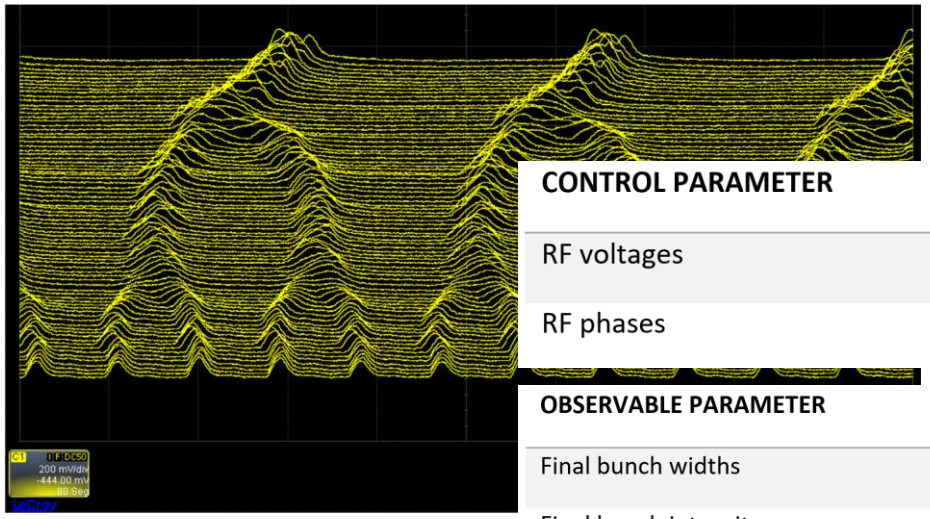
Apply Machine Learning to Improve Beam Polarization in the BNL Hadron Injectors

Yuan Gao

8/25/2023

@BrookhavenLab

Optimization of bunch merge in the Booster and AGS injectors



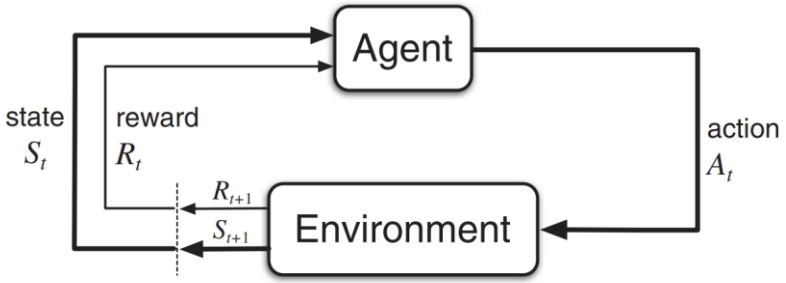
CONTROL PARAMETER	UNITS	NUMBER
RF voltages	V	(# harmonics * # controls points) ~= 20-30
RF phases	deg	# harmonics ~= 2-5

OBSERVABLE PARAMETER	UNITS	NUMBER
Final bunch widths	ns	1
Final bunch intensity	#	1
Bunch shape oscillation amplitudes	ns	1-3
Bunch center oscillation amplitudes	ns	1-3

- Longitudinal bunch merge or split is a useful and common technique in Booster and AGS injectors; Bunch charge can be reduced by splitting, reduce the final emittance;
- Accomplishing these RF manipulations without longitudinal emittance growth is challenging;
- Currently, this is done by experts observing the mountain-range displays of Wall Current Monitor (WCM), and optimizing various characteristics (widths, center -of-charge oscillations);
- Machine learning can be used for auto tuning and optimizing bunch profiles;
- Tunable variables will be RF voltages and phases, objective can be assessed by WCM;

Brookhaven National Laboratory

Reinforcement Learning (RL)



The RL Process: a loop of state, action, reward and next state

Source: Reinforcement Learning: An Introduction, Richard Sutton and Andrew G. Barto

RL is about learning the optimal behavior in an environment to obtain maximum reward.

- The Agent receives a **state S0** from the **Environment**;
- Based on that **state S0,** agent takes an **action A0**;
- Environment transitions to a **new state S1;**
- Environment gives some **reward R** to the agent;
- This RL loop outputs a sequence of **state, action and reward.**
- The goal of the agent is to maximize the expected cumulative reward.
- The cumulative rewards through the game is usually discounted by a factor gamma:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

The larger the gamma, the agents care more about the longer-term reward.

