ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST059

PyEmittance: A GENERAL PYTHON PACKAGE FOR PARTICLE BEAM EMITTANCE MEASUREMENTS WITH ADAPTIVE QUADRUPOLE SCANS[*]

S. A. Miskovich[†], A. Edelen, C. Mayes
SLAC National Accelerator Laboratory, Menlo Park, California, CA 94025 USA

Abstract

The emittance of a particle beam is a critically important parameter for many particle accelerator applications. Its measurements guide the initial tuning of an accelerator and are typically done using quadrupole or wire scans. Quadrupole scans are time-intensive, and it can be difficult to determine scan values that provide a good emittance measurement. To address this issue, we describe an adaptive quadrupole scan method that automates the determination of the scan range. With a given initial set of scanning values, our method adapts the range to capture the waist of the beam, and returns the Twiss parameters and a measure of the beam matching at the measurement screen. With the added capability to repeat beam size measurements when needed, this method provides a reliable measurement of the emittance even with sub-optimal initial conditions. To efficiently integrate these measurements into Python-based machine learning optimizations, the method was developed into a Python package, `PyEmittance`, at the SLAC National Accelerator Laboratory. We present the experimental tests of `PyEmittance` as performed at the Linac Coherent Light Source (LCLS) and the Facility for Advanced Accelerator Experimental Test (FACET-II).

INTRODUCTION

Emittance is a conserved property of a particle beam that characterizes its volume or area in the position-momentum phase space of the particles. A low emittance is desired for ensuring that the entire beam is transported with minimal losses, and is crucial the performance of X-ray free-electron lasers (FELs). The transverse beam emittance ϵ represents the area of the beam in the transverse planes defined by the positions and momenta, x and p_x in the horizontal plane, and y and p_y in the vertical plane. In the following, all references to the emittance indicate the transverse emittance.

The emittance is often measured using a single quadrupole scan with invasive wire or screen beam size measurements [1]. The quadrupole strength is varied over some range, and the consequent change in beam size is measured at a downstream wire (or screen). The range over which the scan is performed is typically selected by trial and error, and requires adjustment when upstream beam line elements are changed. For applications requiring efficient and robust emittance measurements, such as automated tuning algorithms or machine learning (ML) based optimizations, this process is inefficient and limiting. Additionally, the current MATLAB-based emittance measurement software at SLAC National Accelerator Laboratory (SLAC) is difficult to integrate with modern ML software being run and deployed on the machine.

In this paper, we present `PyEmittance`, a new Python package that provides general beam emittance measurements with adaptive quadrupole scans [2]. We show the benefit of the adaptive measurements method when robustness and automation are crucial, and present the test measurements performed at SLAC, at the Linac Coherent Light Source (LCLS) and the Facility for Advanced Accelerator Experimental Test (FACET-II).

IMPLEMENTATION

While `PyEmittance` can be used as an out-of-the-box emittance calculator given a data set of quadrupole strengths and corresponding beam sizes, the data acquisition and adaptive scanning available in the package are useful in cases where hand-tuning of the quadrupole scanning range is unavailable, or when efficient and robust measurements are needed. The package has optional functions that can be performed ahead of calculating the emittance for a flexible way of obtaining a robust emittance measurement. The general method to run a measurement follows the steps below:

1. Select initial scan of 3+ points within some predefined quadrupole range (can be random or specified by the user) and acquire transverse beam sizes for each initial scan point.

2. Given the data collected, calculate a new adapted scanning range to better capture the beam waist. Acquire new beam size measurements within the adapted range (the number of points measured is specified by the user).

3. Check the symmetry of the obtained curve, and if needed, acquire new beam sizes to evenly sample on each side of the minimum.

4. Remove any points beyond the turning point(s) of the curve to maintain a concave curve. If the number of points is now below the number specified, add points within the final range.

5. Finally, calculate the emittance and the match.

[*] Work supported by the U.S. Department of Energy, under DOE Contract No. DE-AC02-76SF00515 and the Office of Science, Office of Basic Energy Sciences.

[†] smiskov@slac.stanford.edu

This is done for each transverse plane separately to provide the ability to measure the emittance when the beam waists are at different quadrupole values. This is helpful when measuring the emittance at locations along the beamline away from the beam waist, or when running ML optimization or characterization algorithms that explore regions where the injector optics are not near optimal.

