Neural Networks for Modeling and Control of Particle Accelerators

Auralee Edelen
SLAC National Accelerator Laboratory

HB2018
Daejeon, Korea
17– 22 June, 2018

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?

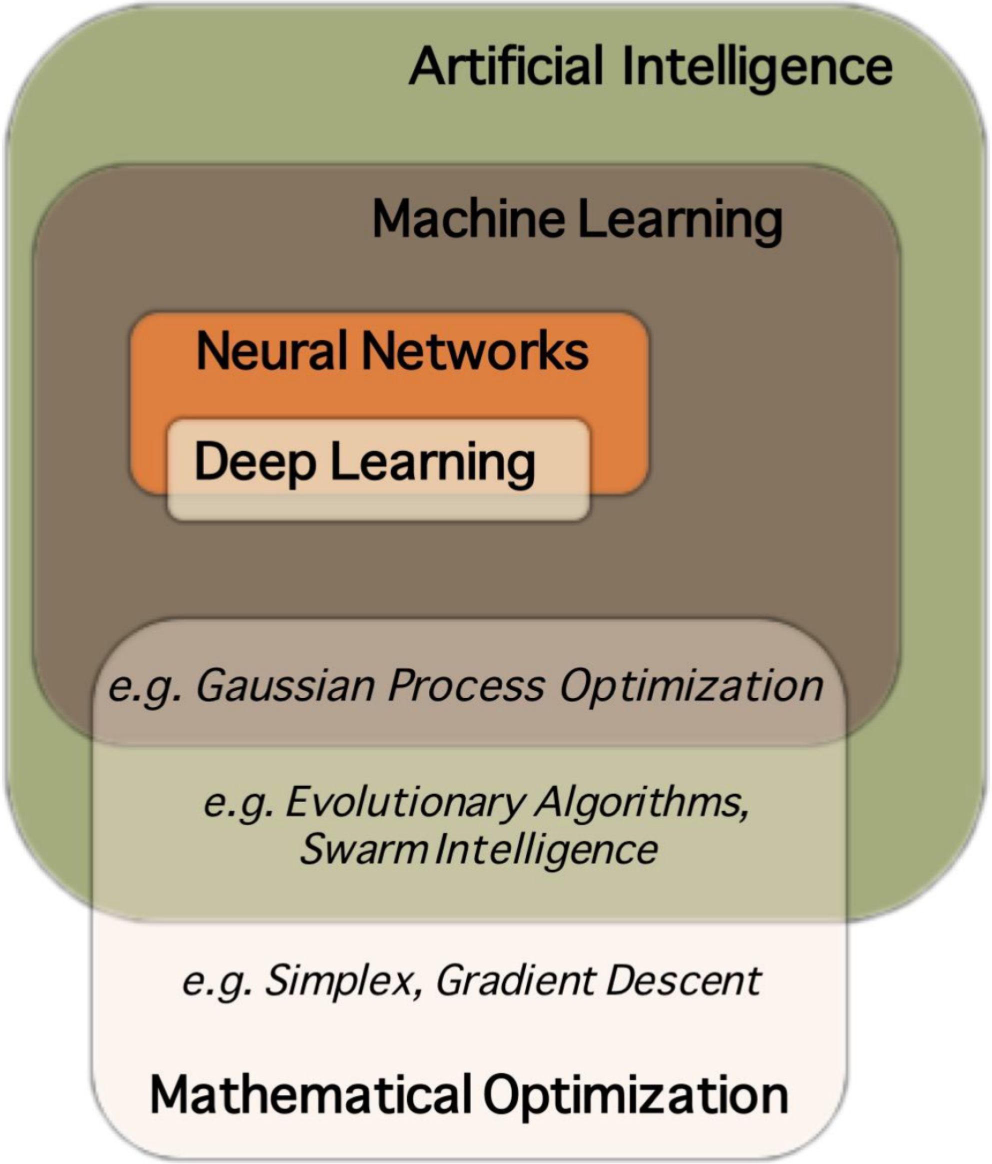
Inspiration from Operators

Fermilab Control Room Photo: Reidar Hahn, FNAL

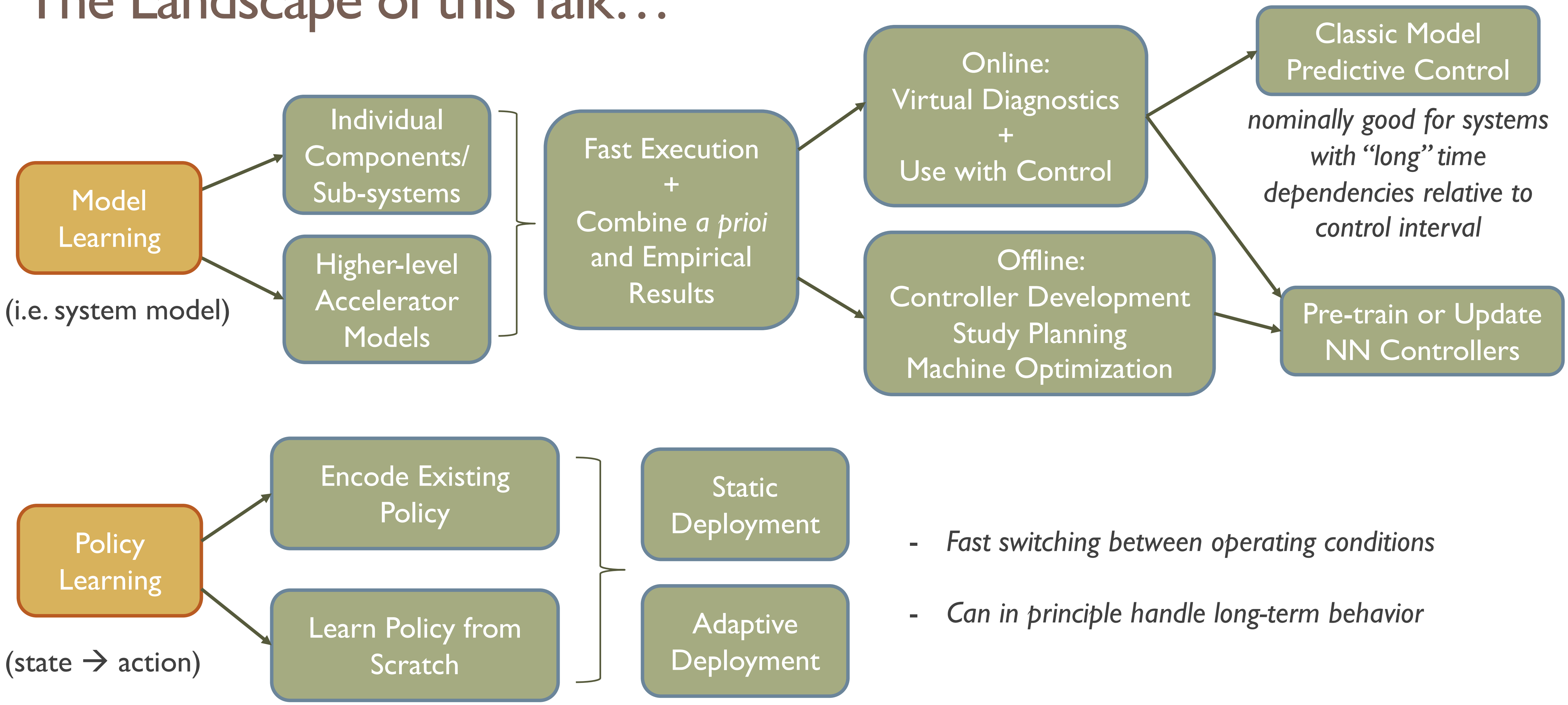
Field Taxonomy (as of now...)

- 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



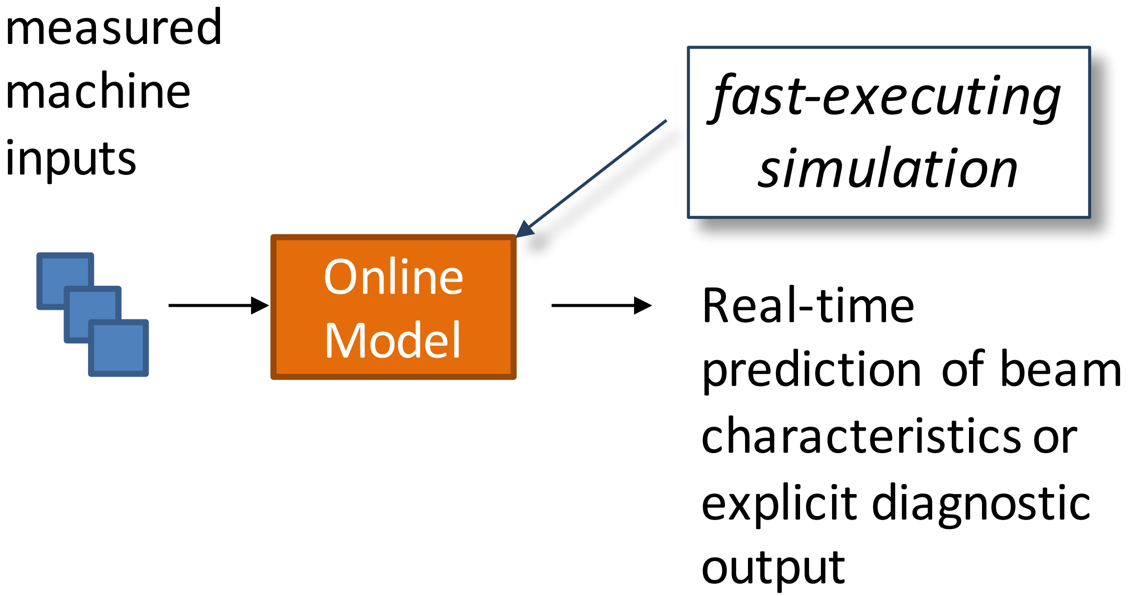
The Landscape of this Talk…



For all of the above, can in principle include image-based diagnostics directly

Online Modeling

- Use a machine model during operation
- 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 diagnostic
  - Predictive control
  - Help flag aberrant behavior
  - *Bonus: control system development*



Online Modeling

- Use a machine model during operation
-

- 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 diagnostic
  - Predictive control
  - Help flag aberrant behavior
  - *Bonus: control system development*

One approach: faster modeling codes

Simpler models (tradeoff with accuracy)

analytic calculations — *e. g. J. Galambos, et al., HPPA5, 2007*

Parallelization and GPU-acceleration of existing codes

HPSim/PARMILA — *X. Pang, PAC13, MOPMA13*

elegant — *I.V. Pogorelov, et al., IPAC15, MOPMA035*

Improvements to modeling algorithms

Lorentz-boosted frame — *J.-L. Vay, Phys. Rev. Lett. 98 (2007) 130405*

Online Modeling

- Use a machine model during operation
-

- 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 diagnostic
  - Predictive control
  - Help flag aberrant behavior
  - *Bonus: control system development*

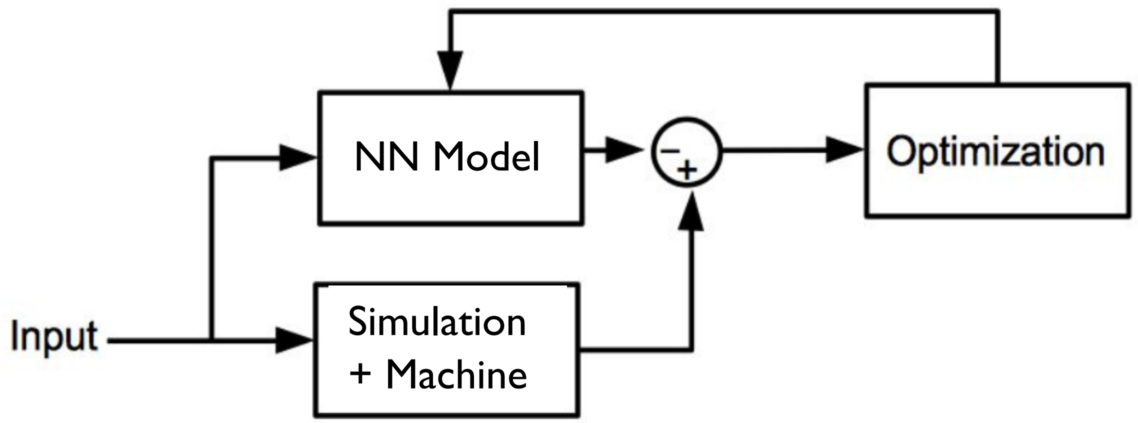
Another approach: machine learning model

Once trained, neural networks can execute quickly

Train on data from slow, high-fidelity simulations

\+

Train on measured data



Online Modeling

- Use a machine model during operation

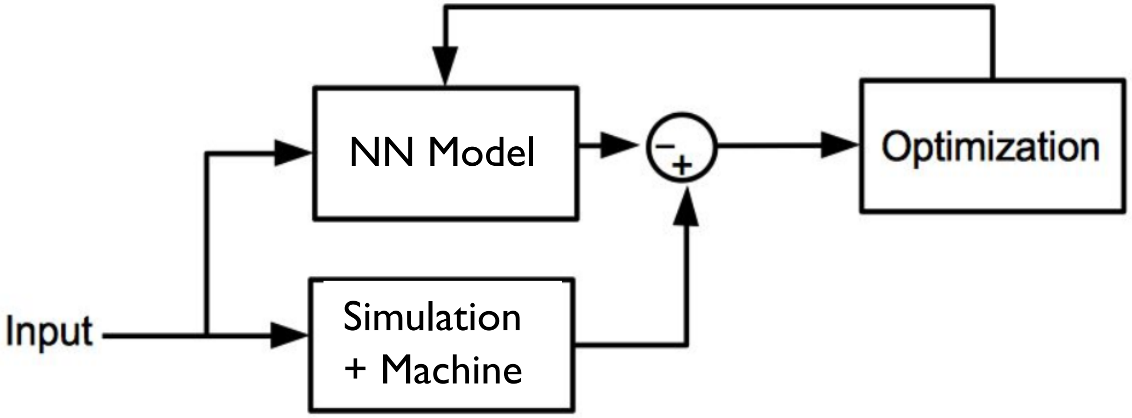
- Ideally:
  - Fast-executing, but accurate enough to be useful
  - Use measured inputs directly from machine
  - Combine *a priori* knowledge + learned parameters

- Applications:
  - A tool for operators + virtual diagnostic
  - Predictive control
  - Help flag aberrant behavior
  - *Bonus: control system development*

Another approach: machine learning model

Once trained, neural networks can execute quickly

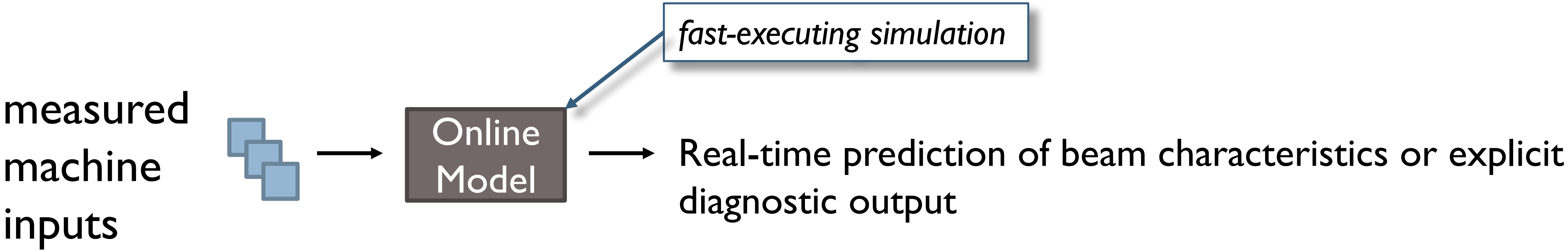
Train on data from slow, high-fidelity simulations
+
Train on measured data



An initial study at Fermilab:
A. L. Edelen, et al. NAPAC16, TUPOA51
One PARMELA run with 2-D space charge: ~ 20 minutes
Neural network model: ~ a millisecond

Virtual Diagnostics

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

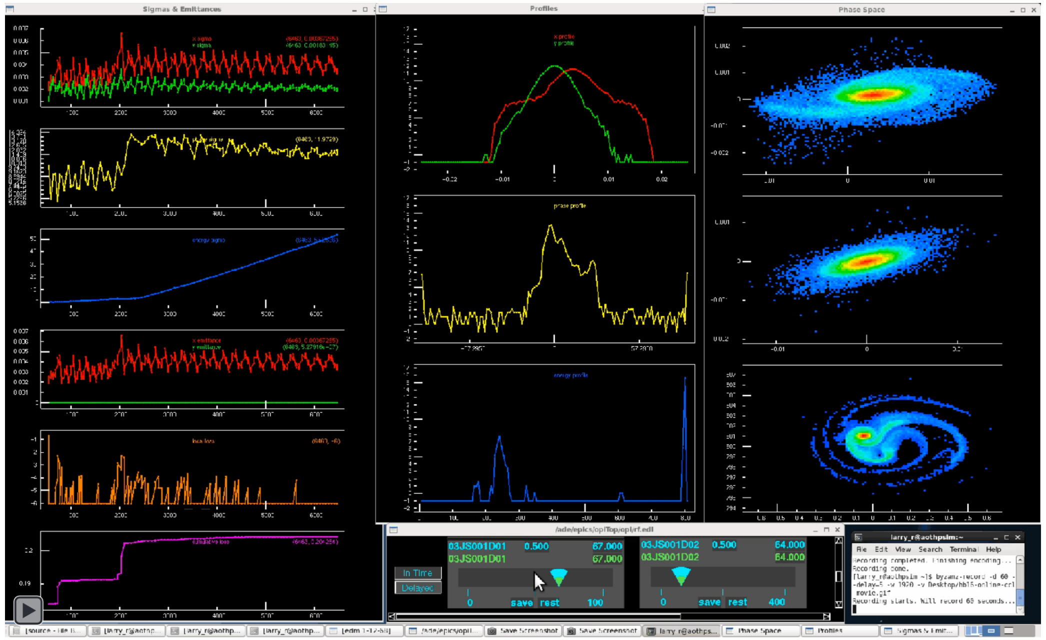


Virtual Diagnostics

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

fast-executing simulation

measured machine inputs → Online Model → Real-time prediction of beam characteristics or explicit diagnostic output

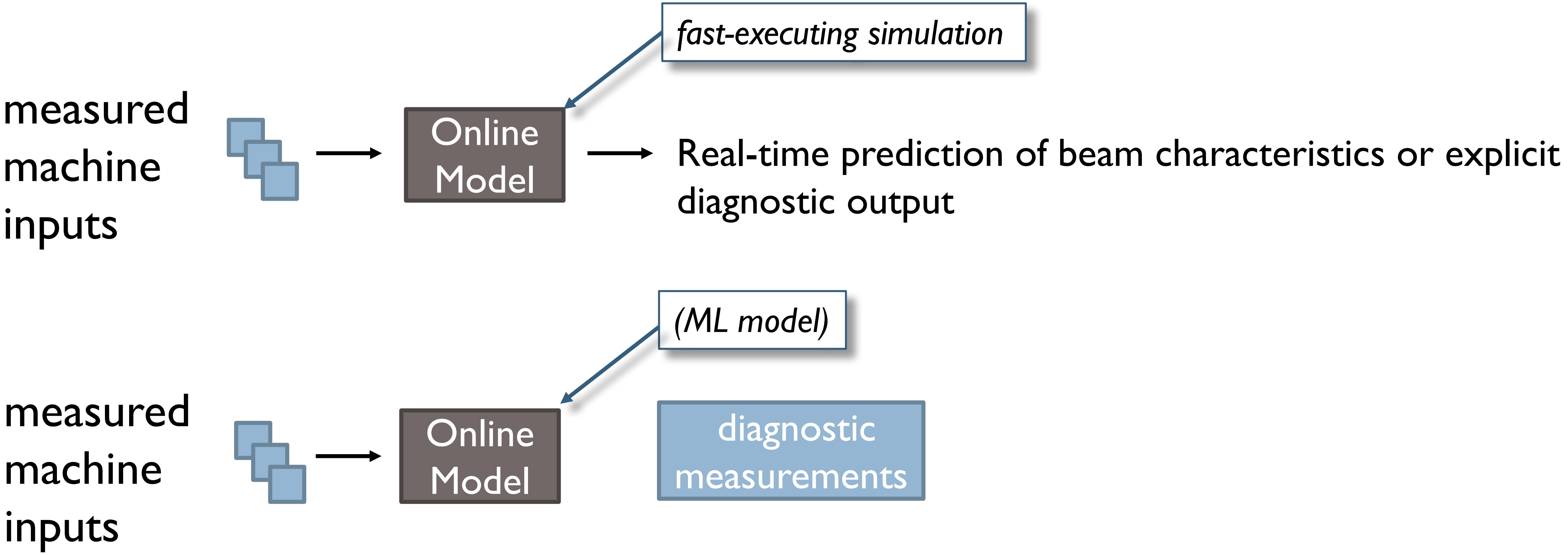


e.g. GPU-accelerated HPSim at LANSCE (based on PARMILA)

