Beyond PID Controllers: PPO with Neuralized PID Policy for Proton Beam Intensity Control in Mu2e

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We introduce a novel Proximal Policy Optimization (PPO) algorithm aimed at addressing the challenge of maintaining a uniform proton beam intensity delivery in the Muon to Electron Conversion Experiment (Mu2e) at Fermi National Accelerator Laboratory (Fermilab). Our primary objective is to regulate the spill process to ensure a consistent intensity profile, with the ultimate goal of creating an automated controller capable of providing real-time feedback and calibration of the Spill Regulation System (SRS) parameters on a millisecond timescale. We treat the Mu2e accelerator system as a Markov Decision Process suitable for Reinforcement Learning (RL), utilizing PPO to reduce bias and enhance training stability. A key innovation in our approach is the integration of neuralized PID controller into the policy fucntion, resulting in a significant improvement in the Spill Duty Factor (SDF) by 9.4%, surpassing the performance of the current PID controller baseline by an additional 2.2%. This paper presents the preliminary offline results based on a differentiable simulator of the Mu2e accelerator. It paves the ground works for real-time implementations and applications, representing a crucial step towards automated proton beam intensity control for the Mu2e experiment.

1 Introduction

We propose a novel RL-enhanced spill regularization system that incorporates a neuralized PID policy function to tackle the beam regularization challenge in the Mu2e experiments [Bartoszek et al., 2015] at Fermilab. Our objective is to create an automated controller that ensures consistent spill (proton beams) intensity during experiments meeting real-time control requirements [Narayanan et al., 2021]. To achieve this, we model the Mu2e accelerator system as a Markov Decision Process and employ the Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017] to cast the spill regulations as sequential decision-making problems. Our main contribution is the integration of a neuralized PID policy function [Zribi et al., 2018] for our RL framework, encompassing the inductive bias of the standard PID controller (i.e. the proportional, integral, and derivative information) to better capture states at different stages. Our experiments on the Mu2e simulator show that we observed a 9.4% improvement in the Spill Duty Factor (SDF). Additionally, our method outperforms the standard PID controller referenced in [Narayanan et al., 2021].

The Mu2e at Fermilab seeks to investigate new physics by studying the decay of muons into electrons. This intricate experiment places stringent demands on the quality of the proton beam directed at the production target, see Figure 1. These requirements are essential to minimize background particle physics processes that could obscure the discovery signal. Key prerequisites for the Mu2e experiment include achieving (i) a highly uniform extracted beam intensity during each spill of protons with 8 GeV kinetic energy, and (ii) maintaining beam losses below 2% of the total beam power to control radiation, thereby reducing equipment activation and minimizing personnel exposure. To achieve this, the Spill Regulation System (SRS) is developed to govern the extraction process of the beam and mitigate various sources of fluctuations in the spill profile. The SRS adjusts the magnetic field to control the spill intensity by adjusting the currents of the magnets throughout the accelerator ring at Fermilab (Figure 1 (c)). Ideally, SRS aims for the spill intensity to be uniform.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

