RECENT DEVELOPMENTS IN LONGITUDINAL PHASE SPACE TOMOGRAPHY

S. Albright[*], A. Lasheen, CERN, Geneva, Switzerland
C. Grindheim, NTNU: Norwegian University of Science and Technology, Trondheim, Norway
A. Lu, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

Longitudinal phase space tomography has been a mainstay of beam diagnostics in most of the CERN synchrotrons for over two decades. For most of that time, the reconstructions have been performed using a highly optimised Fortran implementation. To facilitate increased flexibility, and leveraging the significant increase in computing power since the original development, a new version of the code has now been developed. The new version implements an object-oriented Python API, with the computationally heavy calculations in C++ for increased performance. The Python/C++ implementation is designed to be highly modular, enabling new and diverse use cases. For example, the tracking can now be performed externally and the results used for tomography, or a single set of tracked particles can be used for multiple reconstructions without needing to repeat the tracking. This paper summarises the functionality of the new implementation, and some of the applications that have been enabled as a result.

INTRODUCTION

Longitudinal phase space tomography (hereafter, tomography) is a method of reconstructing the longitudinal phase space distribution from a set of measured longitudinal profiles [1]. As the bunch undergoes synchrotron oscillations, the line density gives a one-dimensional (1D) projection of the two-dimensional (2D) phase space distribution. Over the course of a full synchrotron period, the phase space distribution will be seen from all angles. By recording a number of beam profiles, which are projections of the distribution, it is therefore possible to perform tomographic reconstruction of the phase space distribution.

The reconstruction is an iterative process. On each iteration, the weighting of the reconstructed distribution is adjusted. First, the difference between the measured and reconstructed profiles is measured. Then, this difference is used to adjust the weighting such that it is minimised. Figure 1 shows the measured bunch profiles (left column) of a bunch undergoing synchrotron oscillations, the center column is the reconstructed profiles and the right column is the difference.

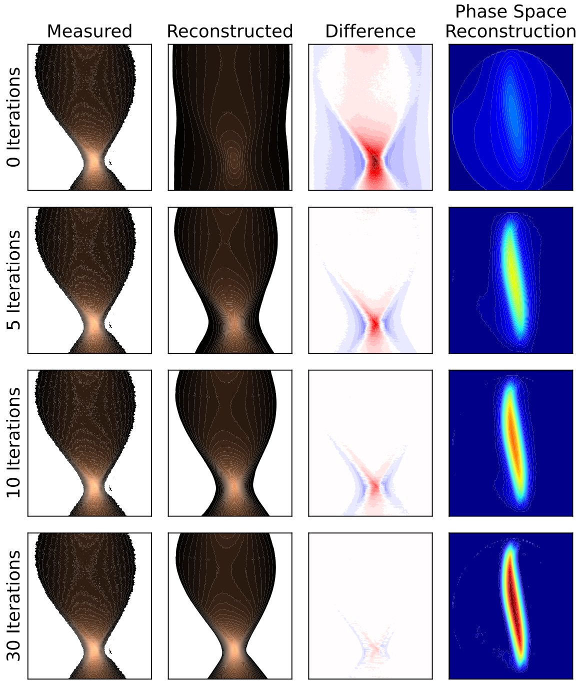


Figure 1: First Column: Waterfall plots showing measured bunch profiles during quadrupole oscillations. Second Column: Reconstructed waterfall plots produced by tomographic reconstruction. Third Column: Difference between measured and reconstructed profiles showing the error getting small as the number of iteration increases. Fourth Column: Reconstructed phase space distribution.

In Fig. 2 the reconstructed phase space distribution part way through the quadrupole oscillations in Fig. 1 can be seen. The measured profile is shown in black on the upper plot, the reconstructed reconstructed profile is in red, as can be seen they are in very close agreement as is the case on all measured profiles.

Tomography is a vital part of the beam diagnostics in the CERN accelerator complex. Recently, to allow further developments and new applications, a new version has been developed in mixed Python/C++. This paper outlines the changes introduced in the new version, and highlights two examples of new applications that were not previously possible. These changes have also simplified existing applications of tomography, such as RF voltage calibration [2, 3].

