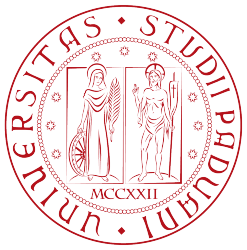


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO
DI
FISICA E ASTRONOMIA

Corso di laurea magistrale in fisica

A study of L1 track trigger algorithms for the CMS detector at HL-LHC

Relatore: Dott. Roberto Rossin
Controrelatore: Prof. Roberto Stroili

Laureando: Giorgio Vidale

Anno Accademico 2016 / 2017

Index

Chapter I CMS tracker upgrade

The phase II upgrade of CMS experiment scheduled for the 2025 aims to increase the integrated luminosity up to 3000 fb^{-1} in order to achieve a massive boost in statistic acquisition in respect not only to current standard (30 fb^{-1}), but also with respect to the planned phase I upgrade, that will take the luminosity up to a factor 10 with respect to current. The scale of the upgrade in both in technical complexity and size such that a fair comparison with the very original CMS construction challenge would not be inappropriate [1].

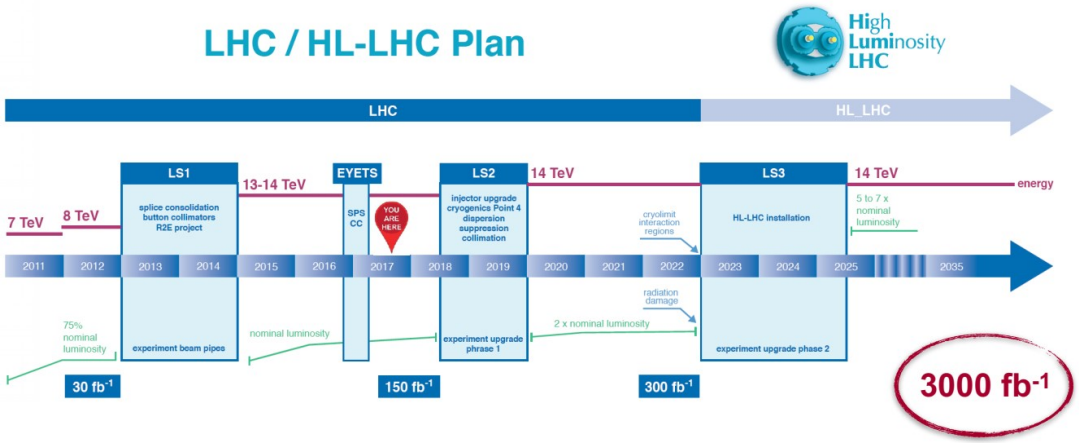


Figure 1 – HL – LHC upgrade schedule

The High Luminosity prefix stresses the main goal of the upgrade: not a boost in energy center of mass, but a boost in the amount of data the machine will be able to produce. This goal, along with the forthcoming technical improvements, is mandatory in order to overcome current machine limits that are already evident when exploring some of the most challenging physical themes that CMS is going to deal with. In fact, the current precision, due to low statistics or – in some processes – the fact the machine is already hitting the systematic wall [1], is not sufficient to probe many different aspects in both standard model and new physics research: in particular, many aspects about the physics of Higgs boson are still open, e.g. the Higgs couplings to the other SM particles have been measured only for a fraction of them, and the self coupling is yet to be measured; moreover, questions raised by theories beyond SM about the existence of different Higgs bosons are yet to be answered. Apart from Higgs searcbes, also searching for the existence of new physics in difficult parameter regions is exposing the limits of current machine. A boost in luminosity will provide massive help by making possible the "hunt for the very rare" or the attack of new difficult regions of phase space; moreover, more statistics will provide useful information to reduce current systematic uncertanties as well.

Raising the instantaneous luminosity up to ~ (5 – 7.5) 10^{34} cm^{-2} s^{-1} will take the number of simultaneous inelastic proton – proton collision per bunch crossing ("pile

up" event, PU) up to ~200, if estimating a total cross section for such processes of 80 mb and a bunch crossing rate of 40MHz [1]; since each collision is likely to produce ~6 charged particles per unit in rapidity, over a range of ± 5 units in rapidity up to 10^4 particles are expected to be produced; the line vertex density is expected to raise up to 2 mm^{-1}, almost a factor 2 over current value.

In order to be able to process efficiently such amount of data, an update of the CMS detector is mandatory; the tracker, among the sub-detectors which will be upgraded, plays a special role: not only it will be fully updated, but it will also be included in the L1 trigger decision chain.

Section 1.01 CMS detector – A brief overview

The detector requirements for CMS to meet the goals of the LHC physics programme can be summarized as follows [2] [3]:

- Good muon identification and momentum resolution over a wide range of momenta in the region $|\eta| < 2.5$ and the ability to determine unambiguously the charge of muons with $p < 1$ TeV/c.
- Good charged particle momentum resolution and reconstruction efficiency in the inner tracker. Efficient triggering and offline tagging of τ's and b-jets, requiring pixel detectors close to the interaction region.
- Good electromagnetic energy resolution and measurement of the direction of photons and/or correct localization of the primary interaction vertex.
- Good E_T^{miss}, requiring hadron calorimeters with a large hermetic geometric coverage ($|\eta| < 5$) and with fine lateral segmentation

At the heart of the experiment is a 3.8 T superconducting solenoid providing large bending power for momentum measurements of charged particles and whose return field is large enough to saturate the iron plates in the return yoke, enabling it to be used for muon momentum reconstruction; the gaps between the plates provide slots for four muon tracking stations. The bore of the magnet is large enough to accommodate the tracking and calorimetry systems. Current CMS silicon microstrip tracker, combined with the strong solenoidal field, provides the required granularity and precision to reconstruct efficiently charged tracks in high multiplicity events and to achieve excellent momentum resolution. In addition three layers of silicon pixel detectors in the barrel region, complemented by two forward disks at each end, seed track reconstruction and improve impact parameter measurements, as well as providing points with sufficient resolution to reconstruct secondary vertices from decays of particles containing b and c quarks. The electromagnetic calorimeter (ECAL) provides coverage up to pseudorapidity $|\eta|=3$ and uses blocks of lead tungstate crystals whose scintillation light is detected by silicon avalanche photodiodes (APDs) in the barrel and vacuum phototriodes (VPTs) in the endcaps. The ECAL is surrounded by

a brass/scintillator sampling hadron calorimeter (HCAL) with coverage up to $|\eta|=3$. Coverage up to $|\eta|=5$ is provided by an iron/quartz-fibre calorimeter (HF), that ensures nearly full geometric coverage for measurement of the transverse energy in the event.

CMS is triggered by dedicated custom electronics which currently form various partial triggers using trigger primitives from the front ends of the calorimeters and muon detectors; these are then sent to the Global Level 1 trigger which is designed to handle up to 100 kHz rate with a latency of 3.6 μs. Data must be stored on detectors during Level 1 processing; when a Level 1 accept occurs, data fragments from individual detectors are sent to the High Level Trigger (HLT), operating on a large computer cluster, to build complete events. The HLT performs a lean version of the off-line reconstruction using full event data and uses the result to decide if the event should be written, together with trigger information, to mass storage for subsequent analysis. A detailed description of the CMS detector is given in reference [3].

Given the high luminosity level target of HL-LHC, Phase two upgrade consists in various modifications to the detector described above, including a complete reworking of the silicon tracker and the modernization and upgrade of the trigger and data acquisition systems to handle higher data volumes. In particular, the new tracker is including: a pixel inner tracker made by 4 barrel layers and 24 disks to get coverage up to $|\eta| = 4$ for vertex reconstruction; a pixel and strip outer tracker made by 6 barrel layers and 10 disks to get coverage up to $|\eta| = 2.4$, which is going to be used to track triggering purposes. A detailed review on the new outer tracker is in section 1.03.

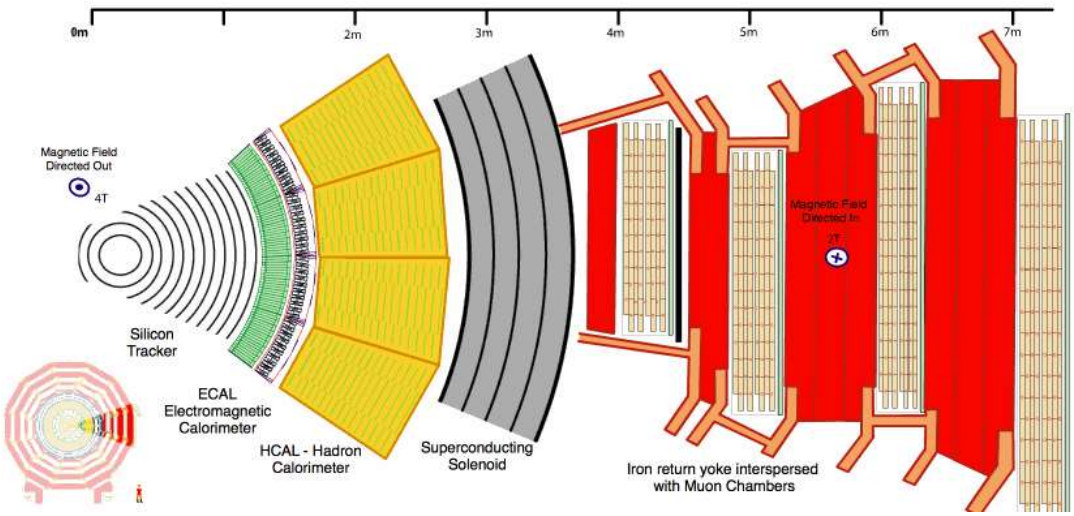


Figure 2 - Schematic representation of CMS detector: the tracker layers, ECAL and HCAL and the muon chamber are shown.

Section 1.02 Need for Speed

In order to satisfy the restrictive triggering time specifications, both hardware and algorithmic solutions at trigger level have been adopted.

On hardware level, a threshold p_T filter directly implemented in tracker sensors is mandatory in order to feed the Level 1 trigger with hits only coming from tracks with p_T higher than the threshold. This is achieved by implementing in the outer tracker double-layer pixel sensor modules [4], whose granularity is sufficient to discriminate track p_T : as seen in Figure 3, when a track crosses a module, it leaves clusters of charges in both the upper and lower layer of the module; the readout electronic is responsible for deciding whether the clusters of charges present in the two sub-layers are coming from the same track, i.e. if their pixel position do not differ more than a few unit pixels; if a coincidence is found, then a "stub" is made. A stub is an object that define the track's hitting point on a module. A p_T selection on stubs is possible by discriminating over clusters of charge separation between the 2 sub-layers.

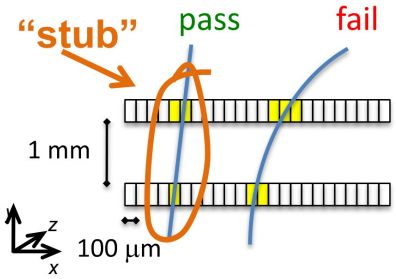


Figure 3 – The idea of "stubs".

A fast and efficient triggering level 1 system is also mandatory in order to select only interesting tracks, provided by the tracker. In fact, found stubs are sent out at each bunch crossing, while the full event is stored locally and is sent out only if requested with a L1 trigger signal.

The new L1 trigger will be able to provide event selection with a rate of up to ~750kHz with a total latency of 12.5 µs [1]. The actual L1 trigger algorithms operate on "primitives", e.g. candidate tracks made by combinations of stubs coming from the tracker layers that are likely produced by the same track. Track trigger Primitives are built from stubs output coming from tracker's modules and different ways to obtain them exist indeed; other examples of primitives are e.g. calorimeter clusters, muon candidates. The total latency allocated for the track reconstruction is about 4 µs. The current upgrade program is oriented toward the implementation of a massive collection of FPGA cards in order to perform track recognition: one approach consists in using FPGA to perform over Hugh transforms together with a Kalman filter in order to select valid stubs combinations; another one makes use of FPGA in testing stubs couple in adjacent layers and extrapolate possible tracks [1]. But different approaches do exist indeed: in this thesis, starting from chapter III, the performance of an Associative Memory approach will be investigated.

Currently, the groups involved have been asked to perform stress test over highly dense events (PU up to 800), in order to better understand the track triggering performance in extreme conditions.

Section 1.03 The outer tracker layout

The selected layout geometry for the tracker is showed in Figure 4 [2]; only the outer tracker will be considered in the AM approach study discussed here.

The outer tracker has cylindrical symmetry around the beam-axis (z – axis), and reflection symmetry around the ideal collision point, at the origin of the coordinate system, as shown in figure. The "barrel" is made by 6 layers parallel to the beam axis, symmetric with respect to $z = 0$; the "endcap" is divided in forward part and backward part, each one made by 5 layers, perpendicular to the beam axis. Each layer is made by sensor modules partially overlapping in order to assure complete coverage both along z-direction and φ-direction.

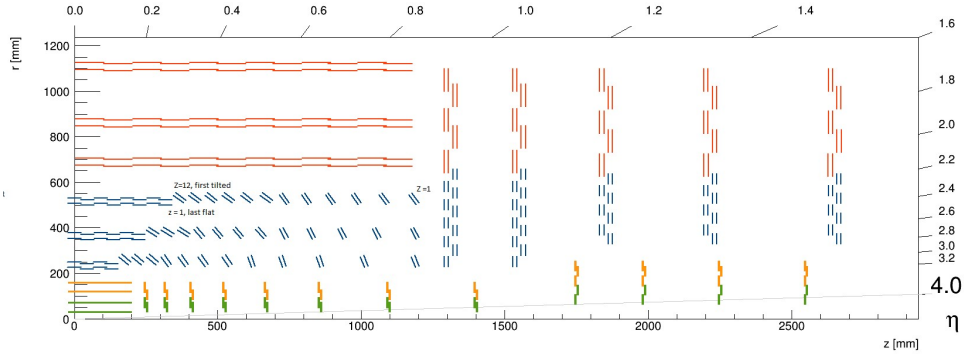


Figure 4 – Layout of the tracker modules: the inner tracker (green and yellow) is made by pixels sensors, while the outer tracker is made by modules built using both pixel and strip sensors.

As said in the previous section, each module itself is actually made by two sub-layer, separated by 1 ÷ 4 mm; with respect to the granularity of the two sub-layers, two kind of modules are implemented [4].

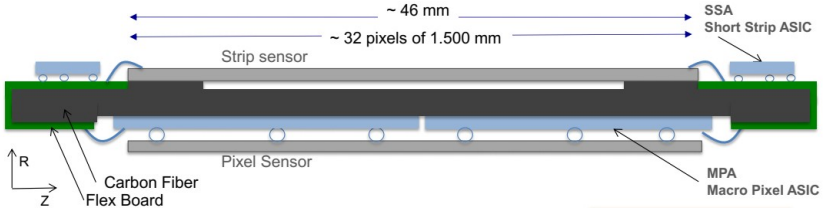


Figure 5 – PS module: the two sub-layers are visible, the upper one made by strips while the bottom one made by pixels. The readout electronics is also visible: the two segments along z which each strip is divided into are read by two indipendent Application Specified Integrated Circuits (ASIC); also the 32 pixel segments are read by two independent ASICs.

- *PS* modules (shown in figure): they are composed by a Pixel and a Strip layer. The strip layer is made by 1920 strips segmented in two parts (2 × 960), each part 25 mm long at 100 μm pitch, that provide optimal resolution in the φ-direction. The pixel layers is made by sensors segmented in 32 parts, that provide good resolution in both φ and z direction, in fact the pixel layer is made by 32 × 960 pixels 1500 μm long at 100 μm pitch. The

stub coordinate info, if a coincidence is found between hit in the two sub-layers, comes from the pixel.

- *2S* modules: they are similar to PS modules, except for the fact that both sub-layers are composed by strip sensors, hence they provide good resolution only in φ-direction.

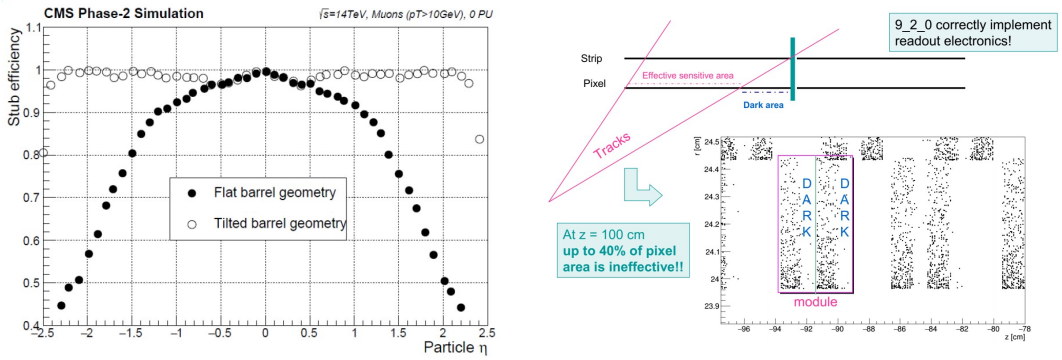


Figure 6 – On the right, schematic representation using simulated tracks of the "dark zones" in the center and at the side of modules. The two halves of each layer of the sandwitch don't communicate, thus a coincidence between hits in upper and lower sublayer is impossible when particles with high angle of incident hit modules in the center or at he side. On the left [12], the efficiency gain of tilted solution for layer 1 in the barrel, with single muons @ p_T > 10 GeV/c.

In order to implement the connectivity between the upper and the lower sensors with reliable and affordable technologies, the readout ASICs are placed at the two sides of each module: each half of the two sub-layers that constitute the module sandwich is read independently by the corresponding side ASIC. The drawback of this solution comes from the fact that the ASIC responsible for the readout of each half doesn't communicate with its counterpart on the opposite side of the module, thus reducing the stub reconstruction efficiency for high η tracks that cross the module near the center down to 40%. A schematic representation of this effect over flat modules in visible in Figure 6.

In order to overcome this limitation, the latest proposal for the tracker layout features progressively tilted modules in the first three barrel layers and up to 60 cm in the r-direction onto disks, thus intercepting tracks with psudorapidity nearly perpendicularly; in the outer layers, due to lower incident angles, there's no need of tilted modules.

The efficiency gain over the "dark zone" near the center (along with the one on the side, due to the fact that the two sub-layers are perfectly aligned), using tilted modules is visible when comparing simulations results (Figure 6) obtained with tilted geometry (the one investigated in this thesis), and the old flat geometry (used until now to characterize AM approach performance in the barrel).