Brookhaven National Laboratory

Monte Carlo vs TD Learning

- Monte Carlo: Collecting the rewards **at the end of the episode** and then calculating the **maximum expected future reward.**

$$V(S_t) \leftarrow V(S_t) + \alpha\,[G_t - V(S_t)]$$

learning rate

Maximum expected future reward starting at that state

Former estimation of maximum expected future reward starting at that state

Discounted cumulative rewards

- Temporal Difference (TD) Learning: Learning at each time step, will not wait until the end of the episode to update **the maximum expected future reward estimation: it will update its value estimation V for the non-terminal states S_t occurring at that experience.**
- This method is called TD(0) or **one step TD (update the value function after any individual step).**

$$V(S_t) \leftarrow V(S_t) + \alpha\,[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Previous estimate

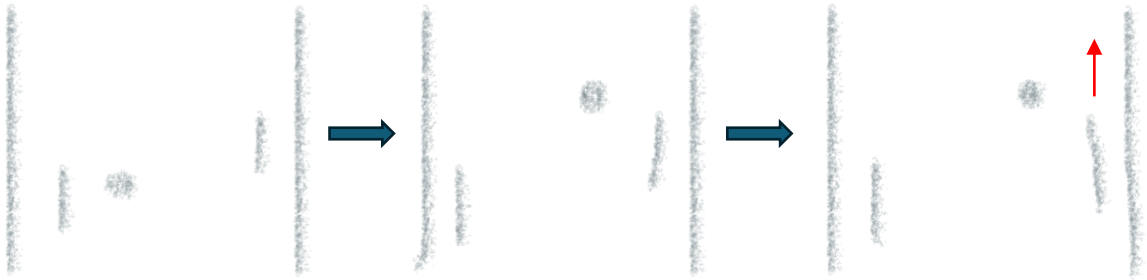
Reward t+1

Discounted value on the next step

TD Target

Brookhaven National Laboratory

Two main approaches for solving RL problems



The policy π will tell the agent which action to take at a given state

- The policy π: the agent's brain
  It's the function that tells the agent what **action to take at a given state.** So it **defines the agent's behavior** at a given time.
- Our goal is to find the optimal policy π*, the policy that **maximizes expected return** when the agent acts according to it. We find this π* **through training.**

There are two main approaches to find the optimal policy π*:

- **Directly,** by teaching the agent to learn which **action to take,** given the current state: **Policy-Based Methods.**
- Indirectly, **teach the agent to learn which state is more valuable** and then take the action that **leads to the more valuable states**: Value-Based Methods.

Policy-based Learning

- In Policy-Based methods, **we learn a policy function directly.**
- This function will define a mapping from each state to the best corresponding action. Alternatively, it could define **a probability distribution over the set of possible actions at that state.**

  Deterministic: a policy at a given state **will always return the same action**.

$$a = \pi(s)$$

  Stochastic: outputs **a probability distribution over actions**.

$$\pi(a|s) = P[A|s]$$

Probability Distribution over the set of actions given the state

Policy-based Learning

- In Policy-Based methods, **we learn a policy function directly.**
- This function will define a mapping from each state to the best corresponding action. Alternatively, it could define **a probability distribution over the set of possible actions at that state.**

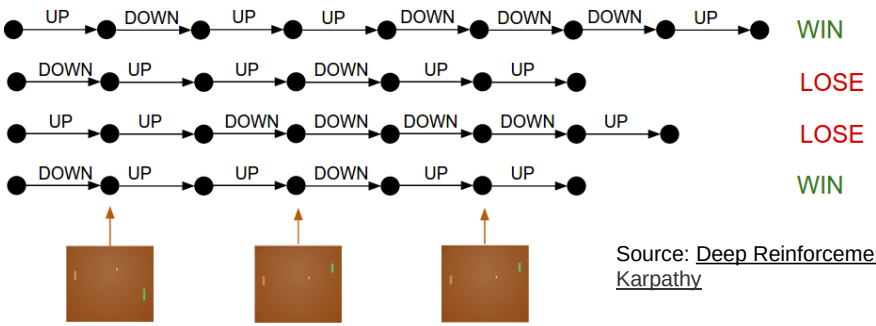
  Deterministic: a policy at a given state **will always return the same action**.

$$a = \pi(s)$$

  Stochastic: outputs **a probability distribution over actions**.

$$\pi(a|s) = P[A|s]$$

Probability Distribution over the set of actions given the state



Source: Deep Reinforcement Learning: Pong from Pixels, Andrej Karpathy

In a game of Pong. Each black circle is some game state, and each arrow is a transition. In this case we won 2 games and lost 2 games. With Policy Gradients we would take the two games we won and slightly encourage every single action we made in that episode. Conversely, we would also take the two games we lost and slightly discourage every single action we made in that episode.

