



Recent Developments and Status of MarlinTPC and Related Software

Jason Abernathy¹, Li Bo², Patrick Conley¹, Klaus Dehmelt³, Ralf Diener³, Jan Engels³,
Keisuke Fujii⁴, Christian Grefe⁵, Jim Hunt⁶, Katsumasa Ikematsu⁷, Matthias Janssen³,
Yukihiro Kato⁸, Martin Killenberg⁵, Frederik Klöckner⁹, Thorsten Krautscheid⁹,
Astrid Münnich⁵, Christoph Rosemann³, Oliver Schäfer¹⁰, Martin Ummenhofer⁹,
Adrian Vogel⁵, Peter Wienemann⁹, Simone Zimmermann⁹

November 9, 2010

Abstract

MarlinTPC is a software package used in the TPC R&D for a future Linear Collider. It is based on the common ILC software framework and comprises simulation, reconstruction and analysis. We give a basic overview of the package and report on the recent developments and related software tools.

¹ University of Victoria, Victoria, BC, Canada

² Tsinghua University, Beijing, China

³ DESY, Hamburg, Germany

⁴ KEK, Tsukuba, Japan

⁵ CERN, Geneva, Switzerland

⁶ Cornell University, Ithaca, NY, USA

⁷ University of Siegen, Germany

⁸ Kinki University, Osaka, Japan

⁹ University of Bonn, Bonn, Germany

¹⁰ University of Rostock, Rostock, Germany

1 Introduction

MarlinTPC is a common software package used in the TPC R&D of the LCTPC collaboration [1] for a future Linear Collider. To use a common reconstruction software has the advantage of minimizing the duplication of effort and of providing a test bench to compare, on an equal footing, various technologies for the TPC read-out. Further, the previously available software packages, that were mostly developed and used inside a single work group, did not offer the functionality and flexibility to reconstruct data from the so-called Large TPC Prototype [2]. The challenges here comprise a much more complicated geometry of the read-out, the need to track a particle over several distinct read-out modules and corrections for field inhomogeneities.

MarlinTPC is building upon the reconstruction framework MARLIN [3], which provides the basic structure implying a processor based structure with defined interfaces. As data model, the common persistency framework LCIO [4] is used, which is the standard in ILC software. Complemented is the usage of LCCD [5] as the standard way of providing access and storage of conditions data. The geometry description is based on GEAR [6]. The MarlinTPC package comprises a simulation part, which offers different levels of detail, a reconstruction chain and analysis functionality.

In total about thirty people contributed so far to MarlinTPC, and recently active are twelve. The code itself resides in a subversion repository [7], and provides working branches for the developers and the common development (called *trunk*). It also provides the means for software releases, which are included in the general ILCSoft installation [8]. There is a project homepage [9], which is a wiki page that allows constant editing and updating of information. In addition there is a mailing list [10] and a discussion forum [11]. To complete the software communication, a bug tracker [12] is available. In addition, a monthly telephone/EVO meeting is held with regular participation from Europe, Asia and the Americas, in which current issues are discussed and recent developments are presented.

2 Reconstruction Overview

MarlinTPC is an implementation of a TPC data model [13], but still incomplete with respect to all described processors and functionality. The data types defined in LCIO are driven through a linear chain of software modules called *processors*. These processors interact on basis of the event.

In LCIO, several data types are defined for tracking, which outline the processing chain; listed in Table 1.

Two main distinctions need to be made in the description of the processing chain. The first is between the basic reconstruction, that is anything before track level, and the higher reconstruction, which includes track finding and fitting. The second distinction is within the basic reconstruction between the different types of data input. The latter difference arises from the different read-out schemes, namely pixelized data and pad-based.

Type	Default Collection Name
TrackerRawData	TPCRawData
TrackerData	TPCConvertedData
TrackerPulse	TPCPulses
TrackerHit	TPCHits
Track	TPCTracks

Table 1: The basic data types defined in LCIO for track reconstruction in a tracking detector. The order of the data types and their (default) name outline the processing chain.