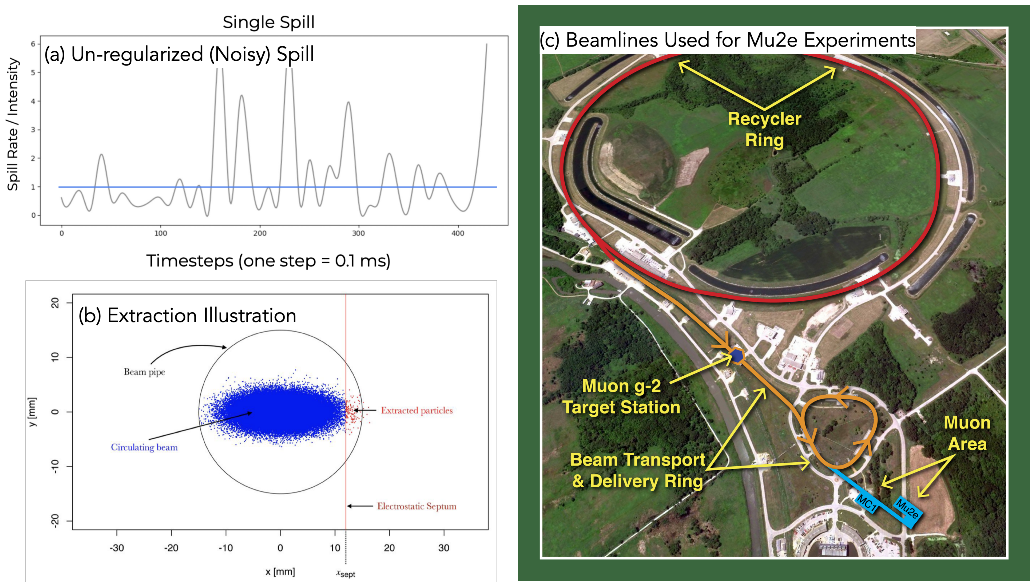

Figure 1: **(a)**: The extraction (or 'spill') of protons from the Delivery Ring is noisy (deviates from 1) without any regulation. **(b)**: A snapshot of the beam in physical space at the extraction location. As the horizontal beam size increases, a slice of circulating beam (that is past the position of the electrostatic septum) is extracted. **(c)**: 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 extracting them from an accelerator ring called Delivery Ring at Fermilab and sending it to the Mu2e production target.

Our approach involves utilizing a Proximal Policy Optimization model with a differentiable Mu2e simulator to rectify random generalized spills. In each step of the process, the simulator generates a series of random spills, each with varying intensities (shown in Figure 1 (a)). Subsequently, the PPO model intervenes in each individual step to correct these generated spills by adjusting the control signal of the Mu2e simulator. The primary goal of our model is to bring all spills as close to a value of 1 as possible. To optimize this objective function while mitigating the influence of excessively high or low intensity spills, we implement an exponential moving average (EMA) to measure the deviation of the sequence from the desired value of 1. Additionally, we incorporate neuralized PID controller, encompassing proportional, integral, and derivative components in the state representation. By employing different random seeds, our simulator can generate a diverse range of spill types, providing a reflection of real-world scenarios.

We demonstrate superior performance by numerically benchmarking our methods with PID controller. Specifically, our experiments compares SDF performance on different seeds. Our results show that our methods consistently improve the SDF in 9.4% for random generalized spills and achieve 2.2% improvements compared to the PID controller.

The rest of this paper commences by a detailed description of our proposed method. Subsequently, we present numerical results to showcase the performance of our approach. Finally, we conclude this paper with a discussion of potential future directions and avenues for further research.

2 Methodology

In this section, we first introduce the machine learning's role in optimizing Fermilab's accelerator parameters. Then, we provide a detailed design of our RL-enhanced regularization system.

2.1 Problem Setup

While increasing beam intensity poses in Mu2e experiment its own challenges, maintaining beam size and minimizing beam losses — particles lost due to interactions with the beam vacuum pipe — are often the primary obstacles. To combat these, we approach the beam intensity regulation challenge, specifically controlling the spill regulation system, as a tracking control problem. The objective is to keep certain signals (spill intensity) close to specific reference values (approximately 1) by controlling the quadrupole currents in the regulation system. By doing so, we adjust the magnetic field, which subsequently adjusts the beam intensity throughout the delivery ring of the Mu2e experiments, as depicted in Figure 1 (c).

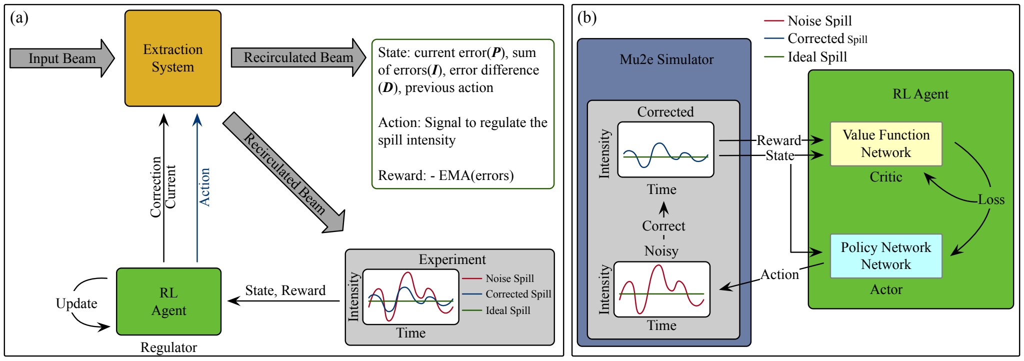

Figure 2: **(a)**: The Mu2e simulator initially generates the beam. It proceeds to adjust the spill and employs this adjusted spill to compute relevant information such as the state and reward. These pieces of information are instrumental in training the RL agent, which in turn offers new actions to refine the spill. **(b)**: The simulator refines (corrects) the spill derived from noisy data. It conveys the state and reward, calculated using the corrected spill, to update the value network responsible for evaluating the quality of the correction. Subsequently, the state and loss generated by the value network contribute to the adaptation of the policy network. The policy network, in response, generates new actions for spill regulation.