ALGORITHM MODIFICATIONS

Originally, the tomography software was written in Fortran95 and optimised using the High Performance Fortran (HPF) extension [4]. The software was designed for optimal speed and memory usage, which were significant considerations due to the computing power available at the

---

[*] simon.albright@cern.ch

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOPT043

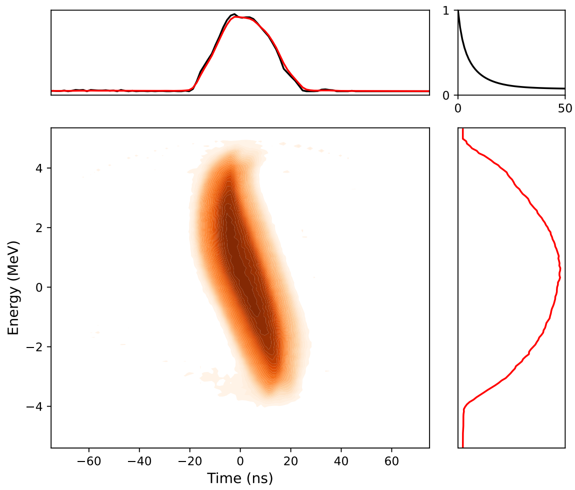


Figure 2: Reconstructed phase space distribution during quadrupole oscillations. The main panel shows the reconstructed phase space distribution. The upper panel shows the measured (black) and reconstructed (red) profiles. The right panel shows the energy projection of the reconstructed distribution. The upper-right plot shows the discrepancy between measured and reconstructed profiles on each iteration.

time. However, due to the structure of the algorithm, introducing new features would require significant effort. To facilitate future developments, the algorithm was refactored into an object-oriented structure, separating the different stages of the calculation from each other [5]. Additionally, the new algorithm is written almost entirely in Python, which facilitates rapid and flexible code development. To avoid excessive run times, the computationally heavy parts of the algorithm are written in optimised C++, which ensures only a modest loss of speed compared to the Fortran code.

The fundamentals of the reconstruction process are unchanged and follow the schematic shown in Fig. 3. Most importantly, the code is now broken into discrete and independent modules, which allows external code to replace any of the default functionality to modify the behaviour.

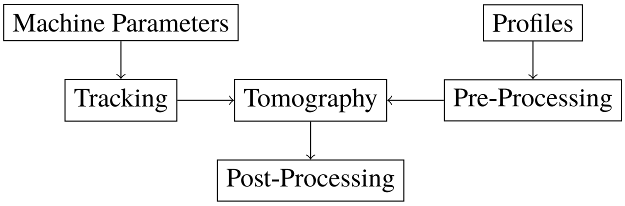


Figure 3: Schematic layout of the main data-flow for tomography. Each node is separate and user-code can replace any step if desired.

In the particle tracking step, the coordinates of each particle are stored at the turns corresponding to a measured profile, where previously they were binned in phase space to save memory. In this way, particle tracking is continuous with the only interruption being to store coordinates in a pre-defined array. The reconstruction still uses the ART method [6], however the weighting is now applied directly to the tracked particles, rather than to the phase space bins to which they had been assigned.

Similarly to the particle tracking, the pre-processing of bunch profiles is now self-contained. By default, the measured bunch profiles are re-binned and the noise floor is cut away.

At each stage identified in Fig. 3, user code can be introduced. In addition, it is possible to store the result of one stage (e.g. tracking) and pass the same data repeatedly to the Tomography step, only changing the other input. This facilitates automated tomography, where the same tracked particles are used with regularly updated profiles, and also simplifies voltage calibration where the same profiles are combined with different tracked particles.

NEWLY ENABLED APPLICATIONS

This section outlines two of the recent progressions in longitudinal tomography, that have been enabled with the new developments. In both cases, the modularity of the code is an essential requirement.

