Slow Spill Regulation with Machine Learning

Replacing PID controllers with neural networks

Introduction

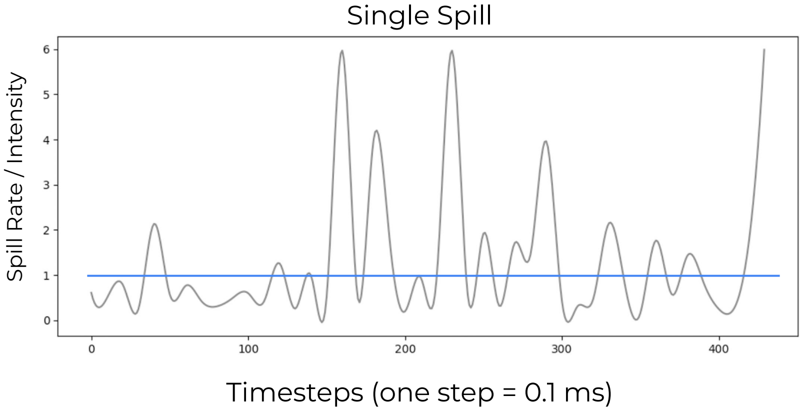
Mu2e is an upcoming experiment at Fermilab that intends to capture Muons in Aluminum atoms and look for new physics in its decay to electrons.

To help increase the signal strength, Mu2e demands pulses of muons arrive at the Aluminum target with ***strict requirements on the rate uniformity***.

To create the muons, proton pulses are made to hit a production target and muons are obtained from the secondaries. The proton pulses with the required time structure are created by the slow extraction of bunched beam from the Delivery Ring.

The extraction (or ‘spill’) of protons from the Delivery Ring is achieved using third integer resonance extraction.

Objective: Regulate the uniformity of the extracted spill - or increase its Spill Duty Factor (SDF) - by regulating the slow extraction process.



$$\boxed{\mathrm{SDF} = \frac{1}{1 + \sigma_{\mathrm{spill}}^2}}$$

Introduction

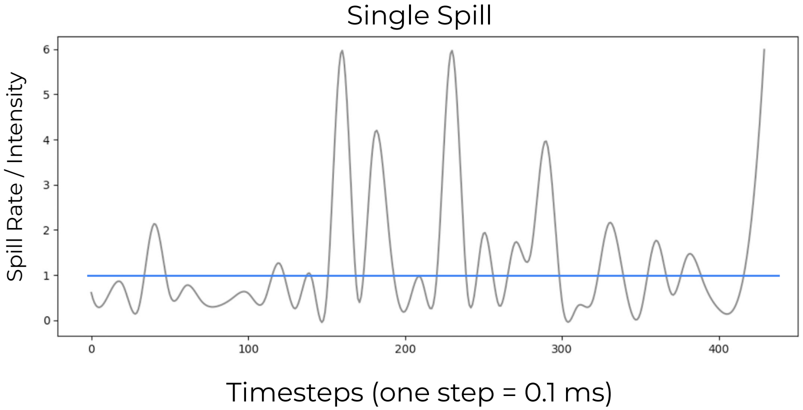
Objective: Regulate the uniformity of the extracted spill - or increase its Spill Duty Factor (SDF) - by regulating the slow extraction process.

Historical approach: PID Controllers

- PIDs are a linear and symmetric heuristic control system with constant parameters, meaning they are designed to operate in domains in which the response of the system is invariant across all operating regions.

Proposed Approach: Learned Controllers

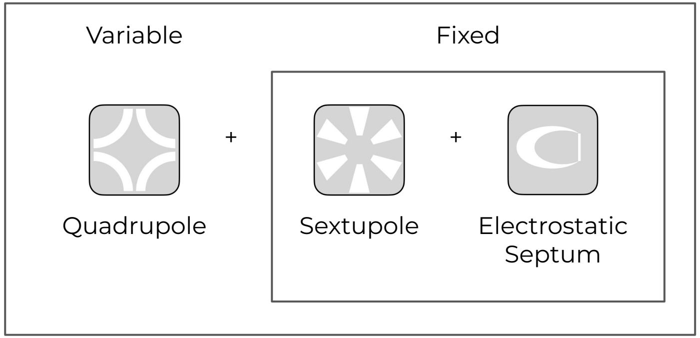
- As we cannot presume the exact noise distribution and possible nonlinearities in the extraction system, a control system capable of adapting to the nonlinearities of the extraction system is warranted.
- Modern neural networks represent a class of arbitrary function approximators and, as such, are a natural solution for extending resonant extraction control systems into the nonlinear regime.



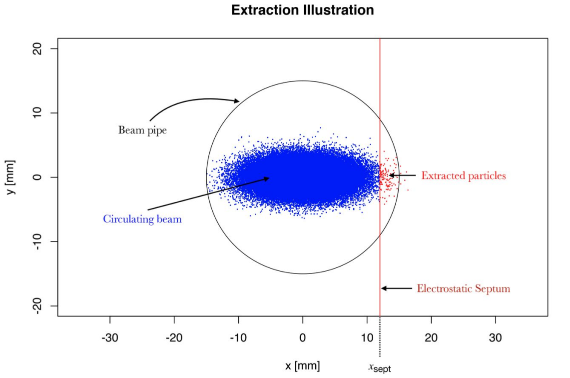
$$\mathrm{SDF} = \frac{1}{1 + \sigma^2_{\mathrm{spill}}}$$

Regulation system at a glance

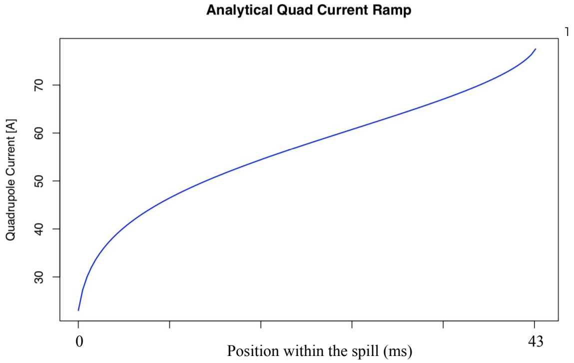
Extraction System



In our setting, we focus on modulating the quadrupole current to control the extraction rate. For our purposes, we consider the sextuple and electrostatic septum as fixed.



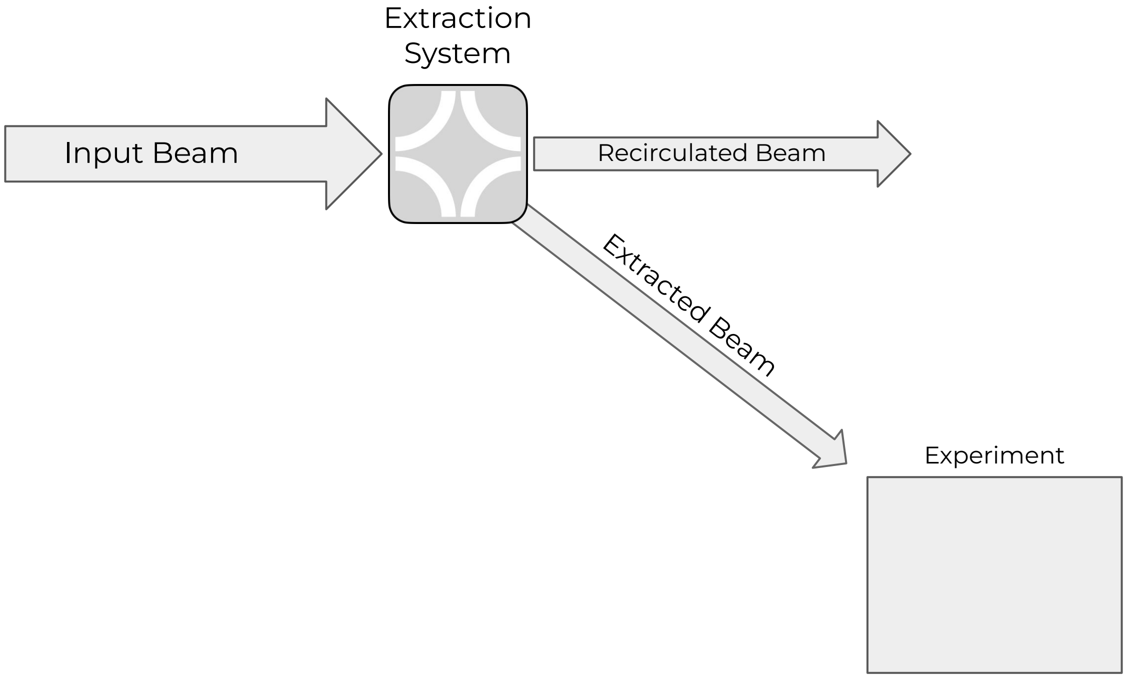
A snapshot of the beam in physical space at the extraction location. As the horizontal beam size increases, a slice of circulating beam (that is past the position of the electrostatic septum) is extracted.



