



DEVELOPMENT OF A LEVEL-1 TRACK AND VERTEX
FINDER FOR THE PHASE II CMS EXPERIMENT
UPGRADE

DAVIDE CIERI

A dissertation submitted to the University of Bristol
in accordance with the requirements for award of the
degree of Doctor of Philosophy
in the Faculty of Science

January 2018

Word Count : 30,000



Davide Cieri: *Development of a Level-1 Track and Vertex Finder for the Phase II CMS experiment upgrade*, January 2018

SUPERVISORS:

Prof. Dave Newbold

Prof. Claire Shepherd-Themistocleous

Bristol

January 2018

Discovery consists not in seeking new lands but in seeing with new eyes.

— Marcel Proust

ABSTRACT

The High Luminosity (HL-LHC) upgrade to the Large Hadron Collider will operate at an increased instantaneous luminosity, up to seven times the design value, in order to collect an integrated luminosity of $3,000 \text{ fb}^{-1}$ in the decade following 2025. Proton bunches at the HL-LHC will cross every 25 ns, producing an average of 140-200 pile-up proton-proton collisions per crossing. A new tracking detector is under development for use by the CMS experiment at the HL-LHC. A crucial requirement of this upgrade is to provide the ability to reconstruct charged particle tracks with transverse momentum above 2–3 GeV within $4 \mu\text{s}$ to be used in the Level-1 (L1) trigger decision.

This thesis presents one of the main proposals for the final L1 Track Finding system, which exploits a fully time-multiplexed architecture based on high-speed FPGA electronics. The developed track finding algorithm makes use of the Hough Transform technique to identify track candidates, followed by a track fitting stage. Several options for the track fitting task have been explored. One of the most promising has been identified in the so-called Seed Filter plus Linear Regression fitter. The chosen track finding algorithm is able to reconstruct efficiently tracks from charged particles with a processing latency well below the assigned budget.

An hardware demonstrator has been built using existing CMS data processing boards, known as MP7, and has proved the feasibility to perform track finding at the HL-LHC within the assigned latency with good reconstruction efficiency.

The thesis explores also a possible use of the L1 tracks to identify primary vertices at the L1 Trigger stage. Several clustering algorithms have been studied. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm has been found as the most promising option for a final application, showing good performance and being feasible for a FPGA implementation.

DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Bristol, January 2018

Davide Cieri

Share your knowledge. It is a way to achieve immortality.

— Dalai Lama

ACKNOWLEDGMENTS

The four years spent in England for my PhD studies have been one of the most intense and pleasing periods of my life. During this time I had the possibility to meet awesome people, without whom it would not have been possible for me to make it.

First of all, I have to thank my two supervisors, Claire and Dave, who have been an outstanding source of inspiration and supported me, especially during the writing period.

Thank all my colleagues at RAL, Alessandro, Ian, Kostas, Kristian, Luigi, Luis, Manny, Tom and Thomas. We shared a great time and, I believe, we made a great team.

Also thank my PhD fellows in Bristol, Chiara, Doug, Emma, Leonor and Gabbie. Without you, my first year in Bristol would not have been the same.

Also thanks to all the fantastic people I met in all the different INFIERI workshops and schools, especially Andr , Benedetta, Daniele, Elena, Maria Teresa, Matteo and Nilou. We shared some very nice moments and I hope to see you all in the future.

Then, I should thank all my friends in England, Cecilia, Claudia, Davide, Danica, Dave, Diego, Dimi, Elisa, Giacomo, Giulio, Marie, Michela, Michele, Monica, Olgar, Sofia, Valentina and all the people I am blamable forgetting. You have been a second family to me and will be always in my heart.

I am also very thankful to the INFIERI collaboration management, especially to Aurore, who gave me the opportunity to explore new places and to meet formidable scientists.

Finally, I have to thank my parents and my brother, who support me in any of my adventures and are always close even being far away.

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n.[317446] INFIERI "INtel-

ligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Research and Industry“.

CONTENTS

1	INTRODUCTION	1
1.1	The Large Hadron Collider	1
1.2	Physics at the LHC	3
1.2.1	The Standard Model of Particle Physics	4
1.2.2	LHC results	7
1.2.3	Open questions in high-energy physics	8
1.3	The High Luminosity LHC Upgrade	9
1.3.1	HL-LHC Physics Motivations	11
1.3.2	Detector Challenges at the HL-LHC	12
1.4	The CMS Experiment	15
1.4.1	Coordinate system	16
1.4.2	The Silicon Tracker	18
1.4.3	Superconducting magnet	22
1.4.4	Electromagnetic calorimeter	22
1.4.5	Hadron calorimeter	24
1.4.6	Muon system	27
1.4.7	Trigger	29
2	THE PHASE II CMS UPGRADE	33
2.1	The Phase II Tracker Upgrade	34
2.1.1	Requirements	35
2.1.2	The Inner Tracker Upgrade	35
2.1.3	The Outer Tracker Upgrade	36
2.2	The CMS Phase II L1 Trigger	41
2.2.1	Calorimeter trigger	42
2.2.2	Muon Trigger	43
2.2.3	The L1 correlator trigger and the L1 Global Trigger	44
2.3	The Track Trigger	44
2.3.1	Track trigger architectures	47
2.4	Trigger algorithms	49
2.4.1	Matching muon trigger primitives to L1 tracks	50
2.4.2	Matching L1 tracks to ECAL clusters	51

3	AN FPGA-BASED TIME MULTIPLEXED TRACK TRIGGER	53
3.1	A fully Time multiplexed system architecture	55
3.1.1	An FPGA-based Time Multiplexed Track Finding system . . .	58
3.2	L1 track finder processor	61
3.2.1	Geometric Processor	62
3.2.2	Track finding with Hough Transform technique	67
3.2.3	Track filtering and fitting	71
3.2.4	Duplicate removal	74
3.3	Seed Filter and Linear Regression algorithm	75
3.3.1	The Seed Filter	75
3.3.2	The Simple Linear Regression Fitter	78
3.4	TMTT Track Finding performance	86
3.4.1	Track Reconstruction Efficiency	86
3.4.2	Track Parameter Resolution	90
3.4.3	Summary	94
3.4.4	Finding tracks below 3 GeV	95
3.4.5	Preliminary Tilted Geometry results	97
4	HARDWARE DEMONSTRATION OF THE TMTT L1 TRACK FINDER PRO- CESSOR	103
4.1	The Demonstrator Setup	103
4.1.1	The Master Processor MP7 board	104
4.1.2	Data Formats, Mappings and Link Specifications	105
4.1.3	FPGA resources	107
4.2	The demonstrator software framework	107
4.2.1	Pattern Writer	109
4.2.2	Unpacker	109
4.2.3	Comparison Software	109
4.3	Track Finding Processor hardware implementations	110
4.3.1	Geometric Processor	110
4.3.2	Hough Transform	114
4.4	Track Fitting with Seed Filter plus Linear Regression	117
4.4.1	Design overview	118
4.4.2	Track Locators	121
4.4.3	Control Unit	122
4.4.4	The Seed Filter plus Linear Regression	124

4.4.5	Duplicate Removal	132
4.5	Demonstrator results	133
4.5.1	Output data rate	133
4.5.2	L1 track reconstruction efficiency	136
4.5.3	Track Fit parameters resolution	137
4.5.4	Latency	137
4.5.5	FPGA resource usage	142
4.6	Final system scaling	142
4.6.1	FPGA resources	143
4.6.2	Latency	144
4.7	Summary	145
5	PRIMARY VERTEX FINDING AT L1 TRIGGER LEVEL	147
5.1	Simulated Data and Definitions	148
5.2	Vertex reconstruction algorithms	149
5.2.1	FastHisto algorithm	150
5.2.2	Agglomerative Hierarchical Algorithm	150
5.2.3	Divisive Hierarchical Algorithms	155
5.2.4	Density-based Clustering Algorithms	156
5.2.5	Partitioning Clustering Methods	159
5.2.6	Algorithms summary and conclusions	162
5.3	Vertex Reconstruction Performance with DBSCAN	164
5.3.1	Primary vertex reconstruction efficiency	164
5.3.2	Primary vertex resolution	165
5.3.3	Pileup effects	166
5.3.4	Vertex-based triggers	169
5.3.5	Changing the p_T -threshold	172
5.3.6	Vertex reconstruction in different final states	177
5.4	Vertexing with a tilted barrel geometry	179
5.4.1	Primary vertex reconstruction performance	179
5.5	Summary	180
6	CONCLUSION	183
A	SIMPLE LINEAR REGRESSION	187
A.1	Regression analysis	187
A.2	Simple Linear Regression	188
B	IMPLEMENTING DIVISIONS IN FPGAS	191

B.1 A serial fixed point division implementation	191
--	-----

BIBLIOGRAPHY	201
--------------	-----

LIST OF FIGURES

Figure 1.1	The CERN accelerator complex. [8]	2
Figure 1.2	Cumulative luminosity versus day delivered to Compact Muon Solenoid (CMS) during stable beams and for p-p collisions. This is shown for 2010 (green), 2011 (red), 2012 (blue), 2015 (purple) and 2016 (orange) data-taking. [11]	3
Figure 1.3	The Standard Model (SM) of Elementary Particles. Sourced from [13].	5
Figure 1.4	(a) Standard Model Higgs boson production total cross sections at $\sqrt{s} = 7, 8$ and 14 TeV. (b) Standard Model Higgs boson decay branching ratios. [28]	7
Figure 1.5	Summary of Higgs boson mass measurements from the individual analyses of ATLAS and CMS and from the combined analysis. [33]	8
Figure 1.6	Projected Large Hadron Collider (LHC) performance through 2038, showing preliminary dates for long shut-downs (LS) of the LHC and projected luminosities. Sourced from [42].	10
Figure 1.7	LHC/High Luminosity Large Hadron Collider (HL-LHC) time-scale. [43]	10
Figure 1.8	Estimated precision of the coupling measurements for a SM-like Higgs boson. The projections assume a centre-of-mass energy of 14 TeV and a data-set with integrated luminosity of 300 (a) and 3,000 fb^{-1} (b). Scenario 1 assumes that the systematic uncertainties are as today, Scenario 2 that the experimental uncertainties scale as $1/\sqrt{L}$ and the theoretical uncertainties halved [46].	11
Figure 1.9	Mass reach of searches for super-symmetry based on simplified models [47].	12
Figure 1.10	Map of the expected particle fluence in the upgraded CMS tracker detector, to an integrated luminosity of 3,000 fb^{-1} , expressed in terms of 1 MeV neutron equivalent fluency [46].	13

Figure 1.11	Mean energy loss rate in different material for protons, muons and pions as a function of the particle momentum [51].	14
Figure 1.12	High pile-up event recorded at CMS during the high pile-up fill of Run 2 at center-of-mass energy of 13 TeV [53].	15
Figure 1.13	Schematic view of the CMS detector. [54]	16
Figure 1.14	Transverse slice of CMS detector in the barrel region. The signatures left by the different particles in the sub-detectors are also shown.	17
Figure 1.15	Layout of the CMS silicon tracker. Each line represents a silicon module with double lines indicating back-to-back modules which deliver stereo hits. [5]	19
Figure 1.16	Comparison of geometrical layouts of the old (bottom) and upgraded (top) CMS pixel detectors [56].	20
Figure 1.17	(a) Transverse and longitudinal primary vertex resolution as a function of the track multiplicity in the vertex for the old (black) and upgraded (red) pixel detector. (b) b-tagging performance for jets with $p_T > 30$ GeV using the old (open points) and the upgraded pixel detector (solid points). These studies have been conducted using top pair production events with an average pileup of 100. [57]	21
Figure 1.18	Longitudinal section (top) and 3D view of the CMS electromagnetic calorimeter (bottom). [5]	23
Figure 1.19	Longitudinal view of the CMS detector, highlighting the locations of the Hadron calorimeter (HCAL) sub-detectors. [5]	25
Figure 1.20	Longitudinal and transverse views of the CMS detectors, highlighting the position of the magnet iron ring and the Hadron outer (HO) layers. [5]	27
Figure 1.21	Transverse view of the CMS detector, showing the layout of the CMS barrel muon Drift Tube (DT) chamber stations (MB). [5]	28
Figure 1.22	The upgraded CMS Level-1 trigger architecture for LHC run II. [66]	30
Figure 1.23	Schematic view of a High Level Trigger (HLT) menu in CMS with its paths. The final trigger decision is the logical OR of the decisions of the single paths. [69]	31

Figure 2.1	Sketch of one-quarter of the Tracker layout in the r - z plane. Outer Tracker: blue lines correspond to PS modules, red lines to 2S modules (see text). In the Inner Tracker, the green lines correspond to pixel modules made of two read-out chips and the yellow lines to pixel modules with four readout chips. Produced with tkLayout [73].	34
Figure 2.2	Sketch of one quarter of the inner tracker in the r - z plane. Green lines correspond to pixel modules made of two read-out chips and the yellow lines to pixel modules with four readout chips. [74]	36
Figure 2.3	Operation of a p_T module. (a) Search for correlation of hits in the two closely-spaced sensors to reject low- p_T particle signals. The selection window for an accepted stub is depicted in green. (b) Wider windows are required at large radii, for the same transverse momentum. (c) Larger sensor spacing is required in the endcap disks to achieve the same rejection capability as in the barrel. [76]	37
Figure 2.4	Schematic views of the assembled silicon p_T modules. [74] .	38
Figure 2.5	Stub finding inefficiency in flat barrel PS modules. Each blue line represents half of a sensor. At the edge of the inner barrel layers a large irrecoverable inefficiency exists for tracks that cross the modules at the separation point of the two halves of a sensor (red lines). [76]	38
Figure 2.6	$r - z$ view of the Outer Tracker. PS (2S) modules are drawn in blue (red) lines. The three sub-regions (Tracker Barrel with PS modules (TBPS), Tracker Barrel with 2S modules (TB2S) and Tracker Endcap Double-Disks (TEDD)) are indicated. [76]	39
Figure 2.7	Block diagram for the Front-End (left) and Back-End (right) electronic system of a 2S module. [76]	40

Figure 2.8	Simplified layout of the Phase II Level-1 (L_1) trigger. The following sub-detectors are labelled in the diagram: the Outer Tracker (TRK), the Endcap Calorimeter (EC), the ECAL Barrel (EB), the HCAL Barrel (HB), the HCAL Forward detector (HF), the Muon Drift Tube Detectors (DT), the Resistive Plate Chambers (RPC), the Cathode Strip Chambers (CSC) and the Gas Electron Multiplier Chambers (GEM). [77] . . .	41
Figure 2.9	Data-flow for a single trigger tower in the Associative Memory (AM)+Field Programmable Gate Array (FPGA) trigger architecture. Data from the trigger tower are time-multiplexed and transmitted to the AM+FPGA processing boards. The pattern recognition tasks is performed by the AM chips, while the FPGA takes care of communication and downstream processing. [82]	48
Figure 2.10	The Tracklet track finding algorithm. (a) Tracklets are formed using pair of stubs (red stars) in different tracking layers. (b) Roads are extrapolated to remaining tracking stations. (c) Only stubs compatible with the trajectory (green stars) are kept. [83]	49
Figure 2.11	(a) Single muon trigger efficiency with a $p_T = 20$ GeV threshold, as function of the generated muon transverse momentum. (b) Single muon trigger rates as a function of the p_T threshold. Results obtained with a standalone muon trigger (red) are compared with a matched muon-track trigger (black). [46]	50
Figure 2.12	(a) Expected efficiency for a single electron trigger with a threshold of 20 GeV in the barrel region for different scenarios: calorimeter only (red), calorimeter photon tuned trigger (green), and calorimeter matched to the track (blue). Also, current trigger (black) performance are shown. (b) Expected trigger rate of single electron calorimeter trigger as a function of trigger threshold, for minimum bias events. [77] . . .	51

Figure 3.1	Total cost as a function of number of units for Application Specific Integrated Circuit (ASIC) (red) and FPGA (black) based systems. In dashed lines are shown the expected changes due to future FPGA developments. Use of ASICs will become advisable only for very large systems. [86]	54
Figure 3.2	Evolution in time of Xilinx FPGA characteristics. [86]	54
Figure 3.3	Schematic comparison between a regionally organised trigger (a) and time multiplexed trigger (b). Each color in the figures represent a set of data belonging to a single bunch crossing. In a regionally organised trigger data from the Trigger Primitive Generators (TPG) are transmitted to regional processors, which require sharing of data between them to handle boundaries. Results are later merged and a final decision is taken by a global trigger. In a time multiplexed trigger data from all TPGs are transmitted to a single processing node for each Bunch Crossing (BX). A multiplexing stage is therefore required and can be a serial interconnection from each TPG to each Time Multiplexed Trigger (TMT) processor. [90]	56
Figure 3.4	High-level view of the Stage-2 trigger architecture.[89]	57
Figure 3.5	Illustration of the data flow from p_T -module front end through the off-detector electronics, which forms the L_1 trigger decision [99]	58
Figure 3.6	(a) Illustration of the DTC cabled octants, also referred as detector octant. Data from each detector octant is readout by 32 Data, Trigger and Control (DTC) boards. Since the outer tracker does not present an eight-fold symmetry in the $r - \phi$ plane, detector octants do not present uniform boundaries. (b) Processing octants divides the tracker into eight equal-size 45 degree sectors in ϕ . Their boundaries are rotated approximately 22.5 degrees with respect to the detector octant.	59

Figure 3.7	The baseline Time Multiplexed Track Trigger (TMTT) system architecture. Time-multiplexing and duplication of stubs at the boundaries are performed in the DTCs, which transmit data to the Track Finder Processors (TFPs). With a baseline time multiplexed period of 18, a total of 144 TFPs are required to build the entire track finding system. [99]	60
Figure 3.8	The TMTT track finding algorithm chain. Input stubs from the DTCs are re-formatted and assigned to finer sub-sectors by the Geometric Processor (GP). An Hough Transform (HT) is then applied to each sub-sectors to find track candidates in the $r - \phi$ plane. The Track Filter and Fitter (TFF) cleans the candidates and computes the track parameters. Finally a Duplicate Removal (DR) rejects duplicate tracks produced by the HT. [99]	62
Figure 3.9	The chosen tracker segmentation on the $r-\phi$ (top) and $r-z$ (bottom) planes. The numbered areas in white are the regions associated to a single sector, whereas coloured regions delineate the overlaps between neighbouring sub-sectors. The dashed lines illustrate the two cylinders of radius $T = 58$ cm and $S = 50$ cm. [99]	63
Figure 3.10	Illustration of the track and stub parameters in the $r-\phi$ plane.	65
Figure 3.11	Illustration of the stub local parameters and their relation with the global track parameters in the $r-\phi$ plane.	67
Figure 3.12	(a) Number of stubs assigned to each (η, ϕ) sub-sector, in top-quark pair production events with pileup 200. (b) Number of sub-sectors each stub is assigned to. Stubs assigned to sectors without (with) the use of the bend information are shown with a red (blue) line. Usage of the bend information reduces significantly the rate of duplication of stubs between sectors.	68

Figure 3.13	The Hough Transform track finding algorithm. Left: Trajectory of a single particle in a quarter of the tracker barrel in the x - y plane. The numbers label the stubs produced by the passage of the particles through the tracking stations. Middle: The produced stubs draw straight lines in the HT space, where the axes correspond to the track parameters ($q/p_T/\phi_T$). Intersection point identifies a track candidate. Right: The same Hough-space discretised with an array of 32×64 cells. Stubs fill a cell only if their local- p_T estimation is compatible with the bin momentum. [95]	69
Figure 3.14	Number of stubs belonging to track candidates at the output of the HT , per each sub-sector. A data reduction of about 90% is achieved.	70
Figure 3.15	Fraction of stubs in a HT track candidate that have been produced by the same particle. A large portion of HT candidates still contain spurious stubs from pileup interactions. A filtering stage is therefore needed to clean the tracks before fitting.	71
Figure 3.16	A Kalman Filter (KF) example procedure for a track candidate in the barrel, shown in the r - z plane. Each step in the fitting procedure is represented by a line segment in the plot. The fitting line is updated at each iteration with increasing radius. Dashed track segments highlight the different results fitting with stub 2a or 2b. State produced with latter is rejected after being extrapolated to stub 4. [99] . . .	73
Figure 3.17	Formation of duplicates in the HT . The yellow cell represents the genuine candidates and the green cells the duplicate tracks produced by the same set of stubs. [99]	74

Figure 3.18	An example Seed Filter (SF) procedure for a HT track candidate in the barrel sub-sector 1. (a) Collecting seeding stubs (blue dots). (b) Creating the seeds (grey lines) (c) Only seeds compatible with a track originating from the luminous region (red bar) and the sector definition. (d) Surviving seeds are extrapolated to remaining tracking layers. (e) Only the closest stubs compatible with seeding lines are kept (green dots). (f) Keep only stubs belonging to the seed with the most activated layers.	77
Figure 3.19	Fraction of stubs in a track candidate that have been produced by the same particle before and after Seed Filter. The SF shows an excellent cleaning power, with about 91% of the surviving tracks having stubs produced by a single particle.	81
Figure 3.20	Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with Pile-Up (PU) ₂₀₀ , before (red) and after (blue) the Seed Filter stage.	81
Figure 3.21	Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU ₂₀₀ , using a High Efficiency Simple Linear Regression (SLR) (blue line), a High Purity SLR (black line) or a Best-of-Both SLR fitter.	82
Figure 3.22	Track finding <i>perfect</i> efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU ₂₀₀ , using a High Efficiency SLR (blue line), a High Purity SLR (black line) or a Best-of-Both SLR fitter.	82
Figure 3.23	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. Results obtained with the three different SLR fitter versions are compared.	84

Figure 3.24	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with the three different SLR fitter versions are compared.	85
Figure 3.25	χ^2 distributions of reconstructed tracks matched (unmatched) with a truth particle, in top pair production events with PU200	86
Figure 3.26	Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU200 . Results obtained with Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).	87
Figure 3.27	Track finding <i>perfect</i> efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU200 . Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).	88
Figure 3.28	Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with different pileup contents (0, 140, 200). These results have been obtained using the SF+Linear Regression (LR) track fitter.	88
Figure 3.29	Distribution of the number of reconstructed tracks using a track finding algorithm with a Seed Filter plus Linear Regression (SF+LR) fitter, for top pair production events with different pileup contents (0, 140, 200).	89
Figure 3.30	Track finding efficiency to reconstruct electrons (blue line) and muons (red lines) in top pair production events with pileup 200, as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b).	90

Figure 3.31	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).	91
Figure 3.32	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).	92
Figure 3.33	Efficiency to reconstruct the primary vertex of the hard interaction in top pair production events with 200 PU, using tracks produced with a SF+LR (blue) and a KF based track finding algorithm.	94
Figure 3.34	Fraction of fake tracks produced in top pair production events at a pileup of 200 by the SF+LR (blue) and the KF (red) track finding algorithms.	95
Figure 3.35	(a) Number of reconstructed tracks in $t\bar{t}$ events with 200 pileup. (b) Track finding efficiency as a function of the inverse transverse momentum $1/p_T$ of the simulated track. The blue line shows the obtained distributions using a modified version of the track finding algorithm, designed to reconstruct also tracks with $2\text{ GeV} < p_T < 3\text{ GeV}$. As a reference, also results obtained with the baseline p_T cut are shown (red line).	96
Figure 3.36	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. The blue line shows the obtained distributions using a modified version of the track finding algorithm, designed to reconstruct also tracks with $2\text{ GeV} < p_T < 3\text{ GeV}$. As a reference, also results obtained with the baseline p_T cut are shown (red line).	98

Figure 3.37	(a) Number of stubs per (η, ϕ) sub-sector and (b) number of track candidates per HT array, in $t\bar{t}$ events at a pileup of 200. The blue line represents the results obtained with novel tracker geometry with tilted module in the barrel region, which have been compared with the old flat barrel geometry (red line).	99
Figure 3.38	Track Finding efficiency as a function the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in $t\bar{t}$ events at a pileup of 200. The blue line represents the results obtained with novel tracker geometry with tilted module in the barrel region, which have been compared with the old flat barrel geometry (red line).	99
Figure 3.39	Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with a tilted barrel geometry (blue line) are compared with that from a flat barrel geometry (red line).	100
Figure 4.1	The Imperial Master Processor MP7 board [97].	104
Figure 4.2	The structure of the MP7 firmware. The algorithm block is abstracted from the physical hardware and service tasks. [97]	105
Figure 4.3	The TMTT microTCA hardware demonstrator crate installed at CERN TIF. The crate is equipped with 11 MP7-XE boards, an AMC13, MCH and required optics. [99]	106
Figure 4.4	Block diagram of the demonstrator slice. Each block in the diagram represents a separate MP7 board, running a separate step in the track finding algorithm. The whole system is made of eight MP7s: two Source, a Geometric Processor (GP), two Hough Transform (HT), two Track Filter and Fitter (TFF) plus Duplicate Removal (DR) and a Sink. [99] . . .	108

Figure 4.5	The demonstrator data taking procedure. Stub and track data produced with a simulated Phase-2 CMS tracker geometry (CMSSW MC Simulation) are digitised and converted into a specific MP7 txt format by the Pattern Writer (PW), which runs also a digitised version of the track finding algorithm (CMSSW Emulation). Txt files are then injected via IPBus [106] into the MP7 Hardware Demonstrator, running the firmware version of the track finding algorithm. Hardware results are then captured via IPBus and reconverted into a CMSSW format by the Unpacker. Finally, hardware and emulation results are compared by the Comparison Software.	108
Figure 4.6	Overview of the entire TFP firmware, showing the main logic components and connections. Different colours highlights the various parts of the chain: GP in yellow, HT in orange and TFF+DR in red. [99]	111
Figure 4.7	The Geometric Processor (GP) routing network. Stubs coming from the DTCs are assigned to 36 finer sub-sectors in (η, ϕ) . The sorting happens through a three-layers router. Stubs are initially grouped in six rough bins in η , then sorted in three finer η bins and finally assigned to the correspondent sector according to their ϕ	112
Figure 4.8	Average number of stubs per η (a) and ϕ sectors. Results obtained with the hardware demonstrator (blue dots) are compared with expectations from the software emulation (red lines). An excellent matching has been observed, greater than 99.9%	113
Figure 4.9	Overview of the firmware implementation of one HT array, assigned to an individual sub-sector. Stubs are propagated to 12 HT blocks, each containing a <i>Book Keeper</i> logic connected to a daisy chain of 2 or 3 <i>Columns (Col.)</i> . [99]	114

Figure 4.10	Logic components within a HT Column, corresponding to a single q/p_T column in the array. Input stubs (<i>stubs</i> in the picture) and stubs consistent with a track candidate (<i>track candidates</i> in the figure) are propagated from Column to Column. [99]	116
Figure 4.11	Distribution of track candidates as a function of the HT track parameters q/p_T (a) and ϕ_T (b). Results obtained with the hardware demonstrator (blue dots) are compared with expectations from the software emulation (red lines). An excellent matching has been observed, greater than 99.4%. . .	118
Figure 4.12	Overview of the Seed Filter plus Linear Regression (SF+LR) firmware. Stubs coming from the HT output channels are temporarily stored in input FIFOs (Track Locators). A Control Unit object readouts the tracks from the Track Locators and distributes them to the SF+LR blocks. To optimise the parallelisation of the firmware, each Control Unit manages six Track Locators and four SF+LRs (Track Fitter Unit). A total of six Track Fitter Units have been implemented on a Virtex-7 FPGA. Produced tracks are then transmitted to a Duplicate Removal (DR) Router, which forwards them to 18 DR modules that reject the duplicate candidates. Surviving tracks are finally output through 18 channels.	120
Figure 4.13	A Track Locator logic component. The module is provided with two BRAM buffers (<i>Stub Buffer A/B</i>) to store stubs from two consecutive events. The arrival of a not-valid stub into the <i>Event Switcher</i> block flags the end of the event. The Event Switcher will, therefore, switch the enabling and disabling of the read/write signals in the two buffers.	121
Figure 4.14	Overview of the Control Unit module. It has been designed as a simple Finite State Machine (FSM), which activates and deactivates the readout from the six Track Locators. Group of stubs with the same HT parameters (m_{bin}, c_{bin}) are transmitted to the four SF+LR modules in a round-robin fashion. .	123

