CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

5 June 2009

Synchronization of CSC Trigger Primitives Arriving at the CSC Track Finder

D. Acosta, G. P. Di Giovanni, J. Gartner, D. Holmes, B. Jackson, A. Korytov, K. Kotov, A. Madorsky, L. Uvarov, Dayong Wang[a)]

University of Florida, Gainesville, Florida, USA.

J. Gilmore

The Ohio State University, Columbus, Ohio, USA

J. Hauser, M. Ignatenko, G. Rakness

University of California at Los Angeles, Los Angeles, USA

F. Geurts, M. Matveev, P. Padley, R.Redjimi

Rice University, Houston, Texas, USA

Abstract

The CSC Track Finder can provide inportant timing information for the inter-chamber synchronization of the CSCs. This note describes the procedures developed for analyzing and monitoring the inter-chamber synchronization status of the CSC system during the two phases of the CMS Magnet Test and Cosmic Challenge (MTCC), minus end Slice Test and the subsequent local/global runs underground, including Beam commissioning in September 2008 and Cosmic Run At Four Tesla (CRAFT). The software implementation, results and limitations of the methods, and how to achieve LHC collision synchronization are also presented.

[a)] Contact person, Email: Dayong.Wang@cern.ch

1 Introduction

1.1 CSC Track Finder in MTCC, Minus End Slice Test, Local Runs and Global Runs

Cathod Strip Chamber Track Finder (CSCTF)[1] provides triggers for the endcap muon dector for CMS[2]. It plays an important role in the commissioning activities.

Synchronization of CSC chambers is important for proper trigger performance. In order for CSCTF to function more effectively, efforts are paid to make the trigger primitives synchronized at CSCTF, more specifically, at the input of Sector Processor (SP).The requirements for synchronization is: DAQ paths need to be timed-in to the trigger for stable readout; Trigger signal should be stable in time with respect to other subsystems; It should not depend too much on where in sector the muons crossed.

In order to test the system and gain experiences for future commissioning, CSCTF entered two phases of MTCC and the following "minus end Slice Test", then the local and global runs. From the cosmic ray data accumulated, we can extract timing information, then adjust the related parameters to improve the synchronization. This note describes the procedures we developed and some results from it.

There are already notes for CSC Track Finder in MTCC. You can find the description of the setup there[3].

For minus end Slice Test, please refer to the "CSC Slice Test" twiki page [4] and links therein. For the following CSC local and global runs, please refer to the twiki page. For the summary of the whole CSC synchronization efforts, please refer to Reference [5].

1.2 Overview of CSC Synchronization

The synchronization of the CSC trigger requires timing alignment in three sequential steps: alignment of the signals from the anode front end boards with each other, maximizing the coincidence of the anode with the cathode strips to make a Local Charged Track (LCT) within a chamber, and alignment of LCTs from different chambers at the Sector Processor (SP). On the path of the trigger primitives from the Front-End Boards (FEBs) to the SP, several set of delay parameters are needed to synchronize the CSC trigger primitives at different locations, namely: the AFEB fine delays, the ALCT-CLCT match delay, the MPC output delay, and the SP Alignment FIFO delay[6].

The alignment of LCTs at the SP arising from synchronous muons that pass through different chambers is factorized into two components, the arrival of the LCTs at the MPCs, and the propagation of the signals from the MPCs to the SPs over different optical fiber lengths.

The variation of optical fiber lengths is corrected for automatically in the firmware of the SP FPGA. At the beginning of every run, a "Resync" signal is sent from the CMS Timing, Trigger and Control (TTC) system to all triggering subsystems. TTC commands are sent over optical fibers to Clock and Control Boards(CCB) which distribute the commands to the electronic components in the CSC, including the MPC and the SP. Upon receipt of the Resync command, each MPC sends a stream of signals to the SP. The FPGA in the SP contains an "Alignment FIFO", which automatically lines up these streams of signals from the 5 MPCs in each sector. The Alignment FIFO corrects for differences in the combined fiber lengths of TTC-CCB and MPC-SP. There is a delay on the TTCrq mezzanine board on the CCB to correct for the TTC-CCB fiber lengths to synchronize TTC commands at the CCBs. This delay is set according to the measurements of the fibers. Mistakes in this delay will result in mistakes in the LCT arrival at the SP.

1.3 Motivations and Backgrounds of This Study

We need to time together trigger primitives as they arrive at the CSCTF. Only after that can combined triggers be made.The aim of the offline analysis of the Sector Processor data described here is thus to align the LCTs at the MPCs.

Two trigger modes were used for CSC's. In the first case, a trigger signal is based on a single valid pattern in a chamber, that generally means ALCT and CLCT are coincident at TMB level ,but ALCT coincidence requirement sometimes varied:3/6 or 4/6. It is the case with widest muon acceptance. For the second case, a coincidence of 2 chambers in 2 disks generate a trigger. It was usually configured to have a wide road, but can be tightened to be pointing to the collision region.

The SP captures 7 consecutive bunch crossings (BX) worth of incoming LCT data, relative to when an L1A was received with some offset. So we can directly see if incoming segments from different chambers and disks are in time with respect to each other.

Figure 1 shows one typical distribution of relative timing of different CSC chambers. It came from CSCTF Data Quality Monitor (DQM)[7], a software program for monitoring the quality of the DAQ data from the CSC Track Finder crate.

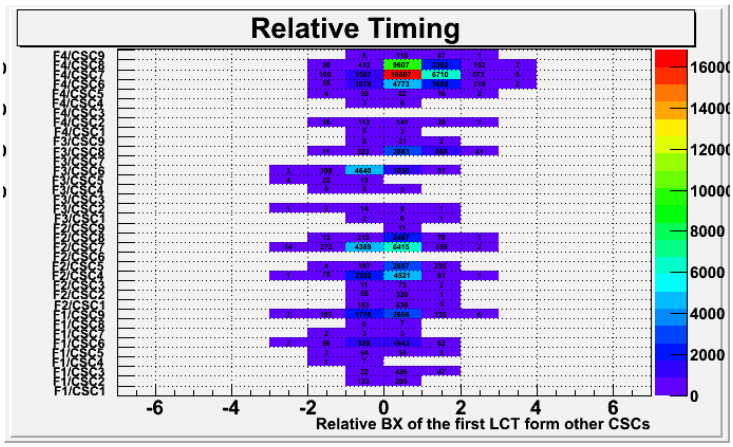

Figure 1: The relative timing of different CSC chambers shown by CSCTF DQM. This plot shows the bunch crossing arrival time of trigger primitives from CSC chamber ME+2/2/29 relative to those from all other chambers in Global Run 2550 during MTCC-I (Aug.25, 2006.

During the phase I of the MTCC, we knew:

- If good synchronization is reached at CSC peripheral crates, the major cause of asychronization at input of SP comes from the cable length differences from TTC to CCB and that from peripheral crates to TF[6].
- Time Of Flight(TOF) effects really enter. The goal of CSC synchronization with cosmic ray muons is to guarantee the muons simultaneously arriving at different chambers to be triggered and read out at the same time.
- There will be much smaller statistics from the non IP-pointing muons than those from the correct direction, because AFEB patterns already restrict the muon direction to be from IP.

During early stage of MTCC, we must determine relative timing of chambers one by one by hand from TF DQM plots(such as the one shown in Figure 1) and offline analyses. We need walk around rings, then intercalibrate rings in one disk to those in another. In principle, we can thus time-in all chambers in a sector to some absolute reference,and then make adjustments. In order to facilitate the process, we first develped the algorithms described in this note during MTCC stage II to digest and offsets chamber by chamber, automatically.

The results from this method was first used during minus end slice test in spring 2007, then it evolved some extent in the following monthly global runs, local runs underground and two phases of CRUZET.

1.4 The structure of this note

The structure of this note is as following: Sec. 2 will focus on the details of algorithm itself. The various tests/verifications with 18 chambers in one sector of ME2 and ME3 are described in the following Sec. 3. More descriptions and results related to the expansion of the algorithm to the full trigger sector will be in Sec. 4. Retrieving results from the whole endcap and issues of combining results from different runs are detailed in Sec. 5. Sec. 6 described the timing results obtained from the first LHC beam commissioning in September 2008. Sec. 7 describes how we should achieve the CSC inter-chamber synchronization for collision beams. Sec. 8 is the summary and outlook.

The appendices provides more details of certain issues mentioned in the body. Appendix A lists the two routes for the naive estimation described in Sec. 2.2.3. Appendix B described the details of the software and dataset for this analysis. Appendix C details the signal propagation estimation mentioned in Sec. 4.3. Appendix D lists the input combinations for one trigger sector, as described in Sec. 2.2.1. Appendix E is a recipe for the whole procedures of running the algorithm over CSCTF dataset and miscellaneous conventions used in the program.

2 Description of the Methods and the Algorithms

2.1 Overview

LCTs at the SP are tagged by the bunch crossing (BX) in which they arrive. For every neighboring chamber pair, the difference in bunch crossing is histogrammed for tracks which contain LCTs from both chambers. The distributions with the highest probabilities of overlap are analyzed to extract the mean of BX difference and error on the mean. With these numbers, we construct a global chi-square, minimization of which can yield optimal timing constants for each chamber simultaneously. Then the values with reverse sign are just the suggestions for delay setting changes from this program. The changes are implemented in approximately 2.2ns steps using AFEB fine delays and in BX steps using an adjustable delay from TMB to MPC.

The basic algorithm is on sector-by-sector basis, i.e. it deals with input data from one sector of all four stations at a time. The inter-sector synchronization can also be easily handled with similar methods. The global results for the 234 chambers in one endcap can be produced by combining the sector-by-sector results with inter-sector results.

To retrieve the optimal information from all the available measurments, we first extract the mean and their error from each histograms. Assign a correction factor to each chamber,then construct a global χ^2 using all favored combinations as:

$$\chi^2 = \sum_{kl} \left(\frac{\Delta_k - \Delta_l - m_{kl}}{\sigma_{kl}} \right)^2 \tag{1}$$

Here, k, l are labels for different chambers under study, Δ is the timing shift between the two channels, m is the measured BX difference and and σ is its measurement error. All these quantities are in unit of bunch crossings(BX). Minimization of the χ^2 defined in Eq. 1 can yield optimal timing shifts for each chamber simutaneously.

This least square problem above can be solved by several different methods. For the purpose of crossing check, both methods of matrix inversion and numerical minimization were implemented.

2.2 Sector-by-sector algorithm

The basic algorithm can handle the data from 39 chambers in one trigger sector together. We focus on only the most important aspects in this subsection.

2.2.1 Input Combinations

There are 118 combinations used for 39 chambers in one sector. The details can be found in Appendix D.

Figure 2 shows distributions of BX differences of tracks penetrating ME1/1 and ME2 as an example of these input combinations.

2.2.2 Error handling

From the input combinations described in 2.2.1, the mean values of BX differences and errors on the means are extracted to serve as input. But the errors thus obtained only reflects the statistical ones. There are also systematical errors that must be considered due to the following reasons:

- The errors of means are based on the assumption of Gaussian distributions, but in real world they are not. There are always some tails in the distributions.
- The results we get is some float numbers while the BX differences are discrete integer values in each event.
- Some other practical effects.

To better handle these errors MTCCII data were carefully studied and the following parametrization was found to be able to describe the measurement errors in total:

$$\sigma_{total} = \sqrt{\sigma_{stat}^2 + C_{sys}^2} \tag{2}$$

Here, σ_{stat} is the error of mean values obtained from the measurements, and the C_{sys} term is used to describe all the systematic effects. The value for C_{sys} is fixed at the value of 0.1BX from the study of MTCCII runs. The C_{sys} here may not necessarily the real systematic error, and it is just the number that can descirbe the data best.