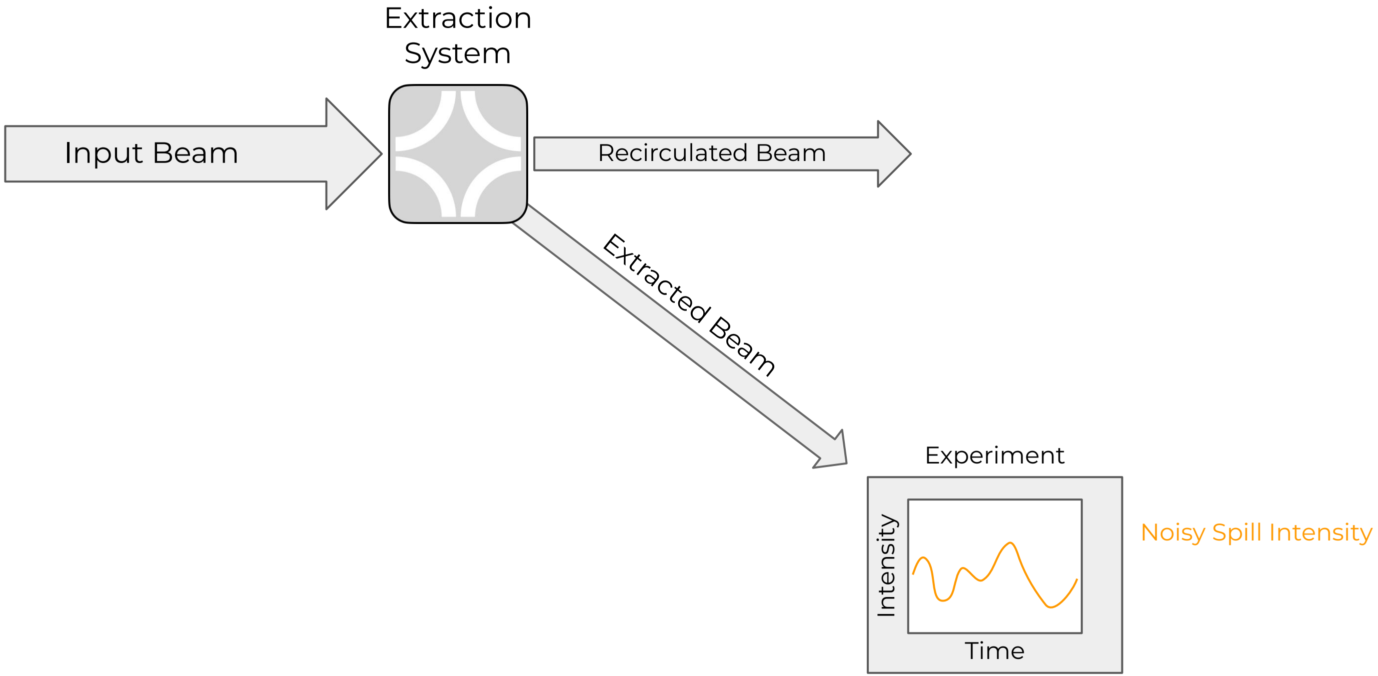
Analytically calculated quadrupole current ramp for one full spill duration.