Figure 4.15	Block diagram of the Seed Filter plus Linear Regression logic. The design can be split in four stages, each representing a different step in the algorithm. Each stage has to wait the arrival of the <i>Free</i> signal only from the successive block, before transmitting/receiving new data.	124
Figure 4.16	Overview of the Seed Filter plus Linear Regression design. Internal components are shown as round boxes, FSMs as squared boxes, FIFOs as blue boxes and BRAM as green boxes. The design can be seen split into four phases, resembling the steps of the algorithm. A detailed description is provided in the text.	126
Figure 4.17	Overview of the Seed Finder module. Each Seed Finder receives in input a copy of the stubs which are stored in <i>Stub Buffer</i> . By means of a FSM it creates pairs of stubs (<i>seed</i>) belonging to PS modules in different stations of the Outer Tracker. The first seeding stub (<i>Stub0</i>) is always the stub at the address <i>i</i> in the Stub Buffer, where <i>i</i> corresponds to the index of the Seed Finder. If <i>Stub0</i> is actually a seeding stub, the module loops over the buffer starting from address <i>i</i> + 1, looking for other seeding stub candidates (<i>Stub1</i>). Pairs of seeding stubs are then stored in a FIFO (<i>Seed FIFO</i>), read-out by the top module. A maximum of three seeds can be stored in the FIFO per Seed Finder.	127
Figure 4.18	Overview of Simple Linear Regression module. The design consists of a state machine that deals the communications between the different components, running each an individual step in the fitting algorithm. More details are provided in the text.	130
Figure 4.19	Floor planning of the implemented SF+LR+DR design in a Virtex-7 690 FPGA [84]. Each colour represents a separate SF+LR module. The fitters are spread along the entire chip to optimise the routing from the Track Locators.	131
Figure 4.20	Architecture of a single Duplicate Removal (DR) block, processing tracks of a single (η, ϕ) sub-sector (<i>Input</i>). Therefore 36 DR modules are needed to process the entire octant. . . .	132

Figure 4.21	(a) Number of fitted tracks and (b) track finding efficiency, in top pair production events with 200 pileup. The plots show the effect of truncation due to the limited size of the input stub packet (blue dots), compared with performance obtained with an limitless packet (red dots). Those results have been obtained in simulation.	134
Figure 4.22	Distribution of track candidate as a function of the H_T indices (a, b) and the fitted helix parameters (c, d, e, f), at the output of the Track Finder Processor (TFP), for $t\bar{t}$ events with PU200. Results obtained with the hardware demonstrator (dots) are compared with expectations from software emulation (blue line). A 95% matching has been observed. .	135
Figure 4.23	Track Finding efficiency in $t\bar{t}$ events at a pileup of 200, as a function of the simulated track pseudorapidity (a) and transverse momentum p_T , measured in hardware (blue dots) and software emulation (red dots). Comparable efficiency have been obtained, with differences due to not-replicable truncation effects and in the use of floating point calculations in some parts of the emulator.	136
Figure 4.24	Track helix parameter resolutions as a function of the simulated track pseudorapidity η , in $t\bar{t}$ events at a pileup of 200, as measured in hardware (blue dots) and software emulation (red dots).	138
Figure 4.25	Track helix parameter resolutions as a function of the simulated track transverse momentum p_T , in $t\bar{t}$ events at a pileup of 200, as measured in hardware (blue dots) and software emulation (red dots).	139
Figure 4.26	Resolution of the r-z helix parameters, for different widths of the stub coordinates (r_T, z), in $t\bar{t}$ events at a pileup of 200. Performance comparable to a full floating point simulation (see Fig. 3.32) can be obtained simply by assigning two extra bits to each variable. Those results have been obtained using the software emulation.	140

Figure 4.27	Time distribution of tracks along the output packet, in top pair production events with pileup 200. Majority of the tracks are transmitted out at two times. The first block is sent out at the beginning of the packet, corresponding to <i>consistent</i> tracks. The second group will arrive at about 600 ns, corresponding to the moment in which the second pass of the <i>DR</i> algorithm takes place.	141
Figure 5.1	Vertex reconstruction performance with a FastHisto clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a <i>PU</i> of 200, for different values of the assumed vertex width d . [46, 113]	151
Figure 5.2	Vertex reconstruction performance with a Single-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a <i>PU</i> of 200, for different values of the assumed maximum gap between tracks in the same cluster.	153
Figure 5.3	Width of track clusters produced with the Single-Link (red line) and the Complete-Link (blue line) algorithms. The Complete-Link tends to create more compact clusters, getting rid of the so-called <i>chaining-effect</i>	153
Figure 5.4	Vertex reconstruction performance with a Complete-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a <i>PU</i> of 200, for different values of the assumed minimum gap between tracks in the two neighbour clusters.	154
Figure 5.5	Vertex reconstruction performance with a Average-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a <i>PU</i> of 200, for different values of the assumed minimum gap between centres of two neighbour clusters.	155

Figure 5.6	Vertex reconstruction performance with a Principal Vertex Reconstructor (PVR) clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum gap between a track and the centre of a cluster.	156
Figure 5.7	Vertex reconstruction performance with a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum distance ϵ of a track from the core track in a cluster.	158
Figure 5.8	Vertex reconstruction performance with a DBSCAN clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum transverse momentum p_T^{min} of density tracks M	158
Figure 5.9	Vertex reconstruction performance with a DBSCAN clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum number of density tracks M	159
Figure 5.10	Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum distance of a track from the centre of the cluster ϵ	160

Figure 5.11	Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the pre-defined number of partitions K	161
Figure 5.12	Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the total number of iterations N_{it}	162
Figure 5.13	(a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200. The results obtained with best configuration of each of the investigated algorithms are compared.	163
Figure 5.14	(a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events at a PU of 200. The results have been obtained with an optimised DBSCAN algorithm.	165
Figure 5.15	(a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events at a PU of 200. The results have been obtained with an optimised DBSCAN algorithm.	166
Figure 5.16	(a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events with different PU contents. The results have been obtained with an optimised DBSCAN algorithm.	167

Figure 5.17	(a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events with different pileup contents. The results have been obtained with an optimised DBSCAN algorithm.	168
Figure 5.18	(a) Number of reconstructed primary vertices in top pair production events with different pileup contents. (b) Distribution of transverse momentum p_T for generated tracks belonging to the hard $t\bar{t}$ interaction in red or to pileup in blue. The results have been obtained with an optimised DBSCAN algorithm.	168
Figure 5.19	(a) Relative resolution of the total vertex transverse momentum as a function of the generated one. (b) Trigger efficiency for selecting signals with $Gen \sum p_T > 100$ GeV, as a function of the cut applied on the reconstructed vertex total transverse momentum. The results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200. . . .	170
Figure 5.20	The background rejection vs. signal efficiency for an inclusive vertex transverse momentum trigger, in $t\bar{t}$ events with PU200, for three different signal definitions ($Gen \sum p_T > 50, 100$ and 200 GeV). For clarity the origin on the x axis has been suppressed.	171
Figure 5.21	Distribution of the missing transverse momentum reconstructed from the calorimeter trigger towers (a) and from the L1 tracks (b), in minimum bias events with an average of 70, 140 or 200 pileup. [113]	172
Figure 5.22	(a) Relative resolution of the tracker-based missing transverse energy as a function of the generated one. (b) Efficiency of a TrkMET-based trigger with $TrkMET > 50$ GeV, as a function of the simulated vertex TrkMET. These results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200.	173

Figure 5.23	The background rejection vs. signal efficiency for an inclusive TkMET trigger, in $t\bar{t}$ events with PU200, for three different signal definitions ($\text{GenTkMET} > 50, 100$ and 200 GeV). For clarity the origin on the x axis has been suppressed.	174
Figure 5.24	(a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.	175
Figure 5.25	(a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.	175
Figure 5.26	(a) Relative resolution of the total vertex transverse momentum as a function of the generated one. (b) Trigger efficiency for selecting signals with $\text{Gen} \sum p_T > 100$ GeV, as a function of the cut applied on the reconstructed vertex total transverse momentum. The results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200, using tracks with different p_T thresholds.	176
Figure 5.27	(a) Relative resolution of the tracker-based missing transverse energy as a function of the generated one. (b) Efficiency of a TrkMET-based trigger with $\text{TrkMET} > 50$ GeV, as a function of the simulated vertex TrkMET. These results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200, using tracks with different p_T thresholds.	177

Figure 5.28	(a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} . (b) Resolution of the primary vertex longitudinal impact parameter as a function of z_{Gen} . These results have been obtained with an optimised DBSCAN algorithm applied to different signal events at a PU of 200.	179
Figure 5.29	(a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events at a pileup of 200. These results have been obtained with an optimised DBSCAN algorithm and the latest available tracker geometry with tilted modules in the barrel region.	180
Figure 5.30	(a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.	181

LIST OF TABLES

Table 2.1	Main parameters of the 2S module and the PS module of the CMS tracker. [74]	37
Table 2.2	Definitions of the objects to be used to estimate the L1 Track Finding performance.	46
Table 3.1	Simulated track finding performance in top pair production events at a PU of 200, for the three investigated linear regression fitter methods.	83
Table 3.2	Simulated track finding performance in top pair production events at a PU of 200, after each stage of the algorithm. The average number of reconstructed tracks per event in the entire tracker is also shown, together with the average number of fake and duplicate tracks that survived the different filtering processes. For benchmark purposes, also results obtained with a KF full chain are reported.	94
Table 3.3	Simulated track Finding performance in top pair production events with 200 PU, using a novel outer tracker geometry with tilted modules in the barrel. Results obtained with a flat geometry are also presented as a benchmark.	97
Table 4.1	Stub data-format at the input of the Track Finder Processors. Stubs are formatted and time multiplexed by the DTC boards, which encode them in a 48-bit data format.	106
Table 4.2	Stub data-format at the input of the HT boards. Stubs are converted by the GP into an useful 64-bit data-format, which ease the load of the downstream processing stages.	110
Table 4.3	Resource utilisation of each pre-processing block (with 48 needed per TFP) and the entire routing block of the GP as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis.	113

Table 4.4	Stub data-format at the input of the TFF+DR boards. Stubs are still encoded in a 64-bit words, while some information has been discarded (i.e. bend), not being of any use for the final filtering tasks. The HT transmits only stubs consistent with a track candidate, appending to each stub the coordinate of the cell in the array.	116
Table 4.5	Resource utilisation of one Column (in the HT array) and of one entire HT array, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources are shown in parenthesis. The entire TFP needs 36 HT arrays. The resources needed to implement the multiplexer are not included here, but are relatively small in comparison.	117
Table 4.6	Track data-format at the output of the TFP . Track information is encoded in a 96-bit word, containing the four track fit (Φ_T , q/p_T , z_T and $\tan \lambda$) and quality (χ^2) parameters. . .	119
Table 4.7	Resource utilisation of one Track Locator module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 72 Track Locators, split in the hardware demonstrator across two MP7s.	122
Table 4.8	Resource utilisation of a single Control Unit module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 12 Control Units, split in the hardware demonstrator across two MP7s.	123
Table 4.9	Resource utilisation of a Seed Filter plus Linear Regression module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 48 Seed Filters, split equally in the hardware demonstrator across the two MP7s. Also, the resources utilised by the single Seed Finder and Seed Checker components are presented.	130

Table 4.10	Resource utilisation of DR router and of a single DR block, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 36 Duplicate Removal blocks, one per each sub-sector of the processing octant, split equally in the hardware demonstrator across the two MP7s.	133
Table 4.11	Measured latency of the demonstrated components of the track reconstruction chain, including the serialisation/deserialisation (SERDES) and optical transmission delays between each board.	140
Table 4.12	Total resource usage for the demonstrator TFP (with time-multiplexed factor $n=36$), as implemented in the Xilinx Virtex-7 XC7VX690T FPGA [84]. The resources needed to construct a complete TFP correspond to the sum of the numbers from the three rows labelled GP, HT, and KF and DR.	142
Table 4.13	Upper limit on the required FPGA resources for a Track Finder Processor board running at 480 MHz. Also the available resources for a number of Xilinx Ultrascale FPGAs [111] are shown.	143
Table 4.14	Latency table for the baseline system, extrapolated from the existing demonstrator. Note that extrapolation to 480 MHz assumes that additional registers are necessary, meaning that rather than a 50% reduction in latency, a 30% reduction is assumed throughout. This number is based on what was achieved with the GP router at 480 MHz. Two SERDES stages internal to the TFP are also assumed, to cover the worst case scenario (for latency) of three daisy-chained FPGAs.	144
Table 5.1	Vertex reconstruction performance in $t\bar{t}$ events with 200 pileup, for the several tested clustering algorithms. Also the time complexity of each algorithm is reported, as well as the average number of reconstructed vertices (RecoVertices) in the event.	163

INTRODUCTION

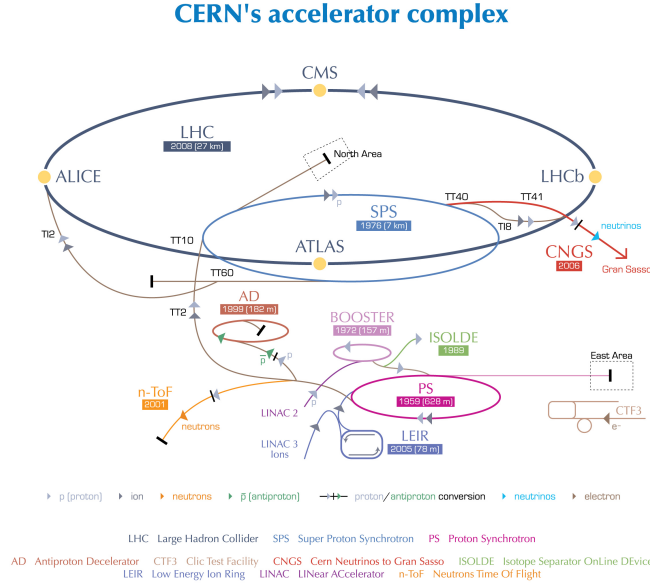
1.1 THE LARGE HADRON COLLIDER

The Large Hadron Collider ([LHC](#)) [1] is the world's most powerful particle accelerator, designed to collide beams of protons and heavy ions in its 27 km long ring of superconducting magnets at a centre-of-mass energy $\sqrt{s} = 14$ TeV with a peak instantaneous luminosity of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$.

Its design parameters represent an impressive improvement, seven-fold increase in energy and a hundred-fold increase in instantaneous luminosity, with respect to previous hadron collider experiments at the Tevatron [2]. The [LHC](#) is located at [CERN](#), 100 m under the Swiss-French border, in the same tunnel that hosted the Large Electron Positron collider ([LEP](#)) [3] until 2000. The [LHC](#) beams cross each other at four collision points where four detectors are installed: the two general purpose experiments [ATLAS](#) (A Toroidal LHC ApparatuS) [4] and [CMS](#) (Compact Muon Solenoid) [5], and the two dedicated experiments [LHCb](#) (Large Hadron Collider beauty) [6], investigating CP violation mechanisms in the B sector, and [ALICE](#) (A Large Ion Collider Experiment) [7], which studies hot [QCD](#) phenomena in heavy-ion collisions.

In order to reach the peak energy of 7 TeV, [LHC](#) beams are pre-accelerated in several stages (Fig. 1.1). The proton beams are formed from hydrogen gas in Linac 2, where they are accelerated up to 50 MeV. Afterwards, they are injected into the Proton Synchrotron Booster ([PSB](#)), where protons are boosted to 1.4 GeV. Then they enter the Proton Synchrotron ([PS](#)), which accelerates them to 25 GeV and separates the protons into bunches spaced in time by 25 ns. Protons enter then the Super Proton Synchrotron ([SPS](#)), where they reach an energy of 450 GeV, before being finally transferred into the [LHC](#).

For any particle physics experiment the available collision energy is the crucial parameter for the production of interesting physical processes. Another important variable certainly is the number of useful interactions (events), especially for rare phenomena with small production cross sections. The rate of events per second



European Organization for Nuclear Research | Organisation européenne pour la recherche nucléaire

© CERN 2008

Figure 1.1: The CERN accelerator complex. [8]

dR/dt depends not only on the production cross section but also on the instantaneous luminosity $\mathcal{L}$, which represents the ability of a particle accelerator to produce the required number of interactions [9],

$$\frac{dR}{dt} = \mathcal{L} \cdot \sigma. \quad (1)$$

For Gaussian-distributed beams containing the same amount of protons per bunch travelling almost at the speed of light, it is possible to write the following equation,

$$\mathcal{L} = f_{\text{coll}} \frac{n_1 n_2}{4\pi\sigma_x\sigma_y}, \quad (2)$$

where $n_{1,2}$ are the number of protons contained into the two bunches, f_{coll} is the bunch frequency and $\sigma_{x,y}$ are the beam sizes along the transverse axes.

LHC operation started on a lower energy of $\sqrt{s} = 7 \text{ TeV}$ on the 30th March 2010, delivering a total of roughly 6 fb^{-1} of 7 TeV collision data to the CMS and ATLAS experiments at the end of 2011. In 2012 the beam energy has been increased

to 4 TeV for an integrated luminosity of around 20 fb^{-1} . After a long technical shut-down (Long Shutdown 1), ended in May 2015, the LHC has reached a centre-of-mass energy of $\sqrt{s} = 13 \text{ TeV}$. At the end of 2016 the CMS experiment has been able to record 45 fb^{-1} of 13 TeV data (Fig. 1.2). The data taking is scheduled to continue till the end of 2018, when another technical shut-down will occur (Long Shutdown 2), in order to upgrade the LHC injectors and perform full maintenance of all the equipment [10].

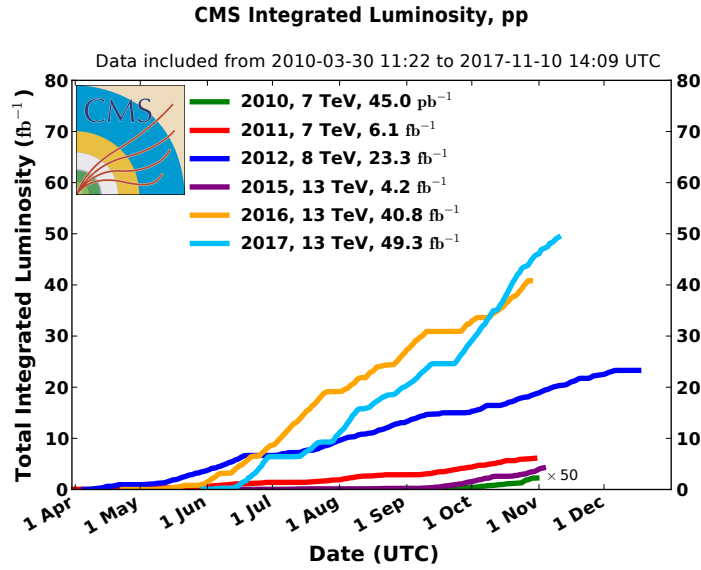


Figure 1.2: Cumulative luminosity versus day delivered to CMS during stable beams and for p-p collisions. This is shown for 2010 (green), 2011 (red), 2012 (blue), 2015 (purple) and 2016 (orange) data-taking. [11]

1.2 PHYSICS AT THE LHC

Thanks to its unique characteristics, the LHC is able to explore a vast programme of physics studies. The main physics motivations for its construction were the testing of the Standard Model (SM), with particular attention to the search for the SM Higgs boson, and searching for Beyond the Standard Model (BSM) physics hints [12].

1.2.1 The Standard Model of Particle Physics

The Standard Model of Particle Physics is the theory that currently provides the best description of particle interactions. The fundamental blocks of the theory are the elementary particles, which can be divided into two main groups depending on their spin. The *fermions* (half-integer spin particles) are the main constituents of matter and interact with each other by means of *bosons* (integer spin), which are the mediators of the fundamental forces.

The SM takes into consideration only three of the four fundamental interactions, not being able to give a qualitative description of gravity, because of its extreme weakness with respect to the other forces and because of the lack of a valid quantum theory for general relativity. The three forces described by the Standard Model are:

- the *electromagnetic* interactions, which act on particles with a non-zero electric charge or with magnetic moments, that move in electromagnetic fields;
- the *weak* force, acting on all particles and characterised by a very low magnitude at low energy scales;
- the *strong* force, which bind the nuclear matter together in the nuclei and is relevant only over a very short distance ($d \ll 1 \text{ fm}$).

Each fundamental interaction is represented by *charge* quantum numbers, and described via the mediation of virtual particles, called *gauge bosons*, which are the quanta of the force fields. Fermions are usually subdivided into two categories: the quarks and the leptons. The quarks are the elementary blocks of nuclear matter and are the only particles in the model to be sensitive to the strong force, while the leptons undergo only the electromagnetic and the weak interactions, depending on their charge. Fermions are also arranged in three generations of matter, where the particles from the first generation form *ordinary* matter. Figure 1.3 lists the elementary particles that compose the Standard Model, outlining the three generations of matter and grouping together leptons, quarks and bosons.

Formally the Standard Model (SM) is a quantum field theory, resulting from the combinations of three previous models: the Quantum Electrodynamics (QED), the Quantum Flavordynamics (QFD) and the Quantum Chromodynamics (QCD).

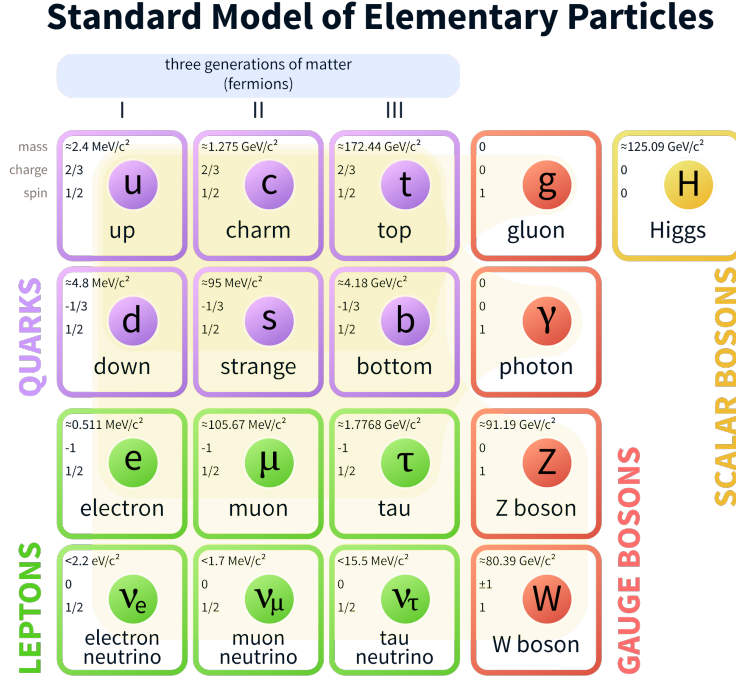


Figure 1.3: The Standard Model (SM) of Elementary Particles. Sourced from [13].

QUANTUM ELECTRODYNAMICS. Quantum Electrodynamics (QED) was the first to be historically developed, thanks to the independent work performed by S. Tomonaga (1946), R. Feynman (1950) and J. Schwinger (1948). [14–16]. The mathematical formulation of QED is the one of an Abelian gauge theory, with the simple unitary group $U(1)$. The quantum of the theory is the photon (γ), which is a massless boson exchanged in the interactions between electrically charged particles.

WEAK INTERACTIONS. Weak interactions have at low energies the smallest scattering and decay amplitude between the forces described in the SM. The first attempt to describe the weak interactions was made by E. Fermi in 1934 [17], when the Italian physicist tried to create a mathematical formulation of the nuclear β decays. He hypothesised a point-like interaction, without the exchange of a mediator particle. This is indeed valid at low energies, but not at a higher scale where such a model leads to a divergence in the cross-section ($\sigma \propto G_F^2 E^2$) and, therefore, to an interaction probability larger than 1.

Hence a better description of the theory involves a non-contact force, mediated by vector bosons, two electrically charged, the W^+ and the W^- with a mass of

~ 80 GeV, and the neutral Z^0 boson ($m_Z \sim 90$ GeV). The existence of these new bosons was verified at the CERN Super Proton–Antiproton Synchrotron (SPPS) in 1983 by the experiments UA(1) and UA(2) [18, 19]. The weak interactions have some unique characteristics, being the only ones that can change the quark *flavours*. Also, it is the only interaction that can give rise to violations of the parity P [20] and the charge-parity CP symmetries [21].

The theory of weak interactions was combined with QED in the so-called *Electroweak* theory by S. Glashow, S. Weinberg and A. Salam in 1980 [22–24]. Here the electromagnetic and the weak forces are treated as two different aspects of the same force, that merge into a single electroweak force once above a critical unification energy of about 246 GeV. This theory is described mathematically by the fundamental representation of the gauge groups $SU(2) \otimes U(1)$.

The presence of massive bosons is an issue for the theory since the gauge invariance of the Lagrangian requires the mediators to be massless. Also, the fermions in the theory are assumed to have zero mass. These issues are solved thanks to the work of Brout, Englert and Higgs in the 60's [25, 26], paved the way for the understanding of the spontaneous symmetry breaking that allows electroweak theory to be compatible with the observed particle content of the SM. In this way the Lagrangian stays invariant under gauge transformations, while its ground state, i.e. the physical vacuum, does not, resulting in a degeneracy that induces scalar fields with non-zero vacuum expectation values [27].

The spontaneous symmetry breaking is obtained by the introduction of a potential density $V(\Phi)$ of a scalar complex field Φ . Three of the real components of this field are transformed into the longitudinal components of $W^\pm$ and Z^0 , giving them masses, while the fourth component, the photon, remain massless. The remaining component of Φ gives rise to the scalar neutral boson, known as the Higgs boson, whose mass is non-zero and it is not predicted by the theory. On the other hand, it is possible to calculate with precision the production cross sections and decay branching fractions of the Higgs boson, once its mass is known (Figure 1.4). The Higgs scalar is also responsible for the fermion masses, which emerge through Yukawa couplings.

QUANTUM CHROMODYNAMICS. Quantum Chromodynamics is the quantum field theory describing the strong interactions [29]. QCD is an exact symmetry under transformation of the gauge group $SU(3)$, meaning that the colour charge is

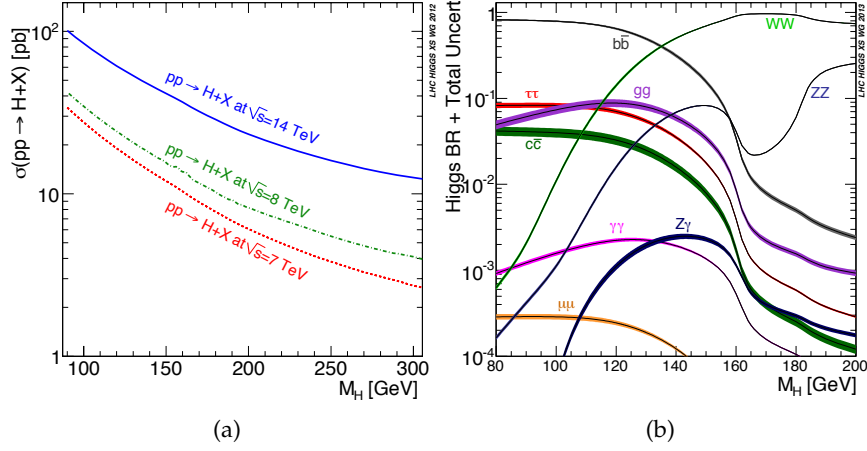


Figure 1.4: (a) Standard Model Higgs boson production total cross sections at $\sqrt{s} = 7, 8$ and 14 TeV. (b) Standard Model Higgs boson decay branching ratios. [28]

always conserved. The group has two fundamental representations, 3 and $\bar{3}$, with three different colours and anti-colours (R, G, B and $\bar{R}, \bar{G}, \bar{B}$). By convention, quarks have a colour charge while anti-quarks have an anti-colour charge. The mediators of the strong interactions are the *gluons*, which are spin-1 bosons and carry colour charge. The *hadrons* are a bound state of coloured quarks and anti-quarks. They are typically divided into two categories: the *baryons*, formed by three quarks or three anti-quarks, and the *mesons*, composite states of a quark and an anti-quark. All hadrons must be *colourless*, meaning that the colour charges of the quarks must form a neutral combination. This is due to the *confinement* property of QCD, which states that only colourless particles can be directly observed, hence no free quark can be detected. Another main characteristic of QCD is the so-called *asymptotic freedom*, which happens at very high energies, where the QCD coupling strength become small enough that quarks will interact weakly, allowing perturbative calculations [30].