Value-based Learning

- In value-based methods, instead of learning a policy function, we **learn a value function** that tells us the value of a given state**.**
- The value of a state s under a given policy π is the **expected discounted return** the agent can get if it **starts in that state and then follows the policy π.**

$$V_{\pi}(s) = \mathbb{E}_{\pi}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$$

Value function — Expected discounted future return — Starting at state s

- The agent will use this value function to select actions at each step:
  Exploration is **finding more information about the environment**, e.g., pick random actions.
  Exploitation is **exploiting known information to maximize the reward**, e.g., pick the action that maximize the value function: $a = max_a V_{\pi}(s)$.

Brookhaven National Laboratory

Value-based Learning

- In value-based methods, instead of learning a policy function, we **learn a value function** that tells us the value of a given state**.**
- The value of a state s under a given policy π is the **expected discounted return** the agent can get if it **starts in that state and then follows the policy π.**

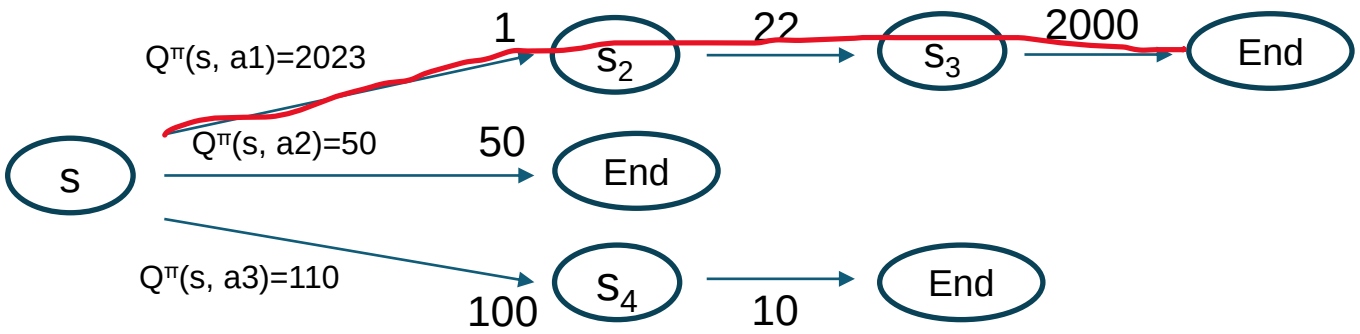
$$V_{\pi}(s) = \mathbb{E}_{\pi}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$$

Value function — Expected discounted future return — Starting at state s

- The agent will use this value function to select actions at each step:

  Exploration is **finding more information about the environment**, e.g., pick random actions.

  Exploitation is **exploiting known information to maximize the reward**, e.g., pick the action that maximize the value function: $a = max_a V_{\pi}(s)$.



