



MPHYS Interim Report

Alex Colvill

02/06/2013



Introduction

Chapter 1 describes the facility where this research year is taking place, the Thomas Jefferson National Accelerator Facility (or Jlab), in Newport News, Virginia, USA. Chapter 2 describes the work completed so far. In short, the initial aim of this research is the design and coding of the reconstruction software for the Forward Time of Flight (FTOF) subsystem of the upgraded Continuous Electron Beam Accelerator Facility Large Acceptance Spectrometer (CLAS12).

Chapter 1 - Overview of Jlab and the CLAS12 particle detector

Jlab is a US Department of Energy owned facility, the centerpiece of which is a 7/8 of a mile long racetrack-shaped electron accelerator called CEBAF, or Continuous Electron Beam Accelerator Facility, which supplies high quality, optionally polarized, beams of electrons to multiple experimental halls, all of which can run simultaneously. Within each experimental hall, the electrons collide with a target (or are converted to photons that collide with a target), and the ensuing debris is recorded by a particle detector. The target is optionally polarized and is typically made of a low Z material such as Hydrogen, Deuterium, or Helium, but higher Z targets such as Carbon are used as well. CEBAF is currently in the middle of an upgrade, the main purpose of which is to increase the maximum beam energy from 5.7 GeV to 11 GeV. With this increased energy, several key scientific benefits emerge [1], namely:

- (1) It will enable three-dimensional imaging of the nucleon, revealing hidden aspects of its internal dynamics.
- (2) It will complete our understanding of the transition between the hadronic and quark/gluon descriptions of nuclei.
- (3) It will test definitively the existence of exotic hadrons.
- (4) Through the use of parity violation, it will provide low-energy probes of physics beyond the Standard model.

The layout of CEBAF prior to the upgrade is shown in Fig. [1], below.

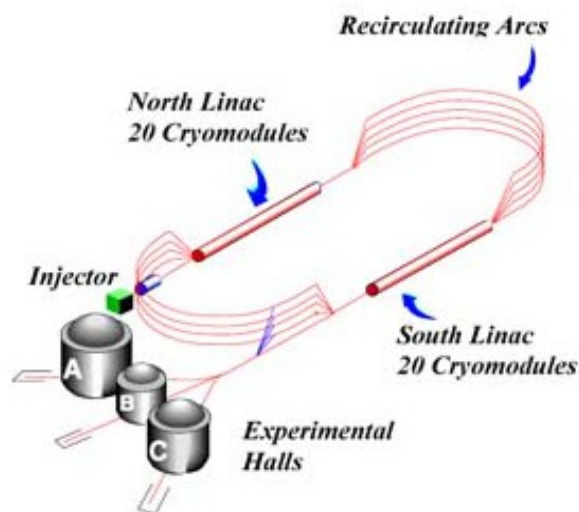


Fig. [1] – Layout of CEBAF prior to the upgrade. [3]

In brief, CEBAF works as follows : the injector generates a low energy stream of electrons which are divided into 'beam buckets', about 2 nanoseconds apart, and about 2 picoseconds long. These beam buckets pass into a linear accelerator, or linac, where they are accelerated in superconducting radio-frequency cavities (eight of which are grouped into a cryomodule, with twenty cryomodules making a linac). At the end of the linac, they are bent round a recirculating arc by powerful magnets, then pass into another linac on the other side of the track. In this manner, electrons complete up to five laps of the track, gaining in energy on each lap, at which point they are diverted to the desired experimental hall to collide with a target.

Alterations to CEBAF as part of upgrade are highlighted in Fig.[2], below.

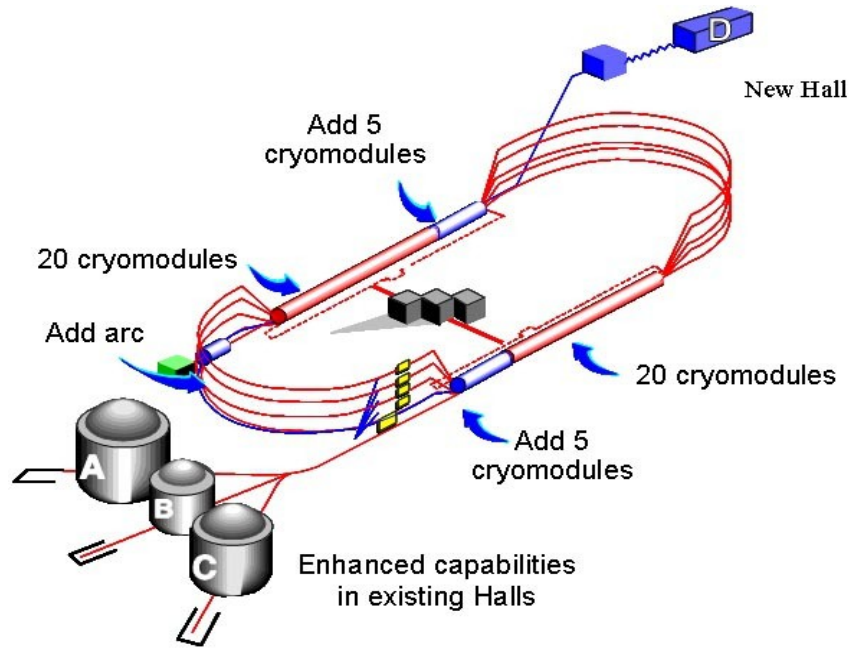


Fig. [2] – layout of CEBAF after the upgrade. [4]

As shown, the main alterations are: the addition of 5 cryomodules to each of the linacs, a new bending arc and a new experimental hall, hall D. The existing experimental halls, A, B and C are also being upgraded. The upgrade to Hall B is the focus of this research. It will contribute to key scientific benefits (1) and (2), as listed on the previous page.