[1] A. Narayanan, THIRD-INTEGER RESONANT EXTRACTION REGULATION SYSTEM FOR MU2E, 2022

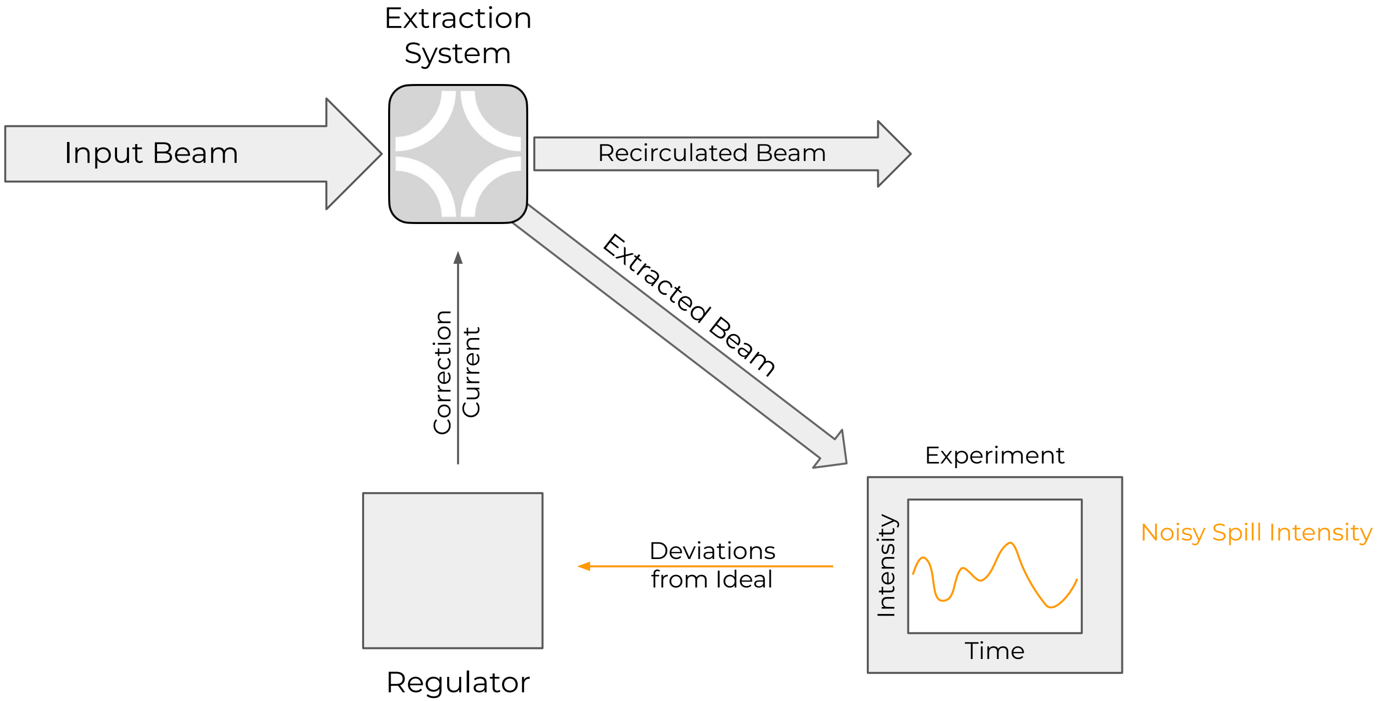
Problem Setup



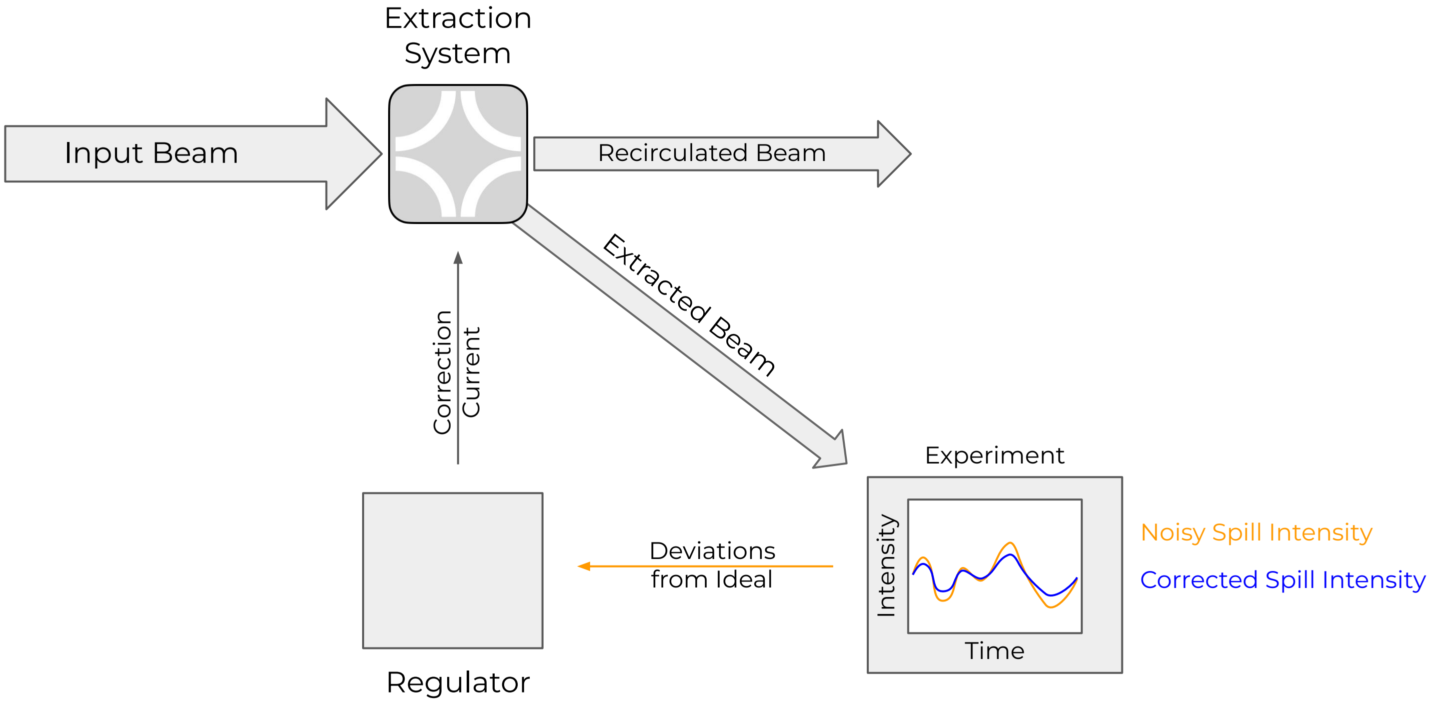
Problem Setup



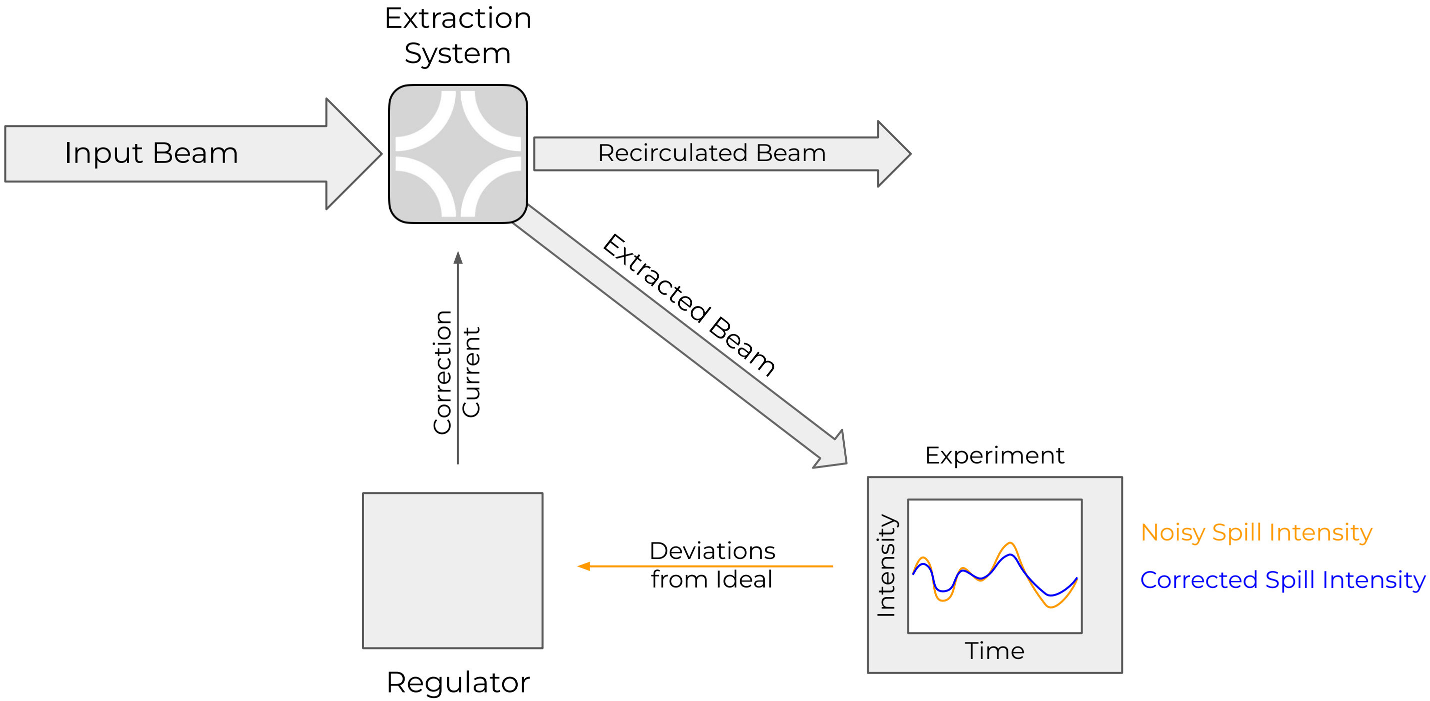
Problem Setup



Problem Setup



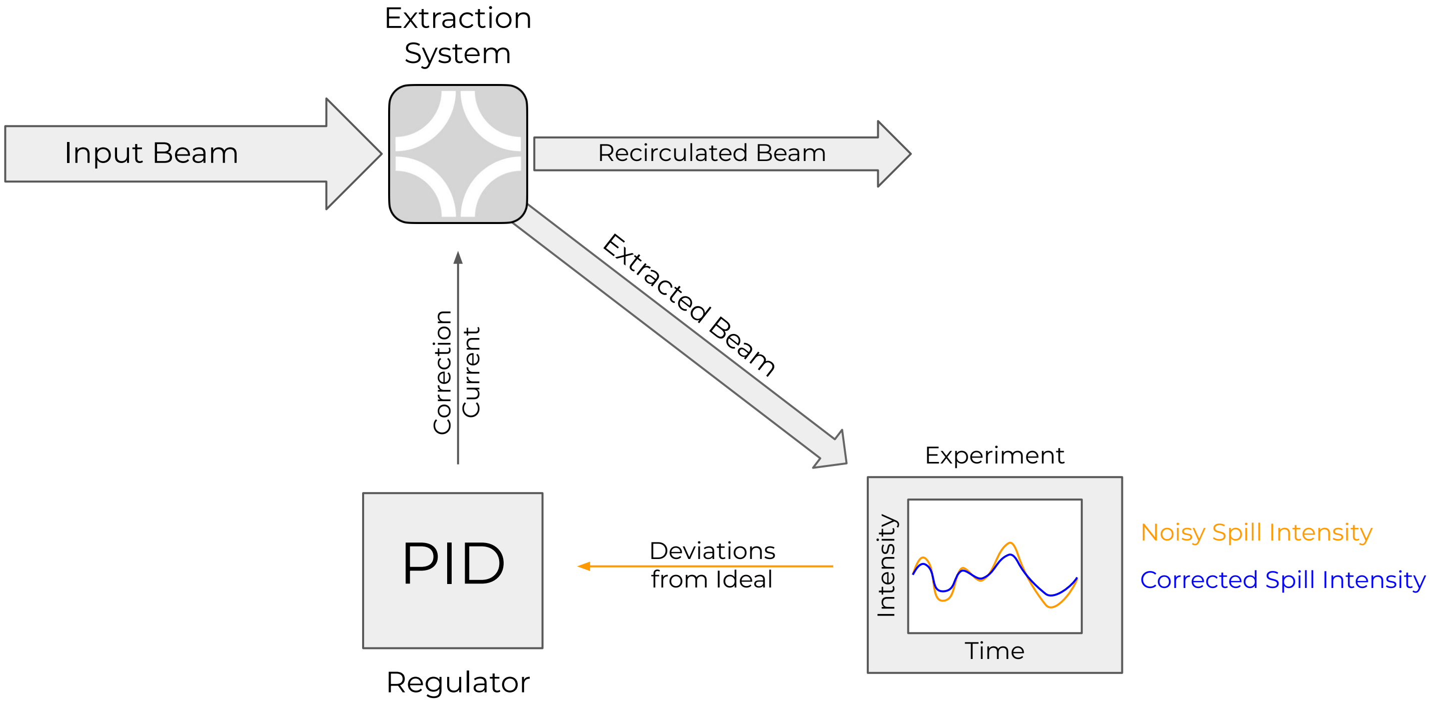
Problem Setup



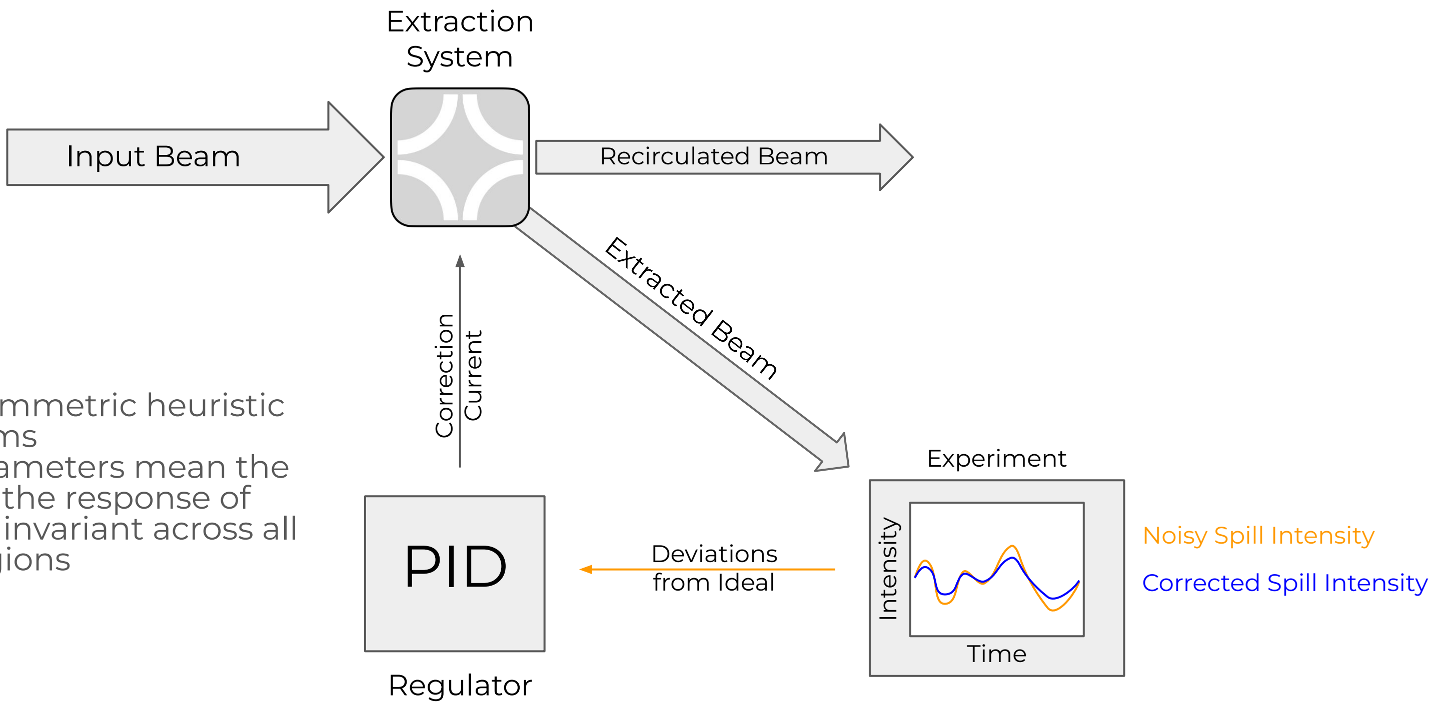
Objective:

Regulate the extraction to produce the *smoothest* corrected intensity profile possible.