Chapter II Tower division

For triggering purposes, the outer tracker needs to be divided into (almost) stand-alone sectors called trigger towers (TT). The AM approach adopts an eight-fold division in φ – each tower having approximately an azimuthal opening of $\Delta\varphi = \pi/4$, and a division in six sections in pseudorapidity η, whose width is to be optimized, in the range [-2.4, 2.4]; as such, the tracker is divided in 48 towers. Each tower consists in a list of tracker modules and is handled independently with dedicated hardware that processes stubs coming from them. One of the goal achieved is to optimize each tower's module list, so that:

- the number of modules shared by adjacent towers – whose stubs are going to be processed more then once – is minimized;
- track reconstruction efficiency is optimized with respect to actual number of modules belonging to the tower, by reducing the number of tracks that escape from the tower due to their curving trajectory.

Section 2.01 Tower definition criteria

In order to obtain the optimized list of modules belonging to a trigger tower, a "dual space" tower definition has been used: "dual" means that is not based on the geometrical space of the modules, but on the parameters phase space (φ_0, η, z_0, charge*p_T) of tracks. Over such space, the parameter distribution functions (pdf) for the 4 parameter are: for φ_0, flat in [-π, π]; for η, flat in [-2.4, 2.4]; for z_0, Gaussian centered in 0 cm with σ = 5 cm; for p_T, flat in $1/p_T$ over [0.3, ∞[. A tower can be defined by a volume in parameters phase space; the corresponding volume in geometrical space is retrieved by mapping the dual tower volume to the geometrical tower volume, i.e. the space covered by all the possible paths of particles whose parameters belong to the tower's dual volume. The modules list is then retrieved by taking the modules that belongs the tower's geometrical volume. It must be stressed that the geometrical definition (using modules) and the dual definition, although different, are in fact defining the same tower: in the dual space each tower is defined as an exact partition of the track parameter space while, in the tracker physical space, as a list of modules; due to modules finite dimension and to optimal coverage needs, it is likely that towers overlap in the physical space. If towers overlap, they share boundary modules, thus reducing the global effectiveness of tracker division in trigger towers.

The tower dual space volume is defined so that at radius R* (to be optimized) the ideal track's $\varphi(R^*)$ belongs to a desired azimuthal interval and $\eta(R^*)$ to a desired pseudrapidity interval, as shown in Figure 7. In comparison with the more trivial request that particle must belongs to a desired azimuthal interval valuated at $R=0$, the R^* definition permits to maximize the number of tracks that remain inside the same

tower for the full length of their motion in the tracker; otherwise, it is likely that a track, that at *R=0* falls into the desired azimuthal, escapes from the geometrical tower volume, due to its curvature path. Another advantage, clear in figure 1, is that by using *R**, the overlapping region between adjacent towers in minimized.

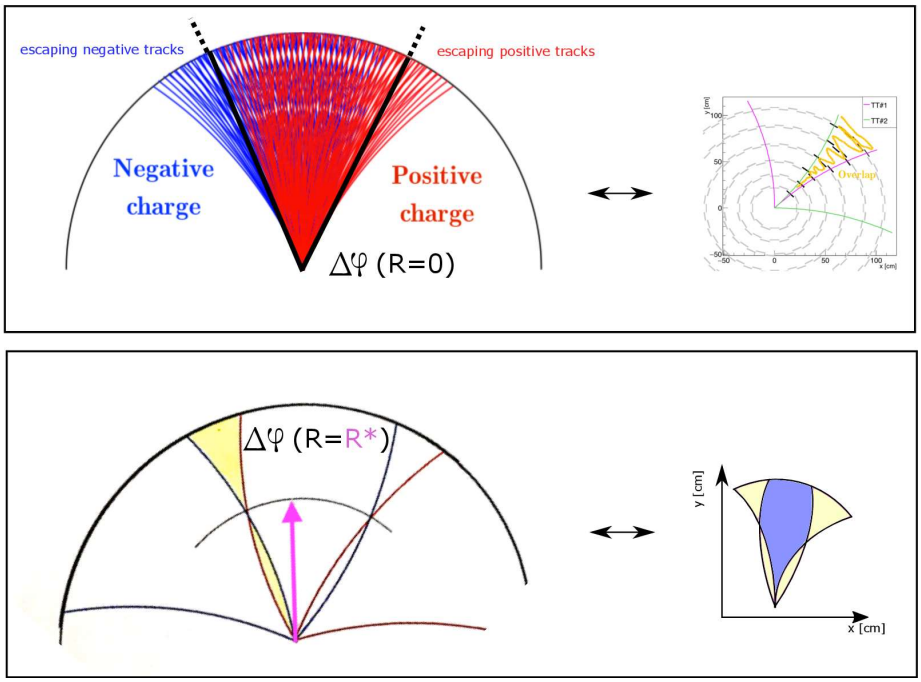


Figure 7 - – Tower definition using R= R* instead of R=0. In the upper figure, disadvantages of using Δφ (R=0) are shown: highly bending tracks are likely to escape tower geometrical space; adiacent towers shares many modules. In the lower figure, R* definition is presented: higly bending tracks are now likely to remain inside the tower, and sharing modules are fairly reduced, with respect to optimal R*.

Optimization of module list includes the minimization of shared modules between towers by picking only modules that are shared between no more than 4 towers, and the choice of the best R*: previous studies have confirmed optimum R* ~ 90 cm [4].

φ and η tower division ranges are set with respect to tracker layout cylindrical symmetry in the transverse plane and with respect to the tracker division in barrel and disks in the beam direction; one of the goal is to chose the optimum η range so that stub count distribution has small variance between towers: this is achieved by tuning η boundaries and thus the total number of modules belonging to a tower.

Each tower is thus ultimately defined by boundaries in track phase space and by a list ofmodules with boundaries in physical space.

Section 2.02 Current tower division

Given the (almost perfect) cylindrical symmetry of the tracker and the z-axis reflection symmetry, there are 3 equivalence classes in η and efficiencies study are per-

formed using a representative of each class; the 3 towers selected respect the condition $\varphi(R^*) \in [\pi/4, \pi/4[$:

- *barrel* towers (TT25 is the class representative) include barrel only modules, with η range [0, 0.73] – as set in precedent barrel efficiencies studies;
- *hybrid* towers (TT33 is the class representative) include tilted modules in the barrel at higher η and the upmost part of disks layer, with η range [0.73, 1.46];
- *forward* towers (TT41 is the class representative) include mostly modules on disks, with η in range [1.46, 2.4].

Forward towers range is set so that towers include the most extension of the last two disks (layer 14 and 15), up to η = 2.4 (tracker η upper limit).

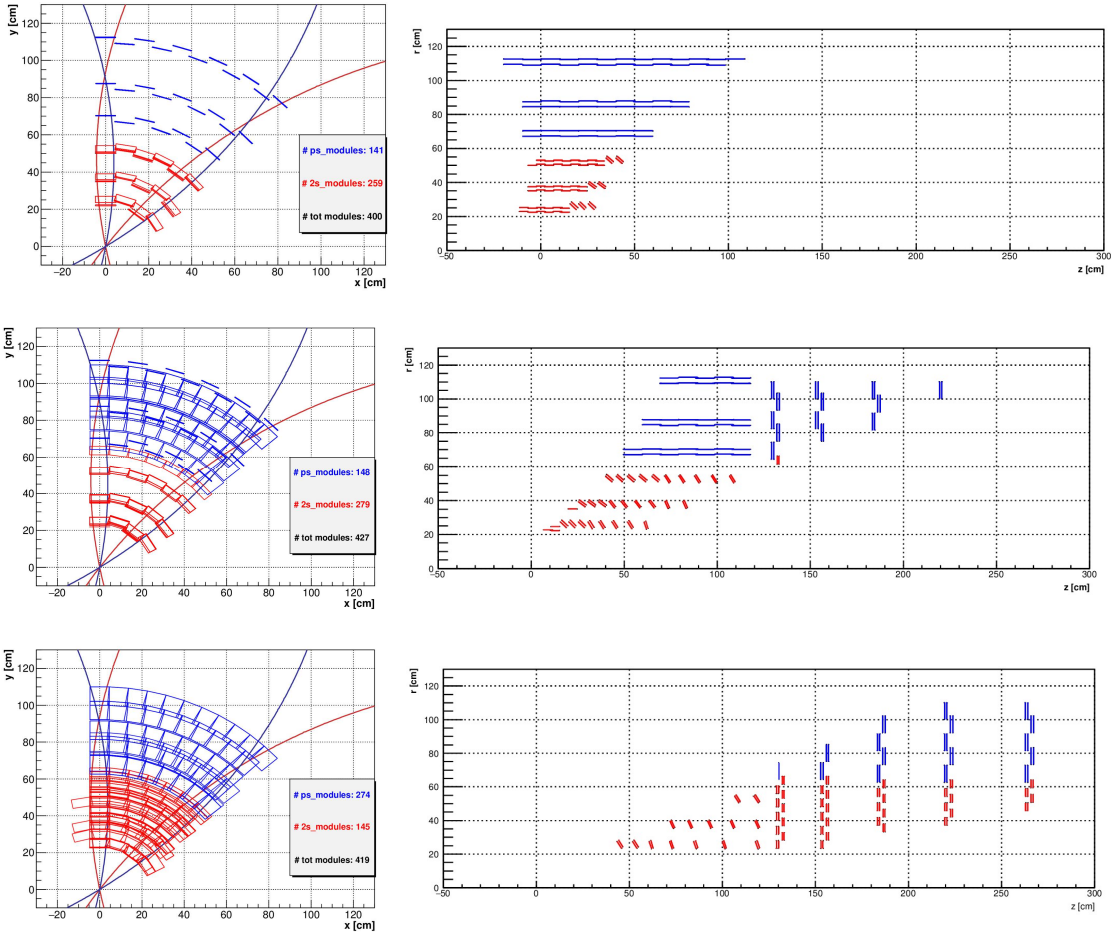


Figure 8 – Tower with phase space cut.

Chapter III AM approach

The fast track reconstruction time goal is achieved by degrading the full tracker stub resolution to a coarse grained one, in which stubs are collected into equivalence classes different for each layer of the tracker, called "superstrips"; an ordered list of superstrips coming from different layers defines a "pattern". The track recognition problem out of a cloud of stubs is thus rescaled to a pattern recognition problem in a volume of patterns – a much more time efficient process, due to its lower resolution.

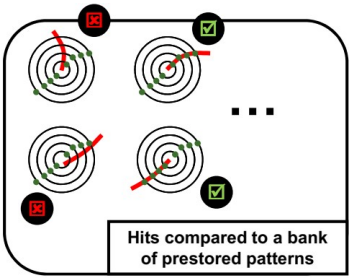


Figure 9 – Identification of a list of hits (green) with stored patterns (red).

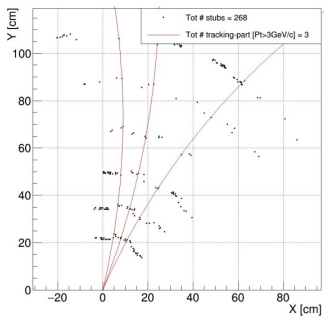
Pattern recognition is performed by confronting raw patterns coming from the

coarse grained representation of the tracker with a bank of ideal track patterns stored in the bank of an Associative Memory, as schematically sketched in Figure 9, where hits (in green) are compared to a list of patterns (in red).

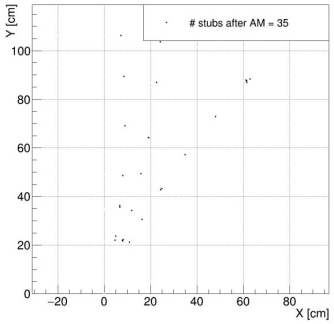
Using AM along with a trained bank of ideal tracks is only one of the possible way to recognize candidate patterns, but its strength relies in the possibility of using massive parallel processing when comparing raw patterns and stored ones.

Matching patterns are flagged as candidates for reconstruction, called "roads", and stubs belonging to them are sent to a fitting stage. Eventually, the full stub resolution is recovered from the coarse grained patterns, and track parameters are extracted by fitting all the possible combination of the stubs belonging to a pattern. The full recognition chain, from the tracker stubs cloud, to the final tracks candidate is show in figure's Figure 10 sequence.

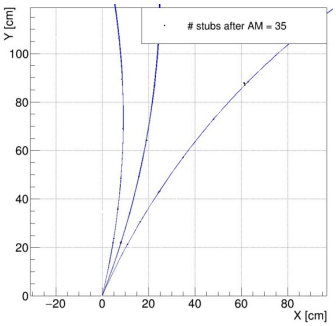
Figure 10 – Stub recognition process



Stubs in tower: 268 stubs in 1 event @ 140PU; 3 tracking particles (p_T > 3 GeV/c).



Roads: stubs only in matching patterns after AM pattern recognition (35 stubs).



After fitting stage, all 3 original tracks are found.

Section 3.01 What is an associative memory

The memory type referred as Content Addressable Memory (CAM or just Associative Memory) differs from conventional Random Access Memory (RAM) in the way data is retrieved: while in RAM stored data words (i.e. multiple bits patterns) are called forth from memory in terms of their address or storage location, in CAM they are selected on the basis of a key transmitted simultaneously to all them, and retrieved upon matching the key with a portion of the stored word data.

With suitable masking techniques, a key may be employed to match any portion or portions of the words in memory; upon recognition of the key, either the number of matching words, the whole words (or any part of it) themselves or the address at which the words are stored, may be retrieved nondestructively.

CAM, as recognition oriented memories, can be implemented in "associative processors" structures, with pattern recognition purposes. The most powerful implementation of these structures are the *bit-parallel/word-parallel* associative processors: a fully parallel associative memory incorporates a word-wide comparator in every row of the associative memory. The effectiveness of this implementation relies on the fact the each digit of the *mask register* is compared in parallel with each corresponding digit of all the words stored in the AM, simultaneously.

This kind of associative processor is used as the powerful track recognition tool at the core of the AM approach discussed in this thesis. In fact, the "words" stored inside in the AM are not made by single digits, as in the simple example discussed above: the stored words are the "patterns" that have been mentioned in the introduction of this chapter, and their "letters" are not single digits, but they are instead set of bits that uniquely identify the classes of equivalence that, in each layers, defines the coarse representation of the tracker, i.e. the superstrips.

Section 3.02 A coarse grained representation of the tracker

Each tracker layer is divided into a coarse grained reticolus made of superstrips; each superstrip is uniquely identified with an ID, called "ssID". In the current implementation a set of six superstrips, built by picking up one ssID from six different layers, defines the "pattern": a pattern is an ordered list of six ssIDs.

In the implementation discussed in the following chapters, a pattern is stored in the AM bank as a word composed by six ssID, each of them is made by 14 bits that encode also the unique address of a module in the tracker (layer, φ position, r or z position).

A particle track is associated with a pattern with respect to the ssID populated in each layer when the particle hit it, leaving a stub. A pattern is therefore populated when a track leaves stubs inside the ssIDs that build up that pattern.

Since tracker layers are more than six, it is possible that a track populates more than six ssIDs, when crossing more than six layers; that is likely to happen in the hybrid and forward tower; on the other hand, for tracks entirely confined in the barrel, that is impossible. Introducing a way to deal with this kind of tracks was mandatory in this study and will be discussed later on.

A fascinating way to represent a pattern is borrowed from statistical mechanics. In a 6-dimension continuous space Γ where the i-th oriented axis is the local i-th layer coordinate $\mathbf{l}_i$, a particle track is a dot defined by the vector $\mathbf{p} = (l_1, l_2, l_3, l_4, l_5, l_6)$. In the discrete six-dimension coarse grained representation space μ, each axis is now divided in discrete ssids intervals, whose length is measured along the local layer coordinates; a pattern is a hypercube in μ-space whose i-th side is long as the ss in the i-th layer. A pattern is then an equivalence class of particle tracks, that fell in the same hypercube in μ-space; the number of tracks that fell in the same hypercube is that pattern's multiplicity. In Figure 9 the μ-space is presented (although with only 3 dimensions); when varying only one of the four track parameters, a curve ϒ in Γ-space is defined: when it enters a pattern volume, it means that tracks obtained with that particular parameters configuration belongs to that pattern.

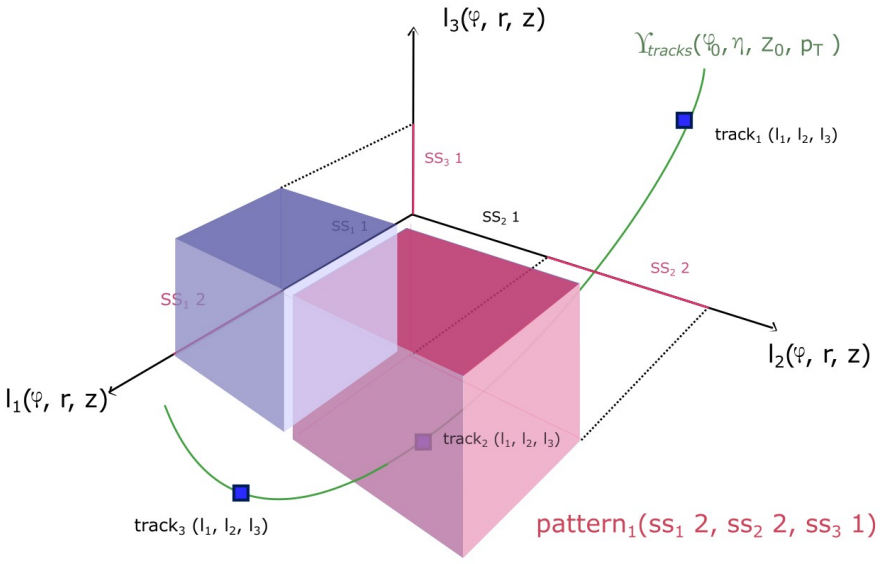


Figure 11 – 3-dimensional Γ-space (local layer coordinates (φ, r, z)) of single tracks is presented, with superimposed its coarsed grain representation μ-space (superstrip layer coordinates ss_i). The gamma tracks green curve is the path in Γ-space defined when varying only one of the four track parameters (φ_0, η, z_0, p_T). When varying all track parameters, covering the tower's dual space, a sub volume of Γ-space and ultimately a sub volume of μ -space are covered: it the subvolume of physical tracks and physical patterns, respectively. Purple cube is a sibling of the red one.

All physical tracks (whose parameters belong to tower's dual space) fall into a subvolume of Γ space; the patterns in coarse grained μ-space that covers that subvolume are the physical patterns. One of the AM approach advantages is to provide a bank with a selection of the physical patterns only: thus most of the random combinations

stubs coming from minimum bias event that don't trigger a physical pattern are not even taken into consideration, providing massive performance improvements.