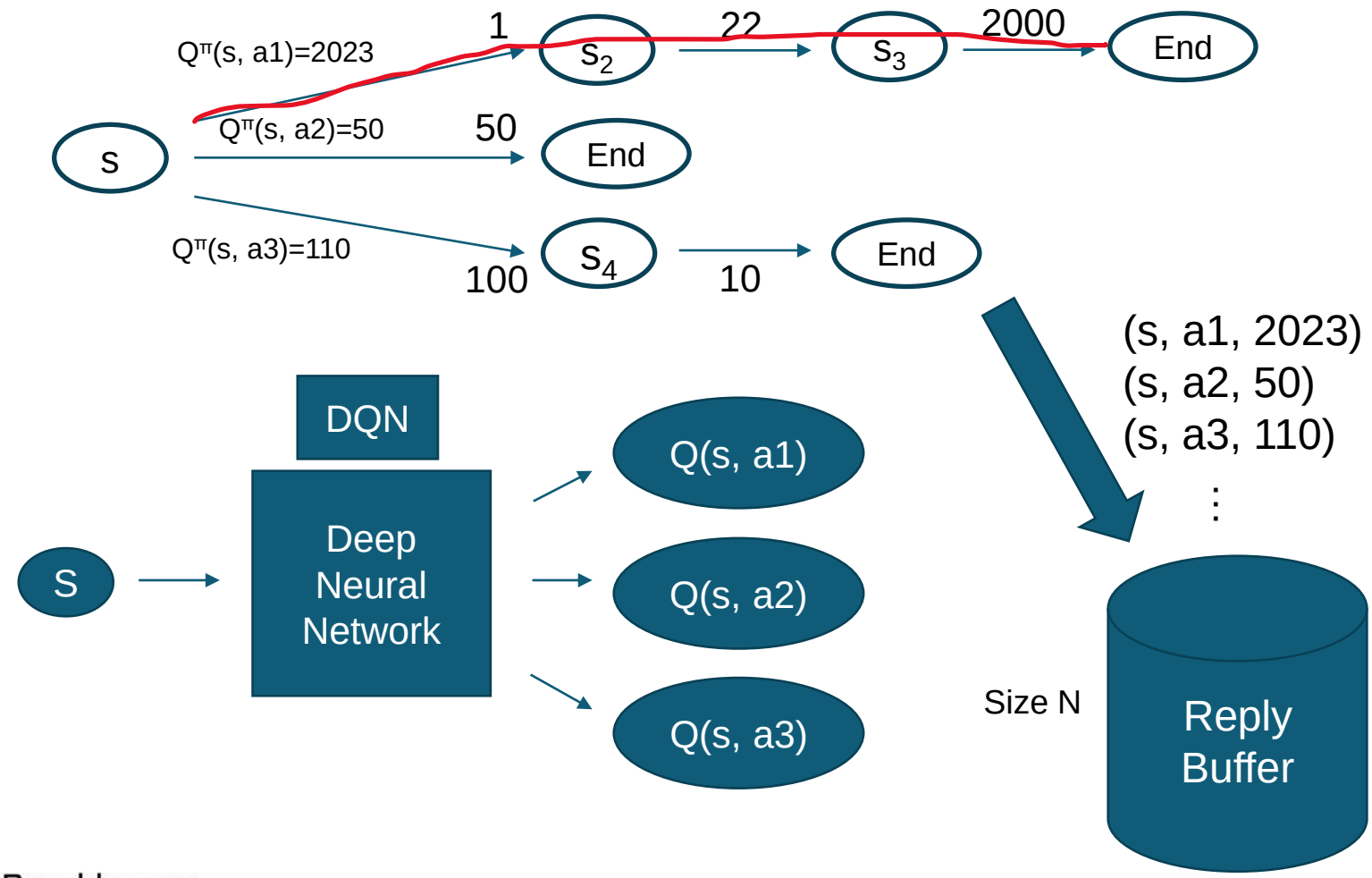
Brookhaven National Laboratory

Deep Reinforcement Learning

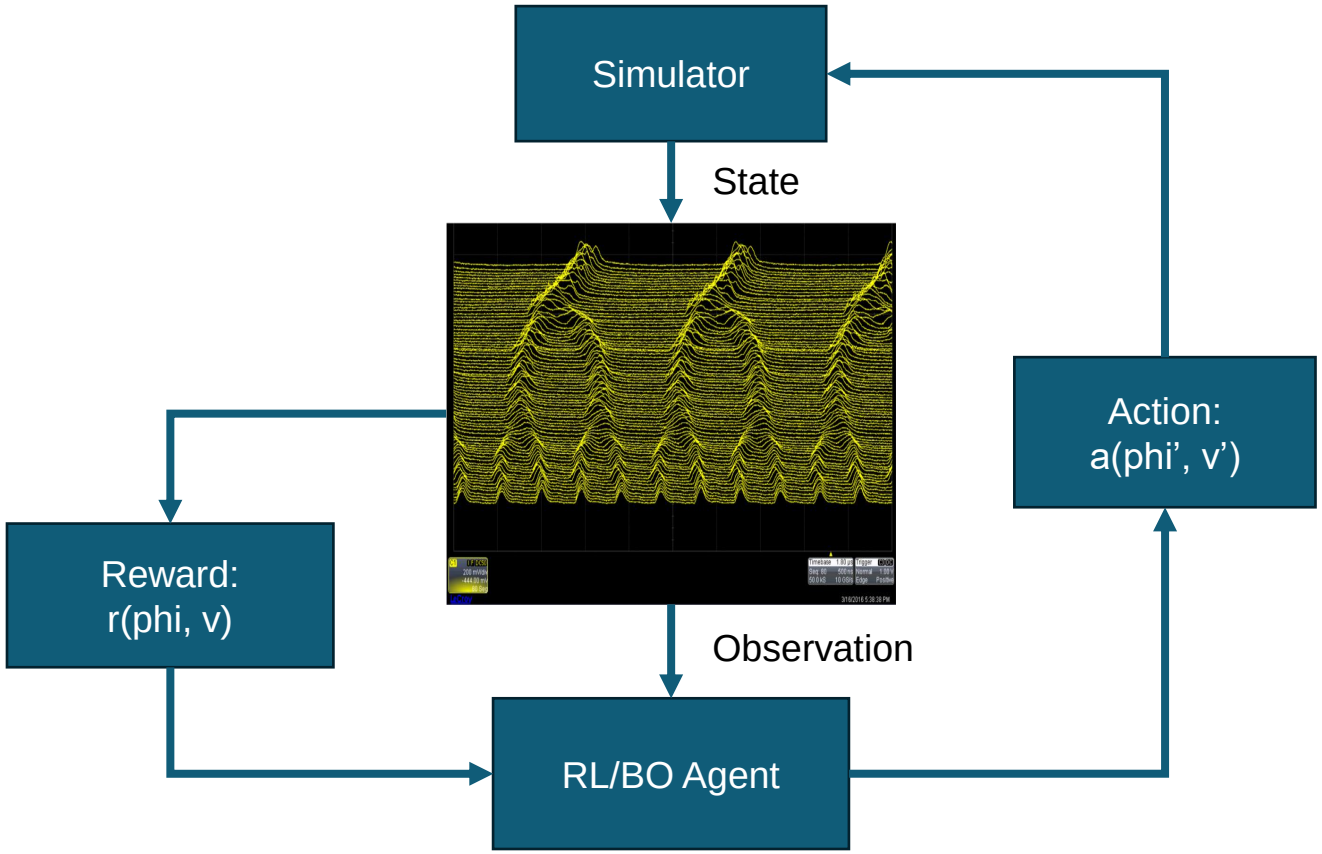
- Deep Reinforcement Learning introduces deep neural networks to solve Reinforcement Learning problems — hence the name “deep”.

