Elettra
Sincrotrone
Trieste

Automation, performance optimisation & ML @ Elettra

G. Gaio

Outline

✓ Reinforcement Learning in a Free Electron Laser

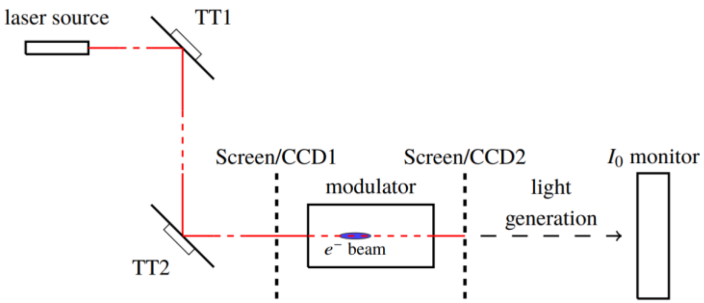
✓ Toward an accelerator autopilot

Reinforcement Learning in FERMI FEL optimization

✓ **Study goal:**
Apply Reinforcement Learning to automatically overlap the seed laser with the electron beam optimizing the radiation intensity

✓ **Seed Laser alignment system:**

- 2 planar Tip-Tilt mirrors (TTs) paired with 2 piezo-motors (hor - ver)
- 2 screens based on Charged-Coupled Devices (CCDs)