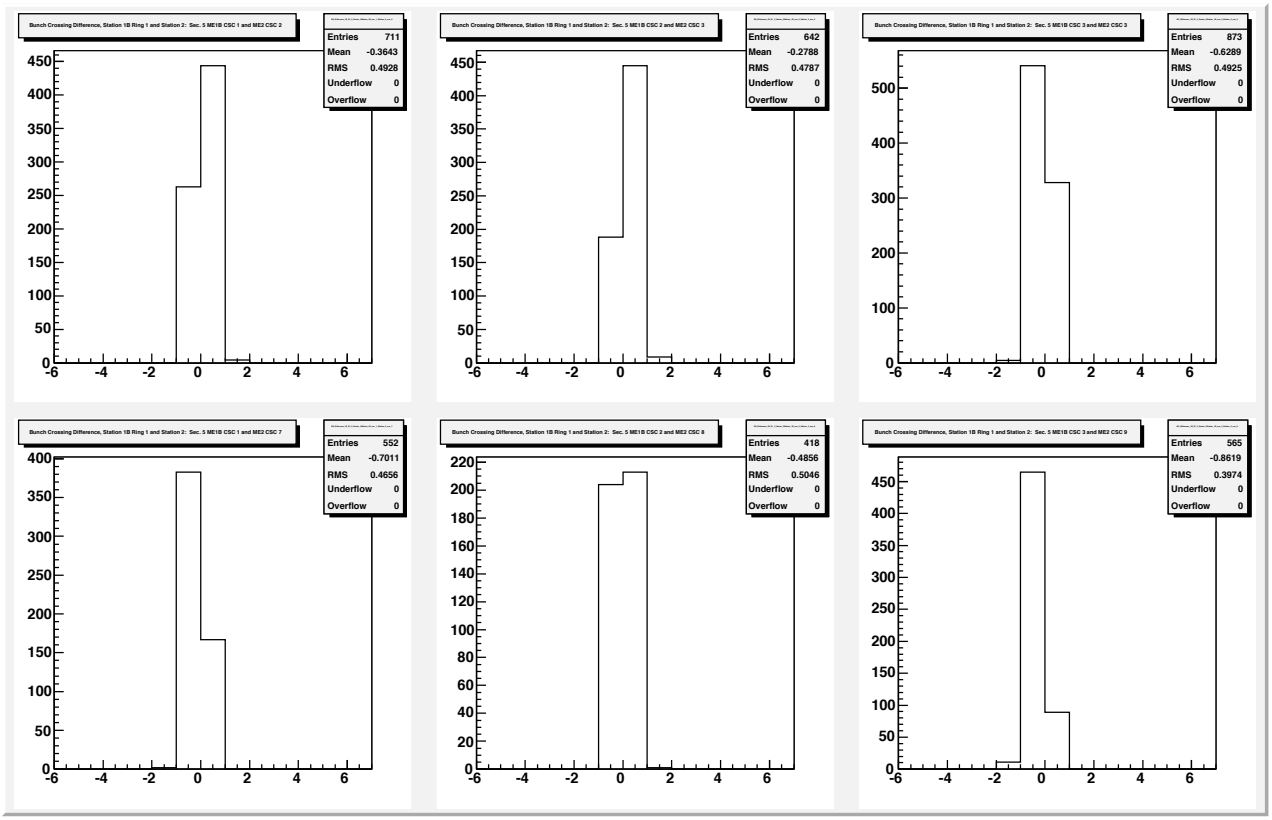

Figure 2: An example of the input combinations: BX differences of tracks penetrating ME1/1 and ME2

Inter-chamber timing program will combine all these errors in the final results. These are the errors for each chamber in individual runs.

For results from multiple runs, some further combination procedures are needed to get the final errors for the combined results. These are discussed in 5.1.

2.2.3 Methods and Procedures

Historically, there have been three methods developed for getting CSC timing constants:

a) naive estimation;

b) minimization using Minuit(there are two options regarding to the staring points);

c) numerical solution of matrix equation;

In all the three options, first the chambers with no data are disabled during configuration, then a scan over all possible combinations is performed to find the best choices of :

1. which chamber should be the reference for this trigger sector;
2. which chambers should be removed;

Then the best choice is used to get the relative timing constants of the chambers for the run with the specific method.

In all cases, it has been observed that the three results from methods b) and c) are in excellent agreements, and they are consistent with method a) in the case when all chambers have data. After the validation, the method c) is taken as default one and method b) serves as cross check purpose. The output of a) method can provide initial values for method b) (optional).

Naive estimation This method origins from simple automation of manual procedures used in MTCC-I mentioned in Sec. 1.3.

We can retrieve the relative BX distribution for segments in one station with respect to L1A. We time-in SP DAQ path so that CSC triggers come in the middle of the readout window. Then look at the overlap between two neighboring chambers on same station and same ring. Look for LCTs in neighboring chambers within appropriate strip region. Then we can compare BX difference of the received LCTs. This should be independent of L1A source as long as it comes from the same muon. Next, the difference in BX for back-to-back chambers in two stations(take ME2 and ME3 as example) are used to get time difference between disks. Look for coincidence in LCT between ME2 and ME3 in same CSC chamber. Then make comparisons ring to ring. They should also be insensitive to L1A source since it is the same muon. We can tie all the information together (still take ME2-ME3 as example): Starting with ring intercalibration, we can define first chamber in each ring as BX=0, and work out the rest. Then an offset is added to one disk and we can calculate ME2-ME3 (with phase) to compare with measured ME2-ME3. Here we assume peripheral crate to peripheral crate timing is already very good. Thus we can recommend changes in timing for ME2 and ME3 based on previous procedures in units of BX.

Here, there are some freedom in choosing the route of comparison to get the timing shift constants for 18 chambers in ME2 and ME3 together. Two different routes (denoted as A and B, refer to AppendixA for details) has been implemented. The comparison of them could be seen from Figure 3. In most cases, Route A can result in smaller error bars, and is chosen as the default for the estimation.

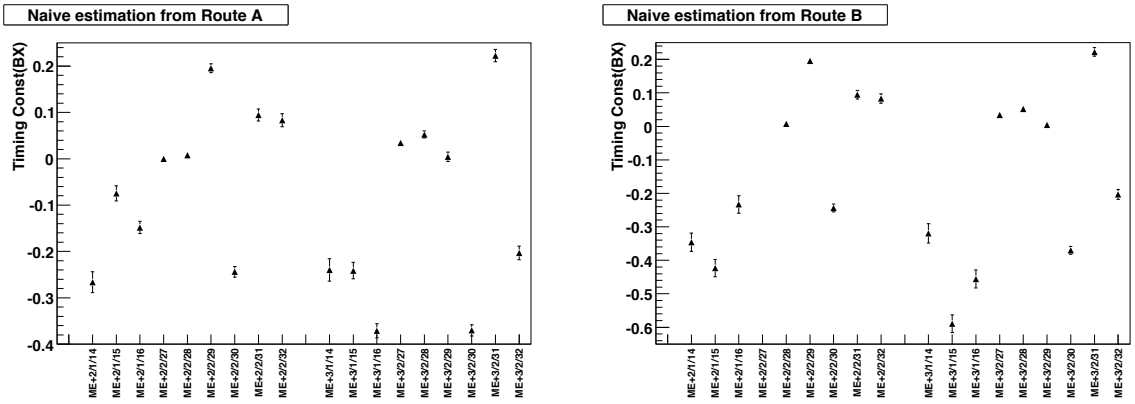

Figure 3: timing shift constants from naive estimation, from Route A(Left) and Route B(Right)

Minimization method A more efficient and accurate way to get the set of constants is the minimization of the χ^2 defined in equation(1).

MINUIT[8] is a widely used physics analysis tool for function minimization, which is also included in ROOT[9], and can be used though Python interface PyROOT[9].

The object function defined as (1) are thus provided. The results from this method are extensively compared with those from matrix method, and they match precisely in most cases. Some examples are shown in Table 1.

The major shortcoming of this method is: it is lack of flexibility. For the runs from MTCC and Slice Test, some chambers have problems from time to time, and thus couldn't provide reliable data. In these cases, this method couldn't function very well and one can't get exact diagnosis message.

Formalism of the matrix method This method can provide this flexibility mentioned above easily and is set as the default method.

The foralism comes from the least square parameter estimation in its matrix form[10]. The χ^2 defined in Eq.1 can be written as:

$$\chi^2(\theta) = (y - F(\theta))^T V^{-1} (y - F(\theta)) \tag{3}$$

Where $y = (y_1, ...y_N)$ is the vector of N measurements at known points x_i. Here θ is the vector of timing constants for each chamber, its dimension is the number of chambers we are studying, denoted as m.

$F(x_i;\theta)$ is a linear function of the parameters, i.e,

$$F(x_i;\theta) = \sum_{j=1}^{m} \theta_j h_j(x_i) \tag{4}$$

Defining $H_{ij} = h_j(x_i)$ for the case of combination algorithm of a whole CSC sector with 109 input measurements. Minimizing χ^2 by setting its derivatives with respect to the θ_i equal to zero gives the estimations of timing constants as:

$$\hat{\theta} = (H^T V^{-1} H)^{-1} H^T V^{-1} y \equiv Dy \tag{5}$$

The error matrix for them $U_{ij} = cov[\hat{\theta}_i, \hat{\theta}_j]$ is given by:

$$U = DVD^T = (H^T V^{-1} H)^{-1} \tag{6}$$

2.3 Inter-sector timing algorithm and global results

The above desciptions are all for studies on the sector-by-sector basis. Inter-sector timing algorithm are with the similar matrix method based on the same principles as described in 2.

For the inter-sector timing algorithm for one endcap, when one sector is chosen as reference (trigger sector 1 serves as reference by default), only 5 constants need to be extracted. To extract these constants, totally 11×6 combinations of neighboring chambers are used. Below the detailed 11 combinations are listed for two neighboring sectors, with one denote as S and another S':

1. S)ME1b/3 - S')ME1a/1
2. S)ME1b/6 - S')ME1a/4
3. S)ME2/3 - S')ME2/1
4. S)ME2/3 - S')ME2/4
5. S)ME2/9 - S')ME2/1
6. S)ME2/9 - S')ME2/4
7. S)ME3/3 - S')ME3/1
8. S)ME3/3 - S')ME3/4
9. S)ME3/9 - S')ME3/1
10. S)ME3/9 - S')ME3/4
11. S)ME4/3 - S')ME4/1

After the inter-sector timing results are obtained, we can get the global timing results for the chambers in the whole endcap with one more step of processing.

For one single run, the sector timing constants (zero for reference sector) are added to the sector-by-sector results. Then all the constants are transformed to the global reference chamber to get the final results, negative values of which yield the delay shifting suggestions.

As a result of group discussion, the chamber ME+1/2/28 and ME-1/2/28 is chosen as the global reference for each endcap.

For results from many runs, the situation gets a little more complex. This will be discussed in 5.1.

2.4 Software, dataset requirements and recipes

In Appendix B, you can find the descriptions of the software actually used to implement the above algorithms. The requirements of data sets is also discussed there.

Appendix E is a recipe and hands-on instruction to get started to use the whole programs step by step.

3 Major Results From ME2 and ME3 Studies from MTCCII data and During Minus End Slice Test

The early developments of the algorithm were based on the analysis of MTCCII data with one sector in ME+1,ME+2 and ME+3. Then it was applied to Minus End Slice Test, with 18 chambers in ME-2 and ME-3. This section describes the results from studies with a system of 18 chambers in ME2 and ME3.

3.1 Validations and Checks of the Algorithm

In order to check the algorithm and understand the results fully, efforts are made in several aspects.

3.1.1 Comparison of timing constants between two rings

Since signals of chambers in the same ring from the same peripheral crate, the fiber lengths and the skrewclear cable lenghs are nominally the same for them. A flat distribution of the quantity $(C_{shift} + C_{ref})_{ME2} - (C_{shift} + C_{ref})_{ME3}$ for the chambers is expected. C_{shift} is the timing constant for the chamber,C_{ref} is the mean of the reference chamber in that ring.

The results shown in Figure 4 are obtained with chambers in ME2/2 and ME3/2, from method of naive estimation. Uniformity from minus end Slice Test has improved compared to that from MTCC run. The residual non-uniformity may reflect the difference at MPC and various other effects such as TOF.

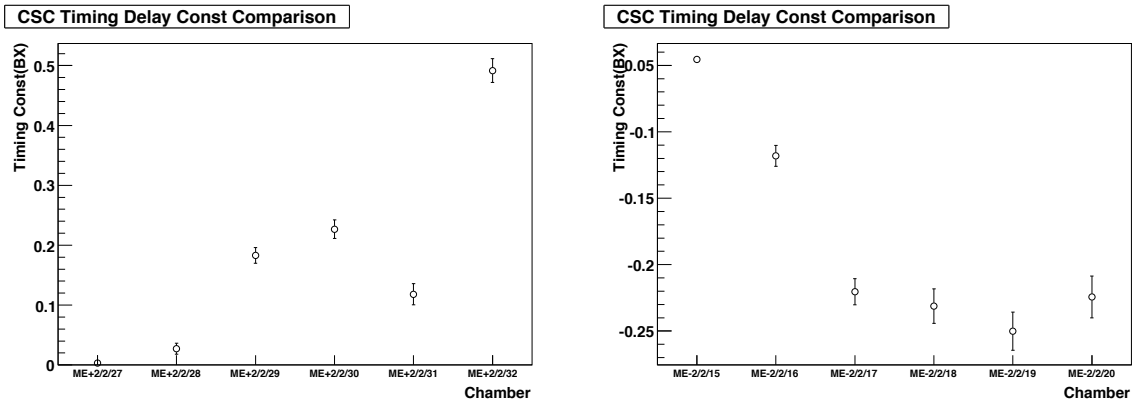

Figure 4: comparison of timing shifts of chambers from MTCC and Slice Test. Results shown are the 6 ring2 chambers in one trigger sector of station ME2, from Run486(Left) in MTCC and Run539(Right) in Slice Test.