Deep Reinforcement Learning

- Deep Reinforcement Learning introduces deep neural networks to solve Reinforcement Learning problems — hence the name “deep”.

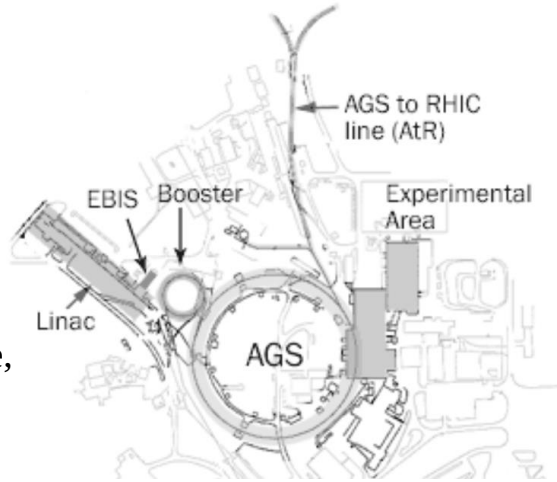


Optimization Workflow for Bunch Merge in AGS and Booster



- A simulator is used for initial experiment, which takes RF voltages and phases as the inputs, and outputs the bunch profile.
- A RL/BO agent is trained to tune RF voltages and phases based on observations from the environment.

Booster Injection Optimization



- Minimize emittance growth:

  <u>The transverse emittances</u>:
  - Mismatch between the trajectory from the Linac and the equilibrium orbit trajectory in Booster, achieved by tuning dipole steering in the transfer line and the Booster.
  - Optical mismatching between the transfer line and the Booster lattice, achieved by tuning quadrupole strength in the transfer line and the Booster.

  <u>Due to scattering at the foil location</u>:
  - Linac outputs H- pulses at 1.1 GeV, duration around 300 us;
  - Revolution frequency in Booster is 1 us, proton will hit the foil many times, causing scattering and emittance growth;
  - Can be minimized by adjusting Booster optics such that the angular spread is maximal at the foil location;

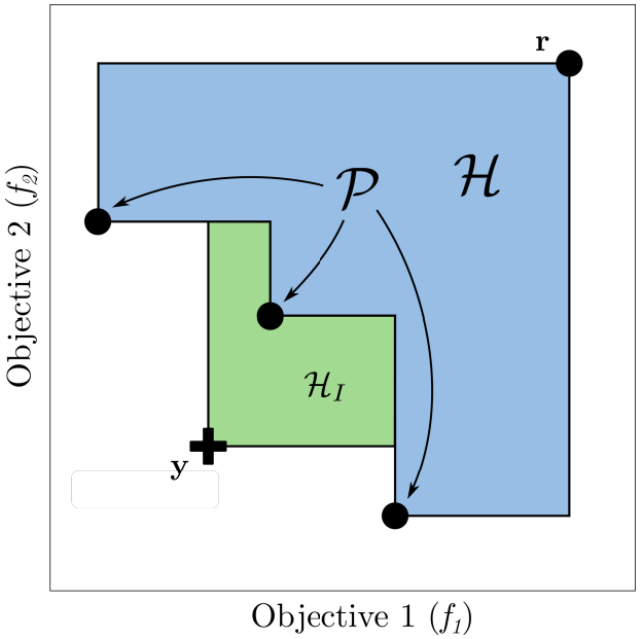
  <u>The longitudinal emittances</u>:
  - Mismatch between the output energy of the Linac beam and the dipole field and frequency of the Booster RF;