The mu-space representation is particularly useful to introduce pattern related concepts, like "sibling" patterns or pattern multiplicity with respect to track parameters.

Section 3.03 Pattern multiplicity, sibling, combinations

By varying track parameters (φ_0, η, p_T, z_0) a locum in μ-space is described (in Figure 11, a curve Υ_{track} laying on such locum is painted in green). It is possible that tracks with close track parameters fell in the same pattern, thus increasing its multiplicity, i.e. the number of generated physical tracks that populated a given pattern. This is to be considered when optimizing the pattern generation process and the superstrip definition. If moving slowly along the locum when varying track parameters, the multiplicity of patterns crossed by the locum is increased.

It is very likely that in a real event with pile up, even physical patterns are partially populated by stubs coming from actually different tracking particle: patterns activated only by stubs coming from minimum bias events are called "combinatoric". The track candidate have to be extracted by fitting all possible combinations of stubs that populate a pattern, and then selecting only those that pass it quality criteria (such a χ^2 cut). Unphysical combinations, i.e. combination of stub coming from different tracking particle that populated the same pattern, will be in this way discarded.

If two pattern shares 5 out of 6 ssids, they are called "siblings": in mu space, all hypercubes aligned in the same direction are siblings; in the current configuration, each pattern has an average of eight to ten siblings, as observed with data set of single muons, without pile up.

Section 3.04 Recognition flow

The complete track recognition process flow is schematically represented in figure:

1. raw stubs info coming from the modules in the tower, e.g. Euclidean coordinates, p_T, module info, are fed into a FPGA;
2. in the FPGA performs the transcoding of stubs Euclidean coordinates into ssIDs; the ssID identify the superstrip which the stub belongs to, as well as the layer the module hit belongs to;
3. ssIDs are sent in the AM chip, to perform pattern recognition. Stubs belongings to output matching patterns are sent back to the FPGA to perform a fitting stage;
4. candidate track are eventually obtained.

Section 3.05 Pattern bank

The associative memory stores a bank of track patterns, that in our study are generated with Monte Carlo methods, i.e. the bank is trained with a sample of ideal mu tracks. Due to hardware bank size limitations, a popularity rule imposes which patterns coming from the training stage are stored in the AM: in case of a 64K bank size, only the first 64K most popular patterns are stored. The output order of matching pattern from the AM is also chosen by the pattern popularity. Since patterns are a set of six layer-ordered ssIDs, in the AM each pattern populate six layer-ordered slots, addressed with the corresponding ssID.

The bank is built by recording all the patterns populated by the tracks contained in the training sample; patterns are stored as a set of six ssid number and are ordered with respect to their multiplicity, also named "pattern popularity".

The full size of the untruncated pattern bank is ultimately determined by the graining the ssID layers subdivision which the tracker is divided into: the finer the ssIDs, the more pattern will be produced. One of the main goal of the AM approach optimization problem is to achieve the best efficiency with the smallest bank size; bank size considered in the study are of 64K, 128K and 256K pattern.

A bank critical attribute is its coverage, i.e. the probability that a real track has its corresponding pattern stored inside, hence found during pattern matching stage. An estimate of it is computed over the ordered set of patterns as a discrete cumulative function of pattern frequency distribution inside the bank:

$$C\left(P_{\max}\right) = C_{global} \sum_{P_i = P_1}^{P\max} \frac{n_{P_i}}{n_{tot}} \tag{5.1}$$

where C_{global} is the global bank coverage, estimated at every iteration of bank training as the probability that a new pattern not already stored in the bank is found in the training sample:

$$\left(C_{gloabl}\right)_{i^{th}\,iter.} = 1 - \left(\frac{n_{new\ patt.}}{\text{banksize}}\right)_{i^{th}\,iter.} \tag{5.2}$$

An adequately big training sample is mandatory in order to have $C_{global} \sim 1$.

When bank's global coverage is $C = \beta$, the frequency of the least populated pattern *P,* such that $P = C^{-1}\left(\beta\right)$ is taken as guarantee for bank's statistical robustness, that is an high confidence level that building the same bank with a different training sample wouldn't end up with different bank's attributes (such as number of patterns and popularity pattern order for pattern with lower popularity than reference *P*). Simulations have proven that a reasonable frequency for pattern P, such that bank's attributes statistical fluctuations are negligible, is in the order of 5.

Section 3.06 Pattern matching

The Euclidean coordinates of stubs coming from modules in the tower are enccoded into ssID in the FPGA and then sent in AM to perform pattern recognition. The matching stage is a parallel process and follows the five out of six pattern's ssIDs matching logic:

1. stubs ssIDs of each layer are sequentially read; layers are processed in parallel;
2. stub ssID in a layer is compared in parallel with each pattern's ssiD of the corresponding layer, looking for matching ssID;
3. in respect of the matching rule, if five out of six ssID of a input subs combination matches with a stored pattern, the pattern is considered as a valid candidate for pattern recognition (Figure 12); matching patterns are said to be "roads fired by the AM"; fired roads contain all the original stubs belonging to the matching patterns' ssIDs; when a road fires, also its siblings can fire too;
4. fired roads are output sequentially from the AM, following the popularity order rule.

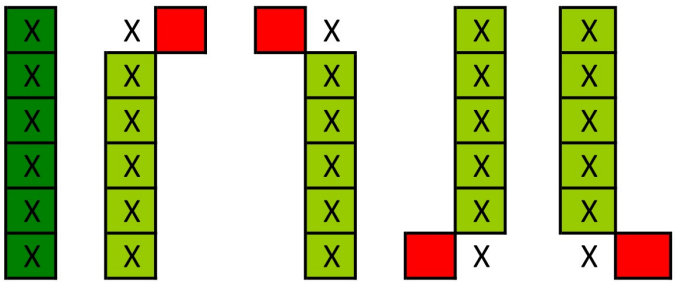


Figure 12 – 5 out of 6 ssIDs matching logic: in dark green is painted a pattern with 6 out of 6 matching ssIDs; in light green are painted patterns that fire with only 5 out of 6 matching ssIDs: they are "siblings" of the dark green pattern.

The fitting stage is taking care of extracting the most likely track candidate by fitting all stubs combinations inside a fired road.

Section 3.07 Fitting stage

Fired roads have only a small fraction of the original full tower stubs, thus a fit on all the possible stubs combinations is extremely quicker than in the original raw stubs cloud.

The output of the fitting stage consists in candidate track parameters associated with each stubs combination that belongs to a road. The full resolution of the tracker has been recovered and track recognition is done.

The time constraints impose a linear approximation fitting method instead of a sophisticated fitting algorithm. Currently, a trained linear fitter is used to perform the fitting stage 53[6]: the linear fitter is completely deterministic, thanks to a training stage - performed using ideal tracks – during which a parameter matrix is build. The linear fitter extracts from each combination of stubs a full set of track parameters, by inverting the parameter matrix built during training stage. In the forward towers, where tracks crosses more than six layers, an optimization study has been done in the past in order to select, for each possible pattern, the six best layers that should be used for fitter training. Then track candidates are selected by a maximum likelihood rule, such as minimum chi squared.

The new forward tower superstrip implementation has been developed on that, in order to decide what layer's ssID should build up the patterns (see section Section 3.11).

Unfortunately, a comparable fitter for the hybrid and forward towers is not available at the time of this study yet, and its development is further beyond the scope of this thesis.

Section 3.08 AM approach optimization

The AM approach solves the pattern recognition time problem by using massive parallel processing. Execution time needed to perform the entire process is partly determined by the AM size: when comparing two banks with equal coverage, the bigger bank with finer patterns will provide the fitting stage with fewer stubs combinations to explore per fired road, thus reducing the total recognition time. While this naive overview of the AM approach optimization problem as a matter of trading AM size with execution time is generally true, in fact optimizing execution time together with road efficiency is a far more complex matter.

The bank size needed in order to reach the desired coverage level depends on the size of the superstrip used in the coarse grained representation of the tracker, in fact using smaller width superstrip more patterns will be needed to get to desired coverage. On the other hand, since the number of pattern fired mostly depends on fake combinatoric track that trigger a physical pattern in the bank, the number of roads sent to the fitting stage together with the total number of combinations to be tested depends on the bank size in a non trivial way.

The optimization study in this thesis focuses in tuning the AM, i.e. the superstrip definition and the banks attributes, in order to get the best efficiency with a reasonable total number of combination to explore in the fitting stage, hence the execution time of the recognition process. This is only one of the two optimization approaches; in fact, other groups involved in the project are currently developing new algorithm to cut the number of possible stubs combination in the fitting stage, after pattern

recognition has been done, e.g using a Hugh transform over stubs combinations inside fired roads.

Section 3.09 Superstrip Δs optimization

In section 3.01 it was stated that the division of tracker layers' geometrical space into superstrips was only one of possible way of defining equivalence classes of stubs. In fact, stubs can be classified not only for their local coordinates along a layer (as has been done until now), but also for their "stub bending" Δs attribute, i.e. the measurement (in units of strip sensor pitch) of the distance between the two charge clusters centroids on the upper and lower sub-layer of a module, that together define the "stub". The Δs is in fact a measure of track's pT, and Δs ∈ [-6.5, 6.5], where Δs ~ 0 is for a high p_T track and |Δs| > 0 is for more bending tracks (sign(Δs) is with respect to their charge); |Δs| = 6.5 is reached for p_T = 2 GeV/c tracks in the outermost barrel layer [14]. A schematic idea of stub bending is showed in Figure 13.

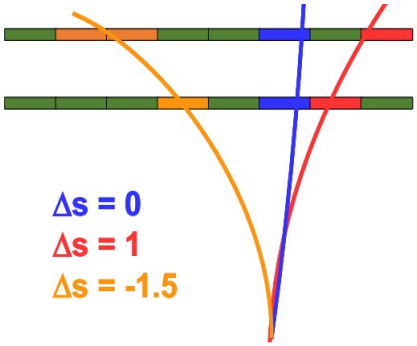


Figure 13 – Stub bending Δs

Current tracker layers' geometrical subdivision can be further segmented using a Δs subdivision (e.g. in 3 classes in the range [-6.5, 6.5]), thus defining the superstrips as equivalence classes of stubs that shares both the same layer local coordinates range and the same Δs range [15]; a patterns becomes a set of 6 ssIDs that contains both a geometrical information and an information of particle's p_T. This superstrip definition has proven during recent studies to be particularly powerful in reducing the combinatoric roads fired when high PU (up to 400) is present: this is due to the fact that stubs coming from minimum bias events, even though they share similar coordinates with the stub coming from track to be reconstructed, belong to a different superstrip because of their different Δs, thus they fire a different – less likely – pattern.

The Δs implementation has proven to be very effective, for barrel towers, up to PU 400. One of the drawbacks lies in the bank size increase necessary to cover the new pattern phase space: in barrel towers a 64K bank is sufficient, even adding Δs division; but for forward towers, where pattern multiplicity is quite low, the bank size increase could be potentially unmanageable. Further studies are mandatory to explore the performance of Δs division in hybrid and forward towers.

Section 3.10 Hugh transform implementation

Currently, the implementation of a generalized Hugh transform is under development for barrel towers; it has proven to reduce the number of combinations before the fitting stage down by a factor 20. Hughes transform implementations is particularly effective when used together with a linearized trained fitter due to the fact that it already makes use of a set of trained confidence interval for track's parameters, characterizing different subset of patterns, in different regions of the tracker. In fact, a discrete coarsed grained μ-space for track parameters (that resembles the one used to describe the idea of "pattern") can be defined: each stubs combination inside a road falls in the appropriate cell in the μ-space, with respect to its candidate track parameters.

Generalizing the original linear Hugh transform [8], each track path in (x, y, z) space is bijectively mapped to a dot in the μ-space of track parameters; since for a single dot in (x, y, z) plane a set of lines passes, a single dot through Hugh transform is mapped to a curve in μ-space (e.g. a line, a sinusoidal curve, depending on the chosen map); experimental stubs are mapped through a Hugh transform a collection of curves in μ-space; if stubs belong to the same track, their respective curves in μ-space intercept in the μ-space point defining the track that stubs belong to.

Given a coarse grained subdivision of the μ-space (e.g. by using the linearized track fitter parameters subdivision for the region of interest, and then further increasing the graining), for each stub an "accumulation matrix" is built, i.e. the collection of all the cells of the grained μ-space reached by applying Hugh transform the stub. Stubs that are mapped to the most populated cell in μ-space are likely to belong to the same track, i.e. the one defined by parameters that fell in that cell.

For barrel only tracks, it is sufficient to consider the transverse plane (r, φ) projection of tracks and only to parameters are involved, i.e. ϱ and φ_0, thus simplifying the transform; on the other hand, when dealing with forward towers, all 4 track parameters are involved, thus making the implementation far more difficult.

Section 3.11 Simulation environment: pattern Bank generation, PU events generation, AM simulation

A Geant4 MonteCarlo based simulation inside the CMS software environment – loaded with the updated tracker geometry[1] – has been used to generate single muon-samples (for banks training), and PU samples (for AM simulation). Muons have been selected to train banks (over other kind of charged particles that could be of interested in being triggered, such as electrons) due to their *nice* nature: they exhibit a very low ionization rate in matter, thus are able to cross all layers in the tracker leaving a full length track; being more massive than electrons, they do not exhibit high cross

[1] CMSSW revision 9_2_6 has been used with a upgrade version of AM simulation software AMsim rev. 8; the tracker layout rev. is the OT463_200_IT4025.

section in bremsstrahlung effect (as electrons do), and are less likely to change their track curvature inside the tracker.[2]

The single muon bank training sample has the attributes of a typical primary event muon distribution: the p_T distribution is flat in $1/p_T$; the η and φ distributions are flat over the range covered by the tower trained; the minimum p_T is set equal to 3 GeV/c (minimal threshold to get a stub). The PU sample has the attributes of a typical bunch crossing output: 200 minimum bias events ("PU200") plus a muon added to the event to test the performances of the reconstruction chain. Currently, events with PU400 and PU800 are investigated, in order to better understand AM approach performance with highly dense stubs cloud in the tracker.

Trigger tower bank's training is performed using event with single muon tracks that belong to tower's dual space. A cleaning stage is mandatory in order to get rid of any anomalous stubs or entire tracks:

- tracks that leave less than 6 stubs in the tracker are discarded;
- events with low p_T secondary charged particles are discarded;
- since it is possible that a track leaves up to 4 stubs in the same layer, due to the fact that modules can overlap due to φ and z coverage needs, only a single stub per layer is kept (the one closer to track's ideal path in that layer).

After the cleaning stage, the pattern generation algorithm selects only 6 layers crossed by track in order to build a pattern.

The 6 layers selection rule implementation was mandatory in order to process tracks that belong to hybrid and forward towers, since they can cross more than exactly 6 layers (as in the barrel). The selection rule has been developed with respect to De Mattia's fitter optimization study [6]: upon all the possible crossed layers combinations, only combinations with exactly 6 layers are considered valid; these has been selected with respect to track fitting resolution (e.g. layer hit in ps module is preferred over layer hit in 2s; layer hit at a bigger distance from vertex is generally preferred in order to achieve better p_T). The valid combinations are ordered in a mask (Table 1), following a popularity rule; result show that the mask must contain at least 15 of the first most valid combinations in order to cover up to 99% of the possible combinations.

[2] In the past, AM banks training with a sample composed by both muon and electron sample (to provide better reconstruction efficiency also for particles that exhibit high bremsstrahlung effect) has been investigted. Such banks were very big in size, though, and currently training is performed with muons only.

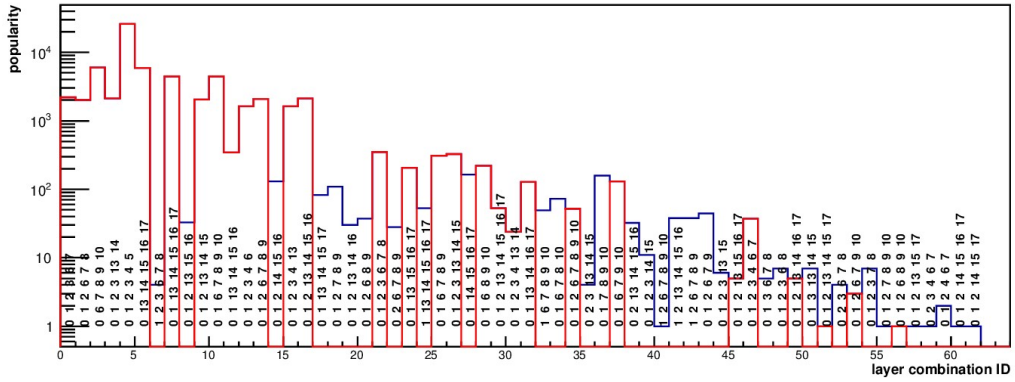


Figure 14 – Popularity distribution of most common layers combinations of track over the whole detector. With the combinations marked in red a global coverage of over 99% of all the possible combinations has been achieved. In red are selcted patterns with 6 or more layers crossed: if more than 6 layers are crossed, De Mattia's selection rule has been applied to select only 6 of them. Please note that also pattern with very low popularity are marked in red: in fact, those are triggered as siblings of the valid patterns with respect to the selection rule adopted.

In order to develop the popularity rule, a study over all the possible layer combinations has been performed, using a sample of single muons over the whole tracker: as expected, 14 of the most popular layers combinations were those already noted by De Mattia in his study, but one more has been added; the popularity histogram showed in Figure 14, marks in red layers combination selected as valid with respect to the mask adopted.

barrel layers						forward disks					backward disks				
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0
1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0
1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0
1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0
1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1
1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1
1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 1 – Layer selection mask of exact 6 layers patterns. To reach a coverage of 99% of all possible layers combinations, a mask with 15 valid combination is needed. A layers belonging to a valid combination is marked with "1"; "0" otherwise. Valid combinations are parsed in order of popularity.

In order to assign the pattern, track crossed layers are compared to the ordered valid layers combinations stored in the mask: the most popular valid combination matching, if found, is then assigned to the training track and stored in the bank.

When pattern generation has been completed, patterns in banks are ordered with respect to their multiplicity.

Chapter IV Superstrip design

The necessary condition that must be satisfied when defining superstrips is that they must cover without superposition all layers in the tracker. If that condition is fulfilled, the design itself of the superstrip can be investigated, by varying superstrip parameters e.g. shape, dimension, layer-dependant parameters, encoding properties. The AM bank size needed to satisfy coverage requirement ultimately depends on superstrip definition. Superstrip optimization is performed by bank studies and road efficiency studies. Until now, just barrel-only definitions have been investigated, in order to perform efficiency studies with track entirely confined in the barrel. The main goal of this thesis is to introduce and characterize new superstrip definitions for the hybrid and the forward towers.

