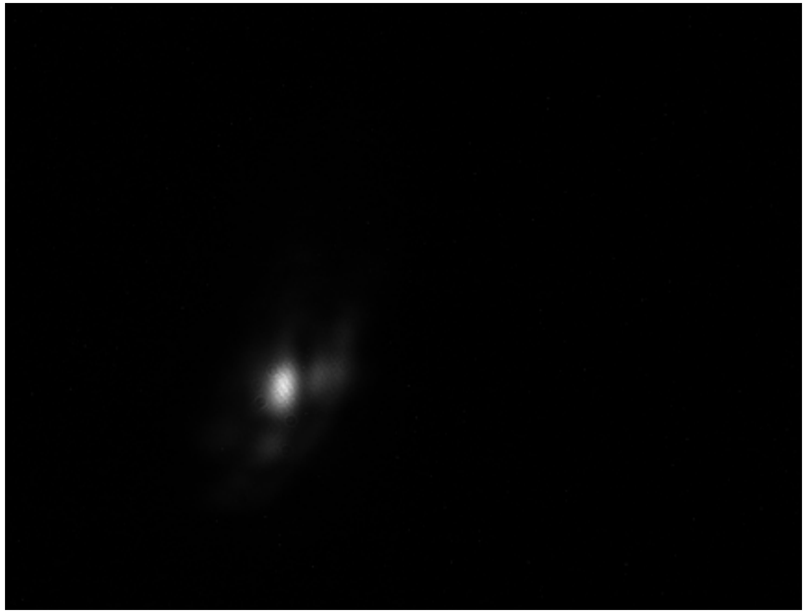
Superconducting Capture Cavities

Initial work with J. Edelen and D. Edstrom, FNAL

Initial Study: Choose Gun Parameters Based on Laser Spot

Motivation:

- Gun phase and solenoid strength tuned daily
- Asymmetries in initial laser distribution result in emittance asymmetries downstream
- Would be nice to obtain optimal gun phase and solenoid strength for a given initial laser distribution automatically (and perhaps prioritize x or y emittance to minimize)



Example virtual cathode image (10 Aug. 2016)

Other perks:

- PARMELA simulation based on survey data already in existence (J. Edelen)
- Try out creating a fast NN modeling tool from slower-executing simulations

Initial Study: Choose Gun Parameters Based on Laser Sp

Motivation:

- Gun phase and sole
- A
- V
  s
  d
  x

Why not just use online optimization?

Why not just fit a Gaussian to the laser spot to get the information instead of using images directly?

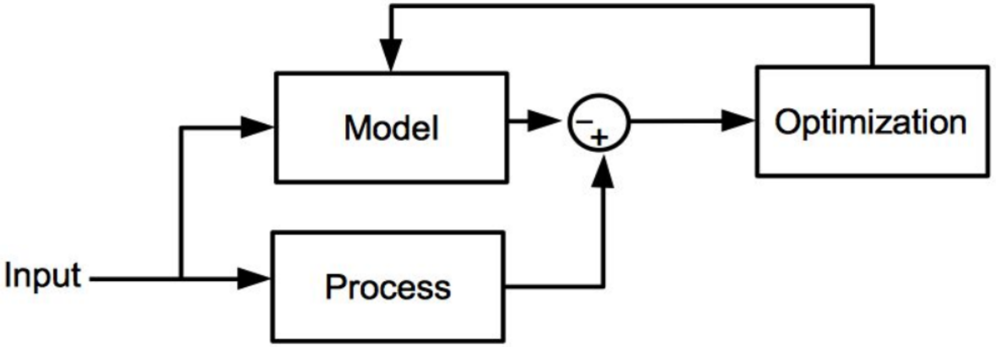
The point of this study: explore this approach on a simple system (it's a stepping stone)

Othe

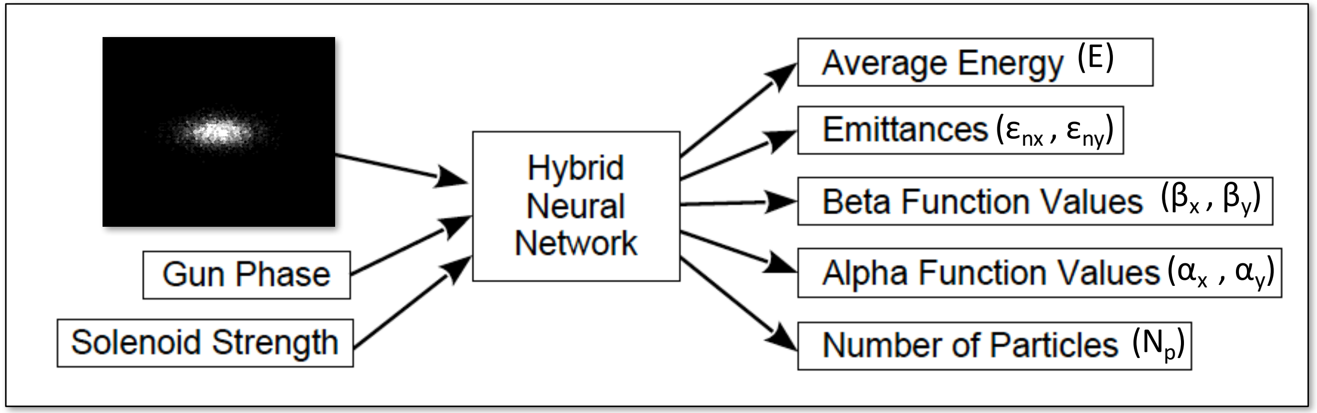
- PAR ta already in existence (J. Edelen)
- Try c NN modeling tool from slower-executing simulations

Initial Study: Steps

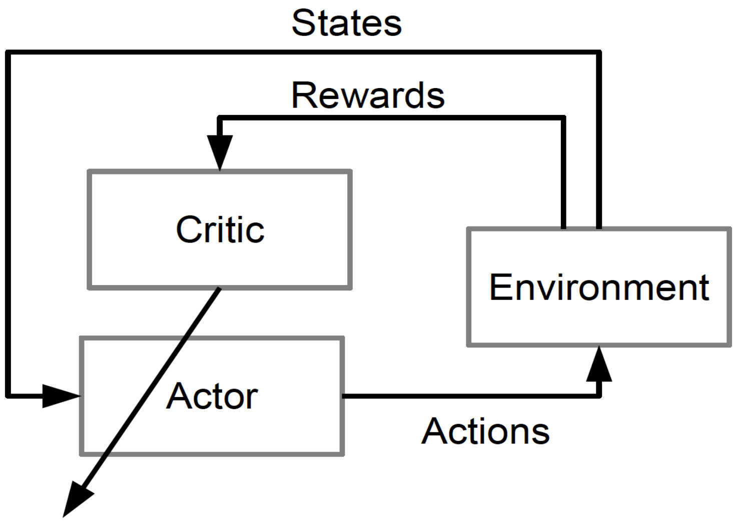
- Gather simulation data from PARMELA scans
- Create a NN model
  - Be certain that the necessary information can be extracted from the image, gun phase, and solenoid strength
- Train a RL controller using that model
- Extension beyond simulation (tentative):
  - Incorporate measured data into model and update controller
  - Carefully test on machine



Model Learning


Gun Phase
Solenoid Strength
Hybrid Neural Network
Average Energy (E)
Emittances (ε_{nx}, ε_{ny})
Beta Function Values (β_x, β_y)
Alpha Function Values (α_x, α_y)
Number of Particles (N_p)

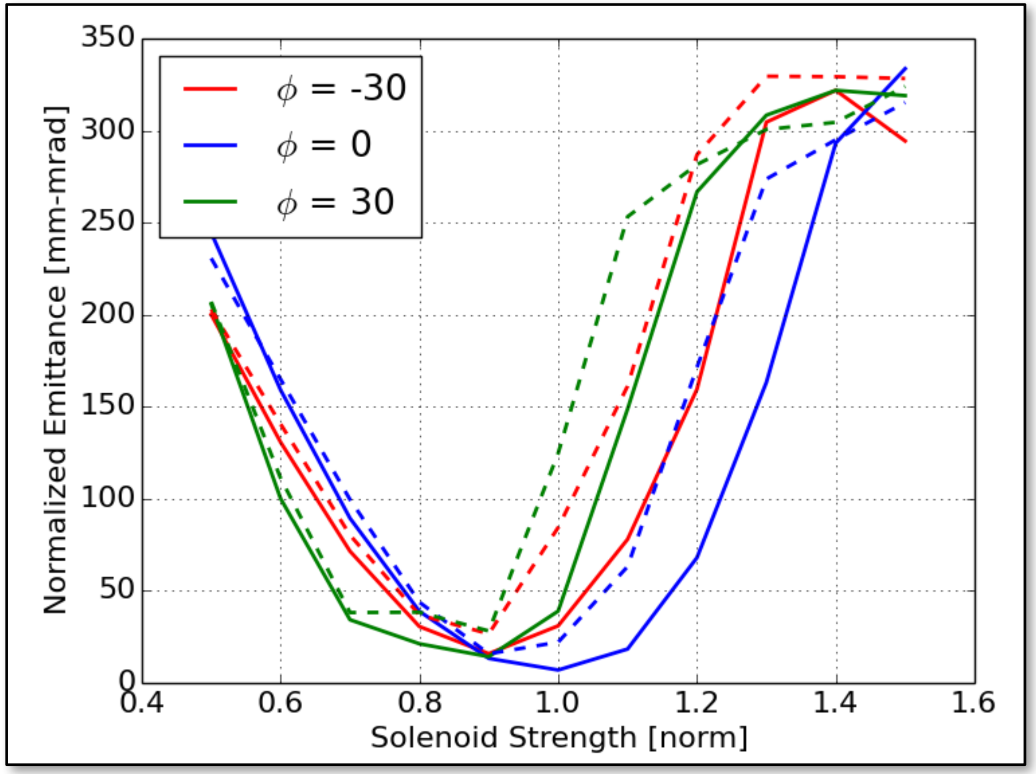
model inputs and outputs



Policy Learning

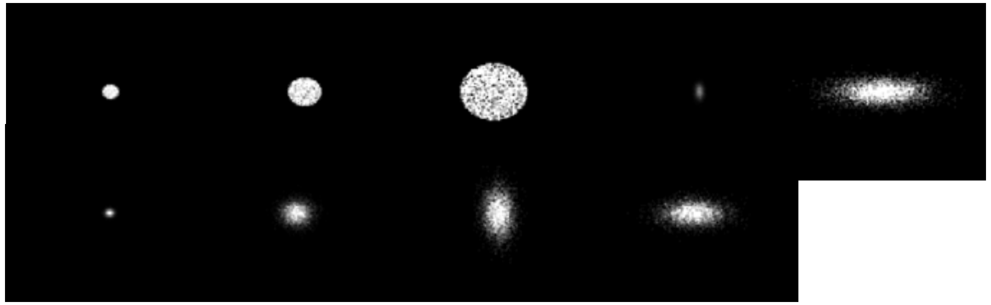
CNN Model: Simulation Data

- PARMELA simulations from the gun up to the exit of CC2
  - 2-D space charge routine
  - Scanned gun phase, solenoid strength, initial beam distribution
- Two sets of data:
  - Fine scans (steps of 5° phase, 5% sol. str.) for sims just past the gun
  - Coarse scans (steps of 10° phase, 10% sol. str.) for sims up through CC2
- Simulated "virtual cathode images"
  - Going from VCI → initial beam distribution ok from prior work
  - Initial beam distribution → simulated VCI probably ok
  - Obviously very "well-behaved" examples



Simulation predictions after CC2. Dashed lines are x-emittance, solid lines are y-emittance.