Objective. To handle the challenges posed by Mu2e experiment, we regulate the uniformity of the extracted spill by increasing its Spill Duty Factor (SDF),

$$\text{SDF} := 1/\left(1+\sigma_{\text{spill}}^2\right), \tag{2.1}$$

by regulating the extraction process. The ultimate goal for SRS in the Mu2e experiment is to achieve a SDF of 0.6 or higher, with an ideal spill having a constant spill rate value of 1 and an SDF of 1.

Mu2e Simulator. We employ a differentiable simulator to replicate the Mu2e experiments realistically. This simulator has the capability to produce spill intensity and compute associated data. It subsequently conveys this data to the RL agent, which aids in training the RL model. Once trained, the RL model transmits control signals back to the simulator, allowing it to regulate its spill based on these signals and provide the modified data to the RL agent. This process is shown in Figure 2 (a).

2.2 Proximal Policy Optimization (PPO) Controller for Spill Regulation System

Reward Function. Let x_t be the observation of one single spill signal at time step t and $\sigma = 1$ be the corresponding reference value. The reward function at t is defined as the exponential moving average

$$r_t = -\text{EMA}(t, \alpha), \alpha \in [0, 1], \text{ where } \text{EMA}(t, \alpha) = \alpha|x_t - \sigma| + (1-\alpha)\text{EMA}(t-1, \alpha). \tag{2.2}$$

EMA gives more weight to recent spills and less weight to older spills. This helps in reducing the impact of short-term fluctuations in the spills, making it easier to identify underlying trends.

PID Controller. A PID controller in discrete-time operation captures past details regarding tracking errors, their integrals, and derivatives within a linear control strategy. We denote the time series of spill signal at time t as $o_t = (x_0, x_1, \cdots, x_t)$. In formal terms, the discrete-time PID controller's policy, characterized by its parameters K_P, K_I, and K_D, is expressed as:

$$\pi^{\text{PID}}(o_t) = K_P\left(x_t - \sigma\right) + K_I \sum_{\tau=0}^{t}\left(x_\tau - \sigma\right) + K_D \frac{\left(x_t - \sigma\right) - \left(x_{t-1} - \sigma\right)}{\Delta t}, \tag{2.3}$$

where K^P, K^I, K^D are tuneable scalar coefficients and Δt is the discrete time interval.

Model: PPO with Neuralized PID Policy. Our model leverages the inductive bias of PID (Proportional-Integral-Derivative) controller, and propose a neuralized PID policy function. Specifically, we incorporate tracking errors, integrals, and derivatives as components of the state vector s_t. Furthermore, we employ a linear network to parametrize the standard PID controller (2.3), and use it as a part of our policy function. Therefore, the policy function of our model not only enables the extraction of external information (based on previous actions) but also includes the PID control signals K_P, K_I, and K_D as its learnable parameters. Remarkably, when learned effectively, our policy network outperforms the standard PID controller (2.3), making it a highly adaptable solution.

Our approach incorporates three key components: the PPO algorithm, the EMA reward function, and the neuralized PID controller. The PPO algorithm, a reinforcement learning technique, plays a central role in optimizing policy functions to enhance decision-making in sequential tasks. We specifically chose PPO to refine our reward function. Rather than relying on a single spill for reward computation,

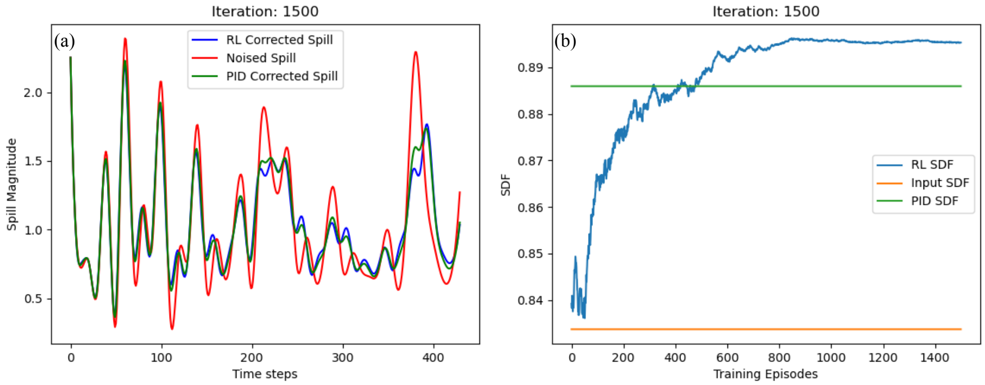

Figure 3: **(LHS)**: Comparison of Spill Intensity: The spill intensity corrected by RL is closer to 1 when compared to the PID-corrected spill. **(RHS)**: Comparison of SDF: After 600 training iterations, the SDF achieved by RL outperforms the SDF obtained through PID.