3 Basic Reconstruction

The most basic unit of pad based data is the time binned ADC data of individual pads. These are grouped together into pulses, which are the convoluted signals on individual pads. In pad based read-out the pulses are combined row wise into hits. These hits are the space points, calculated from the time and the charge weighted position of the individual pulses.

Data from pixel read-out has to be treated differently. Due to the small pixel size, the chip can resolve individual charge clusters, which are spread over several rows. This requires a 2D clustering algorithm. The current Timepix chip can only record either charge or time per pixel, so that interpolation between pixels is necessary. From the reconstructed clusters the hit positions are calculated.

The common result from the either basic reconstruction is the TrackerHit, a 3D space point including charge information and the respective errors.

4 Higher Level Reconstruction

The next step in the reconstruction chain is the track finding and fitting to build tracks from the hits. The first part is a pattern recognition, which finds hits that belong to a single track. This collection of hits is passed to the final fitting, yielding the track parameters.

Track finding and fitting is an iterative process, which needs some logic of steering. Within the linear processing scheme of MARLIN this is hard to achieve without making the construct overly complicated. Some initial effort to implement track finding and fitting for single track events is available, but the recent developments go in the direction of using separate libraries for this task, which are then called by a processor. Most notably a tracking library based on the Kalman Filter technique was developed.

The library KalTest [14] is based on ROOT [15] and provides basic C++ classes for track fitting with Kalman filter technique. The library allows to handle measurement layers of various shapes and coordinate systems in a unified way, while minimizing user implemented code. The library has been tested for test beam data from the Large

Prototype TPC. Currently, the library is being moved out of the MarlinTPC code and into a separate package of the ILC software, so that it can also be used by other projects. Another library that is currently under development is a track finder based on Hough transformation. This will be able to find straight and helix tracks. It is planned to combine this with a χ^2 fitter.

In addition there is a likelihood fitter which can find the most likely track parameters using a model of charge distribution when given a set of signal pulses. A future goal is to implement corrections into the likelihood fitter which can correct the effects of field inhomogeneities.

5 Correction of Large Prototype Data

One of the main reconstruction goals of measurements with the Large Prototype is to show the ability to describe and correct the large inhomogeneities in the electrical and magnetic fields.

An important prerequisite has been achieved by making the measured magnetic field map of the PCMAG¹ [16] available in MarlinTPC [17]. Therefore, the measured and fitted field map is converted to a binned field map. This binned map can be used in MarlinTPC via an existing field interface.

The usage of data taken with the photoelectric system is an approach to test the field inhomogeneities and the correction. This data is taken while shining UV laser light on a pattern on the cathode. Only on the pattern, electrons are liberated by the UV light and drift through the TPC to the read-out. The reconstruction of photoelectric data [18] uses the standard MarlinTPC reconstruction chain up to the pulse level. Once the pulses have been found, the information from pads which have been associated with the projection of the photoelectric geometry are extracted for further analysis. Photo-dots are reconstructed using a likelihood method. It is possible to reconstruct photo-lines using another method.

Output from the reconstruction is imported into ROOT where analysis can be performed. Within ROOT, it is possible to generate displacement maps to analyze the drift of reconstructed photo-dot position for different positions of the TPC. It is also possible to estimate the ability to reconstruct the positions of the photo-dots using the standard deviation of the distribution of the repeated measurements of the x-coordinate of a photo-dot [19].

6 Simulation

The most recent part added to the simulation is the work on the photo-dot signal creation [20]. It is currently possible to simulate the creation of photo-electrons (assuming a

¹The PCMAG is a 1.25 T magnet that is used in the measurements with the Large Prototype TPC. Since it has no return yoke, it has a rather inhomogeneous field distribution. This can be used to test the algorithms for the correction of field inhomogeneities.

uniform distribution of UV light). The drift of these electron clouds can be simulated by evolving a Langevin model of ion drift using numerical methods. Magnetic field values are provided to the simulation by 2D or 3D field maps which were generated by adding measured corrections to a perfect dipole field.