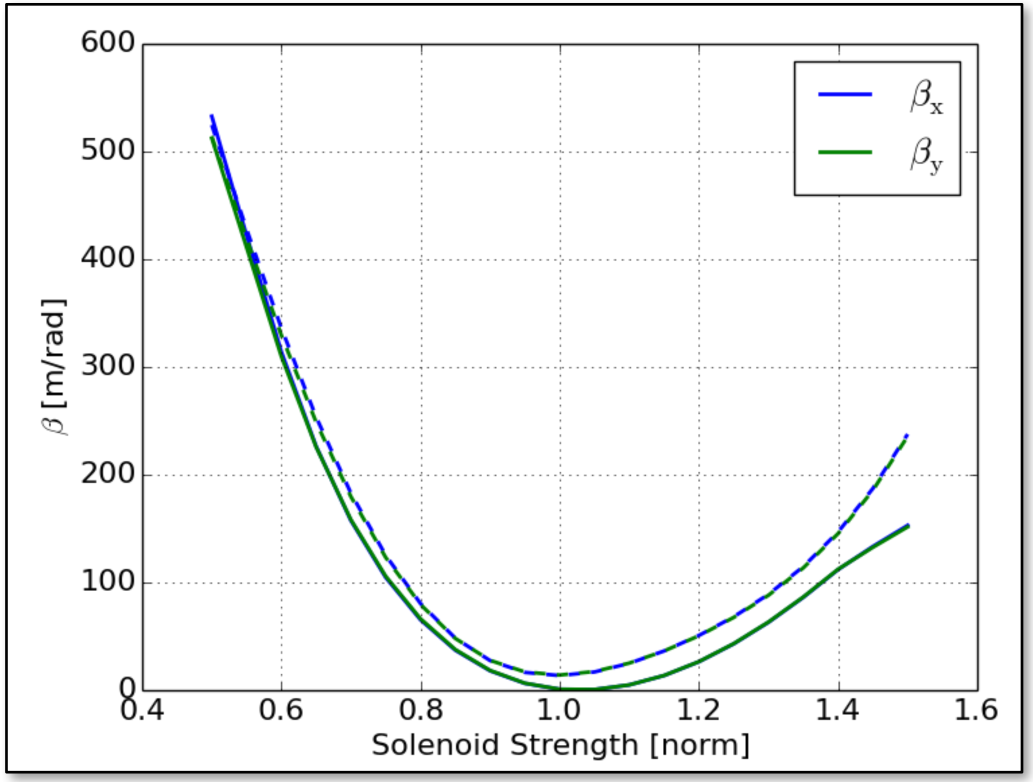
Caveat: doesn't take into account coupling…later changed NN setup to predict sigma matrix, and also used a 3D space charge routine.



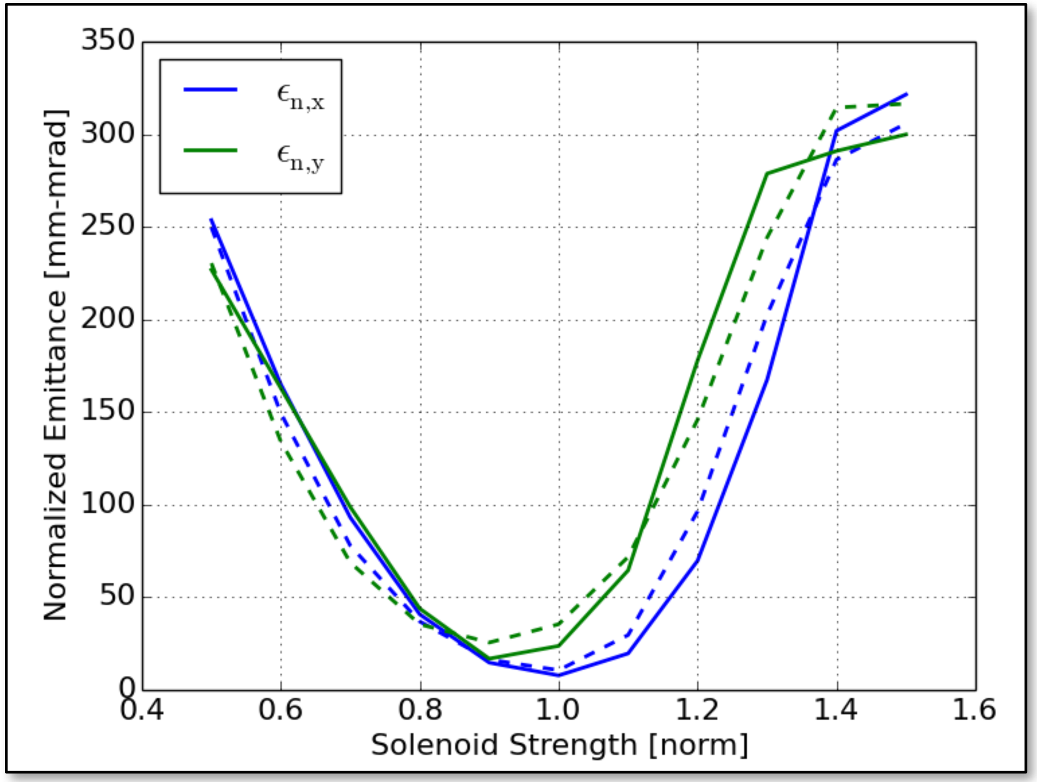
For normalized sol strength, 1 is the setting that produces a peak axial field of 1.8 kG

CNN Model: Two Representative Plots

Dashed lines are NN predictions and solid lines are simulation results



Top-hat initial beam, 0° RF phase, after gun



Asymmetric Gaussian initial beam, 0° RF phase, after CC2

For the gun data, all MAEs are between 0.4% and 1.8% of the parameter ranges.
For the CC2 data, all MAEs are between 0.9% and 3.1% of the parameter ranges.

→ Not bad for such a small training set

Fast Switching Between Trajectories

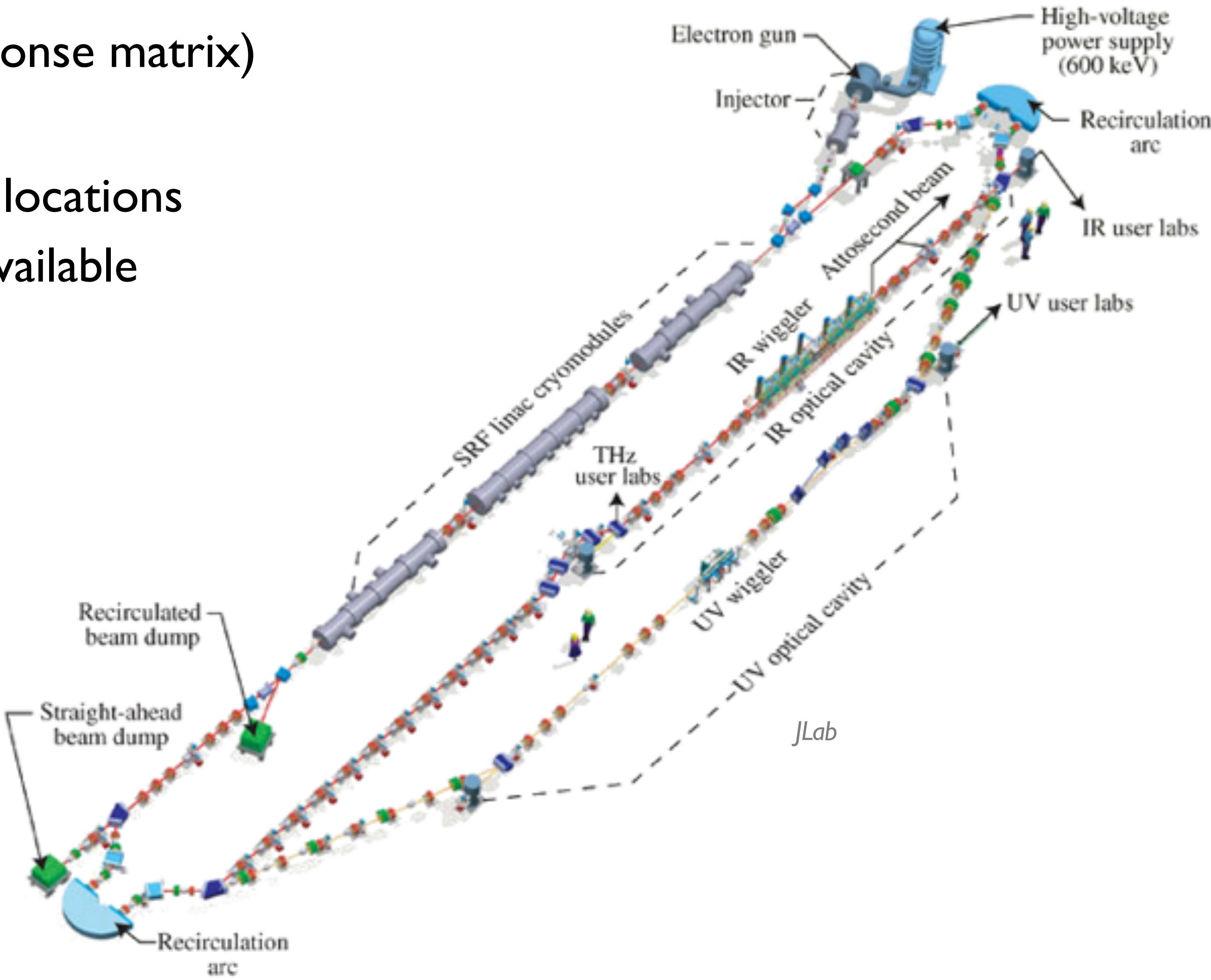
Work with C. Tennant and D. Douglas, JLab

- 76 BPMs, 57 dipoles, 53 quadrupoles
- Traditional approach has never worked (linear response matrix)
- Rely on one expert for steering tune-up
- Want to specify small offsets in trajectory at some locations
- Didn't initially have an up-to-date machine model available

Learn responses (NN model) from tune-up data and dedicated study time:
dipole + quadrupole settings → predict BPMs

Train controller (NN policy) offline using NN model:
desired trajectory → dipole settings
(and penalize losses + large magnet settings)

Test on machine: check to make sure model prediction still accurate and try static controller (non-adaptive)



JLab

Fast Switching Between Trajectories

- 76 BPMs, 57 dipoles, 53 quadrupoles
- Traditional approach has never worked (linear response matrix)
- Rely on one expert for steering tune-up
- Want to specify small offsets in trajectory at some locations
- Didn't initially have an up-to-date machine model available