Overview of the CLAS12 detector

As part of the upgrade, the old detector in Hall B, known as CLAS, or CEBAF Large Acceptance Spectrometer, described in [2], is being replaced by the CLAS12 detector, shown in Fig. 3-4 below. The term large acceptance is used to indicate that a detector is able to detect particles at almost all solid angles.

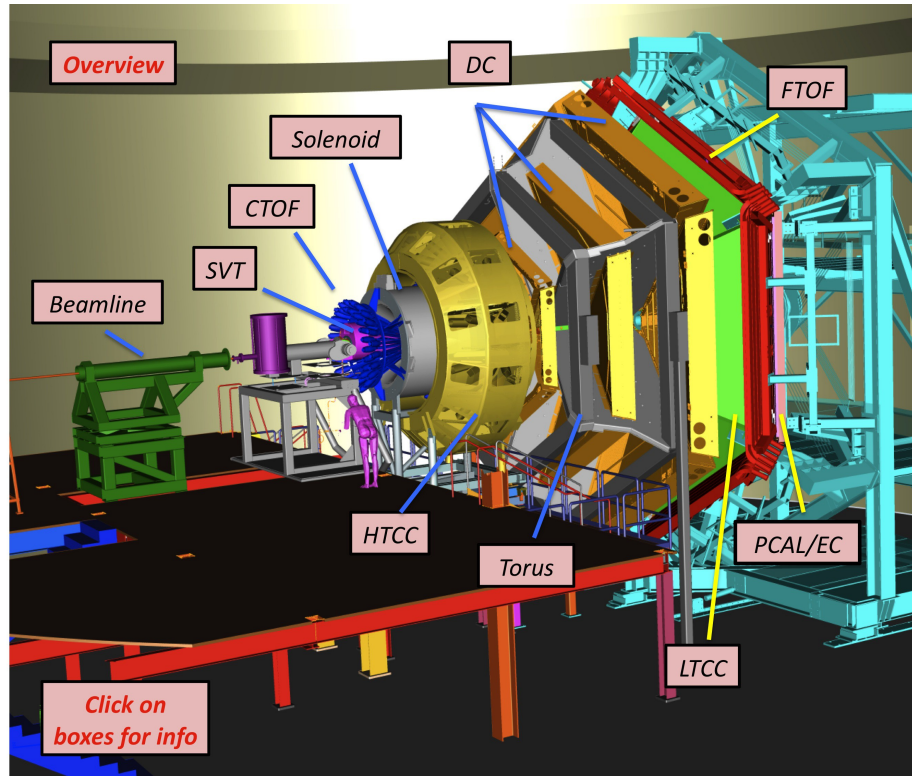


Fig. 3 – CAD drawing of the CLAS12 particle detector. The electron beam enters from the left.
Interactive version available at Ref. [11]

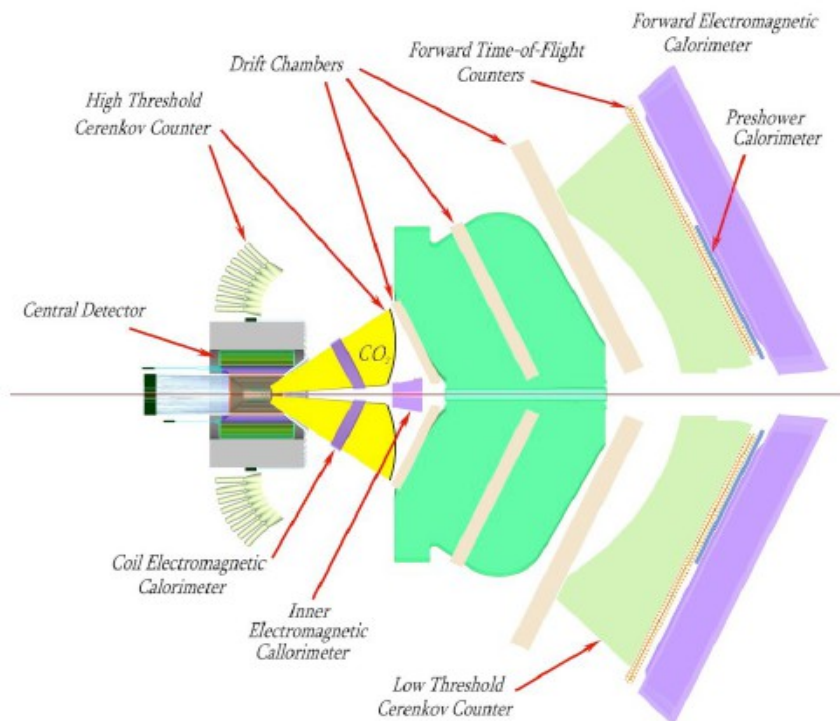


Fig. [4] – Cut away drawing of the CLAS12 particle detector. We are looking down on the detector which has been cut in half horizontally along the beamline. The beamline enters from the left. [5]

Compared to CLAS, the CLAS12 detector will be able to handle a ten fold increase in luminosity, at $10^{35} \text{ cm}^{-2} \text{ s}^{-1}$, and will offer improved acceptance and particle detection capabilities at forward angles. [12]

What follows is a highly simplified description of CLAS12, completely neglecting some components altogether. It is based on the design document [5]. At the highest level, the CLAS12 detector is divided into two main components, the central and forward detectors. The forward detector is based on, and components are being reused from, the CLAS detector. The central detector is new to CLAS12.