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

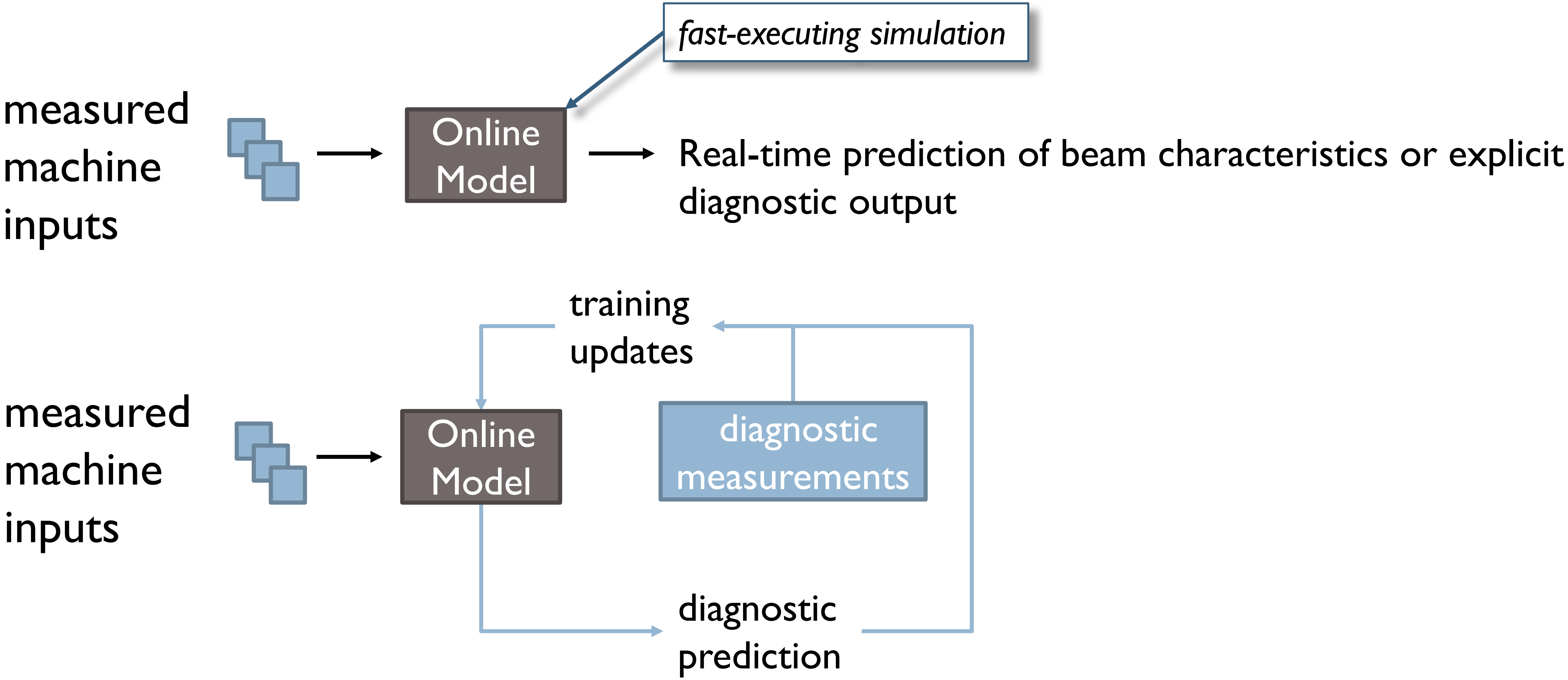
Virtual Diagnostics

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



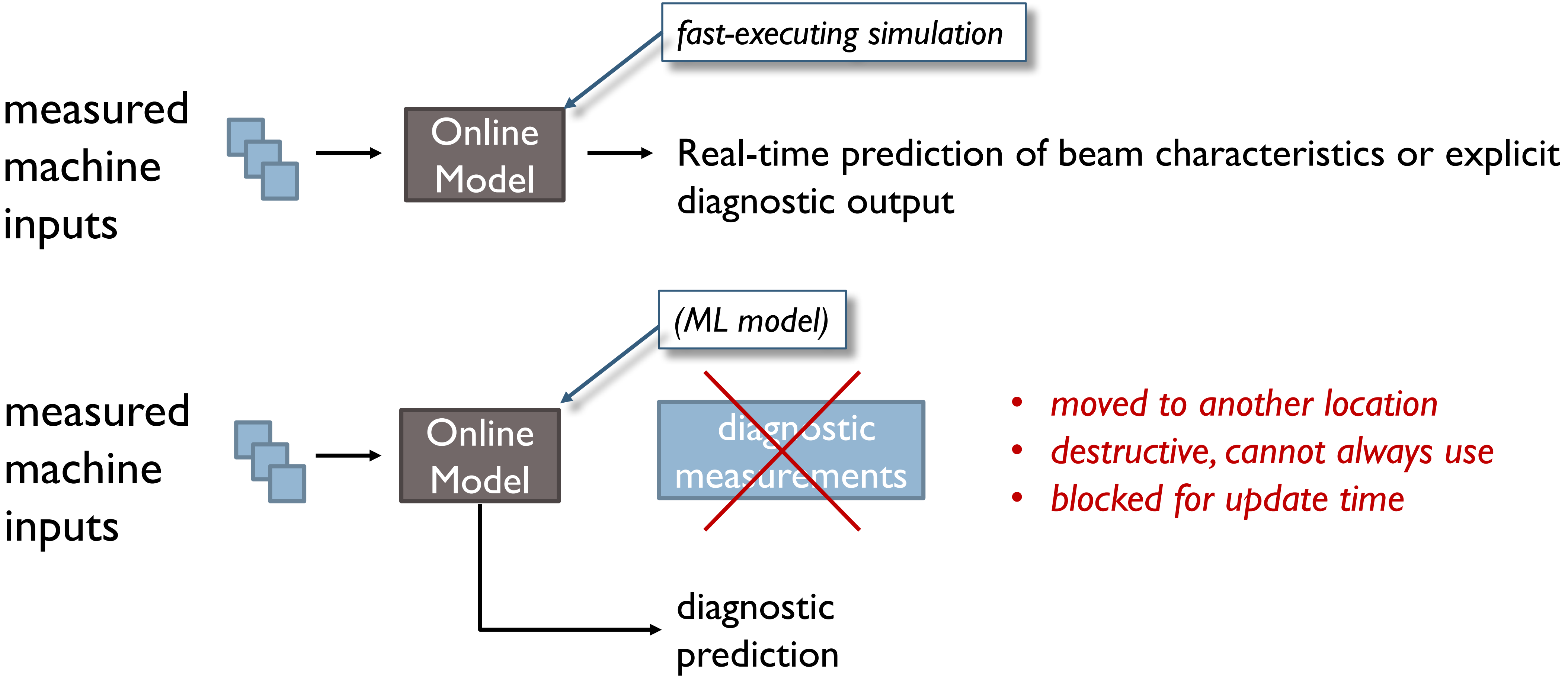
Virtual Diagnostics

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



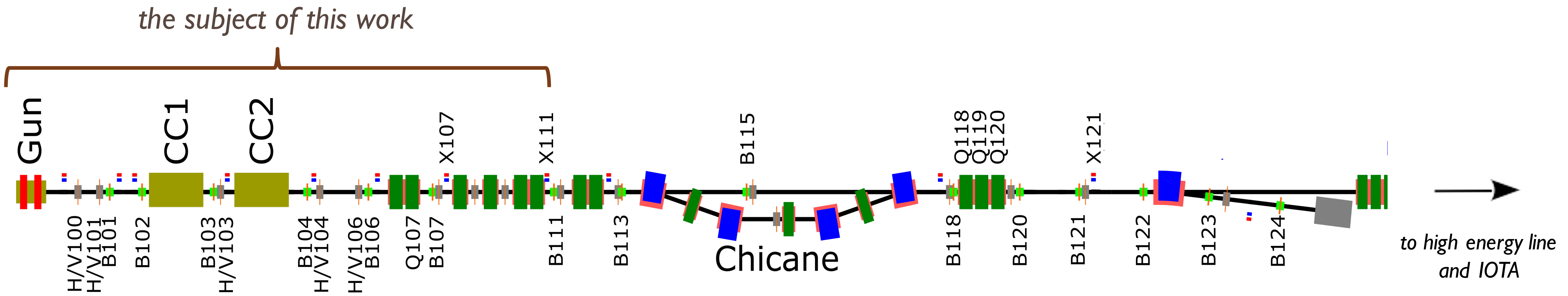
Virtual Diagnostics

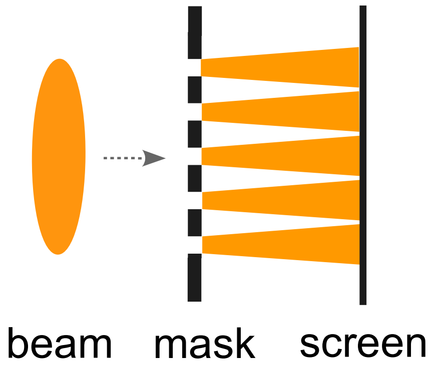
Predict what diagnostics might look like when they are unavailable or don’t exist

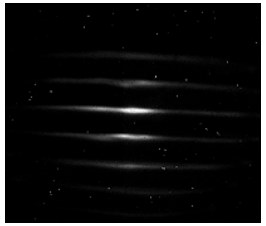


- *moved to another location*
- *destructive, cannot always use*
- *blocked for update time*

Virtual Diagnostics at Fermilab's FAST Facility







fit to obtain subset of phase space parameters

$\langle y^2 \rangle$

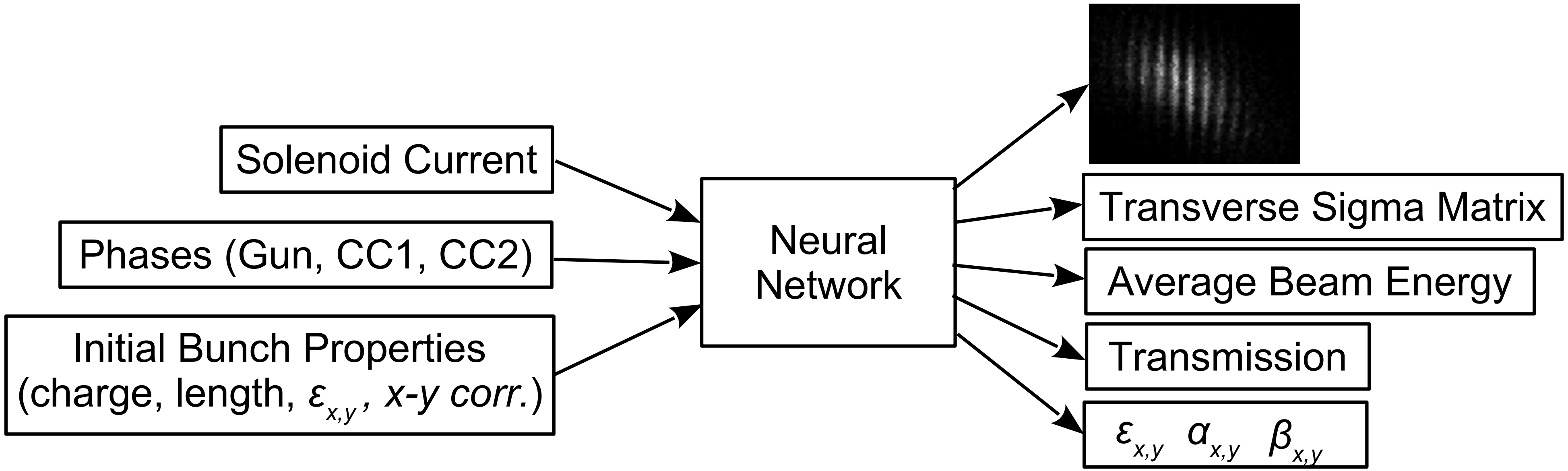
$\langle yy' \rangle$

$\langle y'^2 \rangle$

Multi-slit emittance measurement after the second capture cavity (X107 to X111) takes 10-15 seconds

→ *can we get an online prediction of what this intercepting diagnostic would show?*

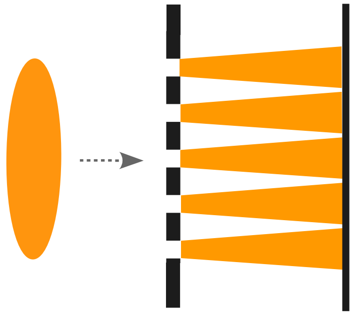
Initially limit the scope…

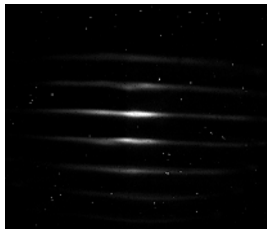


Could in principle use measured data alone, but want to be efficient with machine time

→ use simulation data to fill out the training set

gun phase scans
solenoid current scans
(with two different laser intensities)





fit to obtain subset of phase space parameters

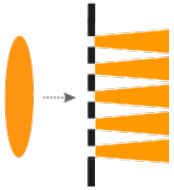
$\langle y^2 \rangle$

$\langle yy' \rangle$

$\langle y'^2 \rangle$

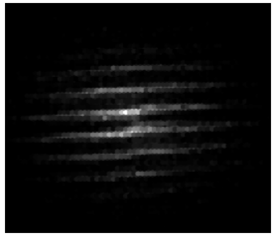
other setting combinations

OPAL



elegant

cathode → CC2
with 3-D space charge routine

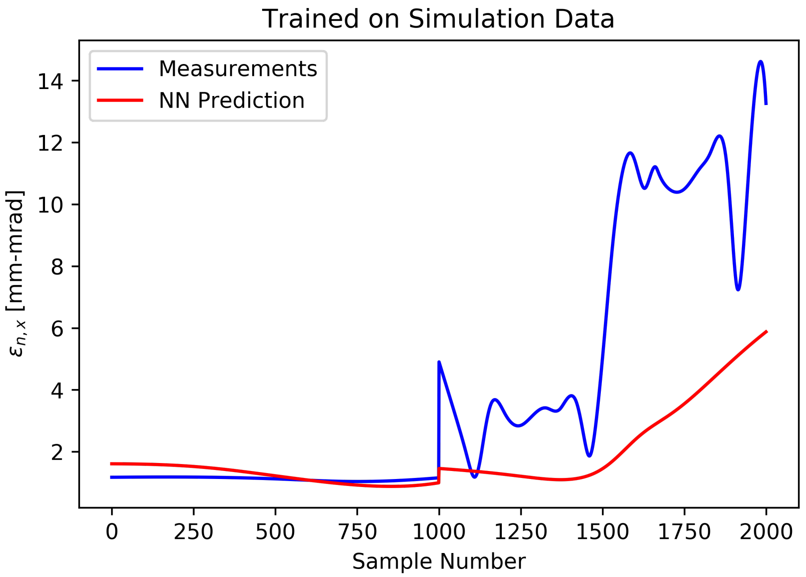


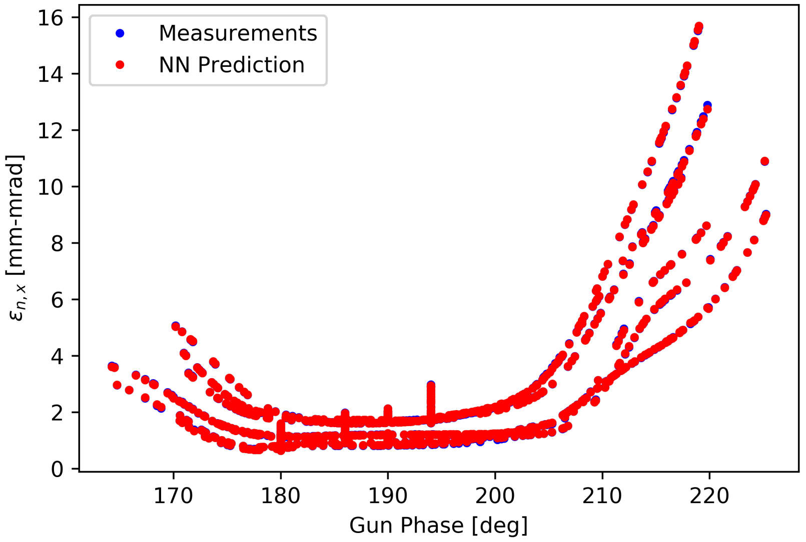
\+ full sigma matrix

Training on imperfect simulations … NN only as good as the simulation

Poor agreement between simulation and measured data for some input/output relationships

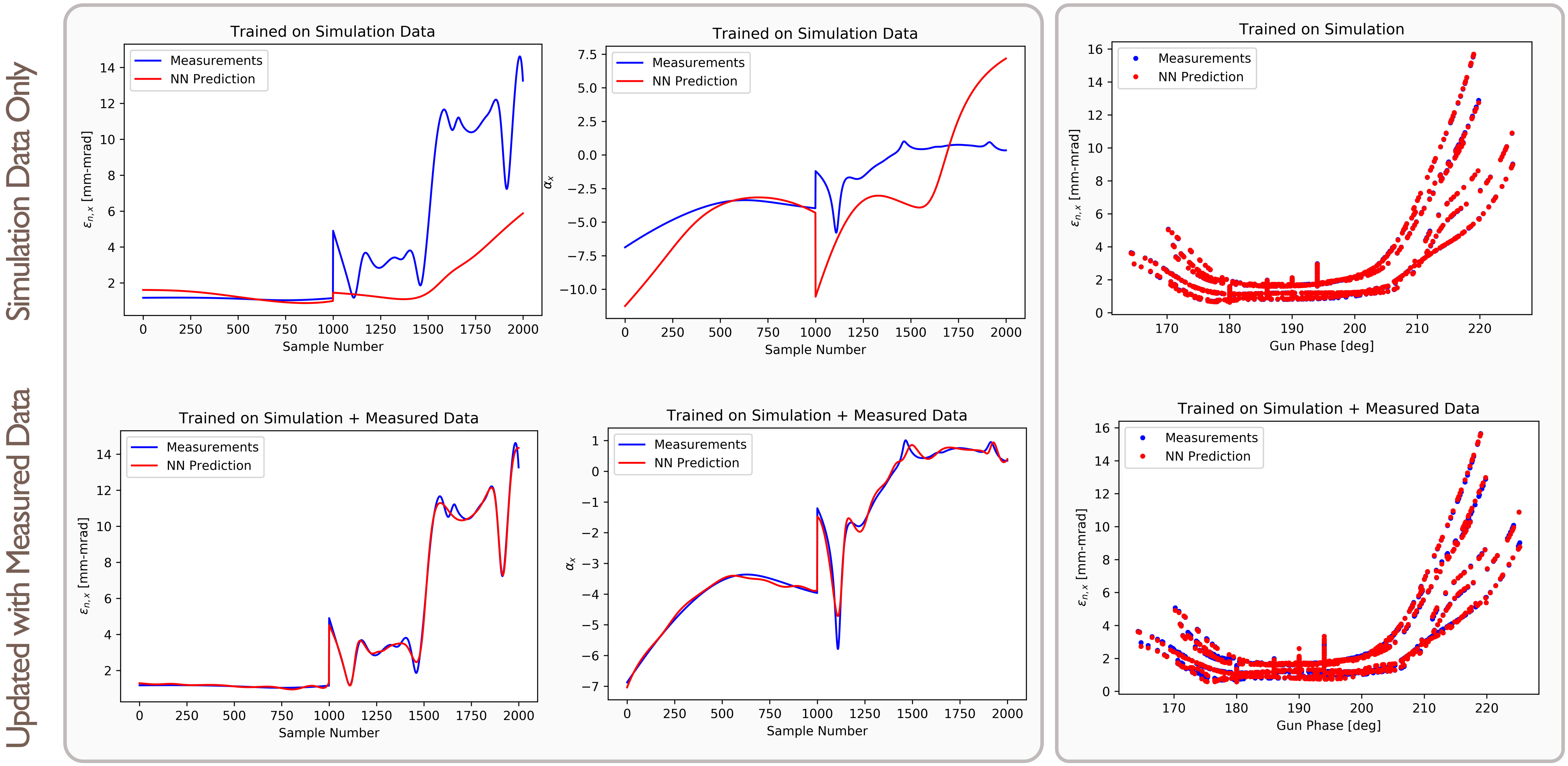
→ can we update the NN model with measured data without disrupting the other predictions?