Learn responses (NN model) from tune-up data and dedicated study time:
dipole + quadrupole settings → predict BPMs

Train controller (NN policy) offline using NN model:
desired trajectory → dipole settings
(and penalize losses + large magnet settings)

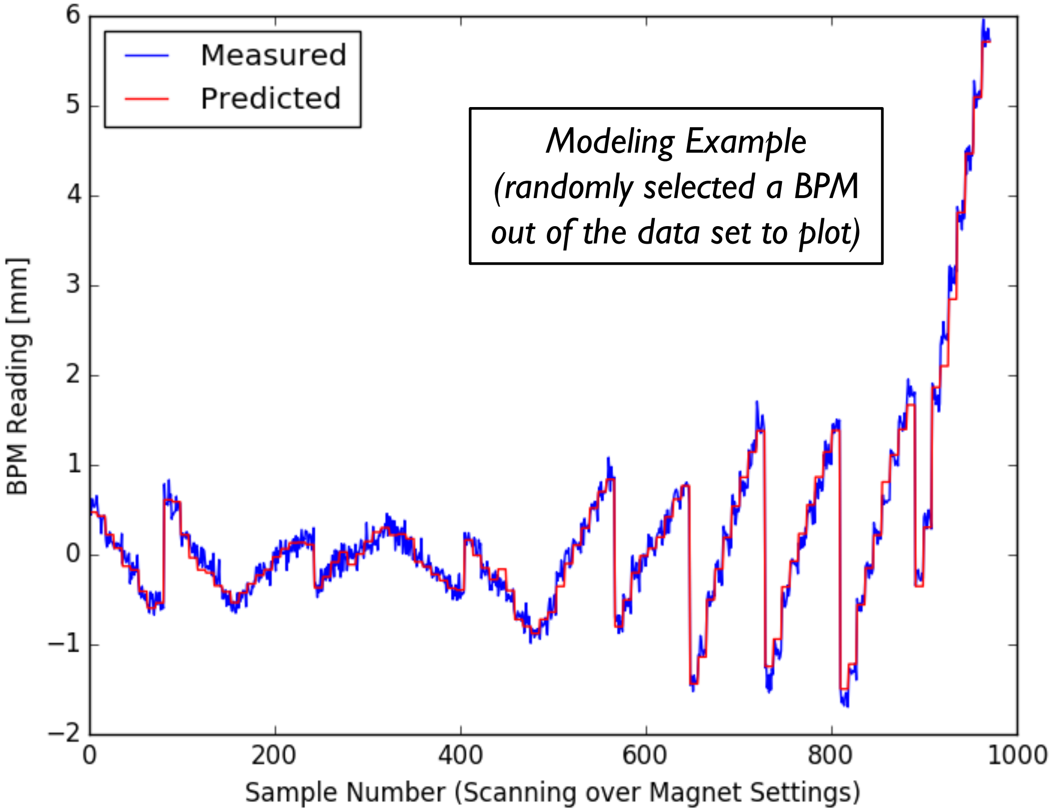
Test on machine: check to make sure model prediction still accurate and try static controller (non-adaptive)

(Very) Preliminary Results:

Model Errors for BPMs:

Training Set:	0.07 mm MAE	0.09 mm STD
Validation Set:	0.08 mm MAE	0.07 mm STD
Test Set:	0.08 mm MAE	0.03 mm STD

Controller: random initial states → on average within 0.2 mm of center immediately



Modeling Example (randomly selected a BPM out of the data set to plot)

Fast Switching Between Trajectories

- 76 BPMs, 57 dipoles, 53 quadrupoles
- Traditional approach has never worked (linear response matrix)
- Rely on one expert for steering tune-up
- Want
- Didn

(Very) Preliminary Results:

Model Errors for BPMs:

Training Set:	0.07 mm MAE	0.09 mm STD
Validation Set:	0.08 mm MAE	0.07 mm STD
Test Set:	0.08 mm MAE	0.03 mm STD

Controller: random initial states → on average

Similar Kind of Task: switching between FEL frequencies (in progress)

→ simulation study with CSU FEL (3 – 6 MeV e- beam → space charge)

→ use optimization iteration output from simulation to train NN model

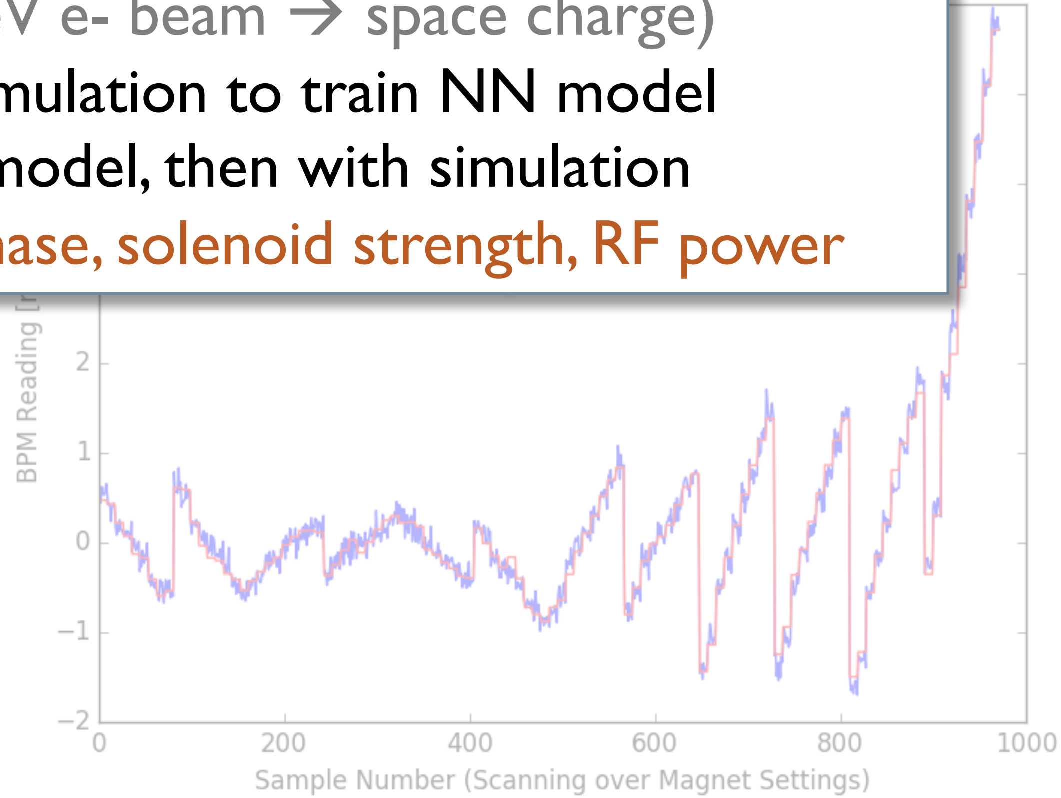
→ train controller via interaction with NN model, then with simulation

→ given target wavelength: set quads, gun phase, solenoid strength, RF power

Learn
dedic
dipole

Train controller (NN policy) offline using NN model:
desired trajectory → dipole settings
(and penalize losses + large magnet settings)

Test on machine: check to make sure model prediction still accurate and try static controller (non-adaptive)



Final Notes: Some Practical Challenges

*large enough parameter range and set of examples to generalize well and complete the task

*you can trust it

Need a sufficient amount of reliable* data*

(but not as much as is sometimes claimed in DL)

Training on Measured Data

Undocumented manual changes
(e.g. rotating a BPM)

Relevant-but-unlogged parameters

Availability of diagnostics

Observed parameter range in archived data

Time on machine for characterization studies
(schedule + expense)

Ideal case:
- comprehensive, high-resolution data archive
- excellent log of manual changes

Training on Simulation Data

How representative of the real machine behavior?

Input/output parameters need to translate directly to what's on the machine (quantitatively)

High-fidelity (e.g. PIC)
→ time-consuming to run

Retention + availability of prior results:
(optimize and throw the iterations away!)

Deployment

Initial training is on HPC systems → deployment is typically not*

- Execution on front-end: necessary speed + memory?
- Subsequent training: on front-end or transfer to HPC?

Software compatibility for older systems:
interface with machine + make use of modern ML software libraries

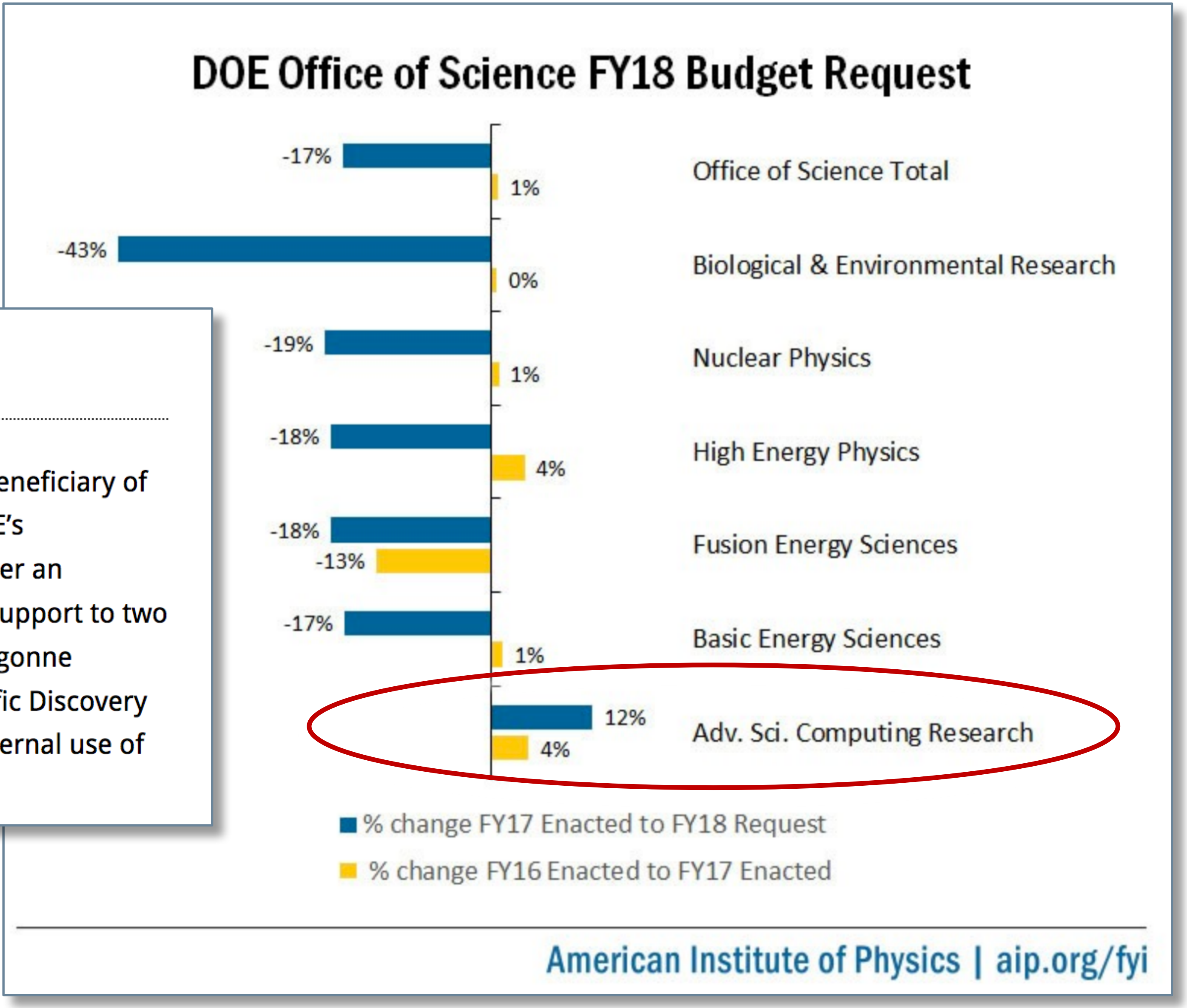
I/O for large amounts of data

* for now…

Final Notes: Funding Climate

Advanced Scientific Computing Research

Advanced Scientific Computing Research stands out as the primary beneficiary of the Office of Science budget. Its budget includes $347 million for DOE's contribution to the interagency Exascale Computing Initiative to deliver an exascale-capable computing system by 2021. It would also increase support to two of DOE's three Leadership Computing Facilities – at Oak Ridge and Argonne National Laboratories. An additional increase would grow the Scientific Discovery through Advanced Computing (SciDAC) program, which facilitates external use of DOE supercomputers.