CLAS12 Central Detector

Cylindrical in shape, the central detector sits physically close to, and is centered on, the target. Its purpose is to detect particles with polar scattering angles greater than 35 degrees (the polar angle is measured vertically up from the beamline). It consists of a solenoid magnet, for momentum analysis, a Silicon Vertex Tracker and a Barrel Micromegas Tracker, for tracking purposes, and a Central Time Of Flight (CTOF) scintillation detector, for precise timing measurements. The CTOF may be relevant to this project at a later date, so will be described in more detail.

Central Time Of Flight (CTOF)

The CTOF detector is shown in Fig [5] below. It consists of 50 scintillation counters, each 66cm long and $3.5 \times 3 \text{ cm}^2$ in cross section. The counters are made from the plastic scintillator Bicron 408 and are located within the solenoid magnet, at a radius of 25 cm from the beam axis. The CTOF detector works as follows: when an ionizing particle passes through a counter, some of its energy is converted to light. This light is transferred along acrylic light guides attached to both ends of a counter. Attached to the end of the each light guide, in an area of lower magnetic field, is a photomultiplier tube (PMT). The PMT converts the light into an electrical signal which is discriminated, digitized and stored. The estimated resolution of the CTOF is 50 ps, which allows for the separation of pions from kaons up to 0.64 GeV, and pions from protons up to 1.25GeV. The CTOF system is also used as part of the trigger; that is, it is used as part of the system that decides when to record data and when to ignore it. In general, knowing the TOF of particles helps in the particle identification process; see section on FTOF, below.

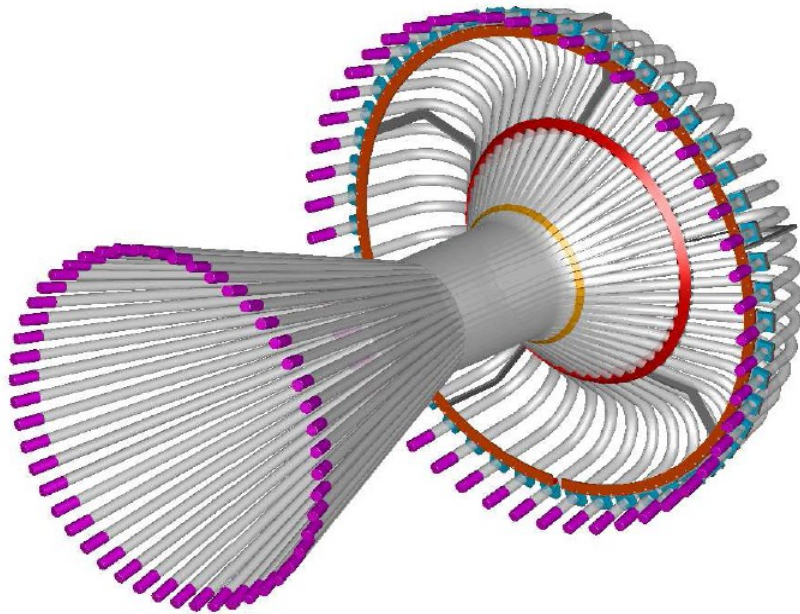


Fig. [5] - The CTOF subsystem of the CLAS12 particle detector shown in isolation. The barrel shaped central section is composed of 50 scintillation counters. Attached on either end of a single counter is a light guide, curved at the front, straight at the back. The purple on the end of each light guide represents a PMT. [5]

CLAS12 Forward Detector

The forward detector is located downstream from the target. Its purpose is to detect particles at polar angles of approx. 5-45 degrees. It is divided into six triangular sectors which circumnavigate the beam line. This rotational symmetry can most easily be seen in Fig. [6], on the next page. The subsystems of the forward detector are listed below, in the order a particle would meet them when traveling downstream from the target.

High Threshold Cherenkov Counter (HTCC)

The HTCC detects particles using the Cherenkov effect, the emission of light by charged particles moving faster than the local speed of light in a medium, which in this case is CO₂. In CLAS12, this effect is necessary to differentiate between electrons and pions.

Drift Chamber + Torus

Outside the HTCC are 3 regions of drift chambers, large gas filled chambers containing thousands of thin, high voltage, wires placed in a regular pattern. When a charged particle passes through a chamber it can knock electrons free from the gas. The electrons are attracted to the wires and when they arrive at a particular wire, this creates an electrical signal, which is amplified and recorded. By determining which wires had signals, the track of the particle can be determined. Moreover, due to presence of a high magnetic field in Region 2 of the drift chambers, the momentum and charge of the particle can be deduced. The magnetic field is supplied by a torus magnet made from six superconducting coils arranged symmetrically around the beam line. The field has a peak value of 3.6T, which is primarily in the azimuthal direction (azimuthal means to the left and right of the beamline, on a horizontal plain cutting through the beamline).

Low Threshold Cherenkov Counter (LTCC)

Outside region 3 of the drift chambers, another Cherenkov counter is used to provide pion / kaon discrimination. The gas in this case is C₄F₁₀.

Forward time of flight (FTOF)

Outside the LTCC is the FTOF subsystem. It operates in a similar fashion to the CTOF and, as it is the focus of this research, is described in greater detail below.

Pre-Shower Calorimeter (PCAL) and Electromagnetic Calorimeter (EC)

Outside the FTOF are two layers of calorimeters, first the PCAL, then the EC, each composed of alternating layers of lead and scintillator. When particles hit the lead, they create showers of particles which in turn create light in the scintillators. Unlike the scintillation measurements in the TOF subsystems, the aim here is to determine the *full* energy of the particle, rather than the crossing time, or partial energy. The PCAL is new to CLAS12. It allows separation of single high energy photons from a π^0 decaying to two photons, which is a critical reaction for the intended physics program at Jlab [12].