1.2.2 LHC results

Experimental proof of the electroweak symmetry breaking mechanism has been a major issue in particle physics for many years, being a fundamental part of the Standard Model theory. Therefore, a main goal of the ATLAS and CMS collaborations has been the search for the Higgs boson. Eventually in 2012, after almost two

years of LHC operations, the two collaborations independently announced the discovery of a new particle with a mass of approximately 125 GeV with characteristics compatible with the Higgs boson [31, 32].

A combined analysis [33], using LHC Run 1 data from both experiments on the $H \rightarrow \gamma\gamma$ and $H \rightarrow ZZ \rightarrow 4l$ decay channels, was published in 2015. The combined mass measurement is $m_H = 125.09 \pm 0.21(\text{stat.}) \pm 0.11(\text{syst.})$ GeV (Figure 1.5).

Another important test of the SM has been conducted by looking at the very rare beauty decay $B_s^0 \rightarrow \mu\mu$. Within the SM, this channel must have a very small branching ratio, occurring just four times for every billion B_s^0 produced. For this reason, any observed difference in the branching fraction with respect to standard model expectations would constitute an evidence of BSM effects. This decay was observed for the first time by the LHCb collaboration in 2012 [34] and later by CMS [35]. The measured branching ratios are in agreement with the standard model expectations, within the current uncertainties.

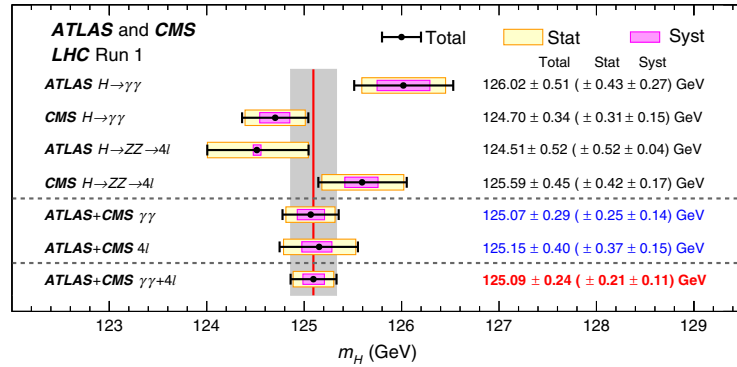


Figure 1.5: Summary of Higgs boson mass measurements from the individual analyses of ATLAS and CMS and from the combined analysis. [33]

1.2.3 Open questions in high-energy physics

Despite describing with extreme precision a large amount of experimental data, the Standard Model does not provide the answer to all the open questions in particle physics. In fact, it is a common belief that the SM is nothing but a low energy effective theory. A short list of issues that have not been addressed by the SM include:

- What is the nature of the dark matter that populates the universe? From cosmological observations [36] we know that the SM describes only a small fraction of the matter in the universe ($\sim 20\%$), while the rest should be constituted by a different kind of matter that interacts weakly with SM fields.
- Why is the Higgs boson so much lighter than the Planck mass (*hierarchy problem*)?
- How do the neutrinos get their masses? Neutrino oscillation experiments have shown that neutrinos indeed have masses [37], which are not predicted by the SM. Adding neutrino mass terms to the model would introduce a series of theoretical problems, i.e. why is there no right-handed neutrino?
- Why is anti-matter mostly absent in the universe (*matter-antimatter asymmetry*)? The Standard Model predicts that matter and anti-matter should have almost the same density in the universe. However cosmological observations clearly show that antimatter is nearly absent [38, 39].

Some of these problems are solved by means of new models known as physics Beyond the Standard Model (BSM). One of the most popular is the SuperSymmetry (SUSY) model, which hypothesises that each SM particle has a super-symmetrical partner particle, called *sparticle*. ATLAS and CMS are both working on direct searches, thus trying to observe the decay of a BSM particle, for new physics and have been already able to place some important limits on several BSM models [40, 41].

1.3 THE HIGH LUMINOSITY LHC UPGRADE

In 2024 physics operations at the LHC will be stopped in order to perform a major upgrade. CERN decided to upgrade the accelerator in order to bring the instantaneous luminosity to a level of $5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$, five times the nominal design value for the LHC [42]. The enhanced collider, formally called the High Luminosity Large Hadron Collider (HL-LHC), will be expected to run for about a decade from 2025, collecting 14 TeV collision data for a total integrated luminosity of $3,000 \text{ fb}^{-1}$.

This upgrade will be necessary in order to achieve the required statistics to maintain scientific progress. In fact, to achieve a reduction of a factor of two of the statistical errors for the measurements, the LHC would have to run for more than

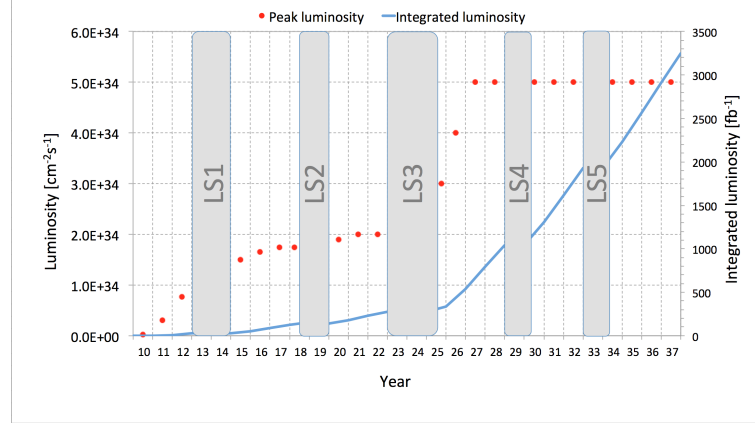


Figure 1.6: Projected LHC performance through 2038, showing preliminary dates for long shut-downs (LS) of the LHC and projected luminosities. Sourced from [42].

ten years from 2020. The same factor can be obtained at the HL-LHC in less than two years (Fig. 1.6).

The HL-LHC installation will take place during the Long Shutdown 3 (2023-2025), in order to minimize the machine downtime (Fig. 1.7), with some modifications planned already for the Long Shutdown 2 (2019-2020). In fact, already after LS2, LHC is expected to reach a peak luminosity of $2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$, delivering an integrated luminosity of over 300 fb^{-1} by 2023. During LS3 the quadrupole magnets that focus the beams at the ATLAS and CMS collision regions will be replaced with new low- β quadrupole triplet. Also, the overlap of the proton bunches will be enhanced by adding crab-cavities at the interaction region. Thanks to these changes, the HL – LHC will be able to deliver up to 250 fb^{-1} of collision data per year for more than 10 years of operation.

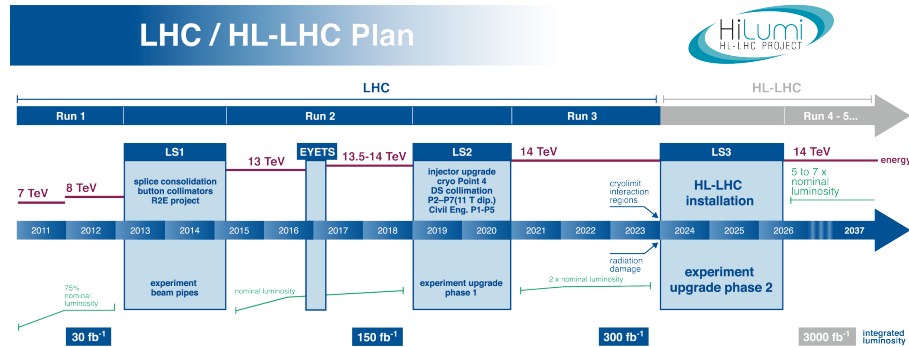


Figure 1.7: LHC/HL-LHC time-scale. [43]

1.3.1 HL-LHC Physics Motivations

Thanks to the increased luminosity, the HL-LHC will be an excellent Higgs factory. It will allow a drastic improvement of the precision of the measurements of the Higgs boson couplings and the search for rare SM and BSM decay channels. Figure 1.8 compares the predicted uncertainties on the Higgs couplings at the beginning and at the end of the HL-LHC data taking period. An improvement about a factor three is expected to be reached for most measurements. Eventually, it will be also possible to observe for the first time di-Higgs production events, which have a very small cross-section (40 fb) and will provide a better understanding of the Higgs boson self-coupling mechanism [44, 45].

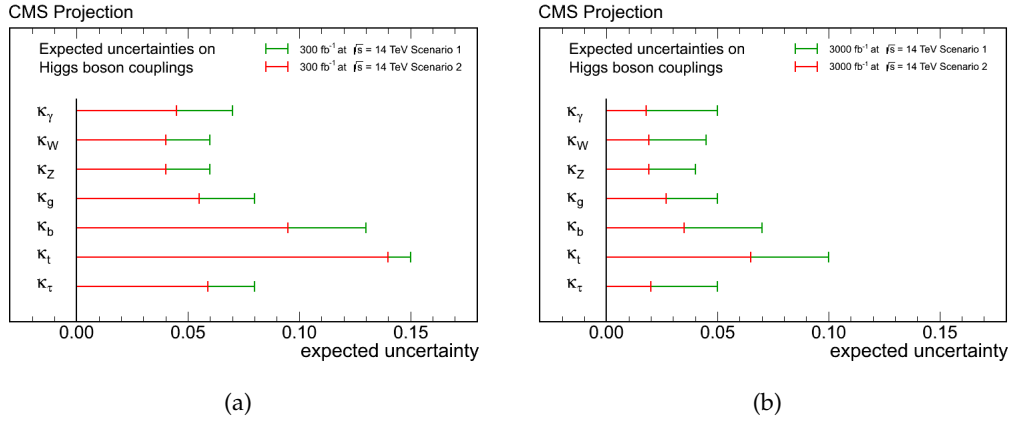


Figure 1.8: Estimated precision of the coupling measurements for a SM-like Higgs boson. The projections assume a centre-of-mass energy of 14 TeV and a data-set with integrated luminosity of 300 (a) and 3,000 fb⁻¹ (b). Scenario 1 assumes that the systematic uncertainties are as today, Scenario 2 that the experimental uncertainties scale as $1/\sqrt{L}$ and the theoretical uncertainties halved [46].

Another crucial study will be the search for new particles at the TeV scale, which have not been observed yet at the LHC. This might be due to the fact that those particles have either masses above the current level of sensitivity or their production cross sections are lower than expected. The increased luminosity of the HL-LHC will improve the sensitivity for searches of new particles. Figure 1.9 shows the new mass scales that the 3,000 fb⁻¹ integrated luminosity will allow the experiments to reach, for some selected SUSY channels.

Other exotica studies will benefit from the increased luminosity. Searches for new heavy gauge bosons with SM couplings will be extended to a mass scale of

6 TeV or more. Also other analyses, like searches for extra dimensions, compositeness and leptoquarks, will see their mass range extended by a few TeV.

Another crucial priority of the HL-LHC physics programme will be the search for evidence of dark matter. This will be done by making use of methods that have been developed during the LHC years, which typically exploit mono-object signatures. An intriguing possibility will also be constituted by novel studies, looking for events where an Higgs boson has been produced in association with a dark matter particle. An increase in integrated luminosity is essential for these studies, because of the small cross sections of their signals.

In addition to searches for new physics, the HL-LHC will improve the precision of many fundamental SM measurements, like the parton distribution functions (PDFs) and inclusive, differential, and double-differential cross sections of events with jets, photons, W and Z bosons, and top quarks.

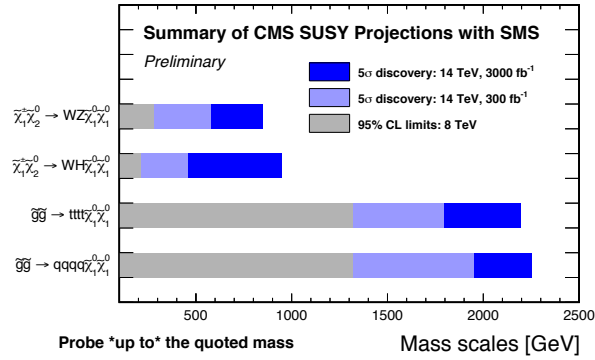


Figure 1.9: Mass reach of searches for super-symmetry based on simplified models [47].

1.3.2 Detector Challenges at the HL-LHC

During LS3, all four collaborations will upgrade their detectors in order to cope with the challenges introduced by the new HL-LHC environment [46, 48–50]. In particular, ATLAS and CMS will have to face a considerable increase in the radiation dose levels, which could lead to detector damage. Another important challenge will be the increased number of overlapping proton-proton per BX (‘pile-up’) caused by the high instantaneous luminosity.

1.3.2.1 Radiation damage

In the process of the detector design, ATLAS and CMS will have to consider the effects of ionising and non-ionising radiation produced by pp collisions at the HL-LHC. Indeed the annual radiation dose that will be delivered to the experiments is expected to be comparable in magnitude with the total dose delivered from the start of the LHC programme to the start of Long Shutdown (LS)₃.

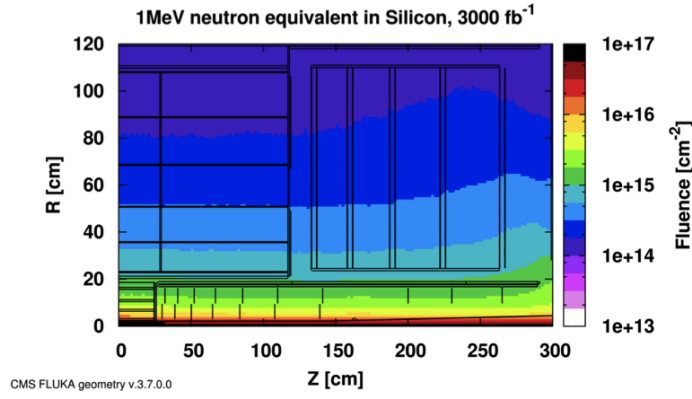


Figure 1.10: Map of the expected particle fluence in the upgraded CMS tracker detector, to an integrated luminosity of $3,000 \text{ fb}^{-1}$, expressed in terms of 1 MeV neutron equivalent fluency [46].

The main source of radiation at the HL-LHC will be constituted by charged particles, typically pions, produced in the collisions. The mean rate of energy loss by ionisation for a mid-relativistic ($0.1 < \beta\gamma < 1,000$) charged particle is described by the well-known *Bethe-Bloch* formula [51],

$$\left\langle -\frac{dE}{dx} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 W_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right], \quad (3)$$

where K is a constant, z the charge number of incident particle, Z and A the atomic number and mass of the absorber material, β is the relativistic velocity, γ the relativistic Lorentz factor and $W_{\max}$ the maximum energy that can be transferred to an electron in a single collision.

Radiation produces different damage depending on the type of detector. The main degradation can be observed in the silicon detectors, since they are usually the closest to the interaction point (Fig. 1.10). Defects can be created in the silicon lattice leading to an increasing leakage currents. The charge collection efficiency

decreases while the full depletion voltage¹ increases, making the detector less sensitive to a crossing ionising particle, hence producing a lower signal [52].

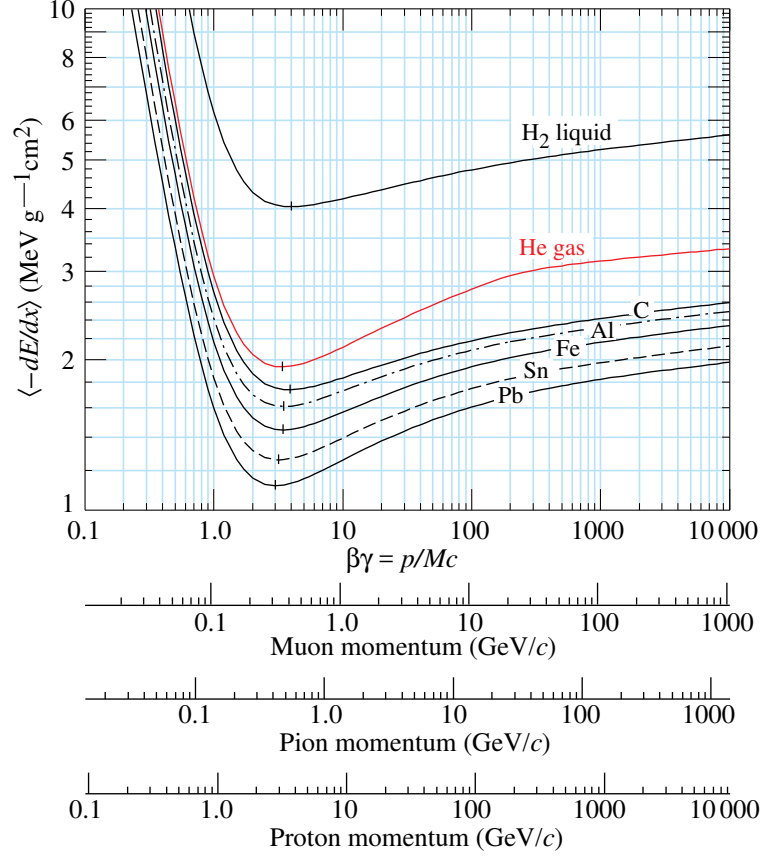


Figure 1.11: Mean energy loss rate in different material for protons, muons and pions as a function of the particle momentum [51].

1.3.2.2 High pile-up

At full operation the HL-LHC will collide bunches of protons every 25 ns with a peak instantaneous luminosity of $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$. A collision between two bunches at the LHC is called a Bunch Crossing (BX). In high-luminosity conditions the average number of pp pile-up interactions will spread between 140 and 200. Most of these will be *soft* collisions, thus not carrying enough transverse momentum to be useful in high-energy physics studies. Only a very small percentage of events can be classified as *hard* interactions, which contain high transverse momentum (p_T) particles.

¹ The depletion voltage is the minimum voltage at which the bulk of the sensor is fully depleted

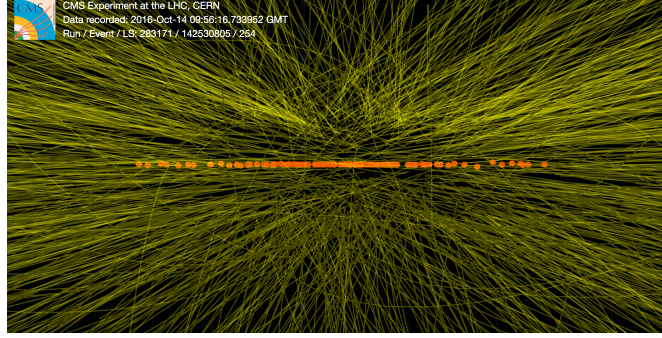


Figure 1.12: High pile-up event recorded at CMS during the high pile-up fill of Run 2 at center-of-mass energy of 13 TeV [53].

Even if soft interactions usually do not carry high p_T particles, sometimes they contain some tracks and energy that superimposed on the actual hard collision. This can lead to misinterpretation in the event selection and reconstruction. Moreover, the increased number of particles produced per BX at the HL-LHC will result in a growth in the amount of data that will have to be processed by the online trigger system and read out for offline analysis.

1.4 THE CMS EXPERIMENT

The Compact Muon Solenoid (CMS) [5] is a general-purpose detector located at the LHC Point 5, 100 m underground close to the French town of Cessy. The detector has been built around one of the four LHC collision points. It has a cylindrical symmetry around the beam axis, with a central part (*barrel*) closed at both ends by two disks (*endcaps*). A schematic view of the detector can be seen in Figure 1.13.

Figure 1.14 shows a slice of the detector in the barrel region. Starting from the interaction point it is possible to list all the different sub-detectors that compose the experiment.

- The **Silicon Tracker** is the closest sub-detector to the interaction points. Made of silicon sensors, it is designed to find the primary and secondary vertices in the collisions and to give precision measurements of the particle transverse momenta.
- The **Electromagnetic Calorimeter** is composed of scintillating PbWO_4 crystals and it is used to measure the energy deposited by electrons and photons.

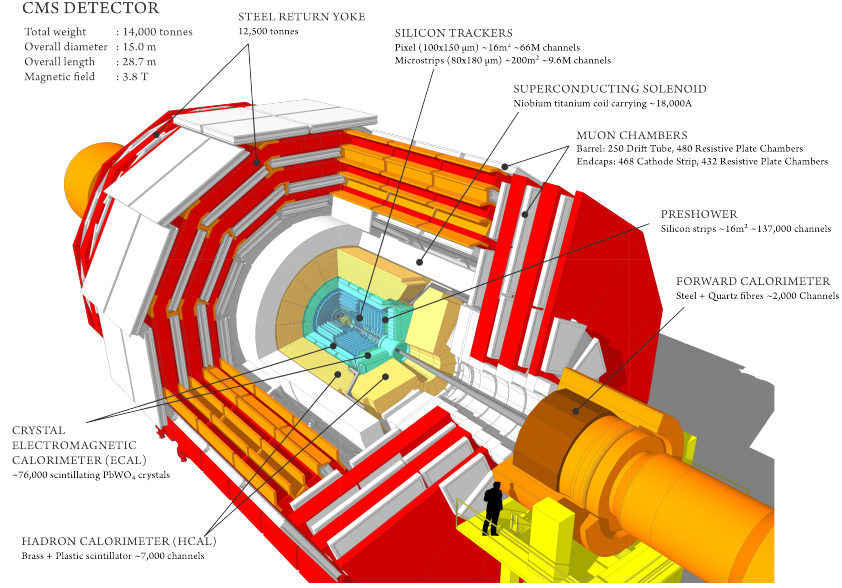


Figure 1.13: Schematic view of the CMS detector. [54]

- The **Hadron Calorimeter** records the energy deposited by hadronic particles in the event.
- The **Superconducting Solenoid** magnet generates an almost uniform magnetic field of 3.8 T along the beam axis, allowing the measurements of the particle momenta in the Silicon Tracker.
- The **Muon Chambers**, located at the edge of the detector, are needed to identify muons and measure their momenta.

1.4.1 Coordinate system

The CMS coordinate system has its centre at the interaction point. The z axis follows the beamline, the x axis points toward the centre of LHC ring, while the vertical y axis points upward.

The cylindrical geometry of CMS makes natural to use polar coordinates:

- $r = \sqrt{x^2 + y^2}$ is the transverse distance from the beam axis z ,
- $\phi = \cot(x/y)$ is the azimuthal angle in the range $[-\pi, +\pi]$,
- $\theta = \cot(r/z)$ is the polar angle measured from the z axis in the range $[0, +\pi]$.

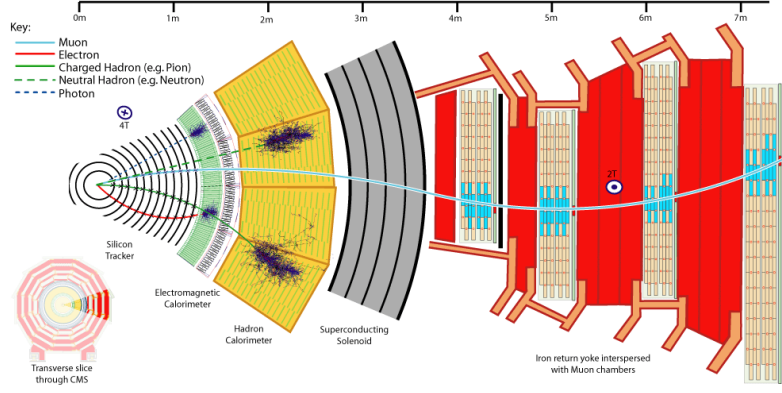


Figure 1.14: Transverse slice of CMS detector in the barrel region. The signatures left by the different particles in the sub-detectors are also shown.

Typically in high energy physics experiments it is more convenient to use the *pseudorapidity* variable η instead of the polar angle θ ,

$$\eta = -\ln[\tan(\theta/2)]. \quad (4)$$

The pseudorapidity could be rewritten in terms of momentum

$$\eta = \frac{1}{2} \ln \left(\frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z} \right), \quad (5)$$

where $\mathbf{p}$ is the particle three-momentum and p_z is the projection on the z axis. By this equation it is easy to see that pseudorapidity coincides with *rapidity* y when the momentum is large, that is when $|\mathbf{p}| \sim E$,

$$y = \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right) \sim \frac{1}{2} \ln \left(\frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z} \right) = \eta \quad (6)$$

The rapidity and the pseudorapidity are very useful variables in particle physics and they are often used in place of the polar angle θ . This is because the particle production is more or less constant with respect to the rapidity, and especially

because differences in rapidity Δy (or pseudorapidity $\Delta \eta$) are Lorentz invariant under boosts along the longitudinal axis. This is fundamental in hadron collisions, where the longitudinal momentum carried by the colliding partons is unknown.

Other useful variables to be defined are the transverse momentum of a particle,

$$p_T = |\mathbf{p}| \cdot \sin \theta, \quad (7)$$

and the *beam spot* length, which is the luminous region around the coordinates origin where the proton bunches collide. The Gaussian widths of this region are 15 cm in z , 16 μm in x and 100 μm in y .

1.4.2 The Silicon Tracker

To meet the various physics requirements CMS has to be able to find the trajectories of charged particles emerging from the collision and to reconstruct all the secondary vertices from particle decays with extreme precision. Those tasks are addressed by the Silicon Tracker system [55].

The tracker is the innermost sub-detector of the experiment, surrounding the beam pipe, with an area of $2.5 \times 5.8\text{m}$ in the r - z plane. On average at the LHC around 20 pile-up collisions take place at the same time, producing $\sim 1,000$ particles per bunch crossing (every 25 ns). These parameters require the tracker to have high granularity and fast response electronics. Unfortunately, this implies a high power density to supply the on-detector electronics, which in turn needs an efficient cooling system. Those requirements are in contrast with the need for a low material mass detector, in order to reduce the effect of particle energy loss due to interactions with matter.

The CMS silicon tracker has been thus built as a compromise for the different necessities. The designed system has a total sensitive area of around 200m^2 , making it the largest silicon tracker ever constructed. Two different silicon module technologies have been adopted. The innermost part of the tracker is formed by pixel sensors distributed along three layers in the barrel ($4.4\text{cm} < r < 10.2\text{cm}$) and two disks in the endcaps. The region up to a radius of 1.1 m is covered by silicon microstrip sensors arranged in 10 layers in the barrel and 3+9 disks in the endcaps. In total the whole tracker constitutes 1440 pixel and 15,148 micro-strip modules.

1.4.2.1 Tracker layout

Figure 1.15 shows a schematic cross-section of the CMS tracker in the r - z plane. Since the average hit rate density in the silicon tracker area spreads from 1 MHz/mm² in the innermost pixel barrel layer to 3 kHz/mm² in the outermost micro-strip barrel layer, the detector requires a high granularity in order to keep the occupancy below 1% per sensor per BX. This low occupancy is needed to resolve the high track density, with an average of 1,000 tracks being produced at each LHC collision. This is achieved by adopting different kinds of silicon modules.

