ML for luminosity maximization

(IP8 optics tunning with crossing angle and short vertex)

X. Gu, G. Robert-Demolaize, Y. Hao

June 14, 2023

Outline:

1) Plan
2) APEX Result
3) GPTune

1) Plan and Procedure

1) Beam Type (proton, Au, …): Au
2) Beam Energy (injection , ramp, store, or at ? GeV): injection
3) RHIC Rings to Be Used: Blue
4) Ramp Name (if a special ramp to be used): Au23-100GeV-e0::injection
5) Buch Filling Pattern: 6 bunches
6) Bunch Intensity: 1.2e10
7) Other Beam Conditions (re-bucketing, cooling, etc.): NO
8) Regular Machine Settings to Be Altered (such as hardware, pet page settings, etc.): NO

1) measure turn-by-turn optics as a base line
2) change s* = 0.5 m at IP12

2) run madx file to create IP12 trim strength

3) inject 6 bunches

4) sent the strength to the rampeditor

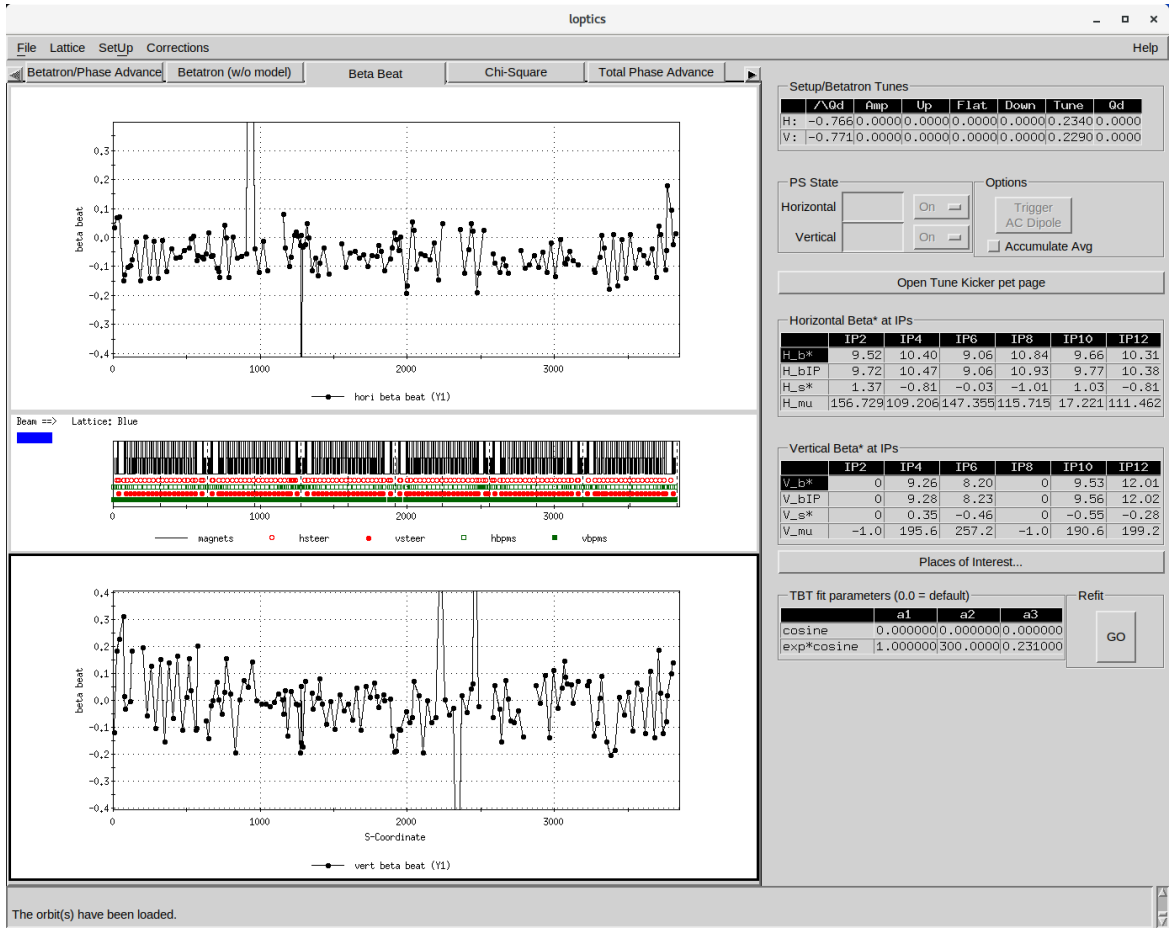
5) activate it

6) check beam loss

7) measure turn-by-turn optics

8) offline analysis.