Overview of FTOF

The FTOF subsystem of the CLAS12 detector is shown in Fig[6], below.

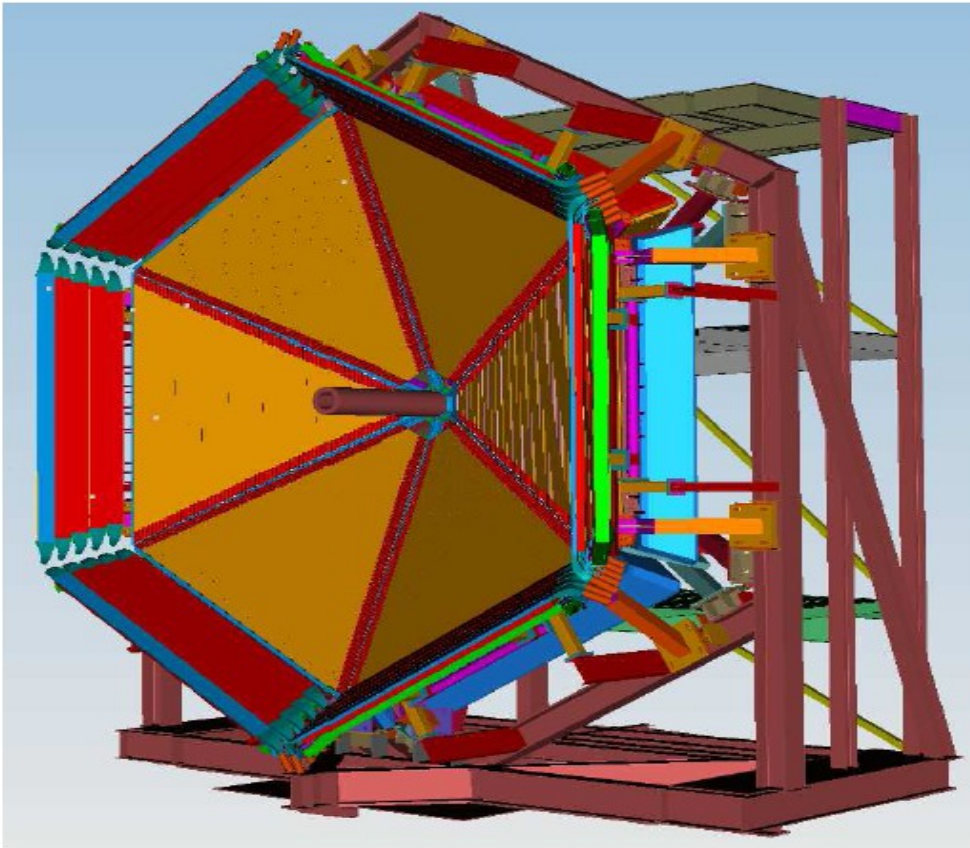


Fig [6] – 3D CAD drawing of the CLAS12 FTOF subsystem shown in isolation. [5]

The FTOF subsystem works using the same basic principle as the CTOF, EC and PCAL. CLAS FTOF is described in [6], designs for CLAS12 FTOF are found in [5]. In each sector of CLAS12, the FTOF subsystem consists of three sets of scintillation counters, called panels. Panel-1b is shown in orange in Fig [6] , above. It is located at forward angles of 5-36 degrees and consists of an array of 62 counters, each 6cm wide by 6cm thick, with a range of lengths from 32cm to 375cm. Located behind Panel-1b, and covering approx. the same area, is Panel-1a, which is being reused from the CLAS detector. It consists of 23 counters, each 15cm wide and 5.08cm thick. Both panels 1a and 1b are necessary to achieve the required timing resolution of 80ps. Panel 2, also being reused, is shown in red in Fig [6], above. It is located at forward angles of 36-45 degrees and consists of 5 counters, each 22cm wide and 5.08cm thick, with a range of lengths from 370cm to 430cm. Panels 3 and 4 from CLAS, located at even wider angles , are not being used in CLAS12. Sector 1 counter positions in the CLAS coordinate system are shown in Fig. [7], below. Sector 1 is the central sector on the left of Fig[6]. The CLAS coordinate system has its origin at the target, with its positive z axis along the beam line, positive y vertically up and positive x horizontally left, when looking downstream along the beam line. The x values should be shown as positive, not negative.