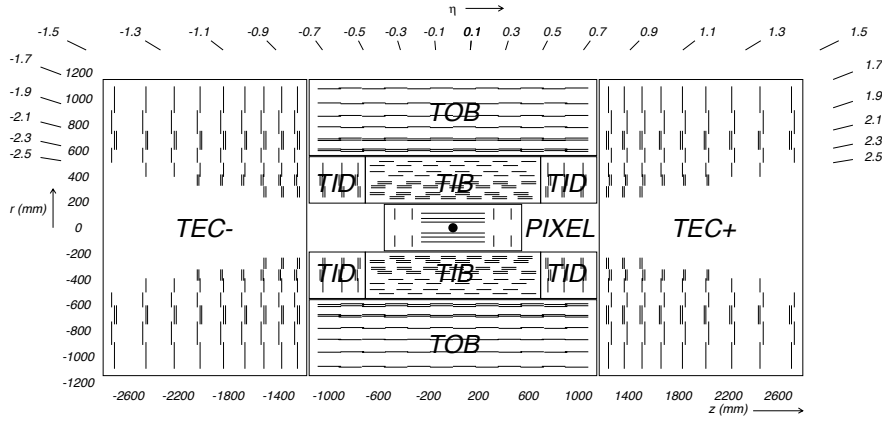


Figure 1.15: Layout of the CMS silicon tracker. Each line represents a silicon module with double lines indicating back-to-back modules which deliver stereo hits. [5]

PIXEL DETECTOR. The innermost detector is built using pixellated silicon sensors with a size of $100 \times 150 \mu\text{m}^2$ per pixel, segmented in the r - Φ plane and along the z axis, which leads to an average occupancy per sensor per BX of around 10^{-4} . The excellent track resolution on both planes is required for an optimal 3D vertex reconstruction, essential for secondary vertices with low track multiplicities.

The pixel detector constitutes three layers in the barrel (BPix) and two endcap disks (FPix), covering the pseudorapidity range $|\eta| < 2.5$. The BPix layers are located in radius of 4.4, 7.3 and 10.2 cm and are 53 cm long in z . The two FPix disks are located at $z = \pm 34.5$ and $z = \pm 46.5$ cm and have a length of 6 and 15 cm in radius.

Since the pixel detector is the closest to the interaction point, it is the system most subject to the radiation damage. For this reason, the CMS collaboration de-

cided to replace the system in the end-of-year shut-down during winter 2016/2017 [56].

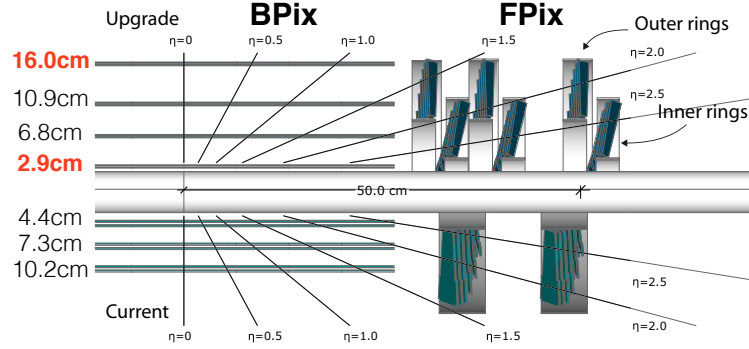


Figure 1.16: Comparison of geometrical layouts of the old (bottom) and upgraded (top) CMS pixel detectors [56].

Figure 1.16 shows a comparison between the old and the upgraded pixel detector. The new system is now composed of four barrel layers, located at radii of 2.9, 6.8, 10.9, 16.0 cm, and three endcap disks at a distance from the interaction point of 29.1, 39.6 and 51.6 cm. The total number of pixels increases from 66 million to 124 million, while at the same time the material budget is reduced. The installation of the new innermost barrel layer has been possible thanks to replacement of the beampipe in 2014, which has a smaller diameter of 45 mm.

Both primary vertex resolution and b-tagging efficiency have improved thanks to the new layer closer to the interaction point (Figure 1.17). Moreover the additional tracking station reduces the combinatorics in track reconstruction, enhancing the overall tracking performance.

MICRO-STRIP DETECTOR. The region between 20 and 110 cm is covered by micro-strip silicon sensors. The detector is composed of three different subsystems.

The Tracker Inner Barrel and Disks (**TIB/TID**) system occupy the area in the region $20 < r < 55$ cm, $|z| < 118$ cm and are composed of four barrel layers and three disks. Different strip pitches have been adopted for each layer/disk. TIB1-2 have a pitch size of 80 μm , resulting in a single point resolution of 23 μm on the r - Φ plane. TIB3-4 uses modules with pitch size of 120 μm , for an r - Φ single point resolution of 35 μm . In the TID the strip pitch sizes spread from 100 μm to 141 μm .

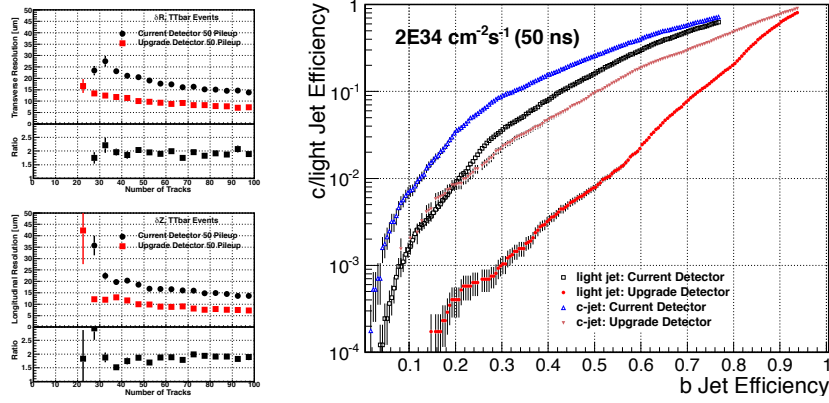


Figure 1.17: (a) Transverse and longitudinal primary vertex resolution as a function of the track multiplicity in the vertex for the old (black) and upgraded (red) pixel detector. (b) b -tagging performance for jets with $p_T > 30$ GeV using the old (open points) and the upgraded pixel detector (solid points). These studies have been conducted using top pair production events with an average pileup of 100. [57]

The region $55 < r < 116$ cm, $|z| < 118$ cm is called Tracker Outer Barrel (TOB) and it is subdivided in six barrel layers. The first four at a smaller radii adopt sensors with a pitch size of $183 \mu\text{m}$, while for the outermost layers it is $122 \mu\text{m}$, leading to single point resolutions of $53 \mu\text{m}$ and $35 \mu\text{m}$ on the r - Φ plane respectively.

Finally the endcap area between $22.5 < r < 113.5$ cm, $124 < |z| < 282$ cm is occupied by the Tracker EndCaps (TEC). These are composed of 18 disks in total, 9 per each endcap, which in turn are formed by up to 7 rings per disk. Again different pitch sizes has been used, from $97 \mu\text{m}$ to $184 \mu\text{m}$.

The micro-strip modules of the tracker also have different thickness, depending on their radial position. This difference in thickness arises from the different levels of radiation fluence that the sensors undergo. Two kinds of radiation damage can occur: bulk effects, caused by the displacement of silicon atoms in the lattice, and surface effects, mainly due to charge trapping at the interface between silicon and oxide. Bulk damage translates into an increase of leakage current, a variation of depletion voltage (sensor type inversion) and a decrease of charge collection. Surface effects mainly cause an increase of the inter-strip capacitance and hence higher noise [58].

The use of a low resistivity material in the region of highest radiation fluence (innermost) offers advantages in terms of depletion voltage. For this reason up to a radius of 55 cm, thin sensors of $320 \mu\text{m}$ thickness have been adopted, with

substrate resistance of $\rho = 1.55\text{--}3.25\text{ k}\Omega\text{ cm}$. On the other hand, at larger radii, sensors of $500\text{ }\mu\text{m}$ thickness and higher resistivity ($\rho = 4\text{--}8\text{ k}\Omega\text{ cm}$) could be adopted, because of the lower radiation level. The increased thickness is necessary in order to increase the signal to compensate for the increased noise caused by the larger strips (load capacitance) [59].

1.4.3 *Superconducting magnet*

A large magnetic field is required in CMS, in order to perform precise measurement of high energy particle momenta. This is provided by a large superconducting solenoid magnet [60], 6.3 m cold bore, 12.5 m length and 220 tons mass, which is installed in the centre of the apparatus. It produces an almost uniform 3.8 T magnetic field in the inner region, containing the tracker and calorimeters.

The magnet conductor is a Rutherford-type NiTi (niobium-titanium) cable co-extruded with pure aluminium. It generates the field with a unique four layer winding system and stores at full current a total energy of 2.6 GJ.

1.4.4 *Electromagnetic calorimeter*

The discovery of the Higgs boson through the ‘golden channel’ $H \rightarrow \gamma\gamma$ would not have been possible without an excellent electromagnetic calorimeter. The CMS Electromagnetic Calorimeter (ECAL) [61] was designed to have very good energy resolution and fine granularity, which allows the separation of two photons and, as a result, to suppress the background constituted by $\pi^0 \rightarrow \gamma\gamma$ decays.

The ECAL is a compact hermetic homogeneous (non-sampling) calorimeter and it is composed of lead tungstate (PbWO_4) crystals, 61200 installed in the central barrel and closed by 7324 other crystals in each of the two endcaps. PbWO_4 has been a perfect choice for operation at the LHC, because of its perfect characteristics. Its high density ($\rho = 8.28\text{ g/cm}^3$), short radiation length ($X_0 = 0.89\text{ cm}$) and small Molière radius ($R_M = 2.2\text{ cm}$) are ideal for the construction of a compact and high-granularity calorimeter. In addition, it shows also a good radiation hardness and it has a fast scintillation decay time, with about 80% of the light emitted in 25 ns.

The scintillation light is emitted in the blue-green wavelengths (420–430 nm) with a relatively low light yield (~ 30 photons per MeV). The light is then collected

by Avalanche Photodiodes (aPDs) in the barrel and by Vacuum Phototriodes (vPTs) in the endcaps.

The ECAL is divided into two part: the ECAL Barrel (EB) and ECAL Endcap (EE), as shown in figure 1.18. The 61200 crystals of the EB are arranged in order to have 360-fold divisions in ϕ and 2×85 -fold divisions in η , covering the pseudorapidity region up to $|\eta| < 1.479$. Crystals are have a truncated pyramid shape, 230 mm long ($25.8 X_0$) with a with a cross-section of $22 \times 22 \text{ mm}^2$ at the front face and $26 \times 26 \text{ mm}^2$ at the rear face, for a resulting granularity equal to 0.0174×0.0174 in η - ϕ . The remaining pseudorapidity region up to $|\eta| < 3.0$ is covered by the EEs. Each endcap is sub-divided in two halves, called *Dees* and made of 3662 crystals. The crystals in the endcaps are arranged in a x-y grid and have all the same shape, with a front face cross section $28.62 \times 28.62 \text{ mm}^2$, a rear face cross section $30 \times 30 \text{ mm}^2$ and a length of 220 mm, which corresponds to $24.7 X_0$.

In front of the EE calorimeter, it is located a pre-shower sampling calorimeter, which has the role to separate neutral pion from the photon showers. It covers the fiducial region $1.653 < |\eta| < 2.6$ and it is made by two alternate layers of lead and silicon strip detectors for a total thickness of 20 cm ($3 X_0$).

The ECAL energy resolution is parametrised with the following formula, valid up to energies of 500 GeV,

$$\left(\frac{\sigma_E}{E}\right) = \left(\frac{a}{\sqrt{E}}\right)^2 + \left(\frac{\sigma_N}{E}\right)^2 + c^2, \quad (8)$$

where a is a stochastic term including the statistical fluctuations of the shower containment and photostatistics contributions; σ_N represents the noise term due to electronic noise and pileup energy and c is a constant term, which includes effects like intercalibration errors or energy loss from the back side of the crystals.

1.4.5 Hadron calorimeter

Behind the ECAL sits the CMS HCAL[62]. The main goal of this scintillating sampling calorimeter is to detect and measure the energy of strongly interacting particles. It is composed of alternating layers of absorber and scintillating materials and it is subdivided into four parts: the Hadron barrel (HB), the Hadron endcap (HE), the HO and the Hadron forward (HF) (figure 1.19).

The **HCAL** is constrained by the inner extent of the magnet, which limits the total amount of material that can be used to absorb the hadronic showers in the barrel region. For this reason, the **HO** has been installed outside the solenoid magnet, in order to detect the tails of the shower. The **HF** covers instead the very high-rapidity region up to $|\eta| \sim 5.2$.

The **HCAL** energy resolution has been measured using charged pion in the energy range 5-300 GeV and is approximately equals to:

$$\frac{\sigma_E}{E} = \frac{84.7\%[\text{GeV}^{1/2}]}{\sqrt{E}} + 7.4\%. \quad (9)$$

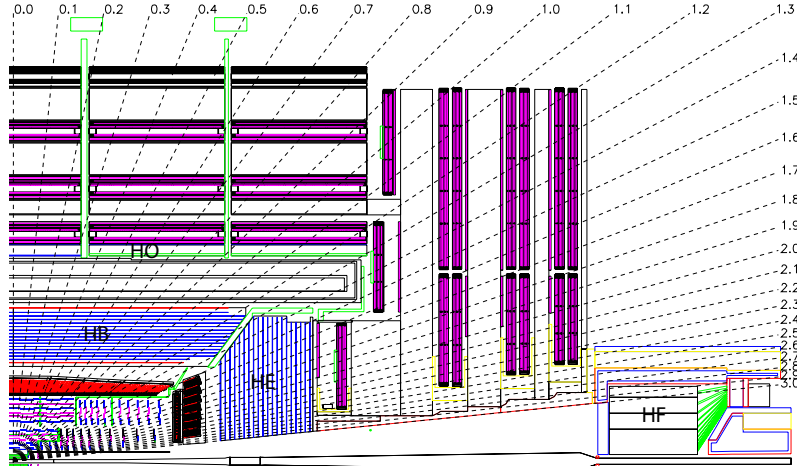


Figure 1.19: Longitudinal view of the CMS detector, highlighting the locations of the **HCAL** sub-detectors. [5]

1.4.5.1 Barrel design (HB)

The sampling calorimeter **HB** covers the barrel region up to $|\eta| < 1.3$. It is divided into two half barrel section (**HB+** and **HB-**), each of them made of 36 identical azimuthal wedges. Each wedge is in turn segmented into four ϕ sectors. The plastic scintillator is also segmented in a η - ϕ grid, with a granularity of 0.087×0.087 , five times larger than the **ECAL**. It is made of about 70,000 tiles of active medium, which are connected to wavelength shifting fibres that collect the emitted light. The absorber consists of 16 steel plates for a total absorber thickness of 5.82 inter-

action lengths (λ_I). The ECAL crystals in front of the HB add an additional $1.1\lambda_I$ of material.

1.4.5.2 Endcap design (HE)

The HE sampling calorimeter covers the pseudorapidity range $1.3 < \eta < 3$, that contains around 34% of the particles produced in the final state. The absorber material of the HE is made from a non-magnetic material (C26000 cartridge brass), since the calorimeter is situated at ends of the solenoid. The calorimeter structure alternates 79 mm-thick brass plates with 9 mm gaps to accommodate the scintillators, for a total length of about $10\lambda_I$. The η - ϕ granularity of the calorimeter varies with η : 0.087×0.087 for $|\eta| < 1.6$ and 0.175×0.175 for $|\eta| \leq 1.6$.

1.4.5.3 Outer design (HO)

Because of the physical constraint constituted by the magnet, it is not possible to contain all the hadronic showers in the HB. For this reason, the HO sampling calorimeter has been installed between the solenoid and the muon chambers, covering the same pseudorapidity region as the HB.

The HB is placed at the first sensitive layers in each of five iron yoke rings (2.536 m wide), necessary to return the magnetic field outside the vacuum tank of the solenoid (figure 1.20). The central ring ($\eta = 0$) has two layers of HO (at radial distances of 3.82 m and 4.07 m, respectively) divided by a 19.5 cm thick piece of iron (the tail catcher iron), because here of the minimal absorber depth in this region. Only a single scintillator layer, placed at a radial distance of 4.07 m is needed in the other rings. Eventually, the minimal length of the calorimeter system is extended to $11.8\lambda_I$.

1.4.5.4 Forward design (HF)

To detect hadron showers in the very high pseudorapidity region ($3 < \eta < 5.2$), CMS exploits the HF calorimeter. Placed outside the muon chambers, it has to survive the highest particle flux. The average deposited energy is in fact 760 GeV per pp interaction, compared to only 100 GeV in the rest of the detector. Hence this sampling calorimeter adopts radiation-hard quartz fibres as active material. The emitted signal, in this case, is the Cherenkov light produced by charged shower

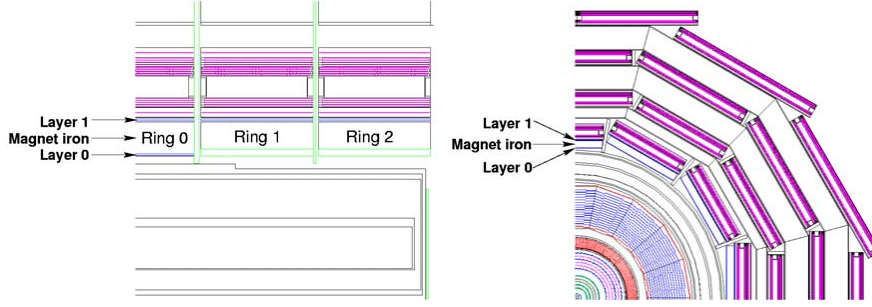


Figure 1.20: Longitudinal and transverse views of the CMS detectors, highlighting the position of the magnet iron ring and the [HO](#) layers. [5]

particles above the Cherenkov threshold ($E > 140 \text{ keV}$ for electrons), which makes the calorimeter sensitive only to the electromagnetic component of the showers.

The [HF](#) has a cylindrical structure with an external radius of 130 cm, where the absorber is made of steel and has a total thickness of 1.65 m. It has a granularity in η - ϕ equal to 0.175×0.175 . Because the [HF](#) provides good precision for energy measurements in the forward region, it has been used for the CMS luminosity measurements.

1.4.6 Muon system

Another golden channel in the search for the [SM](#) Higgs boson is the decay into ZZ or ZZ^* , which in turn decay into four leptons [63]. The best scenario is constituted by the case in which all the four leptons are muons since they are easily detectable and provide the best mass resolution. Therefore it is essential for [CMS](#) to be able to identify precisely the generated muons in the collisions. This task is accomplished by the [CMS](#) muon system [64] that, in addition to the muon identification, must also provide stand-alone measurements of muon momenta. This is allowed by the flux-return yoke of the solenoid magnet. The muon system is also used in the [L1](#) trigger system.

The muon system covers the entire kinematic range of [CMS](#), for a total detector area of about $25,000 \text{ m}^2$. The [CMS](#) collaboration decided to use gaseous detectors to build the muon system since they are inexpensive, reliable and robust. Three kinds of detectors have been installed in the final system.

DRIFT TUBE CHAMBERS In the barrel region ($|\eta| < 1.2$) the system adopts DT chambers with standard rectangular drift cells. They are arranged in four stations (MB) among the layers of the flux return plates (YB) (Figure 1.21). The first four stations consist of eight chambers to measure the position in the bending plane r - ϕ and four chambers to measure the position z along the beam line. The fourth station provides only a measurement in the r - ϕ plane.

CATHODE STRIP CHAMBERS The muon system in the endcap regions suffers from higher muon and background rates, together with a larger and non-uniform magnetic field. Here CMS uses Cathode Strip Chambers (CSCs), which have a fast response time, fine segmentation and are radiation resistant. They provide muon identification for the pseudorapidity region $0.9 < |\eta| < 2.4$ and are arranged in four stations, each of them made by four CSCs, which measure precisely the position on the r - ϕ plane.

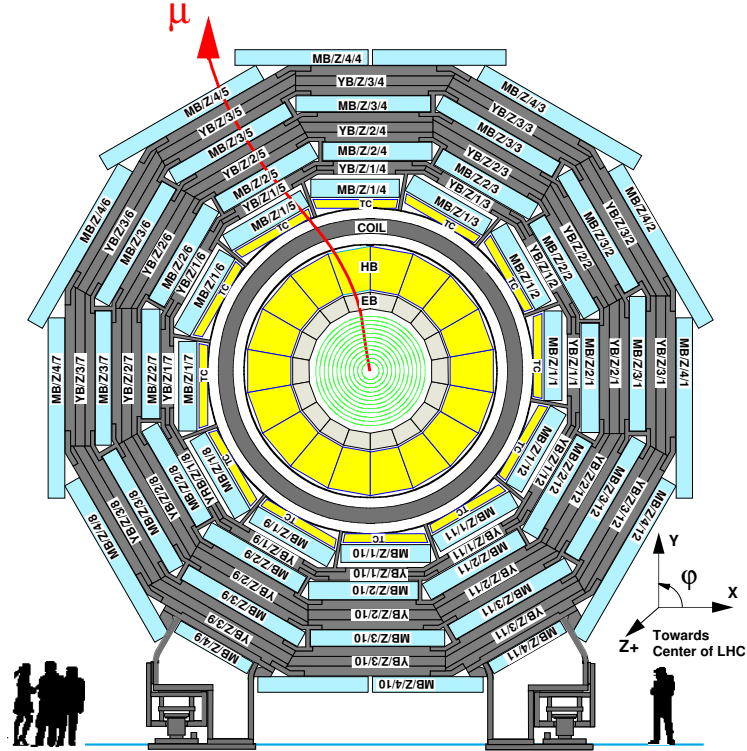


Figure 1.21: Transverse view of the CMS detector, showing the layout of the CMS barrel muon DT chamber stations (MB). [5]

RESISTIVE PLATE CHAMBERS In the pseudorapidity region $|\eta| < 1.6$ a complementary system consisting of Resistive Plate Chambers (RPC) detectors have been installed. The aim of the system is to facilitate the trigger decision, reducing the uncertainty for the background rates and estimating correctly the bunch-crossing time at full LHC luminosity. The RPCs are double-gap gaseous detectors, operating in the avalanche mode, characterised by a fast response. They have a better timing resolution but a coarser spatial resolution than the DTs and the CSCs. In the barrel, they complement the muon system with six additional layers, two in each of the first two DT stations and one in each of the last two stations. The additional layers in the first two stations are necessary to detect low- p_T tracks that do not reach the outermost stations. In the endcaps, a plane of RPC has been installed in each of the first three CSC stations.

The standalone efficiency of reconstructing muons for the muon system is about 95-99%, except in the regions around $|\eta| = 0.25$ and 0.8 between the two DT wheels and around $|\eta| = 1.2$, where there is a gap between the DT and CSC systems. The momentum resolution for the standalone system is around 9% for $p_T < 200$ GeV and between 15-40% at the TeV scale, depending on η . The resolution improves by around one order of magnitude including inner tracker data in the global momentum fit. In any case, the muon system provides a good enough p_T resolution in order to be used in the L_1 trigger, where a resolution of 15% in the barrel and 25% in the endcaps is achieved.

1.4.7 Trigger

At full luminosity, the LHC collides bunches of protons every 25 ns, with approximately 20 simultaneous proton-proton interactions per crossing. This results in a huge event rate of 40 MHz, which needs to be reduced for storage and processing. It is the responsibility of the CMS trigger system [65] to achieve this reduction. The task is divided into two steps: the Level-1 (L_1) trigger, which consists mainly of custom-designed programmable electronics and reduces the rate to a maximum of 100 kHz; and the software-based High Level Trigger (HLT), designed to have an output data rate of $\mathcal{O}(10^2)$ Hz.

1.4.7.1 Level-1 Trigger

The hardware-based Level-1 (L_1) trigger provides the first step in data rate reduction, aiming to lower the rate to 100 kHz. The online L_1 event selection exploits coarsely segmented information from the calorimeter and the muon systems. The full resolution data are temporarily stored in pipelined memories in the front-end electronics and passed to the HLT system in case of a positive decision from the L_1 .

When the L_1 marks an event as accepted, data from the custom front-end boards in each module are sent to the Front-end Readout Link (FRL) boards, which represent the interface to the central Data Acquisition (DAQ) network. Event fragments from the FRLs, for a total bandwidth of 100 GB/s, are input to the Event Builder, which assembles them in a single event that will be passed to the High Level Trigger (HLT) for processing.

The system was upgraded during LS1 [66], in order to cope with the new LHC conditions. In particular, the increased centre-of-mass energy of the collision, the increased instantaneous luminosity and the higher pile-up would increase the trigger rate up to six times the allowed rate, if no change to the trigger thresholds were made. The new system has been designed in order to have an intrinsic flexibility, being able to change architectures and algorithms by means of high-speed optical links and large FPGAs.

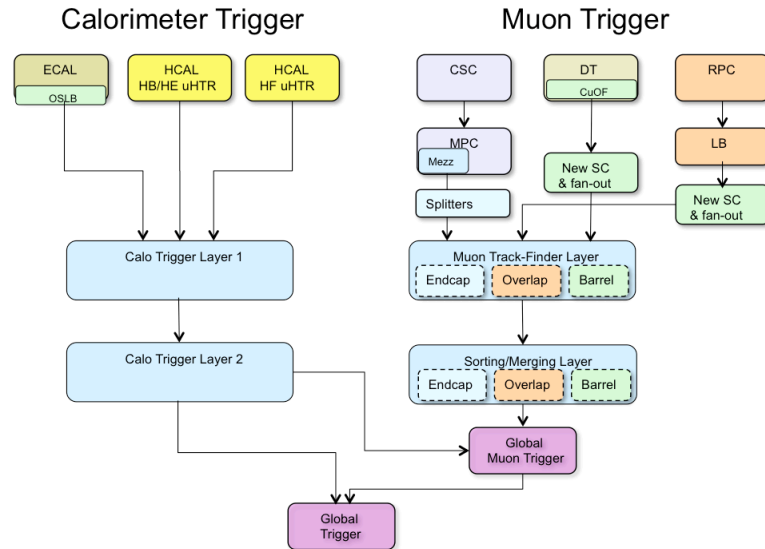


Figure 1.22: The upgraded CMS Level-1 trigger architecture for LHC run II. [66]

The upgraded architecture of the L_1 trigger is shown in figure 1.22. The system is divided into two parts: the L_1 calorimeter trigger and L_1 muon trigger. Their task is to produce the required physics objects (electrons/photons, hadronic tau decays, jets, muons and sum of energies), which will be collected in the Global Trigger (GT), where they will be used by about 400 algorithms in order to achieve a final trigger decision. In order to reduce the complexity of the system, CMS has adopted the telecommunication standard μ TCA, which guarantees additional flexibility and higher bandwidth with respect to the previously used VME framework.

Run II L_1 trigger upgrade constitutes an important basis for the future triggers at the HL-LHC. Many concepts can be also exploited for phase-II applications, for example the time-multiplexing that will be discussed further in Chapter 3.

1.4.7.2 High Level Trigger

The second step in the event selection in CMS is represented by the High Level Trigger (HLT) [67]. The HLT is implemented in software and runs on a large computer farm of thousands commercial CPUs, called the Event Filter Farm (EFF). The algorithms used in the HLT are a streamlined version of the offline reconstruction ones, sharing the same software framework [68].

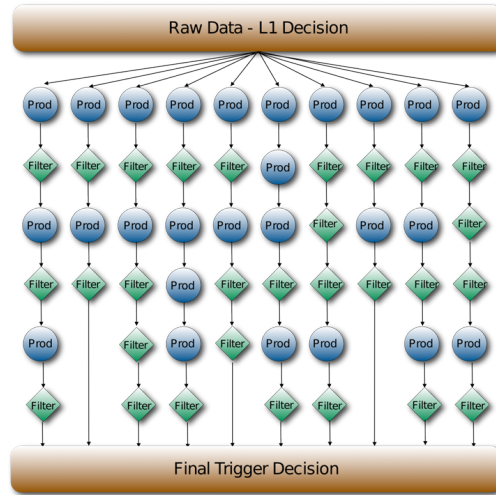


Figure 1.23: Schematic view of a HLT menu in CMS with its paths. The final trigger decision is the logical OR of the decisions of the single paths. [69]

The HLT has a modular structure, being divided into several *trigger paths*, consisting of a sequence of reconstruction and filtering algorithms executed one after the

other (Figure 1.23). The system has been designed in order to have the lowest possible CPU running time. For this reason, the implemented algorithms are arranged in a chain of increasing complexity, meaning that the faster algorithms run first. If an event does not pass the first filtering stage, the rest of the path is skipped and the trigger decision is taken. The HLT has access to the full high-resolution data from the entire detector and takes a decision based on selections for either single physics objects (electrons, photons, muons, etc...), for a combination of them or even using more elaborate algorithms for complicated physics analysis. It also has the capability of randomly discarding a fraction of events (*prescaling*). This is typically applied to signals whose higher rate might exceed the bandwidth limits, e.g. the calibration signal $pp \rightarrow W \rightarrow e\nu$ [70].