Existing Method



Existing Method



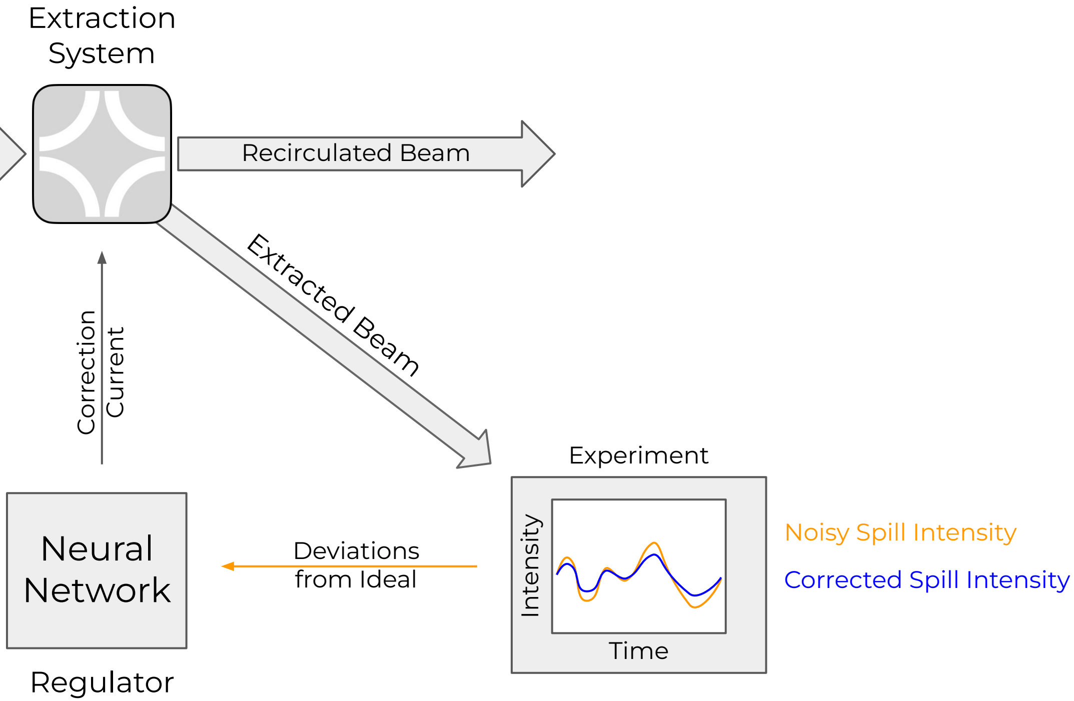
PIDs

- Linear and symmetric heuristic control systems
- Constant parameters mean the PID assumes the response of the system is invariant across all operating regions

Our setting

- Possibly nonlinear noise profile
- Response of the extractor varies across operating regions

Proposed Method



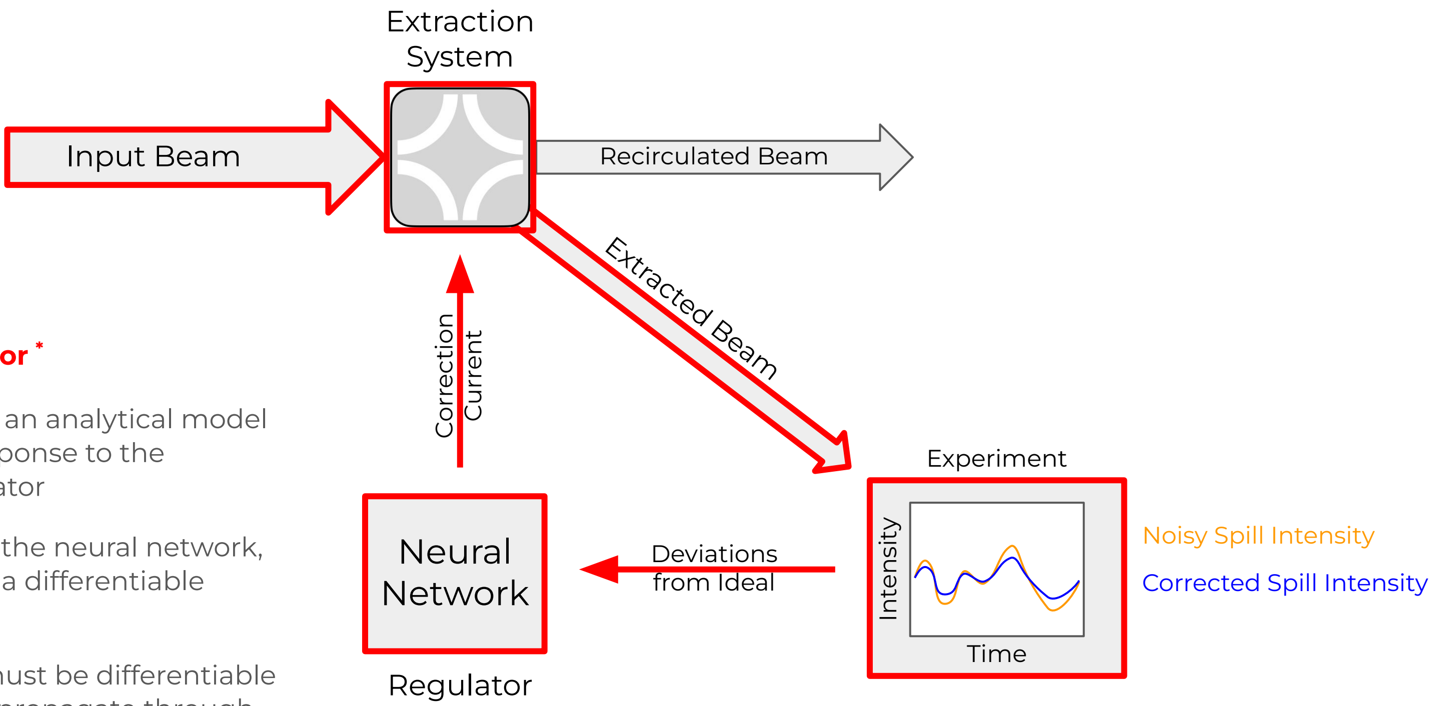
Neural Networks

- Nonlinear function approximators that adapt to the input data
- Can be made ‘domain-aware’ by including positional information

Our setting

- Possibly nonlinear noise profile
- Response of the extractor varies across operating regions