3.1.2 The effects of Time of Flight for cosmic rays

Although the time of flight(TOF) for muons to spend between chambers can reach 0.5BX from the first to the last station, we only use the measurements in neighboring chambers in one station or neighboring stations. In this case, TOF is at the level of several nanoseconds. This is roughly comparable to the precision of the synchronization and may not be neglectable. So, the effects of TOF on the algorithm are studied to understand this better.

With ME2-ME3 algorithm, we tried to add TOF as another unkown parameter, thus get it with the timing constants together. The results showed no improvements on the precision of the timing constants. On the contrary, the χ^2 gets worse and the condition number of the matrix gets worse. This shows there is no need for consideration TOF between ME2 and ME3 with cosmic rays. The reason is understood: for both of the MTCC and minus end Slice Test, there are cosmic rays from both directions of ME2 to ME3 and ME3 and ME2, with similar numbers. Thus, the TOF effects for single track are buried in the spread of distributions, with no bias to chamber timing.

CSCs are large enough so that there is also a significant difference in TOF to top vs. bottom for cosmic rays. For ME1 chambers, due to the geometry, TOF may have much bigger influence. For the commissioning of all trigger sectors within the whole endcap, TOF effects between top chambers and bottom chambers are also quite obvious for cosmic rays. These effects are under study. One possible solutions are to use the model from simulation studies, see the discussion in Section 4.4.

3.1.3 Choice of the constraints and the reference chamber

The algorithm can only track the relative timing shift, so there is always a degree of freedom for choosing the overall scale and one constraint is needed to fix it. There are several choices of constraints. Some conditions, such as $\sum C_i \equiv 0$, are simple in the form, but not so intuitive. The most convenient one is to choose one chamber as reference and define the timing constant of it as zero, then the timing shifts of other chambers are fixed relative to this reference.

To choose the best reference, we usually take the χ^2 as a criterium for MINUIT minimization. For matrix method, χ^2 criterium also applies, but it happens often that two or more choices of reference can yield equal χ^2. Condition number[11] is an important quantity to judge ill-behaviored matrix, and provide an indication of the quality of matrix inversion, so we take it as the criterium to further choose the best reference in case of equal χ^2. The procedures are implemented in two Python modules as described in Appx. B.1.2.

3.2 Comparison of Different Methods

Comparisons of results from different methods have been made with many runs from both MTCC and minus end Slice Test. Comparisons are made among results from four options of three methods:

- numerical matrix solution;
- minimization method with zero starting values;
- minimization method with estimation as starting values;
- naive estimation;

Table 1 show the results for Run 57013 in slice test. From the table, we can see results from different methods can agree with each other very well. Due to the lack of one chamber(ME-2/1/9) in this run, the naive estimation method is restricted from yielding reliable results. The naive estimation works well only if all the chambers have data. For that case, we can take a look at the MTCC Run486. Comparison can be made between left plot in Figure 3 and the right plot in Figure 8. They show good agreements between the two methods of naive estimation (RouteA) and numerical matrix solution.

Table 1: comparison of timing shifts obtained with different methods for Run 57013 in the Slice Test

Chamber ID	numerical matrix solution	minimization from zero	minimization from estimate
ME-2/1/8	0.0000 ± 0.0067	0.0000 ± 0.0068	0.0000 ± 0.0069
ME-2/1/10	0.2555 ± 0.0091	0.2556 ± 0.0090	0.2555 ± 0.0090
ME-2/2/15	-0.1023 ± 0.0017	-0.1022 ± 0.0017	-0.1022 ± 0.0017
ME-2/2/16	-0.0059 ± 0.0020	-0.0058 ± 0.0020	-0.0059 ± 0.0019
ME-2/2/17	-0.1031 ± 0.0028	-0.1030 ± 0.0028	-0.1031 ± 0.0028
ME-2/2/18	-0.0179 ± 0.0034	-0.0178 ± 0.0034	-0.0179 ± 0.0033
ME-2/2/19	0.1324 ± 0.0043	0.1324 ± 0.0048	0.1324 ± 0.0041
ME-2/2/20	0.2904 ± 0.0046	0.2905 ± 0.0045	0.2904 ± 0.0045
ME-3/1/8	-0.1171 ± 0.0087	-0.1170 ± 0.0087	-0.1170 ± 0.0087
ME-3/1/9	-0.0879 ± 0.0083	-0.0878 ± 0.0083	-0.0879 ± 0.0083
ME-3/1/10	0.5883 ± 0.0111	0.5883 ± 0.0114	0.5882 ± 0.0107
ME-3/2/15	-0.2046 ± 0.0000	-0.2045 ± 0.0000	-0.2046 ± 0.0000
ME-3/2/16	-0.1524 ± 0.0024	-0.1524 ± 0.0024	-0.1524 ± 0.0024
ME-3/2/17	-0.0962 ± 0.0026	-0.0962 ± 0.0026	-0.0962 ± 0.0026
ME-3/2/18	0.0785 ± 0.0037	0.0785 ± 0.0036	0.0785 ± 0.0036
ME-3/2/19	0.1980 ± 0.0041	0.1981 ± 0.0040	0.1980 ± 0.0039
ME-3/2/20	0.2353 ± 0.0047	0.2354 ± 0.0047	0.2353 ± 0.0046

3.3 Calibration Test

In order to apply the timing shifts according to the measurements at Sector Processor, we need a strategy to achieve this more effectively. Before using it as a routine tool, we have made some "calibration test" with two more runs. The procedure is as following:

1. Set all the timing settings to the default values. Then take a run with ”all singles” trigger, analyze it as a reference.
2. Take runs with known different delay setting, get to see whether this algorithm can track what has been changed. We can purposely add some known delays to a certain chamber.
3. If the algorithm is validated by this test, we can get corresponding correction to the delays from the analysis results.
4. Use the feedback to adjust the delay settings and take another run to look whether synchronization is improved. If not good enough, make further correction based on analysis results of this run.(this can be repeated until we are satisfied)
5. This analysis can serve as a routine tool afterwards for monitoring purpose.

During the minus end Slice Test, we took the following two special runs to carry out this task:

- Run 608 = run with trigger primitive timing to the SP as it has been during the entire minus-side slice.
- Run 609 = run with changes to two chambers only, assuming 2.2ns per count for the “AFEB fine delay” parameter:ME-3/1/10 subtract 8 counts (0.7bx) from “AFEB fine delay”; ME-3/2/15 add 2 counts (0.2bx) to “AFEB fine delay”.

Figure 5 shows the variation of the timing shift of the two chambers before and after the changes of settings.

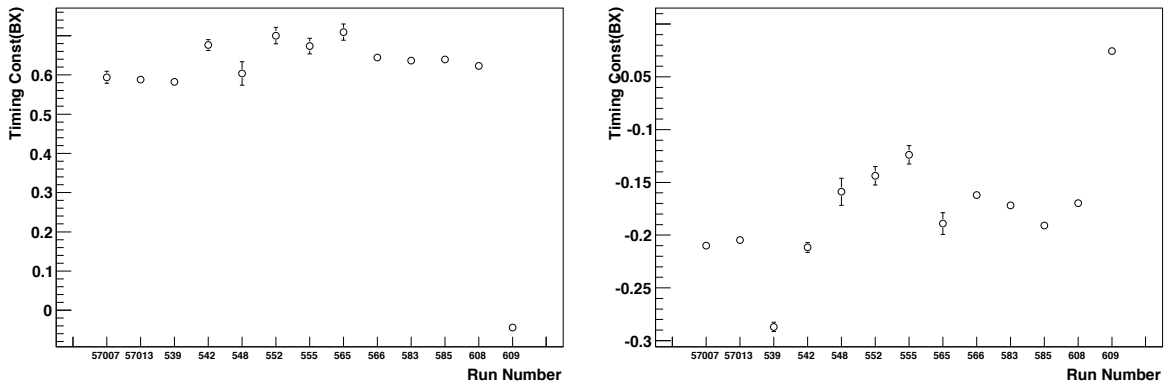

Figure 5: Calibration test of the algorithm with special designed minus end Slice Test runs. The plots show the variations of timing constants among different runs for ME-3/1/10 (left) and ME-3/1/15 (right). The delay settings were changed on purpose between Run 608 and Run 609.

From the figures, it is clearly shown that the 0.7bx and 0.2bx shift between Run 609 and previous runs is exactly what we change. Thus, this calibration test that the algorithm can track well what have been changed, and the results are reliable for the purpose of monitoring and prediction.

3.4 Monitor and Adjustment

After the calibration test with run609, run608 was further studied. A set of timing constants are obtained for each chamber. The suggested adjustments from this analysis were made with AFEB fine delays in a new run: run618.

The results before and after applying timing adjustments suggested can be seen in Figure 6, plotted versus the chambers in the minus side Slice Test. The timing alignment between chambers thus reach the level of better than 0.1bx, which is the resolution of the AFEB fine delay step size.

3.5 Timing Constant Stability

In order to monitor the synchronization status, the algorithm can track the timing constant stability for the same chamber along different runs.

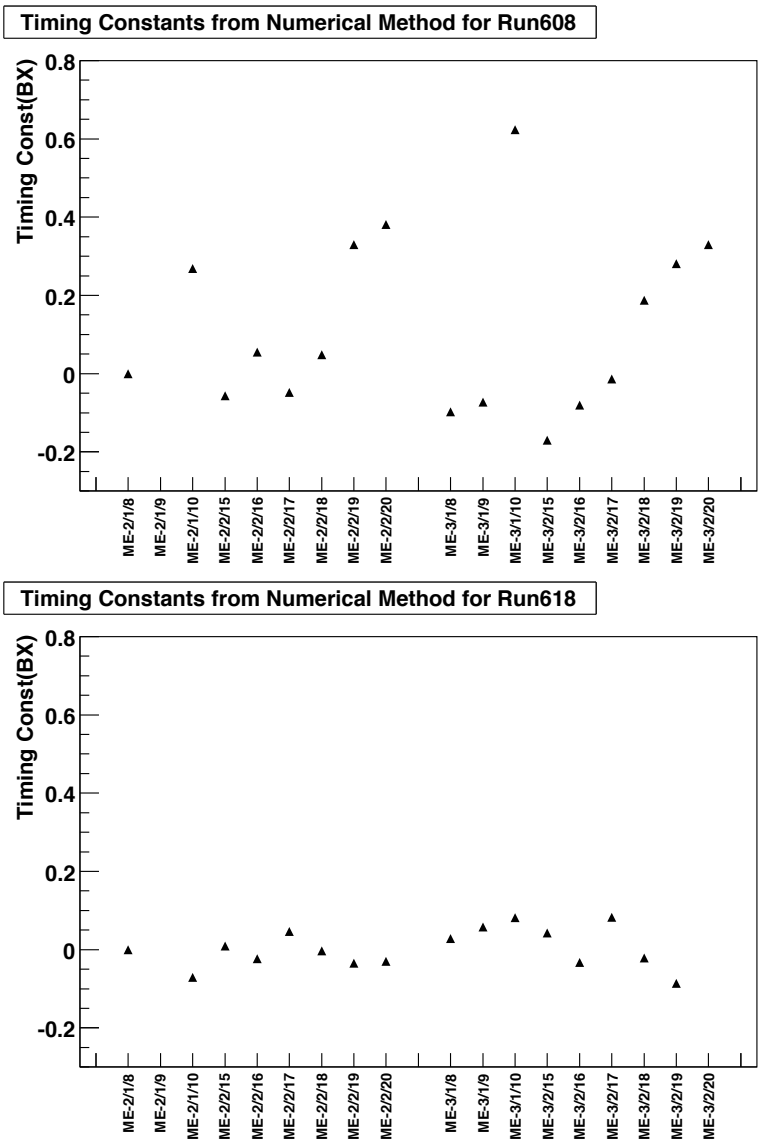

Figure 6: timing constants before(Top,Run608) and after(Bottom,Run618) adjustment

The reference chamber is determined separately for every run according to criteria of minimum χ^2 or minimum condition number in the case of equal χ^2. The reference is transformed to a common reference chamber, while keeping their previous error messages. In the following results, a common reference is taken as chamber ME2/1 in TF numbering(corresponding to ME-2/1/8).

Figure 7 shows stablilities of the timing shift among consecutive seven runs in minus end Slice Test. There are no major changes of timing settings, so the constants are quite stable in this period.

4 Expansion of the Algorithm to a Full Sector

The final goal of this study is to develp an algorithm for the synchronization of the whole system of 468 CSC chambers. So the algorithm with inputs from ME2 and ME3 must be expand to the scale of a full CSC sector first.