THE PHASE II CMS UPGRADE

As already described in Chapter 1, all the detectors at the future HL-LHC will need to be upgraded in order to cope with high instantaneous luminosity ($5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$), which places stringent constraints on the performance, radiation tolerance and longevity of the sub-detectors and read-out systems. Another important challenge is constituted by the increased number of pile-up collisions per BX, which is expected to be around 140.

The CMS collaboration is, therefore, planning a major upgrade during LS3, in order to make the detector ready for the LHC Phase II era. Purpose of the Phase II upgrade [46] will be to maintain the same excellent physics performance as the Phase I detector.

After almost ten years of operations CMS will need to replace its tracker detector [71] and the forward calorimeters [72]. The new tracker will benefit from increased granularity, necessary to maintain the excellent tracking efficiency to identify all the tracks produced by the increased number of charged particles in the collisions. The new endcap calorimeters will have an optimised segment configuration, which will improve the energy resolution, in particular for jet reconstructions.

Higgs boson studies will still be a central part of the CMS physics program also for Phase II. That requires that the p_T (E_T) thresholds for final-state muons (electrons) at the L1 trigger stage must remain low. On the other hand, keeping the same thresholds and the same trigger configuration will not be sustainable at high PU, since the trigger rate will go easily above the limits of the current electronic systems. The only way to keep the trigger rate within an acceptable level, without degrading the event reconstruction efficiency, is to improve the p_T resolution at L1. Further rate reduction can be also achieved by including extra information on the event reconstruction at L1 trigger level, e.g. the identification of the primary vertex in the collision. This will be achieved including data from the Outer Tracker in the L1 trigger system. Tracker data will be processed by the novel L1 *Track Trigger*, responsible for reconstructing tracks from high- p_T charged particles. The choice of the hardware architecture that will be used to build the

Track Trigger system has been matter of discussion in the last few years within the CMS collaboration and several proposals have been made for it. This thesis will discuss in detail one of the most promising approaches, based on FPGA electronics in a time multiplexed architecture. A brief description of the other proposals will be presented in Chapter 2.2.

Despite the additional handle provided by tracker data, the trigger output rate must be increased anyway, in order to keep the same efficiency performance for all physics channels, particularly for triggers involving hadrons and photons, more sensitive to high PU . Moreover, the trigger and the readout electronics will be designed in order to sustain possible luminosity peaks, in order to allow efficient data taking up to a PU level of 200.

The following section will discuss the planned upgrade of the tracker detector, analysing the design of the new silicon modules that will provide a first stage of on-detector data reduction. A full description of the CMS Phase II Upgrade can be found in [46].

2.1 THE PHASE II TRACKER UPGRADE

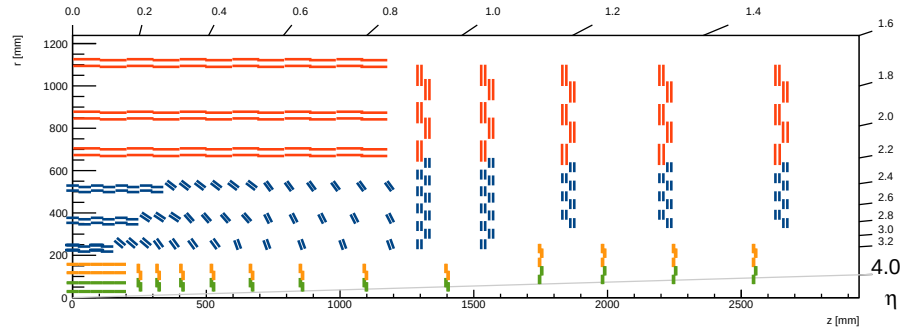


Figure 2.1: Sketch of one-quarter of the Tracker layout in the r - z plane. Outer Tracker: blue lines correspond to PS modules, red lines to 2S modules (see text). In the Inner Tracker, the green lines correspond to pixel modules made of two readout chips and the yellow lines to pixel modules with four readout chips. Produced with tkLayout [73].

Figure 2.1 shows a view of the upgraded silicon tracker detector in the r - z plane. The new tracker will be subdivided into two parts: the Inner Tracker (IT), made of silicon pixel modules, and the Outer Tracker (OT), consisting of two different kinds of silicon modules with strip and macro-pixel sensors. A detailed description of the Phase-2 tracker can be found in [74].

2.1.1 Requirements

The upgraded tracker detector should satisfy some essential requirements in order to operate efficiently at the HL-LHC.

- *Radiation tolerance.* The OT will be designed to operate efficiently throughout the entire HL-LHC data taking period, up to an integrated luminosity of $3,000 \text{ fb}^{-1}$, without any maintenance intervention. The IT will suffer from a very high radiation dose due to its proximity to the interaction point. Therefore it must be designed to be easily accessible, as for the current pixel detector, in order to facilitate any necessary module replacement;
- *Increased granularity.* The density of channels must be increased in order to keep the occupancy under the percent level, a critical condition for an efficient tracking with high PU. The increased granularity should also guarantee improved two-track separation for highly energetic jets;
- *Low material budget.* The amount of material in the tracker must be reduced to prevent secondary interactions of particles with matter (multiple scattering), which degrade the overall event reconstruction;
- *Robust pattern recognition.* Fast and efficient track finding must be ensured at high pileup conditions.
- *L1 trigger integration.* Data from the tracker will be used in the L1 event selection to preserve reconstruction performance in many physics channels;
- *Extended acceptance.* The new tracker will cover an extended pseudo-rapidity region up to $|\eta| = 4$.

2.1.2 The Inner Tracker Upgrade

The innermost part of the tracker detector will be still made of pixellated silicon sensors. The new IT will have to survive the extreme condition of the HL-LHC environment, where the high radiation dose and the increased pileup will place a very strict condition on the radiation hardness and on the front-end electronics.

The baseline of the Phase II upgrade of the pixel detector is shown in figure 2.2. The current proposal is to adopt 100-150 μm -thin pixel modules with a sensitive

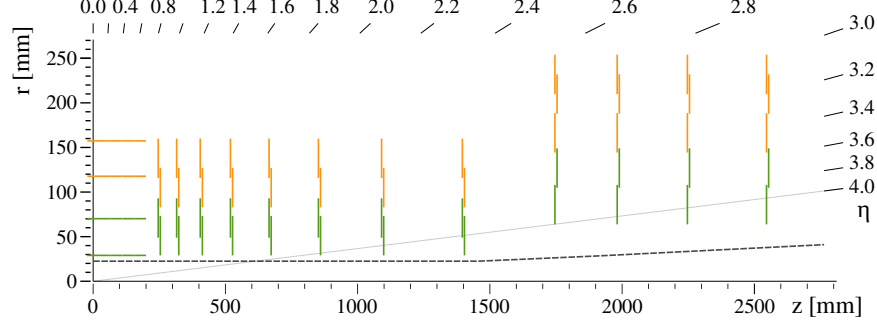


Figure 2.2: Sketch of one quarter of the inner tracker in the r - z plane. Green lines correspond to pixel modules made of two readout chips and the yellow lines to pixel modules with four readout chips. [74]

area of $2500 \mu\text{m}^2$ ($50 \times 50 \mu\text{m}^2$ or $25 \times 100 \mu\text{m}^2$), six times smaller than the current pixel detector, that will guarantee the required granularity, resolution and two-track separation capability. The modules will be arranged in four layers in the barrel, Tracker Barrel Pixel Detector (TBPX), eight small disks in the forward region, Tracker Forward Pixel Detector (TFPX), and other four larger disks in the endcaps, Tracker Endcap Pixel Detector (TEPX), for a total coverage up to $|\eta| = 4$.

The front-end readout chips will be based on 65 nm technology and are currently developed in collaboration with ATLAS in the framework of RD53 [75]. Two kinds of modules are foreseen at the moment for implementation. The innermost layers/disks will adopt pixel modules with two readout chips, while in the outermost part of the IT single readout module will be installed.

2.1.3 The Outer Tracker Upgrade

The Outer Tracker (OT) will cover the region of $200 < r < 1200 \text{ mm}$ and $|z| < 2700 \text{ mm}$. It will be constituted by a new type of silicon modules, the p_T modules, which will perform a first on-detector track selection. The p_T modules will have two parallel strip sensors, separated by few millimetres. Looking at the correlation of signals in the two sensors compatible with a high- p_T particle (*stub*), it will be possible to reject uninteresting hits and to reduce the data rate at the input of the L1 trigger by a factor ten. Figure 2.3 illustrates the operation of a p_T module.

Two different kinds of modules have been developed: the 2S modules with two strip sensors and the PS modules, equipped with a strip and a macro-pixel sensor.

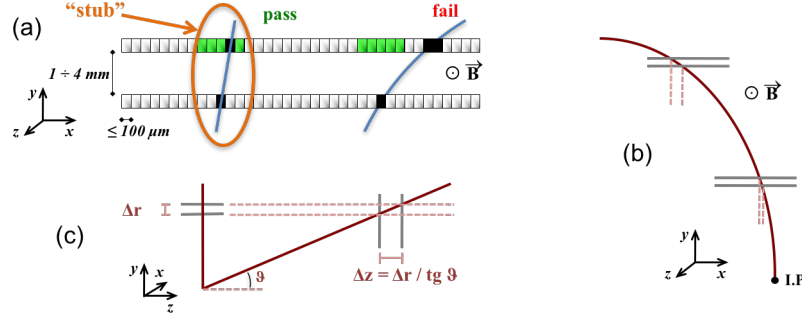


Figure 2.3: Operation of a p_T module. (a) Search for correlation of hits in the two closely-spaced sensors to reject low- p_T particle signals. The selection window for an accepted stub is depicted in green. (b) Wider windows are required at large radii, for the same transverse momentum. (c) Larger sensor spacing is required in the endcap disks to achieve the same rejection capability as in the barrel. [76]

The 2S modules have a strip length of about 5 cm. The PS modules are instead 2.4 cm long and are segmented in 1.5 mm-wide macro-pixels, for a more accurate z measurement. Sketches of assembled p_T modules are shown in Figure 2.4. Table 2.1 summarises the main characteristics of the 2S and PS modules.

Table 2.1: Main parameters of the 2S module and the PS module of the CMS tracker. [74]

2S module	PS module
$\sim 2 \times 90 \text{ cm}^2$ active area	$\sim 2 \times 45 \text{ cm}^2$ active area
2×1016 strips: $\sim 5 \text{ cm} \times 90 \mu\text{m}$	2×960 strips: $\sim 2.4 \text{ cm} \times 100 \mu\text{m}$
2×1016 strips: $\sim 5 \text{ cm} \times 90 \mu\text{m}$	32×960 macro-pixels: $\sim 1.5 \text{ mm} \times 100 \mu\text{m}$
Front-end power $\sim 5 \text{ W}$	Front-end power $\sim 8 \text{ W}$
Sensor power (-20°) $\sim 1.0 \text{ W}$	Sensor power (-20°) $\sim 1.4 \text{ W}$

2.1.3.1 Outer Tracker Geometry

The silicon modules will be arranged into six hermetic cylindrical layers in the barrel region up to $|z| < 1200 \text{ mm}$, which are enclosed by five hermetic double disks in the endcap. PS modules will be deployed in the radial region up to $|r| = 600 \text{ mm}$, while the 2S modules will be adopted in the rest of the tracker. The geometry of the tracker has been designed to guarantee that particles emerging from the luminous region $|z| < 70 \text{ mm}$ will cross at least six layers, in the pseudorapidity region $|\eta| < 2.4$, except in the small gap region around $|\eta| \sim 1$ where only five layers will be crossed. The choice of six layers, and three PS layers has been made in order to ensure a minimal redundancy against minor inefficiencies affecting only one layer.

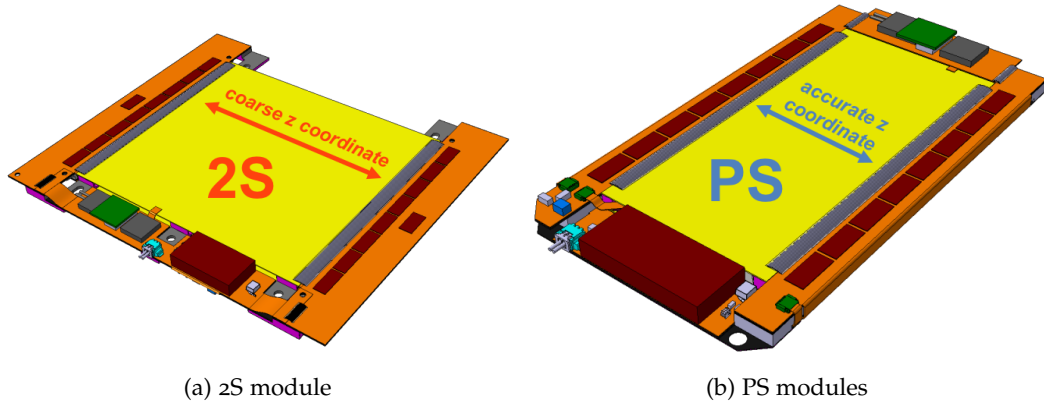
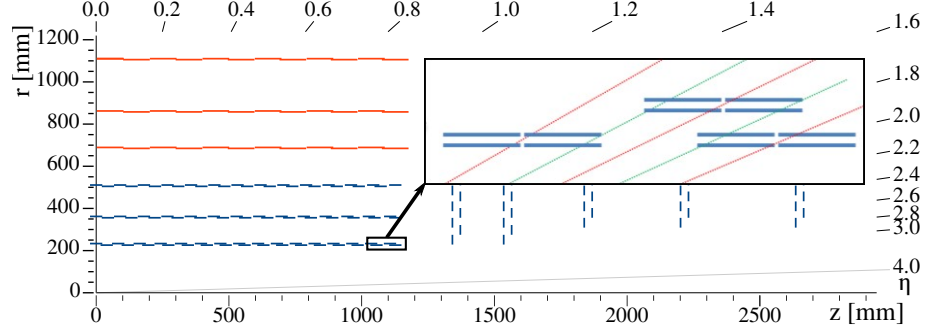
Figure 2.4: Schematic views of the assembled silicon p_T modules. [74]

Figure 2.5: Stub finding inefficiency in flat barrel PS modules. Each blue line represents half of a sensor. At the edge of the inner barrel layers a large irrecoverable inefficiency exists for tracks that cross the modules at the separation point of the two halves of a sensor (red lines). [76]

To find a stub it is necessary that a particle leaves signals in both the top and bottom sensors of a single module. Inefficiencies can emerge in the case a particle crosses a module near the separation point of the two sensor halves (Figure 2.5) in the case of a flat barrel layout. For this reason, CMS opted for an innovative geometry, where the first three layers in the barrel have progressively tilted modules, almost perpendicular to incident particles over the entire barrel length. Figure 2.6 shows the baseline layout with the new tilted geometry.

The flat geometry has been the baseline for the Outer Tracker upgrade for a long period [46], while the tilted geometry has been officially adopted only recently. Therefore the majority of the studies presented in this thesis have been conducted using a flat barrel layout, particularly results that will be shown in Chapters 3 and 4. Some preliminary results with the novel adopted tilted geometry will be also illustrated.

The OT is subdivided into three parts: the TBPS, the TB2S and the TEDD. The TBPS will be made only of PS modules, which will be mounted on two different structures. In the central part, the modules will be installed on *planks*, where the consecutive modules are arranged on the opposite sides of the central plane of the plank, overlapping in z to form a hermetic shape. The side section of the TBPS will be completed with rings, containing modules with a progressive tilted angle. The TB2S will be formed by 2S modules, arranged on *ladders* in a similar way to the planks in the TBPS. Finally, modules in the TEDD will be installed on double-disks. Each disk will be formed by two halves (*dees*), the bottom made of PS modules and the top of 2S. The double-disk design is required in order to achieve overlaps along r .

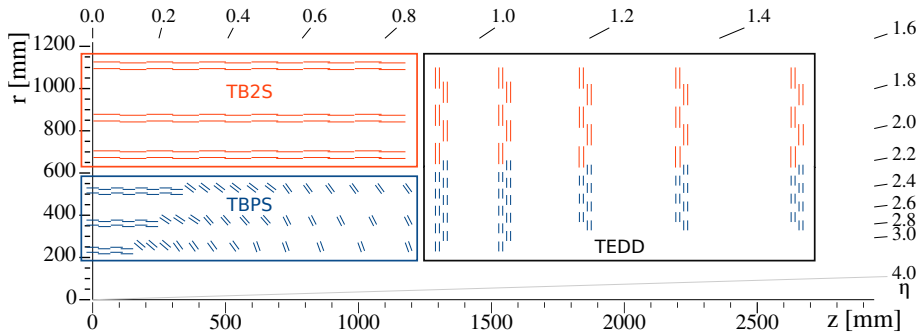


Figure 2.6: $r - z$ view of the Outer Tracker. PS (2S) modules are drawn in blue (red) lines. The three sub-regions (TBPS, TB2S and TEDD) are indicated. [76]

2.1.3.2 Front-End electronics

The OT will deliver data for every bunch crossing to the L1 trigger with a rate of 40 MHz. A block diagram of the electronic system for a 2S module is shown in Figure 2.7.

The Front-End (FE) electronics will be built on-module, surrounding the sensors. In the 2S modules two electronic hybrids are wire bonded to the silicon strips (right, R, and left, L). In the PS, the hybrids are bonded to the strip sensors and to a macro-pixel sub-assembly, integrating the pixellated sensors. The digital data will be produced by eight FE ASIC chips: the CMS Binary Chip (CBC), the Short Strip ASIC (SSA) in the 2S and the Macro-Pixel ASIC (MPA) in the PS. Data are then buffered, aggregated and formatted in the Concentrator Integrated Circuit (CIC).

Each group of 72 modules send their data to the Back-End (BE) DTC boards, for processing. These ATCA boards, based on FPGA technology, will handle three data input/output detector streams: the data acquisition DAQ stream, containing the full resolution data that will be sent out upon receiving an accept signal from the L1 trigger; the trigger TRIG stream, representing the data sent to the L1 Track Trigger; and the timing & control TTC stream, including the clock, the trigger and the configuration signals.

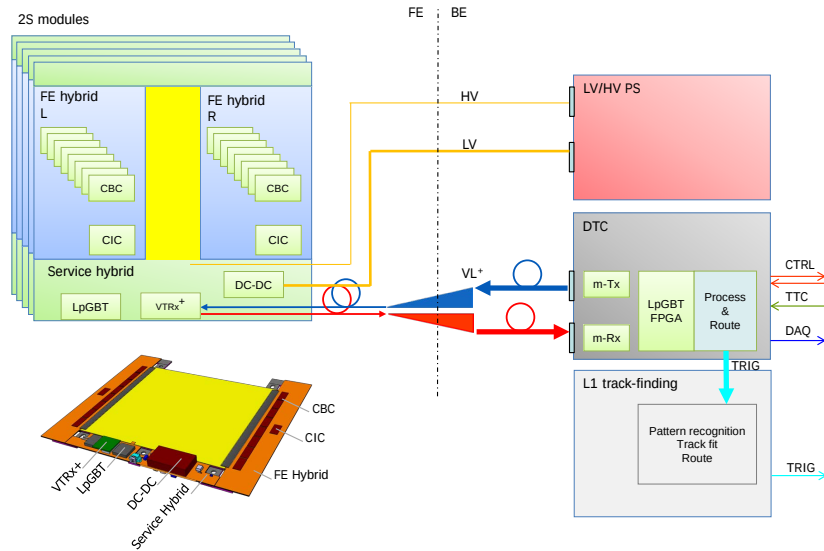


Figure 2.7: Block diagram for the Front-End (left) and Back-End (right) electronic system of a 2S module. [76]

2.2 THE CMS PHASE II L1 TRIGGER

In order to undertake its ambitious physics program, CMS will have to upgrade the L1 trigger system for the Phase II era. The main goal is to guarantee the same or to improve the overall physics acceptance under the extreme condition at the HL-LHC. In particular, the new L1 trigger has to be able to recognise efficiently events from relatively low-mass physics processes (i.e. Higgs production) and perform precise measurements of p_T and Missing Transverse Energy (MET).

In addition to the inclusion of track information at L1, also the detector readout, the L1 trigger and HLT systems will be upgraded. The maximum L1 read-out data rate will be increased to 750 kHz (compared to the current 100 kHz), the L1 latency to 12.5 μ s (currently 4 μ s) and the permanent storage rate to 7.5 kHz (around 1 kHz at Phase I).

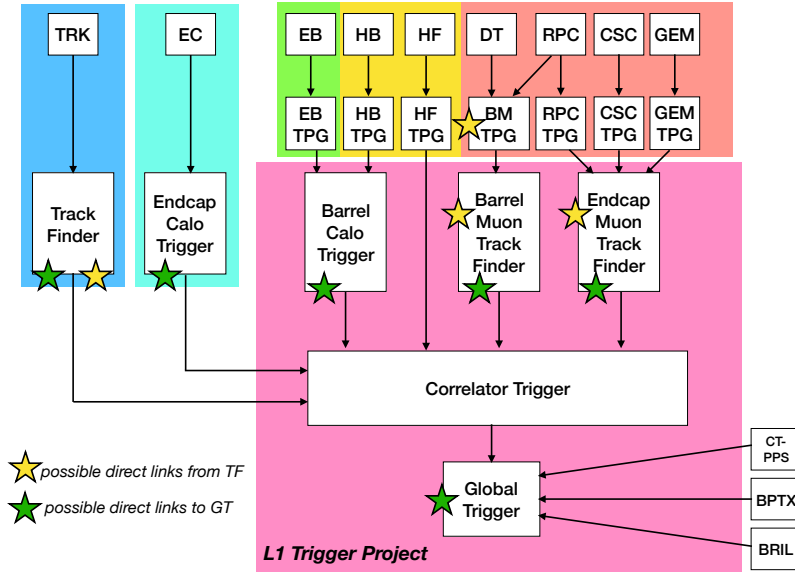


Figure 2.8: Simplified layout of the Phase II L1 trigger. The following sub-detectors are labelled in the diagram: the Outer Tracker (TRK), the Endcap Calorimeter (EC), the ECAL Barrel (EB), the HCAL Barrel (HB), the HCAL Forward detector (HF), the Muon Drift Tube Detectors (DT), the Resistive Plate Chambers (RPC), the Cathode Strip Chambers (CSC) and the Gas Electron Multiplier Chambers (GEM). [77]

These modifications have been motivated by simulation studies in which the same Phase-I trigger algorithm [66] would have been used also for Phase II. Keeping the same L1 menu of Phase-I, it would be necessary to increase the L1 read-out

data rate to 4,000 kHz, with beam conditions of 200 PU, well beyond the possibility of any detector upgrade [46]. Adding tracking information reduces the rate to about 500 kHz. Therefore, taking into account also the uncertainties due to possible simulation errors, the CMS collaboration has decided to increase the L₁ output bandwidth to 750 kHz.

The L₁ trigger latency has to be increased to 12.5 μ s, in order to give sufficient time for the L₁ track reconstruction and for the matching of the three sub-trigger system data. Each trigger subsystem will have a time budget of 5 μ s to produce the trigger primitives, that will be sent to the L₁ correlator where the information will be merged and a global trigger decision will be finally taken. Expecting an acceptance rate of the order of 1%, same as today, a permanent storage rate for the HLT of 7.5 kHz will be sufficient.

Figure 2.8 depicts a simplified layout of the Phase II L₁ trigger [77]. Trigger primitives are generated from separated sub-systems. Data from the EB and the HB are clustered in the Barrel Calorimeter Trigger (BCT), while the HF and Calorimeter endcap (EC) will require a separate endcap system. Muon tracks will be reconstructed separately in the barrel and in the endcaps by the Barrel Muon Track Finder (BMTF) and the Endcap Muon Track Finder (EMTF) processors. Outer tracker stubs will be input to the L₁ Track Trigger, responsible for finding charged particle tracks. Information from the three sub-systems will be collected by the Correlator Trigger (CT), where the reconstruction and identification of the trigger objects, for both *standalone* and *full-detector* objects, will happen. Finally, data are transmitted to the GT, which calculates global quantities and implements the trigger menu. The GT will output an accepted L1A signal, to be fed into the Trigger Control Distribution System (TCDS).

The following paragraphs provide an overview of the upgrades of the individual sub-trigger systems. The novel track trigger system will be described in Section 2.3. The L₁ Correlator Trigger and upgraded L₁ Global Trigger will be addressed in Section 2.2.3.

2.2.1 Calorimeter trigger

Because of the increased pileup conditions, the upgraded calorimeter trigger will need access to the full resolution information to meet the selection requirements. This will be achieved by replacing the front-end and back-end electronics and by

increasing the bandwidth from the detector for trigger information. The upgraded trigger will process data from single calorimeter crystals, differently from the current **L1** Calorimeter trigger that looks into towers made of 5×5 crystals. Also increasing the standalone latency budget to $5 \mu\text{s}$ will permit the implementation of more complex algorithms.

In addition, **CMS** will substitute entirely the endcap calorimeters with a new high granularity sampling calorimeter, using silicon and scintillator as sensitive material [78]. 52 sensitive layers will be placed in each endcap, 28 for the electromagnetic and 24 in the hadronic sections. Only half of the electromagnetic layers will produce data for the **L1** trigger, while all the hadronic layers will be used.

The goal of the **L1** Calo Trigger will be to identify electrons, photons, jets and τ candidates in the **ECAL** and **HCAL**. It will also provide information about some global parameters, such as the total deposited transverse jet energy E_T , the total missing transverse energy MET and the total jet energy H_T . Thanks to these measurements, the Calo Trigger will be capable to give also an estimate on background level in the event on a bunch-by-bunch basis.

Access to the full resolution data will improve also the spatial resolution, which results in a better matching with the other sub-system primitives (tracks, muons) in order to reduce fake rate due to the combinatorial background.

2.2.2 Muon Trigger

The purity of the **L1** muon candidates is expected to be generally high even at high luminosity conditions. The main issue will be constituted by the increased trigger data rate, due to the coarse p_T resolution. The current **L1** muon trigger has to operate in fact with a lower p_T threshold than the off-line reconstruction, which results in a higher data rate. Also, the tails of the momentum resolution contribute by allowing a soft muon candidate to be reconstructed as a high p_T one. This results in a flattening of the rate at $p_T \sim 20 \text{ GeV}$, dominated by mis-measurements.

CMS will provide a solution to this problem already during **LS2**, where additional Gas Electron Multiplier (**GEM**) stations will be installed in the region $|\eta| > 1.6$. This will guarantee enough redundancy with the **RPC** and **CSC** system and enough points for a more accurate track fit. The upgraded **L1** muon trigger will have also access to the full-resolution **DT** information, improving track matching in the longitudinal plane and standalone p_T measurements.

2.2.3 The L1 correlator trigger and the L1 Global Trigger

The task of the L1 Correlator Trigger (CT) is to reconstruct the entire event by processing the incoming data from the three sub-trigger systems. This will require a correct identification of the hard interaction primary vertex, which will be achieved using the L1 tracks obtained by the L1 track finding system. In addition to that, the CT will implement also the reconstruction of particle-flow objects, similar to current offline reconstruction [79], performed by matching tracks with towers in the calorimeters and muon tracks in the muon chambers.

The CT will transmit out to the L1 Global Trigger (GT) the reconstructed physics objects (photons, muons, jets, taus, energy-momentum sums), together with some analysis-like objects, e.g. heavy-flavours tags. Also *standalone* objects will be produced, utilising information from the individual sub-detectors, in order to guarantee redundancy and robustness of the system.