Proposed Method

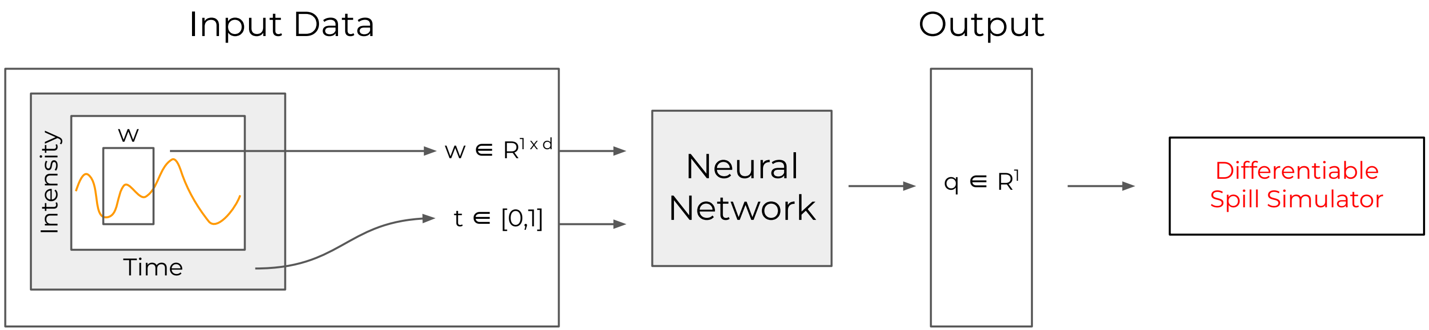


Differentiable Simulator *

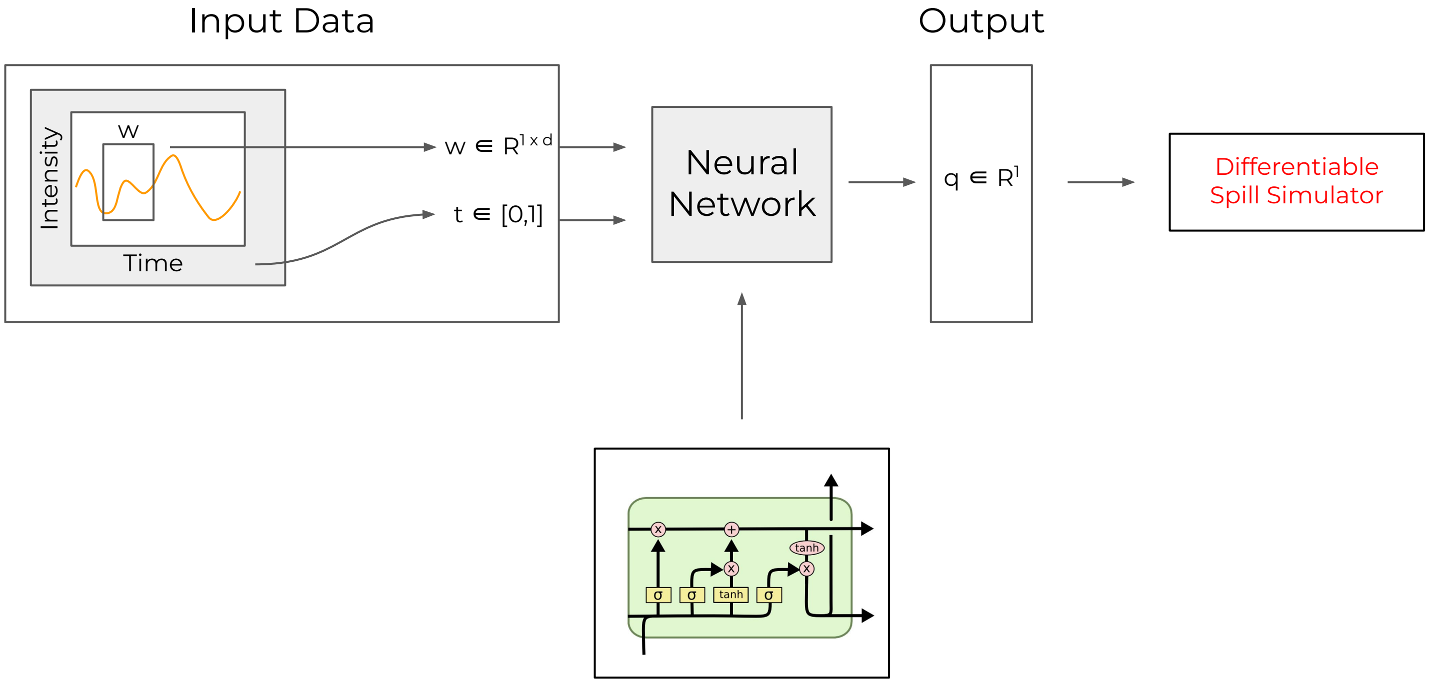
- The simulator is an analytical model of the beam response to the regulation actuator
- In order to train the neural network, we constructed a differentiable simulator.
- The simulator must be differentiable in order to backpropagate through the spill physics to update the neural network weights.

* A. Narayanan *et al.*, "Optimizing Mu2e Spill Regulation System Algorithms", in *Proc. IPAC'21*, May 2021, pp. 4281-4284. doi:10.18429/JACoW-IPAC2021-THPAB243

Formulation for Machine Learning



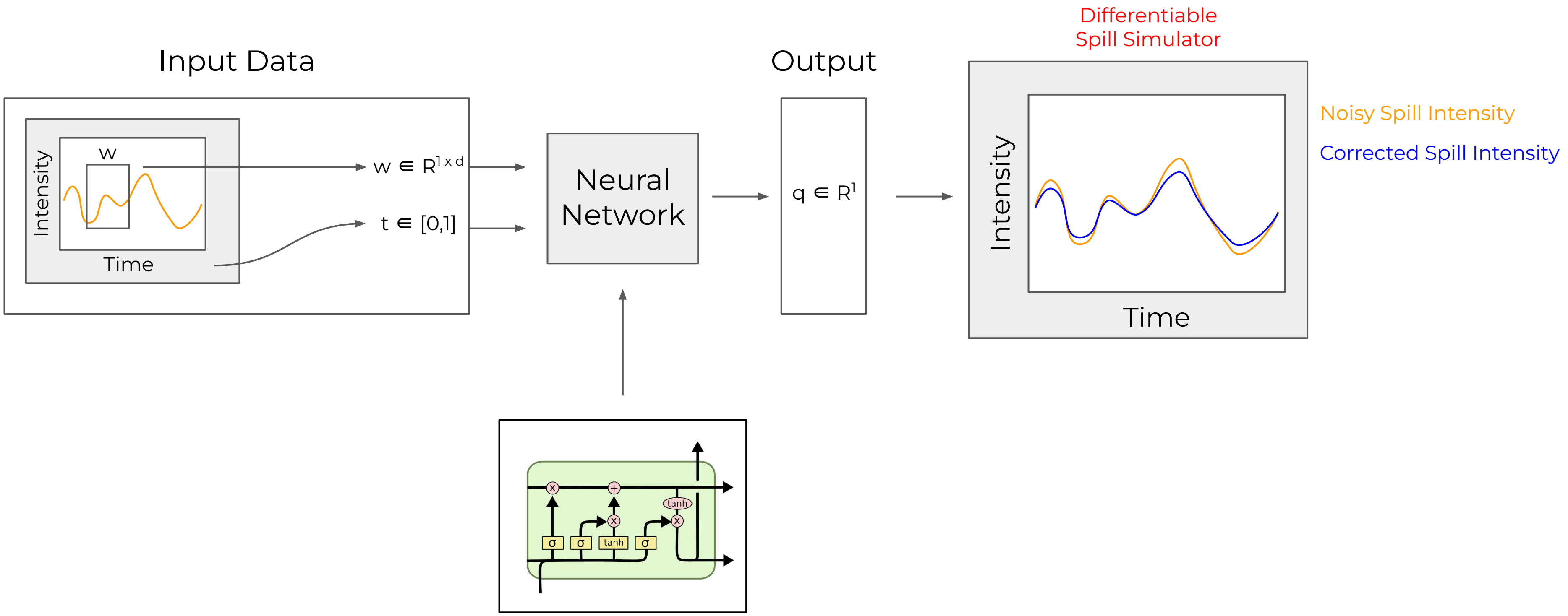
Formulation for Machine Learning



Gated Recurrent Unit (GRU)*

* https://colah.github.io/posts/2015-08-Understanding-LSTMs/

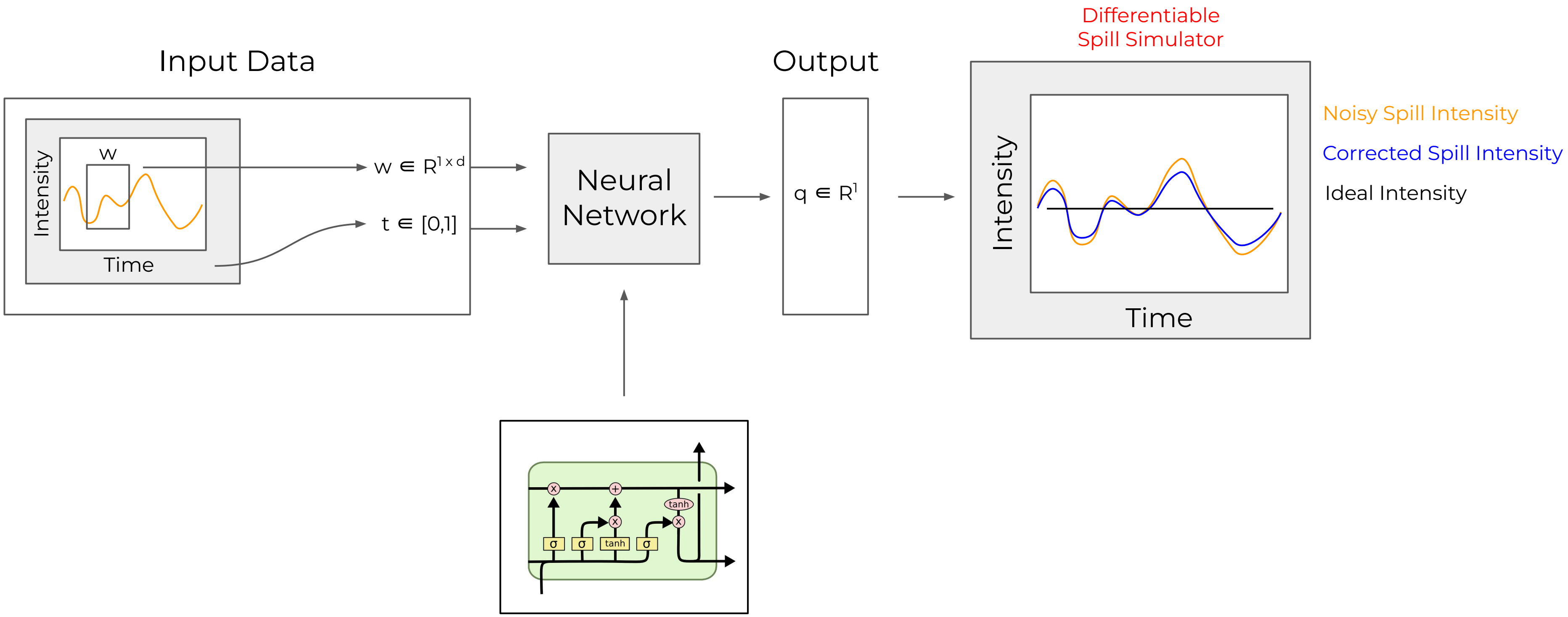
Formulation for Machine Learning



Gated Recurrent Unit (GRU)[1]