External Tracking

With the new code structure, it is possible to do the tracking separately to the tomography. This feature can be used in contexts where the RF voltage is changing in ways that are more complex than can be treated by the original tomography algorithm (maximum of two RF harmonics with programs changing linearly). One example is the longitudinal rotation of bunches before PS extraction for LHC-type beams. For these beams, a combination of adiabatic and non-adiabatic bunch shortening (bunch rotation) is applied, which requires two different RF harmonics and non-linear voltage functions as shown in Fig. 4a.

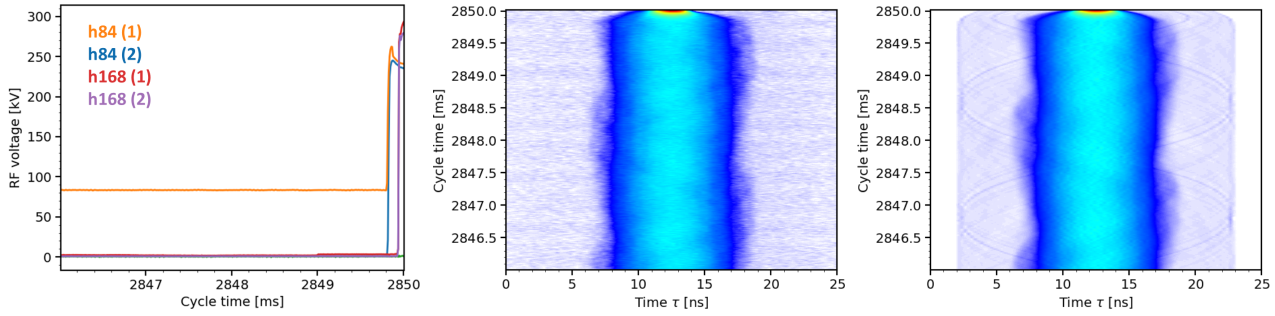
To allow reconstruction during this highly dynamic time period, the longitudinal tracking code BLonD [7] was used to track the test particles. After tracking, the test particles were binned to correspond with the binning of the measured bunch profiles shown in Fig. 4b to allow reconstruction. The time projections of the reconstructed distribution are shown in Fig. 4c; as can be seen the agreement is excellent.

This technique allows tomography to be used without delaying extraction, which was previously required to force static RF voltages. The distribution in phase space at extraction for the example in Fig. 4 can be seen in Fig. 5.

Automated Tomography

In principle, the tomographic reconstructions could be automated with the original Fortran implementation. However, this would be impractical as the full algorithm has to be run for each reconstruction, which is relatively time consuming. With the modular design of the new API, the tracking can be performed ahead of time, and the particle coordinates stored in memory. Then, for each reconstruction it is only necessary to acquire new beam profiles, pre-process them and

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOPT043



(a) RF voltages (b) Measured profiles (c) Reconstructed profiles

Figure 4: Longitudinal tomography during bunch shortening before PS extraction for LHC-type beams. The voltage functions during the final few ms (left) are used with the measured bunch profiles (middle) and external tracking to get the reconstructed bunch profiles (right). Note the good reconstruction of tail particles following synchrotron motion.

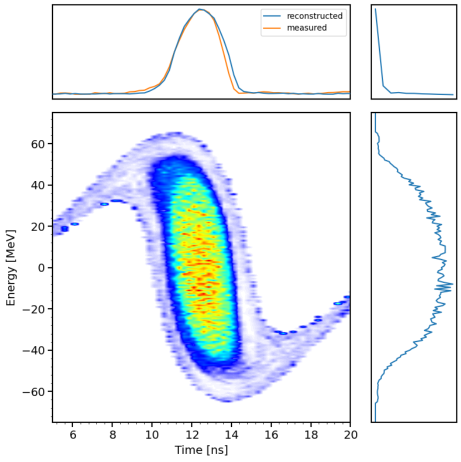


Figure 5: The density in longitudinal phase space at PS extraction for the acquisition in Fig. 4. The tail density located on the top left of the distribution is important information for beam transfer.

then perform the reconstruction with the pre-calculated particle tracks. In production of physics beams, the parameters change very rarely.