The most efficient solution in the barrel consists in a *fountain-like* definition, while for the forward towers, a new original design *flower-like* is investigated, and compared with the previous solution. Details about their design will be shoed in next section.

A reasonable superstrip design should reflect tracker cylindrical symmetry and layout:

1. fixed $\Delta\varphi$ width in the transverse plane; the *scaling factor* parameter fine tune the superstrip width, with respect to an hardcoded value that has been assigned to each layer; the number of φ - subdivision in the transverse plane is determined by the superstrip's $\Delta\varphi$ width;
2. in barrel layers, the number *nz* of subdivisions along the beam axis must be decided;
3. in disks layers, the number *nr* of subdivision along the r direction must be decided.

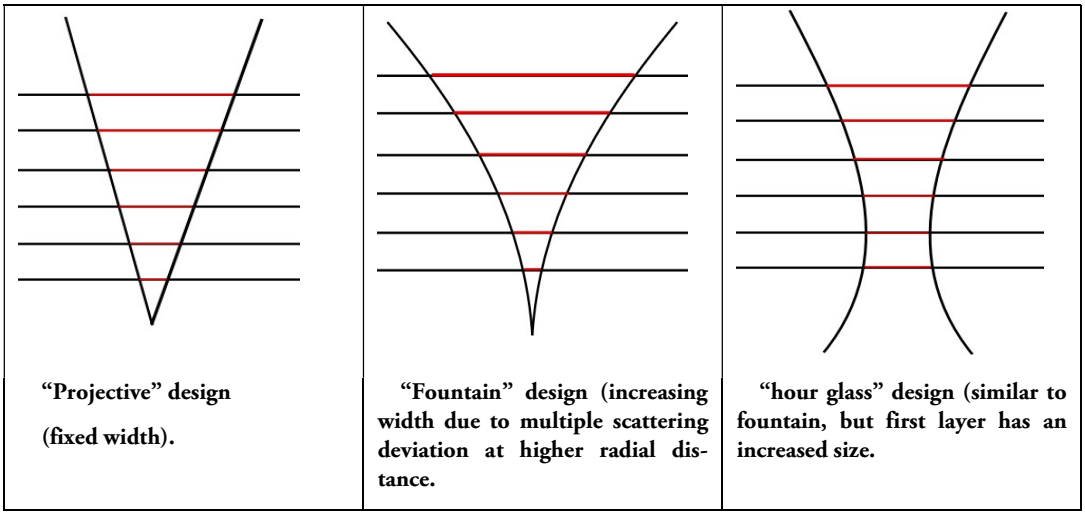
While having an optimal amount of φ subdivision for each layer is mandatory, due to the curved path projection in the transverse plane of a charged particle helicoidal motion in presence of a magnetic field, *nr* and *nz* subdivisions are proven to be futile when dealing with low size AM banks, since *nr* and *nz* >= *1* cause a drastic increase in bank size. In this thesis' studies, *nr* and *nz* are both set to 1.

Section 4.01 Superstrip $\Delta\varphi$ width optimization

Previous studies of tracing particle angular deviation from the ideal path, due to multiple scattering across modules, confirmed that the most efficient superstrip φ width is layer dependent: barrel layers at increasing distance *r* from beam axis have superstrips with increasing φ width, thus are likely to avoid losing tracking particles due to multiple scattering. Scaling factor *sf = 1* is defined with respect to the φ range that covers the 90% quantile of the angular deviation distribution of φ stub coordinate from ideal track φ (due to multiple scattering) calculated using equation (4.1).

In figure are shown three kinds of superstrip design studied until now. Fountain-like design has proven to be the best superstrip design for barrel layers, superseding simpler definitions such as "projective" superstrip with constant φ width in each layer. In fact, Further tuning have proven that a hourglass – like design, with φ width slightly bigger in the first barrel layer, have proven to be even more effective.

Figure 15 – Most studied superstrip design until now.



"Projective" design (fixed width).	"Fountain" design (increasing width due to multiple scattering deviation at higher radial distance.	"hour glass" design (similar to fountain, but first layer has an increased size.

$\Delta\varphi_{SS}$ [rad]	I layer	II layer	III layer	IV layer	V layer	VI layer
$\Delta\varphi_{SS \text{ in barrel}}$	0.00762	0.00439	0.00459	0.00485	0.00523	0.00575
$\Delta\varphi_{SS \text{ in disks}}$	0.0048	0.005	0.0058	0.0064	0.007	

Table 2 – Summary table of superstrips widths – for both flower and fountain – at scaling factor 1. For barrel layer, *sf*=1 is set with respect to the interval that covers 90% of φ angular deviation in barrel; on the other hand, for disks layers, *sf*=1 is set with respect to the interval that covers 95% of φ angular deviation in disks. The unsually big first layer width is due to the fact that it just works better.

In order to study disks implementation, a multiple scattering study in forward towers has been done. In Figure 16 the angular deviation with respect to the ideal track is shown for disks layers; it is noticeable the discontinuity between the angular dispersion in *ps* modules (in blue in the picture) and *2s* modules (in red), due to the fact that low radial resolution of strips in *2s* disk modules propagates to low φ resolution of stubs. When setting superstrip φ width with scaling factor 1 in disks, the *ps* modules φ distribution has been taken as reference, due to their better resolution, and to the fact that φ dispersion for *ps* modules in disks is quite similar to the same quantity for the first 3 layers of the barrel.

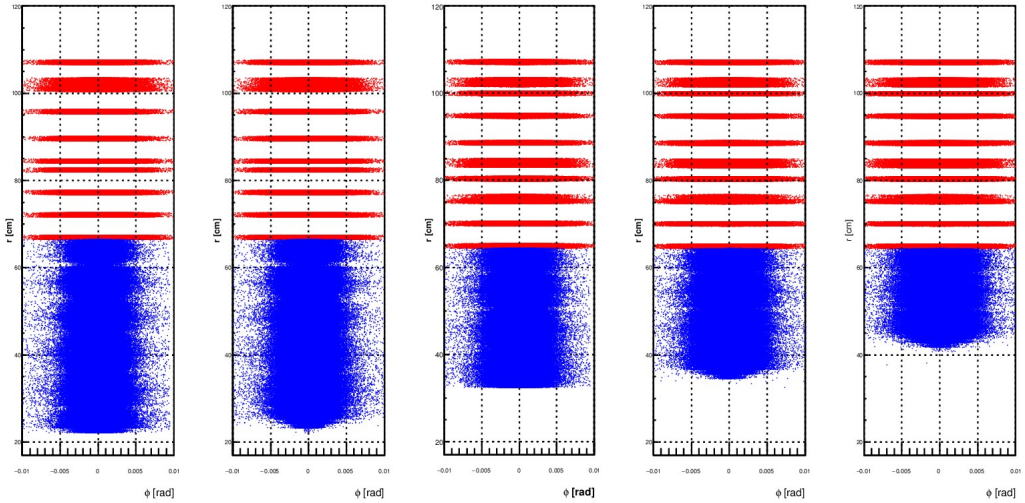


Figure 16 – Angular deviation distribution of φ stub coordinate with respect to ideal φ coordinate at same radius calculated using helicoidal path equation in transverse plane. 5 disks layers are shown.

Section 4.02 The tilted modules' and disks problem

When investigating η turn-on curves in barrel tower with fountain superstrip banks (in fFigure 17, such a plot for tower 41 is showed), an efficiency loss in the η range that covers the tower area with tilted modules presence (for tower 41 it is actually the whole range shown in the picture) has been noticed. This feature is particularly emphasized in the η turn-on curves for forward tower, where full extended disks are present. A pronounced bank size increase in hybrid and forward tower has also been noticed.

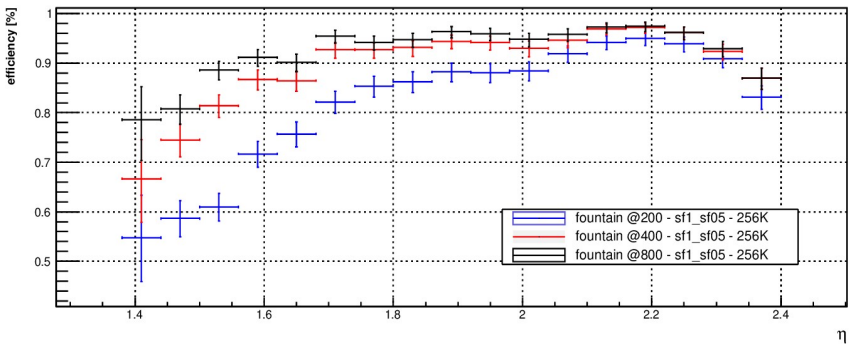


Figure 17 – η turn on curve fot tower 41. Details of this plot will presented in chapter VI, when dealing with the actual road efficiency study. For what matters in this section, please notice that different colors reflect different ways AM ouput fired roads. On ordinate axis, the efficiency in reconstructing particles is shown.

Such features can be explained by the loss of the decoupling effect over z_0 and η, mentioned in the previous section, in tower regions where modules extending along radial direction are present; the loss of decoupling effect reflects to a decrease in pat-

tern multiplicity over such regions. If average pattern multiplicity in tower's banks is low then:

1. more patterns are required to build a bank with a desired minimum coverage, and a bigger training sample is usually needed; if bank size is limited by size constraints, is more likely to cap the maximum efficiency due to coverage limitation;
2. patterns have reduced popularity, thus when pattern matching is performed in the AM, is less likely that these pattern will fire and a reduced efficiency is to be expected in those patterns' track phase space, as observed in η turn on curves.

One of the reason why the average pattern multiplicity is lower in regions where tilted modules are present is to be found in the helicoidal path of particle tracks inside the tracker, in particular, in the dependence of z-coordinate on track parameters. The two parametric equations that describe the motion of a charged particle in the presence of a magnetic field, in cylindrical coordinates, are:

$$r = 2\varrho \sin(\varphi_0 - \varphi) \tag{4.1}$$

$$\varphi_0 - \varphi = \frac{\tan(\theta)(z - z_0)}{\varrho} = \frac{\sinh^{-1}(\eta)(z - z_0)}{\varrho} \tag{4.2}$$

Where φ_0 and η are the particle track angle in the transverse plane and the particle η respectively, measured close to the interaction vertex; z_0 is the z coordinate of the vertex; $\varrho = 0.3 \cdot B \cdot p_T$ is the track curvature determined by its transverse momentum and the tracker magnetic field $B \sim 3.8$ T.

The former equation describes the track path projection in the transverse plane, using polar coordinates: (r, φ) depends on track curvature ϱ and φ_0, while it is independent on the other two track parameters, vertex coordinate along beam axis z_0 and η. The latter describes track path projection in the (φ, z) plane: here all the four track parameters are coupled to define the functional dependence φ(z).

In order to fully determine the stub coordinates (r, φ, *z*) of a track passing a specific layer, both the equation are needed, since they are coupled with respect to φ: r = r(φ; ϱ, φ_0), with $\varphi = \varphi(z; \varphi_0, \varrho, \eta, z_0)$. In the general case of *n* φ-subdivisions, *nr* and *nz* bigger than 1, assigning the ssID to a stub in a layer requires all three stub coordinates and are thus determined by the 4 track parameters coupled together; but when dealing with *nr* and *nz* both equal to 1 the problem simplifies. In fact, given the tracker geometry, ssIDs can be uniquely determined by only two coordinates, i.e. (r, φ) for barrel layers and (φ, z) for disks layers:

- in the barrel, since each layer's module has fixed *r* coordinate (in the flat revision of the layout, at least), the φ stub coordinate on a layer is uniquely determined just by track's (ϱ, φ_0), using the first equation;

- on the other hand, in the disks each layer has fixed *z* coordinate, thus the φ stub coordinate on a layer is uniquely determined by the 4 track's parameter altogether, and the second equation is needed.

In the barrel, ssIDs – and ultimately patterns – are determined by track's ϱ and φ_0 only: varying track's η and z_0 does not affect pattern position in the μ-space, thus significantly increasing average pattern multiplicity.

On the other hand in the disks, since ssIDs are determined by all the 4 track parameters altogher, varying track's η and z_0 does affect pattern position in the μ-space indeed, hence significantly decreasing average pattern multiplicity. Since vertex z coordinate has a standard deviation ~ 15 cm, small compared to disks z-coordinates ≥ 120 cm, in the next sections only η changes are considered interesting for disks.

What actually happens in real space onto the fixed-z disks modules is that by varying η (*fixing all other parameters*), the track is changing the φ-coordinate of the (φ, z) layer crossing point, i.e. is crossing different superstrips along the radial disk projection. The number of crossed superstrip whose width is $\Delta\varphi_{ss}$ in a layer with radial extension r, by tracks with fixed ϱ and φ_0 is given by

$$n_{SS} = \frac{\Delta\varphi_{track}(r_\eta)}{\Delta\varphi_{SS}(r)} \tag{4.3}$$

where $\Delta\varphi_{track}(r) = \arcsin\left(r/2\varrho\right)$, from equation (4.1) is the track angular deviation from straight radial path at radius r, and $\Delta\varphi_{ss}(r)$ is the superstrip boundary's φ coordinate deviation from straight radial path at r; it is important to remember that *r* at which track crosses the layer is coupled to track's η, as stated in equation (4.2).

Ultimately, pattern multiplicity, due to η coupling, reflects the track's η range that is able to maintain the $\Delta\varphi_{track}(r)$ smaller than the $\Delta\varphi_{SS}(r)$ ("η confidence range"): a bigger η confidence range reflects to an increased pattern multiplicity, hence to a smaller bank.

The effect of increasing *nz* in the barrel layers is nothing but to introduce an increasing dependence on track's η and z_0 of pattern position in μ-space; idem for *nr* in the disks.

Flat modules in the barrel are effective in reducing pattern dependence on track's η, due to the fact that they have fixed r-coordinate, and φ stub coordinate do not depends on track's η. On the other hand, module tilting is responsible for an increasing dependence on track's η, similarly to what happens in disks layers: in fact, each tilted module has a not negligible projection ~ 7 cm in the r-direction, hence tracks with varying η have a chance to cross up to 3 superstrips @ p_T = 3 GeV/c.

Moreover, in barrel layers φ coupling to η, as in equation (4.2), depends on z-coordinate, since z in the barrel is not fixed: when fixing track's φ_0 and ϱ, the η confidence range Δη range reduces to:

(4.4) $$\varphi \propto \frac{(z-z_0)}{\sinh(\eta)} \Rightarrow \Delta\eta_{r_2,r_1} \propto \operatorname{arcsinh}\left(\frac{z_{r_1}-z_0}{\varphi_{r_1}}\right) - \operatorname{arcsinh}\left(\frac{z_{r_2}-z_0}{\varphi_{r_2}}\right)$$

In barrel layers contained in hybrid tower, $\eta \in [0.8,\ 1.5]$, $z \in [0.5\text{m}, 1.2\text{m}]$ and $\Delta\varphi$ ~ 1.5 $\Delta\varphi_{ss}$ @ layer 3; under these conditions, the η confidence range that covers the modules' $\Delta\varphi$ increases with z up to a factor 2 between z = 60 cm and z = 100 cm, thus effectively increasing pattern multiplicity, along z. The increased multiplicity due to an increased η confidence range along z–axis should be noticeable in η turn on curves.

The new flower like superstrip design tries to increase pattern multiplicity in tilted modules, by increasing the tracks η confidence range, when fixing φ_0, ϱ and z_0 track parameters.

Section 4.03 Flower superstrip

The new flower design for superstrip aims to reduce the loss of multiplicity of patterns that cover regions with tilted modules by using a curved shape in the transverse plane, following the (r, φ) parameterization of a varying η track. The superstrip shape over disks layer is sketched in next figure.

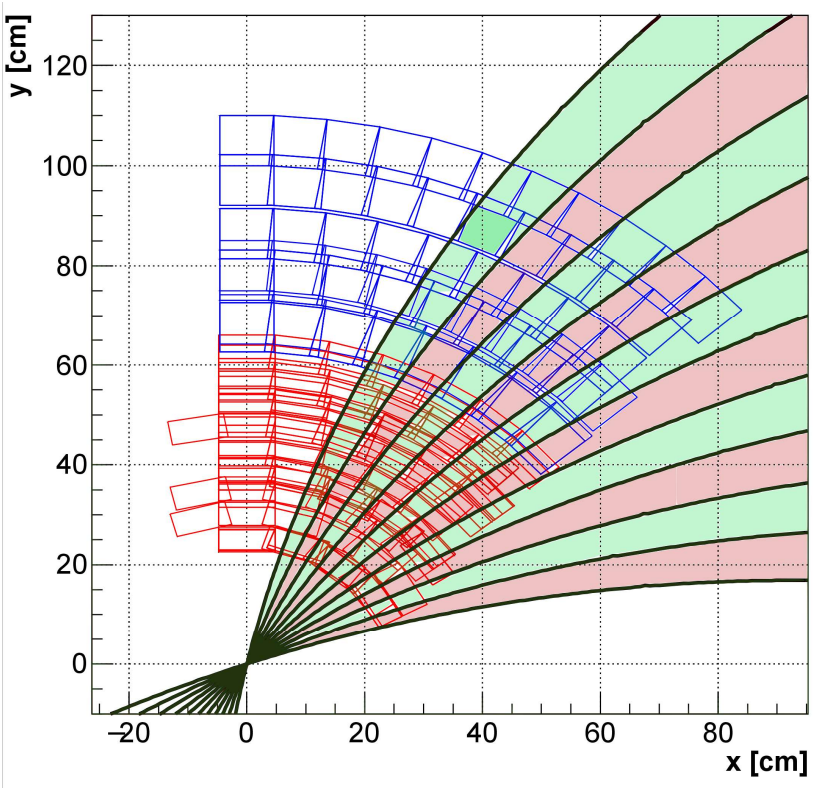


Figure 18 – Flower superstrip over tower 41 disks projection onto transverse plane. Such superstrips have actually their unique angular width for each disk layer. In figure, width has been stressed as well as curvature (here is with p_T = GeV/c, while the actual implementation uses p_T from 8 to 12 GeV/c.

Tuning the φ-boundary path of superstrips such that $\Delta\varphi_{SS}$ is closer to $\Delta\varphi_{track}$ allows to reduce the number of crossed superstrips when varying track η (equation (4.3)), thus effectively increasing the track η confidence range, i.e. increasing average pattern multiplicity.

The (r, φ) parametric equation used to define superstrip boundaries is the (4.2): the superstrip definition sets the p_T (hence the track's curvature) and the charge upon which the bank is tuned. Flower design should be more effective than fountain like with respect to the radial extension of the layer, due to the increased η confidence range.