- The goal is to **maximize the intensity and minimize emittance growth**;
- **Challenge**: Direct access to many of the relevant beam measurements is hard and delayed;
- Beam polarization is not directly impacted, but is strongly affected by emittance produced in the Booster and transmitted through the AGS to RHIC;

Multi-Objective Optimization (MOO)

- Combine multiple objectives into one using weights;
- Pareto Front: PF represents a set of non-dominated solutions, where no other solution can improve one objective without degrading at least one other objective. These solutions are considered Pareto-optimal because they form the best compromise among the multiple conflicting objectives.
- Evolutionary algorithms, NSGA-II …, easy to implement, but very inefficient;
- Multi-objective BO, Hypervolume; BO with constraints;
- Game theory, Nash equilibrium, correlated equilibrium;
- A major practical difficulty for performing multi-objective optimization during online accelerator operations is the measurement of multiple objectives simultaneously. Most currently available accelerator diagnostics are destructive in nature, meaning that measuring multiple potential objectives cannot be done at the same time.

The Pareto front hypervolume H (shown in blue) is the axis-aligned volume enclosed by the Pareto front and a reference point r.
Making a new observation y, that dominates over points in the current Pareto front, leads to an increase in hypervolume (shown in green), referred to as the hypervolume improvement HI .



Source: https://arxiv.org/pdf/2010.09824.pdf

Brookhaven National Laboratory

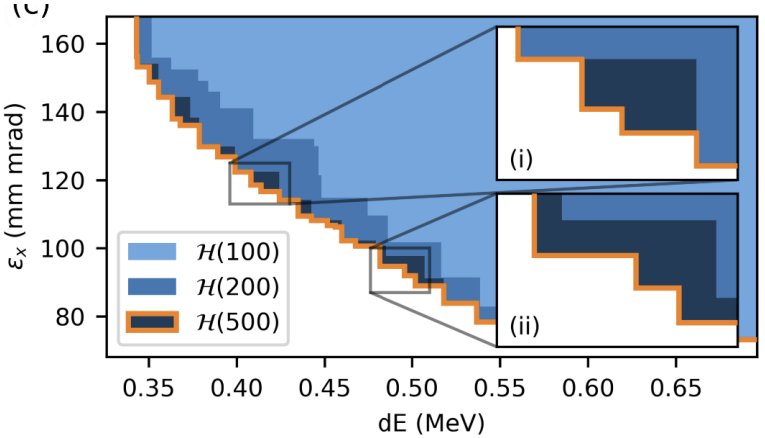
Multi-Objective Bayesian Optimization (MOBO)

- The Expected Hypervolume Improvement (EHVI) acquisition function uses the notion of an increase in PF hypervolume to select points in parameter space.
- Starting with a PF, EHVI predicts the average expected increase in hypervolume as a function of optimization parameters using GP models for each objective.
- As a result, BO using EHVI will select points that are more likely to maximally increase the hypervolume than other algorithms, whereas genetic algorithms select points only based on their optimality.
- When applied to identifying the PF of the AWA photoinjector containing 7 objectives (beam sizes, beam emittances, and energy spread), EHVI was able to converge to a maximum hypervolume several orders of magnitude faster than evolutionary algorithms.

EHVI

Pros: Better optimality, can be used in serial optimization contexts where objectives cannot be evaluated in parallel;

Cons: Computational expensive, scales exponentially with the number of objective functions;