1) APEX



Base line: s*x = -0.81m, s*y = 0.28m

Move s*x -> 0.5 m: s*x = -1.30m, s*y = -0.48
Move s*x -> -0.5m: s*x = -1.09m, s*y = -0.40

Back to base line1: s*x = -1.62m, s*y = -0.28m
Back to base line2: s*x = -0.89m, s*y = -0.20m

Comments:

1. Tested and improved the revised version of s* changing script. Works.

2. Will test it s* changing and measurement with store.

1) GPTune Script testing

```
acnuser06 192>ipython3 demo_APEX_Settings.py
machine: mymachine processor: myprocessor num_nodes: 1 num_cores: 2
PyGMO module cannot be loaded properly. Use SciPy (SearchSciPy) instead.
{'kwargs': {}, 'lite_mode': True, 'RCI_mode': False, 'mpi_comm': None, 'distributed_memory_parallelism': False, 'shared_memory_parallelism': Fa
_parallelism': False, 'verbose': False, 'oversubscribe': False, 'objective_evaluation_parallelism': False, 'objective_multisample_processes': N
threads': None, 'objective_nprocmax': None, 'objective_nospawn': True, 'sample_class': 'SampleLHSMDU', 'sample_algo': 'LHS-MDU', 'sample_max_it
dom_seed': 0, 'model_class': 'Model_GPy_LCM', 'model_kern': 'RBF', 'model_output_constraint': 'LargeNum', 'model_input_separation': False, 'mod
_threads': 1, 'model_processes': 1, 'model_groups': 1, 'model_restarts': 1, 'model_restart_processes': 1, 'model_restart_threads': 1, 'model_ma
er': 0.001, 'model_latent': None, 'model_sparse': False, 'model_inducing': None, 'model_layers': 2, 'model_max_jitter_try': 100, 'model_random_
earchSciPy', 'search_threads': 1, 'search_processes': 1, 'search_multitask_threads': 1, 'search_multitask_processes': 1, 'search_algo': 'dual_a
read_island', 'search_pop_size': 100, 'search_gen': 5, 'search_evolve': 10, 'search_max_iters': 10, 'search_more_samples': 1, 'search_random_se
'TLA_ensemble_exploration_rate': 0, 'regression_logging': False, 'regression_log_name': 'models_weights.log', 'budget_min': 0.1, 'budget_max':
ty_map': None, 'N_PILOT_CGP': 20, 'N_SEQUENTIAL_CGP': 20, 'RND_SEED_CGP': 1, 'EXAMPLE_NAME_CGP': 'obj_name_dummy', 'METHOD_CGP': 'FREQUENTIST',
CSAMPLES_CGP': 500, 'N_INFERENCE_CGP': 500, 'EXPLORATION_RATE_CGP': 1.0, 'NO_CLUSTER_CGP': False, 'N_NEIGHBORS_CGP': 3, 'CLUSTER_METHOD_CGP': '
 'ACQUISITION_CGP': 'EI', 'BIGVAL_CGP': 1000000000000.0, 'selection_criterion_hybrid': 'custom', 'policy_hybrid': 'UCTS', 'exploration_probabil
hybrid': 20, 'n_pilot_hybrid': 10, 'n_find_leaf_hybrid': 1, 'acquisition_GP_hybrid': 'GP-EI', 'random_seed_hybrid': 1, 'bigval_hybrid': 1000000
GPTune History Database Init
[HistoryDB] Create a JSON file at ./gptune.db/GPTune-Demo.json



------Starting MLA with 1 tasks and 20 samples each
[HistoryDB] Found a history database file
no history data has been loaded
NS1:  10
changing sstar now for s* -0.04423
1.51969071e-05
-5.448927355e-05
0
0
-8.026894259e-06
-7.9164283e-05
0
4.016089089e-05
4.172409161e-05
0
-5.247396011e-06
4.155524706e-05
0
0
0
O.K to go! Wait for activate! press <ENTER> to continue
sleep for 20 seconds.
Luminosity =  -2971397006
changing sstar now for s* -0.172287
5.805133606e-05
-0.000209362467
0
0
-3.105954904e-05
-0.0003048807651
0
0.0001574407891
0.0001593498243
0
-2.234451548e-05
0.0001634186074
0
0
0
O.K to go! Wait for activate! press <ENTER> to continue^C----------------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/cfs/ad/owl/xgu/git/GPTune/examples/GPTune-Demo/demo_APEX_Settings.py in <module>
    291
```

```python
def changesstar(x):
    ########################################
    command = "cp ./deltas_backup.dat ./deltas.dat"
    subprocess.getstatusoutput(command)

    with open('./deltas.dat', 'r') as file:
        filedata = file.read()
    filedata = filedata.replace('$S*x$', str(x))
    filedata = filedata.replace('$S*y$', str(0.0))
    with open('./deltas.dat', 'w') as file:
        file.write(filedata)

    command = "madx IR12_Injection > output.dat"
    subprocess.getstatusoutput(command)

    filename = 'IP12knob_Injection.dat'
    with open(filename, 'r') as file:
        data = []
        for line in file:
            row = line.split()[-2]
            data.append(row)
            print(row)
    knobs =  [float(value) for value in data]
    #print(knobs)

    if max(knobs) >= 1e-3 or min(knobs) <= -1e-3:
        print("STOP: Absolute value", "is greater than 1e-3!")
        command = 'rm SendTrim.IP12_Injection'
        subprocess.getstatusoutput(command)

    else:
        # the command for sent the current
        input('O.K to go! Wait for activate! press <ENTER> to continue')


def objectives(point):

    x = point['x']
    print('changing sstar now for s*', x)
    changesstar(x)

    print('sleep for 20 seconds.')
    time.sleep(20)
    Lumi = -list(adoif.get(('sisScaler.8b-scraper.C.11','valueM')).values())[0]
    print('Luminosity = ',Lumi)

    return [Lumi]


def main():

    import matplotlib.pyplot as plt
    global nodes
    global cores
```