Since superstrips must cover the entire layer, the first one, i.e. the layer's reference superstrip, has to be defined with respect to the charge-oriented path direction. For positive bank, the reference superstrip's lower boundary path in the transverse plane is defined as the path that has the φ_0 such that it passes through the most lower layer vertex of the tower (r_{min}, φ_{min}); similarly, the last superstrip is defined using the path that passes trough the upper tower vertex (r_{max}, φ_{max}). The information about these two vertexes is extracted from tower's boundaries map.

The ssID assignment then starts from the layer's reference superstrip, counting the superstrip number with respect to its fixed φ-width. The disks layers subdivision with flower superstrips resembles the distribution of petals around a flower, hence the name.

The original flower superstrip was meant to be used in disks layers, but has also been implemented in the barrel layers, where tilted modules are present. It is expected that the barrel implementation over tilted barrel modules should be as effective as in the disks when compared to fountain like design, in particular for tracks with p_T higher than bank tuning p_T. In fact, in the barrel the r-projection of each layer (due to tilted modules) is indeed small in comparison to the full r-extension of a disk layer, thus geometrically reducing n_{SS} (with respect to fixed track parameters) and the effectiveness of flower design; on the other hand – given the short r-extension of barrel layers – tracks with p_T higher than the bank's tuning p_T are less likely to cross different superstrip in barrel layers than in the disks ones, thus increasing the effectiveness with respect to track p_T change (this should be evident in p_T turn on curves at high values).

It is clear from the flower definition that two banks must exist in order to characterize a tower: the former tuned for positive charged particles, the latter tuned for negative ones. The charge-specific bank design implies a mandatory partial duplication of the hardware AM recognition chain: the FPGA must transcode two different ssIDs for each stub – one for the positive bank, one for the other; the number of road fired is doubled, since two AM banks are now present. The added hardware implementation complexity is indeed a drawback: the efficiency studies for forward and hybrid towers aims to balance the complexity drawback with increased performance,

in terms of both efficiency and fitting time (i.e. number of stub combinations for each road).

The flower bank optimization now must involve also p_T tuning of superstrips: in order to maximize the decoupling effect of the curved superstrip on patterns' η dependence.

Chapter V Flower superstrips implementation in hybrid and forward tower

When investigating the AM approach performance for hybrid and forward tower, fountain design is compared with flower design for superstrips. In order to present results in a more readable way, flower bank optimization study for both tower 33 and 41 are presented first; global road efficiency results will be showed in chapter VI.

Section 5.01 Flower p_T study

For each scaling factor configuration, flower banks have been tuned in p_T in order to minimize pattern number at 90 % coverage. It is interesting to investigate the frequency distribution of charge over average pattern p_T , i.e. pattern curvature, for a given bank configuration and compare it with the analogous distribution for fountain-like design. Being the sample distribution flat in $1/p_T$, frequency distribution reflects pattern multiplicity in bank, high counts correspond to low multiplicity patterns. In Figure 19 an example of such distribution for tower 41 is showed; tower 33 exhibits similar results.

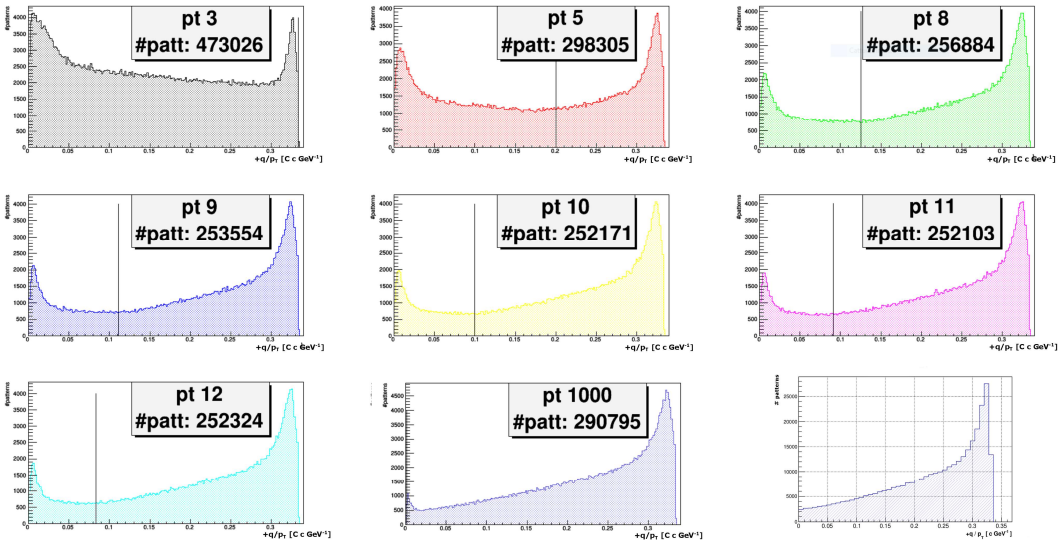


Figure 19 – TT41 bank total size with respect to different pT at scaling factor *sf*=1 in barrel, and *sf* = 1 in disks. A vertical line is placed at the distribution's minimum. The bottom right one is the analogous distribution for fountain like bank, with same scaling factor (only positive charge is showed).

Banks have been generated with a flower bending corresponding to a p_T in the range [3, 15] GeV/c, along with a very high p_T = 1000 GeV/c setting. Curvature distributions show two peaks at the upper and lower limit of 0.3 and 0 $(GeV/c)^{-1}$ respectively, in accordance with a reduced multiplicity for patterns with extremely different curvature with respect to bank's curvature tuning. As expected from equation's (4.2), tracks with bigger or smaller curvatures than superstrip tuning curvature are going to cross different superstrips on same layer when varying η, thus decreasing patterns' multiplicity. A symmetric curvature frequency distribution around 1/5 $(GeV/c)^{-1}$ is obtained when bank p_T tuning is set at p_T = 5 GeV/c: this is not surprising, since the corresponding superstrip curvature is approximately the arithmetic average between the two curvature limits of 0 and 0.33 GeV/c. Lastly, since at very high p_T the flower superstrip shouldn't differ from a fountain-like shape, it is expected that the bank average p_T distribution doesn't differ from the analogous fountain bank plot: this is confirmed in the last plot of Figure 19.

Interestingly, the optimal bank size curvature tuning is at p_T ~ 8 GeV/c, that corresponds to a slightly less curvature that the one defining the symmetric frequency distribution; the optimal p_T value is in general is a unique attribute of each scaling factor configuration and in Figure 20 is showed flower bank size with respect to scaling factor and flower bending p_T.

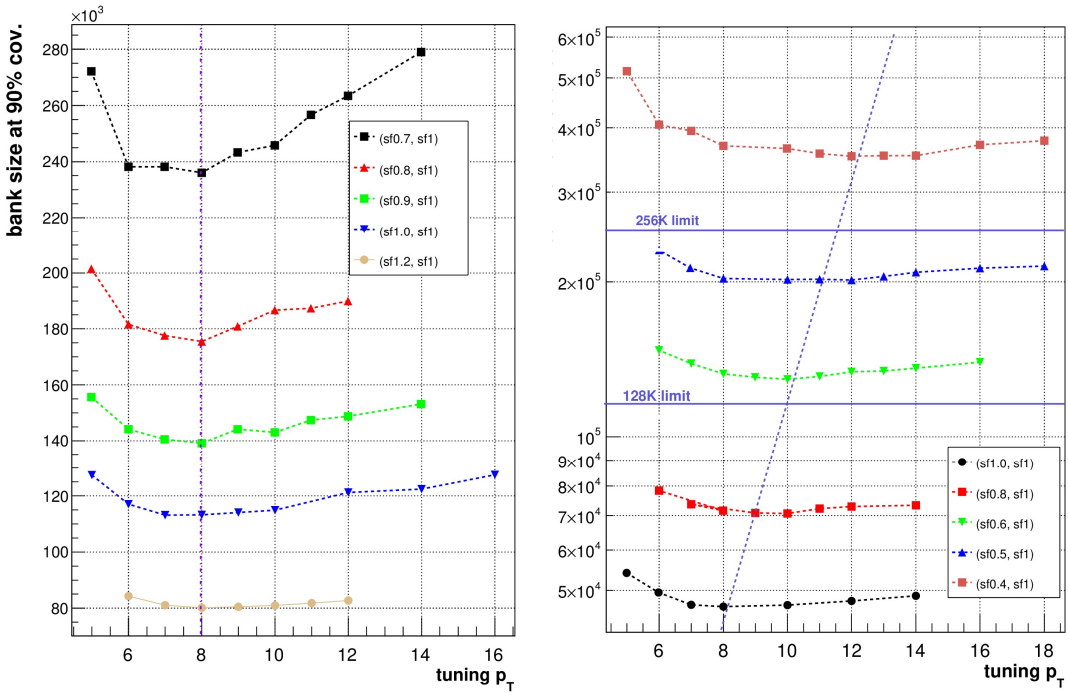


Figure 20 – p_T optimization for tower 33 (on the left) and tower 41 (on the right). In particular, for tower 41 are clearly indicated usual 256K size limit, and 128K size limit .- that will be used as default for flowers configurations. For tower 33, barrel scaling factor (the first sf number between parenthesis in the legend) has been variated; for tower 41, disk scaling factor has been changed (the right number between parenthesis in legend).

As shown in last figure, for tower 33 optimal p_T is independent from barrel scaling factor (check in caption for further details); on the other hand, tower 41 optimal p_T shows a slight dependence over disk scaling factor, even though the minimum is quite large, in the range [8, 12] GeV/c.

Section 5.02 Bank size comparison between fountain and flower

The ratio between fountain-like bank size and flower-like bank size is of great importance when estimating the effectiveness of flower design; in fFigure 21, bank sizes at 90% reference coverage and different scaling factor for hybrid and forward towers, are showed. Due to modules distribution in towers, when dealing with forward tower, the barrel scaling factor have been fixed at $sf = 1$ and disk scaling factor has been set as variable; the viceversa has been done when dealing with hybrid.

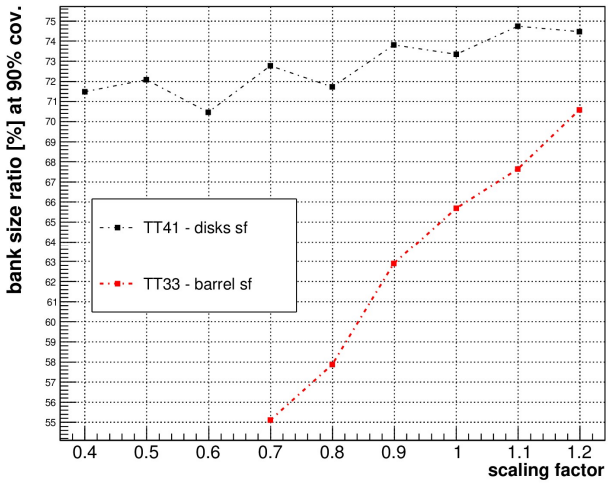


Figure 21 – Bank size ratio at different scaling factor (barrel scaling factor for hybrid tower 33, disk scaling factor for forward tower 41), at 90 % coverage.

Although flower design was originally meant for disks, it is in hybrid tower, when tuning barrel scaling factor, that flower shows its effectiveness: when reducing scaling factor the size ratio with fountain bank sets down to 50% at $sf = 0.7$, while increasing linearly up 70% at $sf = 1.2$. A similar linear dependence, although not as much steep at all, is noticeable when tuning disks scaling factor for forward tower.

The bank size optimization achieved with flower superstrips, brings quite impressive results for hybrid tower, also in terms of effective coverage gain when using fixed bank size of 256K (that is 128K positive charge + 128K negative charge for flower). In Figure 22 bank coverage is showed for both configuration. At 256K the difference in global coverage of the two designs is in order of 10% for hybrid tower (at barrel scaling factor $sf = 0.7$), from 70% fountain to 80% flower; and 5% for forward tower (at disk scaling factor of $sf = 0.5$), from 80% to 85%. The reason of this is that flower

design is very effective on very low multiplicity patterns, i.e. less popular ones, that in global coverage plot contribute in gaining coverage over 70%, where the curve is starting to saturate. If considering more popular patterns to be stored inside a 256K limited bank, the effect of flower is much more limited.

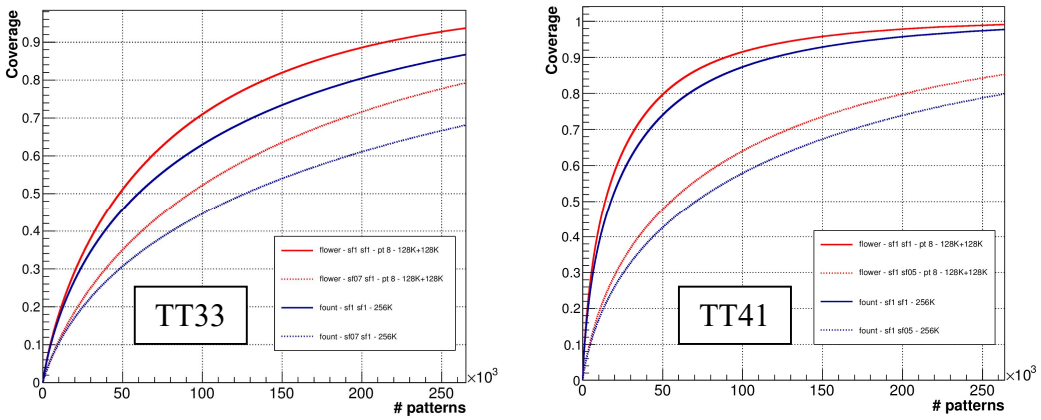


Figure 22 – Coverage (%) for hybrid tower (on the left) and forward tower (on the right). Flower banks results is shown in red, while fountain ones in blue; two different configurations are showed (scaling factor bla blab labl bal blab lab lbal bal).

Section 5.03 Superstrip occupancy comparison

When characterizing a new superstrip design, its mandatory to check the average stub occupancy of the superstrips to ensure that no hot spots will be present.

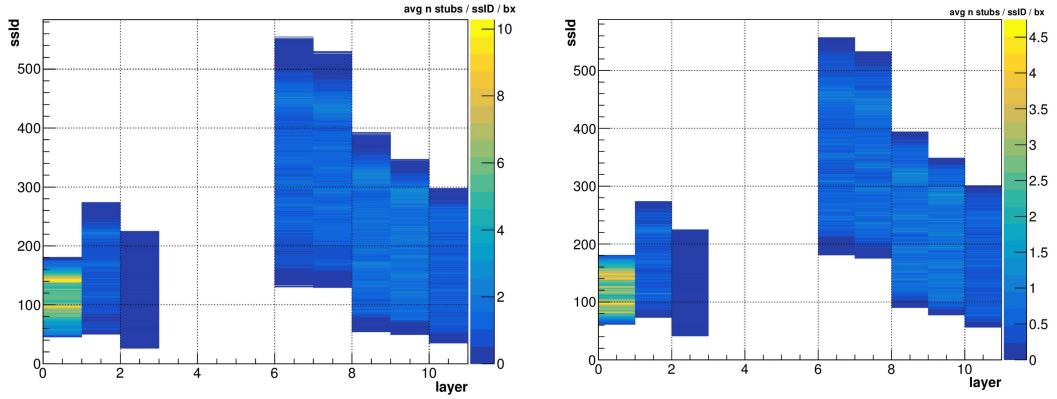


Figure 23 – Superstrip (identified by the uniwue ssID) stubs occupancy, for fountain (left side) and flower (right side), with same disk scaling factor sf = 0.5.

Figure 23 shows average ssIDs stubs occupancies in tower 41 for flower and fountain configurations, using standard scaling factor $sf = 1$ for both barrel and disk; only tower 41 is showed here, since it covers a large part of disks section and also reasonable amount of titled modules on the first three layers of the barrel. Flower configuration needs up to 10% more superstrips in order to cover the full φ extension of

towers, and boundary superstrip are under-used (especially for fountains; flower superstrip, even if close to tower φ boundary, yet they curve towards the center of the layer where a higher stub density is present); this was expected, since boundary superstrips don't cover a big part of the tower, they are less likely to belong to a popular pattern. Stub occupancy is isotropic in φ and with no significant difference between disk layers, for both flower and fountain implementation: on disks, an average number of 4 stubs per superstrip is recorded, a reasonable number at PU200.

Chapter VI Road efficiency studies

The actual performance of the AM approach is ultimately determined by the reconstruction efficiency after the fitting stage. Upon the possible definitions of reconstruction efficiency, e.g. the probability that the fitter output at least one stub combination with track parameters belonging to a fixed confidence range, the most "fitter independent" one refers to "road efficiency": the probability that, between the fired roads, at *least one that contains at least 5 stubs* (in as many superstrips) *belonging to the tracking particle* is found; the reference tracking particle must belong to trigger tower dual space, otherwise it is not considered as a valid particle upon compute efficiency.

When dealing with a simulated environment, all the information about the tracking particles is known, thus making the efficiency estimation a well determined problem. Results are presented as average estimates over sample with 1 muon embedded in 200 PU; road efficiencies confidence intervals are estimated with a C.L. of 68% using Clopper-Pearson "exact" method[3].

The road efficiency definition adopted in this thesis excuses us from further characterization of the fitting stage, that is not available for tilted geometry.

Limits on the number of fired roads output to the fitting stage are imposed, due to the limitation on total fitting time for each event, as well as to the hardware bus transmission rate limitation when outputting roads from the AM chip. Currently roads truncation limits are set to 200, 400 (certainly possible with state of the art hardware implementation) and 800 (extreme configuration).

Along with the road efficiency and the total number of combinations that are sent to the fitter, two more quantities that characterize the road efficiency are:

[3] Efficiency statistical error is asymmetric, but since lower and upper confidence interval limit are similar, only one of them is shown as a symmetric confidence interval.

- number of *duplicates*, i.e. the total number of stub combinations (minus 1) that have at least 5 stubs in as many superstrips coming from the tracking particle;
- number of *fakes*, i.e. the total number of stub combinations that do not have at least 5 stub in as much superstrips coming from the tracking particle.

Road efficiency can be displayed with respect to tracking particle parameters, such as η, p_T, and φ_0, with turn-on curve plots:

in particular, η turn-on curves are very useful to investigate how much the superstrip design is effective in reducing pattern multiplicity over tilted modules and disks; on the other hand, p_T turn on curves are provide information about track trigger performance in selecting only tracks over a p_T threshold. Given the p_T threshold of tracker modules of 3 GeV/c, the ideal efficiency profile in p_T of the complete AM track trigger chain should be a Heavside function $\theta(p_T - 3)$; in fact, in real case it is expected that reconstruction efficiency under threshold is as low as possible, threshold so that, when convoluting with event particles p_T spectrum, no tracks under threshold are passed to trigger alghorithms; on the other hand, the turn on profile is as sharp as possible just over the threshold.