American Institute of Physics | aip.org/fyi

Final Notes

- Neural networks are very flexible tools → far more powerful in recent years
- Mostly preliminary results so far, but making progress (+ more infrastructure in place)
- Lots of opportunities to use neural networks (and ML more broadly) to improve accelerator performance on both existing and future machines

Fermilab has a strong presence in machine learning (especially for neural networks/HEP)

Lots of potential for fruitful collaborations on the accelerator side

→ LBNL, SLAC, LANL, CERN all interested in applying ML to accelerator modeling/controls

Some possible experiments at Fermilab:

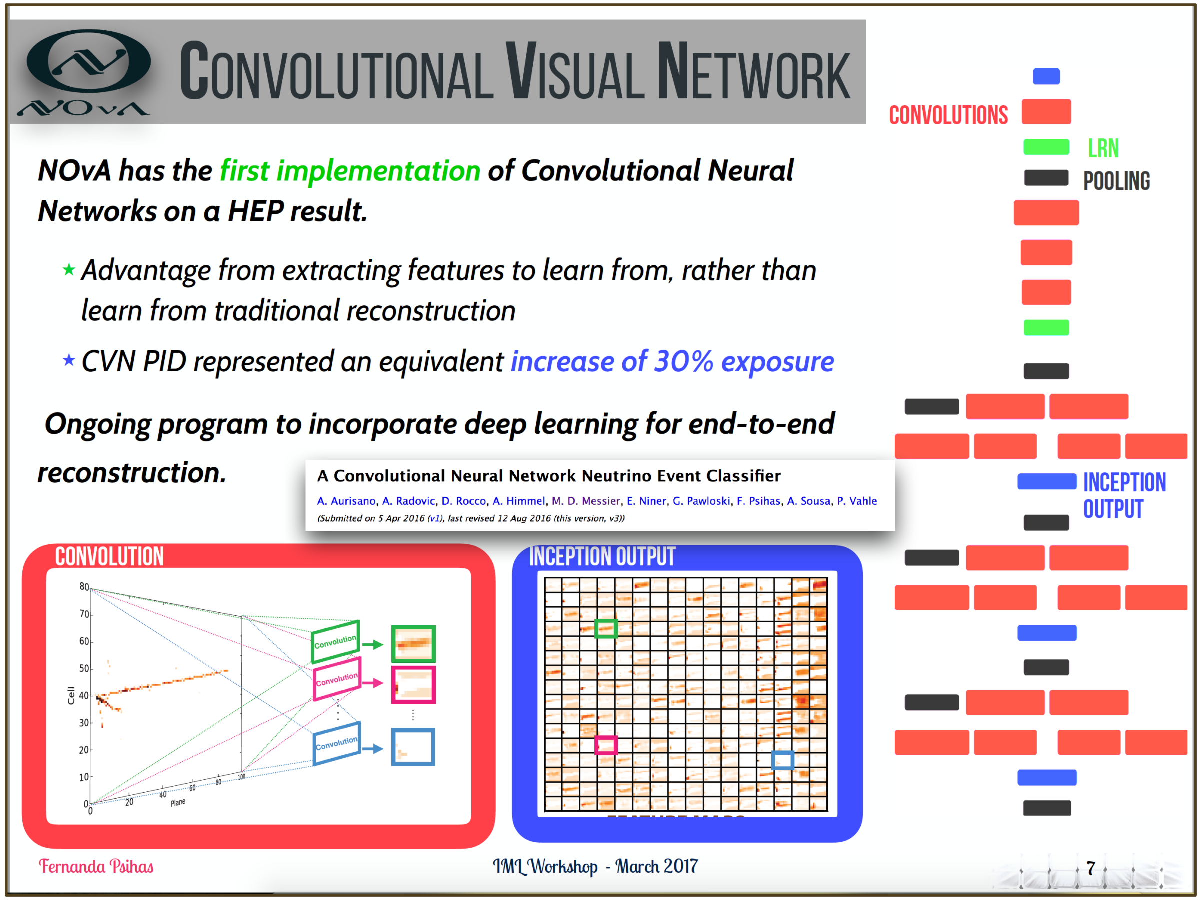
- Ion sources (MPC/RL)
- Cryogenic system control (MPC/RL)
- Fermi Test Beam Facility (fast switching)
- Muon Campus (virtual diagnostics, online modeling)
- Phase space manipulations at FAST (fast switching)

Thanks for your attention!

Final Notes: Fermilab has a strong presence in machine learning (especially for DL/HEP)

- See Fernanda Psihas New Perspectives 2017 talk
- Ramping up HPC resources
- Slack channel: https://hepmachinelearning.slack.com
- Journal Club meetings
- Monthly Intro meetings
- Website: http://machinelearning.fnal.gov/

CNN Applications for HEP
June 9th
10:30 AM, One West



CONVOLUTIONAL VISUAL NETWORK

NOvA has the first implementation of Convolutional Neural Networks on a HEP result.

- ★ *Advantage from extracting features to learn from, rather than learn from traditional reconstruction*
- ★ *CVN PID represented an equivalent increase of 30% exposure*

Ongoing program to incorporate deep learning for end-to-end reconstruction.

A Convolutional Neural Network Neutrino Event Classifier
A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, P. Vahle
(Submitted on 5 Apr 2016 (v1), last revised 12 Aug 2016 (this version, v3))

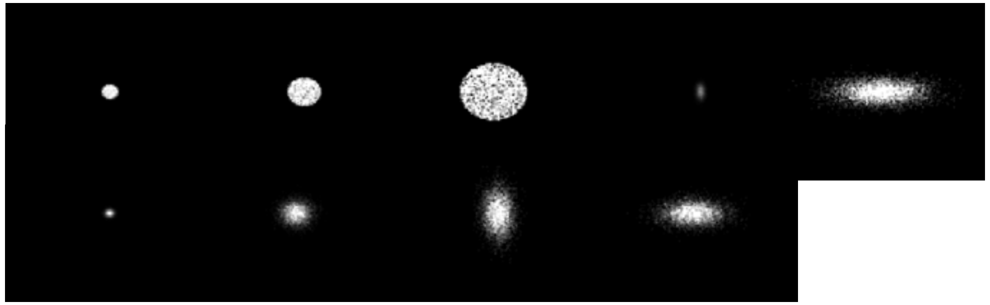
Fernanda Psihas IML Workshop - March 2017 7

CNN Model: Simulation Data

- PARMELA simulations from the gun up to the exit of CC2
  - 2-D space charge routine
  - Scanned gun phase, solenoid strength, initial beam distribution

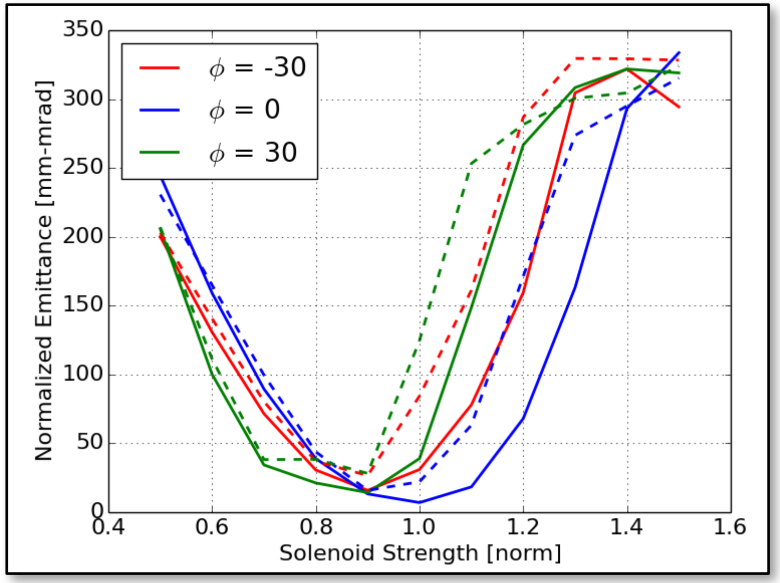
- Two sets of data:
  - Fine scans (steps of 5° phase, 5% sol. str.) for sims just past the gun
  - Coarse scans (steps of 10° phase, 10% sol. str.) for sims up through CC2

- Simulated "virtual cathode images"
  - Going from VCI → initial beam distribution ok from prior work
  - Initial beam distribution → simulated VCI probably ok
  - Obviously very "well-behaved" examples



Parameter Ranges used for Model Training

Parameter	Gun Data		CC2 Data	
	Max Value	Min Value	Max Value	Min Value
N_p	5001	1015	5001	1004
ε_{nx} [m-rad]	2.50E-04	1.60E-06	4.00E-04	9.10E-07
ε_{ny} [m-rad]	2.40E-04	1.60E-06	4.00E-04	8.50E-07
α_x [rad]	14.1	-775.1	0.8	-149.8
α_y [rad]	14.5	-797	0.7	-154.5
β_x [m/rad]	950.4	7.90E-02	820.2	0.7
β_y [m/rad]	896.8	8.40E-02	845.7	0.81
E [MeV]	4.6	3.2	47.2	42.8



Simulation predictions after CC2. Dashed lines are x-emittance, solid lines are y-emittance.

Caveat: doesn't take into account coupling…later changed NN setup to predict sigma matrix, and also used a 3D space charge routine.