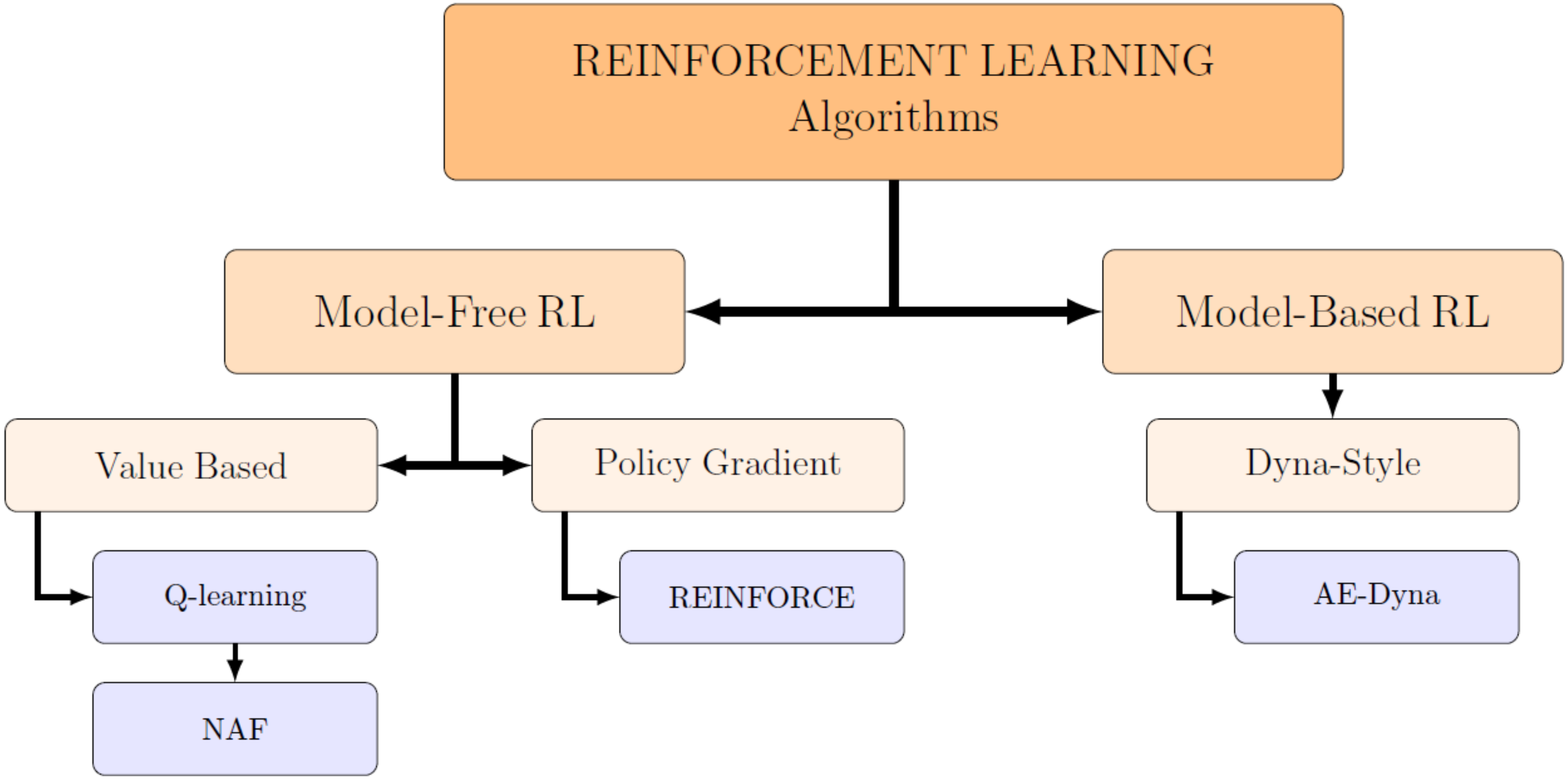


✓ **Final output:**
Intensity acquired by I_0 monitor

Optimization of 4 variables

Niky Bruchon, PhD of University of Trieste
https://arts.units.it/retrieve/handle/11368/2982117/362563/PhD_Thesis_Final_NikyBruchon.pdf

Reinforcement Learning algorithms applied on FERMI



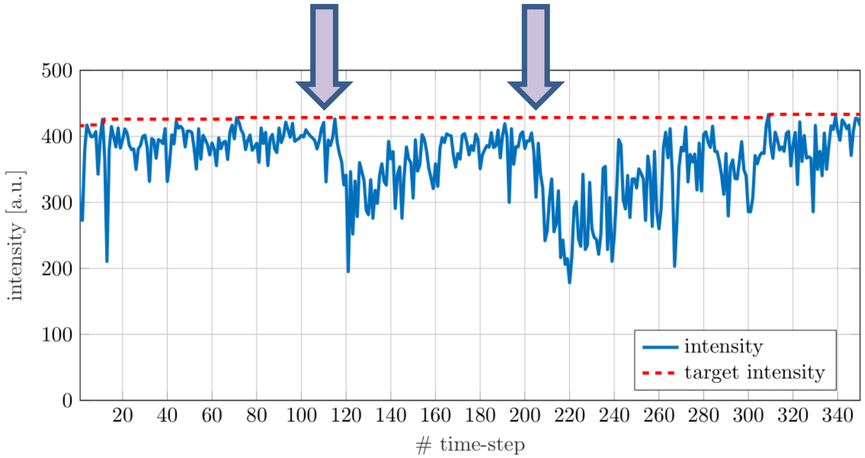
collaboration with CERN

(V. Kain, S. Hirlander)

Elettra Sincrotrone Trieste

Reinforcement Learning results

Recover from manually imposed perturbation (NPG Reinforce)



Policy gradient methods for free-electron laser and terahertz source optimization and stabilization at the FERMI free-electron laser at Elettra
F. H. O'Shea, N. Bruchon, G. Gaio – PRAB 2020

Attainment of the optimal working point starting from random initial conditions

Algorithm	Training data points	Mean num. of steps	Normalized final intensity
Q-learning	3128	11.28	-
NAF	1074	2.56	1.0019
NAF2	824	2.64	0.9995
AE-Dyna (TRPO)	450	4.46	1.0150
AE-Dyna (SAC)	500	3.28	1.0427
GradAscent	1024	3.82	0.9911
iLQR	1024	2.54	1.0019

Not RL

CERTIFIED MANAGEMENT SYSTEM
CQY CERTIQUALITY
UNI EN ISO 9001:2015
UNI ISO 45001:2018

Toward an accelerator autopilot

- ✓ **Decrease ''virtually'' to 0 the number of clicks** on graphical panels in the control room

⬇

- ✓ Move human knowledge and logics inside GUIs **to server side** (*TANGO devices*)

⬇

- ✓ Develop an infrastructure that can scale easily with the complexity of the logics and allows a fast deployment of automatic optimization / feedback systems

⬇

- ✓ Machine physicists and operators should become the developers / mantainers of the logic of the infrastructure

Behavior Trees (BT)