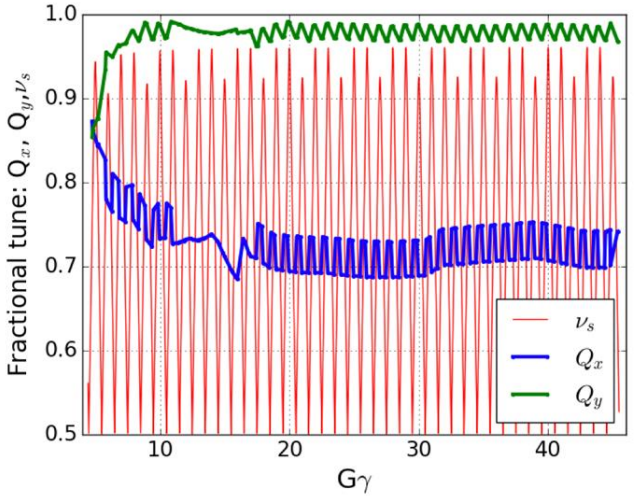


Reference: “Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization”, https://doi.org/10.48550/arXiv.2006.05078

“Multiobjective Bayesian optimization for online accelerator tuning”, Ryan Roussel, Adi Hanuka, and Auralee Edelen

Minimizing Depolarizing Resonance in the AGS

- At the present time the AGS synchrotron is equipped with two partial helices (snakes) which totally eliminate both the imperfection and the vertical intrinsic spin resonances, yielding a 65% final polarization of the proton beam.
- However, the two snakes fall short in eliminating the horizontal spin resonances, which occur through the interaction of the beam's betatron oscillations with the magnetic field of the partial helices.
- To overcome the horizontal spin resonances the method of "jump Quads" is currently being used and the polarization of the beam increases to 70%.
- The initial beam polarization at the exit of the 200 MeV Linac is measured to be 80%. The 10% loss of the polarization is due to the horizontal spin resonances.
- To further increase the polarization by totally eliminating the horizontal spin resonances, a skew quad method is proposed.



What is a Siberian Snake? A helical chain of dipole magnets (or simply a solenoid at low energies) that rotates the spin of a particle by 180 degree about an axis $m\rightarrow$, while it does not disturb the beam orbit outside of the snake.

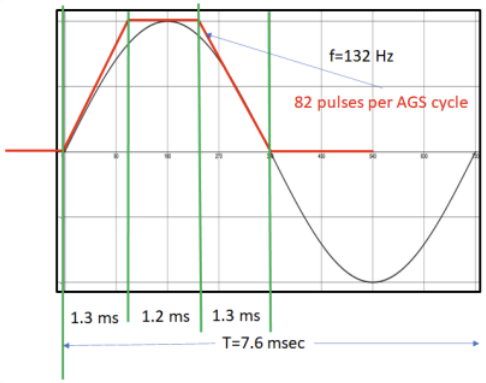
Source: https://doi.org/10.3390/sym13030398

Reference: *"A skew quadrupole for the AGS to minimize the polarization losses of the polarized beams"*, https://www.osti.gov/servlets/purl/1854096

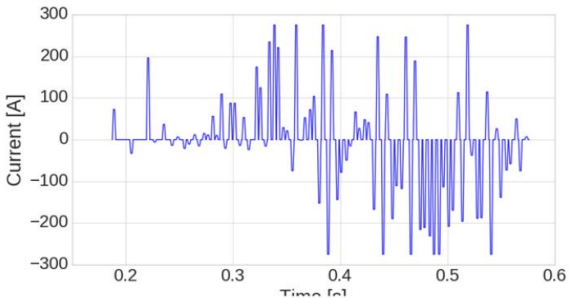
"Using betatron coupling to suppress horizontal intrinsic spin resonances driven by partial snakes", V. Schoefer

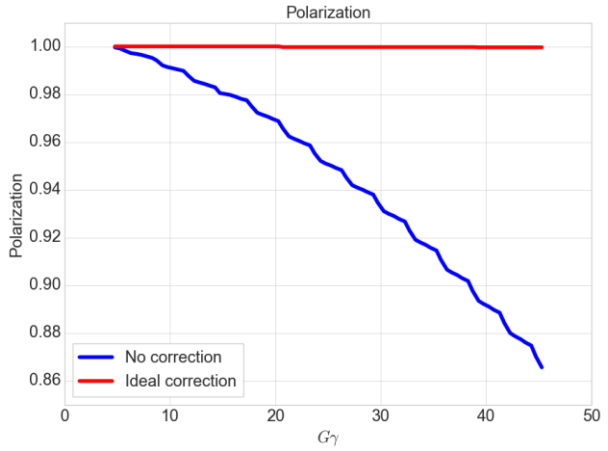
Skew Quad Method

- The skew quadrupole method is based on the linear coupling introduced by the skew quadrupoles which can excites horizontal spin resonances that cancel the ones caused by the partial helices.
- In a model evaluation, this method eliminates nearly all depolarization, even without tune jumps. This correction method relies on accurate resonance timing and on inducing the correct phase and amplitude of the compensating coupling resonance.
- **Challenge**: There are 15 independent skew quads to correct 82 horizontal resonances, each with a unique amplitude and phase. A single 5-minute polarization measurement yields polarization at the 2% precision level, brute-force scanning of parameters resonance-by-resonance is in general not possible.