Solenoid Scan

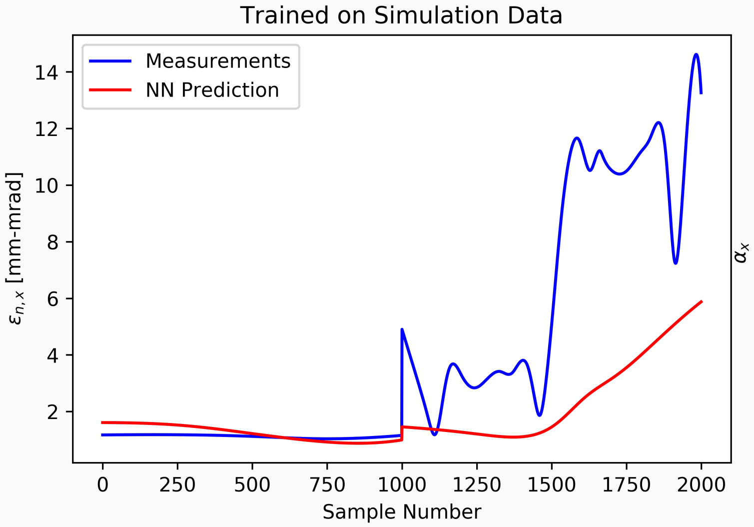
Phase Scan

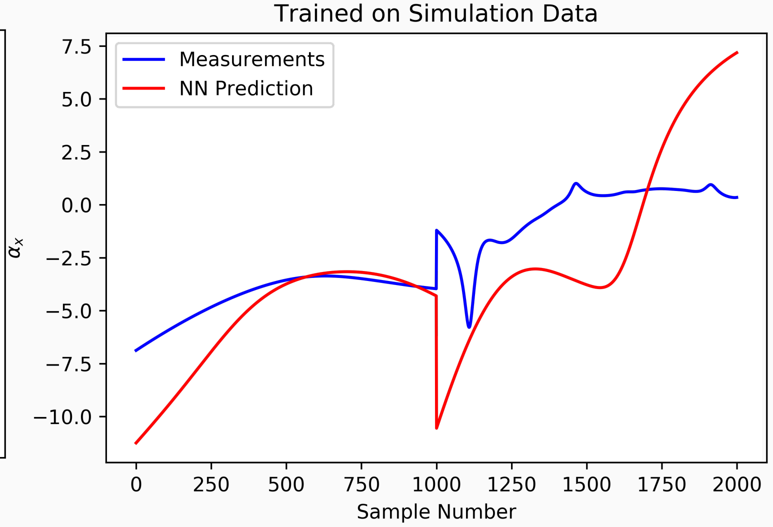


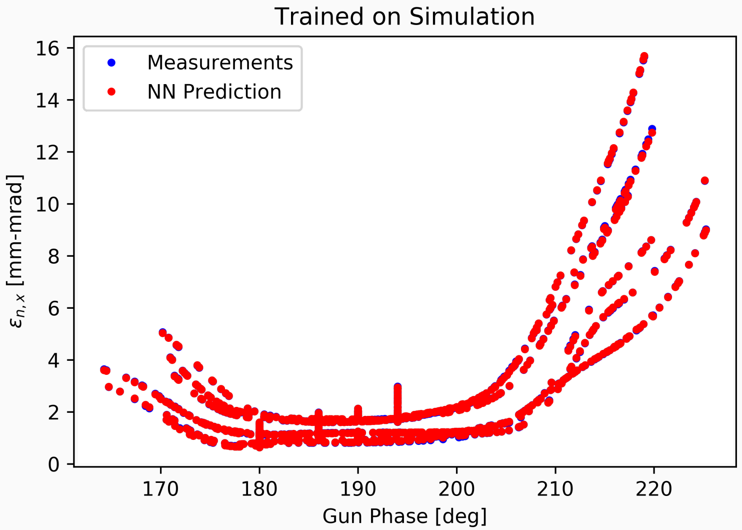
Solenoid Scan

Phase Scan

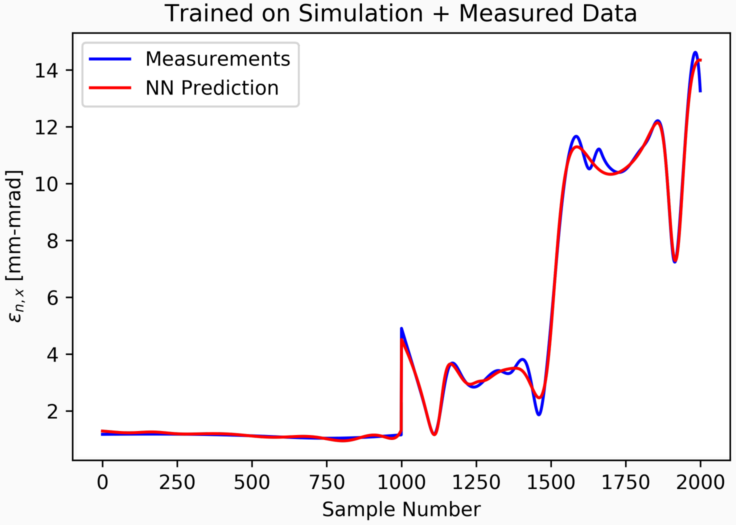
Simulation Data Only

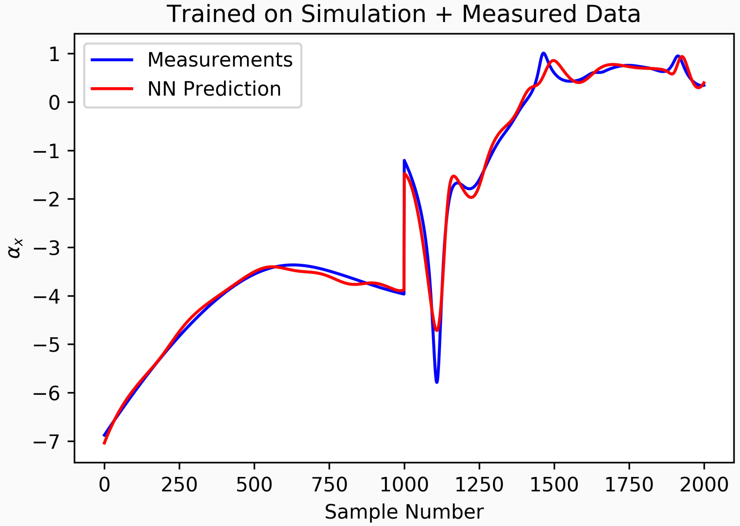


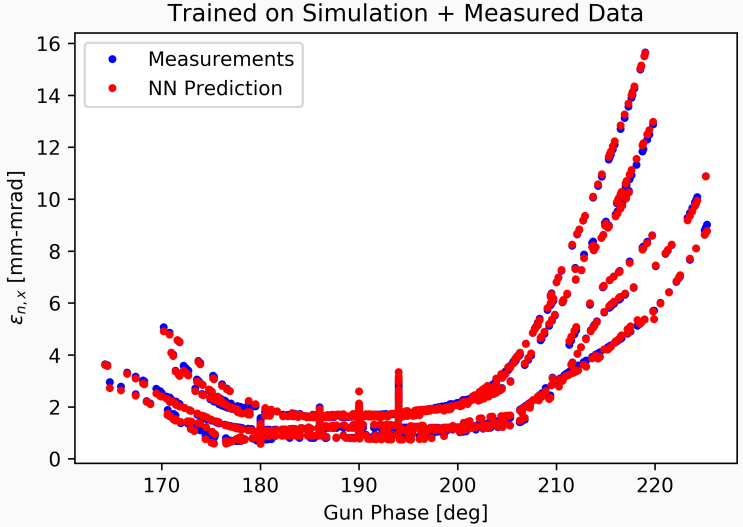




Updated with Measured Data



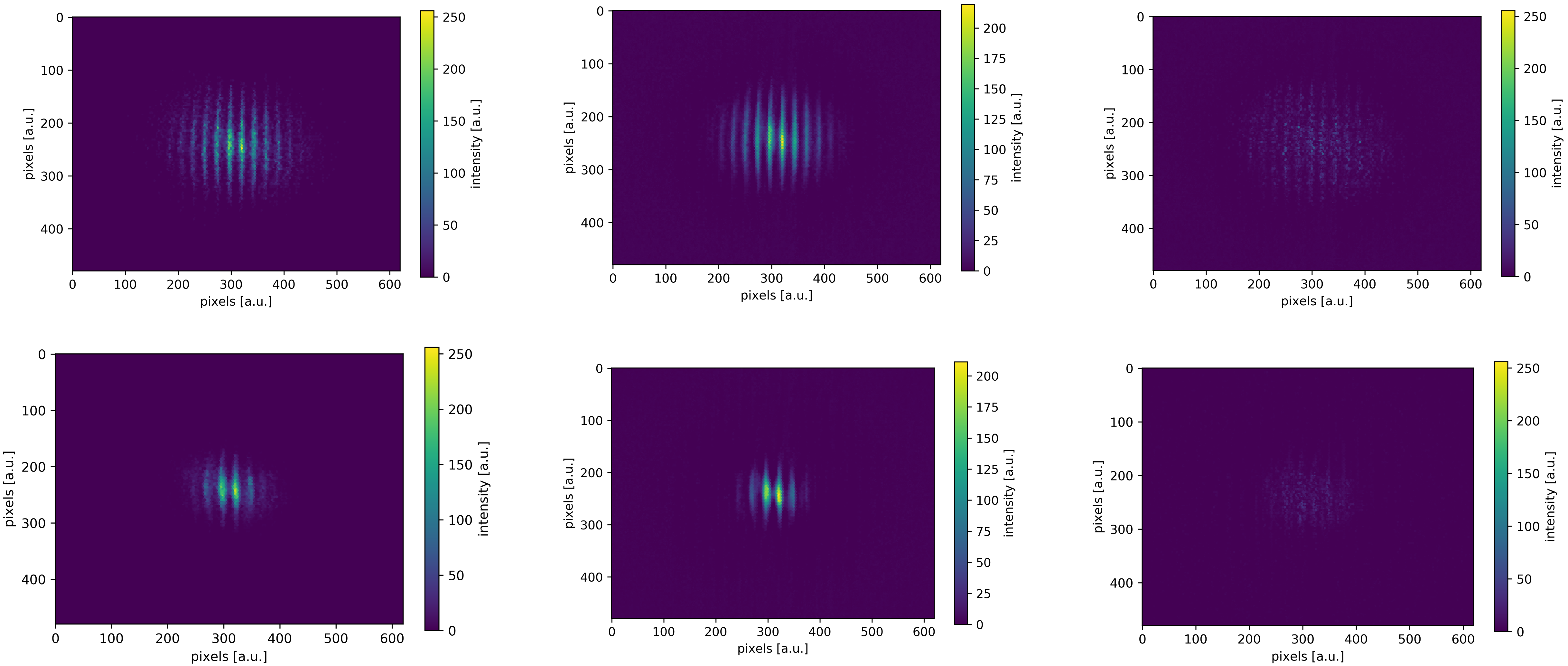




Why bother with simulation at all? → Rough initial solution facilitates training with small amount of measured data

Predicting Image Output Directly

A. L. Edelen, et al. IPAC18, WEPAF040



Simulated

NN Predictions

Difference

Work with J. Edelen, D. Edstrom, A. Halavanau, J. Ruan, P. Piot, A. Romanov, S.G. Biedron, S.V. Milton

Bigger Picture

Fast-executing, accurate machine model

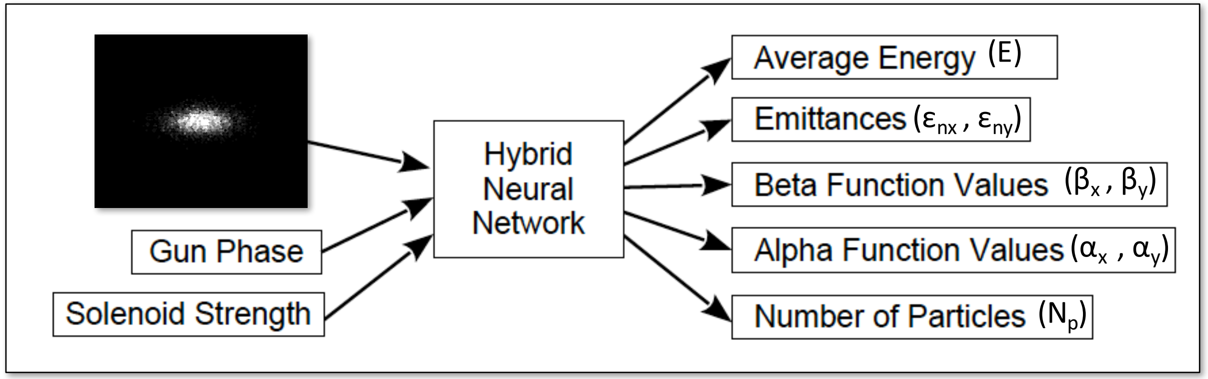
Online: *facilitate studies*

Offline: *study planning*
downstream component design
controller training

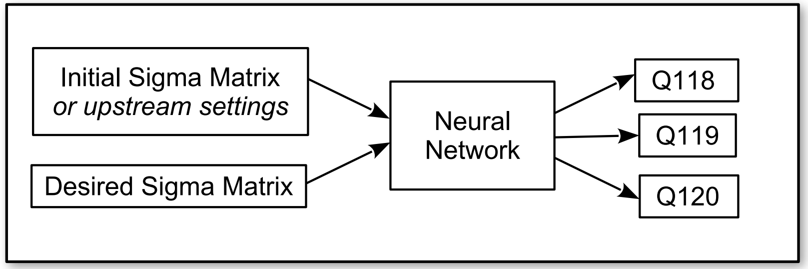
One piece of a larger set of studies:

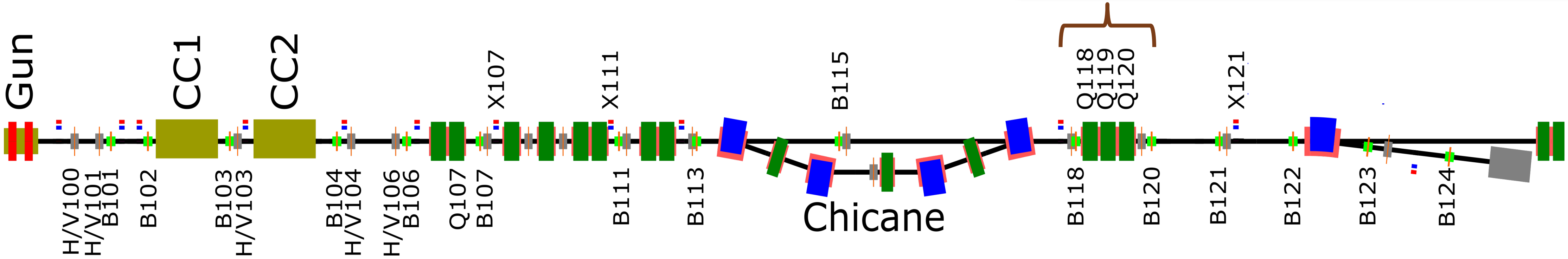
- *Accounting for laser spot changes*
- *NN controller (starting with round-to-flat beam transform)*
- ***The vision is to combine these***

Earlier work: account for changes in laser spot
A. L. Edelen, et al. NAPAC16, TUPOA51



Ongoing work: NN-based round-to-flat beam transform





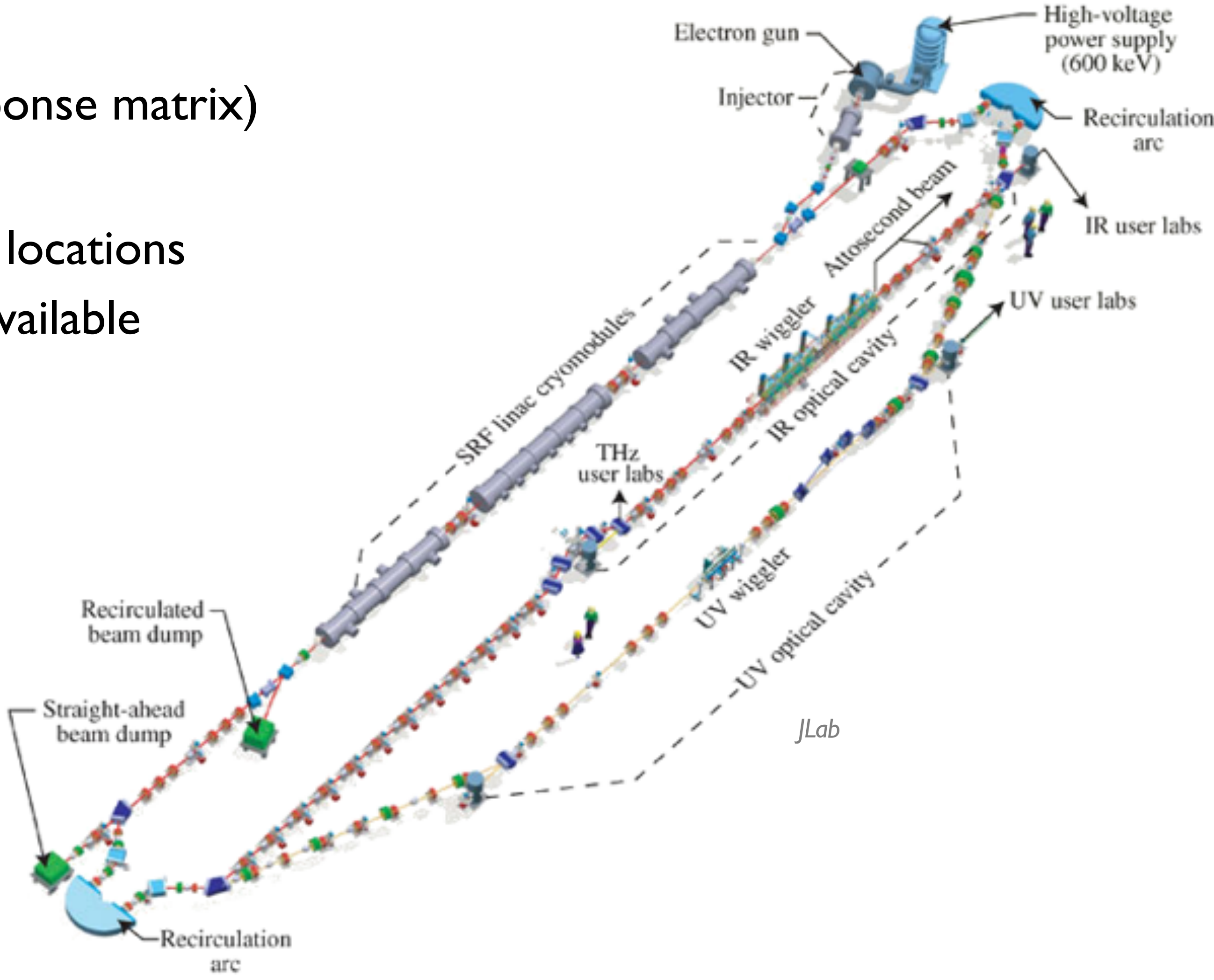
Fast Switching Between Trajectories

Work with C. Tennant and D. Douglas, JLab

- 76 BPMs, 57 dipoles, 53 quadrupoles
- Traditional approach has never worked (linear response matrix)
- Rely on a few experts 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 + transmission

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



JLab

Fast Switching Between Trajectories

Main anticipated advantage of NN over standard approach:

Adaptive control policy → adjust without interfering with operation for response measurements as often?

Handling of trajectories away from BPM center (nonlinear)

But, need to quantify this …

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

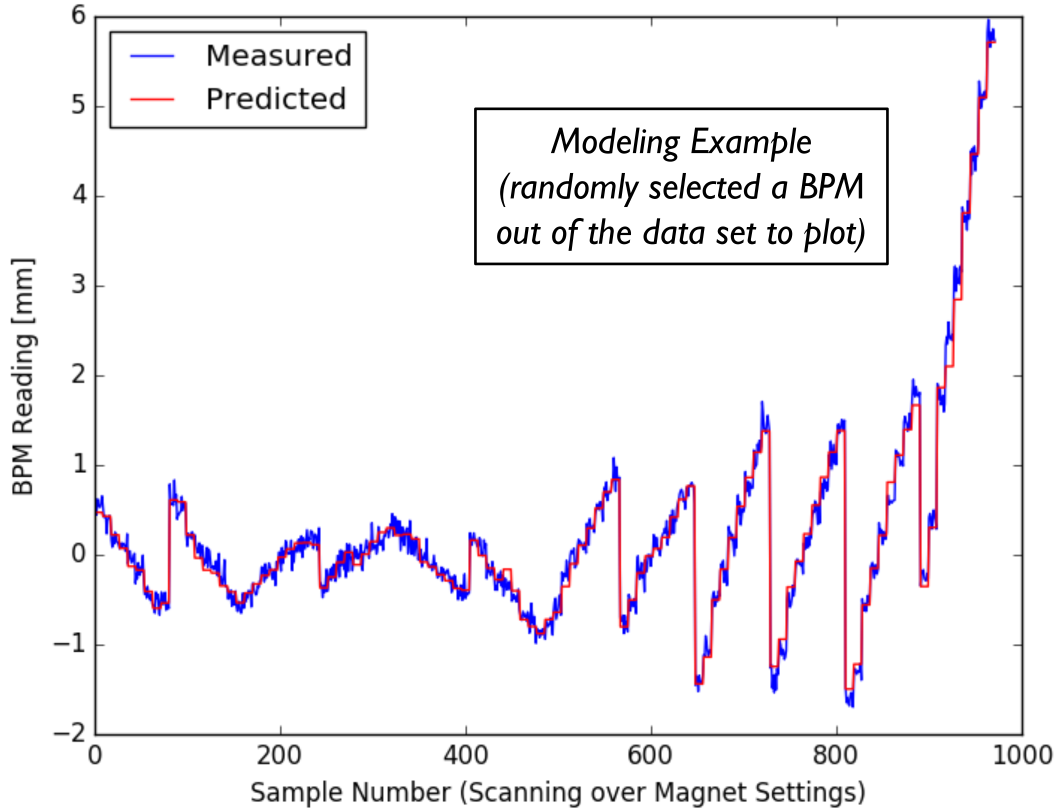
Train controller (NN policy) offline using NN model:
desired trajectory → dipole settings
(and penalize losses + large magnet settings)

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



Switching Between User Requests

- FEL facilities support a wide variety of scientific endeavors *(e.g. imaging protein structures[1], understanding processes like photosynthesis[2], origin of material properties[3])*
- Need to accommodate requests for a wide variety of photon beam characteristics
- May switch as often as every few days
- Have save/restore settings, but these are discrete, and there can be some drift in the machine
- Time spent tuning = reduced scientific output for a given operational budget

e.g. *the Linac Coherent Light Source*

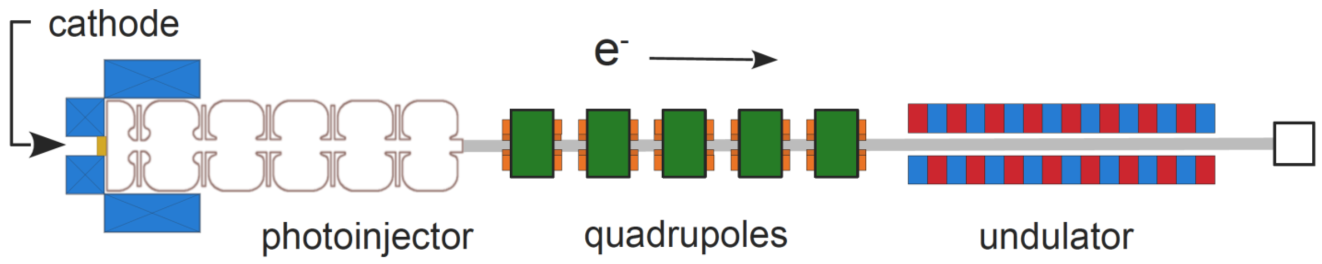
(image: lcls.slac.standford.edu)

Would be nice to have a tool that can quickly give suggested settings for a given photon beam request, is valid globally, and can adapt to changes over time

[1] J.-P. Colletier, et al.,"De novo phasing with X-ray laser reveals mosquito larvicide BinAB structure," Nature , vol. 539, pp. 43–47, Sep. 2016.
[2] I. D. Young, et al., "Structure of photosystem II and substrate binding at room temperature," Nature , vol. 540, pp. 453–457, Nov. 2016.
[3] M. P. Jiang, et al., "The origin of incipient ferroelectricity in lead telluride," Nature Communications, vol. 7, no. 12291, Jul. 2016.

Starting Smaller: A Case Study

Compact, THz FEL design based on previously operational TEU-FEL



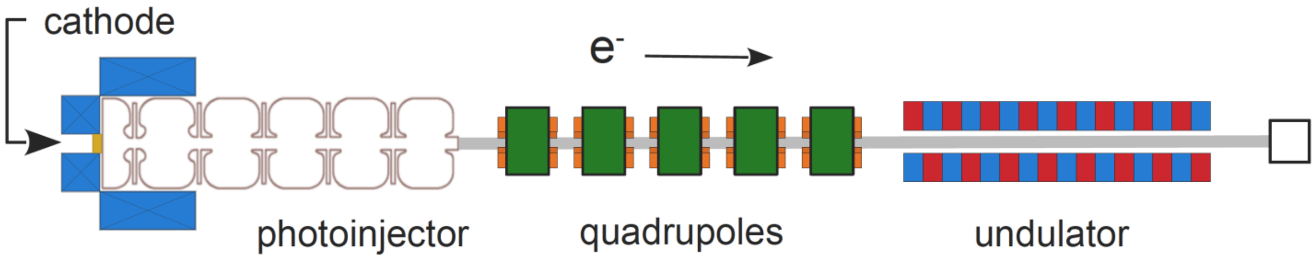
3 – 6 MeV electron beam
200 – 800 μm photon beam

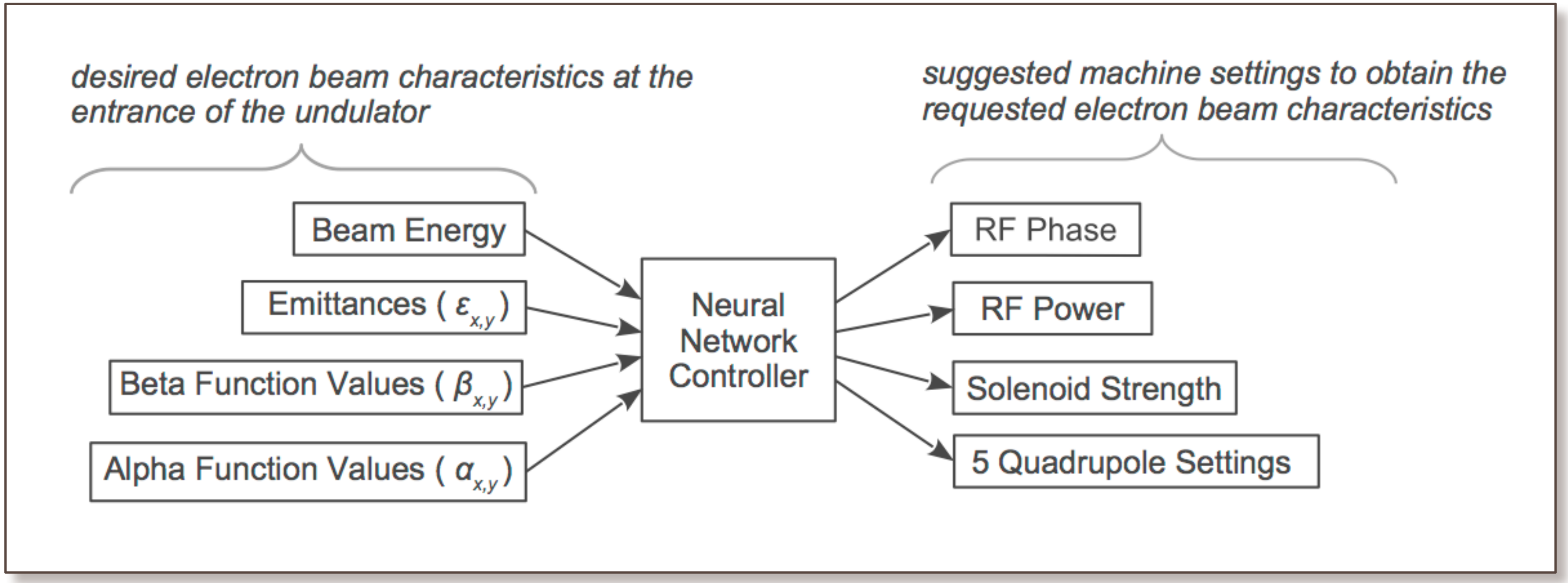
Previously operated at University of Twente in the Netherlands

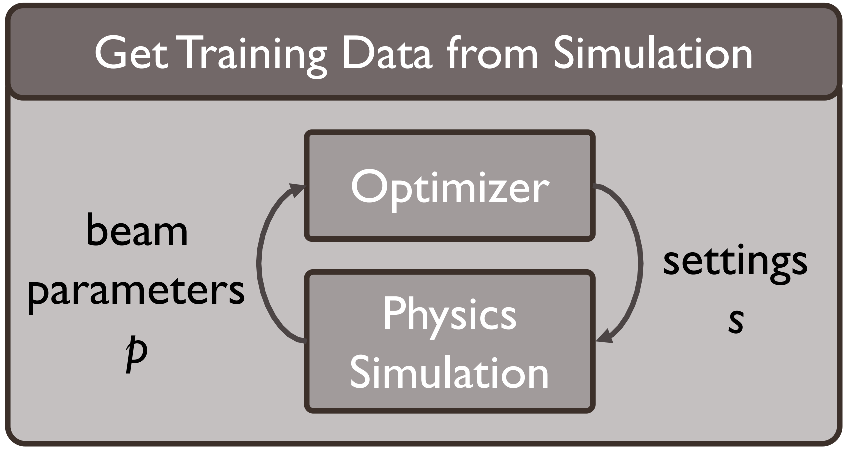
Was going to be re-built at CSU

This is an appealing system for an initial study because it has a small number of machine components, yet it exhibits non-trivial beam dynamics.