For normalized sol strength, 1 is the setting that produces a peak axial field of 1.8 kG

CNN Model: Performance

Parameter	Train. MAE	Train. STD	Val. MAE	Val. STD
N_p	69.5	79.8	70.7	75.7
ε_{nx}	2.30E-06	3.50E-06	2.40E-06	3.20E-06
ε_{ny}	2.30E-06	3.40E-06	2.40E-06	3.20E-06
α_x	9	14.9	10.9	16
α_y	8.8	15.3	10.8	16.1
β_x	12.1	17.6	14.8	18.9
β_y	11.7	16.7	14.3	17.9
E	4.90E-03	4.90E-03	5.50E-03	6.00E-03

Performance for the predictions after the gun

Parameter	Train. MAE	Train. STD	Val. MAE	Val. STD
N_p	103.7	141.2	123.3	176.8
ε_{nx}	1.00E-05	1.20E-05	1.20E-05	1.60E-05
ε_{ny}	1.00E-05	1.30E-05	1.20E-05	1.50E-05
α_x	3.4	6.6	3.1	5.9
α_y	3.4	6.6	3.1	5.9
β_x	16.3	33.5	14.7	27.8
β_y	16.4	33.6	14.8	27.5
E	4.00E-02	3.90E-02	4.60E-02	6.20E-02

Performance for the predictions after CC2

For the gun data, all MAEs are between 0.4% and 1.8% of the parameter ranges.
For the CC2 data, all MAEs are between 0.9% and 3.1% of the parameter ranges.

→ *Not bad for such a small training set*

Present Status and Next Steps

- **Improving the quality of the setup:**
  - Predicting the full sigma matrix
  - More realistic initial distributions
  - Using 3D space charge routine
  - Using locally-connected layers
  - Switching to ASTRA

    (greater execution speed → more training data)

- **Expanding scope to phase space manipulations:**
  - Specify a target sigma matrix
  - Include quads after CC2, capture cavity phases, etc.
  - Collaborating with NIU:
    - RTFB transform is a possible application
    - Alex Halavanau running simulation scans with NIU's newer model → more training data

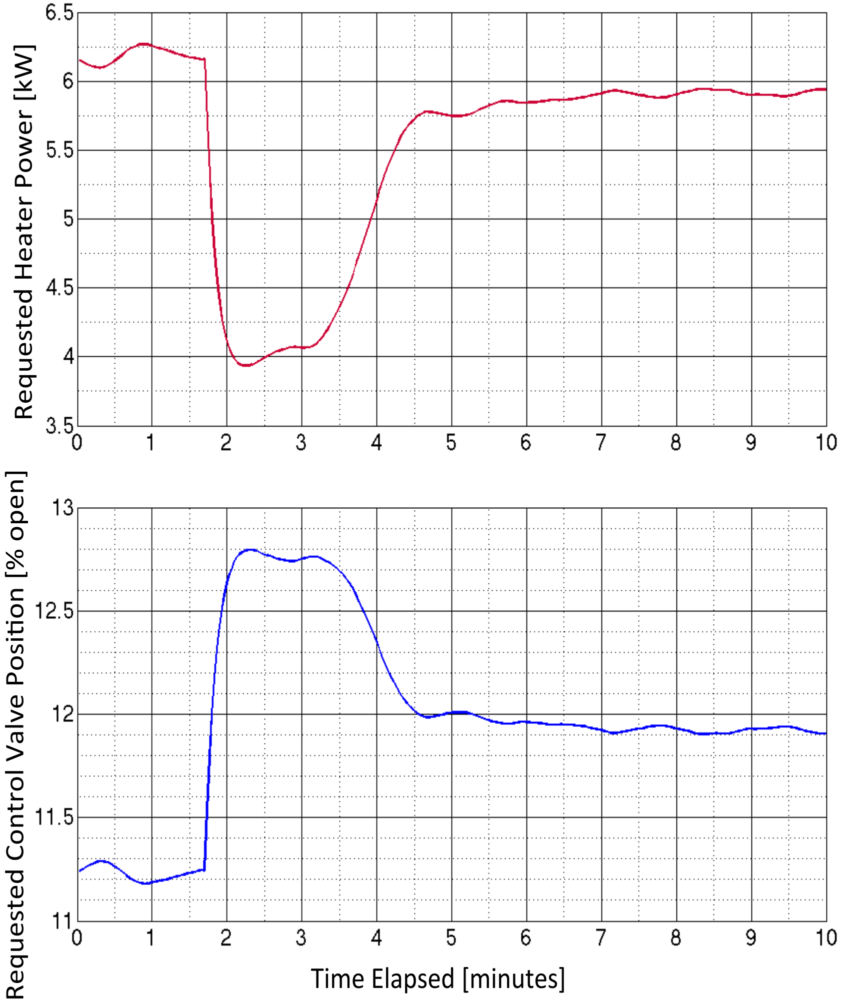
- **Next steps (in tandem):**
  - Finish simulation study with present setup
  - Extend to phase space manipulation simulation study
  - Solidify plans for incorporating measured data and testing controller
    - Need to align available inputs/controllable variables (e.g. sigma matrix vs. info from emittance monitors, rotation of quads, etc.)
    - Also depends on run schedule, status of new emittance monitors, solid time with consistent setup, etc.

Also, if you have some other possible application and have or can easily obtain training data: don't hesitate to get in touch!

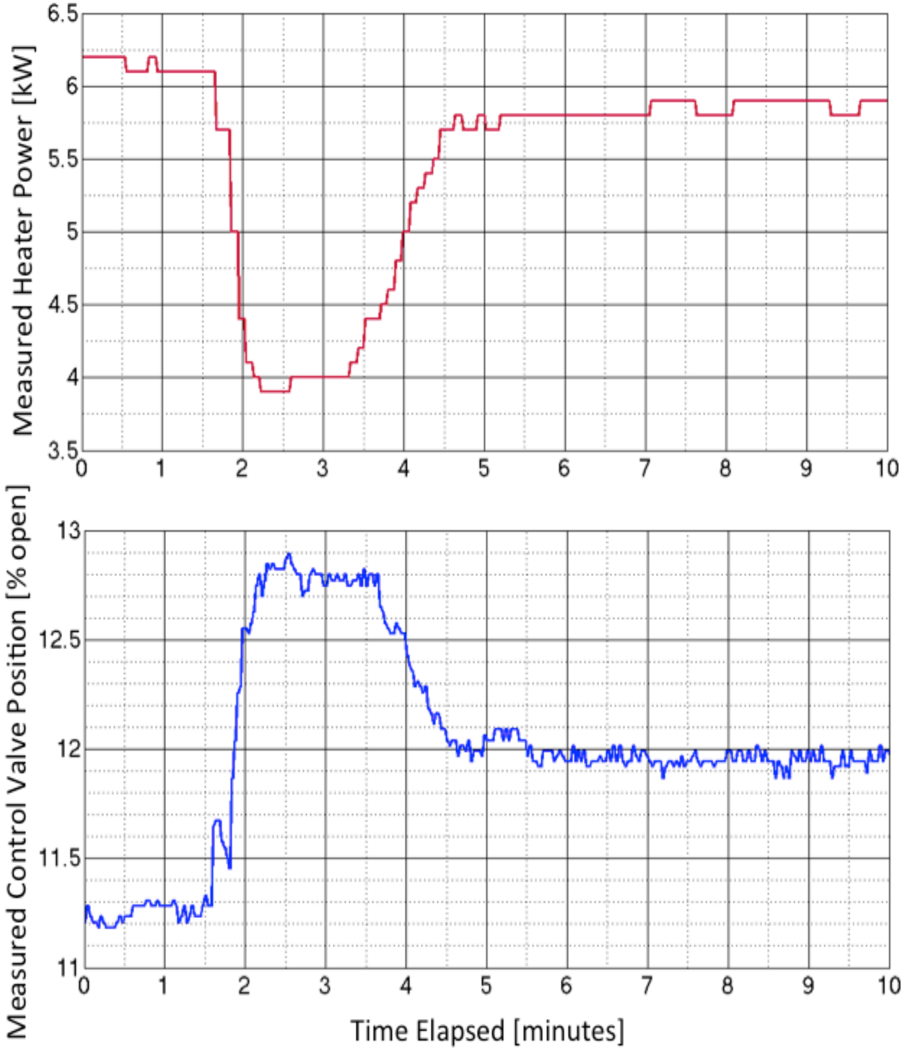
MPC Benchmark Controller: Actions

Could optimize for lower heater power

Requested by Controller



Actual Read-backs



Backpropagation

Vectorized notation: $a_j = f\left(\sum_k w_{jk} x_k + b_j\ \right) \ \rightarrow\ f(wx+b)$

Layer-by layer: $a^l = f\left(w^l a^{l-1} + \ b^l\right) = f(z^l)$

a_j j^{th} node activation

b_j j^{th} node bias

w_{jk} j^{th} node in layer l, k^{th} node in $l-1$

f applied element-wise

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}$$

$$\delta_j^{N_l} = \frac{\partial C}{\partial a_j^{N_l}} f'(z_j^{N_l}) \qquad \rightarrow \quad \delta^{N_l} = \nabla_a C \odot f'(z^{N_l})$$

$$\delta_j^l = \sum_k \frac{\partial C}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} \qquad = \sum_k \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

$$= \sum_k w_{kj}^{l+1} \delta_k^{l+1} f'(z_j^l)$$

$$z_k^{l+1} = \sum_j w_{kj}^{l+1} a_j^l + b_k^{l+1}$$

$$= \sum_j w_{kj}^{l+1} f(z_j^l) + b_k^{l+1}$$

$$\frac{\partial z_k^{l+1}}{\partial z_j^l} = w_{kj}^{l+1} f'(z_j^l)$$

For each training instance:

1. **Forward Pass:**

   For l = 1, 2, 3 … N_l

   $$z^l = w^l a^{l-1} + b$$

   $$a^l = f(z^l)$$

2. **'Error':**

   $$\delta^{N_l} = \nabla_a C \odot f'(z^{N_l})$$

3. **Backward Pass:**

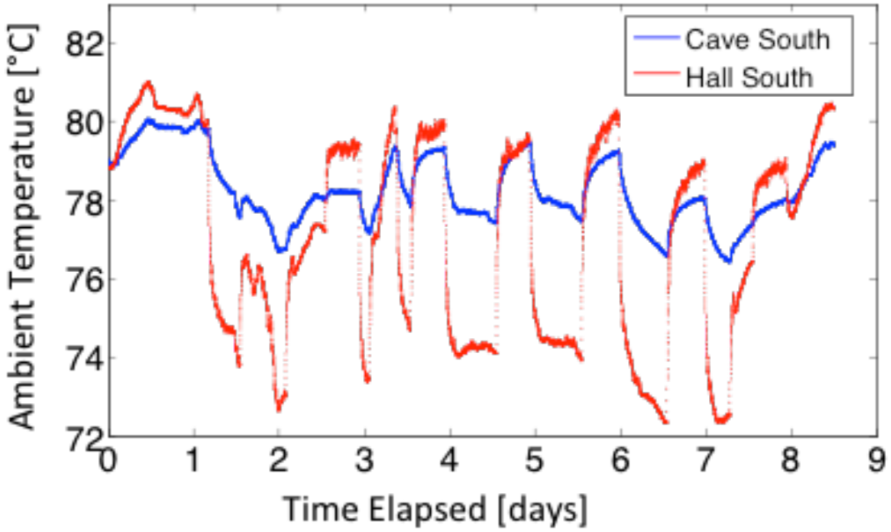
   For l = N_l − 1, N_l − 2, … 1

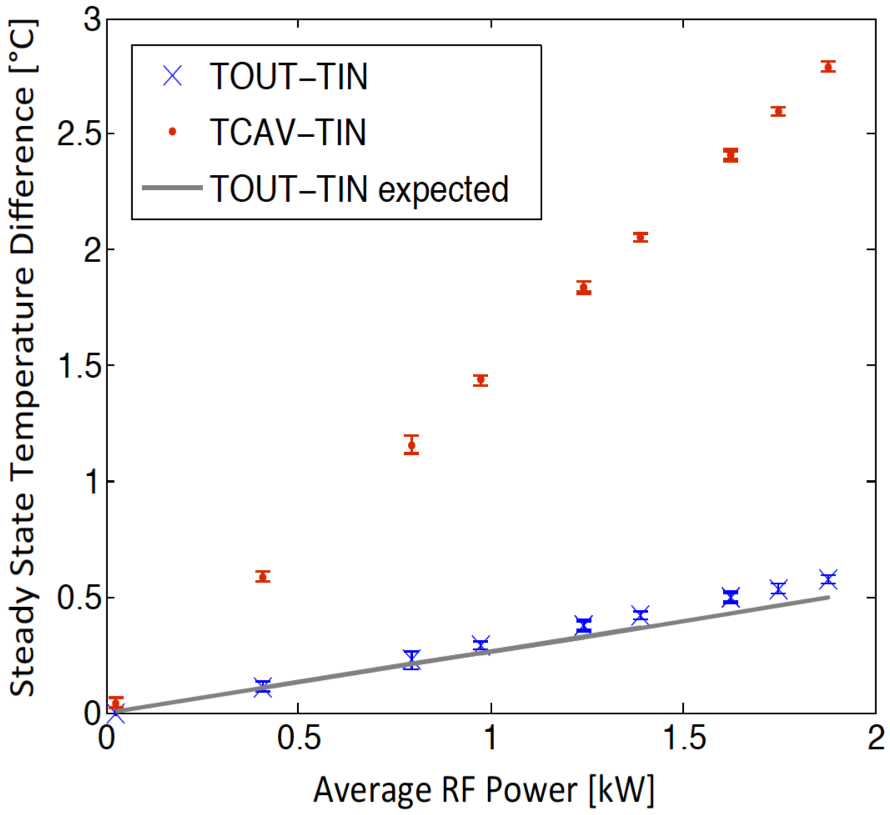
   $$\delta^l = w^{l+1} \delta^{l+1} \odot f'(z^l)$$

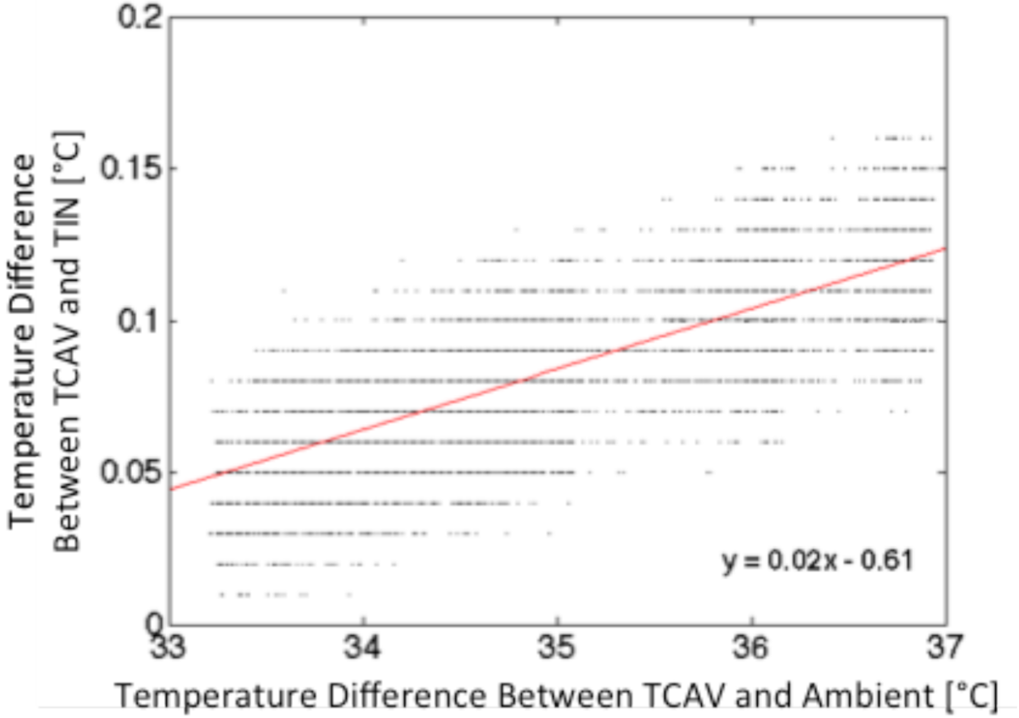
4. **Final Derivatives:**

   $$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \qquad \frac{\partial C}{\partial b_j^l} = \delta_j^l$$

FAST Gun Temperature Considerations





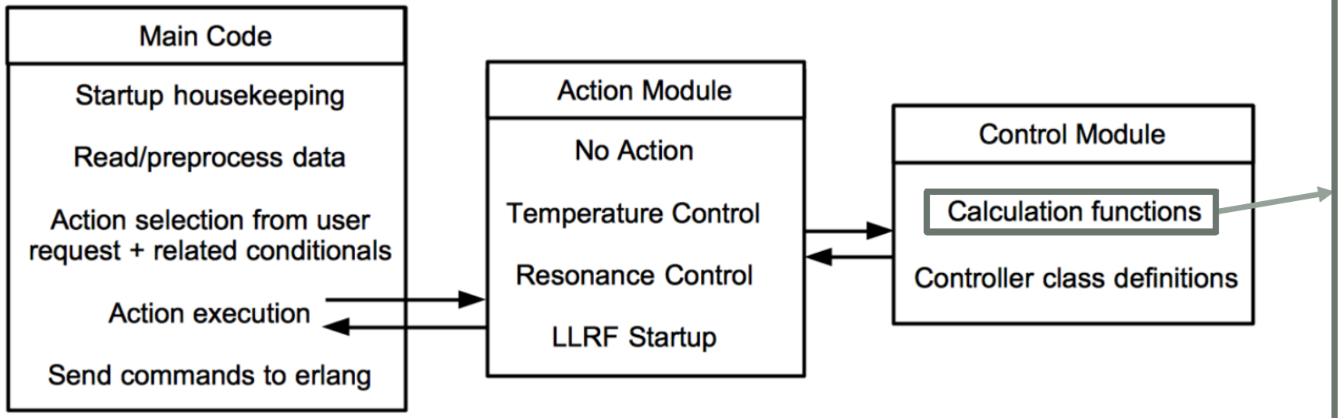


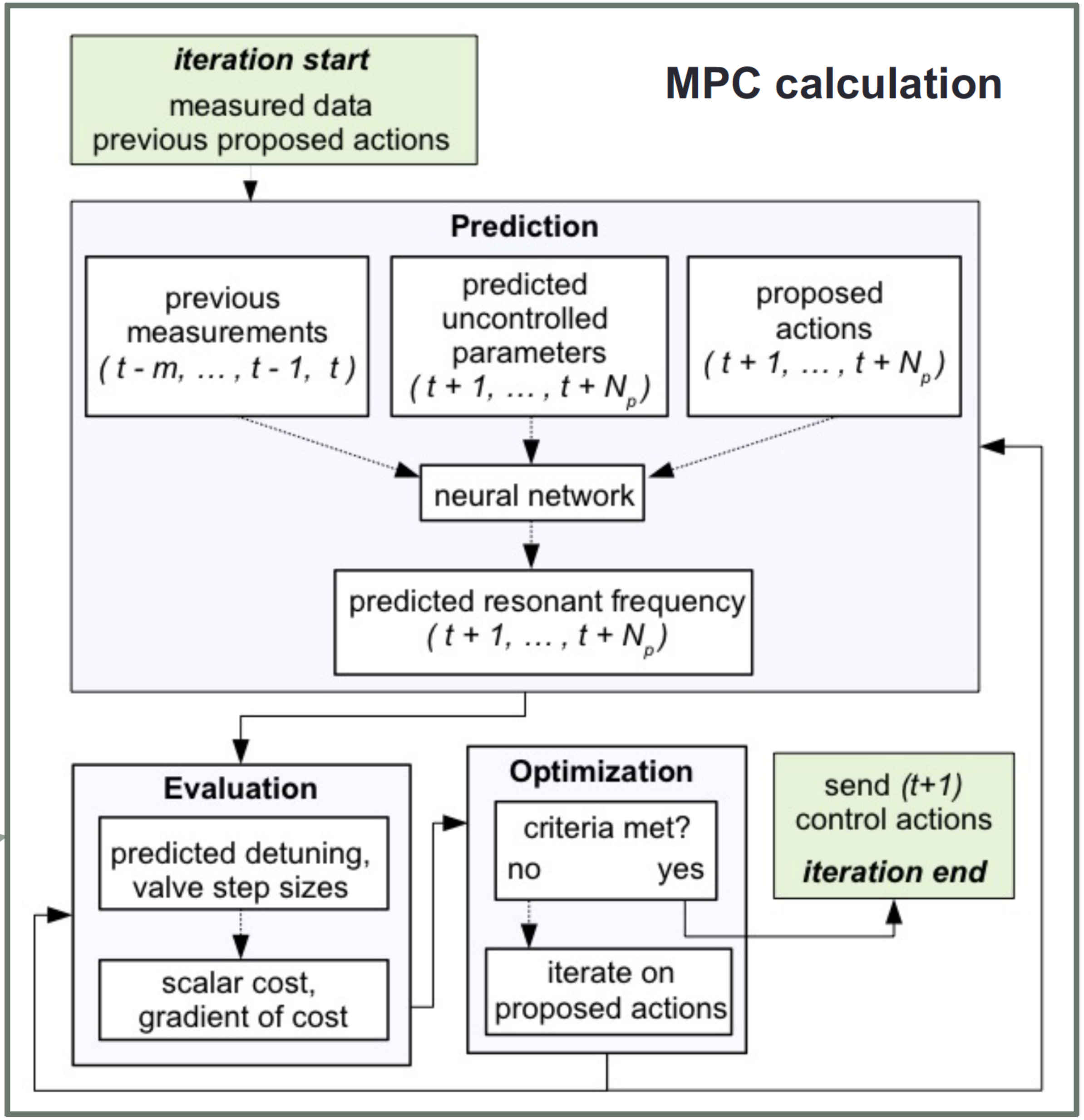
$$P_{cool} = \frac{(T_{OUT}[°C] - T_{IN}[°C]) \times (Flow\ [GPM])}{Water\ Cooling\ Capacity\ \left[\frac{GPM-°C}{kW}\right]}$$

