GeOFF: Generic Optimization Framework and Frontend

Nico Madysa

BE Seminar
1 July 2022
Motivation

many optimization problems in CERN accelerator operations:

- beam steering without optics model
Motivation

many optimization problems in CERN accelerator operations:
- beam steering without optics model
- alignment of electromagnetic septum

Originally developed by D. Bjorkman
many optimization problems in CERN accelerator operations:

- beam steering without optics model
- alignment of electromagnetic septum
- tune optimization
Motivation

many optimization problems in CERN accelerator operations:

- beam steering without optics model
- alignment of electromagnetic septum
- tune optimization
- simple corrector adjustments at Linac 3, ISOLDE, ...
Motivation

- diverse set of machines
Motivation

- diverse set of machines
- theoretical model impossible or unavailable and not worthwhile
Motivation

- diverse set of machines
- theoretical model impossible or unavailable and not worthwhile
- done manually with no “obvious” algorithm to follow
  (experience helps, however!)
Motivation

- many different optimization algorithms exist
Motivation

- many different optimization algorithms exist
- machine learning on the rise!
Motivation

- many different optimization algorithms exist
- machine learning on the rise!
- each package has slightly different API
Motivation

- many different optimization algorithms exist
- machine learning on the rise!
- each package has slightly different API
- most algorithms take control, don’t account for pauses, disturbances, cancellations, replays
Motivation

- many different optimization problems
Motivation

- many different optimization problems
- many different optimizers
Motivation

- many different optimization problems
- many different optimizers
- avoid combinatorial explosion with common optimization interfaces
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages

- Optimization problem
  - Single-objective optimization
  - Multi-objective optimization
  - Function optimization
  - Reinforcement learning problem
  - Septum
  - Steering
  - Tune
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
- extends their interfaces to numeric optimization
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
- extends their interfaces to numeric optimization
- extend Gym metadata system with CERN-specific info
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
- extends their interfaces to numeric optimization
- extend Gym metadata system with CERN-specific info
  ▶ which accelerator?

Optimization problem

- Single-objective optimization
- Multi-objective optimization
- Function optimization
- Reinforcement learning problem

Steering

Septum

Tune
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
- extends their interfaces to numeric optimization
- extend Gym metadata system with CERN-specific info
  - which accelerator?
  - communicates with machines?

Optimization problem

Single-objective optimization

Multi-objective optimization

Function optimization

Reinforcement learning problem

Septum

Steering

Tune

Nico Madysa

GeOFF @ BE Seminar

1 July 2022

5/11
Common Optimization Interfaces (COI)

- standardized interfaces and adapters for various packages
- “20% programming, 80% documentation”
- inspired by OpenAI Gym
- extends their interfaces to numeric optimization
- extend Gym metadata system with CERN-specific info
  - which accelerator?
  - communicates with machines?
  - wants to plot additional data?

Optimization problem

- Single-objective optimization
- Multi-objective optimization
- Function optimization
- Reinforcement learning problem
- Steering
- Septum
- Tune

Nico Madysa
GeOFF © BE Seminar
1 July 2022
Common Optimization Interfaces (COI)

- extensively documented at https://cernml-coi.docs.cern.ch/
Common Optimization Interfaces (COI) extensively documented at https://cernml-coi.docs.cern.ch/

describes expected and allowed behavior for each interface
Common Optimization Interfaces (COI)

- extensively documented at https://cernml-coi.docs.cern.ch/
- describes expected and allowed behavior for each interface
- includes several (2) tutorials
Common Optimization Interfaces (COI)

- extensively documented at https://cernml-coi.docs.cern.ch/
- describes expected and allowed behavior for each interface
- includes several (2) tutorials
- both surface-level user guides and in-depth API references
**Generic Optimization Frontend & Framework (GeOFF)**

- GUI app based on PyQt5 and AccWidgets (BE-developed CERN-specific GUI widgets)
Generic Optimization Frontend & Framework (GeOFF)

- GUI app based on PyQt5 and AccWidgets (BE-developed CERN-specific GUI widgets)
- lists, configures and runs optimization problems
Generic Optimization Frontend & Framework (GeOFF)

- GUI app based on PyQt5 and AccWidgets (BE-developed CERN-specific GUI widgets)
- lists, configures and runs optimization problems
- very extensible: optimization problems are loaded as plugins
GUI app based on PyQt5 and AccWidgets (BE-developed CERN-specific GUI widgets)
- lists, configures and runs optimization problems
- very extensible: optimization problems are loaded as plugins
- built-in list of optimizers plus runtime load mechanism
1. Implement class that inherits from COI (≈ 300 lines of code)

2. Dynamically load package into GeOFF

Optionally:

3. Upload code to CERN package index (provided by our Python team)

4. Notify us to get package integrated into GeOFF
Usage

1. implement class that inherits from COI (~ 300 lines of code)
   ▶ declare metadata