Intermediate goal: get the right beam parameters at the undulator entrance



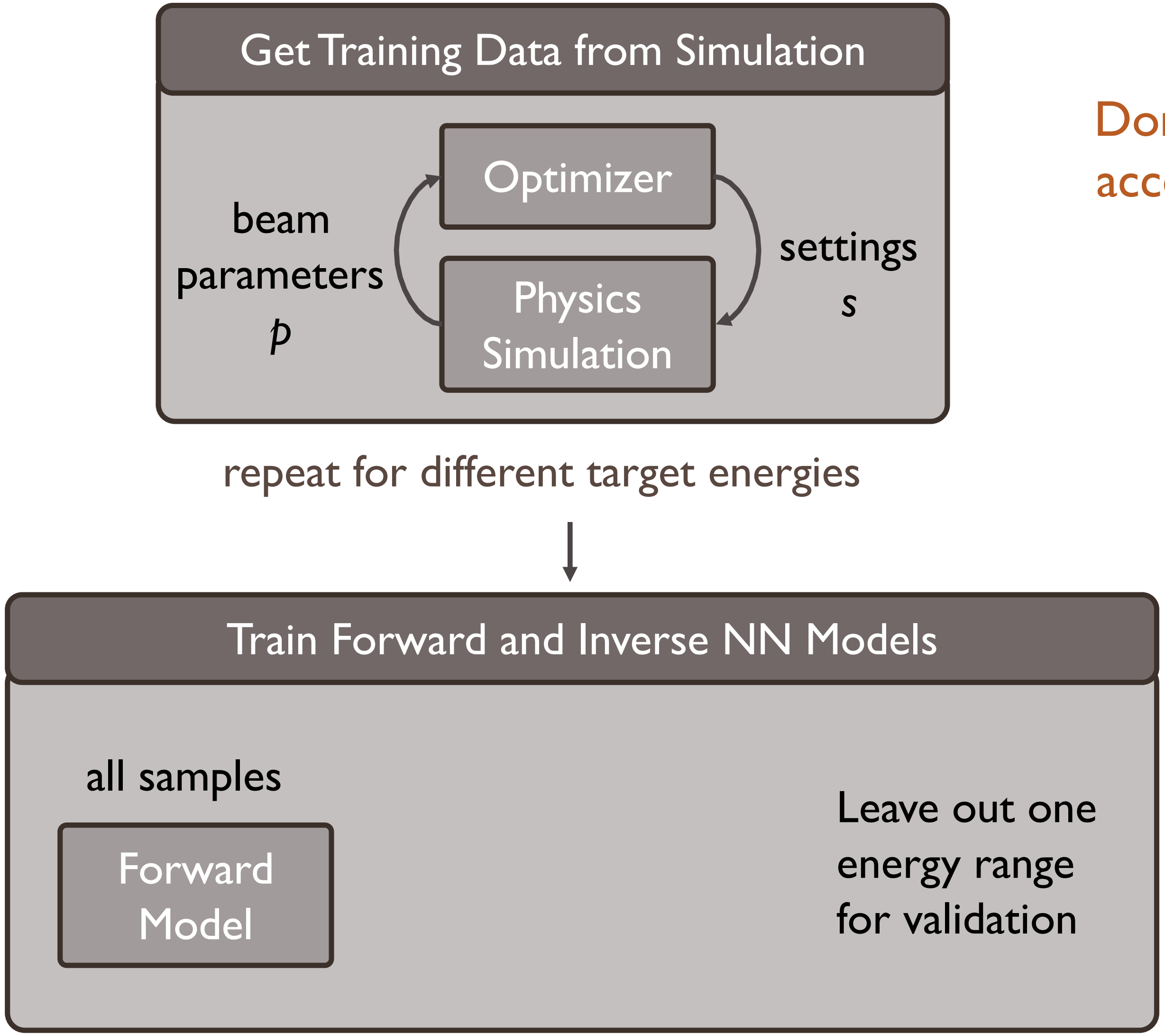




repeat for different target energies

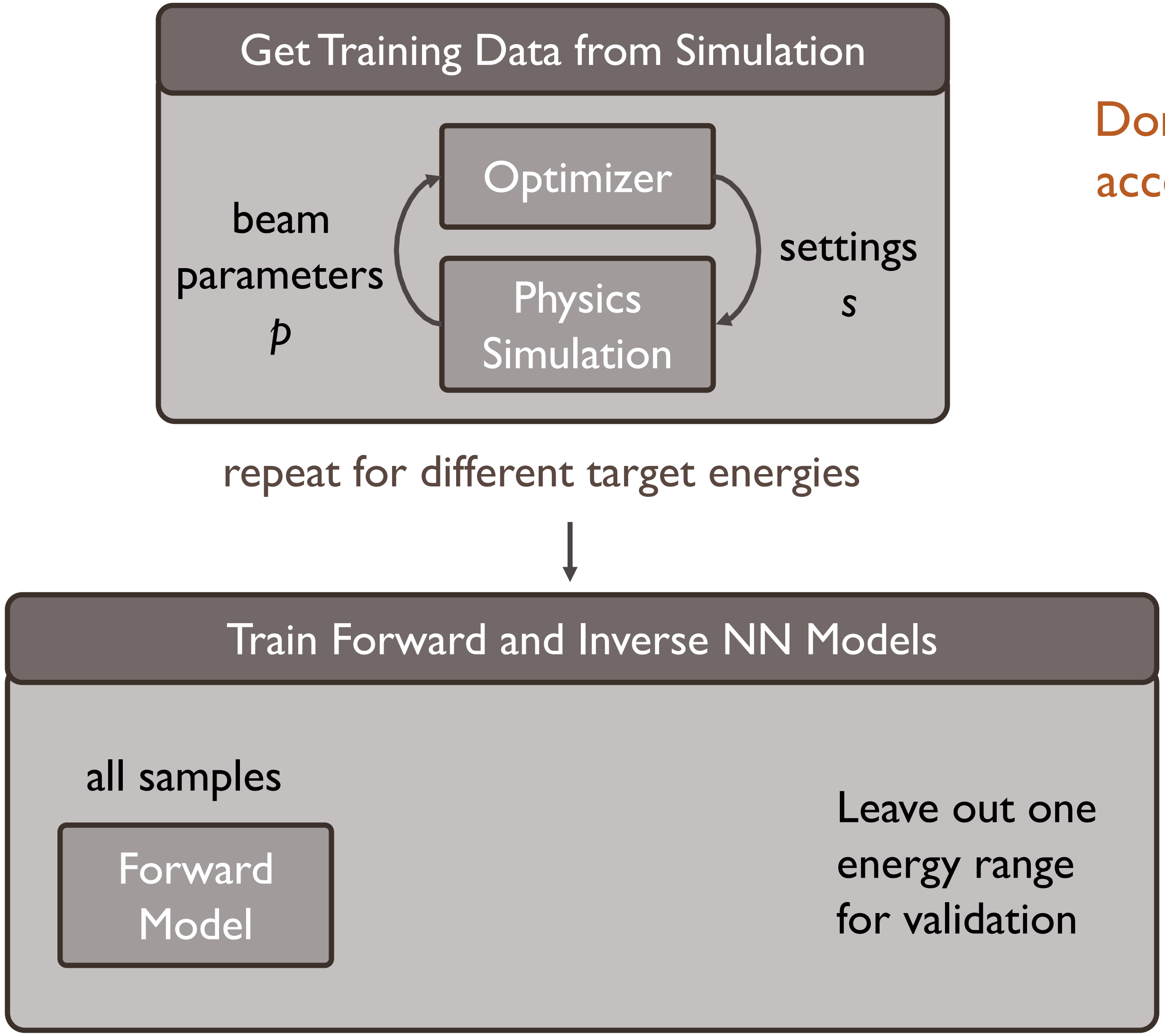
Don't always have a good physics-based model for particle accelerators, so what's in the data archive of a real facility?

Noisy data + tuning around roughly optimal settings



Don't always have a good physics-based model for particle accelerators, so what's in the data archive of a real facility?

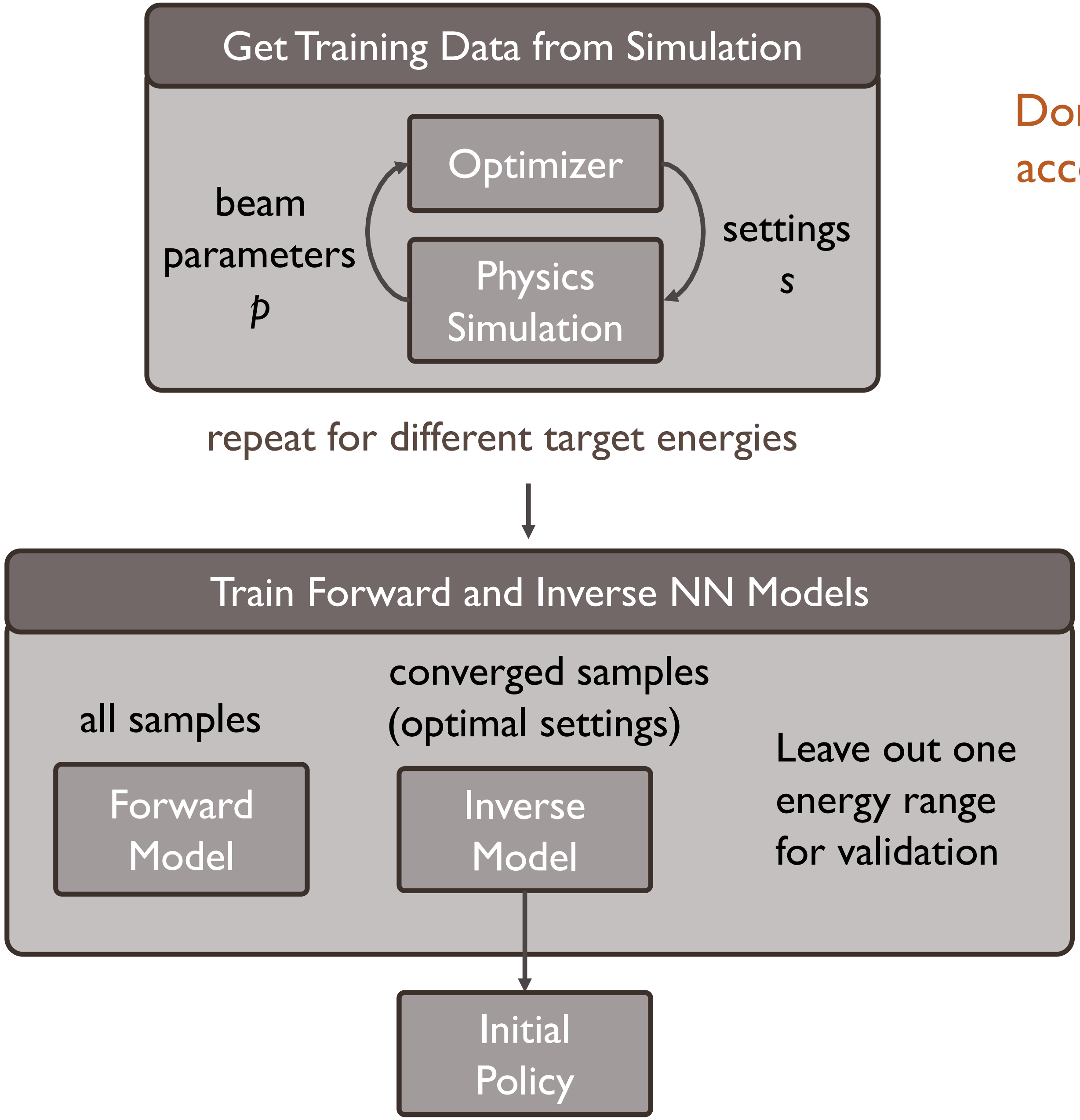
Noisy data + tuning around roughly optimal settings



Don't always have a good physics-based model for particle accelerators, so what's in the data archive of a real facility?

Noisy data + tuning around roughly optimal settings

Want to use the existing data to initialize control policy



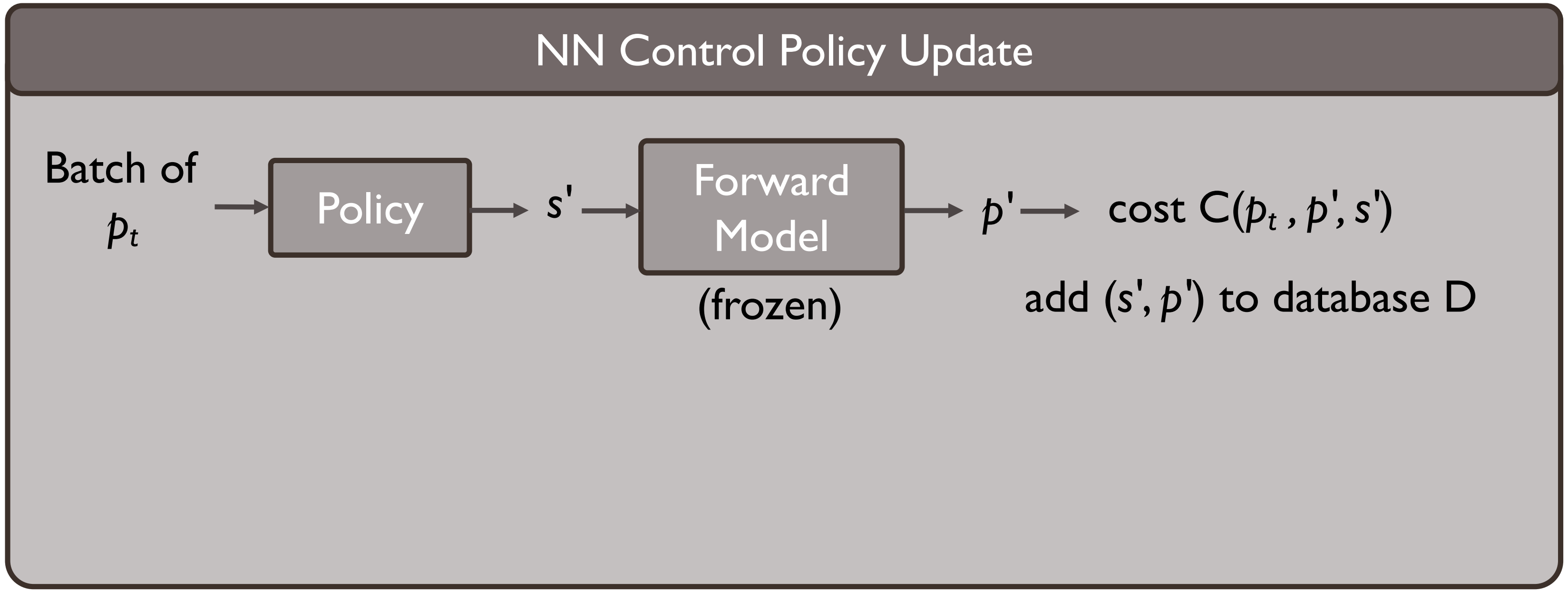
Don't always have a good physics-based model for particle accelerators, so what's in the data archive of a real facility?

Noisy data + tuning around roughly optimal settings

Want to use the existing data to initialize control policy → *model not invertible, but can pre-train policy with converged settings*

Training the Control Policy

- *First: just want to switch to roughly correct settings*
- *Then, two options: efficient local tuning algorithms we already use, or online model/controller updating*



p_t -- target beam parameters

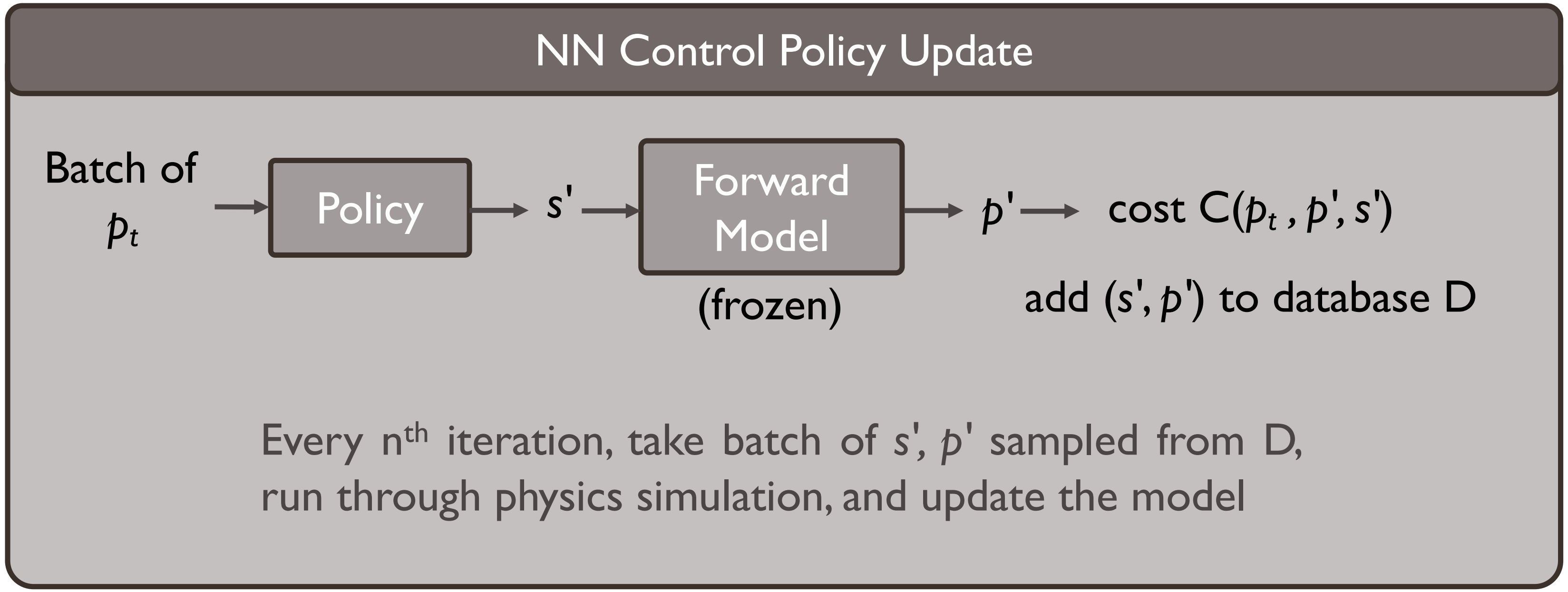
s' -- predicted optimal settings

p' – predicted beam parameters

Cost:
difference between p' and p_t
penalize loss of transmission
penalize higher magnet settings

Training the Control Policy

- *First: just want to switch to roughly correct settings*
- *Then, two options: efficient local tuning algorithms we already use, or online model/controller updating*



p_t -- target beam parameters

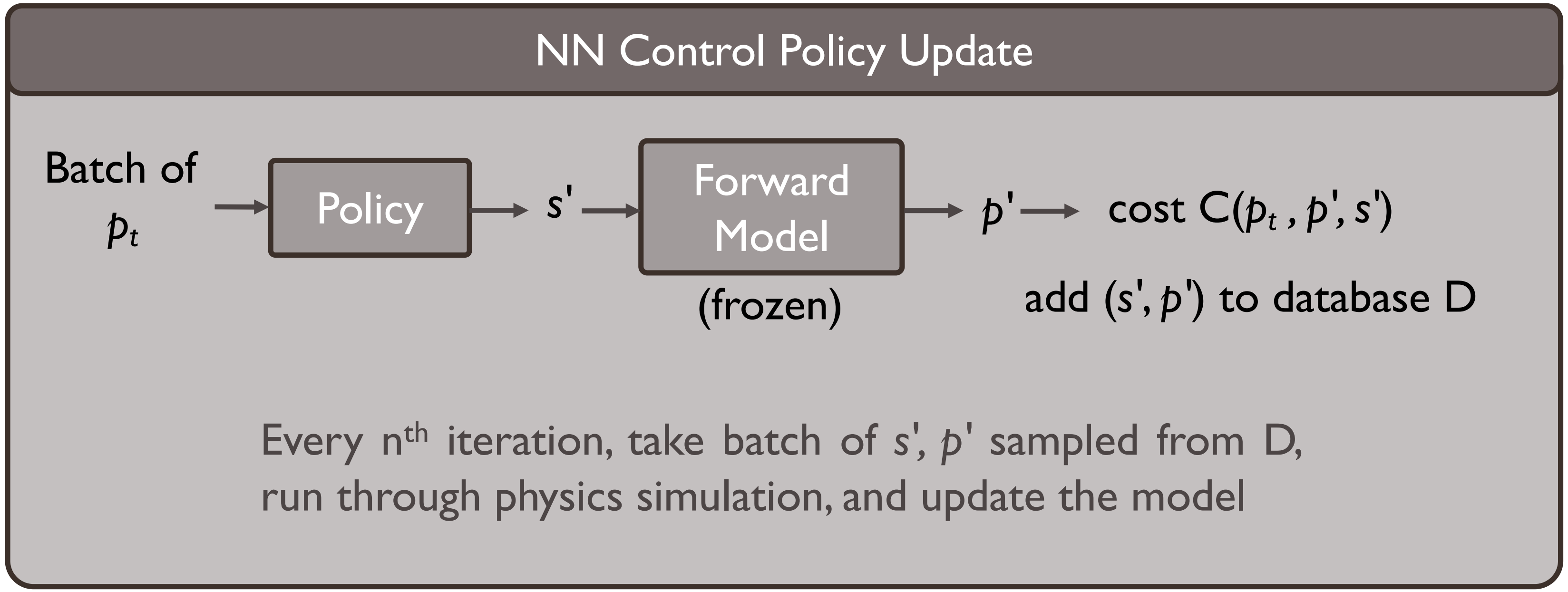
s' -- predicted optimal settings

p' – predicted beam parameters

Cost:
difference between p' and p_t
penalize loss of transmission
penalize higher magnet settings