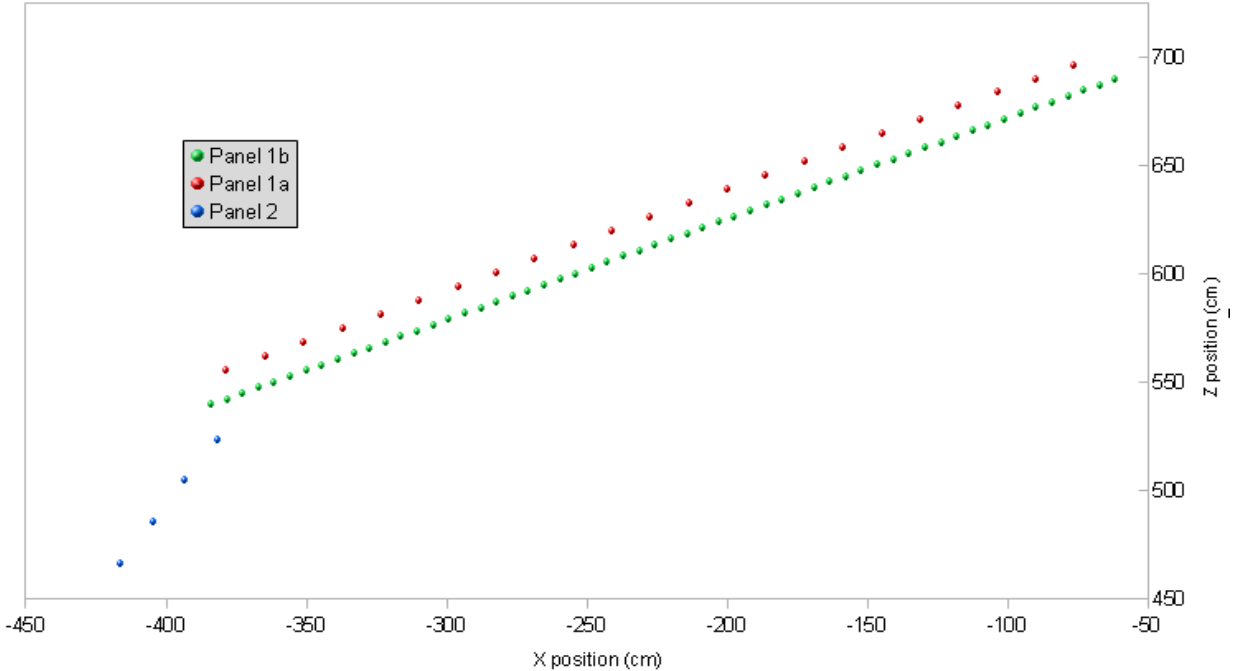


Fig. [7] – Centre positions of the CLAS12 FTOF counters in sector 1. We are looking down the y axis at the edge of the panels from above. The icons used are not representative of the width or thickness of the paddles. The x positions should be positive.

Like the CTOF counters, counters in panels 1a and 2, and the longer counters in panel 1b, are made from Bicron 408. Shorter counters in panel 1b are made from Bicron 404. Each counter is connected to a PMT at both ends, though the make of PMT, and the connection mechanism, varies with panel. Note that there is no need for long light guides here, as in case of CTOF, due to the low magnetic field. Each PMT is connected to an Analog to Digital Converter (ADC), and a Time to Digital Converter (TDC), to provide energy and timing output, respectively. It is the discriminated signals from the ADCs and TDCs that are the input to the FTOF reconstruction software, the focus of this research. Note that some details of the FTOF subsystem relevant to the reconstruction software have yet to be decided, namely the precise trigger and ADC readout methods; new ADCs are being used that could be read out several times in a given event.

The FTOF subsystem has a number of important uses when considered in isolation:

- (1) It allows separation of pions from kaons up to 2.6GeV and of pions and kaons from protons up to 5.6 GeV.
- (2) It is used as part of the level 1 trigger.
- (3) It provides an independent means for identification of slow particles (using energy deposited, rather than flight time).

More generally, when used in combination with other detector subsystems, FTOF, like CTOF, can be used to identify particles. Knowing the precise TOF of a particle, plus the path length (from tracking subsystems), it is possible to calculate the particle velocity. Knowing the velocity and the momentum (from the drift chambers, in the case of FTOF), gives you the mass, which gives you the particle ID. Ultimately, particle IDs are associated with the high level data that experimental physicists require, such as four momenta and scattering angles. It is the job of the reconstruction software to convert the raw detector data into this high level data.

Chapter 2 - Work completed so far

Extraction of a functional specification from the existing CLAS TOF reconstruction code

As the CLAS12 FTOF subsystem is similar to that used in CLAS, the main difference being the number and type of panels, a natural starting point in writing the CLAS12 software was to look at the CLAS software, written in the C programming language. Though the code is not well documented, the outline of the code is described in [7]. [8] describes the calibration variables used. Calibration variables accurately associate raw ADC and TDC values with their corresponding energies and times, respectively. The function of the existing CLAS TOF software can be summarized as follows :

Step 1: Read in calibration and geometry variables appropriate to the data run

Each data run (i.e. data from a period of operation with a specific detector configuration) has associated with it a set of calibration and geometry variables. Geometry variables describe the position and dimensions of the panels and paddles (the word paddle is interchangeable with counter, and will be used for the remainder of the document, sorry). These are read in from an external database and stored in memory.

Step 2: Convert raw ADC and TDC signals to energy and time, respectively

The raw data from a paddle consists of a left and right ADC and TDC value, in channels, plus an address so the paddle can be identified. The raw ADC values are first converted into energy in MeV, using **Eqn. [1]**, below, and the raw TDC values converted to time in ns using **Eqn. [2]**, below. Uncertainties in these values are also calculated. At this stage, left and right values remain separate. Paddle hits with no valid timing information are discarded.

$$\text{Energy in MeV : } \quad \mathbf{E = 10(A-P)/A_{NMIP}} \quad \mathbf{Eqn. [1]}$$

A – raw ADC value in channels, P – pedestal; ADC value with no signal is present, A_{NMIP} – pedestal-subtracted ADC value corresponding to a normally incident minimum ionizing particle (NMIP) which deposits 10 MeV in the scintillator material

$$\text{Time in ns : } \quad \mathbf{T = c_0 + c_1t + c_2t^2 + t_w} \quad \mathbf{Eqn. [2]}$$

t – raw TDC value in channels, c_0 , c_1 , c_2 - fitting constants, t_w – time walk correction, used to correct for the fact that signals of different strengths will pass a discriminator threshold at different times. Details of this correction have been emitted, see [7].