Since minus end Slice Test only contains ME2 and ME3 chambers, MTCC data are reprocessed and studied with the expanded algorithm to prepare for the future use. Here, we use Run 486 of MTCC phase II for this purpose.

The resulted algorithm has been described in Section2, and these algorithm in application is described in later Sections. In this section, the focuses are the various tests and verifications with MTCC data during the developments. The limitations and possible future refinements will also be discussed.

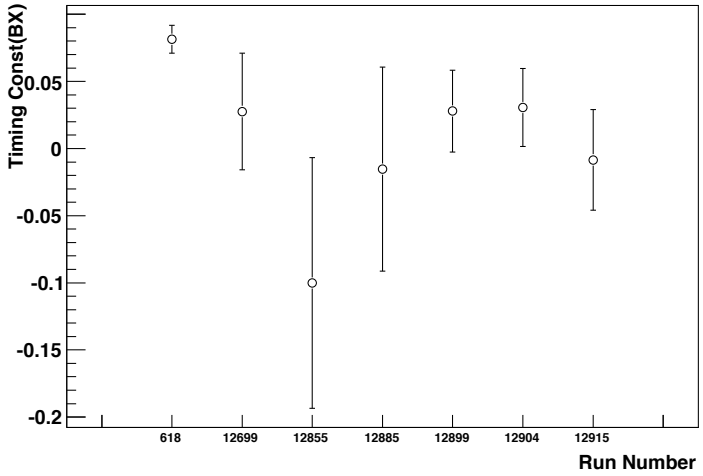

Figure 7: Stabilities of the timing shift among several runs in minus end Slice Test. The timing shifts were from the same chamber of ME-3/1/10.

4.1 ME1 Standalone Algorithm

In order to retrieve timing info from the underground runs, in which it is probable to have ME1 chambers only configuration first, we developed ME1 standalone algorithm. Also, due to the special geometry layout, ME1 standalone studies could help to study the much bigger influence of Time Of Flight.

The basic principle is the same with ME2-ME3 studies. For ME1 standalone algorithm, the number of combinations used is 42, and they are already described in Appendix D.

One restricting factor is the statistic limitation. During MTCC, the participating chambers were from Sector 5, which is nearly at the bottom in position. In this case, cosmic rays come downward and this is a preferred case for collecting events penetrating both ME1/1 and ME1/2 or ME1/3. But the number of events we can get is still very poor due to the steep angles between the rings. Only 7 cosmic events were collected penetrating ME1/1A and ME1/3 out of 1M events of MTCC Run486. For the horizontal sectors, it will be even more difficult to get this kind of combinations.

Figure 8 shows the results from ME1 standalone algorithm for MTCC Run486. Results from standalone ME2-ME3 algorithm are also shown here. The constants for ME1/3 are very poorly determined due to the extremely low statistics.

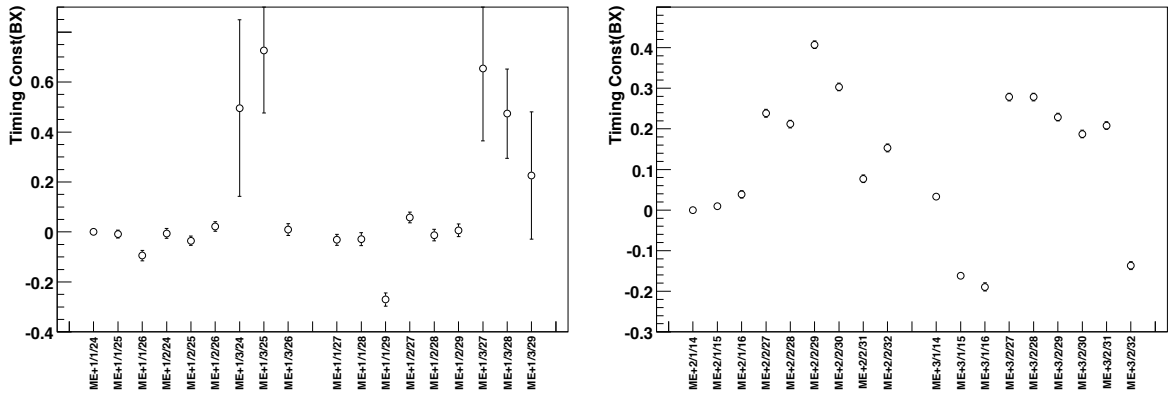

Figure 8: Timing shift constants separately from ME1 standalone algorithm(left) and ME2-ME3 algorithm(right) for MTCC Run486.

Considering the reduction of cosmic flux underground, this ME1 standalone algorithm thus can not be reliably applied for future tests with cosmic rays. Even for muons from beams, due to the steep angle of some combinations, it is also very hard to apply this algorithm alone.

4.2 Combined Algorithm: the Whole CSC Sector as a Whole

As a natural next step, the timing algorithms are extended to reach its goal of including all the chambers in one sector. Since we only had real data of ME1, ME2 and ME3 together in MTCC, the present algorithms contain just these three stations. It should be straightforward to extend to ME4.

In this case, the number of chambers involved is 36 instead of the previous 18. The BX differences used include 42 and 43 combinations in the standalone case of ME1 and ME2-ME3, respectively. In addition, 24 combinations between chambers of ME1 and ME2 are included.

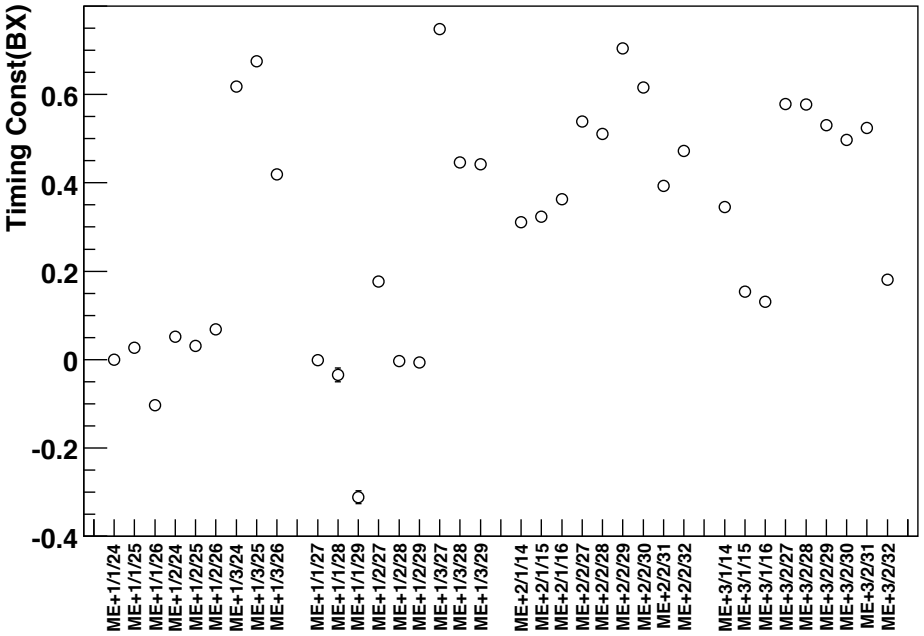

Figure 9: Timing shift constants from combined algorithm for a whole sector. Data used was from Run 486 of MTCCII, systematic errors are not considered here.

Figure 9 shows the timing shifts thus obtained from combined algorithm for a whole sector for MTCC Run486, with very small error bars.

Figure 10 makes a direct comparison of results from the combined algorithm and those from standalone ME1 and ME2-ME3 algorithm. Seen from this figure, the results agree better for ME2-ME3 than ME1. The reason is identified as the statistical limitation mentioned in Sec. 4.1. Therefore, the combined algorithm is more robust compared to standalone ones.

4.3 Signal propagation correction

The inputs to the inter-chamber timing algorithms include many measurements of neighboring chambers in the same ring of the same station. In this case, due to the different configurations of overlapping regions and the positions of AFEBs for different chambers, there are some bias in the ΔBX measurement.This situation is illustrated in Figure 11.

From Figure 11, it is easily seen that the overlapping region is always much nearer to AFEBs of one chamber and thus there is a difference of signal propagation from the wire hit position to AFEB. We obtained estimations of difference for different types of chambers based on the parameters for different chambers. Details are described in Appendix C. Using these values, the inputs to the inter-chamber timing are corrected before any further processing.

Comparison with previous results without correction demonstrate the effects due to this propagation difference in the final results is within ± 0.2 BX.

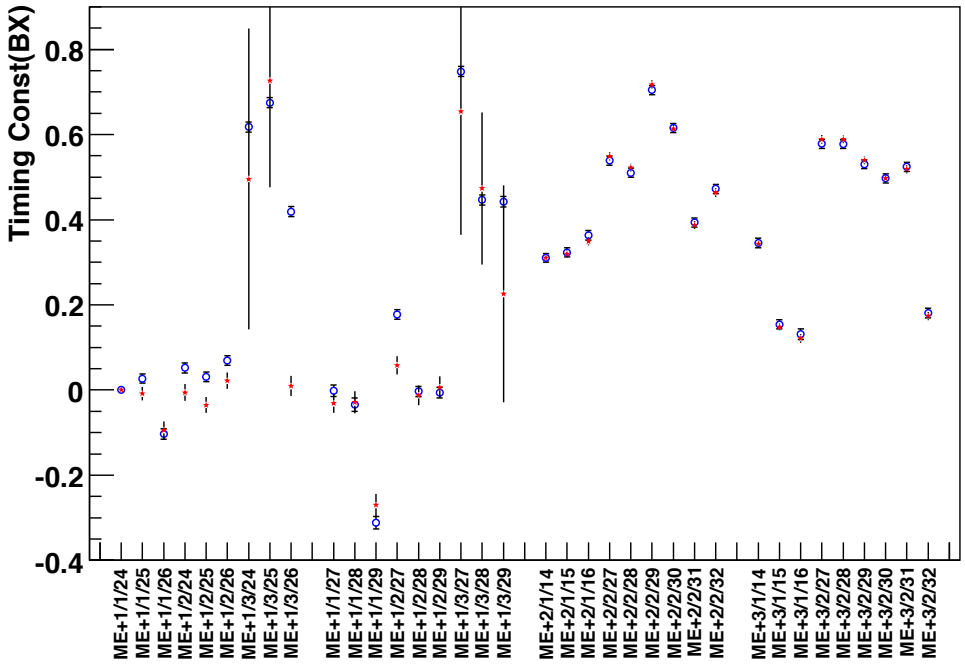

Figure 10: Comparison of results from separate algorithm(Red filled stars with plain error bars) and combined algorithm(Blue empty circle with perpendicular-edged error bars). Take Chamber ME+1/1/24 as reference or both cases, and proper shift has been put for ME2 and ME3 chambers compared to Figure 8.

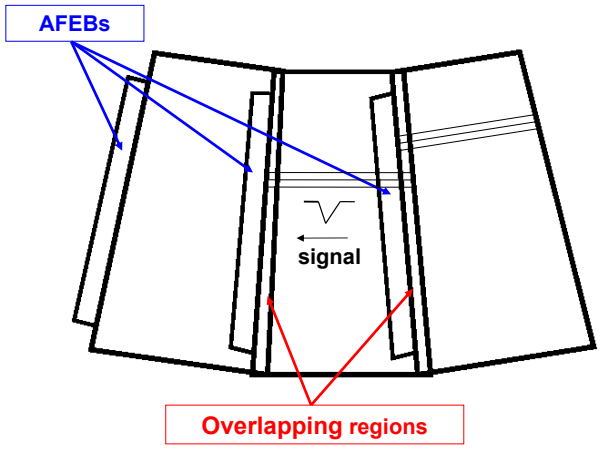

Figure 11: Illustration of signal propagation correction. The figure shows the postion of AFEBs with respect to the chamber and the different time for wire signals in overlapping regions to propagate to the neighboring chambers. In order to get unbiased measurement of ΔBX between neighboring chambers, the correction must be performed.

4.4 Limitations of Present Algorithm

Here is one limitation of present algorithm: it is very common in the commissioning that some chambers are absent in the run due to various failures. This algorithm has already tried the best to tolerate missing data of certain number of chambers due to such failures. When the number of absent chambers exceed some extent, the algorithm couln't work. This threshold is not fixed because different chambers have different "weight" in the algorithm. For example, the chambers in ME-2/1 ring is related to more combinations used, so the influence of absence of them is bigger, while chambers in ME-3/1 may have smaller influences.

Run11961 in the minus end Slice Test demonstrates this limitation clearly:

In this run, chambers corresponing to Peripheral Crates ME-2/1/8, 9, 10 were out due to no high voltage,; ME-3/2/20, ME-2/2/20 not in run due to removed TMB/DMB pair so that Maraton will run. There are totally 5

chambers out of this run, not so bad at first sight, if only judging from the number. But just due to this special configuration of absence chambers, the analysis of this run failed.