Training the Control Policy

- *First: just want to switch to roughly correct settings*
- *Then, two options: efficient local tuning algorithms we already use, or online model/controller updating*



p_t -- target beam parameters

s' -- predicted optimal settings

p' – predicted beam parameters

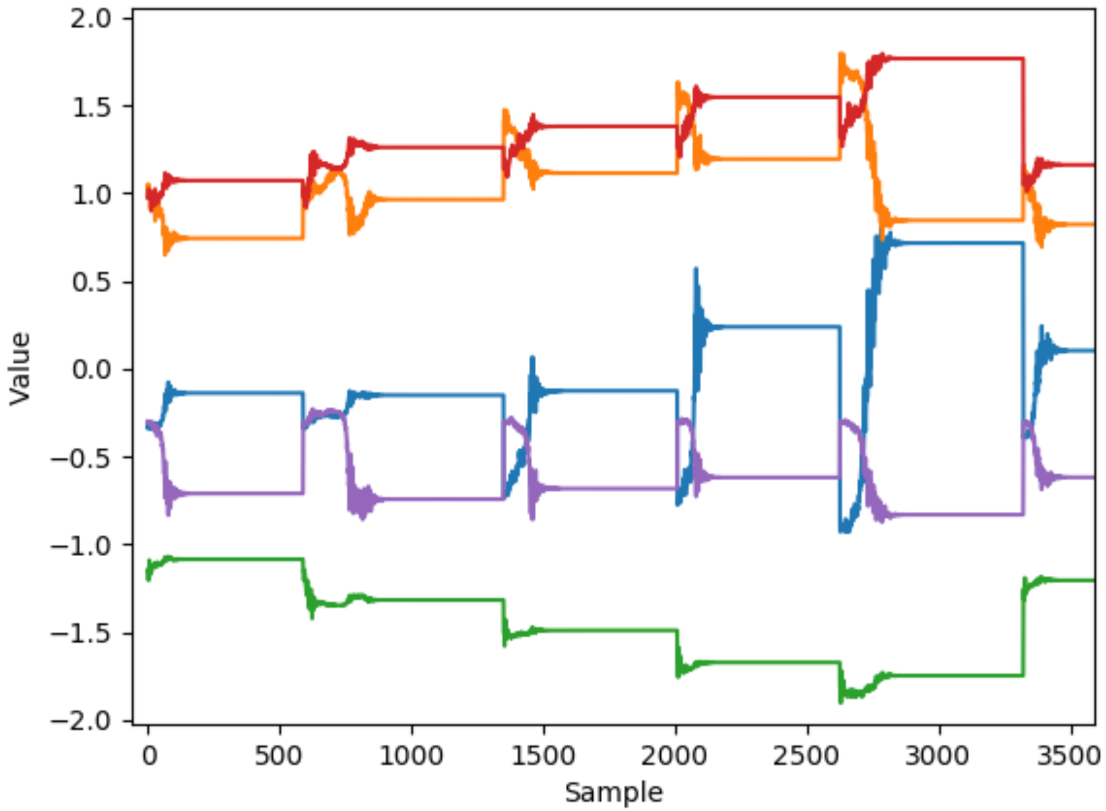
Cost:
difference between p' and p_t
penalize loss of transmission
penalize higher magnet settings

Then test policy directly on simulation

Initial Model and Controller

Training data from simulation:

- output from each iteration of Nelder-Mead, L-BFGS
- 12 beam energies between 3.1 – 6.2 MeV (7195 samples)



Example of what the training data looks like (quadrupoles shown in this case)

Model: 50-50-30-30 tanh nodes in hidden layers

- 8 inputs *(rf power, rf phase, sol. strength, quads)*
- 8 outputs $(\alpha_x, \alpha_y, \beta_x, \beta_y, \varepsilon_x, \varepsilon_y, E, N_p)$
- 5.7-MeV run used for validation set

First study: focus on target α, β for a given energy

Policy: 30-30-20-20 tanh nodes in hidden layers

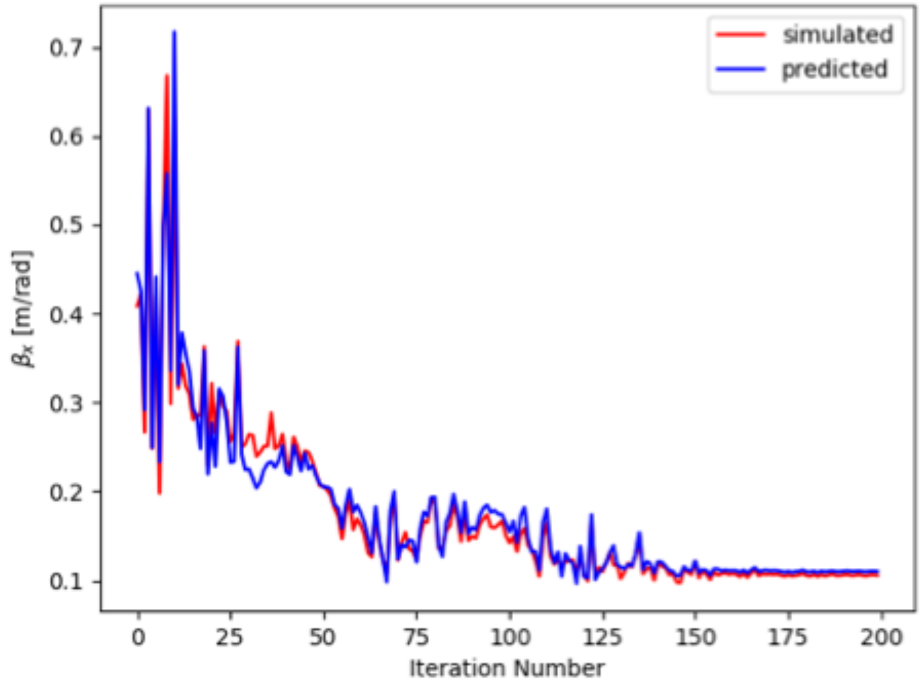
- inputs/outputs opposite the above (except N_p)
- random target energies, $\alpha_{xy} = 0, \beta_{xy} = 0.106$
- exclude 4.8 – 5.2 MeV range for validation

Initial Model and Controller Performance

Summary of Model Performance

Parameter	Train MAE	Train STD	Train Max	Val. MAE	Val. STD	Val. Max
α_x [rad]	0.018	0.042	0.590	0.067	0.091	0.482
α_y [rad]	0.022	0.037	0.845	0.070	0.079	0.345
β_x [m/rad]	0.004	0.009	0.287	0.008	0.012	0.130
β_y [m/rad]	0.005	0.011	0.357	0.012	0.017	0.189

Example of Model Performance

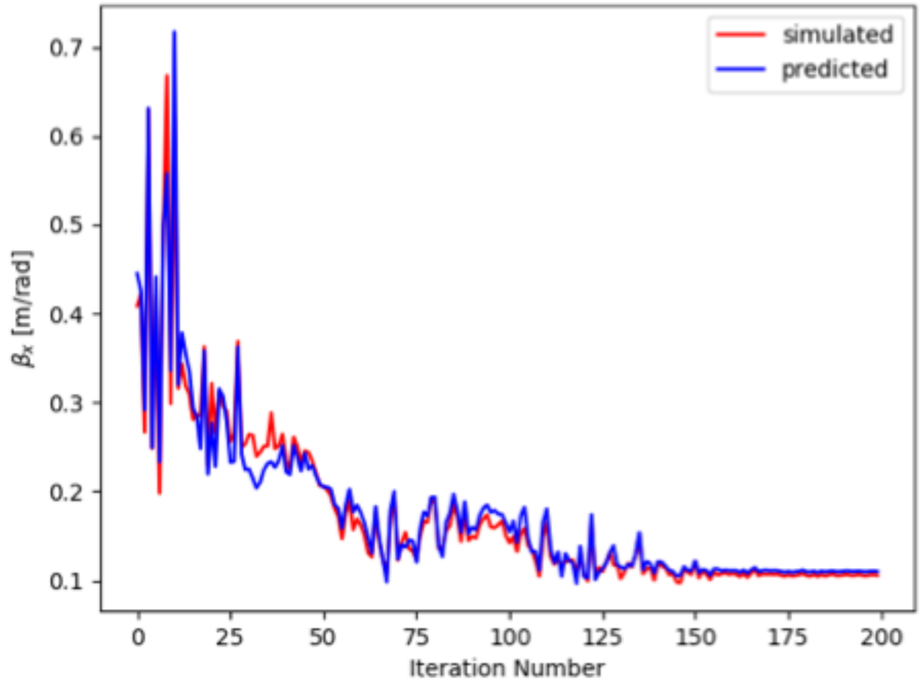


Initial Model and Controller Performance

Summary of Model Performance

Parameter	Train MAE	Train STD	Train Max	Val. MAE	Val. STD	Val. Max
α_x [rad]	0.018	0.042	0.590	0.067	0.091	0.482
α_y [rad]	0.022	0.037	0.845	0.070	0.079	0.345
β_x [m/rad]	0.004	0.009	0.287	0.008	0.012	0.130
β_y [m/rad]	0.005	0.011	0.357	0.012	0.017	0.189

Example of Model Performance



Controller ability to reach $\alpha_{x,y}$ = 0 and $\beta_{x,y}$ = 0.106 in **one iteration**

Parameter	Train MAE	Train STD	Train Max	Val. MAE	Val. STD	Val. Max
α_x [rad]	0.012	0.075	0.011	0.046	0.063	0.141
α_y [rad]	0.013	0.079	0.012	0.045	0.064	0.140
β_x [m/rad]	0.008	0.004	0.006	0.006	0.023	0.008
β_y [m/rad]	0.014	0.011	0.011	0.011	0.069	0.038

What this means: for a given energy, the controller will immediately reach the desired beam size to within about 10% and the beam will be close to a waist, requiring minimal further tuning (assuming no substantial drift…)

A.L. Edelen et al., FEL '17

Dealing with "Long-Term" Time Dependencies: Resonant Frequency Control in Normal Conducting Cavities

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

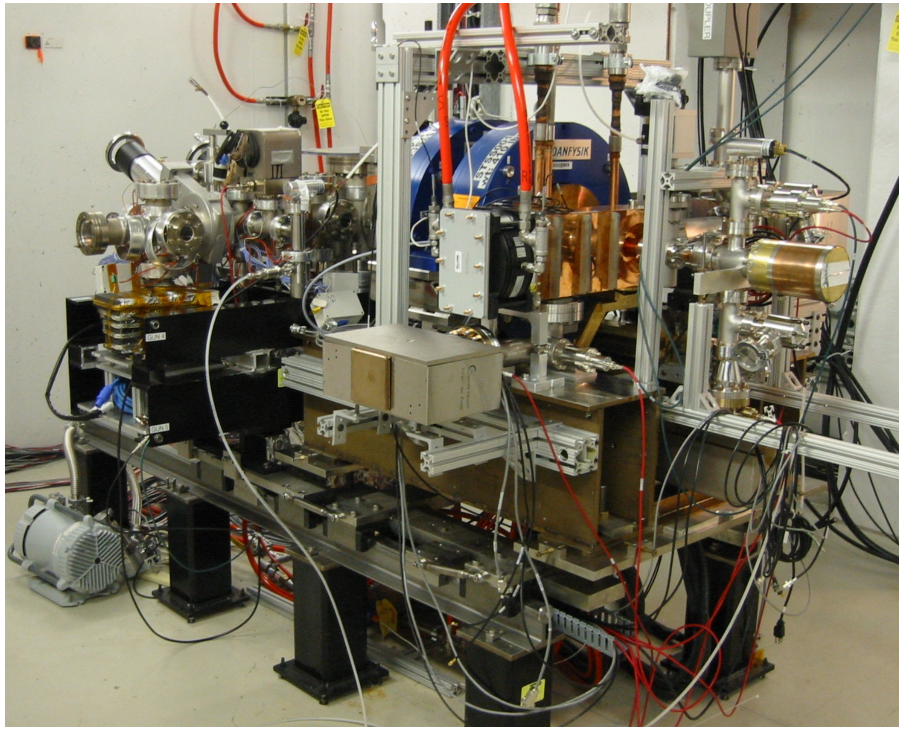


Photo: P. Stabile

Radio frequency quadrupole (RFQ) for the PIP-II Injector Test

Photo: J. Steimel

"long term" in this case means responses lasting many minutes (e.g. 30), with control actions at 0.5 Hz and 1 Hz

Temperature Control for the RF Photoinjector at FAST

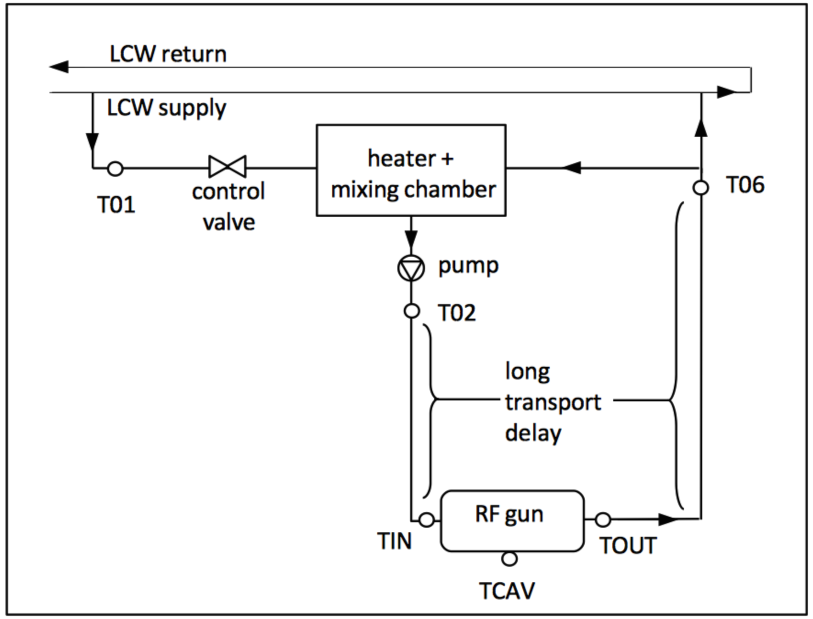
Resonant frequency controlled via temperature

PID control is undesirable in this case:

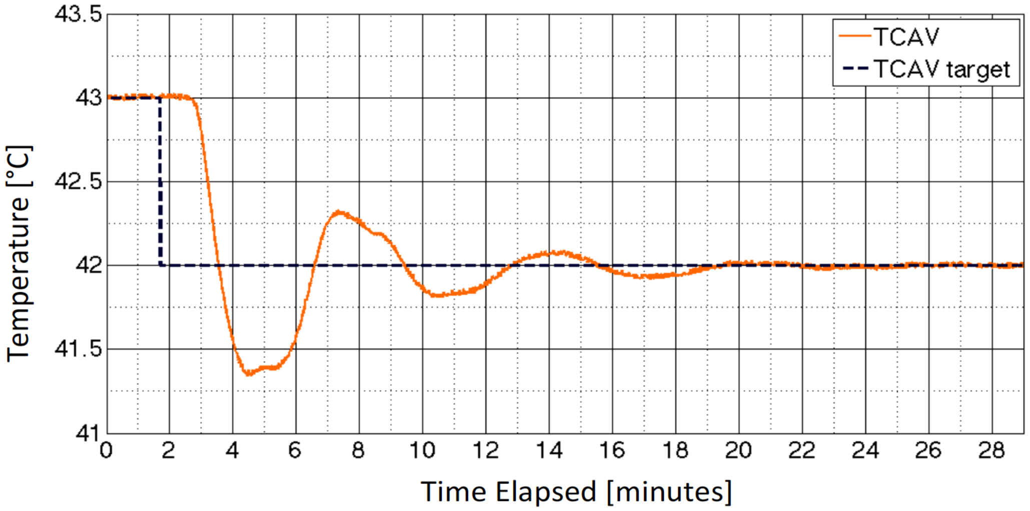
- Long transport delays and thermal responses
- Recirculation leads to secondary impact of disturbances
- Two controllable variables: heater power + valve aperture

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

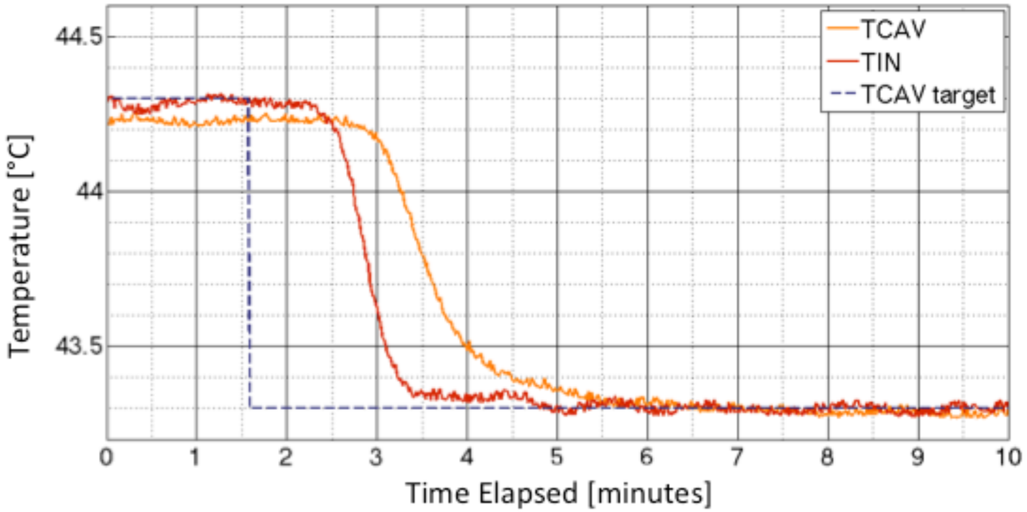
Gun Water System Layout



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

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

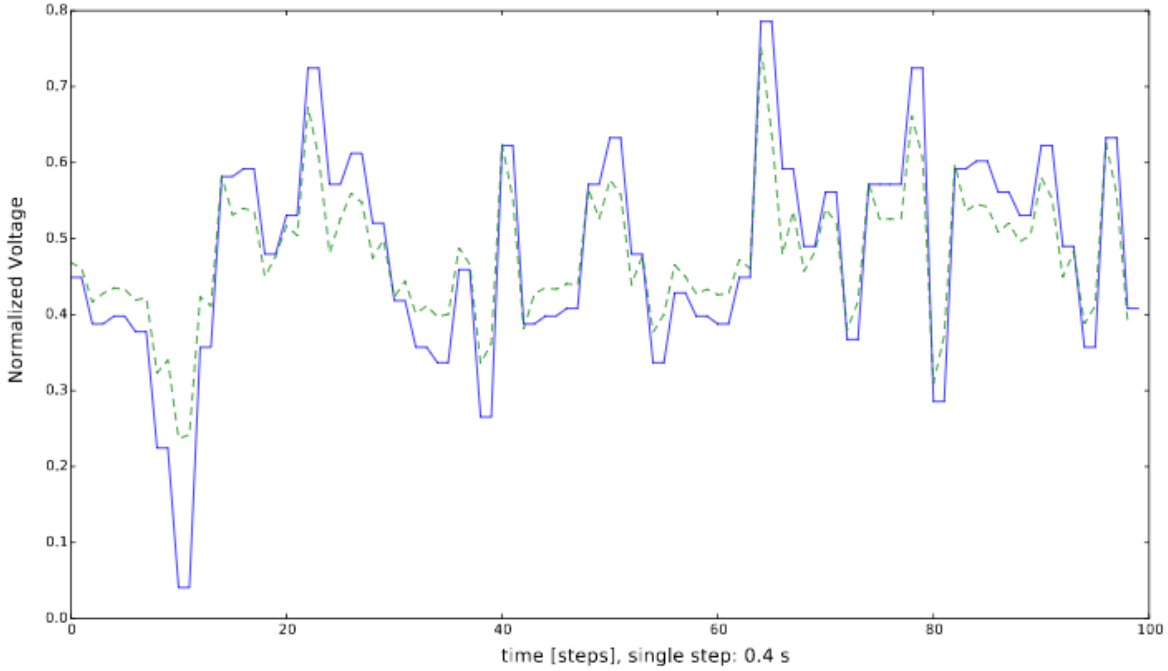
"Some of the most dangerous malfunctions of the magnets are quenches which occur when a part of the superconducting cable becomes normally-conducting."

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



Some Practical Challenges

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

*e.g. not too many unaccounted for inputs or hardware changes, etc.

Need a sufficient amount of reliable* data*

Training on Measured Data

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

Relevant-but-unlogged variables

Availability of diagnostics

Observed parameter range in archived data

Time on machine for characterization studies (schedule + expense)

Ideal case:
- *comprehensive, high-resolution data archive (e.g. including things like ambient temp./pressure)*
- *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

- Neural networks are very flexible tools → *far more powerful + accessible in recent years*
- Lots of opportunities to use neural networks (and ML more broadly) to improve accelerator performance on both existing and future machines
- Transferrable between machines to some degree → *lots of potential for fruitful collaborations!*
- But, not a panacea!
  - *Simpler model-independent online optimization + simpler model-based approaches in many cases may be more appropriate*
  - *Boundaries of usefulness/reliability and tradeoff with time investment have yet to be determined rigorously*

Final Notes

- Neural networks are very flexible tools → *far more powerful + accessible in recent years*
- Lots of opportunities to use neural networks (and ML more broadly) to improve accelerator performance on both existing and future machines
- Transferrable between machines to some degree → *lots of potential for fruitful collaborations!*
- But, not a panacea!
  - *Simpler model-independent online optimization + simpler model-based approaches in many cases may be more appropriate*
  - *Boundaries of usefulness/reliability and tradeoff with time investment have yet to be determined rigorously*
- Growing community → *two very recent workshops on ML for accelerators*

Machine Learning for Particle Accelerators
27 February – 2 March at SLAC
Agenda/Talks: https://tinyurl.com/y988njbl

Intelligent Controls for Particle Accelerators
30 – 31 January at Daresbury Lab
Agenda/Talks: https://tinyurl.com/y9rg3uht

Final Notes

- Neural networks are very flexible tools → *far more powerful + accessible in recent years*
- Lots of opportunities to use neural networks (and ML more broadly) to improve accelerator performance on both existing and future machines
- Transferrable between machines to some degree → *lots of potential for fruitful collaborations!*
- But, not a panacea!
  - *Simpler model-independent online optimization + simpler model-based approaches in many cases may be more appropriate*
  - *Boundaries of usefulness/reliability and tradeoff with time investment have yet to be determined rigorously*
- Growing community → *two very recent workshops on ML for accelerators*

Thanks for your attention!