Step 3: Combine left and right times and energies

Using both left and right signals, the best possible combined hit time, energy, and position, relative to the centre of the paddle, are calculated. How this step is performed depends on the validity of the ADC and TDC data. Non valid data is out of a prescribed range. A status word is associated with the hit, so the validity of the data can be tracked going forward. If both left and right TDC and ADC signals are valid, then:

y Position: $y = v_R v_L (T_L - T_R) / (v_R + v_L)$ **Eqn. [3]**

$v_{R,L}$ – effective speed of light in scintillator when light is going left or right

Time: $T = (T_L + T_R) / 2 - y(v_R - v_L) / (2v_R v_L)$ **Eqn. [4]**

Energy: $E = \{E_L E_R \exp[y(\lambda_R - \lambda_L) / (\lambda_L \lambda_R)]\}^{1/2}$ **Eqn. [5]**

$\lambda_{L,R}$ - attenuation length leftwards, or rightward - the distance at which the intensity of a beam of particles has dropped to 1/e of its original value

If there is only one good TDC signal:

y Position: deduced from external tracking subsystems
 Time: $T = T_{L,R} - y / v_{L,R}$ **Eqn. [6]**

If there is only one good ADC signal:

Energy: $E = E_{L,R} \exp[y / \lambda_{L,R}]$ **Eqn. [7]**

If there is no TDC signal and both ADC signals:

y Position: $y = [\lambda_L \lambda_R / (\lambda_L + \lambda_R)] \ln(E_L / E_R)$ **Eqn. [8]**

Uncertainties in y position, time and energy are also calculated. Note that the X and Y positions of a hit are always assumed to be on the centre line of the paddle, as there is no way of knowing any different.

Step 4: Group adjacent related paddle hits into clusters

To deal with the fact that a single particle can leave signals in multiple adjacent paddles, the final step in the reconstruction is to combine hits into clusters. The clusters are chosen based on an overlap in space and time. If the difference in times or positions is less than three times the uncertainties in the differences, then the two hits are considered a cluster. If three consecutive counters fire in a given event, then the adjacent pair that are better matched in space and time are combined. (the other scintillator is stored as a single hit). If a cluster is found, time is computed from an energy weighted average of the two hits. X and Z positions are simple averages of individual hit positions. Y position is energy-weighted average. Output status word is the sum of the status word for the first hit in the cluster and 100 times the status word for the second hit in the cluster. The ID is the SCR ID for the first hit in the cluster. Uncertainties are calculated in all these combined quantities.

The output from the steps 2, 3 and 4 is stored in permanent data banks, called SC1, SCR, and SCRC respectively, defined in [7].

Conversion and extension of the existing CLAS FTOF reconstruction code

Once it was understood what the CLAS FTOF reconstruction code did, the next step was to convert it from C into Java, as C is unsupported in the CLAS12 era. In making this conversion, *which is largely completed*, the first step was to design the class model. In designing the class model, the guiding principle was one of separation between data and algorithms. In fact, a class model for another of the detector subsystems - the Electromagnetic Calorimeter (EC) - had already been written according to this principle, so this formed the basis of the TOF model. Where relevant, the EC object model was copied wholesale.

By copying the EC class model, the FTOF code was automatically able to run in the ClaRA software framework (CLAS12 Reconstruction and Analysis Framework) [10], in which all the CLAS12 software components must ultimately run. In this context, each detector subsystem software is known as a service, an atomic, self-contained piece of software, which is similar in definition and functionality to a SOA (Service Oriented Architecture) service. Services interact with other services only through a well-defined interface. The full reconstruction chain is built from multiple services, each being sent an input by another service, performing their processing on this input, then forwarding onto the next service. Reconstruction data is sent between services in the Event Input Output format (EVIO). From a programming point of view, turning a piece of software into a ClaRA service simply means implementing the interface of a specific class.

In designing the class model, it was crucial that not only the relevant existing CLAS functionality remained, but that it could easily be adapted and extended for CLAS12. In practice, this meant that, due to the existence of overlapping panels (panels 1a and 1b overlap entirely and panel 2 overlaps marginally with panel 1b), it would be necessary to allow for the clustering of hits *between* panels as well as on a single panel. It is likely that much of the work of the research year will be spent figuring out how to do this, but dummy methods were built into the class model from the start.

An outline of the class structure can be found in Appendix [2].