Emittance Computation

In the variable quadrupole strengths method that we use here, the transfer matrix elements are varied by changing the strength of a quadrupole lens located upstream of a screen (or wire). Given a quadrupole at location s_0, and a screen at location s_1, we calculate the emittance from the reconstructed beam matrix $\mathbf{\Sigma}^0$ at location s_0 following the transformation

$$\mathbf{\Sigma}^1 = \mathbf{R} \cdot \mathbf{\Sigma}^0 \cdot \mathbf{R}^T,$$

where $\mathbf{\Sigma}^1$ is the beam matrix at s_1, and $\mathbf{R}$ is the total transfer matrix of the beamline components between s_0 and s_1. If electromagnetic elements are located between s_0 and s_1, the total transfer matrix $\mathbf{R}$ is provided by the user in the `PyEmittance` configuration files (obtained from an external computation or optical model). Otherwise, the drift length and scanning quadrupole length, defined in the configuration files as well, are used to compute the transfer matrix. The components of $\mathbf{\Sigma}$ are given by $\sigma_{ij} = \langle x_i x_j \rangle$, where $\langle\ \rangle$ denotes an average over all particles in the beam, and the components of the transverse phase-space vector $\mathbf{x}$ are (x, p_x, y, p_y). At s_1, we measure the beam size, *i.e.* $\sigma_{11} = \langle x^2 \rangle$, for at least 3 different quadrupole settings, and we deduce the elements of $\mathbf{\Sigma}^0$ by solving the system of equations defined by

$$\sigma^1_{11} = R^2_{11}\sigma^0_{11} + 2R_{11}R_{12}\sigma^0_{12} + R^2_{12}\sigma^0_{22}.$$

Given a data set of quadrupole strengths, beam sizes, and errors, we fit the measured data using a χ^2-minimization technique, followed by a pseudoinverse method to solve the resulting system of equations and obtain σ^0_{11}, σ^0_{22} and σ^0_{12}. For more details, see Ref. [3]. This computation is done for both transverse planes, and the emittance for each is obtained following the relationship

$$\epsilon = \sqrt{\sigma_{11}\sigma_{22} - \sigma^2_{12}}.$$

In the following, all references to beam emittance refer to the normalized transverse emittance given the geometric mean of the transverse emittance values. The Twiss parameters and the match are also computed and returned based on user-defined model Twiss parameters.

Data Acquisition

In order for the software to measure the beam sizes when needed, a function interfacing with the machine (or simulation) needs to be defined. This function sets the machine to the desired configuration and measures the beam spot size at the defined location, then returns the sizes and errors on the measurements for the horizontal and vertical planes. Given that the user can define this function, `PyEmittance` can be used with any model or machine that is capable of returning beam size predictions or measurements for a given injector configuration.

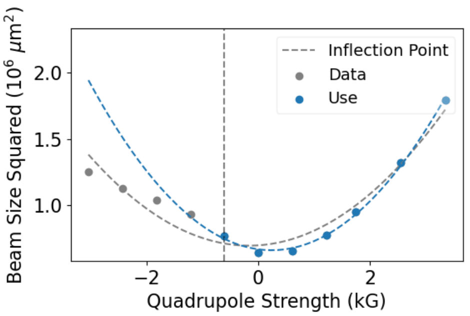

Figure 1: Example of data selection based on the inflection point of the resulting polynomial fit. The blue points are selected to be used in the emittance measurement.

In this work, `PyEmittance` was used with LCLS simulations (surrogate model and Bmad model [4]), the LCLS copper injector line, and the FACET-II injector line. For the latter two, `PyEpics` [5] was used to interface with the control system. In all cases, the package requires system details to be set in the `configs` directory that define the settings required for the measurement. Some examples of how this is set up is shown the `examples` directory in [2].

Adaptive Range Selection and Measurement

When running a measurement, initial quadrupole values have to be defined, typically 3–4 points within the valid range. This data is used to determine a new range in each plane that will scan more finely around the minimum of the beam size: the initial points within some specified cutoff (typically around 2× the minimum) are fitted (if 3 or more points obey that constraint) with a second degree polynomial fit with the quadrupole values as the x-coordinates, the beam sizes squared as the y-coordinates, and the beam sizes and their errors as weights. The resulting coefficients are used to determine if the fit is concave, in which case the new range is taken to be from the minimum of the data to the scanning range limit, in the direction of decreasing beam size. Otherwise, the intersection of this polynomial with some upper limit of the beam size squared is determined to be the new range, if within the allowable scanning range, otherwise it is set to the range limits. The upper limit has been set empirically to 2× the minimum beam size squared (or an approximation if the two do not intersect). A user-specified number of data points is then measured using the user-defined beam size measurement function and added to the data set.

Next, the symmetry of the parabola is checked for each plane. If more points are needed on one side of the minimum, the number and location (uniformly spaced) of the points are determined, and they are measured and added.

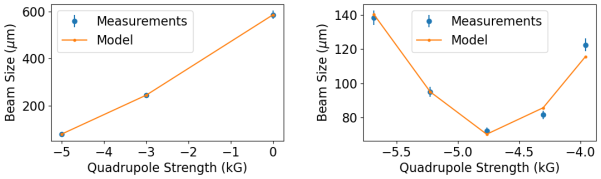

Figure 2: Left: Initial scan based on 3 points selected given an approximate quadrupole range of $[-5, 0]$. Right: Final emittance scan with the full adaptive method presented here.