1)

Comments:

1. Tested and improved the revised version of s* changing script. Works.

2. Will test it s* changing and measurement with store.

3. Measure three times

4. Fix the coupling

1) Motivation--RHIC luminosity Optimization

$$L = \frac{N_1 N_2 f H}{2\pi \sqrt{\sigma_{x1}^2 + \sigma_{x2}^2}\sqrt{\sigma_{y2}^2 + \sigma_{y2}^2}}$$

- Global Parameters:

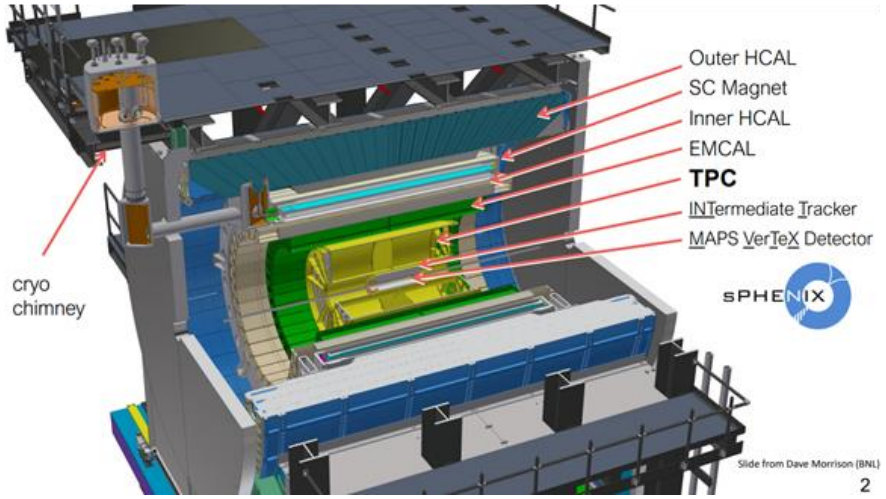
1. Orbit (Dipole)
2. Tune (Quadrupole),
3. Chromaticity (Sextuple)
4. Octupole