we employ EMA within the reward function. This approach allows us to both capture trends across a sequence of spills and mitigate fluctuations. Additionally, we integrate the neuralized PID bias into our policy network. In this setup, our policy network consists of a trainable PID controller and a linear projection of past actions. Let's denote the state at time step t as s_t, the action as a_t. We combine the PID policy π^{PID} and action policy π^{action} to formulate RL policies as π. The variable a_t represents the control signal used for regulating the spill. The state s_t encompasses both the previous action a_{t-1} and the time series of the spill signal o_t. As a result, the action at time t, a_t, can be represented as follows:

$$a_t = \pi(s_t) = \pi^{\text{PID}}(o_t) + \pi^{\text{action}}(a_{t-1}), \tag{2.4}$$

The learning process is shown in Figure 2 (b).

3 Experimental Studies

We validate our method by using the Mu2e differentialble simulator [Narayanan et al., 2021] as the RL environment. Our experiments involve the utilization of both the Mu2e simulator and our RL-enhanced spill regularization system.

Settings. We configure various hyperparameters for both the simulator and the RL agent in our experiment. Specifically, we configure the simulator to generate 430 spills per iteration, equivalent to 10 data points per millisecond within a 43 ms spill duration, aligning closely with realistic settings as detailed in [Narayanan et al., 2021]. In terms of reward calculation, we choose a value of $\alpha = 0.1$ for the EMA component. Regarding the RL agent, we employ the stable-baselines3 PPO model [Raffin et al., 2021]. The actor network is designed as a single linear layer, while the critic network took the form of a two-hidden layer (64×64) MLP network. The learning rate is set at 1×10^{-4}. Notably, despite initially setting the number of iterations to $100,000$, we manually terminated the process if convergence is achieved.

Results. In Figure 3, we examine the spill intensity in scenarios involving unregularized, PID-regularized, and RL-regularized setups. Figure 3 (a) demonstrates that the spill corrected by RL is closer to unity when compared to the PID-corrected spill. Furthermore, we provide a visual representation of the SDF plot during the training process. Figure 3 (b) illustrates that the SDF achieved through RL surpasses that of PID regulation after 600 episodes.

4 Conclusion

We present an innovative RL-enhanced spill control system, utilizing a neural PID controller as the policy function, to tackle beam regulation issues in Mu2e experiments. To simulate real-world spill control scenarios, we utilize a differentiable Mu2e simulator to create spills, improve spill adjustments, and establish reward signals. Furthermore, we harness an RL-based controller for fine-tuning control signals during the regularization process. Our experiments demonstrate the superior performance of our method over the PID-based model.