Actually for the analysis, it is a very bad case that the total ring of ME-2/1 out. The logic here is: when ME-2/1 is out, ME-3/1 is actually excluded from the analysis, because ME-2/1 ring is the only ring that have practical combinations with them(ME-3/1 vs ME2/2 is "wrong direction"). Thus only 10 Chambers left, located in the two "imaging" rings. The combinations here is quite limited now, and thus the algorithm couldn't work. So what really count is not purely the number of Chambers out of data. If it is ME3/1 that are the ring out, or there are 5 chambers out while ME2/1 still has one chamber left, the algorithm can still work. The case of Run 11961 is the worst.

The solution of this limitation is to employ more measurements in this algorithm to make it more robust. From the comparison in Sec. 4.2 ,we already see the improvements in precision and robustness when more measurements are included. More combinations have been studied and added to the algorithm since then.

4.5 Possible Further Refinements

Compared to muons from beams, there is one additional ambiguity for cosmic rays: for the same preferred hit chamber patterns of CSC trigger, there are two possible directions of "heading to IP" and "leaving IP" and thus two different timing patterns. This effect is also different from sector to sector: mainly "heading to IP" for those at the top, and mainly "leaving IP" for the bottom ones, while for the horizontal trigger sectors, both exist.

One possible solution is to make use of the relations of Switched Capacitor Array buffer(SCA) vs time, this can give a more precise information on timing. The spread is about 6ns.

Another solution is to divide the region of study more finely, considering the difference of tracks through different wiregroups. By refining the offline selection to first reconstruct tracks, and constraining them to come from the IP, we can derive time constants for different regions of a chamber. One can use Δ(WG), or Δ(eta). Plot the distribution of $\Delta\eta$ to see whether there is something like "two peak" structure. Each ALCT region is 8 wiregroups long, we may separate delay chip for each region.

Beam halo muons could be used to check the synchronization for chambers in the same ring.

During collision, it would be good to make use of dimuon events that hit non-adjacent regions to check time synchronization, i.e. compare one side of the disk with another with muons that come from the same crossing, or one endcap with another.

5 Application of the Program to Full Endcap and Results from Local/Global Runs underground,CRUZETs and CRAFT

In spring 2008, the whole plus endcap of CSC chambers were commissioned fully underground. Then regular local "golden" runs and several CMS wide monthly global runs, several phases of Cosmic RUn at ZEro Tesla(CRUZET), etc) followed. These provided us an opportunity to apply all the procedures previously developed to timing in all of chambers in the whole endcap. Inter-sector algorithm described in sec 2.3 and the combination method described below in sec 5.1 were also applied to the scale of the whole plus endcap of CSC chambers. A special publication page([12]) was established to communicate the full results from the inter-chamber timing results seen from CSC Track Finder data.

5.1 Combination of results from different runs

For the underground local/global runs, we can get many results from different runs. There are some differences among them but most are consistent with each other. To get the best use of all this information from all the runs, and to make the timing suggestions more reliable, the following procedures are adopted to process the results from different runs:

1. Combine the sector-by-sector timing results: with reciprocal of uncertainty squared as the weight, weighted mean is obtained using sector-based results from all the runs chamber by chamber. Then a 5σ cut is applied to remove all the outliers from the weighted mean. This process is iterated until all the results used agree with each other within 5σ.

2. Combine the inter-sector timing results: with the similar method as the sector-by-sector procedures above.

3. Get the global constant: combine the sector-by-sector and inter-sector results thus obtained. Then change the reference to the chamber of ME+1/2/28. The results are the global constants with errors.

4. Give out the suggestions: the timing shifts to apply have the opposite sign as to the timing constants. Then make further checks and publish suggestions from these runs.

5.2 Results from CRUZETs

There were four stages of CRUZETs at summer 2008. During these period, the established procedures were first excercised to synchronize the two CSC endcaps separately.

There were several iterations of the timing setting changes. The combination methods described in sec 5.1 were used to get the final global timing results for the whole endcap. After the suggested shifts were applied to AFEB fine delay, most of 234 chambers in plus endcap were timed in 0.6BX with each other, as shown by Figure 12[1)]. During CRUZET II, this was also confirmed from other sources, like observations from HCAL data and analysis of DT-CSC correlations.

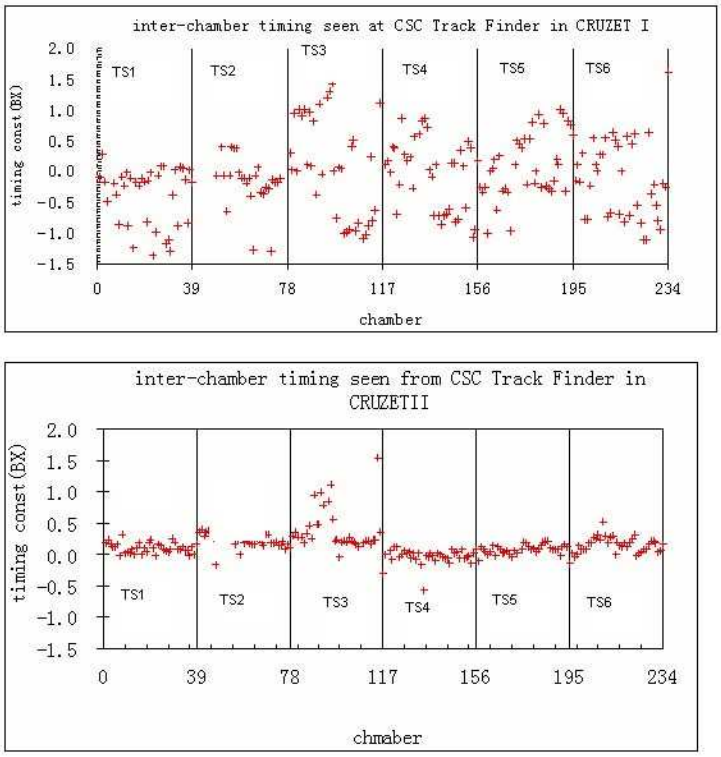

Figure 12: Inter-chamber timing results for plus endcap from CRUZET phase I (top plot) and II (bottom plot). The suggested timing shifts were applied during the period in between. The conventions for chamber numbering (for both endcaps) used in this figure are as following: The chambers are grouped in the order of trigger sectors as indicated in the figure; Inside each trigger sector, the chambers are firstly grouped with the unit of stations and are then ordered by the rings; Thus the numbering goes from 1 to 234 for each endcap. To be more explicit, it goes as: TS1 subdetector1 ME1/1 chamber 1, ... 3, ME1/2, ... ME1/3, subdetector2 ME1/1, ... ME1/3, ME2, ... ME4, TS2, ... TS6 ME4/1 chamber 3.

The same procedures have also been applied to minus endcap chambers shortly before CRUZETIII. The first results of synchronization of minus endcap chambers are shown in Figure 13.

5.3 Results from CRAFT2008

During CRAFT, two important changes were made regarding to CSC inter-chamber synchronization:

[1)] For results after this section, the same conventions for chamber numbering as explained in Figure 12 are used, unless explicitly specified otherwise

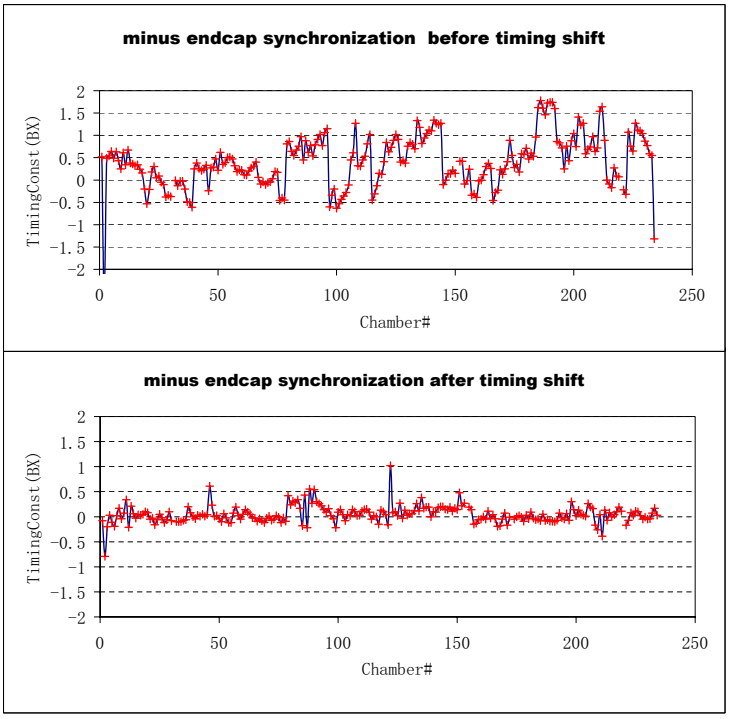

Figure 13: Inter-chamber timing results for minus endcap before timing shift (top plot) and after (bottom plot). The suggested timing shifts were applied during the period in between, shortly before CRUZETIII.

- TTCrxFineDelays were implemented on the basis of TTCrxCoarseDelays, which were implemented during CRUZET August 2008, after the more precise measurement of TTC fiber lengths.
- To be more consistent with other subsystems, extra delays were introduced: +2BX in the top sectors and +1BX in the bottom sectors. So chambers in top sectors are delayed additionally by 1BX relative to those in bottom sectors.

The additional 1BX delay of top w.r.t. bottom was incorporated into the analysis. Then, results of the relative timing analysis from earlier runs were implemented online, into AFEB fine delays. The results are shown in Figure 14.

Figure 15 shows the distribution of the timing constants for the two CSC endcaps. They indicate the precision achieved after this exercise to be around 0.15BX, which is roughly the same order of the error estimation from the method itself.

6 Inter-Chamber Synchronization Results from the first LHC beam commissioning Sep. 2008

LHC had its first beam during Sep. 8 and Sep. 19, 2008. CSCTF has special trigger for halo muons. We didn't implement special delay settings for the single beam, but this set of data provide the first opportunity to exercise the inter-chamber synchronization with beam.

We also tried the trigger synchronization of the CSCs using LHC muons with absolute BXN analysis. The method relies on the synchronous nature of beams and makes maximum use of every beam halo muon. It will be described in Sec. 6.2. The comparison of the results from the two methods will be described in Sec. 6.3.

To avoid events triggered by other subdetectors, the events were selected using Global Trigger information so that only triggers from CSC halos were used. From CSC halo trigger performance in previous cosmic running, we know there were 2Hz cosmic background in the CSC halo triggers. To reduce the possible contamination from cosmic rays, and a BXN selection was performed to only select the events from the BXs matching beam structure at that time. This way, we obtained quite pure CSC halo muons for the studies.

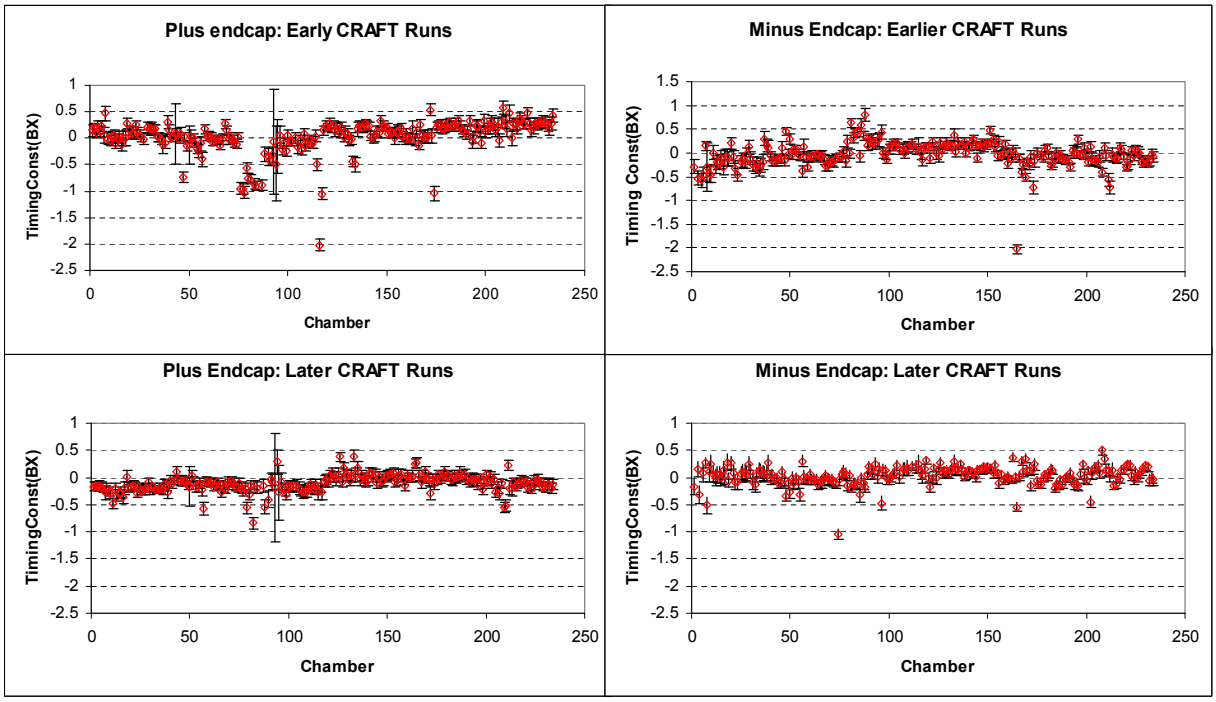

Figure 14: Timing Constants from CRAFT. Top plots show the status of inter-chamber synchronization before setting AFEB fine delays. The results of this analysis were then implemented online, into AFEB fine delays, and bottom plots show results after that.