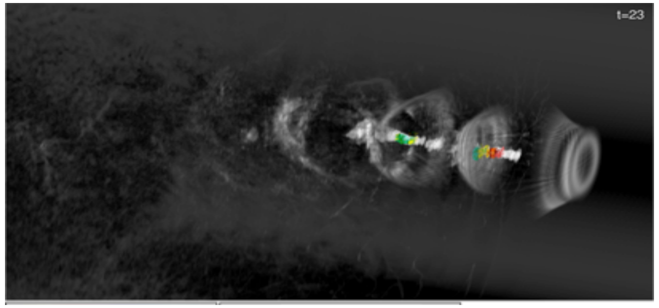
Thanks to many collaborators who contributed to the work shown:
Jonathan Edelen, Dean Edstrom Jr., Alex Halavanau, Nicole Neveu, Andreas Adelmann, Alex Scheinker, Daniel Bowring, Brian Chase, Denise Finstrom, Dennis Nicklaus, Jim Steimel, Elvin Harms, Sandra Biedron, Stephen Milton, Dave Douglas, Philippe Piot, Aleksandr Romanov, Jinhao Ruan, James Santucci, Chris Tennant, and many others

(along with Fermilab's HPC resources and support from Amitoj Singh, Alexei Strelchenko, Gerard Bernabeau, and Jim Simone)

Backup Slides

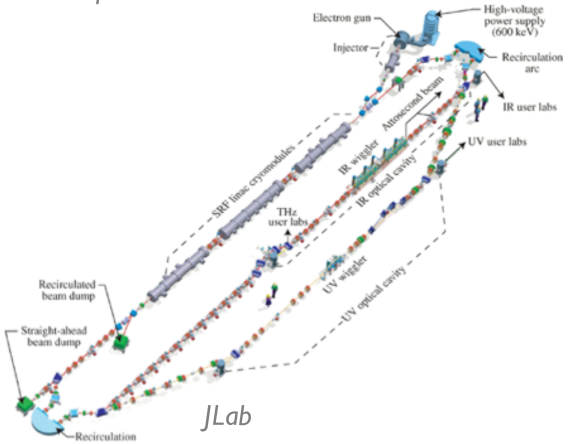
Interesting Technical Challenges

- 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

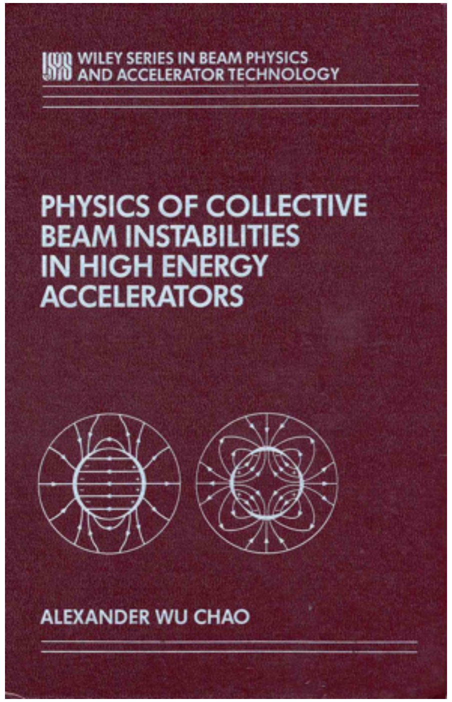
Fermilab



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*

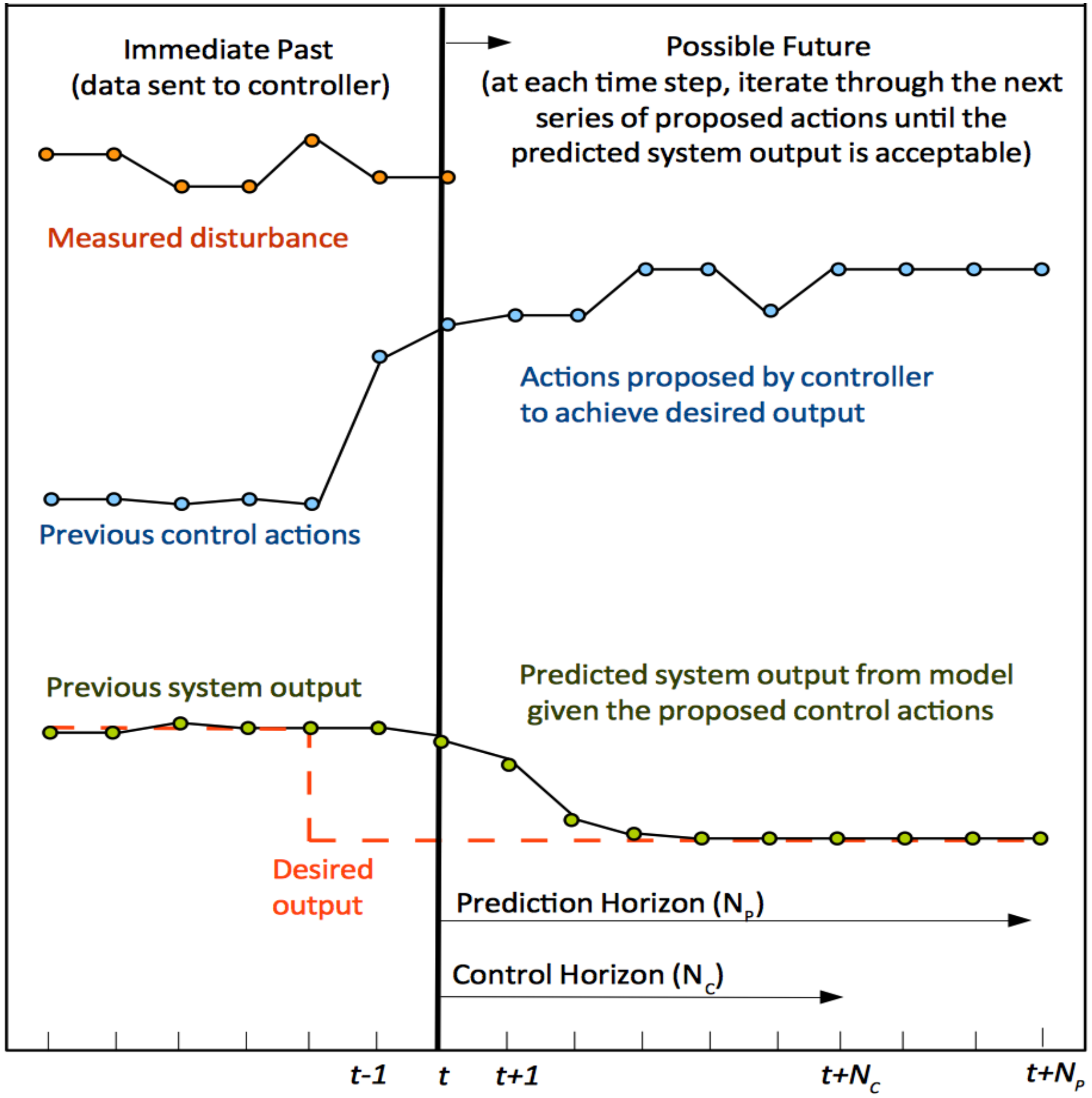


Uncertain, time-varying, nonlinear, many-parameter systems with continuous action spaces:

→ of great interest for research in control and machine learning

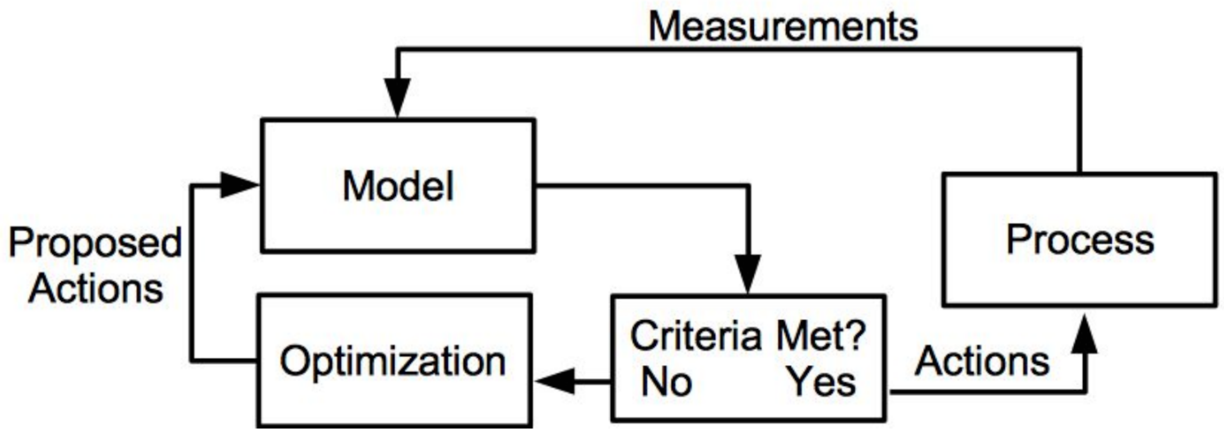
→ lots of opportunity to both gain from and contribute to this area

Model Predictive Control (Prediction + Planning)

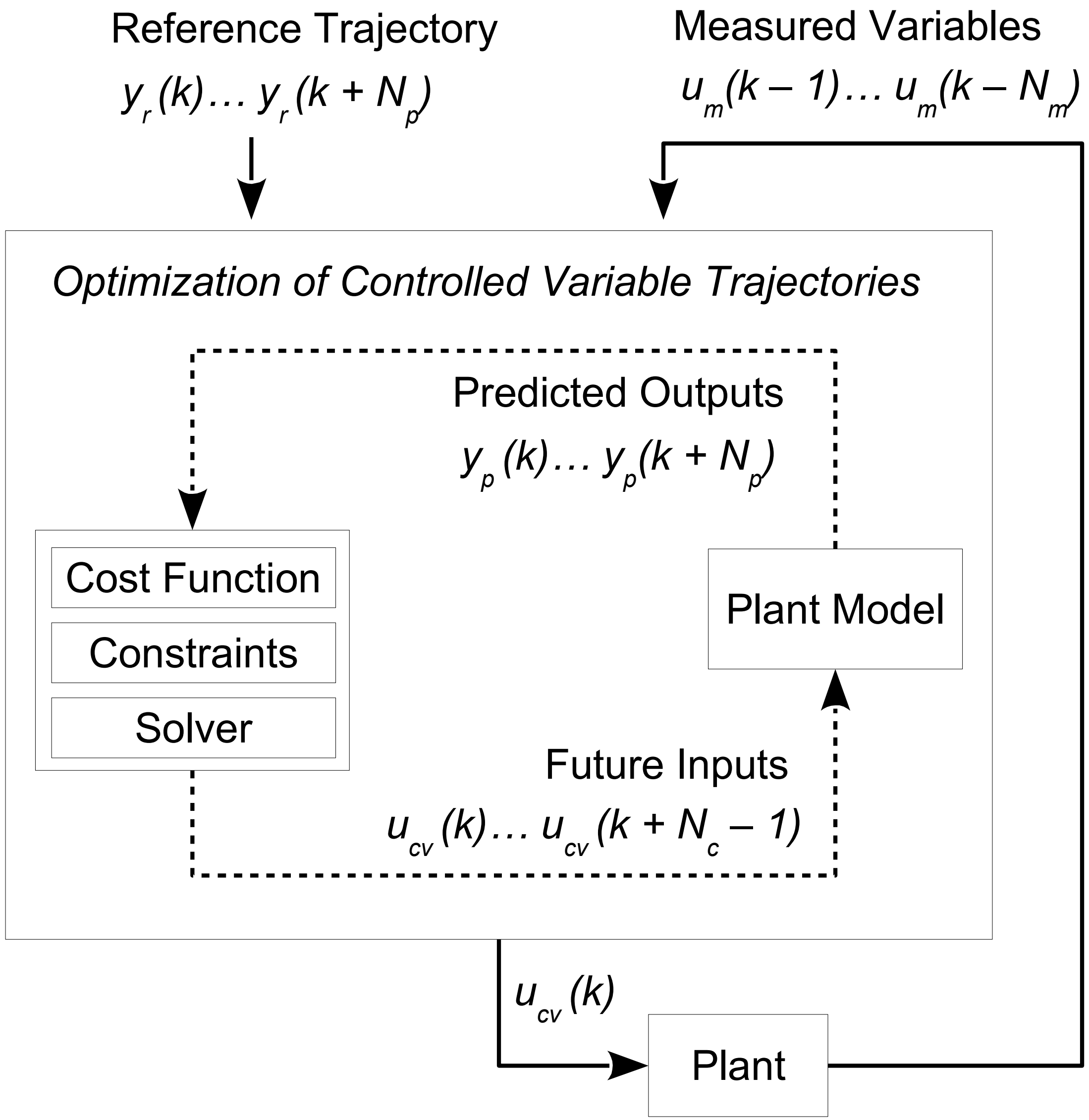


Basic concept:

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



Model Predictive Control (Prediction + Planning)



N_m previous measurements

N_p future time steps predicted

N_c future time steps controlled

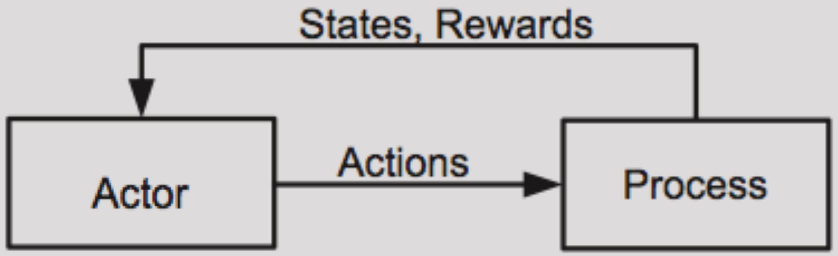
$$\sum_{i=1}^{N_p} \{w_y [y_r(k+i) - y_p(k+i)]\}^2$$
(output variable targets)

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

Neural Network Policies and Reinforcement Learning

Actor-only Methods

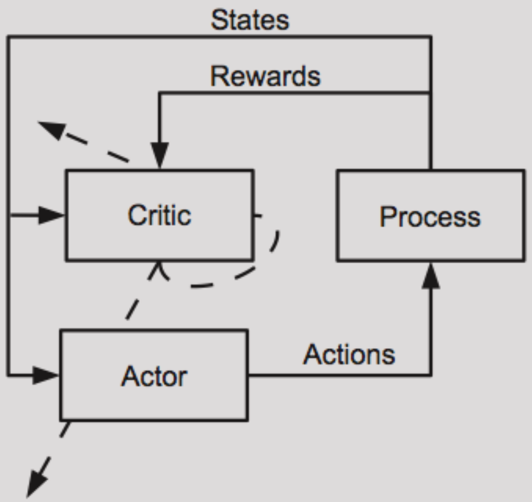


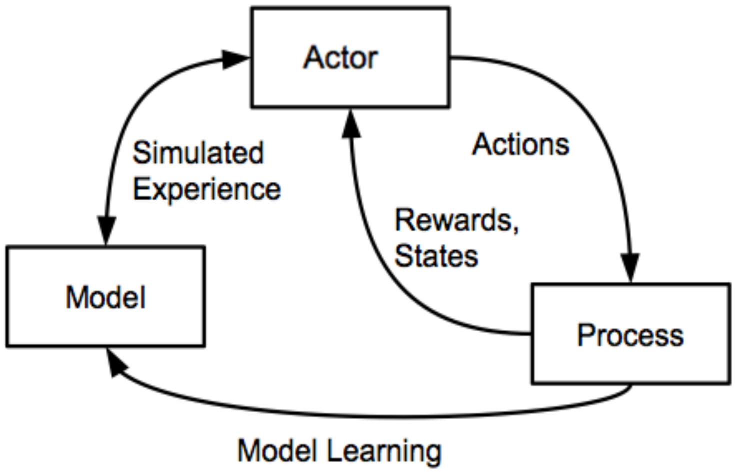
- Actor is a control policy
- Maps states to actions
- Reward provides training signal

Actor-Critic Methods

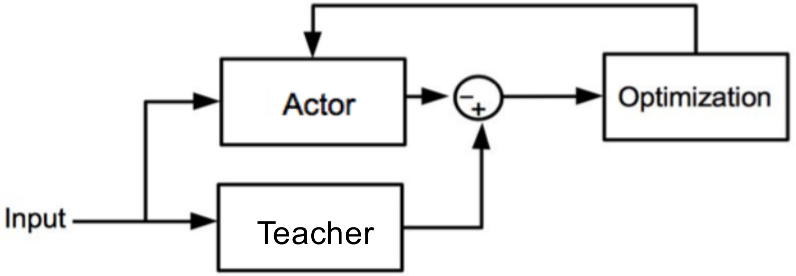
- Critic maps states or state/action pairs to an estimate of long-term reward
- Could be a NN, tabular, etc.
- Critic provides training signal to actor

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

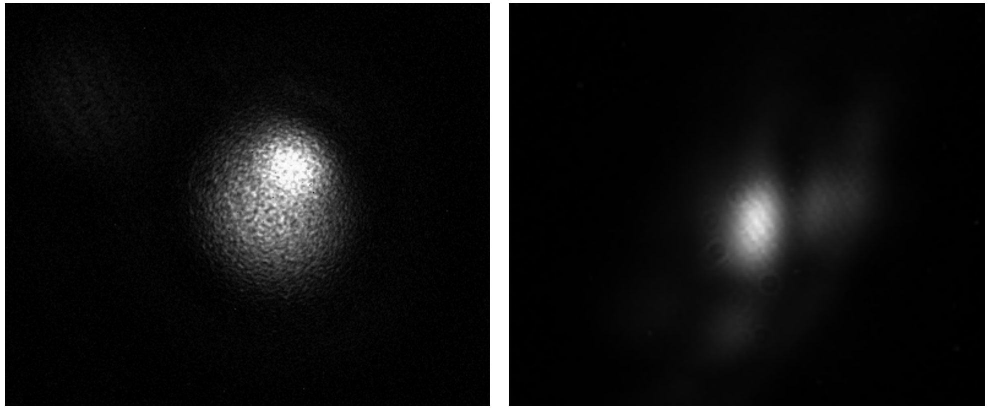
Initial Study: Predict Beam Characteristics After CC2

Motivation:

- Laser spot asymmetries → emittance asymmetries
- Gun phase and solenoid strength tuned daily

- Would like to:
  - Obtain optimal gun phase and solenoid strength for a given initial laser distribution automatically
  - Have an online prediction of emittance (destructive measurement)

  → *an accurate online model is useful for both of these*



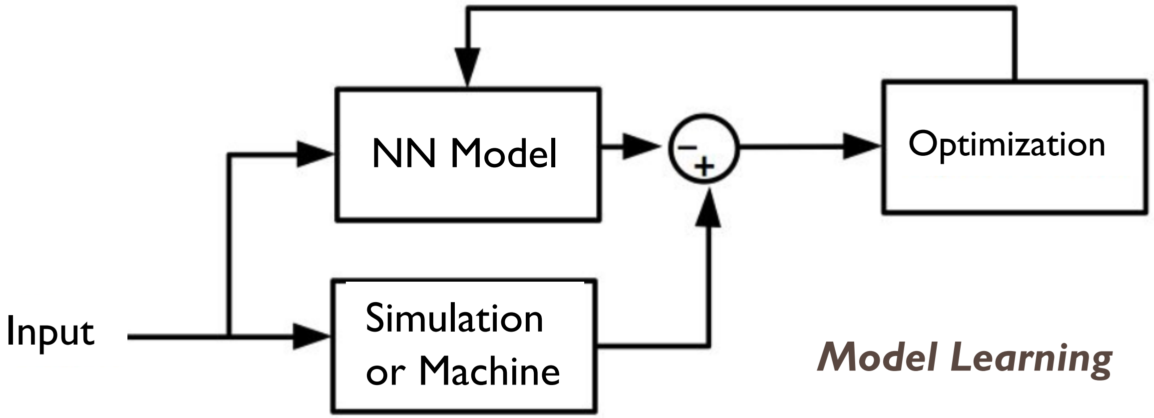
Example images of laser spot (10 Aug. 2016, 11 Nov. 2017)

Other perks:

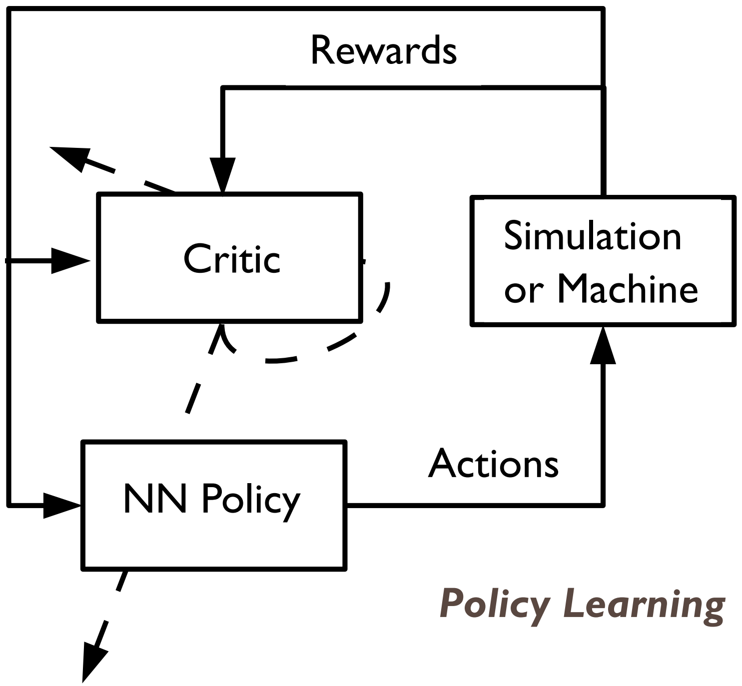
- Test creating a fast NN modeling tool from slower-executing simulations
- Test incorporating image-based diagnostics directly into model/control policy
- Natural extension to facilitate phase space manipulation (e.g. RTFB transform)