There are also several ways to describe the signal creation by particles traversing a TPC. Three different simulation chains are currently implemented in MarlinTPC, differing in the level of sophistication and detail. The most simple is based on the smearing of a Geant4 Hit, with the addition of some electronics simulation to provide the most basic information. On intermediate level, the primary ionization is taken into account, from which charge clouds are drifted, diffused and amplified. The highest detail is provided by a simulation that simulates single electrons in almost every process. This level of detail is needed for instance for pixelized read-out, which can resolve primary charge clusters and even single electrons.

All the chains provide raw data in a format identical to the output of a prototype (or large experiment) read-out. This allows to validate the whole reconstruction chain, from raw data to reconstructed tracks. Recently the input format for all three chains has been made compatible with MOKKA [21] output, so it can be used to test the efficiency of the reconstruction algorithms for real physics cases.

The simulation has also been used to investigate the effect of back drifting ions from the amplification stage [22, 23]. Due to the beam structure of the ILC this will lead to so-called ion disks that drift –compared to the electrons very slowly– towards the cathode. These disks will impact the electric field homogeneity. Therefore it is important to understand this effect and its implications on the possible precision of a large TPC.

Further, there are also several dummy data generators, that simulate pulses or hits, which are mainly for code development purposes.

7 Conditions Data Handling

The MARLIN framework allows the access and handling of conditions data by a special processor, which is common to the whole framework. This *Conditions Processor* is the first in the linear chain and inserts the necessary data into the event stream. The relevant conditions data is then retrievable in any processor downstream in the processing chain. It can also be accessed via so-called *Change Listeners*, which update only, when new data is available and offer a higher flexibility in handling. The source of the conditions data are LCIO objects which can be read from a file or from a MySQL database via the LCCD toolkit.

In MarlinTPC a collection of conditions data classes exist, which are derived from the LCGenericObject data type. These are provided in the TPCCondData, which is automatically compiled and linked with MarlinTPC. The conditions data classes can also be compiled as a separate library without dependency on MARLIN. This allows to link it to a data acquisition program without introducing the full ILCSOft as dependency. The EUDET data acquisition (eudaq [24]) package implements this for the read-out of the Timepix chips to automatically write the chip conditions.

A setup consisting of two servers running MySQL databases to store and provide the conditions data has been set up at DESY. Further on the handling of the conditions data in MarlinTPC is currently under revision to use the latest improvements in LCCD [25].

8 Tools

This year, two projects of summer students were aimed to develop tools related to MarlinTPC. The first project is an event display called BBQ [26]. It is based on TEve from the ROOT framework, uses OpenGL hardware acceleration and supports picking and highlighting. Further, it offers full GEAR and LCIO support. The source code is available from a subversion repository [27].

The second project had the goal of developing a stable and maintainable conditions database browser. The available database browser is based on a command line tool from the CALICE collaboration and uses a TCL script to create a GUI interface. This script is hard to maintain and showed some instabilities and deficiencies in the daily work.

Therefore a rewrite based on C++ and QT was planned. Currently, the basic classes for the GUI interface are nearly ready, but the connectivity to the database and the handling of the condition objects is still missing.

9 Status and Near Future Plans

A short overview of the MarlinTPC software package was given. More information can also be found in previous EUDET publications [28], [29] and [30]. The recent developments comprise the ongoing revisions of the basic reconstruction chain up to the hit level and the conditions data handling. Further, a Kalman filter for track finding and fitting is available. In addition, a track finder based on Hough Transformations is existing and being extended by a fitting tool.

Progress has been made on understanding the necessary corrections and implementing the necessary functionality. A magnetic field map of the magnet used for the Large prototype measurements is available inside MarlinTPC. Also, measurements and simulations have been performed and reconstruction algorithms have been developed to investigate the inhomogeneities using the photo-dot technique. The effects of ion disks from the amplification stage have been studied using a simulation. It is planned to extend the simulation part of MarlinTPC further. In the near future, one important point of development will be to implement methods to correct for field inhomogeneities in the higher level reconstruction.

Work has also started to extend the current data correction and calibration methods, including for example alignment.

This year, data from an Asian test-beam effort at DESY using GEM amplification and pad read-out has been reconstructed using the full MarlinTPC chain up to the determination of the momentum resolution. Much of the used functionality has become fully available over the last year. The groups plan to shift their analysis to MarlinTPC.