The architecture and the design that will be used for the CT are still a matter of discussion within the CMS collaboration at the time of this writing. Several proposals have been made, including a regional/layered architecture or a fully time-multiplexed system [77]. The same considerations can be applied also to the upgrade of the GT, where two proposals for the final architecture have been placed. The first more conservative proposal still relies on the current GT system, made of several processors receiving the full set of trigger objects, but calculating different trigger menu algorithms. A second approach makes use once more of a time multiplexed concept design, in which each processor would implement the full set of trigger algorithms, that will run on consecutive events.

2.3 THE TRACK TRIGGER

The use of the silicon tracker data for L1 processing is not a completely new concept in particle physics. In the past, there have been already successful cases of experiments, which based their event selection on the tracker data [80, 81]. However one can certainly affirm that the use of tracker data in high-luminosity hadron experiments like CMS constitutes a completely different challenge, which requires exploitation and development of novel technologies.

The L1 Track Trigger will process the stub information coming out of the Outer Tracker at a rate of 40 MHz. The stubs finding stage will reject all the hits of tracks

with $p_T < 2\text{ GeV}$, reducing the data volume coming into the track trigger system by a factor 10 to 100. About 10,000 stubs will be produced at each LHC bunch crossing and will be used to form L1 track candidates.

The architecture to be used for the Track Trigger system has been a matter of discussion within the CMS collaboration for several years, where three different approaches have been pursued:

- **Associative Memory plus FPGA approach.**
- **FPGA-based Hough Transform approach.**
- **FPGA-based tracklet approach.**

To understand the feasibility of a L1 Track Trigger, each of the three approaches has been required to demonstrate its capability to efficiently reconstruct tracks with $p_T > 3\text{ GeV}$ out of stubs in a $4\text{ }\mu\text{s}$ latency budget.

The demonstration happened on two different levels: a hardware demonstrator and a software emulation. The hardware demonstrator has been used to evaluate tracking performance in a representative slice of the detector, and to measure the required overall latency together with hardware-specific effects, e.g. truncation, as it will be shown in the Chapter 4. It provided also information about performance, scale and cost a future system. Simulation studies have been utilised to estimate tracking performance for the entire tracker and constituted a benchmark for the hardware results.

The baseline geometry used for the demonstrators corresponded to a Outer Tracker *flat* barrel geometry, as in [46]. Data-sets containing different physics signals ($t\bar{t}$, di-muon production, di-electron production) have been produced and have been used by the three groups. Three pileup conditions have been explored (0, 140 and 200).

TRACKING PERFORMANCE CRITERIA. The main discriminating quantities to evaluate L1 tracking performance are the tracking efficiency, the track parameter resolutions and the rate of output tracks.

The **Tracking Efficiency** (ϵ) is defined as the ratio of the number of L1 tracks that match a truth track to the number of truth tracks.

$$\epsilon = \frac{N_{\text{Matched}}}{N_{\text{Truth}}}. \quad (10)$$

Table 2.2: Definitions of the objects to be used to estimate the L1 Track Finding performance.

Truth Particles	$p_T > 3 \text{ GeV}$ $\geq 4 \text{ stubs in } \geq 4 \text{ layers}$ $ \eta \leq 2.4$ $ z_0 \leq 30 \text{ cm}$ $d_{xy} < 1 \text{ cm}$
L1 Tracks	$ \eta \leq 2.5$ $\leq 30 \text{ cm}$ $\geq 4 \text{ stubs in } \geq 4 \text{ layers}$
Matched Tracks	same as L1 Tracks $\geq 4 \text{ stubs matched with a single Truth particle}$

A **L1** track is considered to be matched with the truth only if it contains four stubs generated by the corresponding generator-level particle. To reduce the population of truth particles to the one relevant for track finding, only particles with at least four stubs in four different tracker layers have been considered. The full description of requirements on **L1** tracks, Truth tracks and Matched tracks are listed in Table 2.2.

The **resolution** σ_p of a parameter p is defined as the interval for which 90% of tracks are contained in the residual distribution x ,

$$\begin{aligned}
x &= p_{\text{Matched}} - p_{\text{Truth}} , \\
\sigma_p \text{ s.t. } P(\bar{x} - \sigma_p < x < \bar{x} + \sigma_p) &= 0.9 .
\end{aligned} \tag{11}$$

Finally the **output rate** f is defined as the average number of **L1** tracks per event,

$$f = \frac{N_{\text{L1Tracks}}}{N_{\text{Events}}} . \tag{12}$$

It is particularly useful for a further understanding of the demonstrator results to divide the rate into contributions from matched tracks f_{good} , tracks that have matched to an already matched tracks (*duplicates*) $f_{\text{duplicates}}$ and from combinatorial fakes f_{fakes} . The definitions of fake and duplicate tracks are different for each approach.

2.3.1 Track trigger architectures

The following sub-sections will provide a brief description of the *AM+FPGA* and of the *Tracklet* approaches. A full description of the *Hough Transform* proposal will be given in Chapter 3.

2.3.1.1 Associative Memories

The *AM+FPGA* approach is based on the use of custom *AM ASICs* to perform a fast pattern recognition with coarse stub information, and of *FPGAs* for the final fitting stages [82]. The *OT* will be divided into 48 independent regions (trigger towers) in η - ϕ , each of them assigned to a single *AM* chip. Every chip is loaded with a set of predefined hit patterns, called roads, corresponding to a valid high- p_T ($p_T > 2 \text{ GeV}$) particle trajectory. Those roads are constructed using low-granularity *super-strips* hits, corresponding to the union of several silicon sensors. The size of the super-strips determines the total number of patterns required to fully cover a trigger region. An optimal value has to be chosen in order to balance efficiency and fake data rate requirements. It has been demonstrated that 1.5 million patterns per trigger tower can guarantee good track finding performance.

About 100 stubs per layer per bunch crossing will be received by each trigger tower, which will be compared with the pre-loaded patterns in the *AM* chip. If a given pattern matches stubs in at least four layers, the road is marked as *fired*, and the pattern address is transmitted to the *FPGA*. Figure 2.9 shows the data-flow for a single trigger tower. Data from different *BX* are time-multiplexed so that each *AM+FPGA* board will be responsible for the processing of the data of a certain trigger tower within a single *LHC* event. The total number of required boards will be hence equal to the number of trigger towers times the time-multiplexed period. The first step is constituted by the Data Organizer (DO), which converts the full resolution stub data into super-strips and transmits them to the *AM* chip for the road matching step. The stubs assigned to a fired pattern are then sent downstream in the *FPGA*, where the linearised track fitting stage takes place and tracks with a bad fit χ^2 are rejected. Finally, a duplicate removal stage is implemented, keeping only the tracks with the best fit probability. Surviving tracks are finally transmitted to the *L1* trigger correlator for downstream processing.

The *AM+FPGA* proposal has been demonstrated in hardware, emulating the functionality of the *AM* chip on an *FPGA*. The system operates with a fixed overall

latency, from DTC to L_1 correlator, of $3.5\ \mu\text{s}$ and tracking efficiency for muons in $t\bar{t}$ events with a pileup of 200 of about 98%.

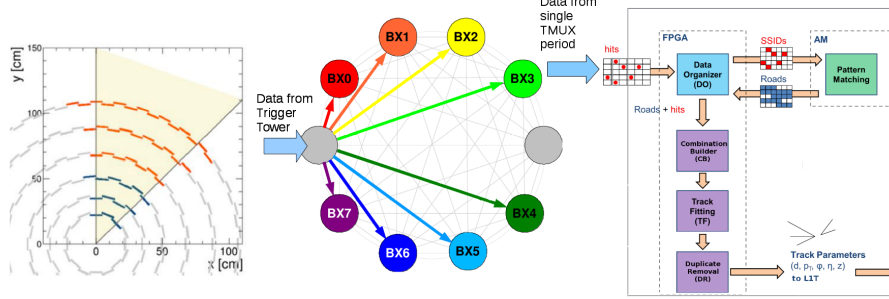


Figure 2.9: Data-flow for a single trigger tower in the AM+FPGA trigger architecture. Data from the trigger tower are time-multiplexed and transmitted to the AM+FPGA processing boards. The pattern recognition tasks is performed by the AM chips, while the FPGA takes care of communication and downstream processing. [82]

2.3.1.2 Tracklet

The tracklet proposal is based on the use of commercially available FPGA circuits to address the track finding task [83]. This approach is based on a road-search algorithm with a pipelined firmware implementation. Figure 2.10 shows a schematic of the tracklet track finding algorithm. Roads are found by forming pairs of stubs in adjacent OT layers and disks (*tracklets*). Initial tracklet parameters are estimated using the two stubs and demanding that tracks originate from the detector origin in the r - ϕ plane. Tracklets are formed using pairs of stubs from barrel layers 1+2, 3+4, 5+6, endcap disks 1+2 and 3+4, and barrel layer 1 + endcap disk 1. Only tracklets compatible with a track with $p_T > 2\text{ GeV}$ and longitudinal impact parameter $|z_0| < 15\text{ cm}$ are then kept. Roads are then extrapolated to the other tracker layers in order to match with other stubs. New track parameters are then calculated via a linearised χ^2 track fit. Finally, a duplicate removal step gets rid of possible duplicate tracks, by comparing pairs of tracklets sharing the same stubs.

The tracklet architecture is based on a time-multiplexed system, with a baseline time-multiplex factor of six. The detector is subdivided into 28 sectors in ϕ and each sector is assigned to a single track finder processing unit. The system has been demonstrated in hardware, resembling three sectors in ϕ and one slice in time. The algorithm has been implemented on a Xilinx Virtex-7 XC7VX690T FPGA [84], mounted on μTCA CTP7 boards [85], running at 240 MHz. In total, the

hardware demonstrator employs four boards, one for the central ϕ sector, two for its neighbours and a board for input/output transmission.

The demonstrator has been able to reconstruct tracks with good efficiency with high pileup, meeting the required latency.

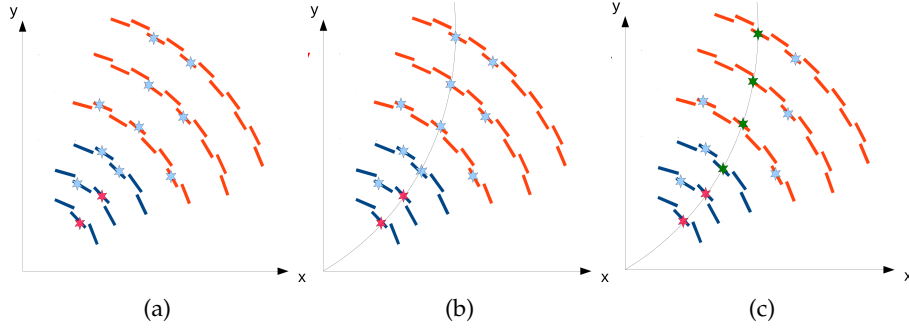


Figure 2.10: The Tracklet track finding algorithm. (a) Tracklets are formed using pair of stubs (red stars) in different tracking layers. (b) Roads are extrapolated to remaining tracking stations. (c) Only stubs compatible with the trajectory (green stars) are kept. [83]

2.4 TRIGGER ALGORITHMS

The main goal of the L_1 trigger upgrade will be to keep similar trigger thresholds as Phase 1 without drastically increasing the L_1 accept output bandwidth. This may be achieved only by improving the performance of object identification algorithms. In particular, it will be of essential importance to be able to reconstruct the primary event interaction vertex and to isolate uninteresting pileup collisions.

Novel trigger algorithms are currently under development, taking advantage of the included tracking information. Three classes of trigger algorithms can be defined: standalone-objects, track-matched objects and particle-flow objects. Standalone algorithms make use of data from independent sub-detectors, providing the capability to make a trigger decision without the need for more complex algorithms. Track-matched algorithms improve performance with respect to standalone algorithms, maintaining a relatively simple design. The last step is constituted by the particle-flow algorithms, which resemble the offline software algorithms. Combining information from all sub-detectors, they are expected to give the best performance but to require the most processing time and resources.

The following sections will present some examples of how the track information can be used to improve trigger performance. A detailed list of the algorithms that have been investigated for the Phase 2 L_1 trigger can be found in [46, 77].

In particular, the availability of track information at the L_1 trigger level gives the opportunity for reconstructing the primary interaction vertex in the event, making use only of L_1 tracks. Several algorithms have been investigated as possible candidates for a vertex reconstruction implementation for the Phase 2 L_1 Trigger. The obtained performance and results are presented in Chapter 5.

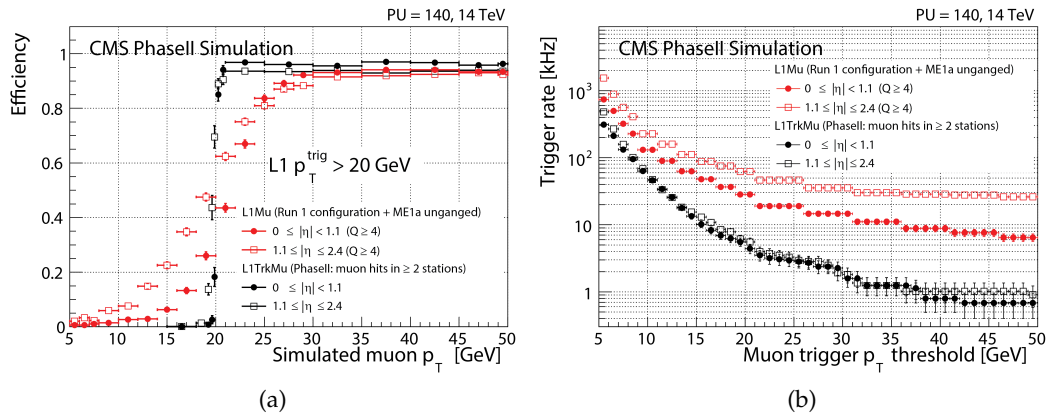


Figure 2.11: (a) Single muon trigger efficiency with a $p_T = 20$ GeV threshold, as function of the generated muon transverse momentum. (b) Single muon trigger rates as a function of the p_T threshold. Results obtained with a standalone muon trigger (red) are compared with a matched muon-track trigger (black). [46]

2.4.1 Matching muon trigger primitives to L_1 tracks

The Phase I Muon trigger is a standalone system, responsible to reconstruct muon tracks using hits in the muon chamber detectors. Despite the Phase I upgrade improvements to the stand-alone p_T resolution of the muon trigger, the system will occasionally assign high momenta to low p_T muons that would eventually pass the trigger threshold. This will result in a flattening of the output muon trigger bandwidth at p_T threshold larger than 20 GeV. Matching a muon track with a L_1 track found in the inner tracker has been shown to improve drastically the p_T resolution, as shown in Fig. 2.11a. Here the efficiency for a standalone muon trigger is compared with a track-matched trigger, for a single muon p_T threshold equals to 20 GeV. The improved resolution translates into a much sharper turn-on

and, in turn, to a large reduction of the trigger rate (Figure 2.11b), about a factor 10 for a p_T threshold of 20 GeV with respect to a standalone muon trigger rate.

Two different matching algorithms have been investigated: an *outside-in* matching algorithm in the central pseudorapidity region, $|\eta| < 1.1$, where a reconstructed muon candidate track in the muon chambers is extrapolated and matched with a L_1 track in the tracker; and an *inside-out* algorithm in the full η range, starting from the L_1 tracks and extrapolating the trajectory to the muon stations [46].

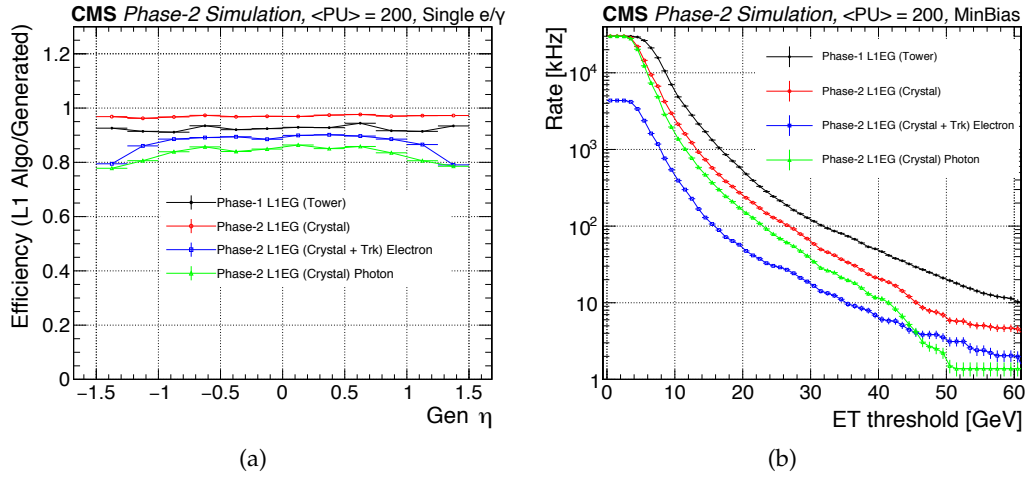


Figure 2.12: (a) Expected efficiency for a single electron trigger with a threshold of 20 GeV in the barrel region for different scenarios: calorimeter only (red), calorimeter photon tuned trigger (green), and calorimeter matched to the track (blue). Also, current trigger (black) performance are shown. (b) Expected trigger rate of single electron calorimeter trigger as a function of trigger threshold, for minimum bias events. [77]

2.4.2 Matching L_1 tracks to ECAL clusters

The upgrade of the electronics for the barrel calorimeter will improve the energy measurement at the L_1 trigger level, taking advantage of the full ECAL granularity of $(0.0175, 0.0175)$ in (η, ϕ) , as opposed to the current trigger tower granularity of $(0.0875, 0.0875)$. The trigger algorithm under study [77] is a simplified version of the one used for the offline reconstruction. Once a seed crystal has been found to have a $p_T > 1$ GeV, a core cluster of 3×5 crystals is defined around it. The spatial position of the cluster is calculated using a weighted sum of the energies of the individual crystals. The shape of the cluster is used to discriminate signals from

electrons and from photons. Finally, the cluster is matched with tracks from the [L1](#) Track Trigger, whose trajectories are extrapolated to the [ECAL](#) surface.

Figure [2.12a](#) shows the obtained efficiency for a standalone [ECAL](#) trigger compared with an electron-track matched trigger and with the current trigger performance. The distribution of trigger rates as a function of the trigger threshold is presented in figure [2.12b](#).

AN FPGA-BASED TIME MULTIPLEXED TRACK TRIGGER

Field Programmable Gate Array (**FPGA**) technology is an alternative to Application Specific Integrated Circuit (**ASIC**) to design and implement complex hardware solutions [86]. An **FPGA** is an Integrated Circuit (**IC**) that can be electrically programmed to become almost any kind of digital hardware [87]. Normally an **FPGA** consists of:

- Configurable Logic Blocks (**CLBs**) that implement some logic functions,
- a programmable routing system to connect the logic blocks,
- I/O blocks connected to **CLBs** to assure off-chip connections.

A **FPGA** chip as sold does not realise any operations. It is function of the firmware developers to program the device to implement the desired system. The design must be firstly written in a Hardware Description Language (**HDL**) or as a schematic design. Then, by means of a synthesis software provided by the **FPGA** vendor, the design is translated in a netlist, which will be fit into the actual **FPGA**.

The unique re-programmability feature of the **FPGAs** makes them a very flexible tools. Indeed, if any modification to the system is required, it is sufficient to load new firmware on the **FPGA**. In order to be able to reprogram the system, however, a **FPGA** usually needs additional silicon area compared to hard **ASIC** implementations of the same logic. Although this increases the cost of the **FPGA** chip by a factor 1.5-10 with respect to **ASICs**, one has to consider also the high development costs that **ASICs** require, in addition to the time needed to design the system.

Figure 3.1 shows how the high cost of an **ASIC** development is justified only by hardware systems composed of a large number of units. Moreover, the increased complexity of the new generation of lithography makes the cost of developing custom **ASICs** increasingly high [88].

For these reasons **FPGA** technology represents a valuable alternative for applications in particle physics experiments, nowadays being widely used in the electronic systems by all the **LHC** experiments and in particular by **CMS** [89]. Over the

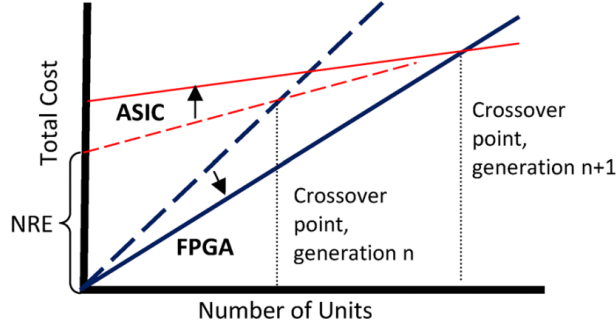


Figure 3.1: Total cost as a function of number of units for **ASIC** (red) and **FPGA** (black) based systems. In dashed lines are shown the expected changes due to future **FPGA** developments. Use of **ASICs** will become advisable only for very large systems. [86]

past 30 years, **FPGAs** increased their capacity (number of **CLBs**) by more than a factor 10,000 and the clock frequency by a factor 100 (Fig. 3.2), making it possible to implement algorithms of higher complexity.

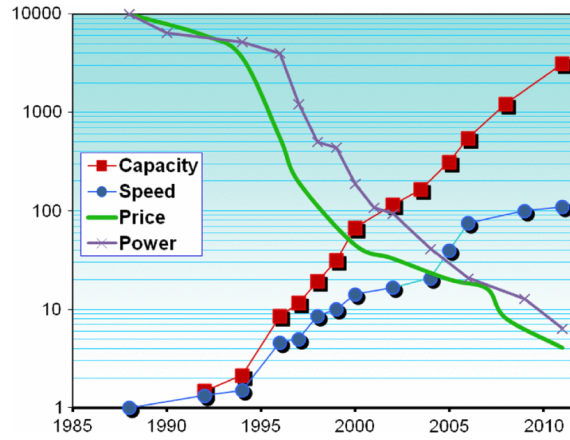


Figure 3.2: Evolution in time of Xilinx **FPGA** characteristics. [86]

The future **L1** Track Trigger for the Phase II upgrade of the **CMS** experiment will require the use of complex algorithms in order to reconstruct tracks from the 10,000 stubs produced at each bunch crossing. This task has to be fulfilled within $5\mu\text{s}$ and therefore can be addressed only by intelligent high-speed electronics. With their flexibility, **FPGAs** represent a very good candidate to build the new Track Finding architecture, giving the possibility of tuning or changing algorithms even after the installation of the final system.

The chosen proposed system has a scalable, configurable and redundant fully time-multiplexed [90, 91] architecture. This chapter will explore the potentiality of time multiplexing, showing in particular how the TMTT concept has been designed in order to cope with the hardware constraints placed by the upgraded CMS Outer Tracker detector and its electronics.

The selected track finding algorithm is based on the Hough Transform (HT) technique [92], utilised to coarsely group stubs consistent with high- p_T tracks into candidates and to reduce the combinatorics [93]. Eventually, stubs associated with HT candidates are filtered and fitted, in order to remove fake tracks and to improve the resolution of the reconstructed track parameters. Two filtering-fitting algorithms have been investigated for the final implementation: the Kalman Filter (KF) and the Seed Filter (SF). In particular the latter represents a personal achievement and constitutes a major part of this thesis. Therefore special attention will be dedicated to the explanation of the algorithm and of the results obtained using this fitting approach. The KF will be utilised as an important benchmark tool, also being a verification of the flexibility and reconfigurability of a system based on FPGAs.

The final section of this chapter will show the expected performance of the track finding system, obtained using a simulation framework developed in CMS simulation software framework (CMSSW) [68]. A demonstration system [94, 95], based on currently available hardware, has been also built and it is the topic of the next Chapter.

3.1 A FULLY TIME MULTIPLEXED SYSTEM ARCHITECTURE

Time multiplexing is a well-known concept in particle physics, and it has been already applied in different areas of the CMS experiment. The fundamental idea of a Time Multiplexed Trigger (TMT) [90, 91] is that all data from a single event flow to a single destination (node) for processing. Figure 3.3 shows a comparison between a regionally organised trigger versus a time multiplexed trigger system. The major benefits of a time multiplexed architecture are the extra latency provided, which facilitates the implementation of complex algorithms, and the independence of each single processor, which does not require any data sharing between nodes. Another important advantage is that very few nodes are needed to demonstrate a

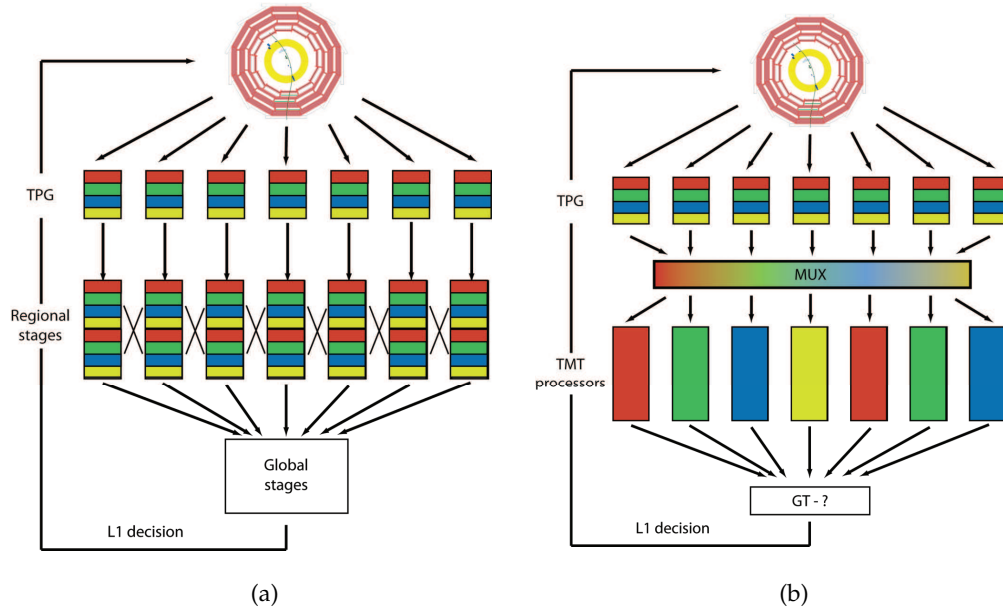


Figure 3.3: Schematic comparison between a regionally organised trigger (a) and time multiplexed trigger (b). Each color in the figures represent a set of data belonging to a single bunch crossing. In a regionally organised trigger data from the Trigger Primitive Generators (TPG) are transmitted to regional processors, which require sharing of data between them to handle boundaries. Results are later merged and a final decision is taken by a global trigger. In a time multiplexed trigger data from all TPGs are transmitted to a single processing node for each BX. A multiplexing stage is therefore required and can be a serial interconnection from each TPG to each TMT processor. [90]

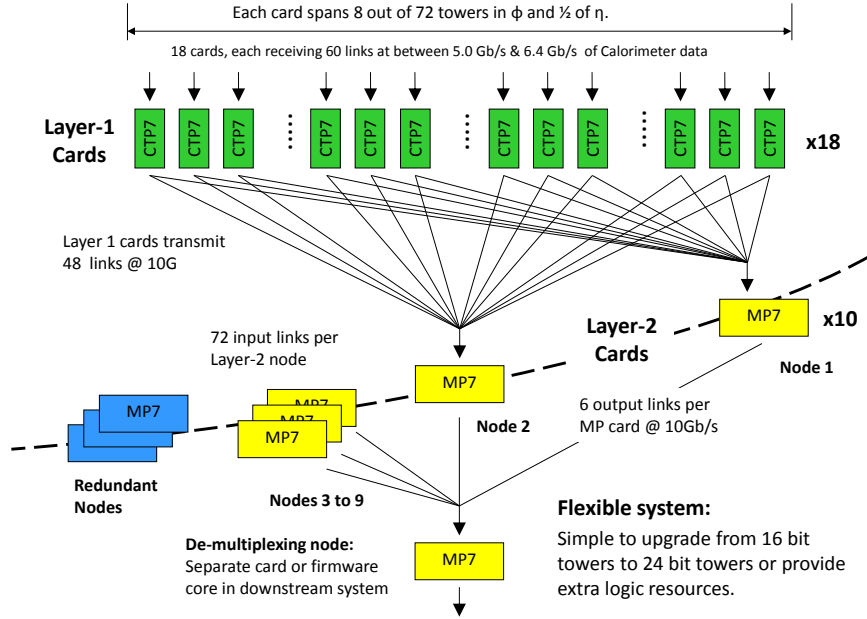


Figure 3.4: High-level view of the Stage-2 trigger architecture.[89]

full trigger system since in a time multiplexed system all nodes are identical and run the same algorithm.