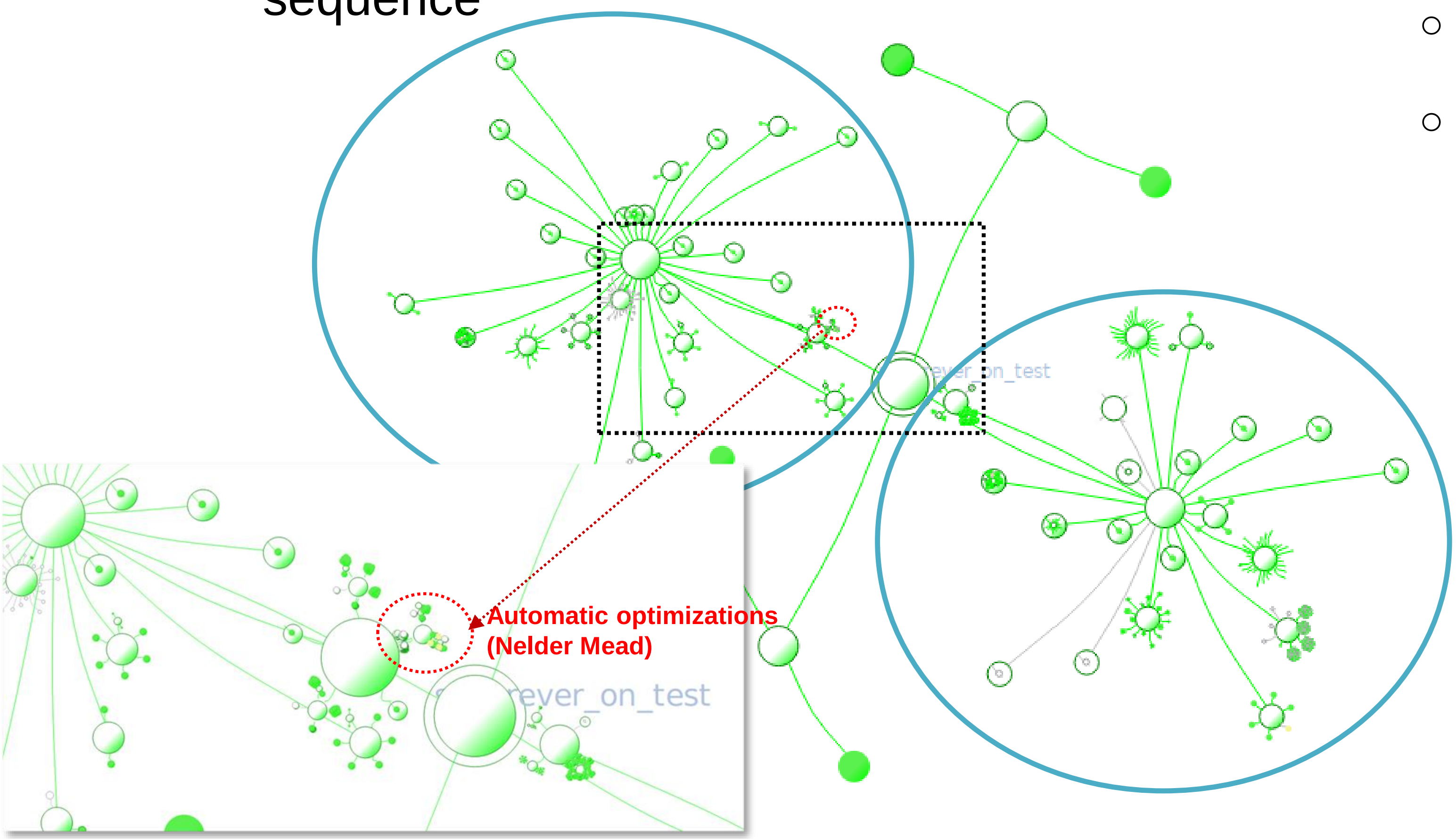
- ✓ BT are used for **in-game AI player opponents**, **UAV** and **robotics**
- ✓ They are able to create very complex tasks composed by simple decoupled self-contained tasks, regardless how they are implemented
- ✓ The tree-structure is composed by:
  - a root node
  - intermediate nodes (composite, selector, decorator) that control the flow
  - leaf nodes
- ✓ In the control system:
  - Each **leaf/node is a TANGO device** that executes a specific task (leaf) or launch in series or parallel other tasks (intermediate/root node)
  - In-house basic scripting language to execute simple reading/setting of variables after receiving a Start command; it supports if/else statement;
  - Can execute *Python*, *Matlab,* bash scripts…
  - Native support of **retry** and **fallback** actions
  - It controls a programmable TANGO device server which implements **feedback** / **numerical optimization schemes**
  - At Elettra BT are known a **SEQUENCERS**

A framework for high level machine automation based on behavior tree - in submission to ICALEPCS2021

Elettra full automation

- ✓ Recover beam loss
- ✓ Injection, beam to users
- ✓ It manages:
  - *HW devices*
  - *Optics correction*
  - *Slow/Fast Orbit Feedbacks*
  - *Automatic Optimizations*