Since for track under threshold it can be both due both to the modules inefficiency in recording a stub of a particle under 3 GeV/c, and both to the inefficiency of the AM in recognizing under threshold tracks, in following results two kind of p_T turn on curves will be discussed: the full turn on curve for particle that belongs to a slightly wider tower dual space, i.e. with $p_T > 1$ GeV/c that still satisfy tower dual space condition about φ (R*) and η (R*); a p_T turn on curve for particles with $p_T > 3$ GeV/c (usual PU sample), but computed over roads fired using a AM bank trained with a particle sample with p_T over 5 GeV/c. The second kind of p_T turn on curve will provide useful information on specific AM inefficiency in reconstructing particles with p_T lower than the AM bank training sample particles.

Since the flower design makes use of two distinct pattern banks – one for positive charge and one for negative, the complete road output for each event is doubled if compared to standard fountain bank. In order to fairly compare results between the two design, performance quantities related to fountain-like banks are computed with a road truncation limit doubled the one used for flower-like; moreover, total number of combinations and road fired, is presented as the sum of matching charge bank and not matching charge bank.

It should be noted that the total number of road fired from flower banks (positive and negative altogether) is comparable with the total number of roads fired from fountain banks, as will be showed in detail in section 02 and 03; this is expected, since most of the road fired are combinatoric. The small difference in number of

roads fired between matching-charge and not matching-charge flower bank is indeed due to the average number of siblings of the single correct pattern, fired from the matching charge bank only. Since in average the 85% of roads fired are shot with the logic 5 out of 6 (and not 6 out of 6) it is to be expected that average number of duplicates combinations is a good estimation of the average number of siblings of a pattern in the bank.

For each event is mandatory to process the output of both flower banks, even though the don't matching-charge one output up to 99% of fake combinations, independently from scaling factor.

Efficiency results for flower configurations are therefore presented in terms of matching bank performance only, with only a reference to not matching charge bank, for sake of completeness.

The road efficiency optimization consists in investigating the optimal AM bank configuration that manages to produce the best reconstruction efficiency with respect to total combinations. AM bank tuning includes: setting the scaling factor in barrel and disks; setting the bank size and, for flower-like, tuning the optimal p_T.

Section 6.01 Barrel Tower

Tower 25 has been investigated using fountain-like superstrip with AM bank size of 64K and 256K; barrel tower has been the first attempt in exporting the old configuration over the new tilted geometry and since in barrel towers modules don't extend along r-direction, except for the first ring of *ps* modules at inner layers boundaries, flower design was not implemented for this tower. Future analysis could focus in studying flower design implementation also in tower 25, as a way to further improve current results.

Barrel scaling factor range [0.5, 1.2] haa been investigated and the best configurations with the tilted geometry have been compared with old studies results on flat geometry layout.

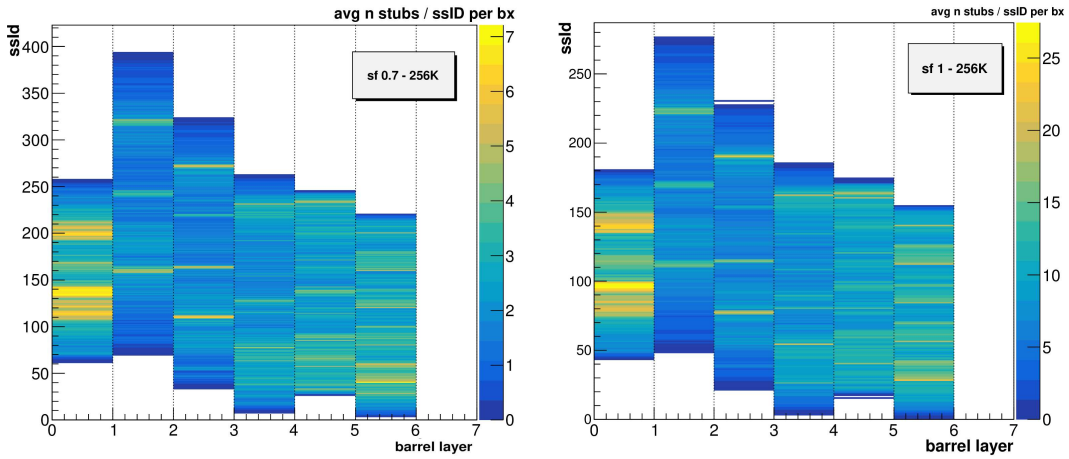


Figure 24 – Occupancy studies for TT25 using a 256K bank: on the left with barrel scaling factor sf = 0.7; on the right with sf = 1.

As showed for the hybrid and forward tower, in Figure 24 are showed superstrips stubs occupancy for barrel tower, using the reference barrel scaling factor of sf=0.7 (best case for 256K) and sf = 1 (reference scaling factor), for the 256K bank configuration. The well established barrel layers superstrip φ widths produce occupancy plots not as much homogenous in different layers as the newly defined disks superstrips widths do: in fact, for the best configuration at 256K bank size with barrel scaling factor sf = 0.5, the number of stubs per superstrip keeps increasing on outer tracker layers, from an average of 2 per bunch crossing to 4 on layer 5; this indicates that superstrip widths for outer layers are slightly to wide with respect to local stub density. But since in first three layers stub occupancy is quite homogeneous and given that these settings have proven to work quite well no modification in barrel superstrip width has been made.

A feature in barrel layers is that some well defined groups of superstrips are particularly populated. In fact this could be explained by looking at tower layout projection in (r, z) plane. Given the regular φ distribution of such highly populated superstrips, the reason of such density is to be found in overlapping modules in layer, whose particular disposition is mandatory to have best coverage in φ and z direction; in overlapping portions of modules the same track can leave up to 4 stubs, due to the fact that it is crossing many modules, thus incrementing local average stub density. Moreover, due φ and η tower dual space cut together with module disposition, it is possible that same modules are only partially populated, thus when generating patterns it is less likely to generate patterns in φ such regions and to populate them when firing roads.

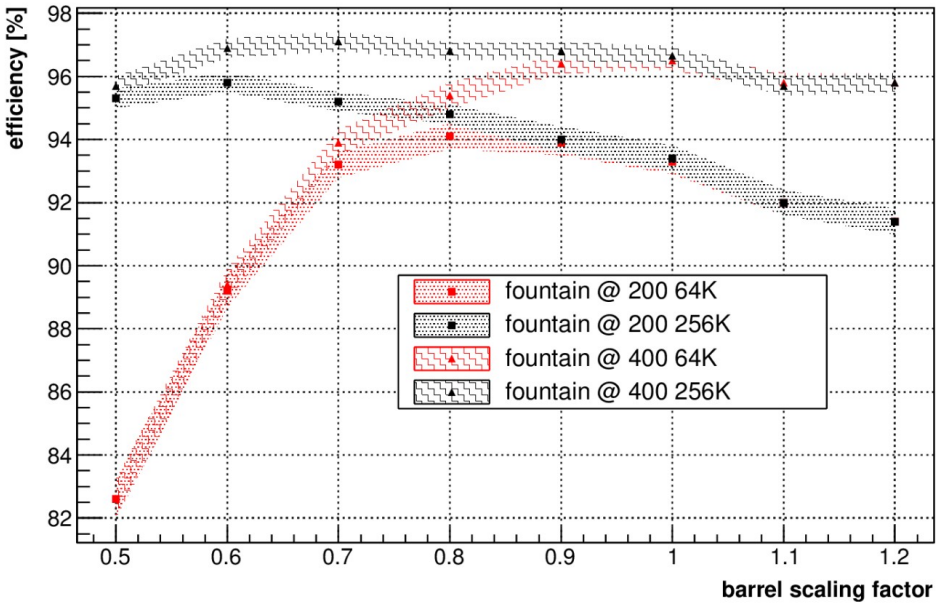


Figure 25 – Comparison between 64K and 256K; road truncation at 200 and 400 road are showed (red for 64K, black for 256K).

FFigure 25 shows the road efficiency global comparison for the examined configurations. Efficiencies depends on bank size, scaling factor and road truncation:

1. with a 64K bank (red in Figure 25), with 400 roads truncation limit the best efficiency of 96.5±0.3% is achieved at scaling factor 1 in the barrel, while with 200 roads truncation limit a the maximum is set at 93.2±0.3% at scaling factor 8 in the barrel;
2. with a 256K bank (black in the figure), with 400 roads truncation limit the maximum efficiency is rised to 97.1±0.3% at scaling factor 0.7 in the barrel, while with 200 roads truncation best efficiency is 95.8±0.3% at scaling factor 0.6.

The efficiency plot reflects the effects of both varying scaling factor, having different bank size limits, and cut the roads output from the AM.

With respect to scaling factor, advantages of a bigger bank are visible: when reducing superstrip width, efficiency loss due to coverage limitation are avoided; on the other hand, at scaling factor smaller than 0.8 efficiency @64K rapidly diminishes.

In fFigure 256, road truncation effects are also visible: since the average number of road fired increases (almost) linearly with respect to scaling factor, a higher road truncation limit is mandatory in order to achieve better efficiency at higher factors.

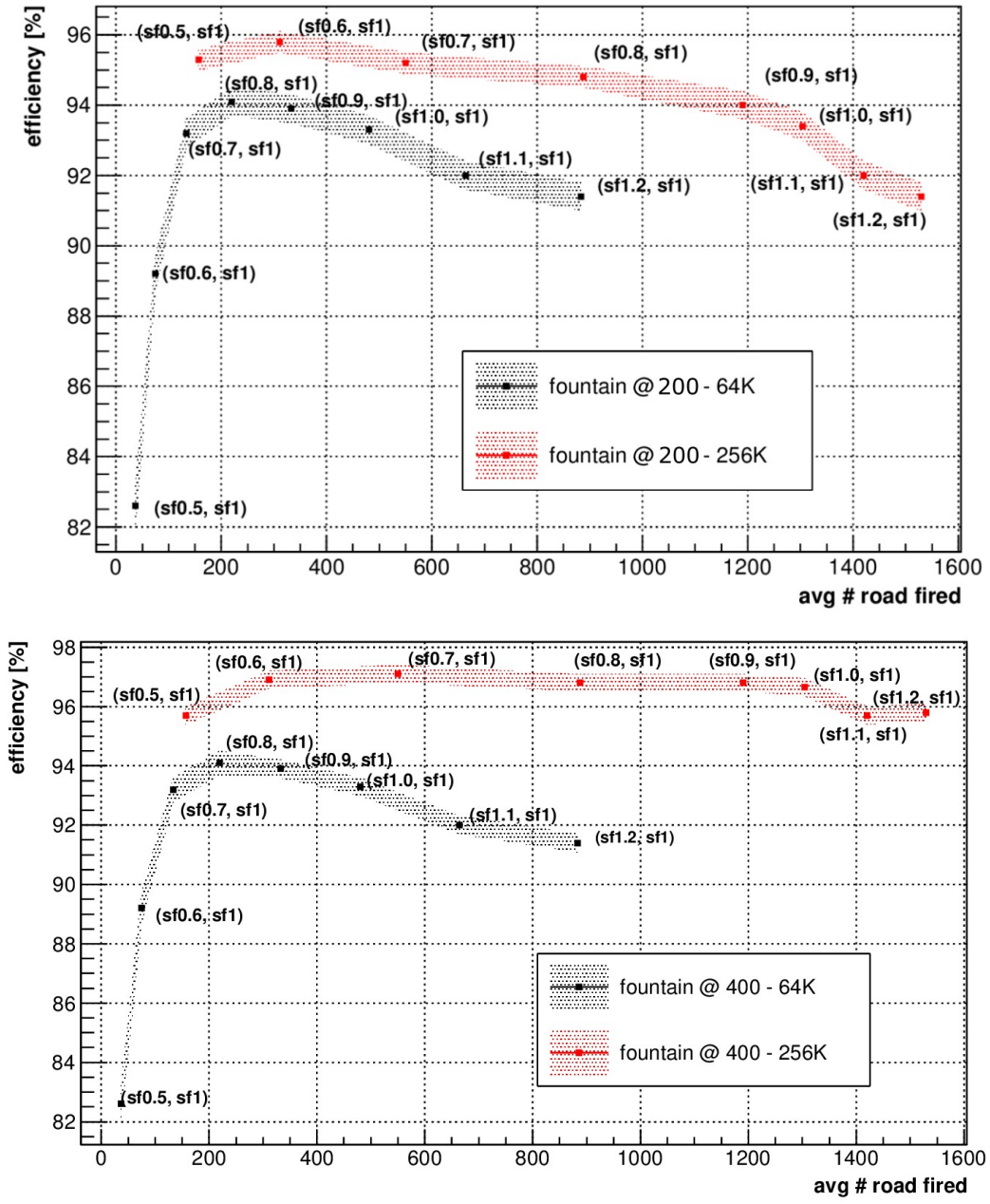


Figure 26 – Road efficincy vs average number of road fired, at 200 roads truncation limit (the previous plot) and 400 roads truncation limit (this one). Banks with bigger size are likely to fire more roads, with respect to scaling factor.

When checking for AM bank performance, an important information comes from bank size versus the total number of combinations output to fitting stage; data show that bank size $B_{size} \propto \beta / n_{combs}$. Bank size reflects the hardware cost (and implementation risk,) due to added complexity in miniaturizing bigger banks); on the other hand, the average total number of combinations reflects the total time needed to perform track reconstruction. If dealing with banks not limited in size, the best configu-

ration would simply consists in using the smallest possible scaling factor (without making superstrips smaller than the size of a module strip).

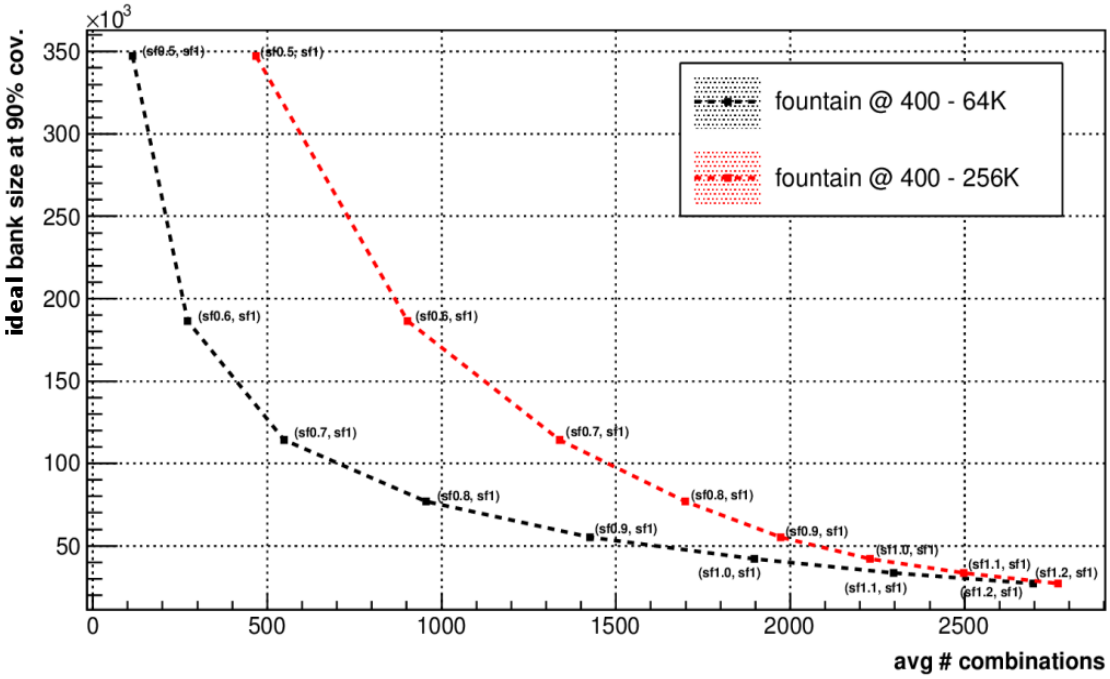


Figure 27 – Ideal Bank size at 90% coverage versus to average total number of combinations output to fitting stage at 400 roads truncation. The optimal working scaling factor should be located as close as possible to origin.

Since AM banks have fixed limited size, when firing roads only a fraction of the full ideal bank is used, depending on the scaling factor. At small scaling factors bigger banks are mandatory to reach the desired coverage; if size is limited by hardware constraints, the number of roads fired (thus the total number of combinations) as well as efficiency are limited by loss of coverage. An increasing difference between the ideal bank size and the actual bank size reflects on a decreasing β factor, as visible in Figure 27, where results for 64K and 256K bank at different scaling factor are compared. The optimal working range at fixed bank size is thus around scaling factors that are able to reduce the total combinations number but that also need banks whose ideal size is not exceeding too much the actual limited size.

	avg # roads fired	avg # tot combs @200	eff. [%] @ 200	avg # tot combs @800	eff. [%] @ 800
flat	122	460	99.2 ± 0.2	533	99.2 ± 0.2
tilted	157	390	95.8 ± 0.3	495	97.7 ± 0.3

Table 3 – Results comparison between best case configuration for flat geometry (sf = 0.5, 256K), and for tilted geometry (sf = 0.6 @ 200 roads truncation, 256K; sf = 0.7 @ 800 roads truncation, 256K).

Similar numbers in terms of total combinations and average road fired are noticeable when comparing best case results with 256K bank and $sf = 0.5$ obtained with the flat geometry, as expected since modules layout doesn't change dramatically in the barrel tilted configuration and average occupancy per superstrip with PU200 is almost the same.

Section 6.02 Hybrid tower road efficiency

Tower 33 road efficiency has been investigated by comparing fountain and flower banks, varying scaling factor in the barrel in the range sf $\in$ [0.7, 1.3], until the optimal working point is found. As reported in flower p_T study section, for each scaling factor the bank has been tuned in p_T in order to obtain the best decoupling from η parameter, and minimize the number of patterns to get 90% coverage. Results are shown in Figure 28; in table are collected the best case scenarios for both flower and fountain.