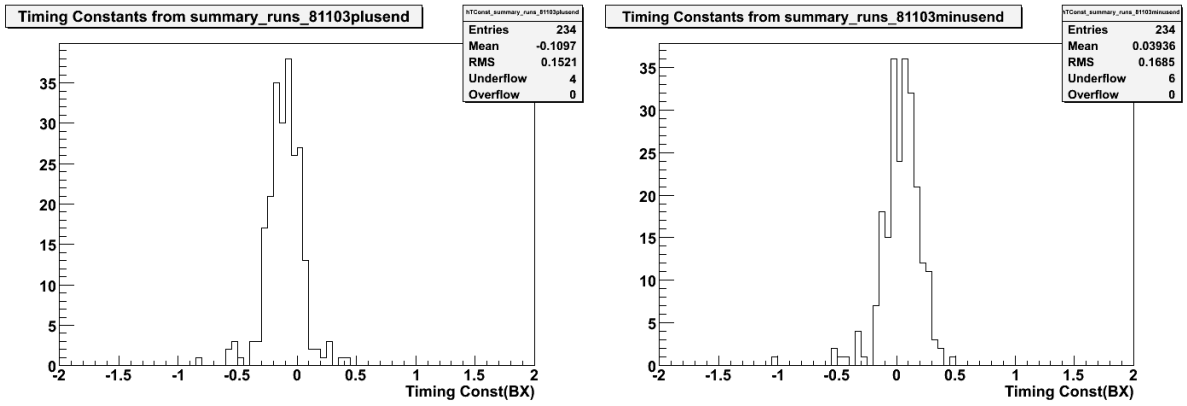

Figure 15: Distributions of timing constants from CRAFT. The precision achieved after this exercise to be around 0.15BX.

6.1 Results from relative timing method

Several runs with the most statistics were used for the analysis. Run62232 and Run62384 taken on Sep. 12 were used for the first trial. They have beam 2 captured during this run and the BXN jumped for beam crossing. The full procedure of the relative timing methid described in Sec. 2 was applied.

The left plots in Figure 16 show some typical results from two trigger sectors.

Halo muons have very different timing behavior compared with cosmic rays. They fly mostly in parallel to the beam, from one endcap to another. Since the system was configured for cosmic ray muon timing, so the bigger spread of the timing constants is as expected. The full results can be found in the results publication page([12]).

6.2 New method for synchronous muon sources: absolute BX number analysis

This is a new way to do the trigger synchronization of the CSCs using LHC muons. Since the halo (or collision) particles are periodic, coming at a specific point in time (specific BX), the absolute BXN of the recorded hits will provide important timing information.

This method makes a BXN histogram for every chamber and fill it every time that chamber had an LCT. The quantity filled is the BXN recorded by the SP (it's the BXN of the L1A arrival) corrected by the relative BX in the SP readout. It is computed as: BXN_LCT = ME_BXN + TIME_BIN_LCT - 3 This assumes that the LCTs are normally timed-in to timebin=3 (counting from zero) in our readout of 7 time bins. In fact, subtracting 3 is irrelevant, and we could drop it.

For cosmics these histograms will be flat. For beam, there will be a spike in the BXN-LCT at the BXN corresponding to the filled bunch. If we fit these spikes, and compare to one reference chamber, this gives us the offset from the reference chamber. So if the reference chamber has a spike at BXN=a, and another chamber has a fitted peak of a-0.5, we will know that chamber is early by 0.5 BX.

This method of absolute BXN synchronization makes maximum use of every muon associated to the beam. It uses the advantages of synchronous particles over cosmics, and complements well the approach of relative timing analysis.

Some typical results from the absolute BXN analysis of single beam data are shown in right plots of Figure 16.

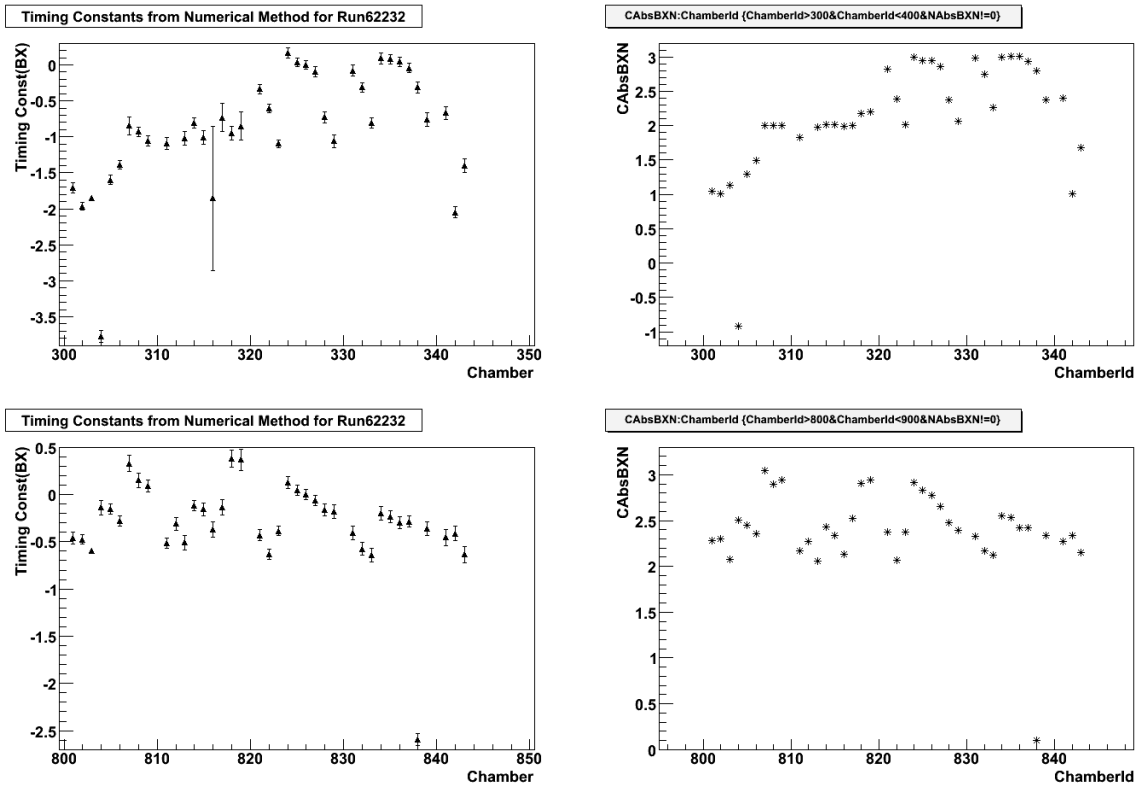

Figure 16: Timing constant results of two methods mentioned in Section 6, single beam data, Run 62232. The left plots are from relative timing methods; the right plots are from the absolute BX number analysis of data of the same sector, the absolute reference of this analysis is arbitrarily chosen. The numbering for chambers is: Sector_Id*100+Station_Id*10+Chamber_Id

6.3 Comparison of results from the two methods

Comparing these results with the results of the same trigger sectors from relative timing analysis in left plots of Figure 16, you can see they agree with each other quite well. The tiny differences between them are well below the size of the errors.

This comparison is also a verification of the relative timing analysis method we were using, from an independent source. The relative timing method can be applied to both comsic and beam halo data, and it should be able to handle the future collision data as well.

The Absolute BXN analysis provids redundancy for cross check and another independent means to get the inter-chamber synchronization information. It can also serve as a for synchronization of colliding beams, as discussed

in Sec. 7.1.2.

7 Towards Collision Synchronization

Most of previous sections are devoted to discussions of CSC inter-chamber synchronization for cosmic rays. But our final goal is the synchronization for LHC colliding beams.

Synchronization with beam is quite different from the cosmic one, and in fact a easier case. With the beam, nearly all the muon tracks origin from the Interaction Point (IP). Different sectors are symmetrically located with respect to beam and thus no need to consider the differences of different sectors as we must do for inter-sector for cosmic ray case. All available programs for cosmic rays and single beams should work directly for collision beams.

7.1 Three proposed methods towards collision synchronization

There are three different approaches to achieve the goal of CSC synchronization with collision data. Below they are described one by one.

7.1.1 Method 1: transformation to collision synchronization based on simulation studies.

Cosmic rays, halos from single beams and muons from collision are quite different in their timing behaviors, and it is not apparent regarding to scaling time of flight effect from one to another.

We can make a first round of corrections to time-in CSCs for collisions before first collisions, to be refined after the run. Basically, we can use the MC simulation of cosmics and the MC simulation of collisions to provide the guide for transformation of the current cosmic synchronization to the one for collisions.

Once the simulation of cosmic rays, beam halos and colliding beams is verified to be able to reproduce major features properly as real data, we can use it as a guide to understand what transformation is needed to go from cosmic and single beam synchronization to collision synchronization.

The basic procedures for translation from cosmic to collision synchronization:

- Simulate cosmic and colliding beams with the same detector and trigger settings separately and get the timing constants of each chamber with the relative timing method described in Sec. 2
- Get the difference of the timing constants from the two sources for each chamber. This will be the transformation constant for this chamber.
- Get reliable inter-chamber timing constants from real data for cosmic rays.
- Apply the transformation constant of each chamber to the timing constant obtained with cosmic rays, then you will get rough estimation of what it be like for collision data.
- Derive the proper delay settings to compensate timing constants thus obtained to achieve the first order synchronization for colliding beams.
- After accumulation of enough collision data, finer adjustment can be achieved by the relative timing or absolute BXN analysis.

The 2008 single beam data provided an unique opportunity to verify the method by comparing the results from real halo data with what was predicted from cosmic-halo translation with similar method as mentioned above. The results from comparison of MC and 2008 beam halo data is shown in Figure 17. It is just a qualitative comparison due to the discrepancies with some of the real run conditions in the simulation. But we can see that MC behavior is quite similar to what we really observed for cosmic/halo comparison. So MC should be rather reliable to use for this transformation.

The same procedures can be also applied to transformation of halo muons to collision. It will provide additional means to cross check the predictions from cosmic results. If the predicted collision synchronization from cosmics and beam halos agree well, the probability of having well timed-in system for collisions is higher.

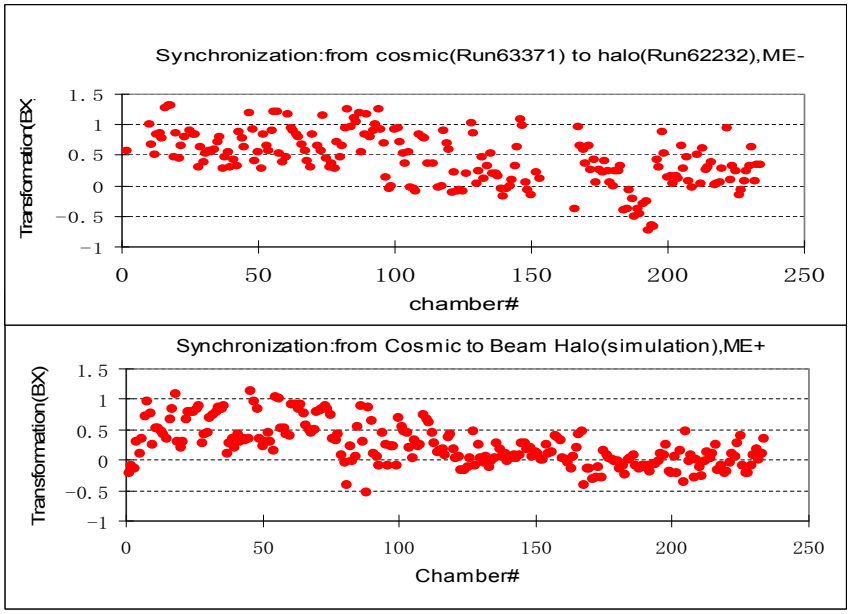

Figure 17: Verification of synchronization transformed from cosmic to beam halo. Top plot shows the synchronization transformation constant for each chamber in ME-; run63371 of cosmic rays and run62232 of single beam with beam2 profile were used. Bottom plot shows the results from simulation of cosmics and single beams of beam1 profile. Since the circulating directions of beam1 and beam2 are opposite, the reults of ME+ should match what are from ME- in real data above.