$$P_{cool} = P_{IN} \approx P_{RF_{avg.}}$$

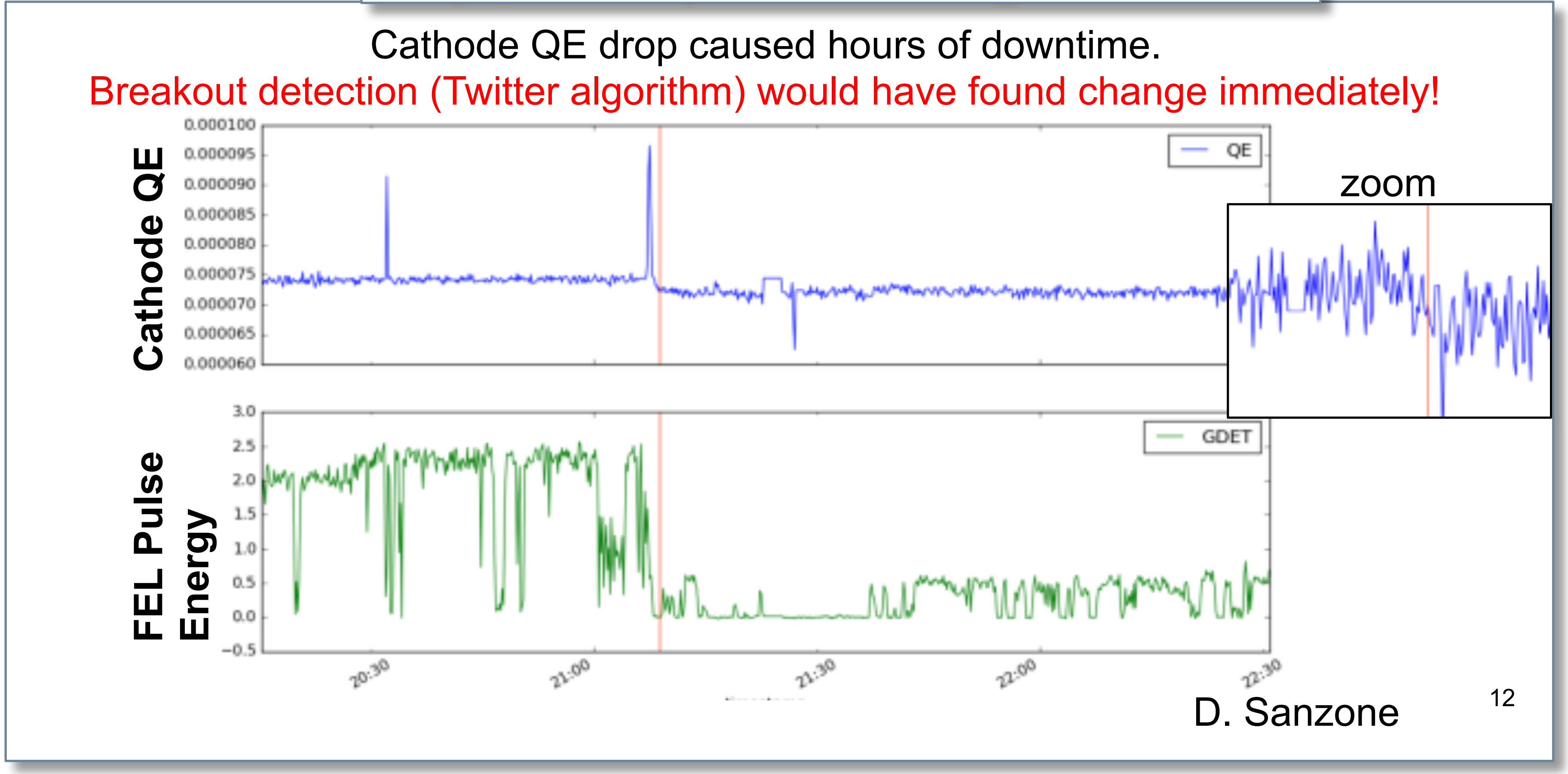
In the resonance control framework

- N_p future time steps
- *m* previous measurements for each input variable
- “actions” are vane and wall flow valve settings





Example from D. Ratner presentation at 2017 DS@HEP:



Cathode QE drop caused hours of downtime.

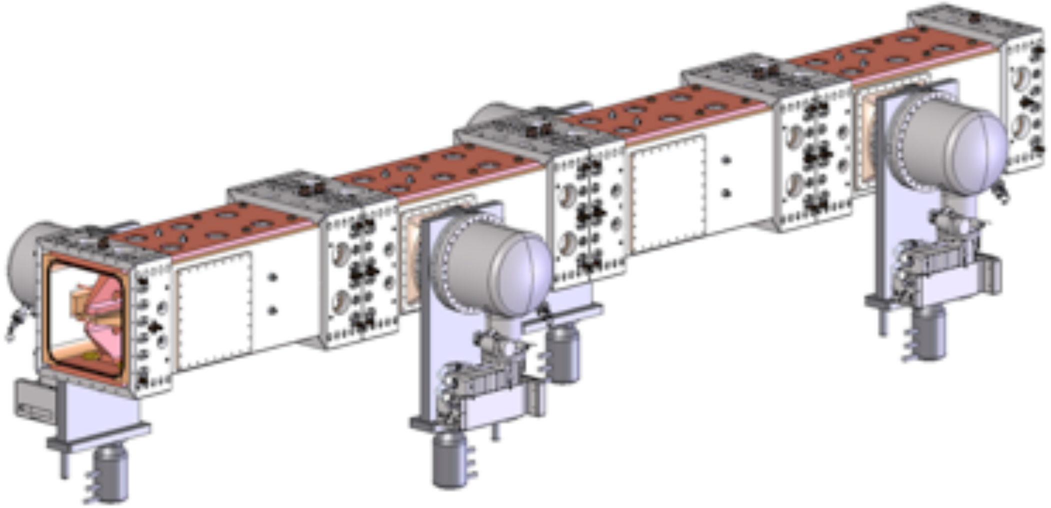
Breakout detection (Twitter algorithm) would have found change immediately!

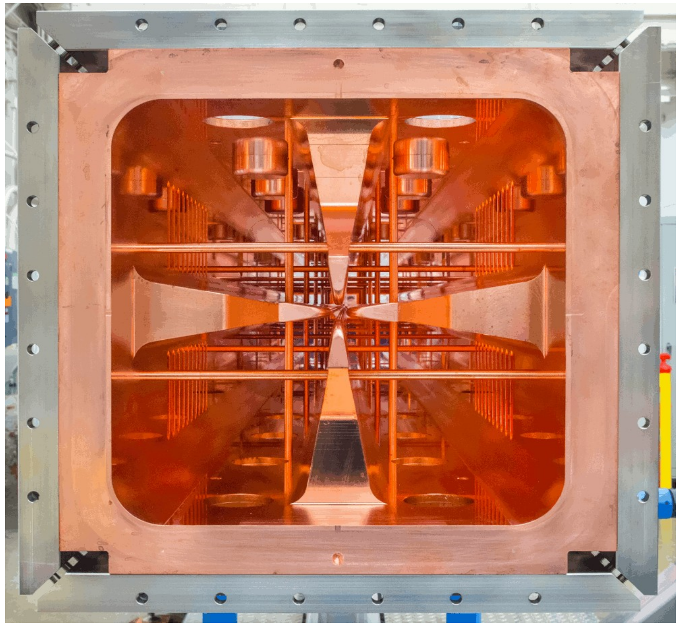
D. Sanzone

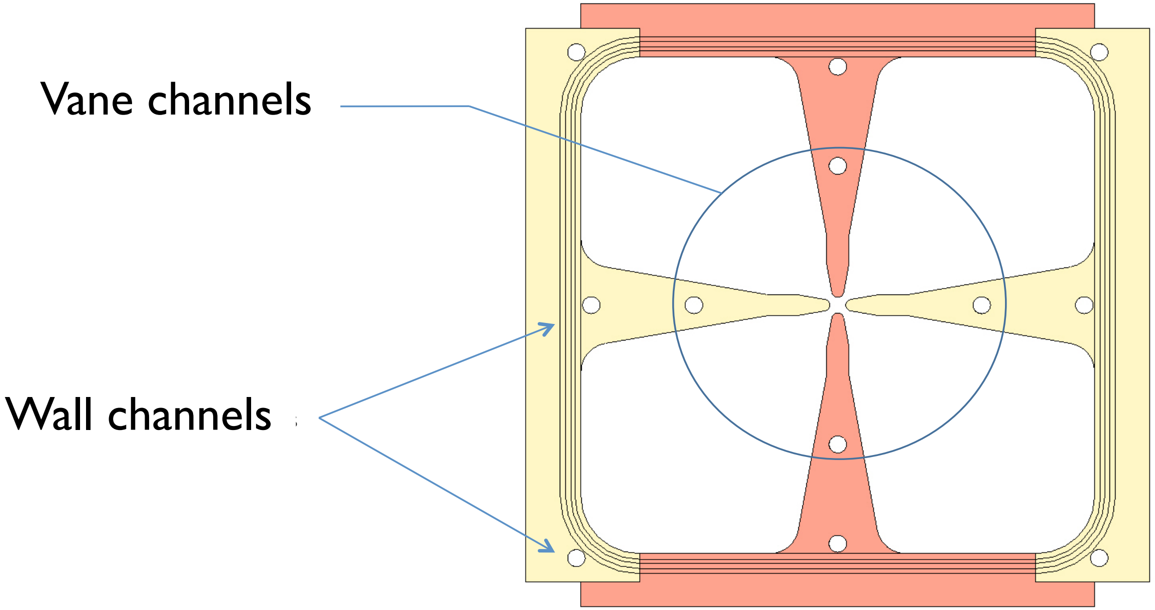
PXIE RFQ

3-kHz max. freq. shift

0.1-°C water stabilization







Vane channels

Wall channels

All images courtesy LBNL, D. Li, A. Lambert

FAST Photoinjector

RF electron gun at the Fermilab Accelerator Science and Technology (FAST) facility

— Long, variable time delays
— Tight tolerances
— Recursive behavior
— Two controllable parameters

FAST RF Gun Parameters	
Gun Parameters	
Type	Photoinjector
Number of cells	1½
RF Mode	$TM_{010,\pi}$
Loaded Q	~11,700
RF Frequency	1.3 GHz
Frequency Shift	23 kHz/°C
Nominal Operating Parameters	
Macropulse Duration	1 ms
Repetition Rate	1–5 Hz
Bunch Frequency	3 MHz
Design Gradient	40–45 MV/m
Power Source	5 MW Klystron