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

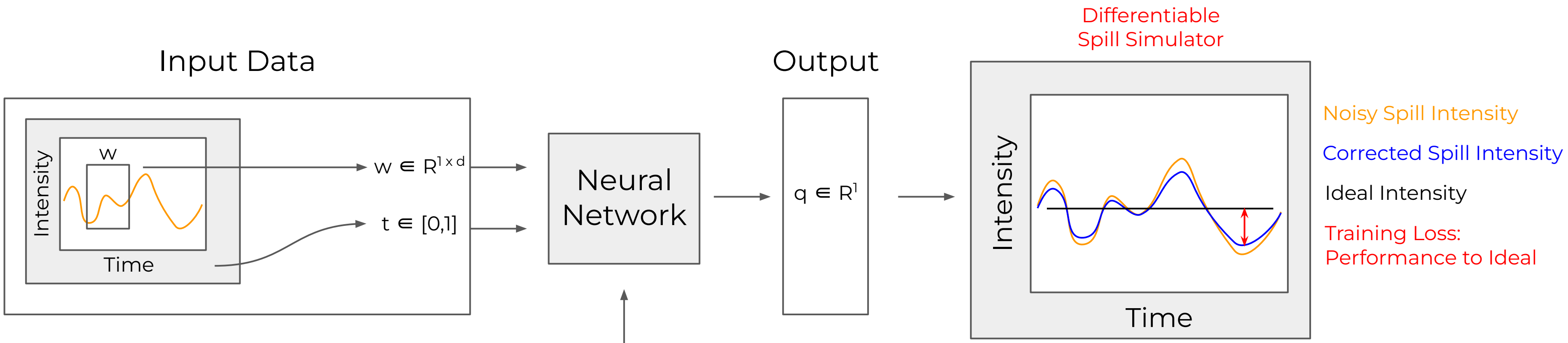
Formulation for Machine Learning

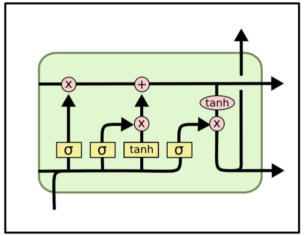


Gated Recurrent Unit (GRU)[1]

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Formulation for Machine Learning





Gated Recurrent Unit (GRU)[1]

Training Loss = *MSE*(Corrected Spill Intensity - Ideal Intensity)

Supervised Learning

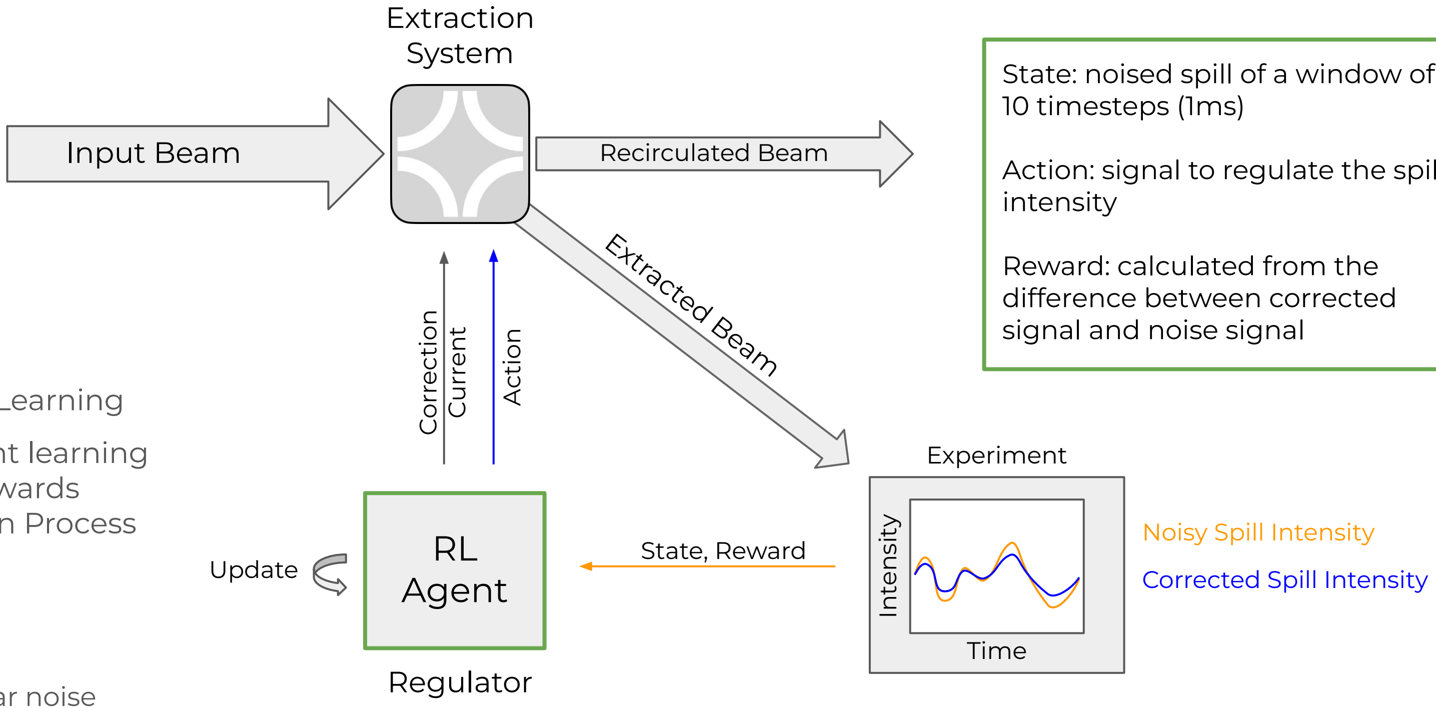
The model is trained to minimize the difference between the Corrected Spill Intensity and the Ideal Intensity.

Reinforcement Learning

The model is trained to maximize the inverse of the difference between the Corrected Spill Intensity and the Ideal Intensity.

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Proposed Method: Reinforcement Learning



State: noised spill of a window of 10 timesteps (1ms)