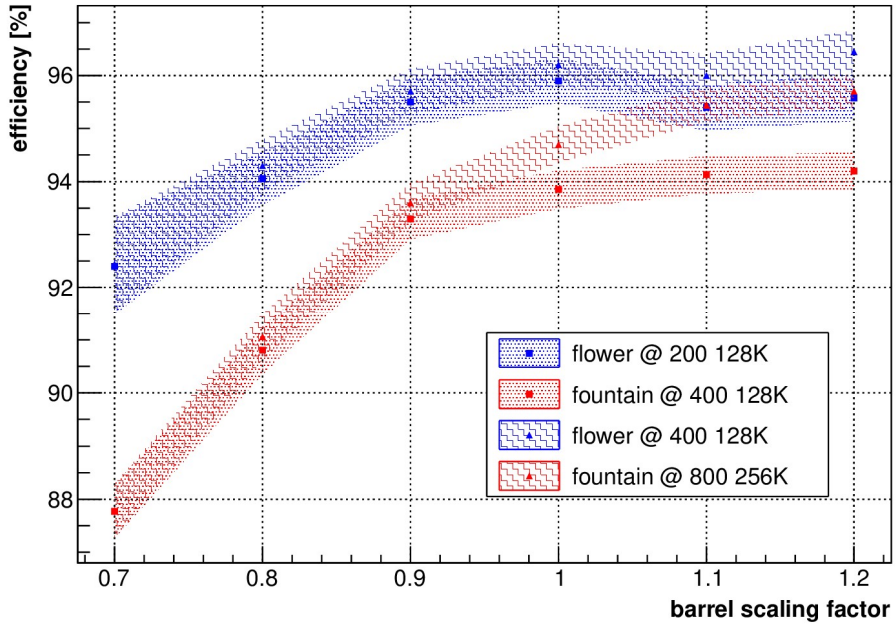


Figure 28 – Road efficiencies for flower (in blue) and fountain (in red) at different scaling factors and road truncation. A fair comparison has been made using fountain 256K bank and flower 128K + 128K bank.

Efficiency saturation exists when scaling factor exceeds $sf = 1$; on the other hand, road efficiency quickly decreases, when scaling factor in the barrel decreases under $sf < 0.9$. Since efficiency loss is not recovered by increasing the roads truncation limit, under $sf = 9$ bank coverage limit is hit, and fewer than 200 roads (for flower; fewer than 400 for fountain) are fired, as confirmed by checking the average number of road fired per event. Table 4 compares the best cases, with the maxium road trunca-

tion limit of 400 for flower and 800 for fountain, achieved in saturation region; the relative difference in this region between the two configurations is reduced.

	Avg road fired	Avg # combinations	Road efficiency
Fountain @256K	555	2306	95.8 ± 0.4 %
Flower @128K matching charge	283	1190	96.5 ± 0.4 %
Flower @128K not matching charge	249	1014	7.0 ± 0.5 %

Table 4 – Best case scenario (saturation). Fountain @ 800 roads and flower @400 roads truncation, with disk scaling factor sf = 1.2.

Under *sf* = 0.9 efficiency profile at road truncation 400 and 800 are overlapping for both flower and fountain, due to the fact that roads fired are well under 400; on the other hand, over *sf* = 0.9 significant difference is visible.

If considering the more conservative configuration of 400 roads truncation limit for fountain, the best efficiency is achieved for fountain with *sf* >= 1 at ~ 94% (saturating) with an average number of total combinations of ~ 1000 (figure 28); for flower best efficiency is achieved with 200 roads truncation (plus 200 from don't matching bank) and *sf* = 1 at 96% (saturating), with a similar to fountain number of total combinations. Therefore the inefficiency drops from 6% to 4%, quite an improvement.

	Avg road fired	Avg # combinations	Road efficiency
Fountain @256K sf =	438	1800	95.4 ± 0.3 %
Flower @128K matching charge	130	500	95.7 ± 0.4 %
Flower @128K not matching charge	109	390	4.0 ± 0.4 %

Table 5 – Efficiency threshold at 95%. Best case flower manages to achieve better efficiency at a lower scaling factor (thus less combinations).

Flower-superstrip design due to its better coverage at lower scaling factor is able to keep up efficiency even when fountain is hitting coverage limit. Table 5 shows best case when imposing a minimum efficiency threshold of 95%. At *sf* = 0.9 flower manages to beat fount efficiency with bigger scaling factor *sf* = 1.1 with 95.7 ± 0.3% efficiency with respect to 95.4%, but with down to ~ 50% less total combinations (900 with respect to 1800) due to smaller superstrip, and to the fact that fewer roads are firing at lower scaling factor.

Flower design prove to achieve similar efficiency levels (if not more) but with a lower scaling factor.

Don't matching charge bank has efficiency under 8 % at 200 roads truncation limit, slightly dependent from scaling factor, with up to 99% of fake combinations, the same holds for tower 41. At last, following figure 29 shows in detail efficiencies and average number of combinations for TT33.

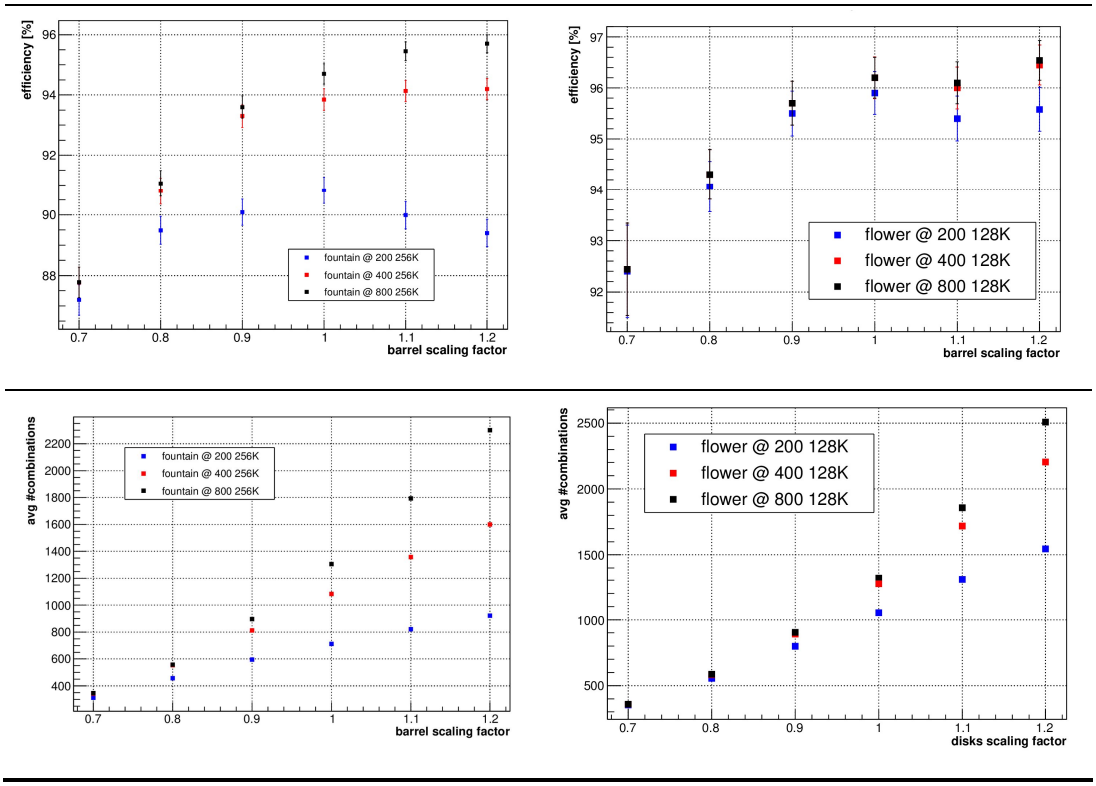


Figure 29 – Road efficiency vs average number of road fired, for different barrel scaling factor, for flower (128K + 128K) and fountain (256K). For flowers, total combinations are the sum of combinations output from matching charge bank and not matching charge bank.

Section 6.03 Forward tower

Tower 41 road efficiency has been investigated by comparing fountain and flower banks, varying scaling factor in the disks sections only (since the majority of the modules are located there) in the range sf ∈ [0.4, 0.8]. Once the best scaling factor for disks has been found, further configurations have been tested, trying to improve the barrel scaling factor. Bank p_T study at different disks scaling factor has proven that optimal p_T is in range 8 - 12 GeV/c, slightly decreasing with disk scaling factor.

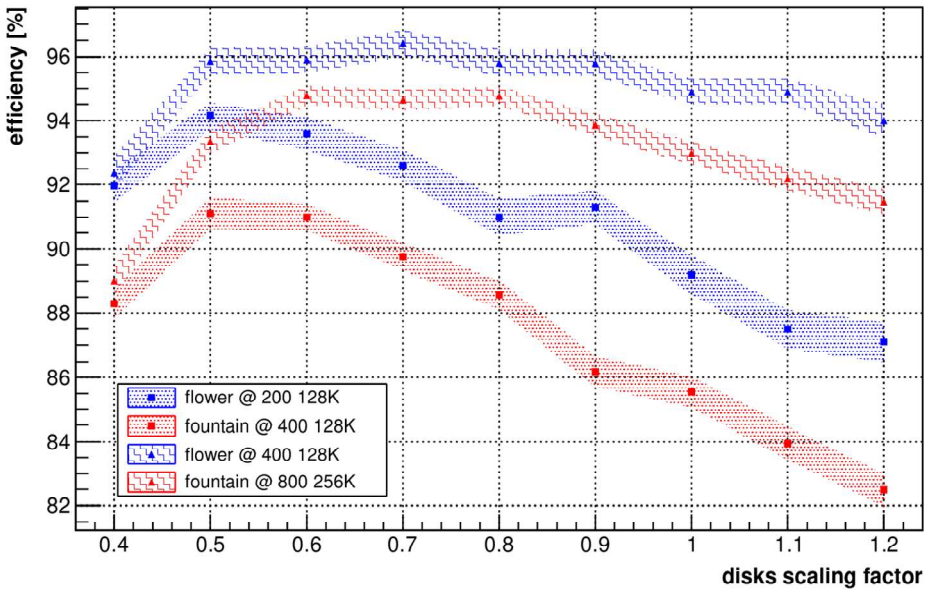


Figure 30 – TT41 road efficiency comparison. With flower truncated at 200 roads and fountain truncated at 400 roads, best disk scaling factor is for both set to sf = 0.5. Flower shows at sf = 0.9 with 200 roads truncation a probable statistical fluctuation.

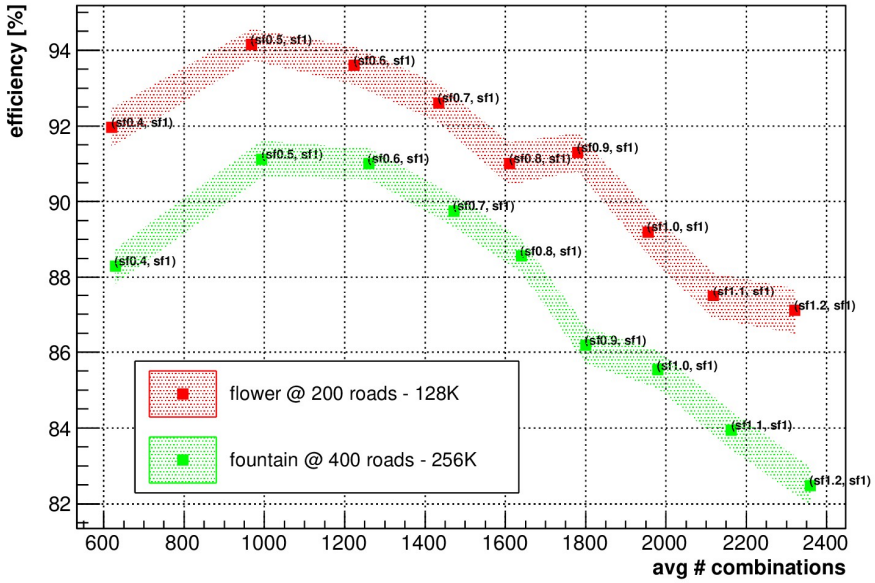


Figure 31 – Road efficiency comparison between fountain @400 roads and flower @200 roads, with respect to the average number of total combinations (for flower, the sum of matching and not matching roads is showd). Flower shows at sf = 0.9 a probable statistical fluctuation.

Tower 41 efficiency profile as a function of the disk scaling factor is quite different from the hybrid tower one. Figure 30 and Figure 31 show that best efficiency is achieved with roads truncation set at 400 and 800 for flower and fountain espec-

tively; results are collected in table 6. Flowers manage to reduce inefficiency from 5.2% to 3.6%, again a great performance.

	Avg road fired	Avg # combinations	Avg # fake combs	Road efficiency
Fountain 256K sf 06	750	680	672	94.8 ± 0.4 %
Flower 128K sf 07 matching charge	510	1304	1286	96.4 ± 0.3 %
Flower 128K not matching charge	482	1171	1170	13.0 ± 0.7 %

Table 6 – Tower41 best case with fountain road truncation 800 at sf = 0.6, and flower road truncation 400, at sf = 0.7. Average number of fakes combinations is also showed; the number of duplicates is around 8 for fountain, and up to 18 for positive matching bank.

When increasing scaling factor an almost linear efficiency loss is noticeable, in contrast to the efficiency saturation that characterized hybrid tower when increasing scaling factor in the barrel.

If setting an efficiency threshold at 94% and considering the best case with 400 and 800 for flower and fountain road truncation limit respectively (table 7), flower manages to reduce inefficiency from 5.2% to 4.1%, with a reduction in average total combinations down to 37% from 2000 to 1369: a massive global improvement indeed. Flower design prove to achieve similar efficiency levels (if not more) but with a lower scaling factor.

	Avg road fired	Avg # combinations	Avg # fake combs	Road efficiency
Fountain 256K sf 06	715	2000	1983	94.8 ± 0.3%
Flower 128K sf 05 matching charge	218	685	670	95.9 ± 0.4%
Flower 128K sf 05 not matching charge	199	584	584	9.7 ± 0.6%

Table 7 – Road efficiency threshold at 94%: best case are shown that satisfy the threshold. Fountain with disk scaling factor sf = 0.6 and flower with sf = 0.5.

In figure 33 a detailed view over the average number of combinations comparison is shown.

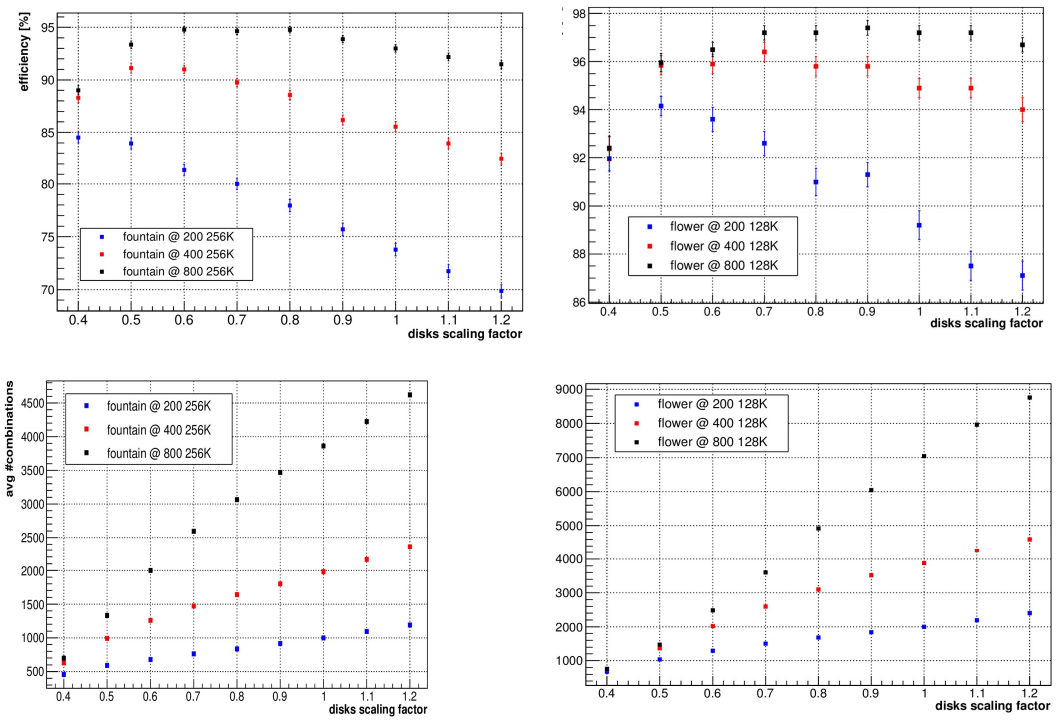


Figure 32 – (top) Road efficiencies at different road truncation limits, for fountain (256K) and flower (128K + 128K); (bottom) average number of total combinations, for fountain and flower with same settings as top, flower combinations are show as the sum of matching charge bank and not matching charge bank output roads total combination.

Section 6.04 Full turn on curves for forward tower

The main reason for which flower design has been developed is to improve road efficiency when modules spanning different r values are present, such as in inner barrel layers or disks. Usual fountain design exhibits a low global road efficiency over forward and hybrid tower, in particular, it lacks efficiency in reconstructing tracks with p_T lower than 10 GeV/c, while a general inefficiency is present over the whole tower η range.

In this section's figures, fountain results will be shown with usual roads truncation limit of 200 (blue), 400 (red) and 800 (black), while flower configuration will be show with a low road truncation limit of 100 (blue), then 200 (red) and 400 (black); this choice reflects the fact that flower banks have proven to fire less than 400 roads usually.

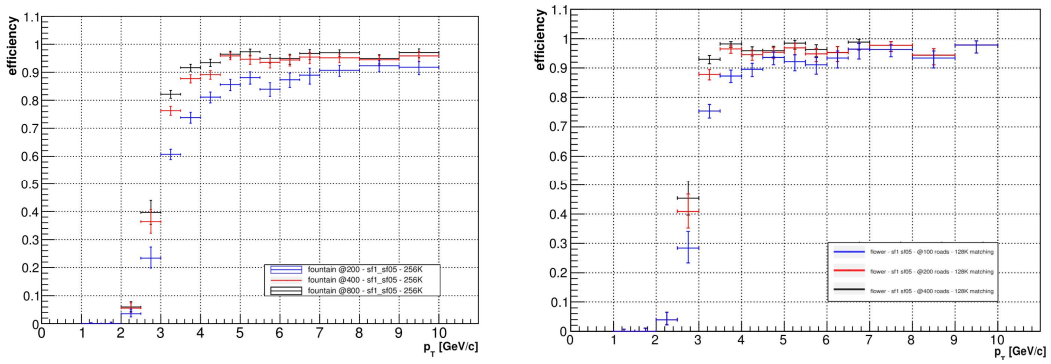


Figure 33 – Full p_T turn on curve, focus on lower p_T. On the left, fountain bank with 256K, disk scaling factor sf = 0.5 (best case @ 400 roads truncation); on the right, flower bank with 128K + 128K, disk scaling factor sf = 0,5 (best case @200 roads truncation for fair comparison). Check the below text for color info, due to small legend.