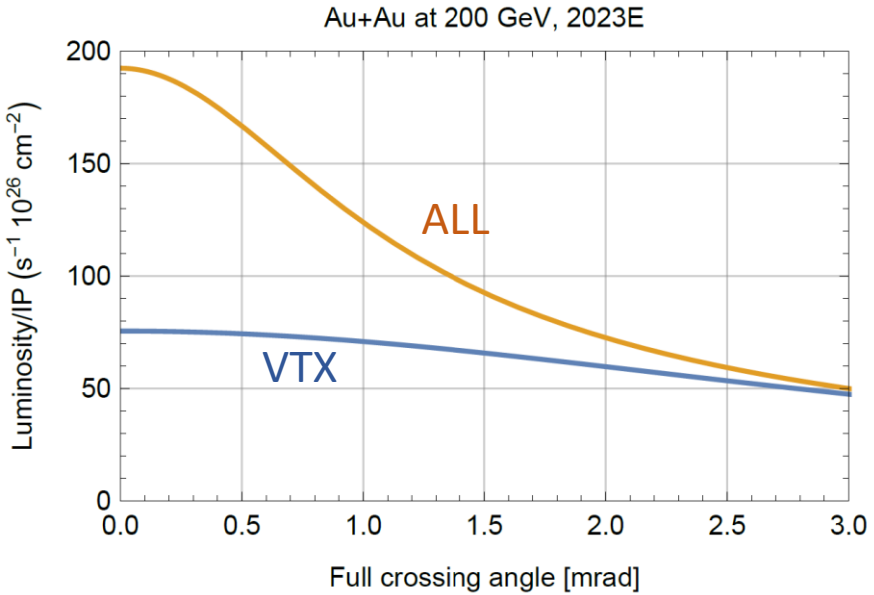
- Local (IR8) Parameters:

1. Beta*
2. S* (more sensitive than head on)
3. Transverse offset

- Other Parameters:

1. RF Voltage
2. Collimator Position





Courtesy of W. Fischer

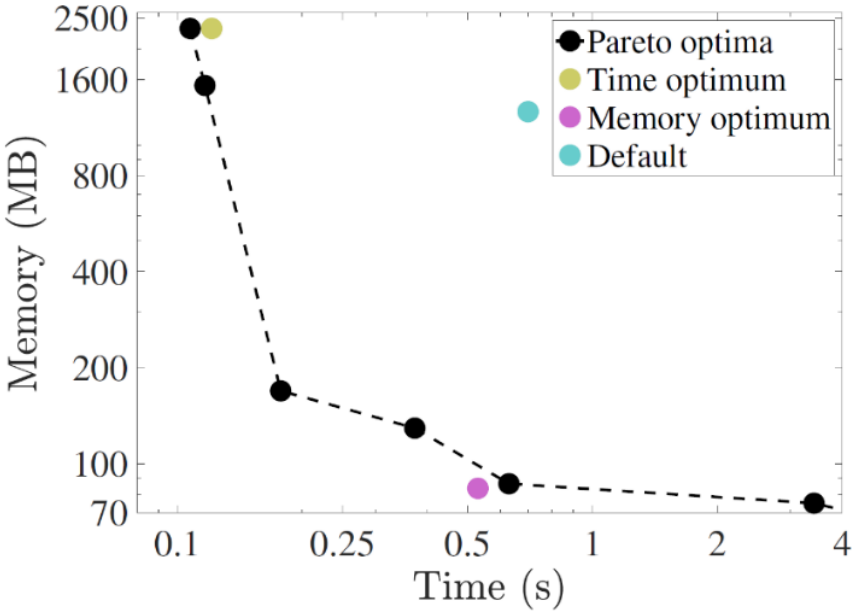
- sPHENIX:

1. VTX (+/-10 cm)
2. Crossing angle (2mrad)
3. S/N - Background

2) Bayesian optimization at LBNL GPTune

Several features of GPTune (BLNL) are very useful for HPC simulation codes, including:

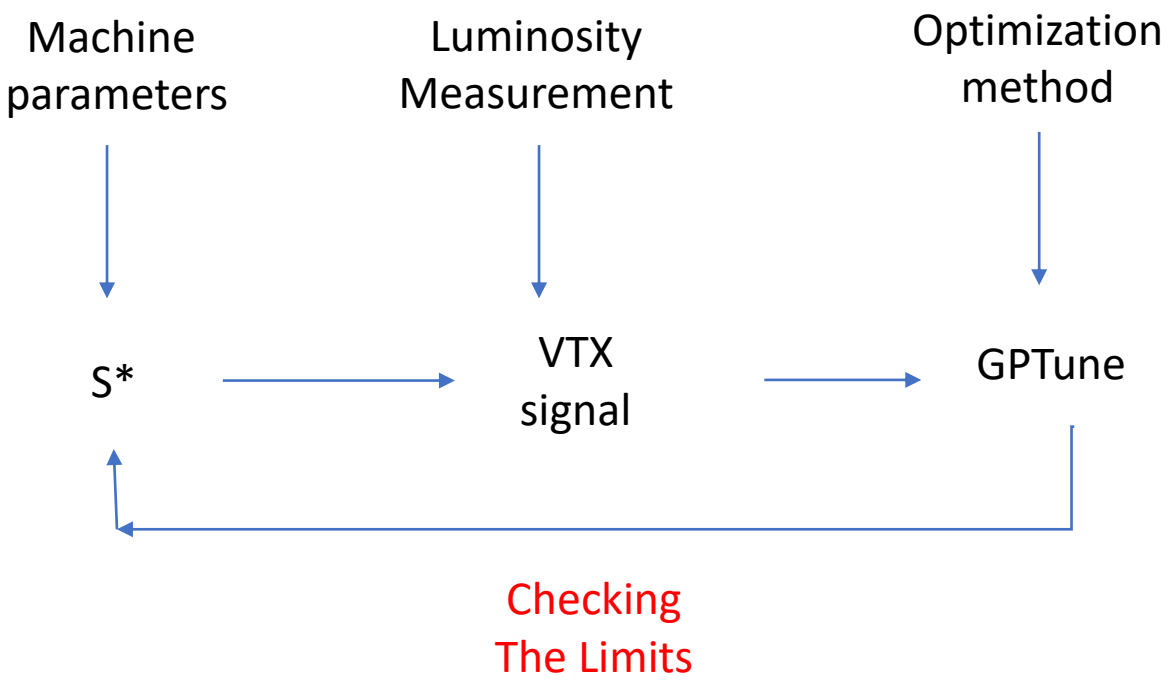
(1) relies on dynamic process management for running applications with varying core counts and GPUs
(2) can incorporate coarse performance models to improve the surrogate model
(3) allows multi-objective tuning such as tuning a hybrid of computation, memory and communication
(4) allows multi-fidelity tuning to better utilize the limited resource budget
(5) supports checkpoints and reuse of historical performance database.



Application:
Conduct parameter optimization for several HPC codes. The most notable result is for the multiscale production-level full-blown simulation codes, M3D-C1 and NIMROD that are used in the fusion Tokamak design.

https://github.com/mkturkcan/GPTune
https://nimrodteam.org/meetings/team_mtg_5_21/nimrod_meeting_YangLiu.pdf

4) Online Luminosity Optimization



S* and beta* changing scripts: ready for testing:

1. change the target s*, beta* within ‘deltas.dat’ file;
2. run ‘madx job.madx_Au16-e0::store’ command, will get ‘IP8knob.dat’ file;
3. run ‘CreateSend.IP8’ command, will get ‘SendTrim.IP8’ file;
4. run ‘SendTrim.IP8’ command.

Guillaume Robert-Demolaize.