2. dynamically load package into GeOFF

   Optionally:

3. upload code to CERN package index (provided by our Python team)

4. notify us to get package integrated into GeOFF
Usage

1. implement class that inherits from COI (~ 300 lines of code)
   - declare metadata
   - implement machine communication

2. dynamically load package into GeOFF
   - Optionally:
     - upload code to CERN package index (provided by our Python team)
     - notify us to get package integrated into GeOFF
implement class that inherits from COI (≈ 300 lines of code)

- declare metadata
- implement machine communication
- add plotting (if any)
Usage

1. Implement class that inherits from COI (~300 lines of code)
   - Declare metadata
   - Implement machine communication
   - Add plotting (if any)

2. Dynamically load package into GeOFF
Usage

1. implement class that inherits from COI (~ 300 lines of code)
   - declare metadata
   - implement machine communication
   - add plotting (if any)

2. dynamically load package into GeOFF

Optionally:

3. upload code to CERN package index (provided by our Python team)
Usage

1. implement class that inherits from COI (~ 300 lines of code)
   ▶ declare metadata
   ▶ implement machine communication
   ▶ add plotting (if any)

2. dynamically load package into GeOFF

Optionally:

3. upload code to CERN package index (provided by our Python team)
4. notify us to get package integrated into GeOFF
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:**
  - LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances
  - longitudinal blowup (superseded by theoretical model)
  - splitter loss optimization between North Area experiments

- **ISOLDE:**
  - generic transfer line optimizer under investigation
Use cases

- PS:
  - steering from PS to n-TOF
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering

- **LEIR:**
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3: LEBT steering**

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances
  - longitudinal blowup (superseded by theoretical model)
  - splitter loss optimization between North Area experiments

- **ISOLDE:**
  - generic transfer line optimizer under investigation
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3: LEBT steering**

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3**: LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3: LEBT steering**

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances

---

Nico Madysa
GeOFF @ BE Seminar
1 July 2022
9/11
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3: LEBT steering**

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances
  - longitudinal blowup (superseded by theoretical model)
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3:** LEBT steering

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances
  - longitudinal blowup (superseded by theoretical model)
  - splitter loss optimization between North Area experiments
Use cases

- **PS:**
  - steering from PS to n-TOF
  - resonance compensation (also in PSB)

- **Linac3: LEBT steering**

- **LEIR:**
  - transfer line from Linac3 and injection
  - multi-turn injection
  - transfer line to PS

- **SPS:**
  - ZS alignment
  - crystal shadowing
  - tune adjustments to avoid resonances
  - longitudinal blowup *(superseded by theoretical model)*
  - splitter loss optimization between North Area experiments

- **ISOLDE:** generic transfer line optimizer under investigation
Next steps

- turn GUI into modular components
Next steps

- turn GUI into modular components
  - provide each part *and* the whole GUI as a library
Next steps

- turn GUI into modular components
  - provide each part and the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
Next steps

- turn GUI into modular components
  - provide each part *and* the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process
Next steps

- turn GUI into modular components
  - provide each part *and* the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process

- data recording, automatic model training
Next steps

- turn GUI into modular components
  - provide each part \textit{and} the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process
- data recording, automatic model training
- multi-objective optimization
Next steps

- turn GUI into modular components
  - provide each part and the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process

- data recording, automatic model training
- multi-objective optimization
- integration with Machine Learning Platform
Next steps

- turn GUI into modular components
  - provide each part *and* the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process

- data recording, automatic model training

- multi-objective optimization

- integration with Machine Learning Platform
  - apply stand-alone models to given optimization problem
Next steps

- turn GUI into modular components
  - provide each part *and* the whole GUI as a library
  - experiments & machines can package GUI with their own plugins
  - gives control to the users, avoids bottleneck in release process

- data recording, automatic model training

- multi-objective optimization

- integration with Machine Learning Platform
  - apply stand-alone models to given optimization problem
  - upload trained models
many optimization problems and many algorithms to choose from
Summary

- many optimization problems and many algorithms to choose from
- we present a standardized interface to connect the two
many optimization problems and many algorithms to choose from
we present a standardized interface to connect the two
extensive documentation with guides, examples and in-depth references
Summary

- many optimization problems and many algorithms to choose from
- we present a standardized interface to connect the two
- extensive documentation with guides, examples and in-depth references
- prototype GUI to present this platform to the user
many optimization problems and many algorithms to choose from
we present a standardized interface to connect the two
extensive documentation with guides, examples and in-depth references
prototype GUI to present this platform to the user
strong focus on customizability and plugin mechanisms
Summary

- many optimization problems and many algorithms to choose from
- we present a standardized interface to connect the two
- extensive documentation with guides, examples and in-depth references
- prototype GUI to present this platform to the user
- strong focus on customizability and plugin mechanisms
- successful use at several machines