In Figure 33 a comparison between p_T turn on curves for tower 41 is showed, including both fountain and flower results. Turn on efficiencies are collected in following table 8 for the best case 200/flower and 400/fountain both with disk sf = 0.5.

	Fountain (256K, disk sf = 0.5)			Flower (128K +128K, disk sf = 0.5)		
p_T	Eff @ 200	Eff @ 400	Eff @ 800	Eff @ 100	Eff @ 200	Eff @ 400
1 – 1.5	0%	0%	0%	0%	0%	0%
2	4%	6%	6%	4%	4%	4%
2.5	24%	36%	40%	28%	40%	45%
3 (threshold)	60%	76%	81%	75%	88%	94%
3.5	74%	88%	92%	88%	96% (platau)	98% (plateau)
4	82%	89%	94%			
4.5	85%	95% (plateau)	95% (plateau)			

Table 8 – Turn on curve for trasverse momentum, detail until plateau is reached. Flower reaches it already at 3.5 GeV/c with respect to 4.5 GeV/c, with a sharper turn on.

Flower design is effective in improving AM performance for low p_T events, whose p_T is in fact close to the bank's tuning p_T. With disk scaling factor 0,5 (the best case for both fountain and flower), the sharpness of p_T turn on curve over 3 GeV/c threshold is clear, even at 200 roads truncation limit (in red in the picture): the lowest efficiency is set to ~5% at 2 GeV/c, rising at 40% at 2.5 GeV/c, while at 3 GeV/c is already at 90% (95% at 400 roads truncation limit). In fact, due to the low number of road fired from the matching charge bank, no significant difference is present at different roads truncation, and already at 200 roads truncation the maximum efficiency is achieved over 3 GeV/c, where a plateau is noticeable. On the other hand, fountain performance is poor under 10 GeV/c; even at 800 roads truncation the turn

on curve is not sharp and 90% efficiency is achieved over 4 GeV/c (95% at 6 GeV/c); the lowest efficiency is set with 400 roads truncation at slightly over 5%, rising just under 40% at 2.5 GeV/c, similarly to flower performance. At high p_T (over 20 GeV/c, not shown in the plot due to its dimension) both fountain and flower p_T efficiencies reach a plateu with road truncation over 200. In fact, with 100 roads truncation, flower performance at high p_T is decrising, due to the fact that low bending tracks are less likely to trigger a banks pattern tuned for low p_T.

It is worth stressing here that these turn on curves do not define the full L1 track trigger turn on curves, but only measure the function of track candidates below threshold that the AM erroneously promotes and passes to the fitting stage. It is indeed at that stage, where stubs are processed with their full special resolution, that track parameters (among there, the p_T) are obtained.

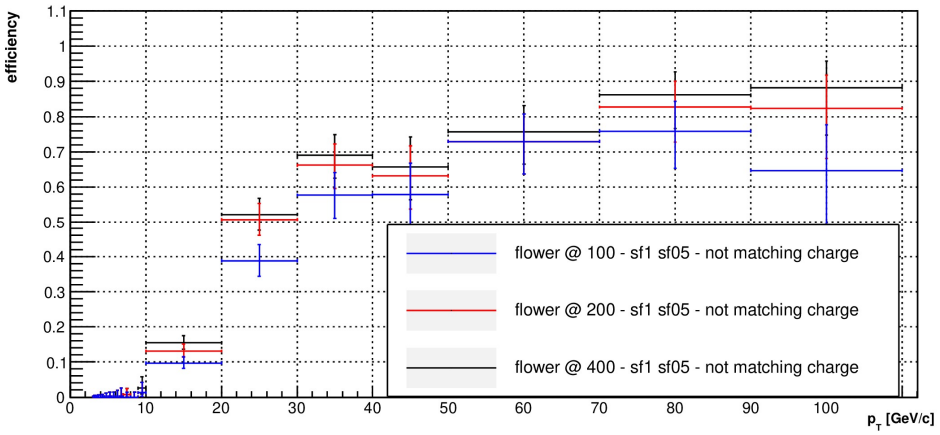


Figure 34 – p_T turn on curve for not matching charge flower bank, with 128K + 128K size at disk scaling factor sf = 0.5. Bins over 10 GeV/c have increased size, following sample p_T distribution. Usual colors are shown for 100, 200, 400 roads truncation limit, although not much difference is noticeable.

In order to fully characterize flower performance, it is interesting to check also not matching charge bank results; until now, not matching charge output has not been mentioned in detail, except that for the very low efficiencies (as expected, under 10% slightly depending on scaling factor). In Figure 34 the turn on curve for not matching bank with scaling factor sf = 0.5 (same as best case shown before), is show, with particular emphasis over higher p_T range. As expected, the road efficiencies is very low under 10 GeV/c (around bank's tuning p_T), on the other hand, efficiency is increasing with p_T until a sort of plateau around 80% is reached. The increased efficiency at higher p_T is expected, since particles whose charge don't match with bank charge are more likely to find a pattern with small curvature, instead then a pattern with high curvature in the wrong direction at low p_T .

The last interesting question about p_T turn on performance yet to answer is the AM intrinsic efficiency in recognizing tracks above a threshold that is independent from

module threshold; with this goal banks of both fountain and flower at disc scaling factor sf = 0.5 have been generated using a training sample of muons with p_T > 5 GeV/c, thus setting the turn on threshold at 5 GeV/c; road efficiency has been computed using the usual sample of PU200 with muons with p_T starting from 3 GeV/c. Figure 36 shows the results: flower manages to get a sharper profile between a very low efficiency (compatible with 0%) under 3.5 GeV/c to the maximum efficiency of over 95% with 200 roads at p_T = 5 GeV/c; on the other hand, fountain still show up to 7% efficiency with 400 roads at 3 GeV /c, and similar performance to flower over 5 GeV/c.

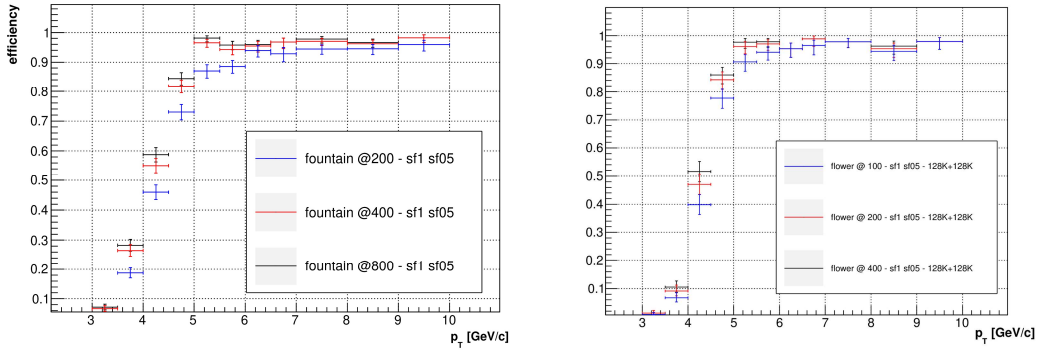


Figure 35 – Turn on curves for a 5 GeV/c threshold bank. Fountain on the left and flower on the right. Flower roads truncation limit is set at 100, 200, 400 roads for a easier comparison with fountains colors.

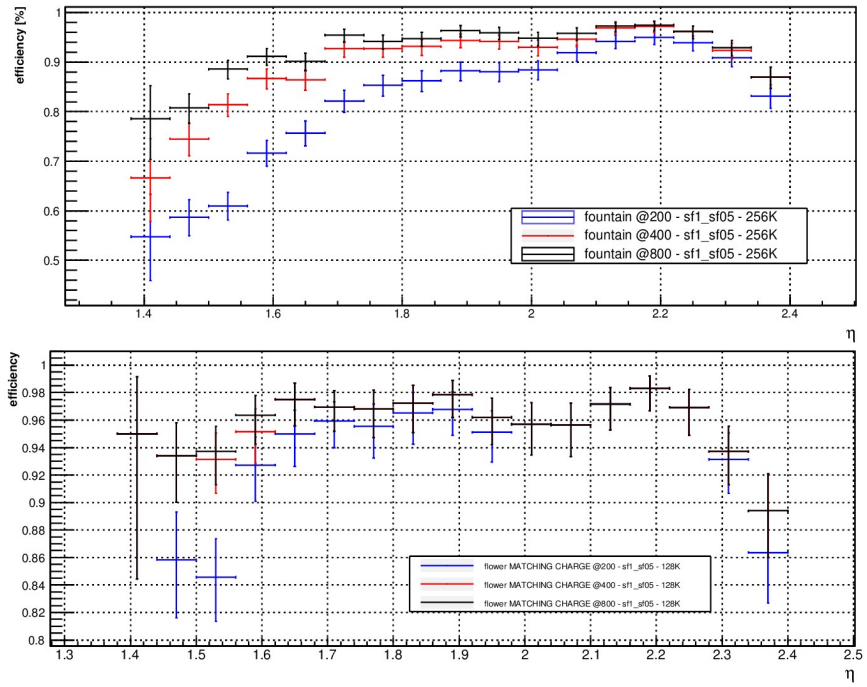


Figure 36 – η turn on curves for fountain (at the top) and flower matching charge bank (at the bottom).

Flower design shows performance improvements over fountain design also over η tower range. In fountain η turn on plot (figure 26) are visible around η = 1.45 and η = 2 two jumps in efficiency due to particles having to cross one less layer of tilted modules in the barrel at each jump; moreover is clear the linear dependence over z-coordinate (that is, over η after a remapping), explained when discussing flower superstrip design.

Flower design manages to get over 10% better efficiency over the whole range, in particular at lower η where tilted modules are present. A loss of efficiency at high η is still visible in both pictures, that no increase in roads truncation limit is able to resolve; this is likely due to a lack of coverage at very high η, a region where tracks struggle to leave at least 5 stubs in disks layers and are thus removed from pattern generation.

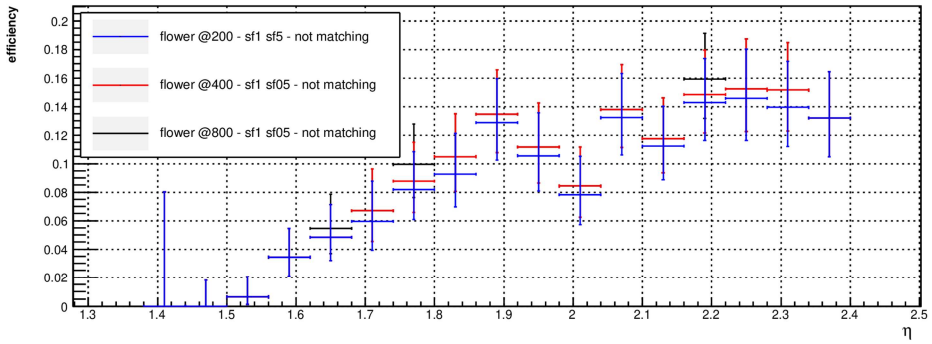


Figure 37 – η turn on curve for TT41 flower not matching bank, 128K +128K with disk sf = 0.5.

The not matching charge flower bank has bad performance over the whole η range (under 14% efficiency); the efficiency profile resembles the fountain one, except for the very low efficiencies due to not matching charge.

Chapter VII Conclusions and future development

An approach based on Associative Memories (AMs) for the CMS L1 track trigger upgrade has been investigated, by extending current studies – limited to the flat barrel geometry only – to the whole tracker and with an updated geometry. The AMs allow to perform track recognition over a coarse-grained representation of the tracker, in which tracker layers are divided in sections called “superstrip” (SS); tracks are therefore collected in equivalence classes of superstrips called “patterns”. The patterns generated by physical tracks are stored into banks of patterns and subsequently used

by the AM to perform massive parallel pattern recognition. This is obtained by matching the patterns stored in the bank with the ones generated by the O(200) PU interactions per event. This approach "moves" offline (e.g. precomputing the physical patterns) part of the load of discriminating between the hits (called "stubs") coming from interesting (i.e. "high-p_T") tracks from the low momentum ones. Stubs belonging to the matched patterns are then sent to a linearized fitting stage in order to extract track candidates to be used as trigger primitives for L1 trigger, with latency limit up to 4 μs.

A way to handle the stubs from more than six layers (the current AM design adopts 6-SS patterns) has been developed thus making the AM approach feasible also the in hybrid and forward region where up to 8 layers can be crossed by a single track.

The current superstrip design (referred as "fountain") has been extended to the whole detector and it has been shown to have good performance in the barrel region. An original new "flower" design has been developed and compared to the old one, and performance improvements have been demonstrated in the forward region, where the tracker modules are arranged in disks and also in the barrel region where tilted module are present.

In the barrel tower, the fountain design has been exported onto the new modules layout, showing only a few differences with respect to the old flat barrel geometry: the presence of tilted modules at the tower boundaries is sufficient though to reduce road efficiency down to 97% in the best case, with respect to over 99% achieved with the old tracker flat geometry.

The fountain design shows its limit when dealing with tilted modules: a moderate efficiency loss (~10 %) for tracks with p_T <10 GeV/c pointing towards the tilted modules has been measured. The average reconstruction efficiency reaches 96% in forward tower and up to 96.5% in hybrid tower, with an average of 1000 combinations output to the fitting stage, in the best cases.

The flower design aims to gain better efficiencies in the hybrid and forward regions by adopting a new original design tuned upon track's curvature. The flower design calls for more complexity since it needs two separate banks to guarantee the full coverage for positively and negatively charge tracks.

Flower design manages to increase pattern bank coverage with 256K standard size for both hybrid and forward tower from 70% to 80%. In both hybrid and forward towers, although with quite different efficiencies profiles due to different superstrip size tuning, flower design manages to achieve up to 4% efficiency increase (reaching 95%) with respect to same configuration settings for fountain, which struggles to achieve up to 90% efficiency. When comparing equivalent efficiency levels, the flower design is able to achieve similar efficiency levels as fountain design, with an average number of combinations sent from AM to the fitting stage reduced by over 30%, both in the hybrid and the forward region.

Flower design improvements are noticeable also in p_T turn on curves, at $p_T < 10$ GeV/c: in fact, flower exhibits sharper turn on curves with the maximum efficiency plateau reached at modules p_T threshold of 3 GeV. η turn on curves show efficiency improvements up to 10%, in particular in η regions that cover tilted modules layers.

As mentioned, the drawback of tilted design lies in the need to partially duplicate the AM chain: two banks are needed, and FPGA must transcode twice the stubs to feed both banks. Moreover, the roads fired from the not-matching-charge bank are mostly (>90%) fake roads; and when computing total combinations, up to 99% (on average) of them are from random stub combinations.

In conclusion, the AM approach has been successfully exported to the whole tracker, and current design superstrip has been improved: when dealing with restrictive time constraints, as for the L1 trigger update project, managing to achieve similar efficiencies results as with an old barrel optimized design, but with over fewer total number of combinations to pass to fitter or with higher efficiencies for the same number of combinations to be processed.

Future developments are focusing on investigating AM approach performance with high PU events (over 400) do determine the limits of the chain with respect to events with a very high stub density. Current studies on barrel only have already proven that AM approach struggles to achieve optimal efficiencies due to the high number of combinatoric roads fired. In order to overcome this problem, new AM optimization method are currently investigated, e.g. the stub bending optimization, which focuses on discriminating patterns in the AM using track curvature information coming from modules themselves. In high density events each fired road contains many more combinations to be read out and processed by the following stages (e.g. the fitting one), thus dramatically increasing the total reconstruction time. In order to reduce the stubs combinations to fit for each fired roads, a Hugh transform implementation is being studied: it has already proven to be extremely effective for the barrel region, where only two track parameters are necessary to discriminate tracks (φ_0 and ϱ); an implementation of the Hugh transform in forward towers, where all four track parameters are needed, is currently under development.

In order to complete the performance study on hybrid and outer towers, exporting the current linearized fitter to those towers is mandatory and its implementations over disks will require to fully rework parameters optimization by re-training the fitter using tracks that covers the whole detector; once the fitter will be online, it will be possible to measure time performance of the whole track trigger chain.

The performance study of stub bending optimization along with flower superstrip design over the full track trigger chain with a reworked linearized fitter will be of great interest in investigating AM performance over hybrid and forward towers in presence of high PU events.

Bibliography

[1] Günther Dissertori, "The LHC endeavour, achievements and future plans", DFA Colloquium, Padova, 11/15/'17.

[2] CMS Collaboration, "Technical proposal for the Phase-II upgrade of the Compact Muon Solenoid", *CMS-TDR-15-02*, 2015.

[3] CMS Collaboration, "CMS Physics", *CMS TDR 8.1,* Vol. 1, 2006.

[4] Kostas Kloukinas, "CMS Pixel-Strip Project", *ACES 2014 – Workshop for LHC upgrades,* 03/19/'14.

[5] CMS Collaboration, "The CMS experiment at the CERN LHC", *JINST* 3 (2008) S08004, doi:10.1088/1748-0221/3/08/S08004.

[6] M. De Mattia et al. "Track fitting for AM+FPGA approach", 01/19'16, https://indico.fnal.gov/event/11322/session/4/contribution/41/material/slides/0.pdf.

[7] Anargyros Krikelis, Charles C. Weems, "Associative Processing and Processors", *Computer*, Vol. 27, Iss. 11, 1994.

[8] Richard O. Duda, Peter E. Hart, "Use of the Hough Transformation to detect lines and cuves in pictures", *Comm. ACM,* Vol. 15, No. 1, pp. 11-15, 1972.

[9] S. Jindariani, T. Liu, L. Ristori, S. Das, I. Furic, JF Low, Jacobo Konigsberg, "Revisiting AM superstrip configurations", *L1 AM meeting*, 21/01/'15.

[10] A. Honma, A. Mussgiller, P. Rose, "Approval f Upgrade Results", *Phase II Modules and Mechanics Approval Meeting,* 03/06/'15.

[11] Olmo Cerri, "CMS L1 Upgrade TT Definition", *Fermilab summer student midterm presentation,* 23/08/'16.

[12] S. Nahn, A. Canepa, "Outer Tracker Overview", *Technical Design Review,* 09/13/'17.

[13] Luciano Risotri, "Real time track finding with associative memories", 03/05/'14.

[14] R. Rossin et al.,"Benchmarking the AM approach @ LPC".

[15] Miquèias M.A., "Stub bending in the AM – Performance studies @64K"m 08/17/'17.