A TMT requires two processing layers with a switching network between them, which can be a dynamic traffic scheduler, as in the CMS HLT [96], or a passive optical fibre network like the one used in the Stage-2 of the upgraded L1 Calorimeter Trigger [66]. In particular the latter represents an important proof of the concept since it is currently operating successfully.

The time multiplexing architecture of the Stage-2 L1 Calorimeter trigger is shown in figure 3.4. The first processing layer, Layer-1, is responsible for the pre-processing and for the formatting of the data. All data from a given LHC bunch crossing are then transmitted to single processing node in Layer-2. Here particle candidates and global energy sums are built or computed. Finally, data is passed to a demultiplexer board, that formats them before sending to the Global Trigger.

The chosen electronics for the Stage-2 L1 Calorimeter trigger is based on microTCA boards equipped with Xilinx Virtex-7 FPGAs [84]. The first layer is formed by CTP7 cards [85], each processing a smaller region of the detector. Layer-2 implements instead MP7 processing boards [97], each of them having access to the full

event data. A total of eighteen CPT7s and ten MP7s (nine as Layer-2 processors and one as the demultiplexer) are currently in use at the Stage-2 calorimeter trigger. Clock and timing signals are distributed to the system by AMC13 [98] cards, which take also the L_1 trigger decision and perform readout to the DAQ system.

3.1.1 An FPGA-based Time Multiplexed Track Finding system

The same TMT concept can be applied to a future L_1 track finding system. As described in Section 2.1.3.2, stub data produced in the outer tracker will be pre-processed by the Data, Trigger and Control (DTC) boards and then transmitted to the downstream L_1 track finding layer (Figure 3.5).

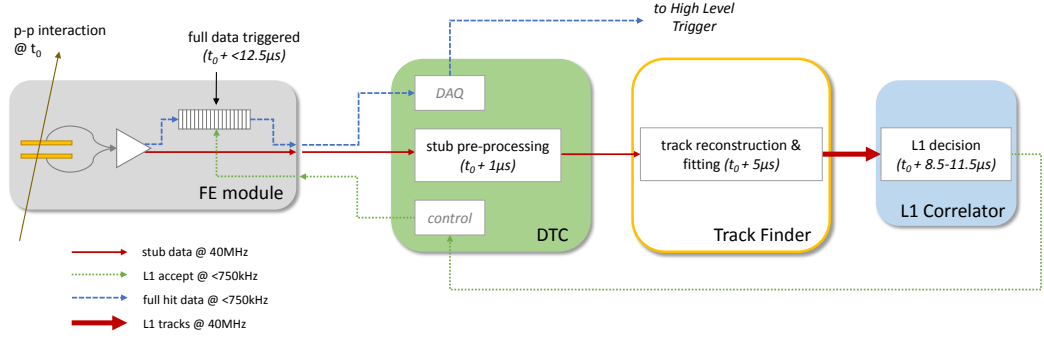


Figure 3.5: Illustration of the data flow from p_T -module front end through the off-detector electronics, which forms the L_1 trigger decision [99]

A TMT can be then designed by considering the DTCs as the first pre-processing stage. Each DTC board will receive stub data belonging to several modules up to a proposed maximum of 72 [74]. Stub data will be encoded in different data formats, depending on the p_T module type. DTCs will receive information about the centroid of the seed cluster in the stub (in half-strip units), written in an 11-bit address, plus the stub *bend* information, defined as the distance in strips between the two clusters in the stubs, related to the local bend of the particle trajectory, which will be encoded in a 3-bit (PS) or 4-bit (2S) word.

Several possible configurations have been proposed as to how to cable the front-end electronics to the back-end system. For the purpose of this thesis, it has been assumed that a set of 32 DTCs will process all data from an approximate octant (i.e. $45^\circ\phi$ -sector) of the outer tracker, for a total number of 256 DTCs board to serve

the whole detector. Since the tracker layout does not have an eight-fold symmetry, these *detector octants* do not have uniform boundaries, as shown in Figure 3.6a.

Similarly to what happens in the Stage-2 calorimeter trigger, data from a single event should be streamed into a single processing node. However, the maximum number of input links and the total bandwidth of a single *FPGA* cannot handle the huge amount of data produced by the entire tracker. Therefore it is necessary to have more than one processing card, which will be responsible for the track finding task in the pre-defined detector region for a single *BX*.

This second processing layer will be formed by *FPGA*-based Track Finder Processor (*TFP*) boards, each of them receiving a full set of stubs from a single octant and from a single event. Therefore, the total number of *TFP* boards needed per octant would be equal to a fixed time-multiplexed factor n , with each board processing a new event every $n \times 25$ ns.

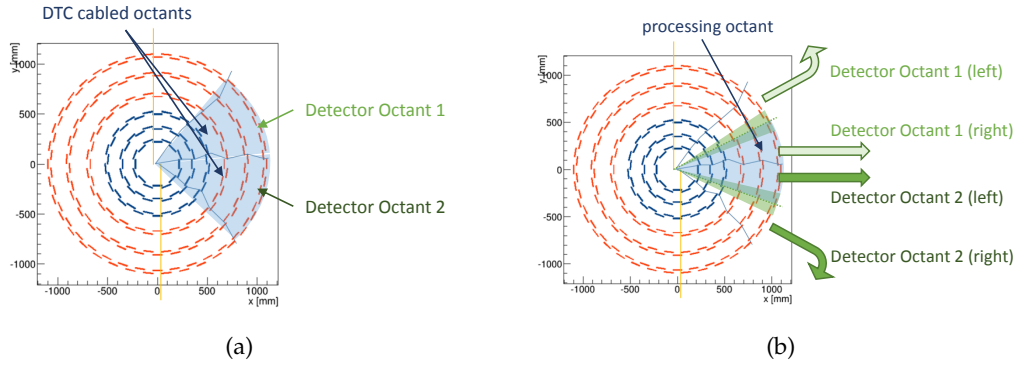


Figure 3.6: (a) Illustration of the DTC cabled octants, also referred as detector octant. Data from each detector octant is readout by 32 *DTC* boards. Since the outer tracker does not present an eight-fold symmetry in the $r - \phi$ plane, detector octants do not present uniform boundaries. (b) Processing octants divides the tracker into eight equal-size 45 degree sectors in ϕ . Their boundaries are rotated approximately 22.5 degrees with respect to the detector octant.

3.1.1.1 Mapping between DTCs and TFPs

Each *TFP* board will be responsible for identifying tracks in particular sub-region of the tracker. A first option would be to assign a single *TFP* to individual detector octant. However, this approach would require the transmission of data into the *TFP* also from the *DTCs* assigned to the two neighbour octants, in order to avoid losses of tracks at the boundaries.

Another possibility exploits a different mapping scheme, which simplifies the stub duplication task. New *processing octants* might be then defined (Figure 3.6b) in a way that they divide the tracker into eight equal-size 45-degree sectors in ϕ . The processing octants boundaries are rotated approximately 22.5 degrees with respect to the physical detector octant. Each TFP will then be assigned to a single processing octant and will now receive data from two DTC boards.

The task of the DTC will be to convert the stubs information into a new global coordinate system, encoded in a 48-bits stream. A detailed description of the adopted data format will be provided in 4.1.2. The stub is then assigned to one of the two processing octants, or in case it is consistent with a track that could be contained in either octants, it will be duplicated.

Data belonging to a single processing octant will be transmitted on a dedicated set of links into a single TFP, without any further downstream data sharing or duplicate removal required after the track finding process. In this way, the whole architecture becomes easily scalable, since each TFP would be designed to operate independently from the others.

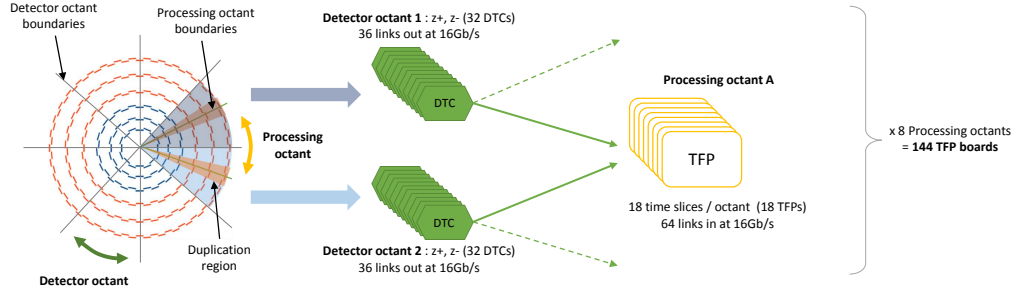


Figure 3.7: The baseline TMTT system architecture. Time-multiplexing and duplication of stubs at the boundaries are performed in the DTCs, which transmit data to the Track Finder Processors (TFPs). With a baseline time multiplexed period of 18, a total of 144 TFPs are required to build the entire track finding system. [99]

Figure 3.7 illustrates the baseline for a the TMTT system architecture. Each DTC will transmit out a total data rate of 600 Gb/s, corresponding to around 36 links at 16.3 Gbps¹. This large bandwidth is needed in order to handle high-occupancy fluctuations in the innermost tracker layers at 200 PU, though the average rate of stubs per DTC will be much lower.

¹ Maximum transceiver speed a Kintex Ultrascale FPGA can handle. [100]

Since each **DTC** will have to transmit data to two processing octants, a maximum time multiplexing period $n_{tm} = 36/2 = 18$ is achievable. Therefore each **TFP** will process stubs from a single processing octant every 450 ns. The total number of **TFPs** boards will be thus equal to

$$N_{TFP} = n_{tm} \times n_{octants} = 18 \times 8 = 144. \quad (13)$$

A time-multiplexed design gives also the possibility of developing new algorithms in parallel during the data taking. This can be achieved simply by incorporating extra nodes, which can receive data duplicated from $1/n_{tm}$ events without affecting the performance of the rest of the system. Additional nodes can be incorporated for redundancy.

3.2 L1 TRACK FINDER PROCESSOR

The designed track finding algorithm is composed of several stages. This simplified both the firmware development and the construction of a hardware demonstrator system based on currently available hardware since each step of the algorithm can be implemented in a separate logic component. The algorithm chain is composed by:

- the **Geometric Processor (GP)**, which pre-processes the stub data for the downstream stages. Stubs are assigned to finer sub-sectors in η and ϕ , with respect to the processing octant, to simplify the track-finding task and increase parallelisation;
- the **Hough Transform (HT)**, which performs the first step in the track finding chain, collating sets of stubs corresponding to a track hypothesis in the r - ϕ plane with coarse resolution, in order to reduce the combinatorial background;
- the **Track Filter and Fitter (TFF)**, which removes the stubs wrongly assigned to track candidate by the **HT** and performs a precision track fitting algorithm, as a means to reduce the fake track rate;

- the **Duplicate Removal (DR)**, which implements the final cleaning stage, exploiting the precise fit information to reject possible duplicate tracks produced by the Hough Transform.

Figure 3.8 shows a simplified flow of the data into the track finding algorithm chain.

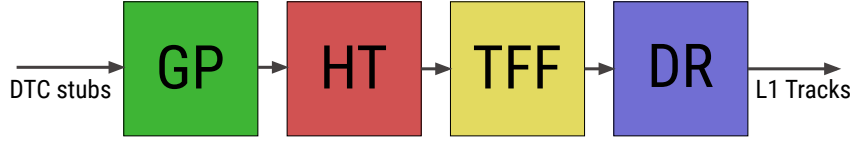


Figure 3.8: The TMTT track finding algorithm chain. Input stubs from the DTCs are re-formatted and assigned to finer sub-sectors by the Geometric Processor (GP). An Hough Transform (HT) is then applied to each sub-sectors to find track candidates in the $r - \phi$ plane. The Track Filter and Fitter (TFF) cleans the candidates and computes the track parameters. Finally a Duplicate Removal (DR) rejects duplicate tracks produced by the HT. [99]

3.2.1 Geometric Processor

The first step in the TFF chain is constituted by the Geometric Processor (GP). It will take as input the 48-bit stubs coming from the DTCs and it will repack them into a more useful 64-bit format, containing additional parameters computed by the GP that will be useful for the downstream processing stages. Each stub will be also assigned to specific geometric sub-sectors of the considered processing octant.

Figure 3.9 shows the sector segmentation adopted. Each octant is sub-divided into 36 sub-sectors, slicing the whole tracker area in 16 parts in ϕ and 18 parts in η . This additional division allows the performance of track finding stage independently and in parallel for each sub-sector. Since charged particles will draw a straight line trajectory in the r - z plane, the sector width in pseudorapidity can be relatively small.

Each (ϕ, η) sector is defined with respect to ϕ_T and z_S coordinates, that correspond to the coordinate of a track trajectory relative to the point where it crosses

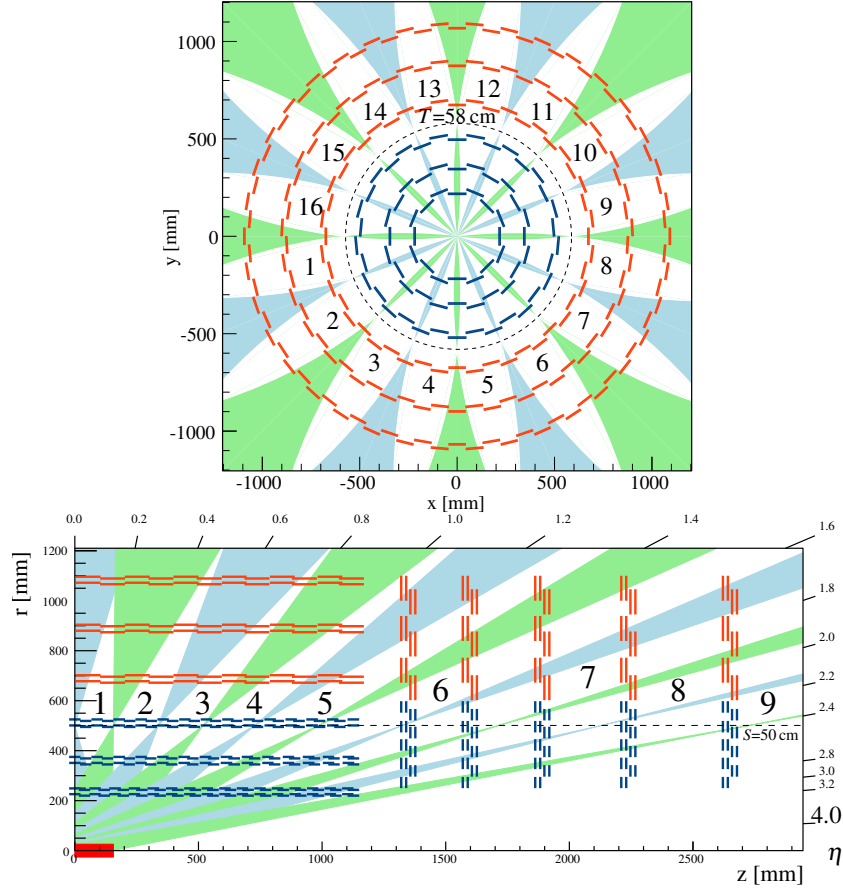


Figure 3.9: The chosen tracker segmentation on the r - ϕ (top) and r - z (bottom) planes. The numbered areas in white are the regions associated to a single sector, whereas coloured regions delineate the overlaps between neighbouring sub-sectors. The dashed lines illustrate the two cylinders of radius $T = 58\text{ cm}$ and $S = 50\text{ cm}$. [99]

a cylinder of radius T or S , centred on the beam-line. The values of T and S have been chosen in order to minimise stub duplication between neighbouring sectors,

$$T = 58 \text{ cm}, \quad S = 50 \text{ cm}. \quad (14)$$

Sector sizes have been chosen to optimise the stub occupancy in each sector. This results in sector equally sized in r - ϕ and varying in r - z . As shown in Figure 3.12a, the average number of stubs per sector is about 90 in $t\bar{t}$ events at PU200. Even if this number is relatively small, it is important to take into account the large tail of the distribution. In fact, sectors containing a large number of stubs are likely to include tracks belonging to highly energetic jets, and thus to interesting physics collision.

A stub with coordinates (r, ϕ, z) is assigned to a certain sub-sector, with boundaries in r - z equal to $(z_S^{\min}, z_S^{\max})$, only if it fulfills the following equation,

$$\frac{r \cdot z_S^{\min}}{S} - w \cdot \left| \frac{r}{S} - 1 \right| < z < \frac{r \cdot z_S^{\max}}{S} + w \cdot \left| \frac{r}{S} - 1 \right|, \quad (15)$$

where $w = 15 \text{ cm}$ is the half-width of the luminous region along the beam axis z .

In addition to this, a stub has to be compatible with the sector in the r - ϕ plane. Let us consider a sub-sector i centred on

$$\phi_i = \frac{2\pi i}{N_\phi}, \quad (16)$$

where $N_\phi = 16$ is the total number of segmentations of the tracker in ϕ and $0 \leq i < N_\phi$. A stub belongs to a particular sector only if it can be assigned to a track that lies completely in that particular sector. The kinematics of a charged particle of charge q with transverse momentum p_T in a magnetic field B with radius of curvature R satisfies by this equation,

$$p_T = qcBR, \quad (17)$$

where c is the speed of light. For any stub (r, ϕ, z) on the track,

$$R = \frac{r}{2 \sin \Delta\phi}, \quad (18)$$

where $\Delta\phi$ is the difference between the azimuthal angle of the stub ϕ and the production angle of the track Φ_0 . Figure 3.10 illustrates the used the utilised track parameters in the r - ϕ plane. For small angles it is possible to approximate $\sin \Delta\phi \sim \Delta\phi$. Hence, the maximum allowed $\Delta\phi^{\max}$ is obtained for tracks of charge q with $p_T = p_T^{\min}$,

$$\Delta\phi^{\max} = q \cdot \frac{cB}{2p_T^{\min}} \cdot r. \quad (19)$$

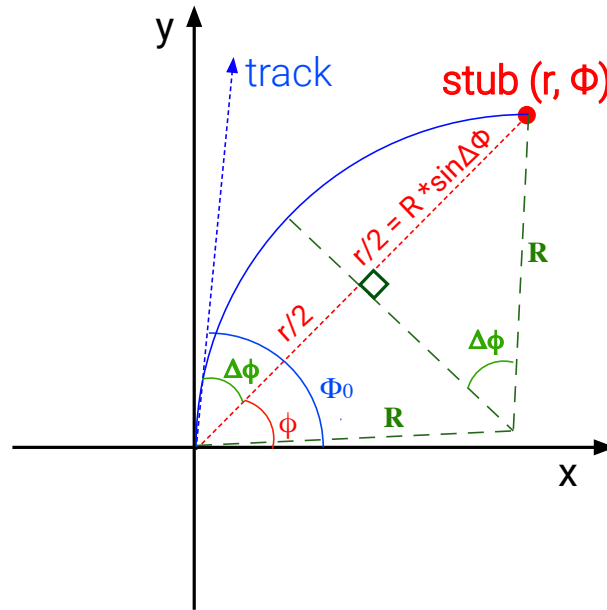


Figure 3.10: Illustration of the track and stub parameters in the r - ϕ plane.

Since sub-sectors are defined with respect to a radial position T , it is useful to write,

$$\Delta\phi_T^{\max} = q \cdot \frac{cB}{2p_T^{\min}} \cdot |r_T|. \quad (20)$$

Eventually, the sub-sector ϕ condition is,

$$|\phi_i - \phi| < \frac{\pi}{N_\phi} + \Delta\phi_T^{\max}, \quad (21)$$

where $\frac{\pi}{N_\phi}$ is the sector half-width. It is possible to place another criterion, exploiting the stub local bend measurement b , defined as the distance in strips between the two clusters in the stubs in units of the strip pitch. In fact the bend constrains the allowed q/p_T range of a track,

$$\frac{2\rho \cdot (b - k_b)}{cBr} < \frac{q}{p_T} < \frac{2\rho \cdot (b + k_b)}{cBr}, \quad (22)$$

where $\rho = (p/s)$ for barrel stubs and $\rho = (p/s) \cdot (z/r)$ for endcap stubs, and p and s are the pitch and separation parameters of the two sensors in a module. The *flat* barrel layout allows only eight different values of (p/s) . k_b is configurable cut parameter, which takes into account the bend resolution that has been assumed to be the same for the whole tracker and it is around $\sqrt{2/12}$. The default value k_b is then set to 1.25, approximately three standard deviations. Figure 3.11 shows a schematic view of the local stub parameters and their relation with the global track coordinates.

Therefore, the maximum allowed $\Delta\phi^{\max}$ curvature can now be expressed as,

$$\Delta\phi_T^{\max} = \rho \cdot (b + k_b) \frac{|r_T|}{r}, \quad (23)$$

leading to another criterion for the sub-sector,

$$|\phi_i - \phi| < \frac{\pi}{N_\phi} + \rho \cdot (b + k_b) \frac{|r_T|}{r}. \quad (24)$$

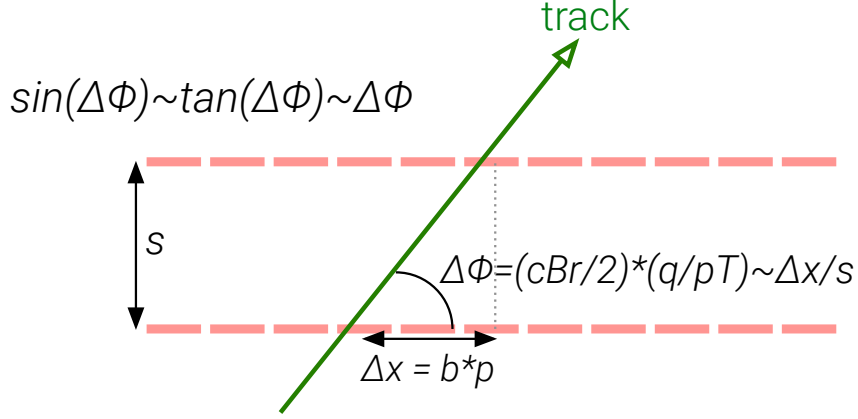


Figure 3.11: Illustration of the stub local parameters and their relation with the global track parameters in the r - ϕ plane.

Requiring both conditions 21 and 24 reduces the rate of stubs duplicated across the sub-sectors (Figure 3.12b). In a typical $t\bar{t}$ event with PU200 the GP assigns a stub to 1.8 sub-sectors on average.

3.2.2 Track finding with Hough Transform technique

The Hough Transform (HT) [92] is a method widely used in high-energy physics, as well as in other fields, e.g. digital imaging processing [101]. The simplest version of this technique permits detection of straight lines. In the real space, a line is defined as a set of points (x, y) , which the HT represents in terms of the slope-intercept parameters (m, c) .

$$y = mx + c \Rightarrow c = -xm + y \quad (25)$$

In this way, each point in (x, y) real space represents a line in (m, c) parameter space, so that a line in the real space will be identified by the crossing point in the parameter space, corresponding to the exhaustive set of lines that can possibly pass through this point.

In the TMTT track finding algorithm, the HT is applied to the stub coordinates (r, ϕ) , which have the best resolution in CMS. An independent HT is used for each of the sub-sectors defined by the GP.

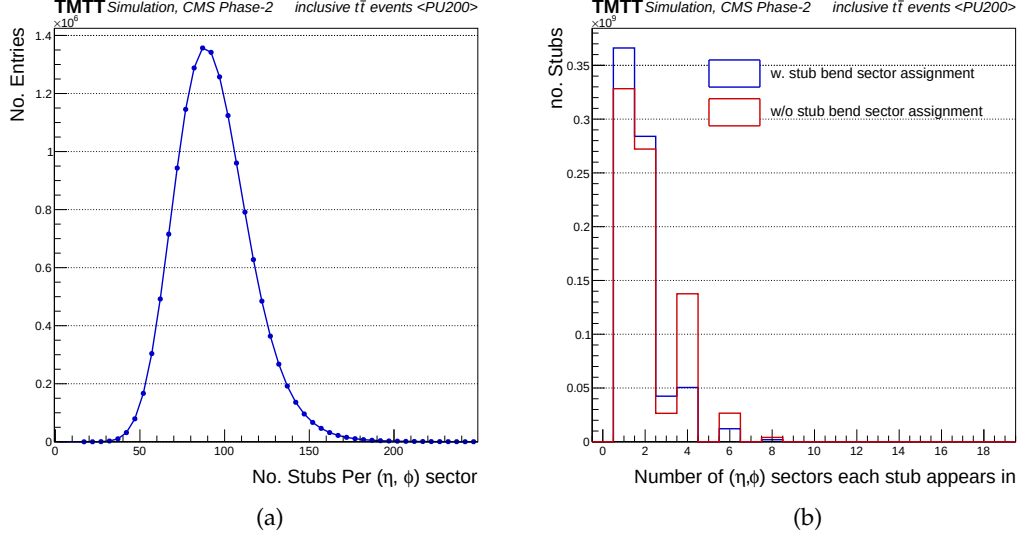


Figure 3.12: (a) Number of stubs assigned to each (η, ϕ) sub-sector, in top-quark pair production events with pileup 200. (b) Number of sub-sectors each stub is assigned to. Stubs assigned to sectors without (with) the use of the bend information are shown with a red (blue) line. Usage of the bend information reduces significantly the rate of duplication of stubs between sectors.

A charged particle travelling through the CMS magnetic field is bent with a radius of curvature,

$$R = \frac{p_T}{cB}. \quad (26)$$

Each coordinate (r, ϕ) on the particle trajectory has to fulfil the following equation,

$$\frac{r}{2R} = \sin(\phi - \Phi_0) \sim \phi - \Phi_0, \quad (27)$$

where Φ_0 is the production angle of the particle at the origin. The small angle approximation is considered to be valid for tracks with $p_T > 2 \text{ GeV}$.

Combining the two equations, gives [93, 95],

$$\Phi_0 = \phi - \frac{qcB}{2p_T} \cdot r. \quad (28)$$

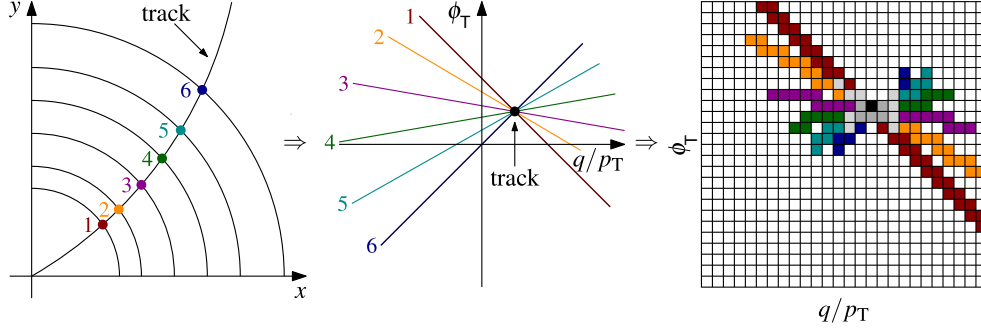


Figure 3.13: The Hough Transform track finding algorithm. Left: Trajectory of a single particle in a quarter of the tracker barrel in the x - y plane. The numbers label the stubs produced by the passage of the particles through the tracking stations. Middle: The produced stubs draw straight lines in the HT space, where the axes correspond to the track parameters ($q/p_T/\phi_T$). Intersection point identifies a track candidate. Right: The same Hough-space discretised with an array of 32×64 cells. Stubs fill a cell only if their local- p_T estimation is compatible with the bin momentum. [95]

This relation maps the stub coordinates (r, ϕ) in real space to the track parameter $(q/p_T, \Phi_0)$ in parameter space. Stubs produced from the same particle will intersect at a single point in transform space, in the case that coordinates were measured with perfect precision and in absence of secondary effects like multiple scattering. The gradient in eq. 28 is always positive, since it is proportional to the radius of the stub r . It is more convenient therefore to apply the Hough Transform to the translated space (r_T, ϕ) , where $r_T = r - T$. In this way, negative slope parameters are allowed and a more uniform filling of the HT array is achieved. The HT relation then becomes,

$$\Phi_T = \phi - \frac{qcB}{2p_T} \cdot r_T, \quad (29)$$

where Φ_T is simply the azimuthal coordinate of the track relative to the point where it crosses a cylinder of radius T .