Over the last few years, the MarlinTPC project has reached maturity and developments towards version 01-00 are nearing completion. The next steps will include the calibration and correction methods and the revision and extension of the build-in detailed simulation for TPC prototypes.

Acknowledgment

This work is supported by the Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RII3-026126.

References

- [1] LCTPC Homepage: <http://www.lctpc.org>
- [2] LCTPC Collaboration, *The Linear Collider Time Projection Chamber of the International Large Detector*, Report to the DESY PRC 2010, Section 2.2, Oct. 2010
- [3] MARLIN Homepage: <http://ilcsoft.desy.de/marlin>
- [4] LCIO Homepage: <http://lcio.desy.de>
- [5] LCCD Homepage: <http://ilcsoft.desy.de/lccd>
- [6] GEAR Homepage: <http://ilcsoft.desy.de/gear>
- [7] MarlinTPC repository (public access): <https://svnsrv.desy.de/public/MarlinTPC/>
- [8] ilcinstall Homepage: <http://ilcsoft.desy.de/ilcinstall>
- [9] MarlinTPC Homepage: <https://znwiki3.ifh.de/MarlinTPC/>
- [10] MarlinTPC mailing list: ilcsoft-MarlinTPC@desy.de
- [11] Linear Collider Forum, sub forum *Tracking & Vertexing*: http://forum.linearcollider.org/index.php?t=threadt&frm_id=39
- [12] Software issue tracker: <https://jira.slac.stanford.edu/browse/MAR>
- [13] TPC Data Model: <http://www.desy.de/~wiene/MarlinTPC/tpcdatamodel.pdf>
- [14] KalTest Homepage: <http://www-jlc.kek.jp/subg/offl/kaltest/>
- [15] ROOT Homepage: <http://root.cern.ch>
- [16] Christian Grefe, Diplomarbeit, *Magnetic Field Map for a Large TPC Prototype*, Jun 2008, <http://www-library.desy.de/cgi-bin/showprep.pl?desy-thesis-08-052>

- [17] Christian Grefe Presentation: Magnetic Field Map in MarlinTPC, presented on the MarlinTPC meeting, Thursday 15 Apr. 2010
- [18] Jason Abernathy Presentation: Photoelectric Reconstruction, presented on the MarlinTPC meeting, Thursday 19 Aug. 2010
- [19] Jason Abernathy Presentation: Magnetic Field Study via Photoelectric System, presented on the MarlinTPC meeting, Thursday 16 Sep. 2010
- [20] Jason Abernathy Presentation: Photoelectric Simulations, presented on the MarlinTPC meeting, Thursday 19 Aug. 2010
- [21] <http://polzope.in2p3.fr:8081/MOKKA>
- [22] Thorsten Krautscheid, Diplomarbeit, *Simulationen zur Ionenrückdrift in einer Zeit-Projektionskammer am ILC*, Aug 2008, in German, <http://lhc-ilc.physik.uni-bonn.de/thesis/krautscheidthesis.pdf>
- [23] Thorsten Krautscheid, Presentation: Simulation on the ion backdrift in a large TPC for the ILC, presented on the 114th workpackage meeting of LCTPC, Sep 2010
- [24] <http://projects.hepforge.org/eudaq/>
- [25] Steve Aplin, Presentation: LCCD Status, presented on the EUDET Annual Meeting 2010, Sep 2010
- [26] Ching Bon Lam, Presentation: BBQ, A TPC event display, presented on the 112th workpackage meeting of LCTPC, Aug 2010
- [27] BBQ subversion repository: <https://svnsrv.desy.de/public/bbq/trunk>
- [28] J. Abernathy et. al., *MarlinTPC: A Marlin based common TPC software framework for the LC-TPC collaboration*, EUDET-Report-2007-04
- [29] J. Abernathy et. al., *Latest developments in the MarlinTPC software package*, EUDET-Report-2008-09, 2008
- [30] J. Abernathy et. al., *Update on the Status of MarlinTPC*, EUDET-Memo-2009-16, 2009