Originally proposed in [8], an automated tomography demonstration system is currently being tested in the CERN PS Booster. Here, a new cycle is played every 1.2 s and up to four reconstructions are required per cycle. To facilitate this, the reconstruction runs on the UCAP system [9], which is designed for efficiently running and publishing data analysis on acquired accelerator parameters. For each reconstruction, a set of tracked particles is computed and stored on disk, then on each acquisition the UCAP system runs the iterative reconstruction and publishes the results.

For this application, it is essential both that the reconstructions run fast enough to allow up to four measurements per cycle and also that they have adequate precision. For a selection of representative measurements, the reconstructions were computed with a variety of input parameters (number of tracked particles, number of profiles, number of bins). Comparing the precision and run-time for the reconstruction stage (ignoring the tracking stage) shows that precise reconstructions are possible with very fast computation.

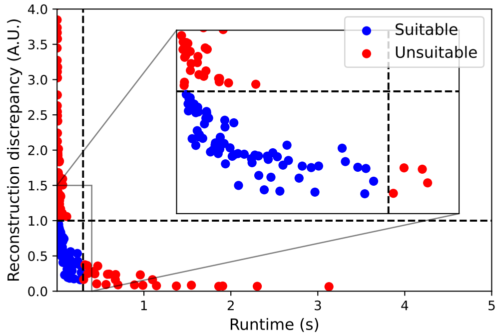


Figure 6: The reconstruction discrepancy versus runtime for various combination of reconstruction parameters, without the tracking stage.

Figure 6 shows a scatter plot of the reconstruction discrepancy (vertical axis) vs the computation time (horizontal axis) for a selection of input parameters. A reconstruction with discrepancy less than 1 and runtime below 0.3 s (indicated by the dashed lines) is suitable for automated tomography in the CERN PS Booster.

CONCLUSION

Longitudinal phase space tomography continues to be a vital beam diagnostics tool for the CERN accelerator complex. For many years, a highly efficient Fortran based algorithm was used, with excellent results. A new Python/C++ version has been developed, which facilitates further developments and more flexible applications of the technique by introducing modularity to the algorithm. Two examples; automated tomography with pre-tracked particles, and tracking with an external tracking code were discussed here. In the future, more developments are planned, such as multi-bunch tomography and hardware acceleration on Graphical Processing Units.

REFERENCES

[1] S. Hancock, "A Simple Algorithm For Longitudinal Phase Space Tomography", PS/RF/Note 97-06, CERN, Geneva, Switzerland, 1997

[2] D. Quartullo, S. Albright and H. Damerau, "Frequency-Dependent RF Voltage Calibration Using Longitudinal Tomography in the CERN PSB", presented at IPAC'22, Bangkok, Thailand, June 2022, this conference.

[3] D. Quartullo *et al.*, "RF Voltage Calibration Using Phase Space tomography in the CERN SPS", presented at IPAC'22, Bangkok, Thailand, June 2022, this conference.

[4] S. Hancock, M. Lindroos and S. Koscielniak, "Longitudinal Phase Space Tomography With Space Charge", *Phys. Rev. Spec. Top. Accel Beams*, vol. 3, p. 124202, 2000. `doi:10.1103/PhysRevSTAB.3.124202`

[5] C. Grindheim, S. Albright, "Longitudinal Phase Space Tomography Version 3", CERN-ACC-NOTE-2021-0004, CERN, Geneva, Switzerland, 2021

[6] R. Gordon, "A Tutorial on ART (Algebraic Reconstruction Techniques)", *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 78–93, 1974.

[7] CERN Beam Longitudinal Dynamics code BLonD. `http://blond.web.cern.ch`

[8] S. Albright, "A Proposal for an Automatic Longitudinal Tomography Demonstration Project in the Proton Synchrotron Booster", CERN-ACC-NOTE-2020-0014, CERN, Geneva, Switzerland, 2020

[9] L. Cseppento and M. Buttner, "UCAP: A Framework for Accelerator Controls Data Processing @ CERN", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, paper MOPV039. `doi:10.18429/JACoW-ICALEPCS2021-MOPV039`