Initial Study Steps

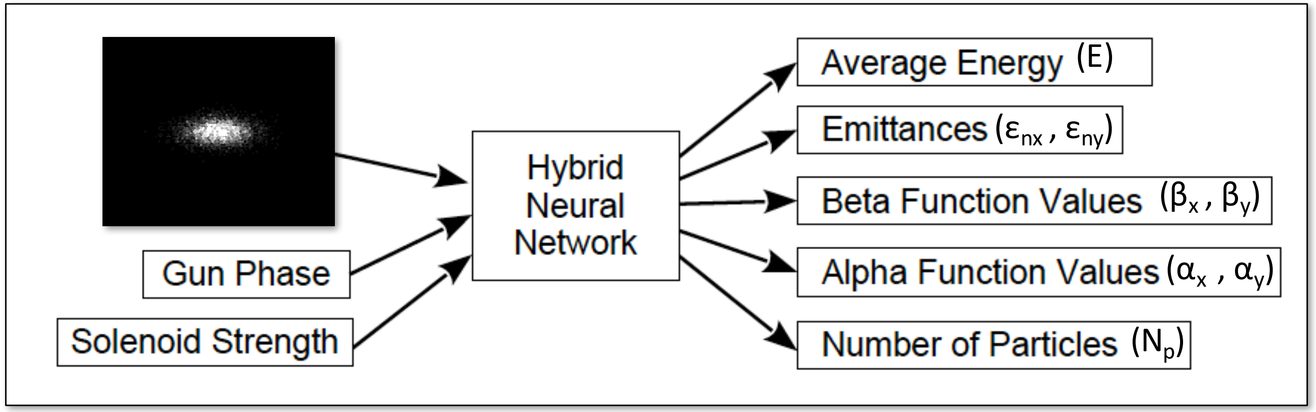
- Gather simulation data from PARMELA scans
- Create a NN model
- Gather measured data
  - Incorporate measured data into model
  - Validate simulation results
- Decide whether to expand the model inputs/outputs
- Train RL controller using updated model
- Carefully test model and controller on the machine



Model Learning



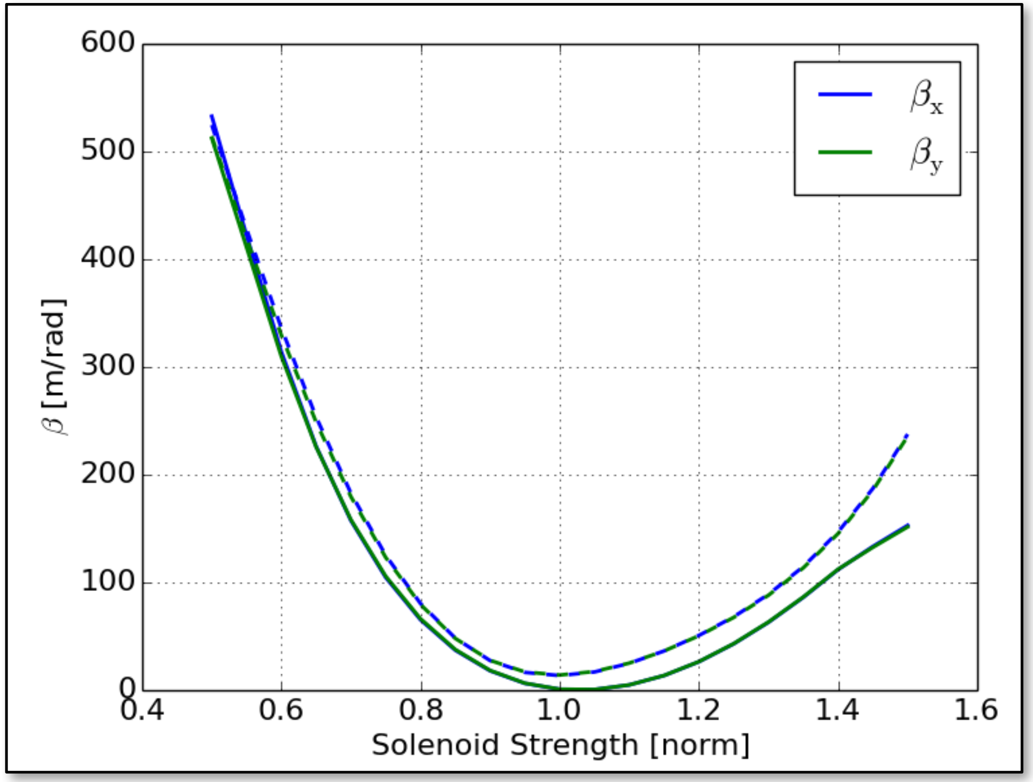
Policy Learning



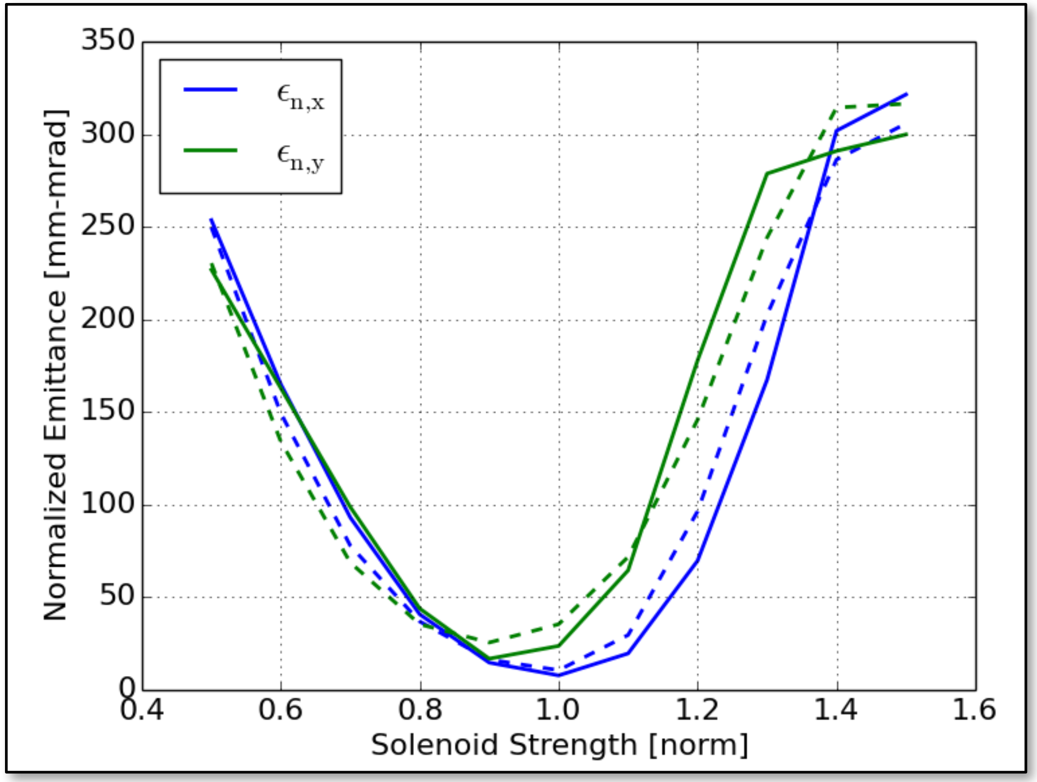
Initial Model Inputs and Outputs *(later expanded these)*

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

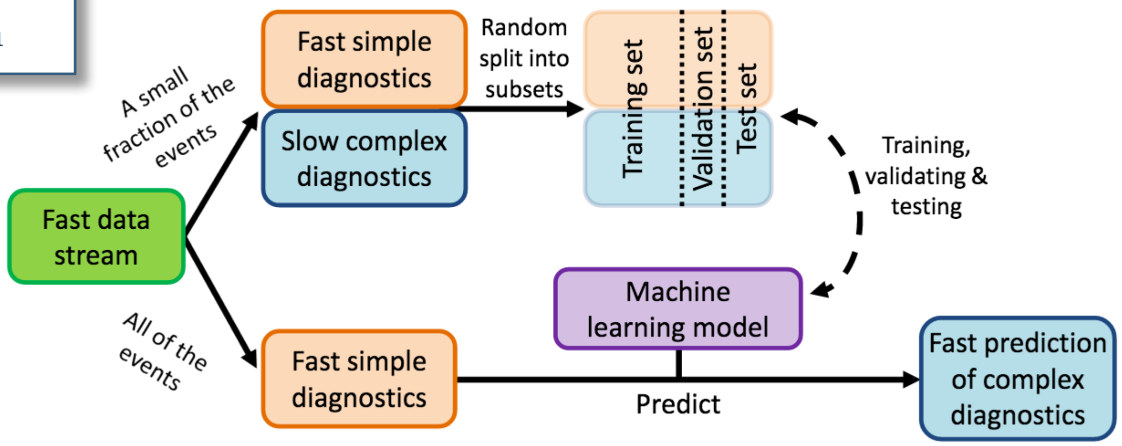
→ *Surprisingly good for such a small training set*

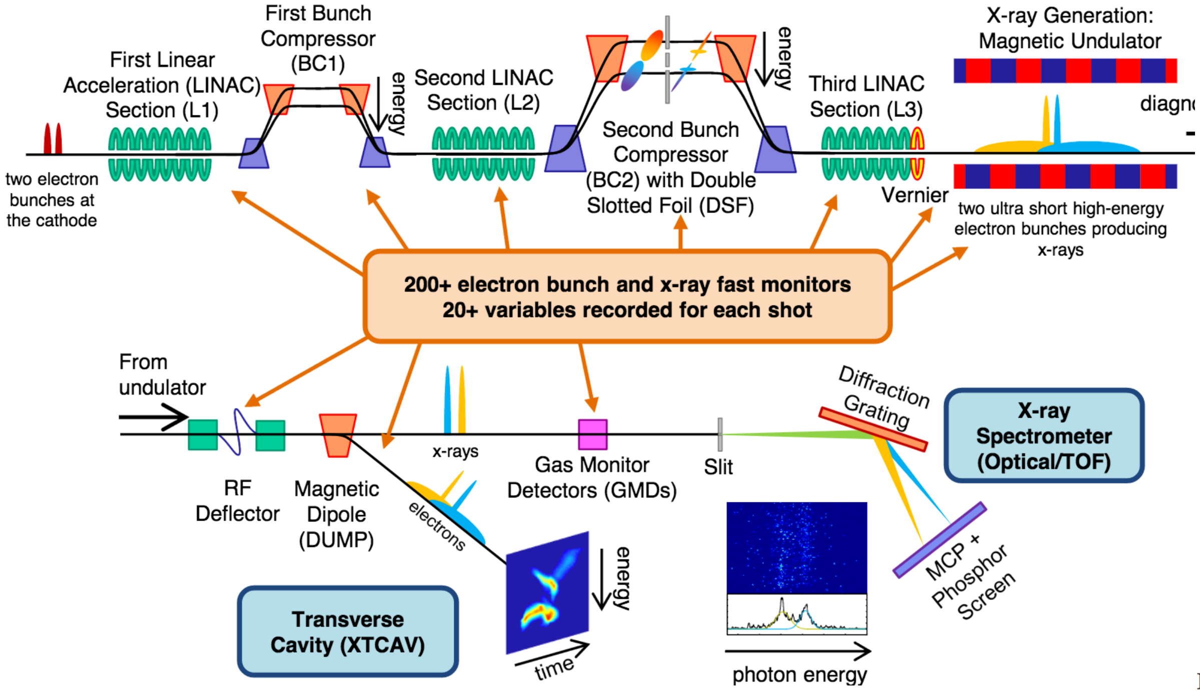
Machine learning applied to single-shot x-ray diagnostics in an XFEL

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





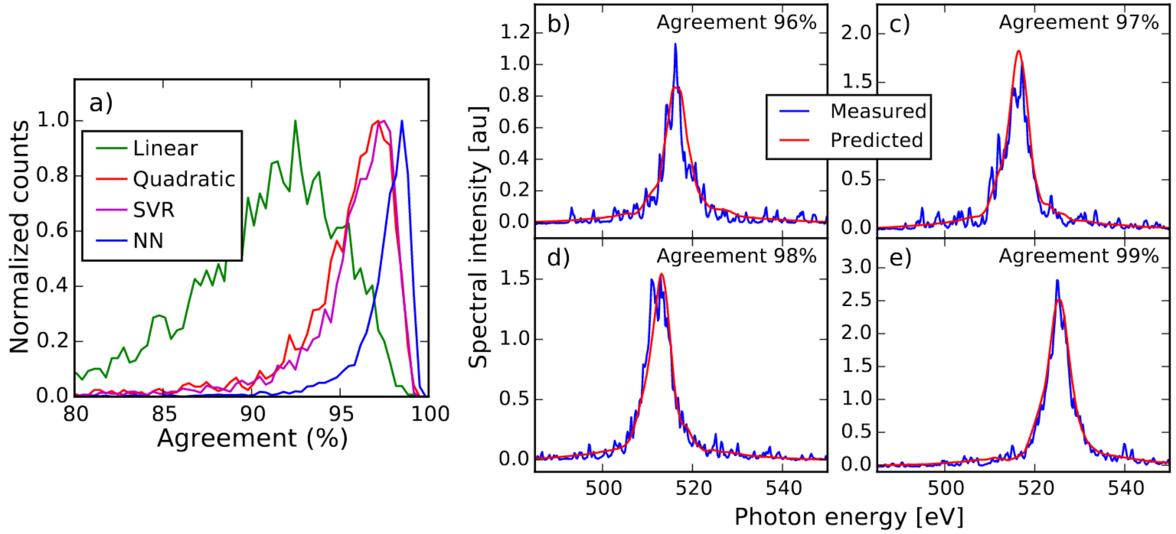


FIG. 4. Spectral shape prediction for a single pulse. (a) Histogram of agreements between the predicted and the measured spectra for the test set using the 4 different models. (b-e) Examples of the measured and the predicted spectra using a neural network to illustrate the accuracy for different agreement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

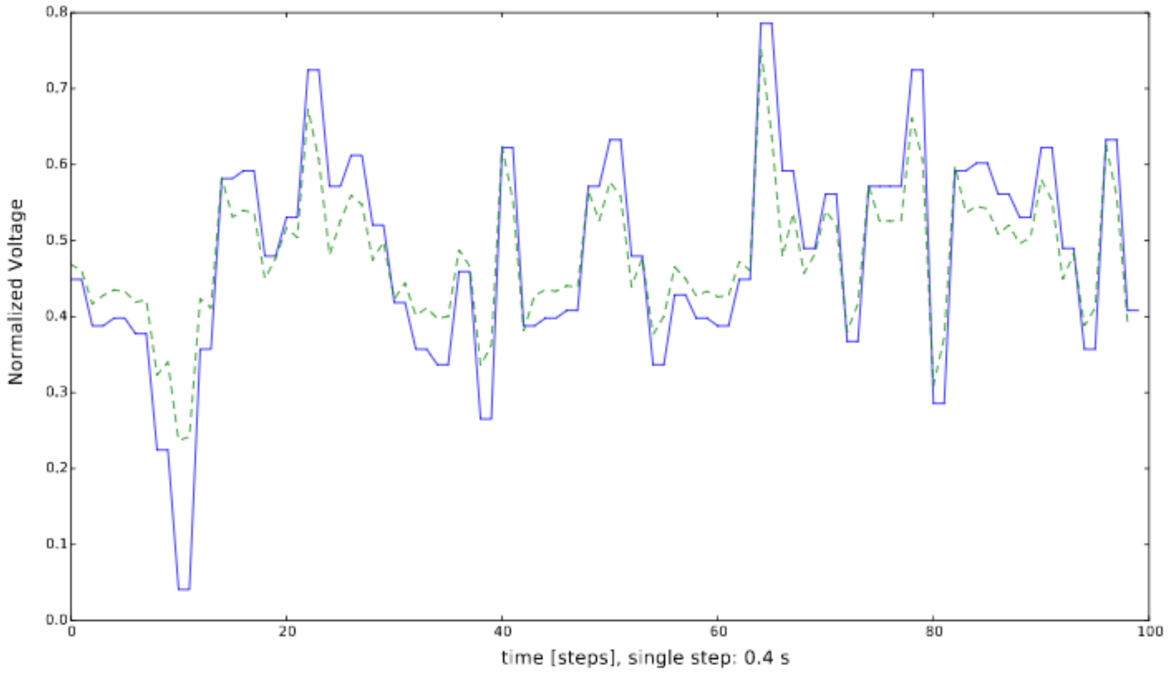
"Some of the most dangerous malfunctions of the magnets are quenches which occur when a part of the superconducting cable becomes normally-conducting."

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



PIP-II Injector Test RFQ

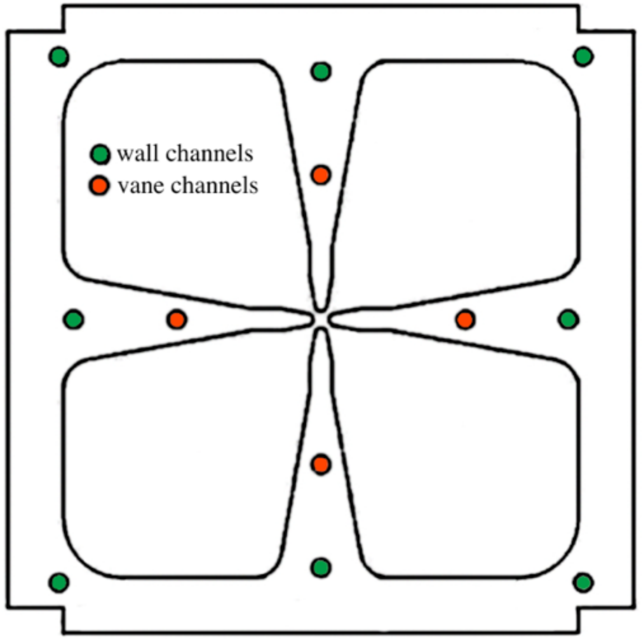
variable heating

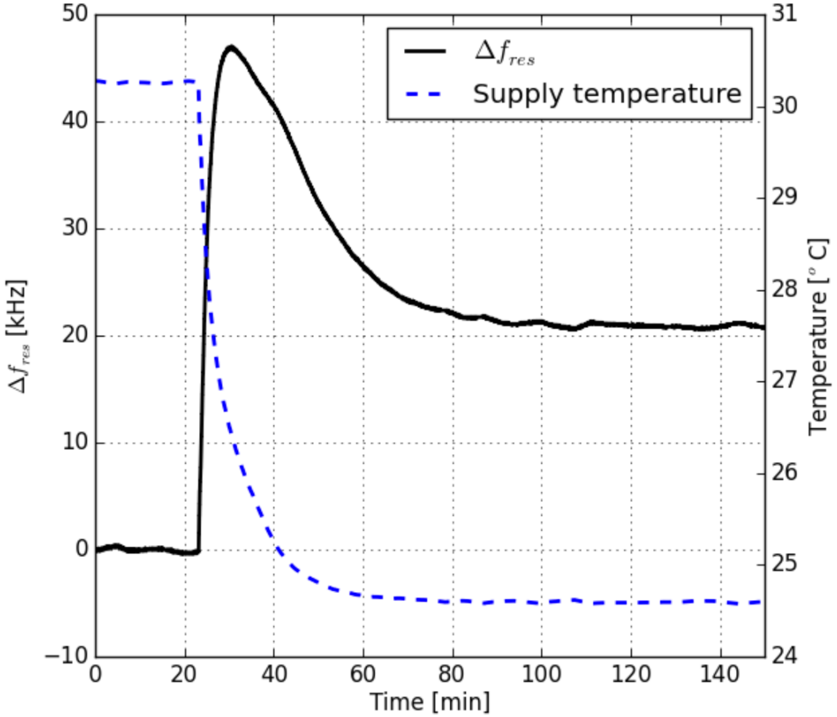
Specification for GDR: 3-kHz maximum frequency shift

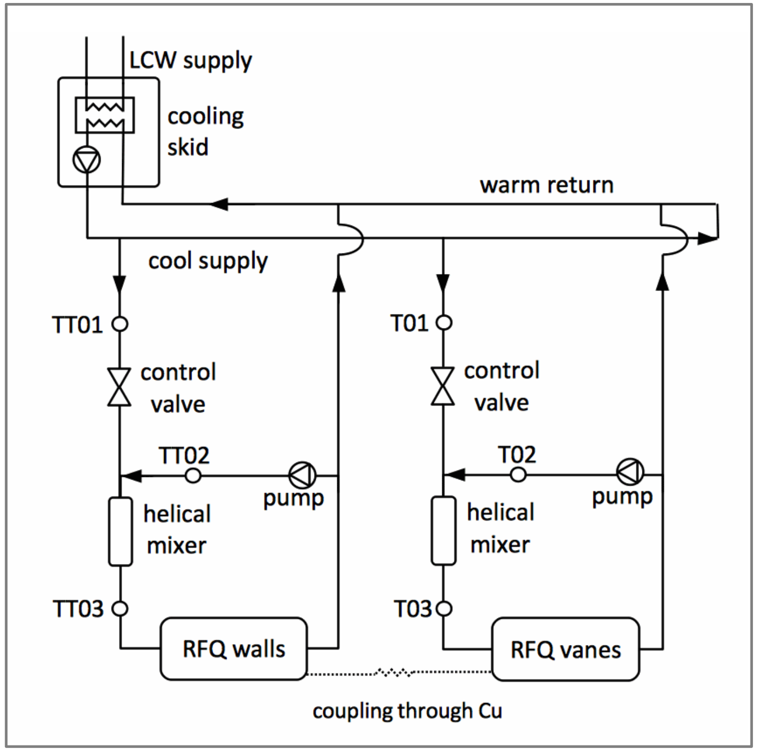
Range of RF duty factors and pulse patterns (up to CW)

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

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







Why does this matter for normal-conducting cavities?

The LLRF system will compensate for detuning by increasing forward power

Why does this matter for normal-conducting cavities?