The last check performed is useful when the scanning range is too large and deviations from the convex parabolic shape is seen towards the edges. In these cases, the switching points in the data are computed, and points outside of the convex region are removed. An example of this is shown in Fig. 1, where the gray and blue points indicate the data to be removed and to be used, respectively. If the resulting data set has less than the user-specified number of points, an optional function is run to add points uniformly around the existing data within the final convex scanning range.

In all of the above, the data can be reused so no measurements are repeated within a scan if the ranges overlap.

An example of running an emittance scan in simulation mode using this method is shown in Fig. 2. The left image shows the initial scan given 3 quadrupole strength values naively selected within an approximate quadrupole range of $[-5, 0]$. The right plot displays the final curve using the procedure described above, requiring 5 points within the final range. The emittance value can be 10–15% lower and its uncertainty reduced by 50% using this method compared to measurements done by randomly selecting scanning values along the quadrupole range as shown in the left plot.

EXPERIMENTAL TESTS

The PyEmittance package is being used at SLAC to measure the emittance at the LCLS and the FACET-II injector lines. Both non-coherent optical transition radiation (OTR) screens and wires were used to measure the beam spot sizes. In the case of OTR screens, the image processing was done within the PyEmittance package.

We compared the measurement outputs of PyEmittance to measurements using the existing MATLAB software at SLAC and found the results to be in agreement when the scanning ranges are similar. Figure 3 shows a comparison of the scans with each method for both machines. Note that for the MATLAB scans, the quadrupole range was iteratively hand-tuned to obtain the final measurement displayed. As we expect, PyEmittance finely scans around the minimum to the desired scale, and the closer the hand-tuned MATLAB and PyEmittance ranges get, the closer the outputs.

An advantage of a Python package is the ease of integration with other Python-based software being developed at SLAC and elsewhere, especially for online tuning and ML-based procedures. We have used several ML-algorithms with PyEmittance to optimize the emittance in LCLS and FACET -II, with efficient and reliable performance. Example evaluation functions that algorithms can call are given in `pyemittance.emit_eval_example` in [2].

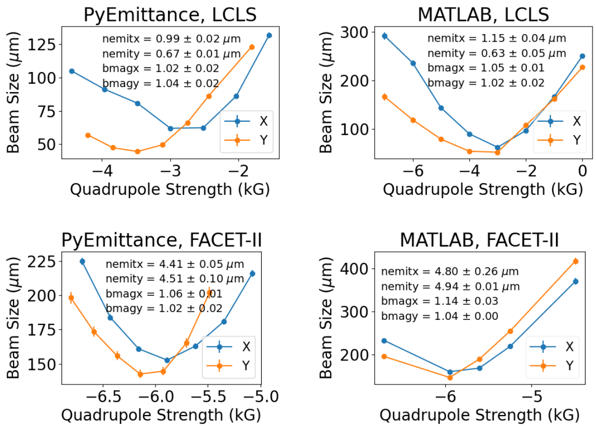

Figure 3: Top: Comparing emittance measurements in LCLS to the existing MATLAB application. Bottom: Comparing emittance measurements in FACET-II.

CONCLUSION

We present PyEmittance, a new Python package for general particle beam emittance measurements with adaptive quadrupole scans. It is designed for robust and flexible integration with other Python-based software, with online accelerators, and with simulation software. Future work includes improved configuration settings for interfacing with machines/simulations, support for bunch length measurements, and for multi-wire emittance measurements. It is an open source software, and can be installed via the command `pip install pyemittance`. Contributions and suggestions to the GitHub repository [2] are welcome.

ACKNOWLEDGEMENTS

The authors would like to thank Timothy Maxwell, Nicole Neveu, and Finn O'Shea for their input while developing this package.

REFERENCES

[1] M. Minty and F. Zimmermann, *Measurement and Control of Charged Particle Beams*. Berlin, Germany: Springer, 2013.

[2] PyEmittance, https://github.com/slaclab/PyEmittance/.

[3] F. Loehl, "Measurements of the transverse emittance at the VUV-FEL", DESY-THESIS 2005-014 and TESLA-FEL 2005-03, Diploma Thesis, Phys. Dept. Hamburg University, Germany, 2005.

[4] D. Sagan, “Bmad: A relativistic charged particle simulation library”, *Nucl. Instrum. Methods*, vol. A558, pp. 356–359, 2006. doi:10.1016/j.nima.2005.11.001

[5] PyEpics, http://cars9.uchicago.edu/software/python/pyepics3/.