References

L. Bartoszek, E. Barnes, J. P. Miller, J. Mott, A. Palladino, J. Quirk, B. L. Roberts, J. Crnkovic, V. Polychronakos, V. Tishchenko, P. Yamin, C. h. Cheng, B. Echenard, K. Flood, D. G. Hitlin, J. H. Kim, T. S. Miyashita, F. C. Porter, M. Röhrken, J. Trevor, R. Y. Zhu, E. Heckmaier, T. I. Kang, G. Lim, W. Molzon, Z. You, A. M. Artikov, J. A. Budagov, Yu. I. Davydov, V. V. Glagolev, A. V. Simonenko, Z. U. Usubov, S. H. Oh, C. Wang, G. Ambrosio, N. Andreev, D. Arnold, M. Ball, R. H. Bernstein, A. Bianchi, K. Biery, R. Bossert, M. Bowden, J. Brandt, G. Brown, H. Brown, M. Buehler, M. Campbell, S. Cheban, M. Chen, J. Coghill, R. Coleman, C. Crowley, A. Deshpande, G. Deuerling, J. Dey, N. Dhanaraj, M. Dinnon, S. Dixon, B. Drendel, N. Eddy, R. Evans, D. Evbota, J. Fagan, S. Feher, B. Fellenz, H. Friedsam, G. Gallo, A. Gaponenko, M. Gardner, S. Gaugel, K. Genser, G. Ginther, H. Glass, D. Glenzinski, D. Hahn, S. Hansen, B. Hartsell, S. Hays, J. A. Hocker, E. Huedem, D. Huffman, A. Ibrahim, C. Johnstone, V. Kashikhin, V. V. Kashikhin, P. Kasper, T. Kiper, D. Knapp, K. Knoepfel, L. Kokoska, M. Kozlovsky, G. Krafczyk, M. Kramp, S. Krave, K. Krempetz, R. K. Kutschke, R. Kwarciany, T. Lackowski, M. J. Lamm, M. Larwill, F. Leavell, D. Leeb, A. Leveling, D. Lincoln, V. Logashenko, V. Lombardo, M. L. Lopes, A. Makulski, A. Martinez, D. McArthur, F. McConologue, L. Michelotti, N. Mokhov, J. Morgan, A. Mukherjee, P. Murat, V. Nagaslaev, D. V. Neuffer, T. Nicol, J. Niehoff, J. Nogiec, M. Olson, D. Orris, R. Ostojic, T. Page, C. Park, T. Peterson, R. Pilipenko, A. Pla-Dalmau, V. Poloubotko, M. Popovic, E. Prebys, P. Prieto, V. Pronskikh, D. Pushka, R. Rabehl, R. E. Ray, R. Rechenmacher, R. Rivera, W. Robotham, P. Rubinov, V. L. Rusu, V. Scarpine, W. Schappert, D. Schoo, A. Stefanik, D. Still, Z. Tang, N. Tanovic, M. Tartaglia, G. Tassotto, D. Tinsley, R. S. Tschirhart, G. Vogel, R. Wagner, R. Wands, M. Wang, S. Werkema, H. B. White Jr. au2, J. Whitmore, R. Wielgos, R. Woods, C. Worel, R. Zifko, P. Ciambrone, F. Colao, M. Cordelli, G. Corradi, E. Dane, S. Giovannella, F. Happacher, A. Luca, S. Miscetti, B. Ponzio, G. Pileggi, A. Saputi, I. Sarra, R. S. Soleti, V. Stomaci, M. Martini, P. Fabbricatore, S. Farinon, R. Musenich, D. Alexander, A. Daniel, A. Empl, E. V. Hungerford, K. Lau, G. D. Gollin, C. Huang, D. Roderick, B. Trundy, D. Na. Brown, D. Ding, Yu. G. Kolomensky, M. J. Lee, M. Cascella, F. Grancagnolo, F. Ignatov, A. Innocente, A. L'Erario, A. Miccoli, A. Maffezzoli, P. Mazzotta, G. Onorato, G. M. Piacentino, S. Rella, F. Rossetti, M. Spedicato, G. Tassielli, A. Taurino, G. Zavarise, R. Hooper, D. No. Brown, R. Djilkibaev, V. Matushko, C. Ankenbrandt, S. Boi, A. Dychkant, D. Hedin, Z. Hodge, V. Khalatian, R. Majewski, L. Martin, U. Okafor, N. Pohlman, R. S. Riddel, A. Shellito, A. L. de Gouvea, F. Cervelli, R. Carosi, S. Di Falco, S. Donati, T. Lomtadze, G. Pezzullo, L. Ristori, F. Spinella, M. Jones, M. D. Corcoran, J. Orduna, D. Rivera, R. Bennett, O. Caretta, T. Davenne, C. Densham, P. Loveridge, J. Odell, R. Bomgardner, E. C. Dukes, R. Ehrlich, M. Frank, S. Goadhouse, R. Group, E. Ho, H. Ma, Y. Oksuzian, J. Purvis, Y. Wu, D. W. Hertzog, P. Kammel, K. R. Lynch, and J. L. Popp. Mu2e technical design report, 2015.

Aakaash Narayanan, KJ Hazelwood, Michelle Ibrahim, Han Liu, Seda Memik, Vladimir Nagaslaev, Dennis Nicklaus, Peter Prieto, Brian Schupbach, Kiyomi Seiya, et al. Optimizing mu2e spill regulation system algorithms. Technical report, Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States), 2021.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Ali Zribi, Mohamed Chtourou, and Mohamed Djemel. A new pid neural network controller design for nonlinear processes. *Journal of Circuits, Systems and Computers*, 27(04):1850065, 2018.