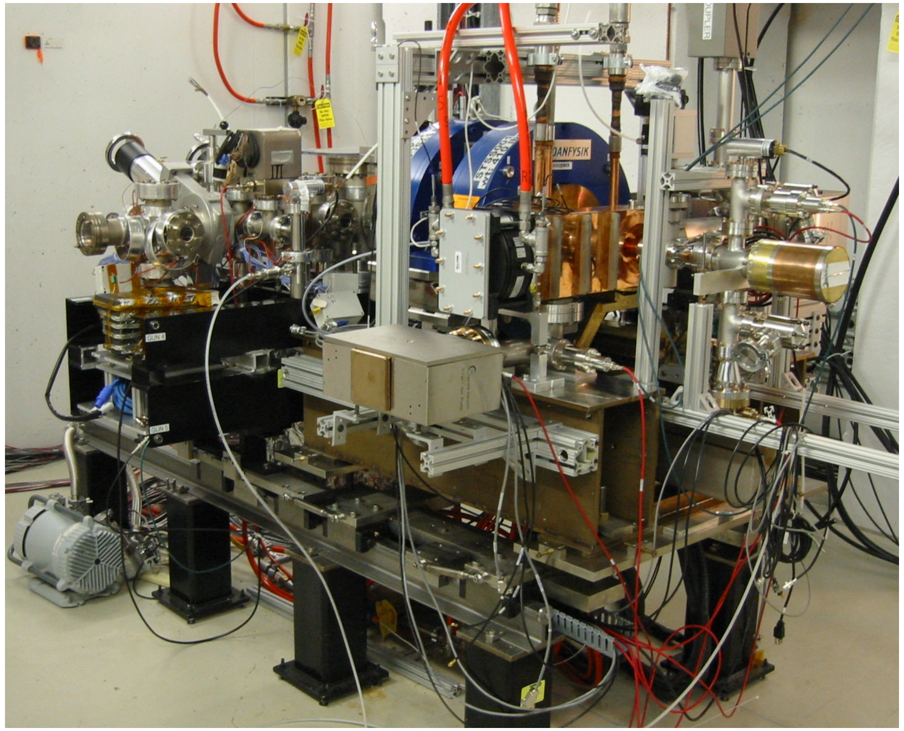


Photo: P. Stabile

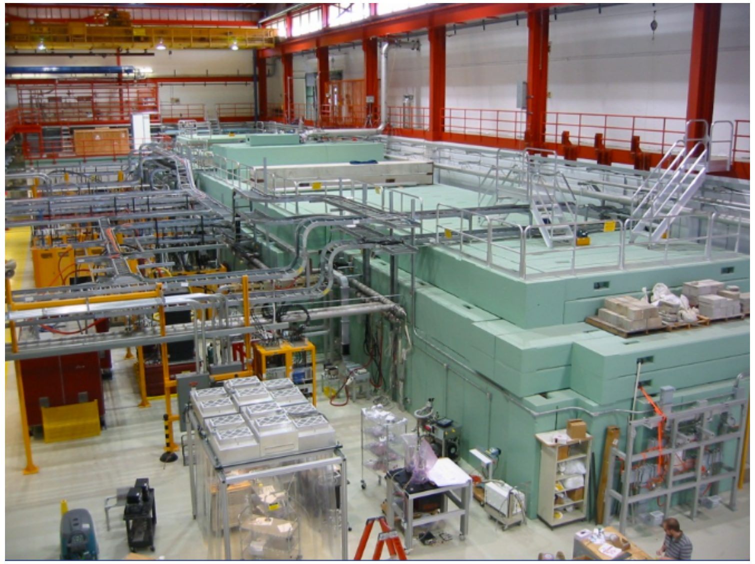


Photo: E. Harms

PIP-II RFQ

Right now: 100s to 5ms pulse at 10 Hz

~100 kW forward RF power

PXIE RFQ Parameters	
RFQ Design Parameters	
RF frequency	162.5 MHz
Q-factor	~13,900
Loaded Q	~7,000
Physical Length	4.45 m (2.4 wavelengths)
Vane-to-Vane Voltage	60 kV
Estimated Power Dissipation	< 100 kW
RF Repetition Rate	pulsed – CW
Beam Parameters	
Current	0.5 – 10 mA (nominal 5 mA)
Input Energy	30 keV
Output Energy	2.1 MeV

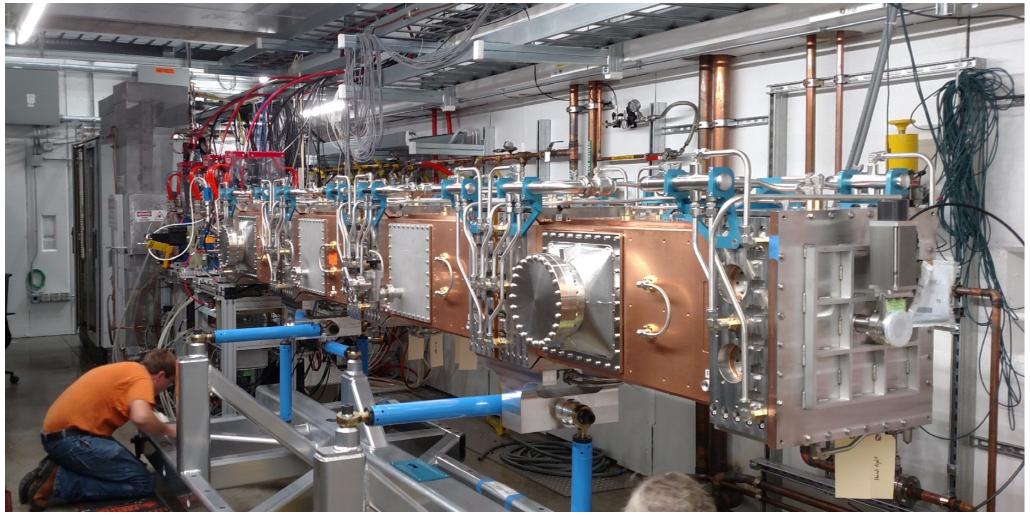
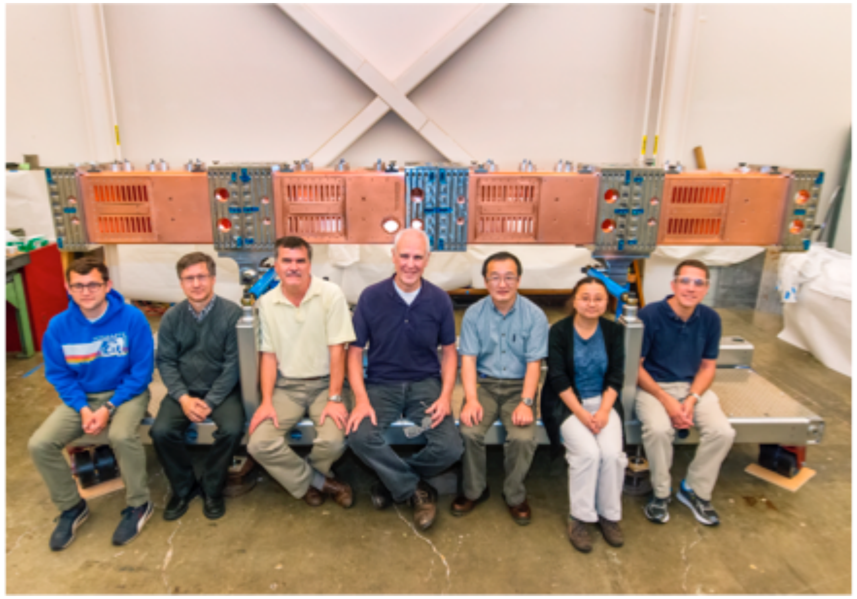


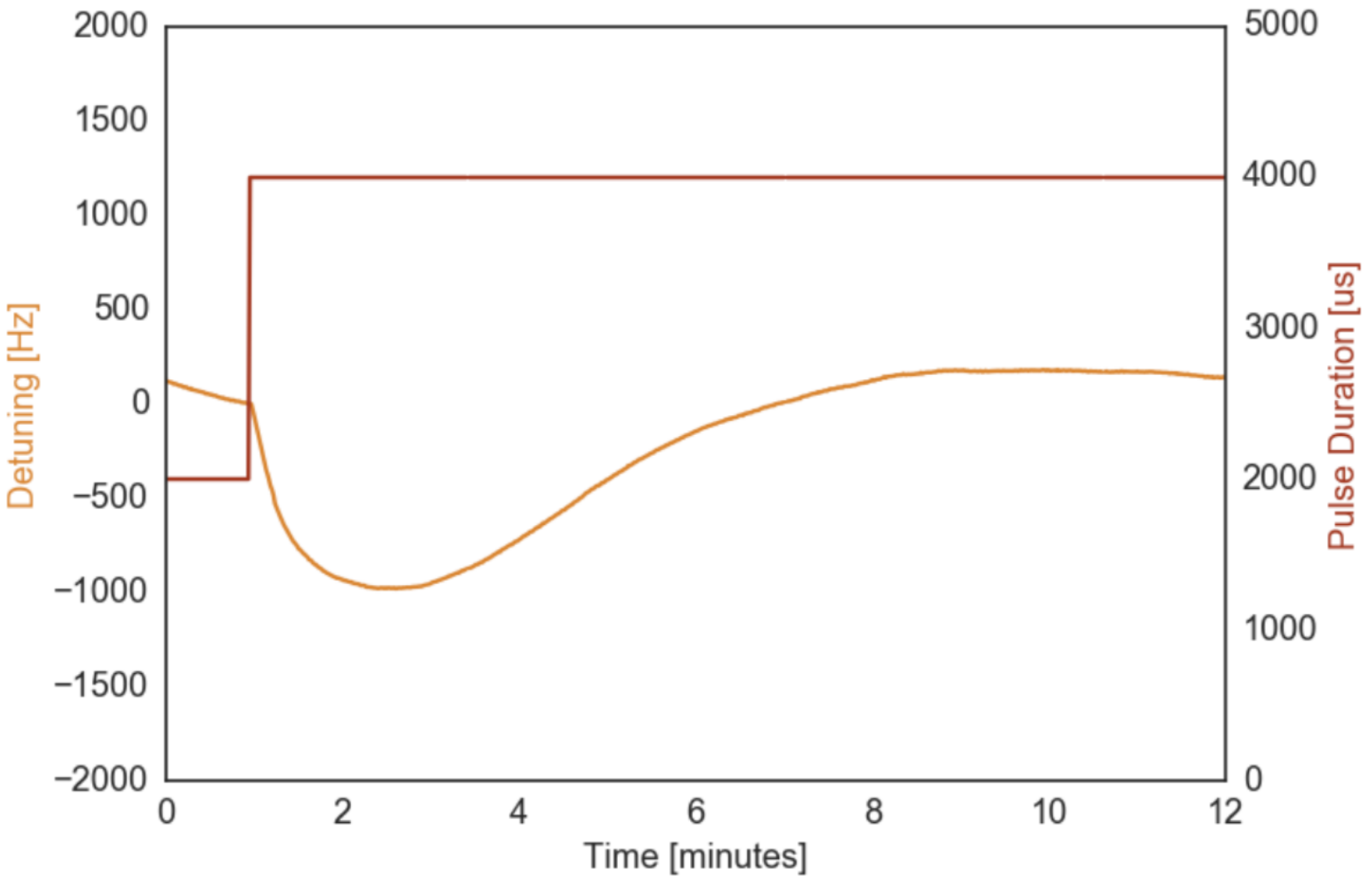
Photo: J. Steimel

Constructed by LBNL



High-intensity RFQ for the PIP-II Injector Experiment (PXIE)

- — Time delays
- — Large, dynamic frequency response
- — Tight tolerances
- — Coupling
- — Recursive behavior
- — Three controllable parameters



PI frequency control during pulsed RF operation for a 2-ms increase in pulse duration and a cavity field of 65 kV