Action: signal to regulate the spill intensity

Reward: calculated from the difference between corrected signal and noise signal

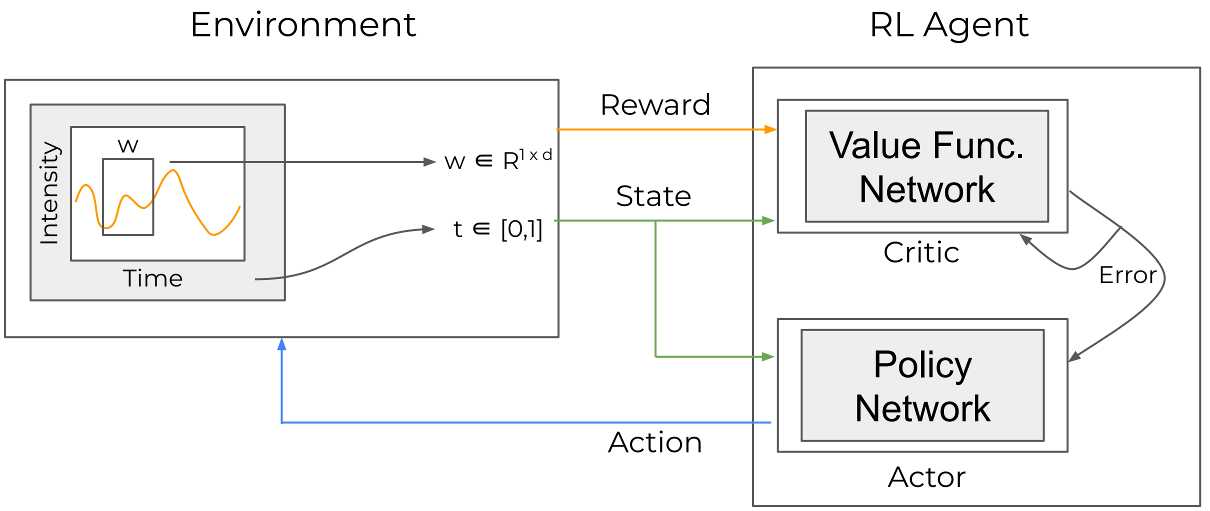
Deep Reinforcement Learning

- Intelligent agent learning to maximize rewards
- Markov Decision Process

Our setting

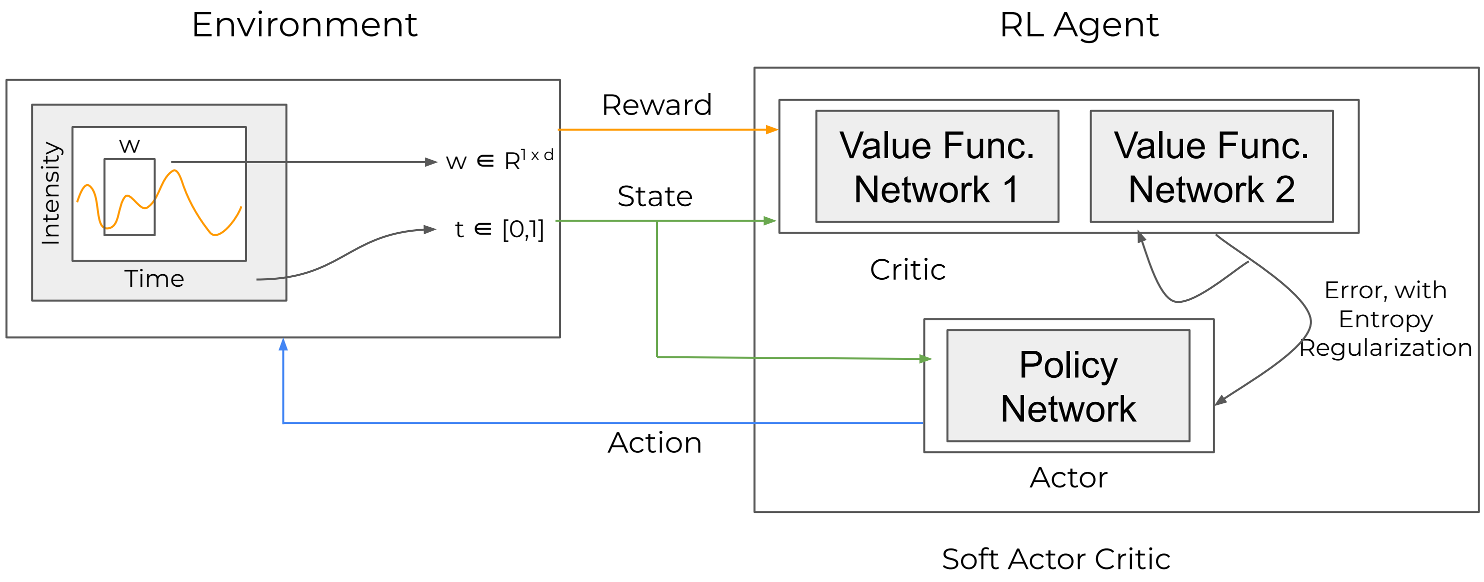
- Possibly nonlinear noise profile
- Response of the extractor varies across operating regions