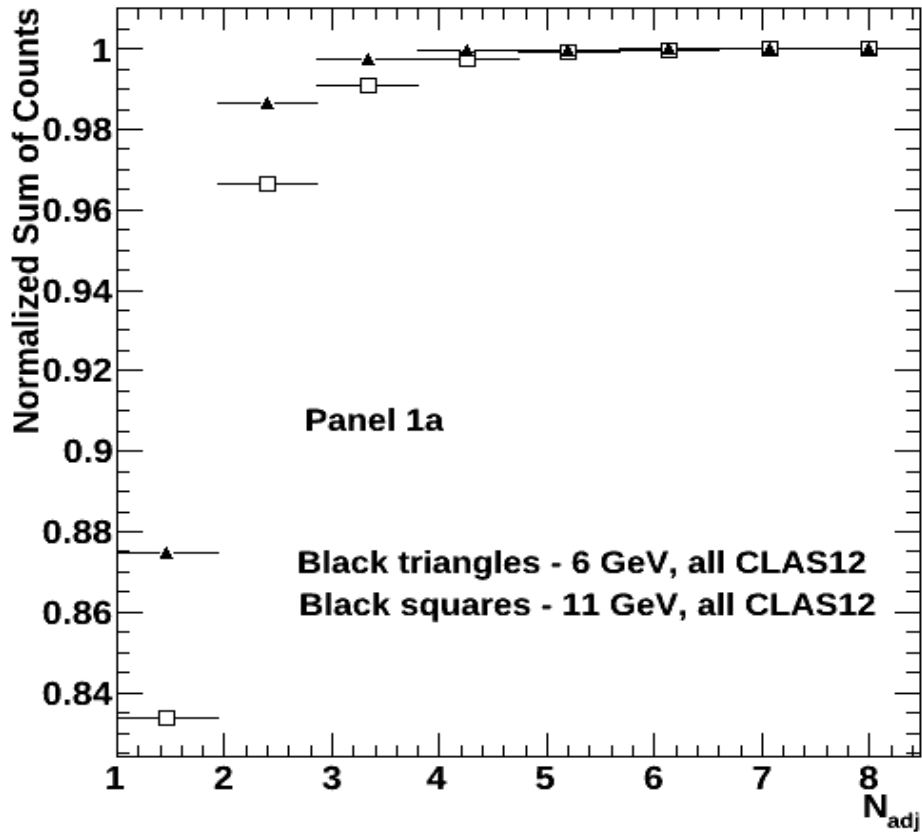
Testing the software

As the CLAS12 detector is not yet operational, it is necessary to rely on simulation to provide a realistic input to the reconstruction for test purposes. The simulation used is called GEMC, or GEANT4 Monte Carlo (GEANT4 is a software library for the simulation of particles passing through matter). Within GEMC, the CLAS12 detector exists in virtual form. Given a set of input particles specified in LUND format [13], it can calculate the path these particles take in the detector, including the time of, energy deposited by, and position of, the particles when they hit the FTOF paddles. It then digitizes these parameters, approximating how the real ADC and TDC operate. The output format of GEMC is given in Appendix[1]. In practice, this digitization is not sophisticated enough, so instead, the FTOF software performs the digitization itself, using a reversed version of the reconstruction algorithm. Although formal unit testing is yet to be performed, it has been verified that going backwards and forwards through the algorithm give results that are as close to identical as would be expected, given the loss of resolution due to digitization. It has also been verified that GEMC and the FTOF software have identical geometry.

Adaptation of single panel clustering algorithm for CLAS12

Most recently, work has begun on GEMC simulations to decide if the CLAS clustering algorithm needs to be altered for CLAS12. As a reminder, it currently looks at three adjacent paddle hits, and creates a

cluster from the two that match the best in space and time. Given that the beam energy has doubled, and that panel 1b paddles are significantly narrower (6cm) versus panel 1a (15cm), it seems logical that it might be possible for a single particle to hit more than 3 adjacent paddles. To this end, a series of simulations have been run that measured the frequency with which N adjacent paddles were hit on a given type of panel. Results for panel 1a are shown in Graph [1], below.



2013-05-26 22:22:18

Graph[1] – comparison of how often N number of adjacent paddles (N_{adj}) are hit at CLAS (6GeV) versus CLAS12 (11GeV) beam energies, with all detector components present. Author: Jerry Gilfoyle.

Our initial conclusion is that it may be necessary to increase the number of adjacent paddles we look at from 3 to 4 on panel 1a, though this is still under discussion. We reached this conclusion as the normalized sum of counts at 6 GeV and $N_{adj} = 3$, is the same as the normalized sum of counts at 12 GeV and $N_{adj} = 4$. (ran out of time to explain this better, sorry)

Summary of work completed so far

A functional specification of the existing CLAS6 TOF reconstruction code has been written. The bulk of this has been converted into a working Java program that operates as a service in the ClaRA framework. Work has begun on simulations to decide if to, and how to, alter the clustering algorithms on a single panel.

Future work

- Decide how many hits to combine, and how to combine hits, when clustering on a single panel.
- Decide how to match and combine clusters between panels.
- Implement these decisions.
- Integrate the FTOF reconstruction service with FTOF geometry and calibration services, and external tracking services.
- Possible alterations due to trigger and ADC readout methods.
- Possible implementation of the CTOF reconstruction, which will be similar to the FTOF reconstruction.

References

- [1] The US Nuclear Science Advisory Committee, 'The Frontiers of Nuclear Science: A Long Range Plan' (2007)
- [2] B. Mecking, *et al.*, (The CLAS Collaboration), 'The CEBAF Large Acceptance Spectrometer', Nucl. Instr. and Meth., 503/3, 513 (2003).
- [3] C.W. Leemann, W. Christoph, D. R. Douglas, and G. A. Krafft, "The Continuous Electron Beam Accelerator Facility: CEBAF at the Jefferson Laboratory.", Annual Review of Nuclear and Particle Science, 51.1, 413 (2001)
- [4] www.jlab.org/12-gev-upgrade accessed on 01/06/2013
- [5] 'CLAS12 Technical Design Report', version 5.1, www.jlab.org/Hall-B/clas12_tdr.pdf, accessed on 01/06/2013
- [6] E.S. Smith, *et al.*, 'The time-of-flight system for CLAS', Nucl. Instr. and Meth., A432, 265 (1999)
- [7] S.J. Taylor, 'Radiative Decays of Low-Lying Excited-State Hyperons', Ph.D. Thesis, Rice University, USA (2000) available at https://clasweb.jlab.org/wiki/index.php/CLAS12_Time-of-Flight_Software
- [8] K. Loukachine, 'Electroproduction of the $\phi(1020)$ Vector Meson at 4GeV', Ph.D. Thesis, Virginia Polytechnic Institute and State University, USA (2000)
- [9] G.F. Knoll, 'Radiation Detection and Measurement', 3rd edition, Wiley (2000)
- [10] V. Gyurjyan, *et al.*, "CLARA: A Contemporary Approach to Physics Data Processing.", Journal of Physics: Conference Series, 331/3 (2011)
- [11] <http://www.jlab.org/Hall-B/clas12-web/> accessed on 01/06/2013
- [12] 'CLAS12 software document; Hall B 12 GeV upgrade', <http://www.jlab.org/Hall-B/clas12-software.pdf> accessed on 01/06/2013
- [13] https://gemc.jlab.org/gemc/Documentation/Entries/2011/3/18_The_LUND_Format.html accessed on 01/06/2013