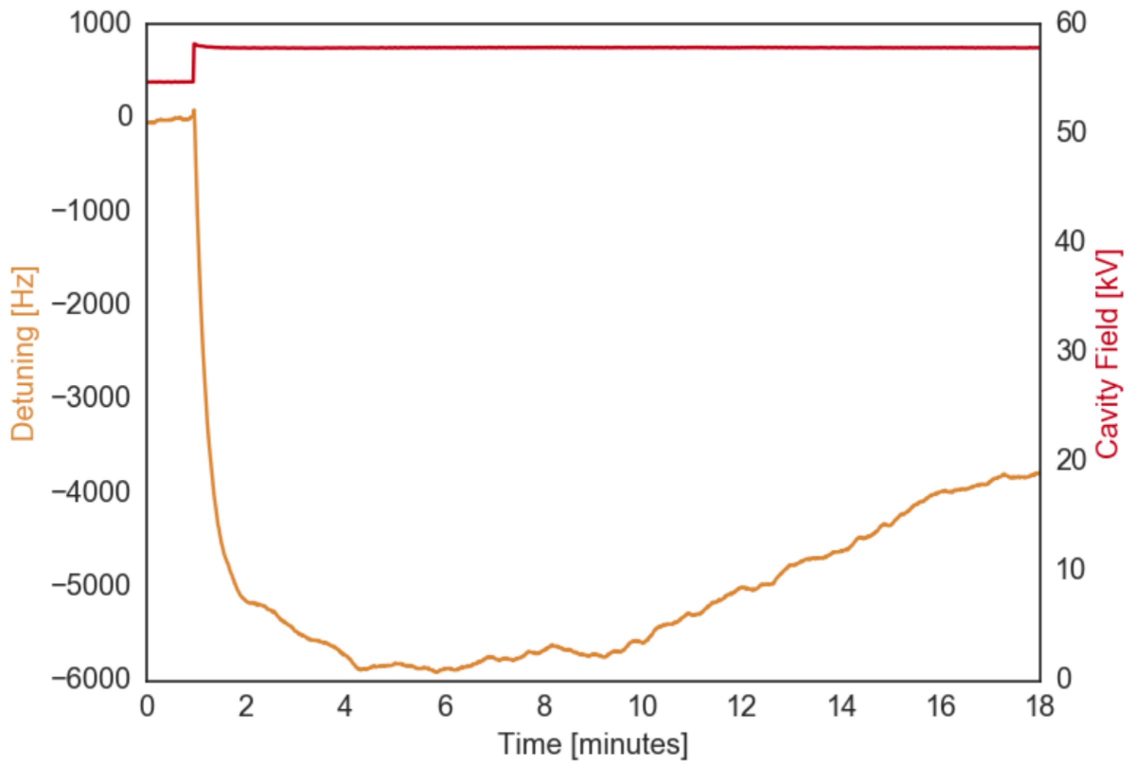
The LLRF system will compensate for detuning by increasing forward power

But…

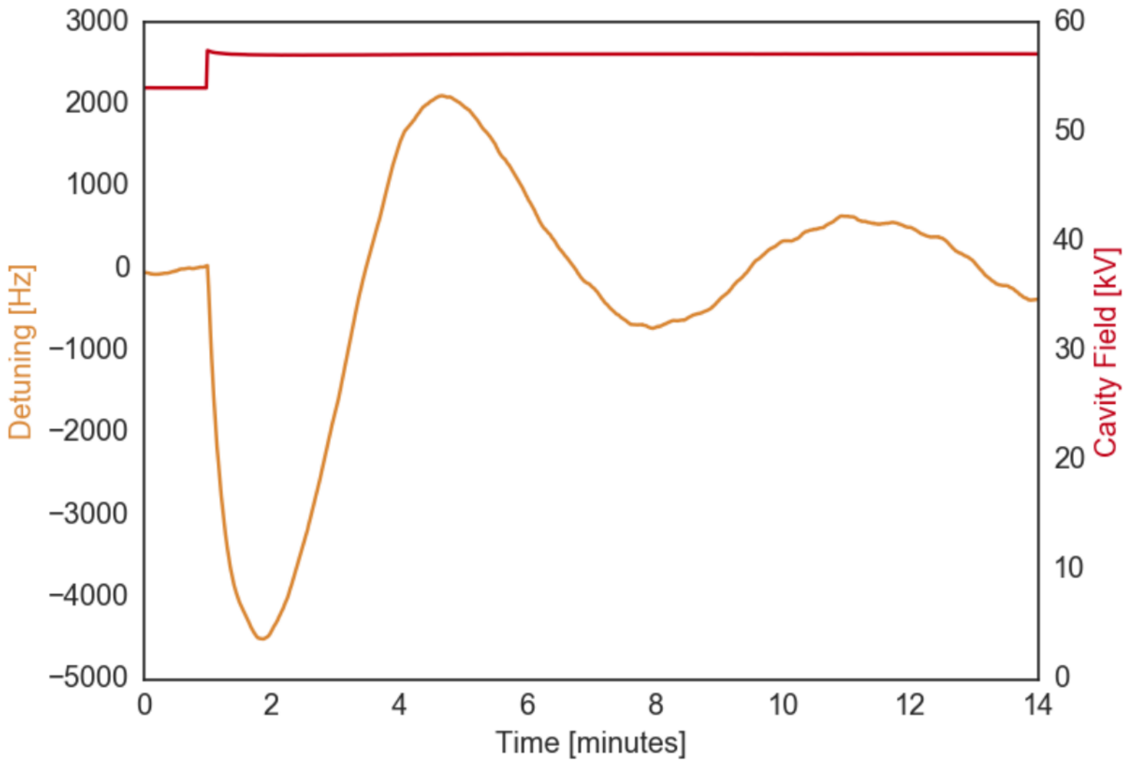
- Ability to do this bounded by the amplifier specs
- If detuned beyond RF overhead → *interrupt normal operations*
- RF overhead adds to initial machine cost and footprint
- Using additional RF power → *increasing operational cost*
- Increased waste heat into cooling system → *increasing operational cost*

RFQ Detuning in CW Mode

For a small change in cavity field (55 kV to 58 kV)…

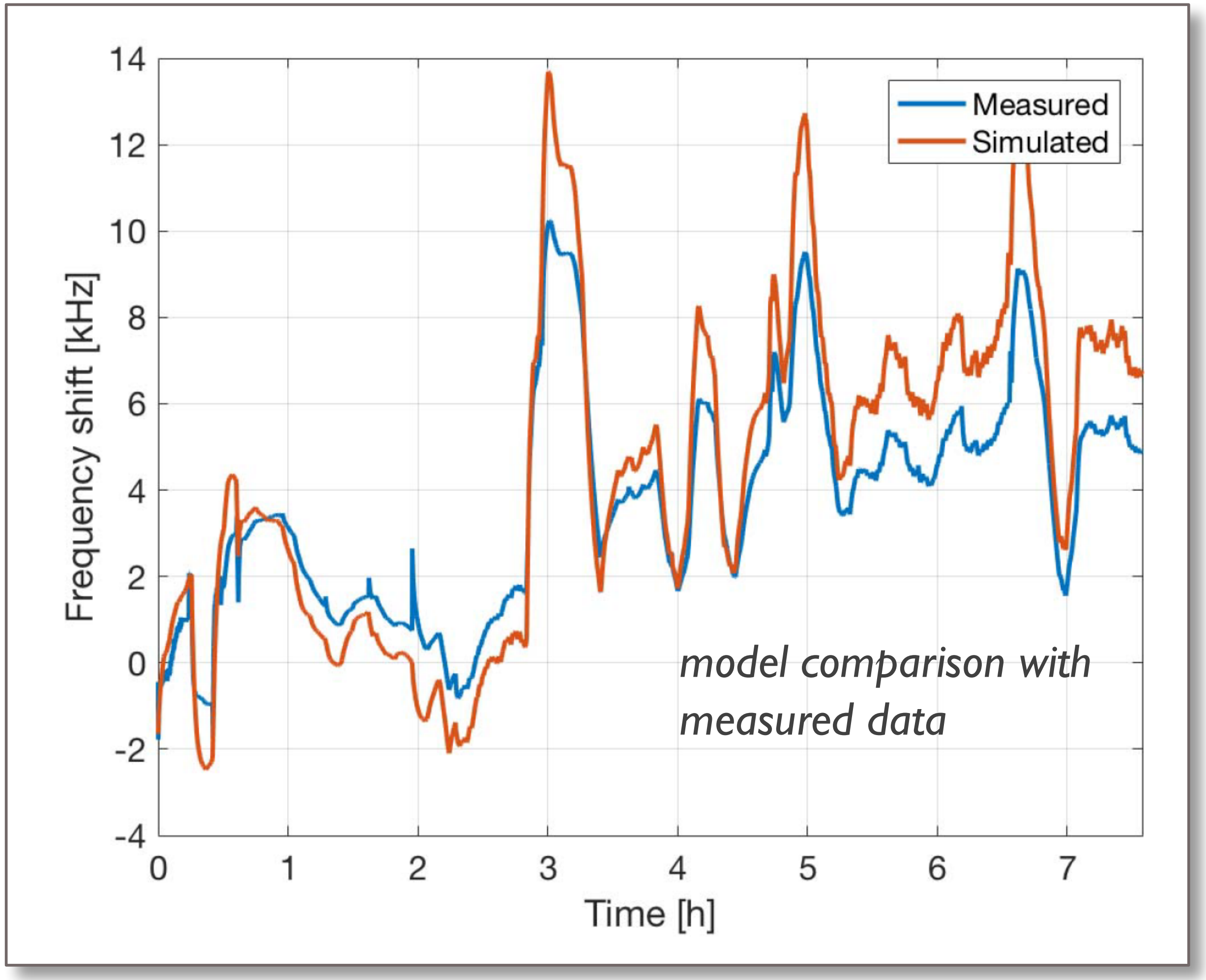


Uncontrolled



PI Frequency Control

Created a fast first-principles model, so why not use that in MPC instead of a NN?



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

Model needs to be sufficiently accurate for MPC

Assessed performance using measured input data:
4 ms RF pulse duration, 10 Hz rep rate
variety of valve and power settings

1.67 kHz RMS error
4.01 kHz max error
Maximum acceptable detuning is 3 kHz

Not accurate enough for control with MPC!

even after extensive tuning of uncertain parameters using an optimizer

Also looked at a linear learned model: still too poor
1.13 kHz RMS, 2.66 kHz max error

Initial Neural Network Modeling

*wanted to make sure we **could** model the response before moving forward*

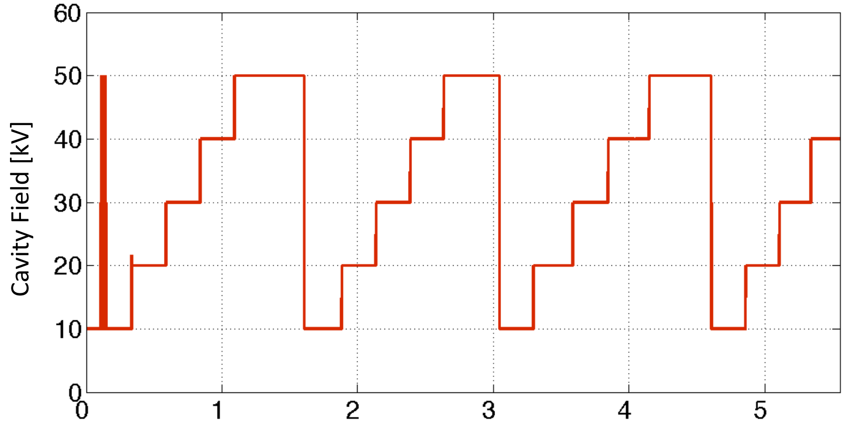
Feed-forward, Fully-connected Network

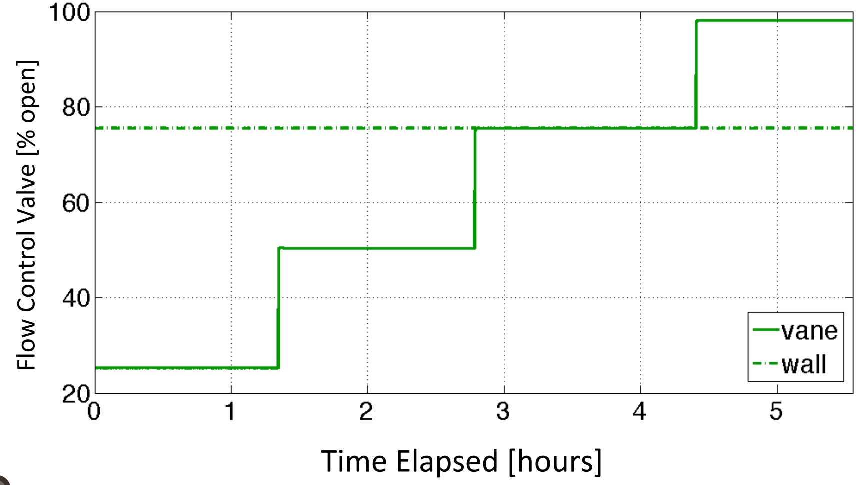
x_{t-hist}
⋮
x_t
→ NN → f_{t+1}

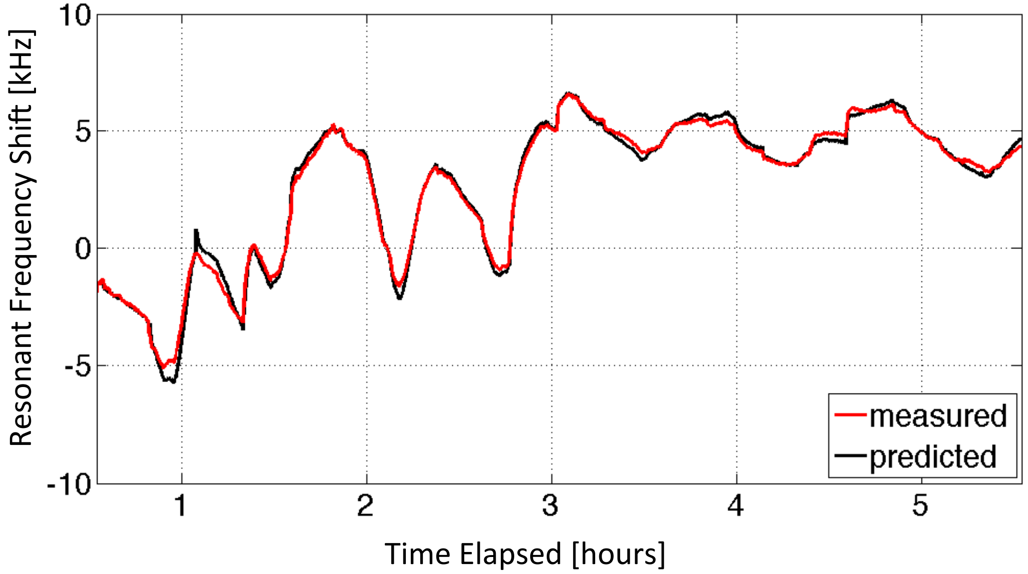
x includes:
- Valve settings
- Average RF power
- Water temperatures
- Cave temperature
- Cave humidity

Two hidden layers: 25 and 7 nodes

hist: 30 minutes at 1 Hz







Mean Absolute Error

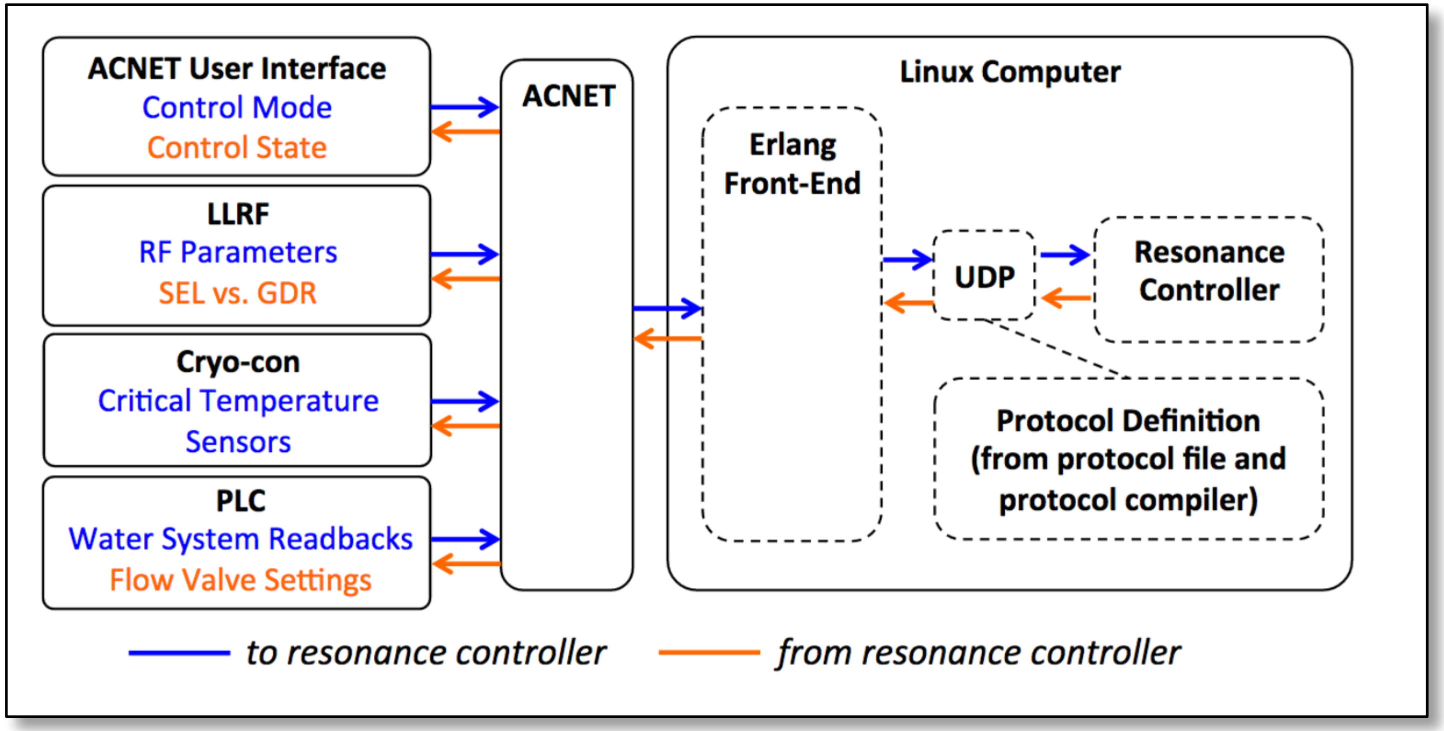
346 Hz – test set
98 Hz – validation set
115 Hz – across all sets

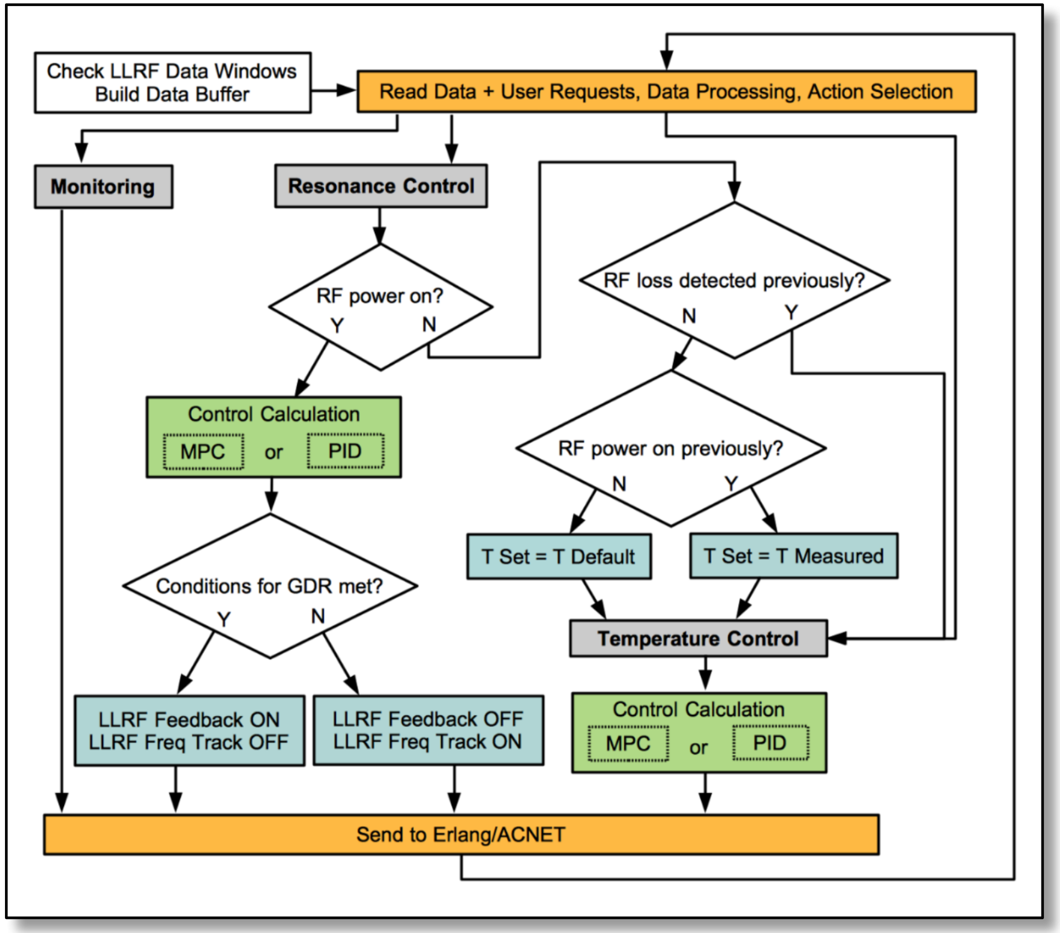
Training Data

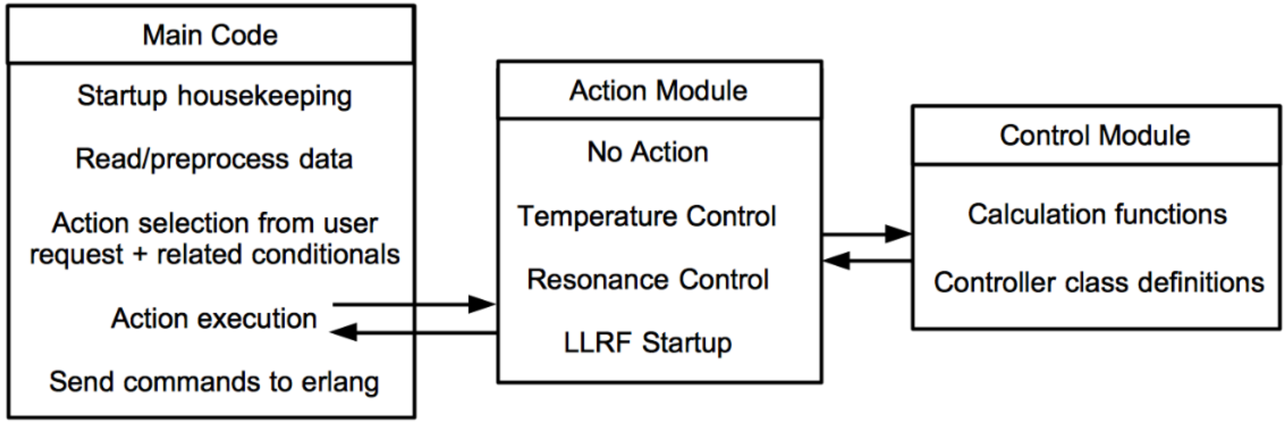
~ 64 hours of measurements
Scanned average RF power, valves
Includes RF trips, startup/shutdown

Built a python-based control framework

- *Executes on controls network linux computer*
- *PI control in regular operational use*
- *Designed to be portable + modular*
- *Supports the use of ML libraries*







Recap of Application Areas and Examples

- Model Predictive Control with Neural Network Models
  - Especially useful for systems with long-term time dependencies
    - *PIP-II RFQ*
    - *FAST RF gun*
- Modeling using Measured and/or Simulated Data
  - Create a fast simulation tool for online modeling
    - *FAST linac*
    - *FEL energy switching study (see tomorrow's talk)*
  - Create models from measured data alone
    - *JLab trajectory control*
    - *PIP-II RFQ*
    - *FAST RF gun*
  - Combine observed behavior and a priori knowledge
    - *FAST linac, PIP-II RFQ*
- Neural Network Control Policies
  - Tuning and changing operating state
    - *JLab FEL trajectory control*
    - *FEL energy switching study (see tomorrow's talk)*
  - Learning from existing control policies
    - *Present PIP-II RFQ work*
- Incorporating Image-based Diagnostics Directly into Control Policies
    - *FAST linac study*
- Virtual Diagnostics
  - Predict beam parameters when diagnostic not available or not in use
    - *FAST linac study (measurements from an intercepting diagnostic)*

Predicting the full horizon at once

Example of Qualitative Performance
on Farthest-Ahead Time Step
(for one of the many models trained)