Formulation for Reinforcement Learning

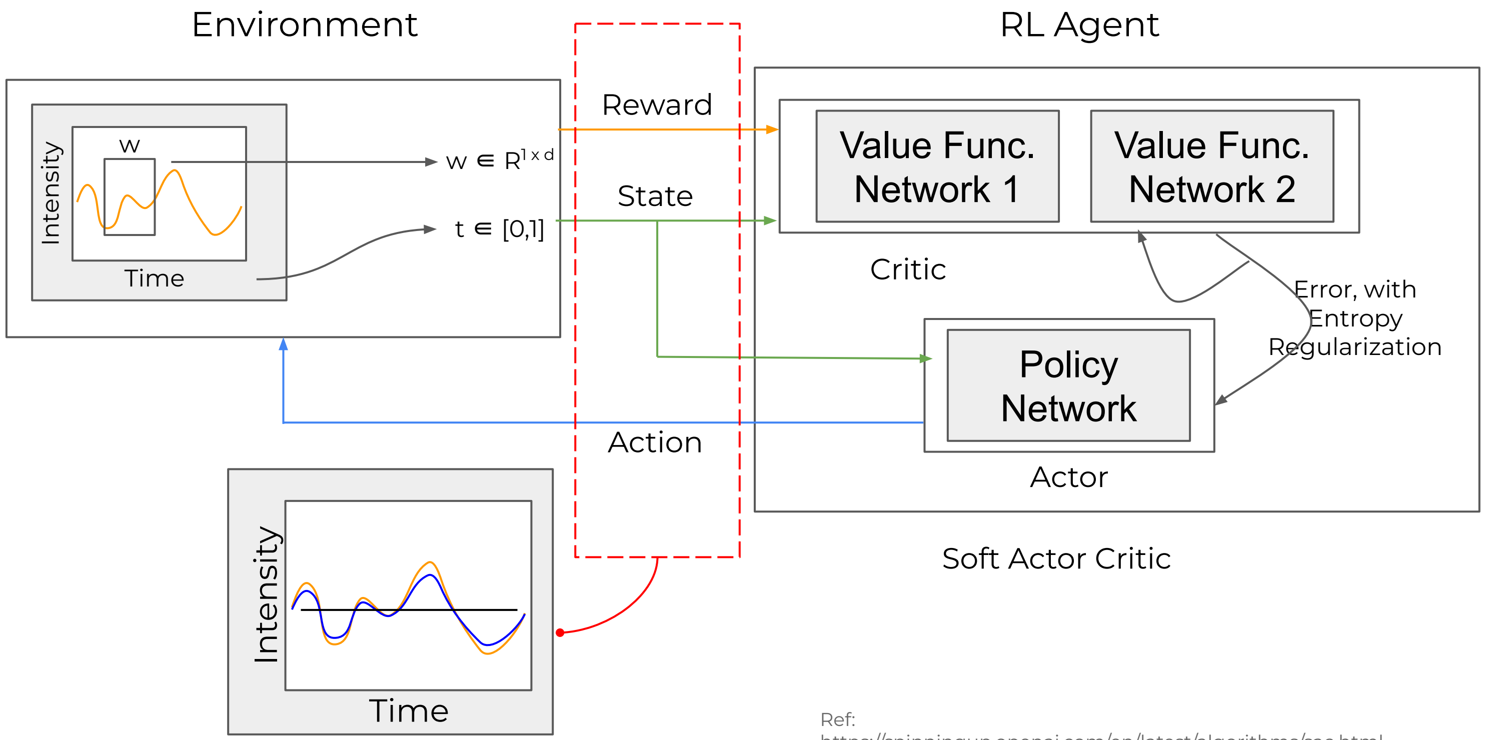


Actor Critic Algorithm

Formulation for Reinforcement Learning

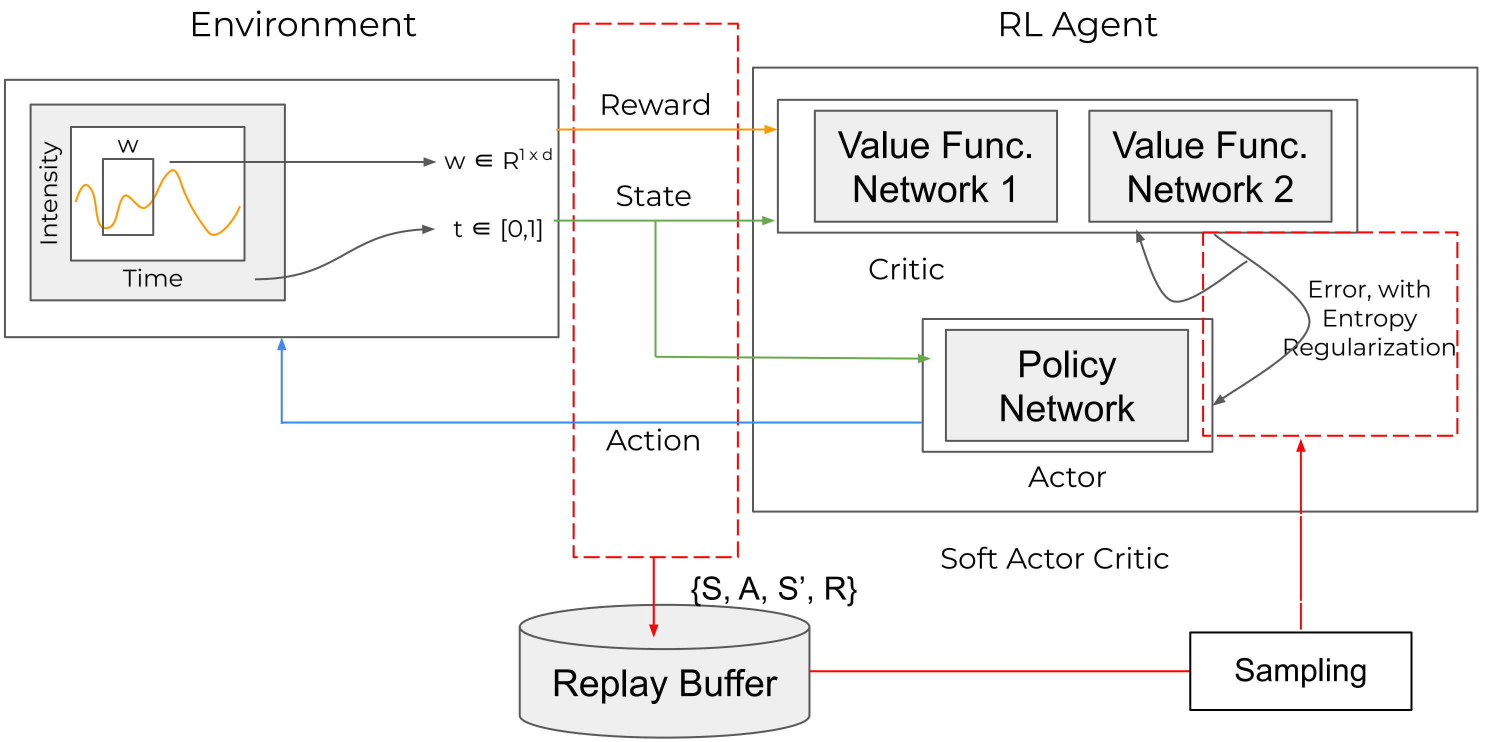


Formulation for Reinforcement Learning



Ref: https://spinningup.openai.com/en/latest/algorithms/sac.html

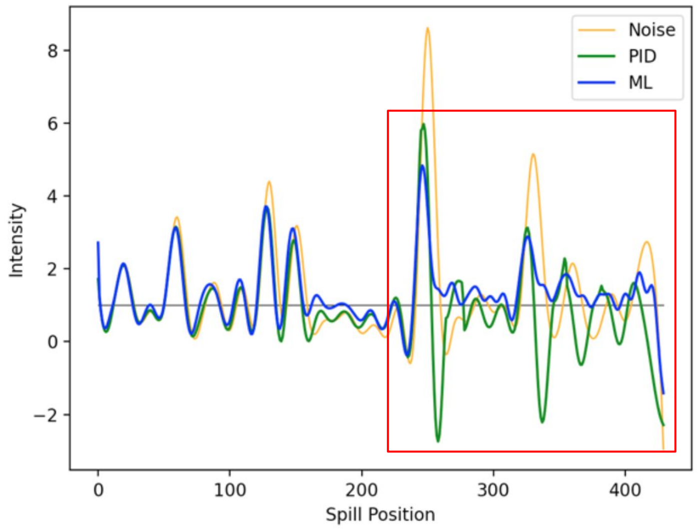
Formulation for Reinforcement Learning

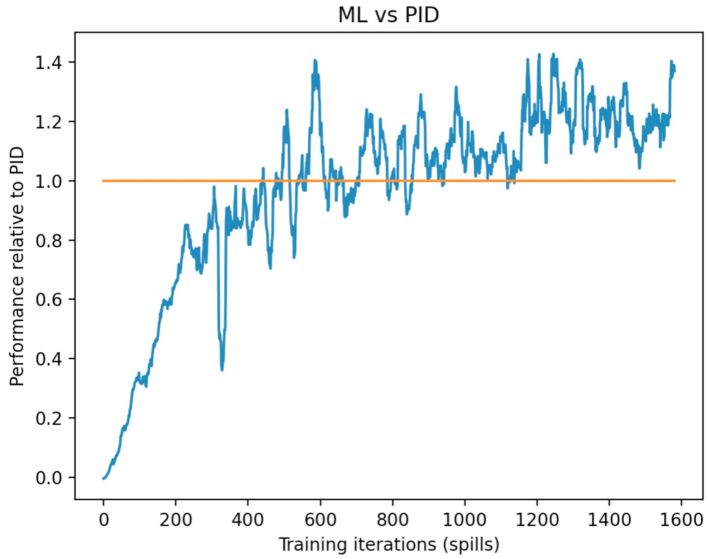


Results: Supervised Learning

As training proceeds, the GRU gradually exceeds the PID regulation performance.

Single Spill Example, PID vs ML

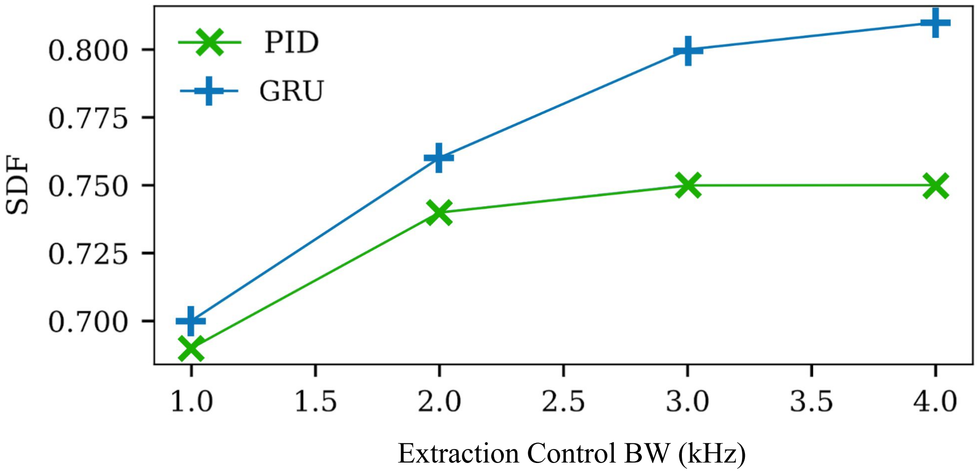




Results: Supervised Learning

The GRU model outperforms the PID in all scenarios, with the performance difference *increasing* as we increase the system bandwidth.

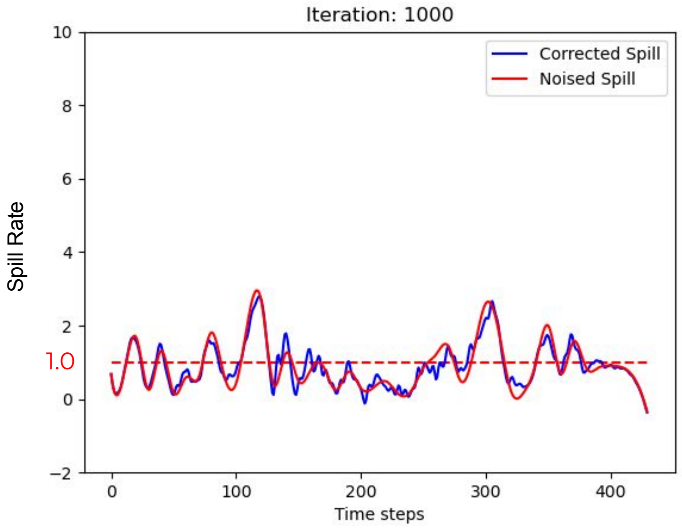
$$\mathrm{SDF} = \frac{1}{1 + \sigma_{\mathrm{spill}}^2}$$



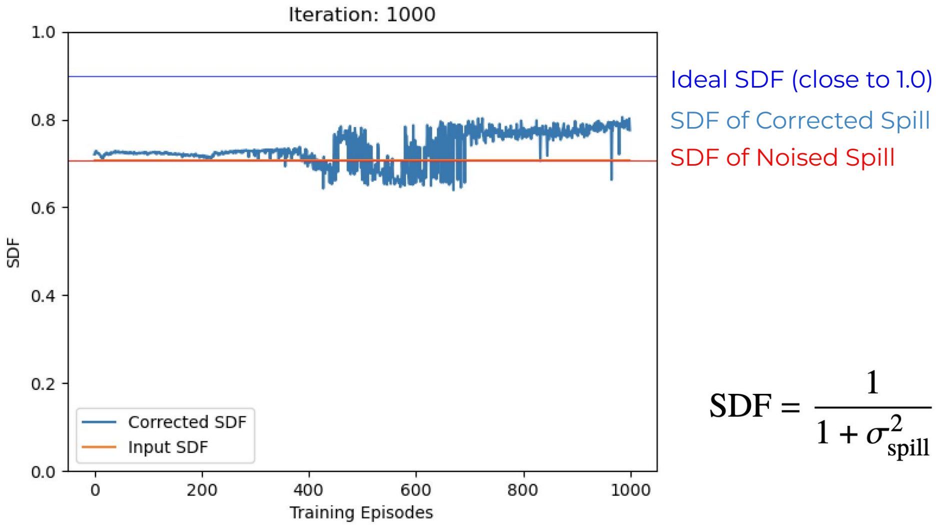
Average corrected SDF over 1,000 spills. Input noise has an average of SDF of 0.51.

Preliminary Results: Reinforcement Learning

The RL agent is capable of generating control signals at each timestep that regulate the spill towards 1.



Corrected spill vs. Noised spill at episode 1000



$$\mathrm{SDF} = \frac{1}{1 + \sigma_{\mathrm{spill}}^2}$$

Average SDF of corrected spill over 1000 training episodes

Summary

- Constructed a differentiable slow spill extraction simulator that allows us to train neural networks to regulate the slow spill extraction rate.
- Showed that a simple recurrent neural network (GRU) can outperform an optimized PID controller (as measured by relative increases in the SDF of the corrected spill).
  - Further showed that this difference increases as we increase the system bandwidth.
- WIP: Showed encouraging results using an RL-based regulation system

Acknowledgements

M. Thieme *
J. Jiang *
A. Narayanan *
V. P. Nagaslaev

J. Arnold
M. Austin
J.R. Berlioz
P. Hanlet
K.J. Hazelwood
M.A. Ibrahim
D.J. Nicklaus
G. Pradhan
P.S. Prieto
B.A. Schupbach
A. Saewert
K. Seiya
R.M. Thurman-Keup
N.V. Tran
D. Ulusel

S. Memik
R. Shi
H. Liu

Fermilab

Northwestern University

* Equal contribution

Appendix

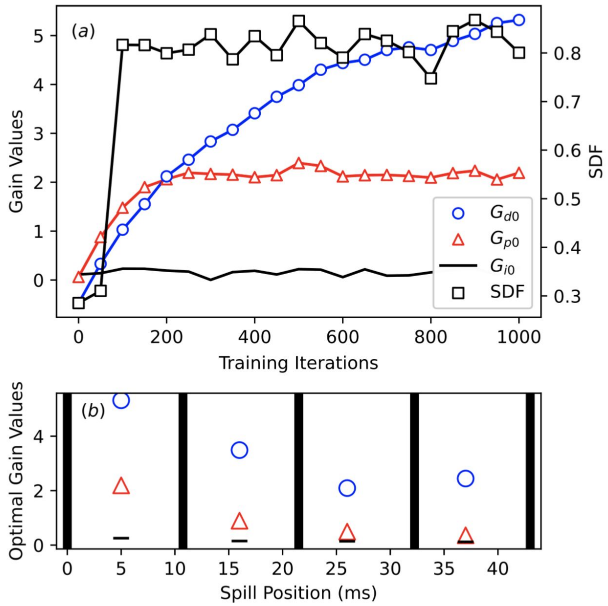


Figure 2: Top (a) Evolution of the PID gains in domain-0 (leftmost subdomain of bottom plot) over the full spill, as well as the SDF. Bottom (b) Four subdomains of the spill are segmented by vertical bars. Optimal gain values within each subdomain are shown on the vertical axis.