GPTune is ready for test with scripts:

1. Installed and tested
2. Did optimization with Eq. as an input

Vertex: contacted with sPHINEX people:

1. Did it with PHENIX before
2. Send the vertex data several ways.

4) Preliminary Plan

1. First, we will test the scripts without beam (with magnet?, injection)

2. Then, we can test them at other IP (10, or 6) with one or 6 bunches, head on and then with 2mrad angle. And find the s* operation limit manually. (injection)

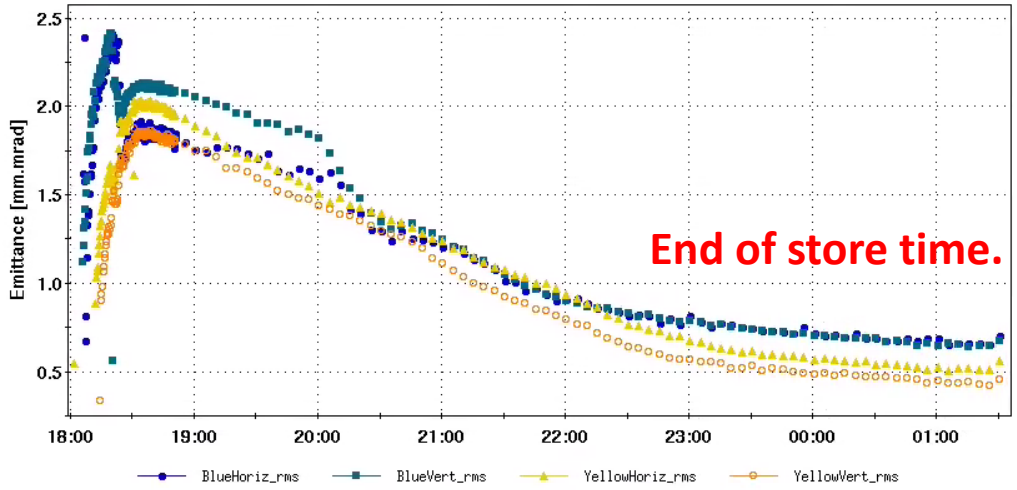
3. Then, repeat above procedure for IP8 without detector, 1 or 6 bunches. Will ask the permission for this.

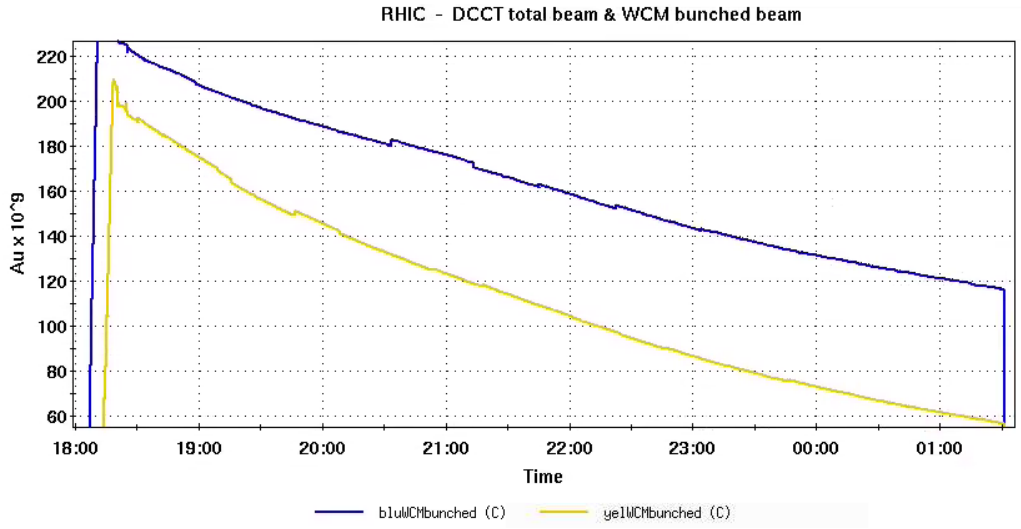
4. Test the scripts and GPTune with detector ON;

5. Finally, we can optimize the Vertx signal.

5) Challenges

- Changing the Dispersion at IP
- Changing the optics around the ring
- The beam emittance is changing during the store





5) Resource and Collaboration

- BNL: CAD support, Run coordinator, control, operator, RF and AP group
- MSU: Yue Hao, Will Fung
- LBNL: Ji Qiang, Sherry Li, Yi-Kai Kan

Backup Slides