Global Minimum Assessment in the PFAD Detector Offset Reconstruction Using Multiple Optimization Methods

TECHNISCHE UNIVERSITÄT DARMSTADT

Contact: aobertelli@ikp.tu-darmstadt.de, eplastinin@ikp.tu-darmstadt.de

Abstract

To investigate the structure of neutron-rich nuclei, we are developing a silicon tracker and a liquid hydrogen target which together form the STRASSE detection system. This system will allow to perform missing mass spectroscopy of radioactive nuclei from quasi-free scattering. PFAD is the demonstrator of the STRASSE detector, consisting of one third of the STRASSE tracker. It establishes a baseline for the performance capabilities and ensures the working principle of the future STRASSE tracker. During this thesis, experimental data acquired with PFAD will be used to develop and benchmark different optimization algorithms, which will then be compared to a neural-network alignment trained on simulations. This will lead to the best achievable alignment and position resolution. The work of this thesis will lay the groundwork for STRASSE.

1 PFAD

Within the STRASSE project, PFAD (Prototype For Advanced Detectors) is a demonstrator and prototype of the STRASSE array. Instead of the six modules of STRASSE, PFAD only consists of one third of the detectors. Therefore, the PFAD detectors are arranged in a two-arm system with two detectors on the left and two on the right side of the target (see Figure 1).

Each detector module consists of two single-sided silicon strip detectors. These type of detectors are segmented. Therefore, the deposited charge created by the passing of a charged particle can be used to determine the position where the charged particle passed through the detector. By using a second layer, the two reconstructed hit positions can be used to reconstruct the track of the charged particle. To determine the y- and z-coordinates of the hits respectively, two single-sided silicon wafers are needed. Each of the detector planes is placed along the x-axis. Therefore, the x-coordinate of the hit is not sensitive during the measurement but can be simply derived from the position of the plane.

This results in a total of 40 position coordinates needed to describe the full setup. Each module (with two planes) can be described by 4 translational coordinates (the x-coordinates of the two planes need to be treated individually) and 6 rotational coordinates (3 for each plane).

The position coordinates of the detectors during the experiment were measured beforehand. Due to handling and uncertainties in the position measurement, certain offsets to each coordinate are expected. If those offsets are not taken into account, the position resolution will suffer. Therefore, it is beneficial to determine the offsets leading to the optimal position resolution.

2 BOBYQA

Currently, the detector position coordinates are optimized using the python implementation of the BOBYQA [1] algorithm called Py-BOBYQA [2, 3]. BOBYQA is an abbreviation for "Bound Optimization BY Quadratic Approximation". It uses a quadratic model to approximate an objective function within predefined boundaries by evaluating the function at selected interpolation points from a predefined starting point.

To determine 37 out of the 40 coordinate offsets for PFAD, BOBYQA needed roughly

$$192635\ \text{s} \approx 53.51\ \text{hrs} \approx 2.23\ \text{days}. \tag{1}$$

The possibility of solving a high-dimensional minimization problem in a realistic time frame is the main advantage of such a model-based algorithm like BOBYQA. The disadvantage of BOBYQA is the locality of the minimization. BOBYQA is not developed to find global minima, therefore the algorithm can easily be trapped in a local minimum instead.

3 Different Optimization Algorithms

With the use of different approaches to the optimization of the objective function, a more cost efficient, more precise and possibly global optimization algorithm can be developed. The following list mentions some of the possible optimizers that can be used. Each has their own advantages and disadvantages.

- **Constrained Optimization BY Linear Approximation (COBYLA):**
  This optimizer works similarly to BOBYQA but with a linear model instead of a quadratic model. Even though the model used by COBYLA is simple, a relatively flat objective function would greatly benefit. Furthermore, simplifying the approximation process might reduce the total evaluation cost at the expense of accuracy.

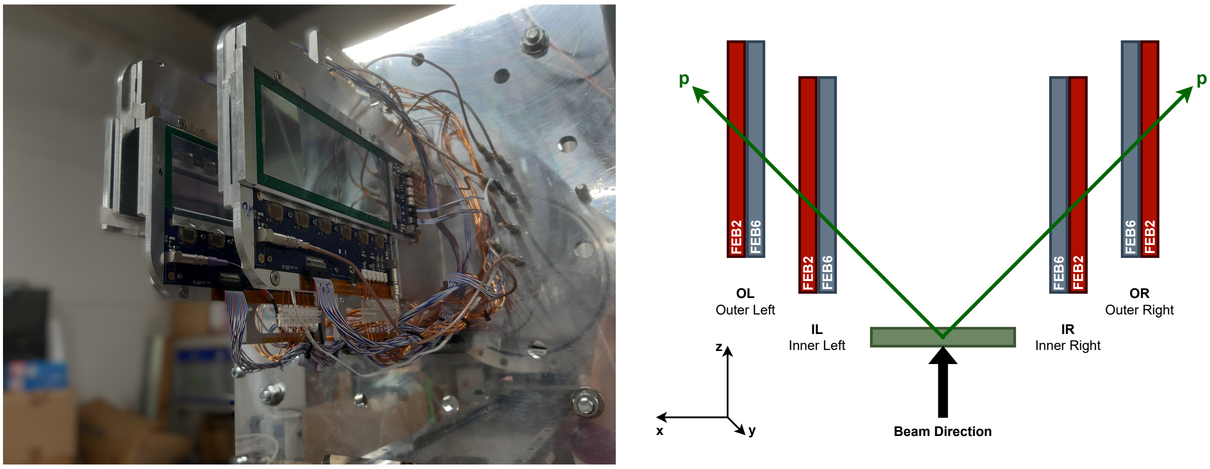

Figure 1: Picture of the PFAD demonstrator with its associated electronics (left) and a schematic representation of the arrangement of the four PFAD modules relative to the target (right).