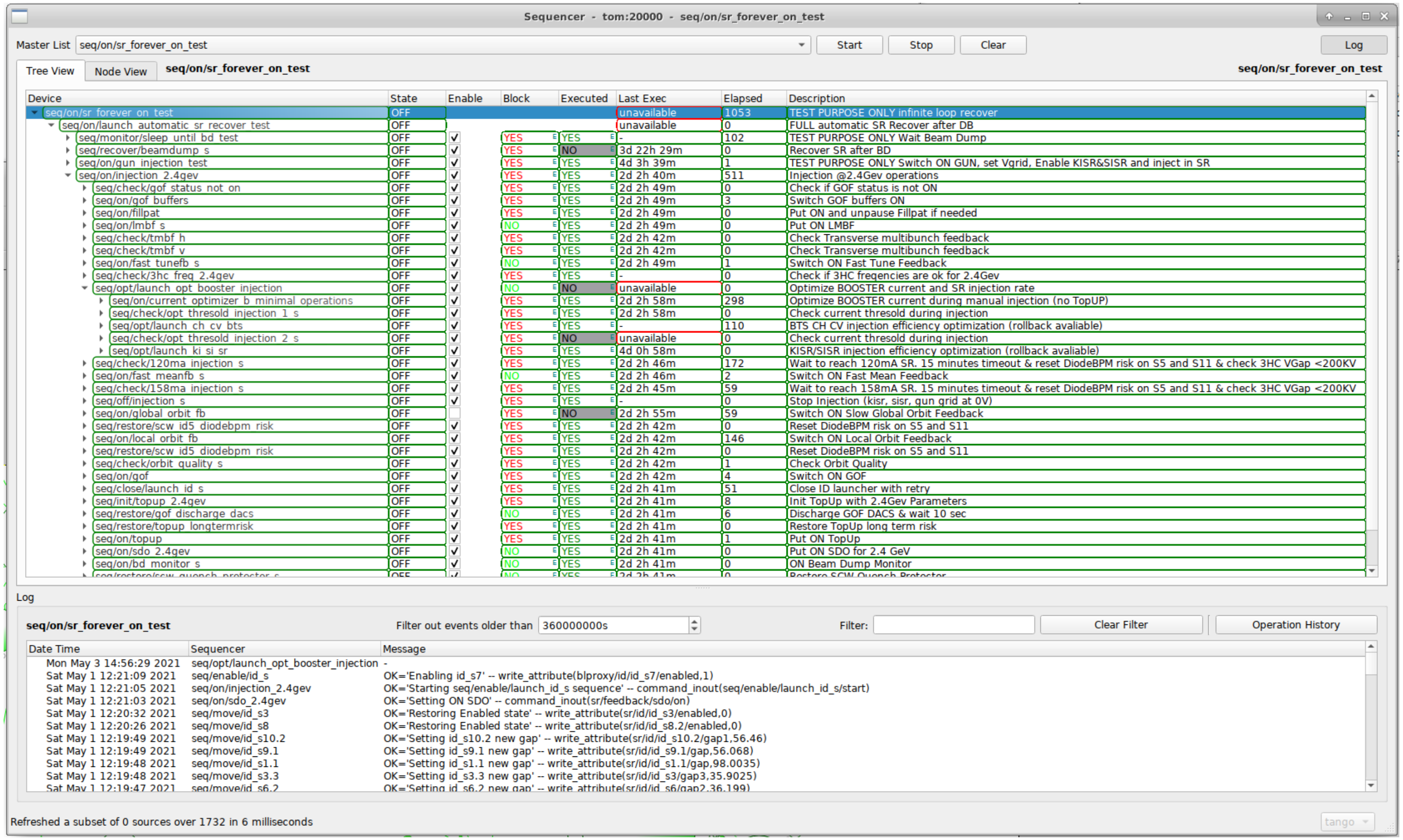
Injection- beam to user sequence



Recover from beam-loss sequence

Sequencers monitoring

Qt-based dynamic panel explores and monitors the execution of sequences (web interface available)



- ✓ Next steps:
  - Integration of *scikit-learn* and *OpenAI Gym* (optimizers)
  - Analyze drifts in the execution time to detect anomalies
  - Add BT learning capabilities to achieve more flexibility

Thank you!

N. Bruchon, P. Cinquegrana, G. Gaio, S. Krecic, G. Scalamera, G. Strangolino, F. Tripaldi, M. Trovo', L. Zambon