7.1.2 Method 2: absolute BXN analysis

The method itself has already been described in Sec. 6.2.

It has the following features :

- Low requirements on the events accumulated, so can provide very quick feedback.As an example, the results shown in right plots of Figure 16 mainly came out of about 12 minutes' beam capture and the previous injection attempts.
- It works for both single beam and colliding beams.
- This means we don't need rough adjustments beforehand. Just take a (short) run with real beam, then analyse it, then put the corrections back and we should get rather good synchronization.

So this method can be excercised in conjunction with Method1 detailed in 7.1.1.

7.1.3 Method 3: emulating the collision profile

Since there are such complexities of cosmic synchronization, it would be favorable if we can emulate the collision profile from runs with cosmic rays. Then the information we obtained from these special cosmic samples shouldn't be quite different from that for collsion beams. We can use it for initial stage of colliding beam commissioning and improve it later.

There are two possible options:

- Design and take cosmic data some special trigger: the possible bottom sectors only trigger is already discussion in CSC and trigger community, maybe we can try it out in some later MWGR or CRUZET in 2009.
- Relative timing analysis of the subsample of cosmic ray data that resembles collision muons. The sample can be obtained by offline selection that constraints the direction of the cosmic tracks from reconstructed information. The efforts of CRAFT08 data is ongoing, and we can also repeat such studies with future CRAFT09.

7.2 In practice: use the three methods together

These three methods above are complementary to each other, so their results can be compared and combined to provide us more robust estimation. We can apply all the three methods in most of the time.

For data from synchronous source (halo and collision), we can use the methods simultaneously in most cases. But there are some cases that we have to choose only use one or two of them due to some constraints. For example,

- When data sample is small, it is better to use absolute analysis only.
- If there are more complex BXN patterns due to beam conditions, we have to rely on relative analysis.

8 Summary and Outlook

The procedures and methods described here were developed during the phase II of MTCC, validated with runs from MTCC and the subsequent minus end Slice Test and fully used during local/global runs underground. It can extract time differences for all relevant chamber combinations in one run, derive a set of global corrections for the timing parameters for all 234 chambers in one endcap simultaneously. This implies making timing changes to each chamber just once. Before using it as a routine tool, some ”calibration tests” with new runs have been passed. Some suggestions have been made for chamber tunings and were proved to be effective during minus end slice tests and underground runs. The stability of results over time is confirmed through many runs, and the major variations are understood.

The synchronization of a whole CSC endcap can be done together with inter-sector timing results. The Global Runs, CRUZETs, beam comissioning and CRAFT all provided good opportunities to run the procedures and workflows and CSC inter-chamber synchronization has reached the level of less than 0.2BX. We can use different methods to get reliable inter-chamber synchronization for LHC collisions.

References

[1] D. Acosta, M. Stoumimore, and S.M. Wang. Simulated Performance of the CSC Track-Finder. CMS Note 2001/033, CMS Collaboration, 2001.

[2] CMS Collaboration. The Muon Project Technical Design Report. CERN/LHCC, CMS Collaboration, 1997. CMS TDR 3.

[3] D. Acosta, D. Holmes, K. Kotov, L. Uvarov, D. Wang, et al. The Cathode Strip Track Finder at the 2006 Magnet Test and Cosmic Challenge. CMS Internal Note CMS IN-2007/57, CMS Collaboration, 2007.

[4] CSC Slice Test Wiki page. https://twiki.cern.ch/twiki/bin/view/CMS/CSCSlice, 2007.

[5] G. Rakness, J. Hauser, and D. Wang. Synchronization of the CMS Cathode Strip Chambers. In *Proceeding of TWEPP 2007*, 2007.

[6] D. Acosta, D.Holmes, D.Wang, et al. Commissioning of the CSC Track Finder. CMS Internal Note CMS IN-2008/053, CMS Collaboration, 2008.

[7] CSC Track Finder DQM project. http://cms-csc.web.cern.ch/cms-csc/DQM/TrackFinder/manual.html.

[8] F. James. *MINUIT: Function Minimization and Error Analysis, Reference Manual*. CERN, cern program library long writeup d506 edition, 1998.

[9] ROOT: An Object-Oriented Data Analysis Framework. http://root.cern.ch/.

[10] S. Eidelman, K.G. Hayes, K.A. Olive, et al. Review of Particle Physics. *Physics Letters B*, 592:1+, 2004.

[11] Condition number: From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Condition_number.

[12] Dayong Wang. Publication Page for Inter-Chamber Timing Results from CSC Track Finder Data. http://cms-csctf-sw.web.cern.ch/cms-csctf-sw/timingresult/WelcomePage.html, 2008.

[13] CMSSW: CMS Offline Software. http://cms.cern.ch/iCMS/jsp/page.jsp?mode=cms&action=url&urlkey=CMS_OFFLINE.

[14] NumPy: Numerical Python library. http://numpy.scipy.org/.

[15] GSL: GNU Scientific Library. http://www.gnu.org/software/gsl/.

[16] CSC Timing Studies with Track Finder Twiki Page. https://twiki.cern.ch/twiki/bin/view/CMS/CSCTimingSeenAtTrackFinder.

[17] CMS CSC Track Finder: Software page. http://wangdy.web.cern.ch/wangdy/csctf_sw.html.

[18] CVS Repositry CSC Track Finder Timing Studies. http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/UserCode/wangdy/?cvsroot=CMSSW.

[19] Jay Hauser et al. CSC Strip, Wire, Chamber, and Electronics Conventions. CMS Internal Note IN-2007/024, CMS Collaboration, 2007.

A Route A and Route B in naive estimation

There are different routes of comparison to get the timing shift constants for 18 chambers in ME2 and ME3 together with naive estimation described in Sec. 2.2.3. Two of them have been implemented, denoted as Route A and Route B. The Route specifies the sequence in which the comparison results of neighboring chambers are accounted for to get the overall timing shift for each chamber. From studies with data from MTCC, Route A has better precision and serves as default route and Route B serves for consistency check. Here are their meanings:

- Default Route (denoted as Route A):
  Station2Chamber5-9→Station3Chamber4-9→Station2Chamber1-3→
  Station3Chamber1-3→Station1(a,b)Chamber1-9
- Comparison route (denoted as Route B)
  Station2Chamber5-9→Station1(a,b)Chamber1-9→Station2Chamber1-3→Station3Chamber1-9

B Software and Data Set

B.1 Descriptions of Major Packages

B.1.1 TimingCSC

This package consist of CSCTimingAnalyzer, an EDAnalyzer under CMS offline software framework CMSSW[13]:

- Processing different chamber combinations used later.
- Extract nave estimation of the timing constants chamber by chamber, error bars are considered according to different routes.
- ROOT file I/O: quantities retrieved from CSCTimingAnalyzer stored in Trees, as input to analysis modules below.

B.1.2 Dedicated Python Modules

The algorithm to extract timing shift constants are all written with Python programming language, in the form of modules. Major benefits from this choice are :

- With NumPy library[14], the modules can be extensible easily, while performances remain equivalent to other scientific libraries such as GSL[15].
- The libraries are available to everyone, in CMS maintenance.
- Easier for the configuration: Python modules are highly configurable at runtime, with inline documentation.

Below is some brief introduction to the major modules:

findana and findmin : These two modules provide automatic tools to scan over all possible combinations and find the best choices of the algorithm(see above), for matrix method and minimization method, respectively.

tfconst : This is the core module to perform the algorithm and extract the timing constant for each chamber. Its input is TimingConst.root from CSCTimingAnalyzer under CMSSW, and it implements all the three methods described in Sec. 2. For each chamber in the data taken, there is one timing constant for it to account for the cable length and other effects. A global parameter could also be included to account for time of flight between different stations (Optional). This module can be configured at runtime. It could automatically determine all the invalid measurements and excluded them. Users can disable some parameters by keyboard input. For minimization method, users could also choose to start from zero or naive estimation values.

runstable : This module performs the check for stalility of timing constants for the specified list of runs, for all the three methods with different runtime configurations. It also performs transformation of reference chamber and automatic locate and load necessary results out of tfconst.

fillroot :With the increase of numbers of combinations used, there is a bigger probability of not having enough statistics for some channels. This module helps the development and debugging of the algorithm by providing a randomly filled "fake" data in the same format.

timing_intersector :This module handles inter-chamber timing as described in 2.3.

B.1.3 Documents and CVS

For more details and ongoing activities of this project, there is a Twiki page at [16].

I maintained a webpage for some CSCTF related softwares at [17] , there contains Doxygen document for package of TimngCSC. The document for python packages to be added later.

In order to ease the development and maintainance of the packages, I have put them into CVS repository at [18].

B.2 Data Set and Requirements of Statistics

The algorithms were first developed and tested with MTCC runs. For the minus end Slice Test, the requirements on phi matching parameters are loosened. For all the algorithms, 1M events each run are enough to get reliable results.

The aim is to expand this program to deal with increased number of chambers and ultimately to the whole 468 CSCs. In order to reduce the requirements for data set, the ME2-ME3 analysis was tried with fewer events. It showed that 300k events can give nearly the same result for the run618, with biggest error bar from 0.01BX to 0.022BX. If 100k events are used, biggest error bar is about 0.036BX. Considering the best possible precision we can achieve is about 0.05BX, 300k events is the recommended size for the dataset (2–3 sigma). If necessary, data set of 100k to 200k is also acceptable, providing there is no similar problem as run11961 in minus end Slice Test. (see Sec. 4.4)

For ME1 standalone algorithm, as described in Sec 4.1, only several cosmic events could be collected penetrating ME1/1A and ME1/3 out of 1M events of MTCC Run 486.

In the case of combined algorithm of the whole sector, the requirements on the statistics could be much looser. At present(Dec 2007) with runs on surface, 1M cosmic events are well enough to obtain reliable results. Since the cosmic rays rates in underground tests are much lower, the requirements may be reduced to 300-500k. Studies with simulated cosmic samples inside UXC show the event rate of CSC is about 0.2-1Hz/chamber as shown in left plot in Figure 18. This has been confirmed later in local/global runs with plus endcap CSCs, as shown in the right plot in Figure 18.

From these rates, a run with enough statistic may take about 4 hours. This doesn't include many practical issues, and may be optimistic. In order to further reduce the statistical requirements, additional handling before the algorithm was tried for chambers with lower rates(such as inner rings, as shown in Figure 18). For the combinations with no events or very few events, the mean value of BX difference is set to zero, with a bigger error estimation than those with more events. Except for some extreme cases, this works fine. That means the statistical requirements could be reduced to 150k in one trigger sector, corresponding to data-taking time of 1-2 hours. Several runs with this size of sample were confirmed with good results, see Sec. 5.

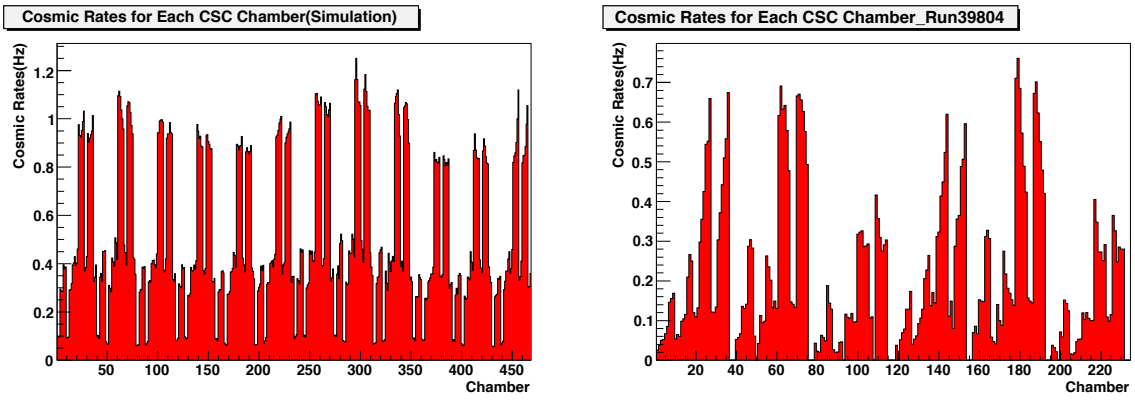

Figure 18: Cosmic rates in CSC chambers in UXC. Left: estimation for two endcaps from standard CMSCGEN cosmic simulation samples with total rates of cosmic muon 650Hz; Right: cosmic rates observed during local Run29804 (plus endcap).