- **Nelder–Mead [4]:**
  Nelder-Mead is a simplex search algorithm which is a heuristic direct search. A set of n+1 test points are selected and the function values at these points are evaluated and ordered according to their value. The worst value is then replaced by either reflection, expansion or contraction. Even though this method is rather robust, it is highly inefficient for a high number of dimensions and can easily get stuck at flat minima. Furthermore, it heavily depends on the selection of starting points.

- **Differential Evolution (DE) [5]:**
  Differential Evolution is a population-based stochastic. Here, a random population of test points is created and by mutating the points, better values are found and kept. This is done over multiple generations. This algorithm is a simple global minimization algorithm which would need many objective evaluations.

- **Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6]:**
  This algorithm is a global stochastic optimizer that creates new test points using a multivariate normal distribution. The population is sampled from a normal distribution described by a mean, step size and covariance matrix. The best values from the population are selected and based on these values a new distribution is determined. In this process the covariance matrix learns about the geometry of the objective function and can therefore be used to determine the minimum. This algorithm can determine the global minimum of the function but also needs many evaluations.

- **Trust-Region Bayesian Optimization (TuRBO) [7]:**
  The TuRBO algorithm is a state-of-the-art global optimizer that uses Bayesian Optimization within multiple trust regions. In each trust region a sample of testing points is created. A Gaussian Process is then fitted with these points. With an acquisition function new points are suggested. Wether or not these points are better than the ones before will decide if the trust region expands or shrinks.

4 Timeline

The time frame for a bachelor's thesis is set at 13 weeks. Here is an estimated/suggested schedule:

- **Week 1:**
  - Literature reading on STRASSE [8] and the current PFAD optimization algorithm (Master's thesis E. Plastinin)
  - Setup of the needed codes to create root files with and without optimized geometry
  - Familiarize with BOBYQA-based algorithm
- **Week 2-3:**
  - Minimization of needed data to reduce computation cost
  - COBYLA Implementation
- **Week 4-5:**
  - CMA-ES Implementation
  - TuRBO Implementation
- **Week 6-7:**
  - Neural Network Implementation and Training
- **Week 8:**
  - Comparison of different optimization methods
  - Assessment of most reliable and promising optimization method
- **Week 9-12:**
  - Writing
  - Analysis in parallel
  - Comments in parallel
- **Week 13:**
  - Final Comments
  - Thesis Submission

5 Requirements

Necessary requirements:

– Basic understanding of Python (main language to be used for implementing the different optimization algorithms; criterion and BOBYQA-based algorithm already implemented)

Optional (can also be acquired during the Bachelor's thesis):

– C++, Root (resulting data files will be based on root)

– Bash (running scripts to create root files automatically; already written, only handling)

– Linux (working on LDAS server)

References

[1] M. Powell. "The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives". In: Tech. Report, Dept. of Applied Math. and Theo. Physics (Jan. 2009).

[2] Coralia Cartis et al. "Improving the Flexibility and Robustness of Model-based Derivative-free Optimization Solvers". In: ACM Trans. Math. Softw. 45.3 (Aug. 2019).

[3] Coralia Cartis et al. "Escaping local minima with local derivative-free methods: a numerical investigation". In: Optimization 71.8 (2022), pp. 2343–2373.

[4] J. A. Nelder et al. "A Simplex Method for Function Minimization". In: The Computer Journal 7.4 (Jan. 1965), pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.4.308. eprint: https://academic.oup.com/comjnl/article-pdf/7/4/308/1013182/7-4-308.pdf. URL: https://doi.org/10.1093/comjnl/7.4.308.

[5] Helder Pereira Borges et al. "An Adaptive Metaheuristic for Unconstrained Multimodal Numerical Optimization". In: Bioinspired Optimization Methods and Their Applications. Ed. by Peter Korošec et al. Cham: Springer International Publishing, 2018, pp. 26–37. ISBN: 978-3-319-91641-5.

[6] Nikolaus Hansen. The CMA Evolution Strategy: A Tutorial. 2023. arXiv: 1604.00772 [cs.LG]. URL: https://arxiv.org/abs/1604.00772.

[7] David Eriksson et al. Scalable Global Opt. via Local BO. 2020. arXiv: 1910.01739 [cs.LG]. URL: https://arxiv.org/abs/1910.01739.

[8] H. N. Liu et al. "STRASSE: a silicon tracker for quasi-free scattering measurements at the RIBF". In: The European Physical Journal A 59.6 (June 2023). ISSN: 1434-601X. DOI: 10.1140/epja/s10050-023-01018-3. URL: http://dx.doi.org/10.1140/epja/s10050-023-01018-3.