Skew Quadrupole Current





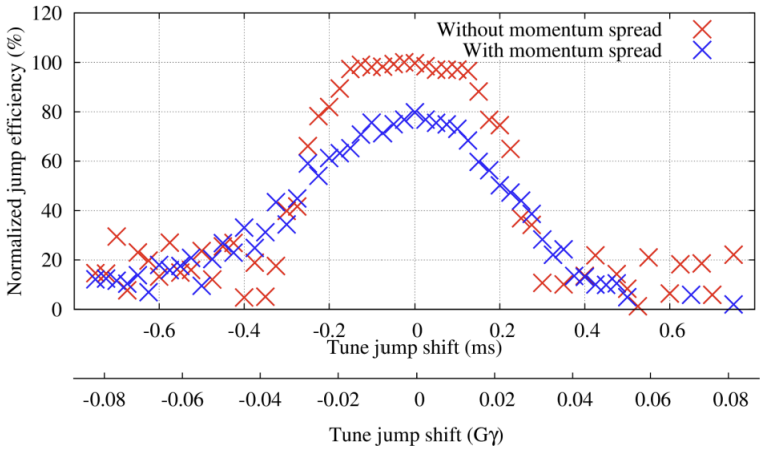
Skew Quad Model Evaluation

Energy Measurement Calibration

- Spin depolarizing resonances occur at very specific energies, the time at which the beam energy reaches these spin resonance conditions determines the time at which any compensation efforts must take place, often within very tight tolerances (of order 100 microseconds). Thus, determining the beam energy as a function of time is therefore crucial for improving beam polarization.

- A mistiming of $G_\gamma = \pm 0.02$ would cause measurable deterioration of the polarization.

- There are two conventional energy measurement methods



- Normalized tune jump efficiency as a function of the tune jump shift with or without momentum spread.
- The second axis gives the equivalent shift in energy at the acceleration rate of d[Gγ]/dt = 110 s−1 .

Reference: *"Energy Calibration and Tune Jumps Efficiency in the PP AGS", Y. Dutheil*, etc.

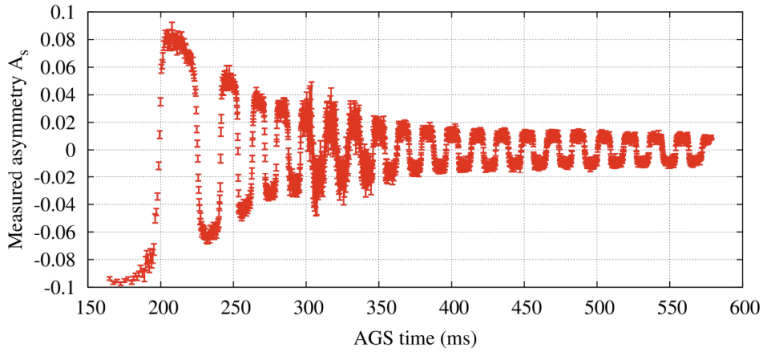
Energy Measurement Calibration

- Method 1 uses the measured RF frequency f and average radial shift of the beam dR:
- Method 2 uses the measured field (Binj + Bclock/Cscal) and the average radius:

$$G\gamma = G\frac{1}{\sqrt{1-\frac{1}{c^2}\left(\frac{f}{h}\right)^2(2\pi)^2(R_0+dR)^2}}$$

$$G\gamma = G\sqrt{\left[\frac{(1+\gamma_{tr}^2\, dR/R_0)\rho_0 c\left(B_{\text{inj}}+B_{\text{clock}}/C_{\text{scal}}\right)}{M_0}\right]^2+1}$$

- The parameters in red (f , dR and Bclock) are measured quantities while the blue ones are machine parameters that can be adjusted, and the black are fixed physical constants.
- The two formulae result in different calculations of the energy as a function of time. The result is cross calibrated: the machine parameters in both equations are adjusted manually until the difference between the two energy calculations is minimized along the ramp.
- The partial snakes of the AGS cause the spin of the protons to flip across every integer in Gγ. It was proposed to use the spin flip measured by the polarimeter to accurately determine the crossing time of every integer Gγ during the ramp.
- The **optimization goal** is to combine the three energy measurements into one using uncertainty minimization.



Brookhaven National Laboratory

Discussions