C Estimation of signal time propagation effect across chambers

Since the input to the relative CSCs timing comes from the event when particles cross the overlapping regions of neighboring chambers it is important to take into account the time propagation of the anode signals along the wires. This effect related to particular design of the CSCs - the anode front end boards (AFEBs) are located at the one side of the chambers. In case of a muon crossed the overlapping region, an anode signal was created near the AFEBs of one CSC , but an anode signal in the another chamber must crossed the whole chamber to reach the AFEB, as one can see from the Fig.11.

The corrections have been calculated base on the assumption that cosmic ray particles uniformly distributed across the sensitive volume of a chamber. Then the average efficient length of the wires has been calculated for each CSC type.

$$L = \frac{\int l^2(h)\, dh}{\int l(h)\, dh} \tag{7}$$

where L is average efficient wire length, l is the wire length at the distance h from the base of the chamber.

Finally, the time correction was calculated as $T = k \times L/c$, where c is the speed of light in vacuum, k is the coefficient, which takes into account the size of the overlapping region for cosmic ray particles. It was found from the results of the single chamber trigger run, which was taken during MTCC test. In the Figure 19 the strip occupancy in the overlapping region is shown for the chamber, which was not sending triggers.

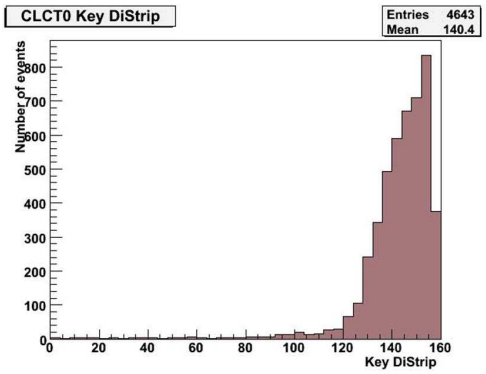

Figure 19: Strip occupancy in the chamber which located next to the trigger chamber

The absolute values of corrections finally used for different types of chambers are as following:

```
ME1/2     T = 0.072 bx
ME1/1     T = 0.044 bx;
ME1/3     T = 0.085 bx (this is not really used, since no overlapping)
```

```
Others    T = 0.104 bx
```

Due to the different positions of AFEBs on the chambers of different stations, the corrections with different signs are applied accordingly. The corrections for stations ME1 and ME2 are positive and those for stations ME3 and ME4 are negative.

D Details of Input combinations for one trigger sector

CSC chamber numbering scheme is defined in note[19]. Figure 20 shows CSC numbering scheme within one trigger sector.

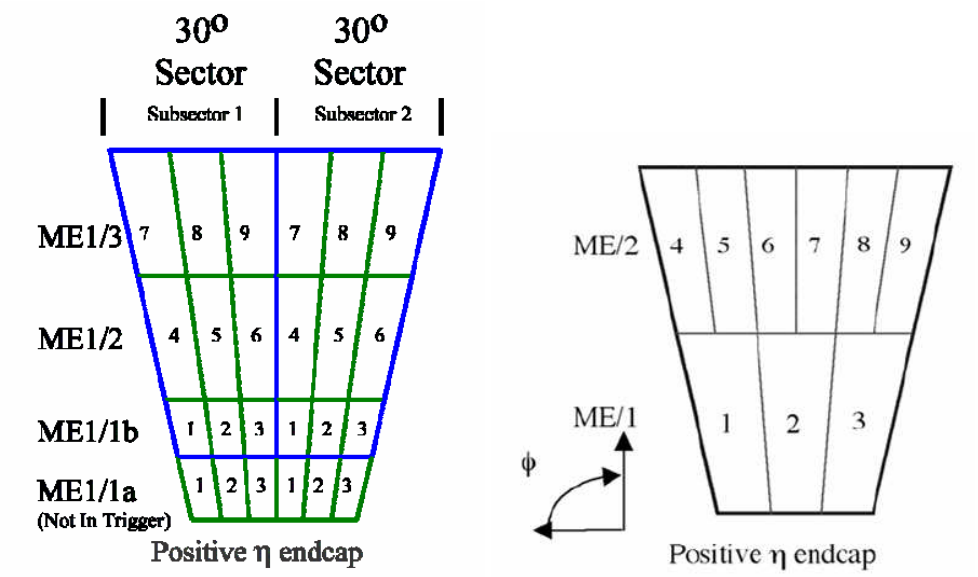

Figure 20: CSC numbering scheme within one trigger sector. Left: for ME1 chambers; Right: for ME2/3/4 chambers. For more details see[19].

The present algorithm can deal with all chambers of one sector together. There are 118 combinations used in total, 43 among which are for ME2 and ME3, 42 for ME1 only and another 24 for ME1 and ME2, also 9 for ME3 and ME4. For the convenience of description, these combinations are classified into several groups:

- Combinations between different rings within ME1A or ME1B. Due to the geometry restrictions, the realistic combinations are restrict to between ME1/1 and ME1/2 or ME1/3. There are 7 each for ME1/1-ME1/2 and ME1/1-ME1/3 combinations, so totally 28 combinations in this group.
- Combinations between ME1A and ME1B chambers. Since there is no overlap between neighboring chambers in ME1/3, so there are totally 6 combinations in this group.
- Combinations between chambers in the same rings in the same stations. There are 4 each for ME1A and ME1B, 7 each for ME2 and ME3, and 2 for ME4, so it adds up to 24.
- Combinations between chambers of ME1 and ME2. There are 6 for ring 1 and ring 2 respectively, for both ME1A and ME1B. So the total number here is 24.
- Combinations of chambers between ME2 and ME3. Total number of combination is 29. They can be divided into three subgroups:
  1. Chambers with the same CSC ID in ME2 and ME3. Total number is 9.
  2. Chambers in the same ring in ME2 and ME3. There are 4 for ring 1 and 10 for ring2. Total number is 14.
  3. Chambers in different rings in ME2 and ME3. 6 combinations are used at present.
- Combinations of chambers between ME3 and ME4. Total number of combination is 7. They can be divided into two subgroups:

1. Chambers with the same CSC ID in ME4. Total number is 3.
2. Chambers in the same ring in ME3 and ME4. Total number is 4.

E Recipes for Running the Inter-Chamber Timing Program on CSCTF Data

We assume people are working in an standard CERN computing environment, for example, on lxplus. The following steps should be followed from scratch to get some first results of chamber-to-chamber timing from CSCTF data.

Any problem/comment/feedback, please contact me.

E.1 Preparations: Establish a CMSSW working area and check out all necessary codes/tools

E.1.1 Establish a CMSSW working area

Below is the basics to establish a CMSSW working area:

```
scramv1 project CMSSW CMSSW_1_8_0

## Set your runtime environment:
cd CMSSW_1_8_0/src
eval ‘scramv1 runtime -csh‘
## or
eval ‘scramv1 runtime -sh‘ ## depending on your shell

## Set the CVS root so that you can access modules from the CVS repository.
project CMSSW
```

E.1.2 Establish extra environments needed

- ROOT: if you are working on lxplus, ROOT is already there; In other cases, just follow the instructions from ROOT homepage (http://root.cern.ch) to establish related environments for ROOT
- Python: if you are working on lxplus, you just need to execute:

  ```
  Source /afs/cern.ch/cms/sw/slc4_ia32_gcc345/external/python/2.4.3\\
  /etc/profile.d/init.sh;
  ```

  Othercases, you should the follow instructions from Python homepage (http://www.python.org)
- Numpy: Since the program uses Numpy, a numerical package written in Python, you should get a Numpy and establish environment first. If you are working on lxplus, you just need to execute:

  ```
  source /afs/cern.ch/cms/sw/slc4_ia32_gcc345/external/py2-numpy/\\
  1.0.1/etc/profile.d/init.sh;
  ```

  Othercases, you should follow the instructions from Numpy homepage (http://numpy.scipy.org/)

E.1.3 Checkout necessary codes/tools

You also need the following components to get things running:

- CSCTF Unpacker: usually there is already the unpacker in standard CMSSW release, but sometimes there are some latest changes/patches not covered there. So you may need to check out the latest version to your CMSSWHOME/src: EventFilter/CSCTFRawToDigi.

- TimingCSC: you should check out the latest version to your CMSSWHOME/src: EventFilter/TimingCSC. Note the codes are not in CMSSW repository, but at UserCode/wangdy/TimingCSC.
- pygram: you should check out the latest version to wherever you like(not necessarily under CMSSWHOME). Note the codes are not in CMSSW repository, but at UserCode/wangdy/pygram.

You need to compile the two packages of CSCTFRawToDigi and TimingCSC before you go any further.

E.2 Unpack CSC Track Finder Data and Pre-processing

Now you have the tools ready. Let us go to the real job step by step.

E.2.1 Get the raw data

First, you should get CSC Track Finder raw data to your working area; consult experts for details if you don't know where to get them. Since the raw data sometimes are big, you may need to put them to CASTOR or some tmp areas if you work on lxplus to avoid quota excess.

In the following context, I assume that you have put raw data in tmp area with the following name conventions:

```
/tmp/yourusername/XXXXX/csc_000XXXXX_EmuRUI00_Monitor_000.raw
```

Here(and in all context below), XXXXX is the 5-digit run number

E.2.2 Unpacking and pre-processing

- Go to CSCTiming/test area, execute the following command to unpack data:

  ```
  ./dyrunRawToDigi XXXXX
  ```

  You will find the digis in dir of Generated_Digis/RunNum_XXXXX

- execute the following command to pre-process data:

  ```
  ./dyrunAnalyzeDigi XXXXX
  ```

  You will find the necessary histograms in dir of Generated_Histograms/RunNum_XXXXX. Some related plots are generated in dir of Generated_Plots/RunNum_XXXXX.(Note: this option is turned off at present to reduce memory usage, you can uncomment certain part of the codes to enable it, maybe I will add a flag in cfg file later)

- I also provide some script to do the above steps together and put the data onto CASTOR, etc. You may need to modify a bit to fit your need.

E.3 Get Timing Information From CSCTF Data

Now comes the last steps to get the results.

Note The steps described in this section are applicable to the following version of components:

tfconst version 1.20 and earlier ones

findana version 1.3 and earlier ones

runstable version 1.12 and earlier ones

- go to the dir of pygram, and make a subdir: output there

- copy the histograms with the name of "***TimingConst**.root" you get to the present directory of "pygram". At present, there are 6 of them, corresponding to the present +Z Endcap runs. Do the following steps for each of them:

  1. Rename the root file under stuty to "TimingConst.root"
  2. type the commands: "python tfconst.py"
  3. there will be prompt for you to put the run number (you'd better also put sector info into it), disabled chambers, reference chamber, need plot or not, method used and which station combinations. They are easy to follow, and just put in your choice or press "enter" to use default values.
  4. finally you get the results in subdir of output. They are available in both numerical and graphical formats(eps and png, you need to choose graph option in previous step of course)

- There are also several other python tools to use in the same directory, but they are more or less in a "expert" level. When you finish the above steps, usually you already get the basic results you need.

E.4 Conventions of the results

For the routine use of the algorithm, "findana" module should be used first to determine the best reference chamber. Then specify this when start "tfconst". After this algorithm, both the numerical and graphical results are generated in local directory of "/output". Module "runstable" could be run as a next step (optional), generating results in local directory of "/result".

The file containing the name of "tconst_anaXXX" is the timing consts of the chambers for runXXX, in it the chamber number is defined as Sector_Id*100+Station_Id*10+Chamber_Id. The files containing "hStabXX" shows the const stability of chamberXX for these runs. If the chamber is not available in all the runs, it is not include. The numbering in this case is just simply from 1 to 18 (or from 1 to 36).

E.5 Get global results with batch jobs

The steps described in the above subsection E.3 can only handle one trigger sector at a time. There are more advanced options to handle the problem for all 6 sectors in one endcap and many runs in batch mode, and together with inter-sector results, a compiled results can also be obtained in the end. To do this, you need the following extra preparations:

- Software versions later than those spefified in E.3.
- Prepare the root results for all 6 sectors in advance and put them under /data subdirectory in your working directory.
- If you want to publish the results to the webpages as I do for underground local/global runs, you need to get some more scripts in the same directory, modifiy the directory used in your case and choose all the publication options.

It is also possible to handle part of the endcaps, to handle data from limited number of stations, to publish the results to places other than default ones etc. Some of these options are ready to choose directly at runtime from prompt. You can follow the docstrings in the programs for further customizations.

I hope this recipe is helpful for the interesting people to get started in using the program. Good luck!