Appendix [1]

Format of data output by GEMC simulation software:

Type : 0 = integer , 1 = double

Activated : 0 = not in the output , 1 = in the output

| name | id | type | activated | description |
|-------------|-----------|-------------|------------------|---|
| ETot | 1 | 1 | 1 | Total Energy Deposited |
| | 2 | 1 | 1 | Average global x position |
| | 3 | 1 | 1 | Average global y position |
| | 4 | 1 | 1 | Average global z position |
| | 5 | 1 | 1 | Average local x position |
| | 6 | 1 | 1 | Average local y position |
| | 7 | 1 | 1 | Average local z position |
| | 8 | 1 | 1 | Average time |
| pid | 9 | 1 | 1 | Particle ID |
| vx | 10 | 1 | 1 | x coordinate of primary vertex |
| vy | 11 | 1 | 1 | y coordinate of primary vertex |
| vz | 12 | 1 | 1 | z coordinate of primary vertex |
| E | 13 | 1 | 1 | Energy of the track at the entrance point |
| mpid | 14 | 1 | 1 | Mother Particle ID |
| mvx | 15 | 1 | 1 | x coordinate of mother vertex |
| mvy | 16 | 1 | 1 | y coordinate of mother vertex |
| mvz | 17 | 1 | 1 | z coordinate of mother vertex |
| sector | 20 | 0 | 1 | ADC Left |
| paddle | 21 | 0 | 1 | ADC Left |
| ADCL | 22 | 0 | 1 | ADC Left |
| ADCR | 23 | 0 | 1 | ADC Right |
| TDCL | 24 | 0 | 1 | TDC Left |
| TDCR | 25 | 0 | 1 | TDC Right |

Appendix [2]

Package and class structure

The current class model, which implements the bulk of the CLAS6 functionality, and can run as a ClaRA service, is divided into the following packages:

org.jlab.clas12.ftof.detector

Provides very basic classes to maintain information about the detector, including the class `Panel`, which holds information about the properties of all the panel types, 1a, 1b and 2.

org.jlab.clas12.ftof.calibration

Provides classes to store calibration data. The calibration data has been divided into three types. A `DetectorParameter` only has one value for the entire detector. A `PaddleParameter` has a unique value per paddle. An adjustment parameter is used to change a `PaddleParameter`'s value when it is first read from the database, but is subsequently not stored. Currently all parameter values are hard-coded, but will, at some point, be retrieved from an external calibration service.

org.jlab.clas12.ftof.geometry

Provides classes to store geometry data. The hierarchy is `DetectorGeometry->SectorGeometry->PanelGeometry->PaddleGeometry`. Each of these classes stores geometry data relevant to its level in this hierarchy. Currently all the values are hard-coded, but will, at some point, be retrieved from an external geometry service.

org.jlab.clas12.ftof.event

Provides classes to store the data of the event being reconstructed. The hierarchy is *EventData->SectorData->PanelData->Paddle->Hit*. *EventData* contains all the data of a single event reconstruction. It has a list of *SectorData* objects, unique to a sector, which in turn has a list of *PanelData* objects, unique to a *Panel*. Each *PanelData* object has a list of *Paddle* objects. A *Paddle* object is where the corrected energy and time from the left and right PMTs of a real paddle is stored. A *Paddle* is only created if there is valid timing information from one TDC. A *Paddle* has a *Hit* object that stores the combined left and right PMT times and energies. Groups of two *Hits*, called *Clusters* are stored in *PanelData*, and, tentatively, combined *Clusters* between panels 1a and 1b, called *SuperClusters* are stored in *SectorData*. *SuperClusters* contain the information you want the program to output.

org.jlab.clas12.ftof.event

Provides classes to perform the reconstruction algorithm by manipulating the event classes just described. *PaddleCreator* constructs the *Paddle* objects by getting the input from the FTOF EVIO input bank. It saves each *Paddle* into the correct sector and panel. A sector of the FTOF detector is reconstructed with the *SectorReconstruction* class, that in turn uses *PanelReconstruction*, which in turn uses *PaddleReconstruction*. *PaddleReconstruction* involves combining the PMT data on both sides of

a Paddle to create a *Hit*. *PanelReconstruction* involves finding and making *Clusters* of *Hits* between the paddles of a panel. *SectorReconstruction* involves combining the *Clusters* from panels 1a and 1b in a sector. It is unspecified at this stage exactly how it does this, and what the output is (currently called *SuperClusters*). Reconstructed data is turned into output EVIO by the *FTOFOutputCreator*.

org.jlab.clas12.ftof.services

Currently only one service is defined – *FTOFReconstruction*. It reconstructs a single event in the FTOF. It takes an EVIO input bank, runs the reconstruction algorithms just described and appends the output bank to the input EVIO bank.