Figure 3.13 shows how the Hough Transform builds the track candidates. The parameter space $(q/p_T, \Phi_T)$ is subdivided into an array of 32×64 cells; the horizontal axis q/p_T is limited by the choice of $p_T^{\min}$, $|q/p_T| < q/p_T^{\min}$, while the

vertical axis Φ_T coincides with the sector Φ_T range. For each stub a value of Φ_T is calculated per each bin in q/p_T , filling the respective cell. Once all stubs have entered the array, track candidates are selected by identifying cells that have stubs in a minimum number of tracker layers or disks. This threshold has been set to five layers, except in the pseudorapidity range $0.89 < |\eta| < 1.16$ where it has been reduced to four in order to take into account the acceptance gap between barrel and endcaps.

Finally, to reduce the total number of *activated* cells, hence the possible duplicate tracks, a stub enters a cell only if the cell q/p_T is in the range allowed by the stub bend measurement, as in eq. 22. To reduce further the total number of stubs to be transmitted to the downstream stages, a limit on the maximum number of stubs per cell has been imposed and it has been set to 16. Any stub entering the cell afterwards would be neglected. Figure 3.14 shows the obtained reduction in terms of stubs per each (η, ϕ) sector. On average only about 10% of the input stubs survive the Hough Transform while maintaining an overall tracking efficiency above 96%.

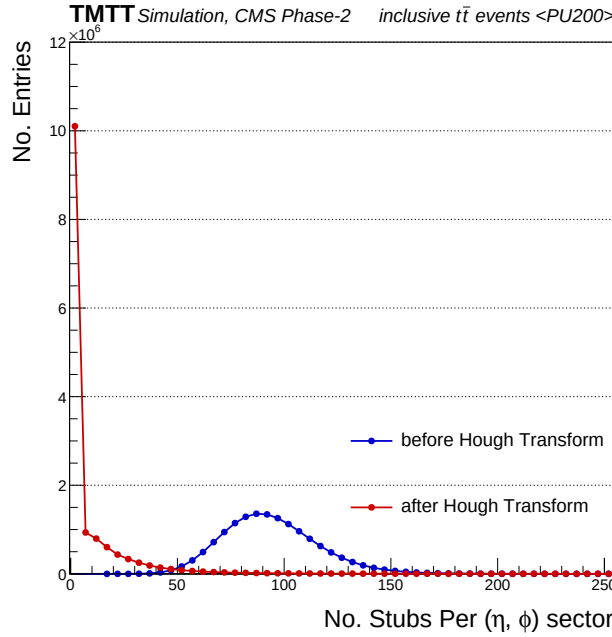


Figure 3.14: Number of stubs belonging to track candidates at the output of the HT, per each sub-sector. A data reduction of about 90% is achieved.

3.2.3 Track filtering and fitting

While the HT significantly reduces the number of stubs to be processed and gives a rough estimation of the q/p_T and Φ_0 parameters of the tracks, it does not provide a measurement in the r - z plane. This means that many of the track candidates built by the Hough Transform may not have a straight line trajectory in the r - z plane. Indeed about 45% of the HT tracks do not match any truth particle, as defined in Tab. 2.2. In addition, most of the matched tracks ($\sim 60\%$) still have extra stubs assigned, which, if not removed, can invalidate the final track fit measurement (Fig. 3.15).

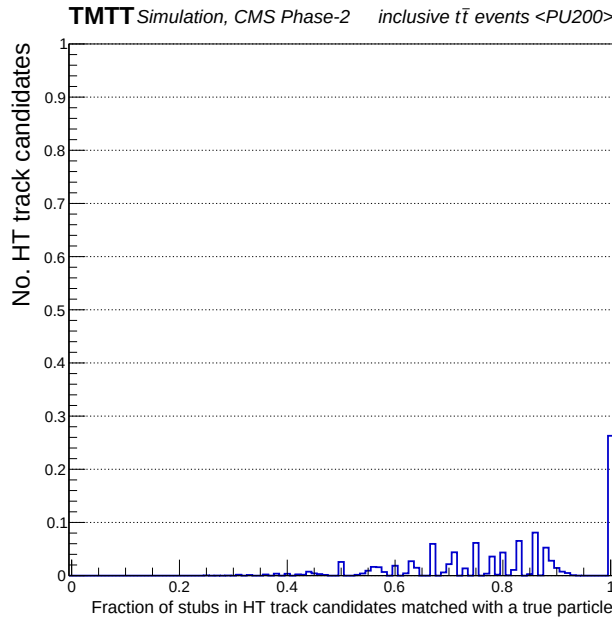


Figure 3.15: Fraction of stubs in a HT track candidate that have been produced by the same particle. A large portion of HT candidates still contain spurious stubs from pileup interactions. A filtering stage is therefore needed to clean the tracks before fitting.

For these reasons, it is necessary to implement a further cleaning stage that keeps only the best stubs for the final fit. Several algorithms have been investigated and two of them have been found to give similarly good performance:

- the **Seed Filter plus Linear Regression** is a two-stage algorithm. In the first step, the HT track candidates are *cleaned*, by keeping only one stub per tracking station and checking the compatibility of the track with a straight line

hypothesis in the r-z plane. If this hypothesis is satisfied, the surviving stubs are then used to perform a fit with a simple linear regression technique;

- the **Kalman Filter** is a single stage algorithm, performing both track filtering and fitting tasks in the same process, starting with a coarse estimation of the track parameters.

The following paragraph will provide a brief overview of the Kalman Filter algorithm, while the Seed Filter plus Linear Regression approach will be discussed in detail in Section 3.3.

3.2.3.1 Kalman Filter

The Kalman Filter (KF) [102] is an iterative algorithm that uses a series of measurements to estimate unknown parameters by using Bayesian inference and estimating a joint probability distribution over the variables for each new measurement. The filter is named after Rudolf E. Kalman, one of the primary developers of the theory.

The Kalman filter permits to estimate the evolution of a dynamic system, assuming that the state $\mathbf{x}_t$ of the system at time t is linked to state $\mathbf{x}_{t-1}$ in which the system was at a time $t - 1$,

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t, \quad (30)$$

where $\mathbf{u}_t$ is control input vector; $\mathbf{F}_t$ is the state transition matrix, which relates that state of the system at the time $t - 1$ to the new state at t ; $\mathbf{B}_t$ is the control input matrix, applying the effect of the control input vector $\mathbf{u}_t$ to $\mathbf{x}_t$; and $\mathbf{w}_t$ is the vector containing the noise terms for each element in $\mathbf{x}_t$ [103].

The implemented algorithm runs over the stubs from each individual HT track candidate, starting from an estimate of the track parameters and their uncertainties (*state*). Each stub is seen as an independent measurement and it is used to update the state and to decrease the uncertainty. Track parameters are updated for each new state, taking into account the relative uncertainties of the measurements.

The KF calculates for each set of stubs four track parameters: the inverse of the radius of curvature R , as defined in eq. 26, the azimuthal angle of the track in the transverse plane at the beam-line Φ_0 , the polar angle θ and the longitudinal impact

parameter z_0 . For tracks originating from the beam-spot ($r = 0$), the following equations are valid,

$$\phi = \frac{r}{2R} + \Phi_0 \quad (31)$$

$$z = \cot \theta \cdot r + z_0. \quad (32)$$

Since both relations are linear in r , it is natural to use the radius as a stepping parameter for the **KF** over the entire tracker. In the endcaps, where modules follow an orthogonal orientation, the uncertainty in r due to the strip length is folded into the z uncertainty using $\sigma_z^2 = \sigma_r^2 \cdot \cot^2 \theta$.

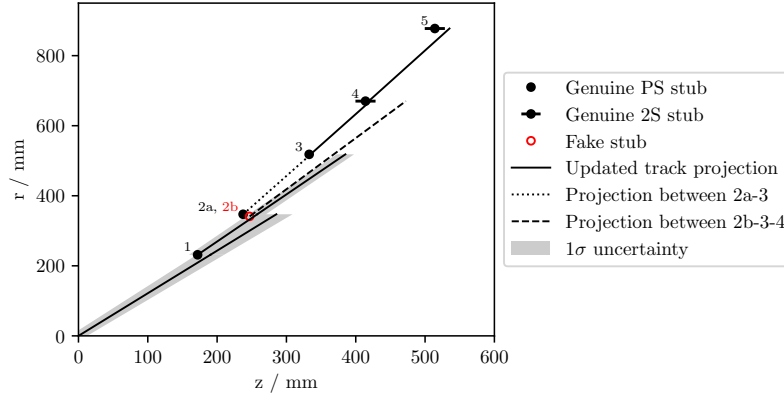


Figure 3.16: A **KF** example procedure for a track candidate in the barrel, shown in the r - z plane. Each step in the fitting procedure is represented by a line segment in the plot. The fitting line is updated at each iteration with increasing radius. Dashed track segments highlight the different results fitting with stub 2a or 2b. State produced with latter is rejected after being extrapolated to stub 4. [99]

A **KF** fitting procedure for an example track candidate in the barrel is shown in Figure 3.16. The first state is obtained from the **HT** array coordinates (q/p_T , Φ_T) and the η sub-sector in which the track has been found. The starting state is updated then using stubs ordered by their radius from inside-out. To take into account possible detector inefficiencies or absence of a compatible stub on a given tracker station, up to two non-consecutive layers may be skipped. If two stubs on a given layer are found to be compatible, each combination of stub and state is propagated separately. The best state is then selected as the one with fewer

missing layers and with the smallest χ^2 , and it is finally used to calculate the track parameters.

3.2.4 Duplicate removal

The final step in the [TMTT](#) algorithm chain is constituted by the Duplicate Removal ([DR](#)). At this stage, half of the track candidates are unwanted duplicate tracks, which have been produced by the [HT](#). As shown in [Figure 3.17](#) duplicate tracks are generated because of the coarse array resolution of the Hough Transform, which is needed to maintain a high tracking efficiency. The same set of stubs can, in fact, *activate* more than one cell in the array. When these [HT](#) track candidates are fitted, they produce the same track fit parameters, which should correspond to the point where the stub lines actually intersect in the transform space.

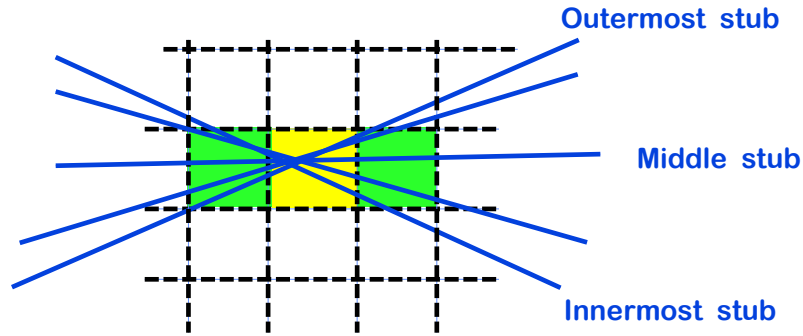


Figure 3.17: Formation of duplicates in the [HT](#). The yellow cell represents the genuine candidates and the green cells the duplicate tracks produced by the same set of stubs. [99]

The [DR](#) algorithm exploits the fitted parameters to remove the unwanted duplicates. If the fitted parameters of the track do not correspond to the ones found by the [HT](#), the track is rejected. Once all tracks have been processed, a second pass through the rejected tracks is then performed, in order to avoid efficiency loss due to resolution effects. In this second step, the algorithm retrieves the tracks whose [HT](#) coordinates do not correspond to the ones of a track that passed the first cycle.

3.3 SEED FILTER AND LINEAR REGRESSION ALGORITHM

A second proposal for the track fitting task is represented by the **SF+LR** algorithm. This two stages-algorithm cleans up firstly the track candidates produced by the Hough Transform, by checking their compatibility with a straight line trajectory in the r - z plane, and later performs a track fit based on the linear regression technique.

3.3.1 The Seed Filter

It is possible that the **HT** produced candidate containing stubs, which lie tens of centimetres from the real track in the r - z plane. The goal of the Seed Filter (**SF**) is therefore to identify and eliminate those stubs. The **SF** process is illustrated in Figure 3.18 for an example candidate in the barrel (η region 1). The algorithm process can be summarised as follows:

1. **Collecting seeding stubs (Fig. 3.18a)**: all stubs from PS modules in the candidate, belonging to the three innermost layers in the barrel or to the four innermost endcap disks (two per side), are collected and marked as first seeding stubs. Second seeding stubs are then selected also including PS stubs in the third disk of the two endcaps;
2. **Creating the seeds (Fig. 3.18b)**: for each pair of seeding stubs s_1 and s_2 with coordinates (r_1, z_1) and (r_2, z_2) a line is computed (*Seed*) in the r - z plane,

$$\frac{z - z_1}{z_1 - z_2} = \frac{r - r_1}{r_1 - r_2}. \quad (33)$$

A limit on the maximum number of pairs n_{pairs} per each first seeding stub s_1 can be set, together with a cut on the total number of seeds n_{seeds} to be considered. The default values for these selection parameters are $n_{\text{pairs}}^{\text{max}} = 3$ and $n_{\text{seeds}}^{\text{max}} = 15$;

3. **Beam-spot and sector filter (Fig. 3.18c)**: seeds are extrapolated both towards the beam-spot and towards the radius S , calculating the values where the

lines intersect the beam-spot axis z_0 and the sector definition radius S , z_S . Only if the following relations are satisfied is the seed kept:

$$|z_0| = \left| z_1 - \frac{(z_1 - z_2) \cdot r_1}{r_1 - r_2} \right| < w, \quad (34)$$

$$z_S^{\min} < z_S = z_1 - \frac{(z_1 - z_2) \cdot (S - r_1)}{r_1 - r_2} < z_S^{\max}, \quad (35)$$

here w is the half-width of the beam spot along z and $(z_S^{\min}, z_S^{\max})$ define the sub-sector covering range, as explained in Section 3.2.1. A limit $n_{\text{good}}^{\max}$ on the number of *good seeds* can be imposed. In the default configuration only the first 10 seeds that pass the beam-spot and sector filter are considered in the next steps of the algorithm;

4. **Stub checking (Fig. 3.18d):** *good* seeds are then extrapolated towards the remaining layers. For each seed, only stubs compatible with the line are kept. This is guaranteed by the following condition,

$$\Delta_{\text{stub}} < \sigma_{\text{stub}}, \quad (36)$$

where, for each stub with coordinates (r, z) ,

$$\Delta_{\text{stub}} = |(z - z_1) \cdot (r_1 - r_2) - (r - r_1) \cdot (z_1 - z_2)| \quad (37)$$

$$\begin{aligned} \sigma_{\text{stub}} = & (\sigma_z + \sigma_{z_1}) \cdot |r_1 - r_2| + (\sigma_{r_1} + \sigma_{r_2}) \cdot |z - z_1| \\ & + (\sigma_r + \sigma_{r_1}) \cdot |z_1 - z_2| + (\sigma_{z_1} + \sigma_{z_2}) \cdot |r - r_1|, \end{aligned} \quad (38)$$

with $\sigma_{r,z}$ being the uncertainty on r, z of the stub position, which is assumed to be equal to half of the module pitch length. Only one stub per tracking layer is kept, keeping only the first one to be found compatible with the seed;

5. **Building candidates (Fig. 3.18e):** only seeds that still have more than four stubs in different stations of the tracker are kept;
6. **Seed Comparison (Fig. 3.18f):** if more than one seed survives the process, only the candidate with fewest missing layers is kept, or, where several candidates have the same number of activated layers, the first one found is kept.

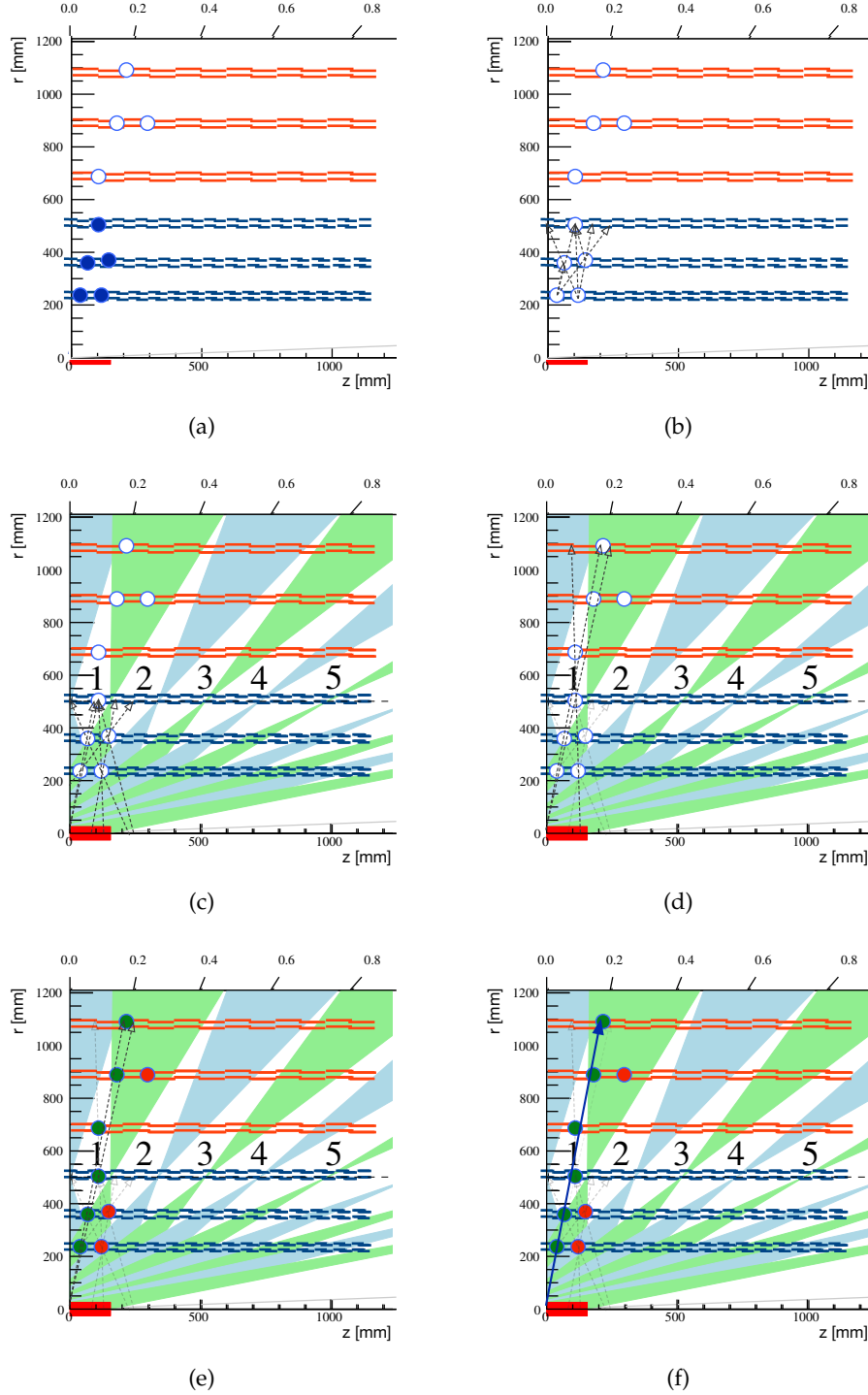


Figure 3.18: An example Seed Filter (SF) procedure for a HT track candidate in the barrel sub-sector 1. (a) Collecting seeding stubs (blue dots). (b) Creating the seeds (grey lines) (c) Only seeds compatible with a track originating from the luminous region (red bar) and the sector definition. (d) Surviving seeds are extrapolated to remaining tracking layers. (e) Only the closest stubs compatible with seeding lines are kept (green dots). (f) Keep only stubs belonging to the seed with the most activated layers.

3.3.2 The Simple Linear Regression Fitter

Track candidates that survive the Seed Filter selection are then fitted via a Simple Linear Regression (SLR) fitter. The fitter is based on the assumption that the particles with enough transverse momentum ($p_T > 2 \text{ GeV}$) draw a straight line trajectory on both r - ϕ and r - z planes. Therefore two independent straight line fits are performed in the two planes. A brief description of the Simple Linear Regression model is presented in Appendix A.

The fitter calculates four track parameters as follows,

$$\frac{1}{2R} = \frac{n\sum_i(r_i\phi_i) - \sum_i r_i \sum_i \phi_i}{n\sum_i r_i^2 - (\sum_i r_i)^2}, \quad (39)$$

$$\Phi_0 = \frac{\sum_i r_i^2 \sum_i \phi_i - \sum_i r_i \sum_i (r_i \phi_i)}{n\sum_i r_i^2 - (\sum_i r_i)^2}, \quad (40)$$

$$\tan \lambda = \frac{n\sum_i(r_i z_i) - \sum_i r_i \sum_i z_i}{n\sum_i r_i^2 - (\sum_i r_i)^2}, \quad (41)$$

$$z_0 = \frac{\sum_i r_i^2 \sum_i z_i - \sum_i r_i \sum_i (r_i z_i)}{n\sum_i r_i^2 - (\sum_i r_i)^2}, \quad (42)$$

where (r_i, z_i, ϕ_i) are the coordinates of the stubs in the track with $1 < i < n$. Different possible implementations of the SLR have been explored:

1. **High-efficiency SLR.** Fit parameters are calculated straight-away without any further check on the stubs. r - ϕ parameters are calculated using all the stubs in the track candidate, while only PS stubs are used to estimate z_0 and $\tan \lambda$;
2. **High-purity SLR.** Track fit parameters are computed first only for in the r - ϕ plane. Then residuals are calculated for each stub and only the four stubs with the smallest distance from the fit line are kept, guaranteeing always the presence of at least two stubs from PS modules. Finally track fit parameters are computed again with the surviving set of stubs, using only PS stubs for the r - z fit;
3. **Best-of-both SLR.** Following the same process as the high-purity SLR, this time removing from the track candidates all stubs that have a residual distance larger than a predefined value ($\sim 5\sigma$). The final fit parameters are then

calculated as in the other two approaches, utilising only PS stubs for the r-z case.

The goodness of the track fit is estimated by calculating independently the χ^2 distribution on both planes:

$$\chi_{r\phi}^2 = \sum_i \left(\frac{d_{r\phi}^i}{\sigma_{r\phi}^i} \right)^2, \quad (43)$$

$$\chi_{rz}^2 = \sum_i \left(\frac{d_{rz}^i}{\sigma_{rz}^i} \right)^2, \quad (44)$$

where d^i is the residual distance of a stub i from the fit line and σ_i is the standard deviation. The equations for the two lines in the r- ϕ and r-z planes are,

$$\phi = \frac{1}{2R} \cdot r + \Phi_0, \quad (45)$$

$$z = \tan \lambda \cdot r + z_0, \quad (46)$$

it is possible to define the residual distances, for a stub i ,

$$d_{r\phi}^i = \phi_i - \frac{1}{2R} \cdot r_i - \Phi_0, \quad (47)$$

$$d_{rz}^i = z_i - \tan \lambda \cdot r_i - z_0. \quad (48)$$

It is useful to define a global χ^2 as the quadratic mean of the χ^2 of the two fits:

$$\chi^2 = \frac{\sqrt{(\chi_{r\phi}^2)^2 + (\chi_{rz}^2)^2}}{2}. \quad (49)$$

Finally a cut on χ^2 of the track can be applied, in order to reject all the candidates which have poor fit parameters.

3.3.2.1 Seed Filter plus Linear Regression simulated results

SEED FILTER. The Seed Filter has been found to have excellent performance in terms of fake track rejection and track cleaning. Figure 3.19 shows the distribution of the fraction of stubs in track candidate that are matched with a truth track, before and after the Seed Filter stage. A cleaning of the track candidates is obtained,

with about 98% of the stubs assigned to a matched track candidate that actually belong to a truth particle.

This is achieved without affecting the track finding efficiency, as shown in Figure 3.20. The excellent track finding performance of the Hough Transform is not lost, with an observed drop in the efficiency of only about 0.9%. Furthermore ~ 96% of the matched tracks after the SF do not have any incorrect stubs assigned to them, easing the work of the subsequent track fitting stage.

SIMPLE LINEAR REGRESSION. Key performance criteria to compare the three linear regression implementations are the track finding efficiency and the resolution of the fit parameters. Figure 3.21 shows the track finding efficiency for all charged particles in inclusive $t\bar{t}$ events with pileup 200 for the three approaches. As expected, the High-Efficiency SLR is the one showing the highest efficiency, on average 0.8% higher than the best-of-both approach. The High-Purity approach presents the lowest tracking efficiency, since sometimes it rejects the wrong stub on a track, violating the four stubs matching criteria.

The high efficiency of the High-Efficiency SLR is obtained by keeping all the stubs in a track candidate without trying to reject stubs not compatible with the track fit, which translates into poorer track parameter resolution. It is useful to define the perfect efficiency as,

$$\epsilon_{\text{perf}} = \frac{N_{\text{FullMatch}}}{N_{\text{Truth}}}, \quad (50)$$

where $N_{\text{FullMatch}}$ is the total number of reconstructed tracks which are matched to a truth particle and do not have any extra stubs from other particles assigned to them. Figure 3.22 shows the distribution of the perfect efficiency for the three SLR approaches as a function of $1/p_T$ and of η . It is easy to observe that the High-Efficiency SLR has the lowest perfect efficiency, meaning that many of the track candidates out of this implementation contain impurities. On the other hand, the High-Purity SLR shows the highest candidate purity, even if the best-of-both approaches presents a very good perfect efficiency above 94%.

The resulting track finding performance for the three approaches are summarised in Table 3.1.

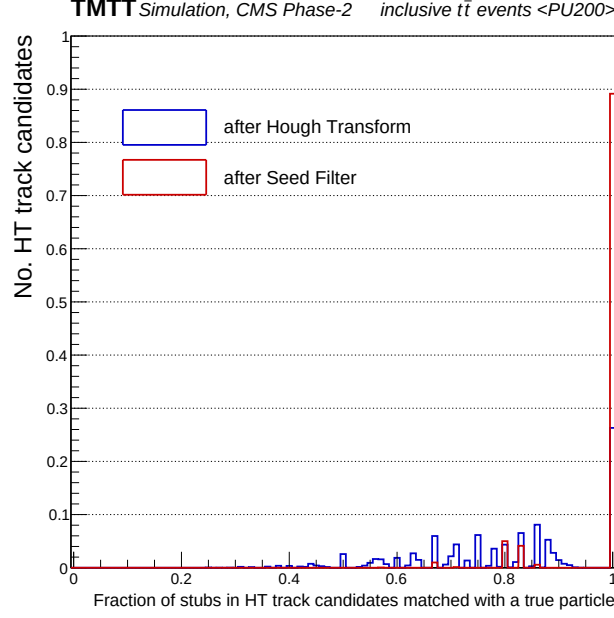


Figure 3.19: Fraction of stubs in a track candidate that have been produced by the same particle before and after Seed Filter. The **SF** shows an excellent cleaning power, with about 91% of the surviving tracks having stubs produced by a single particle.

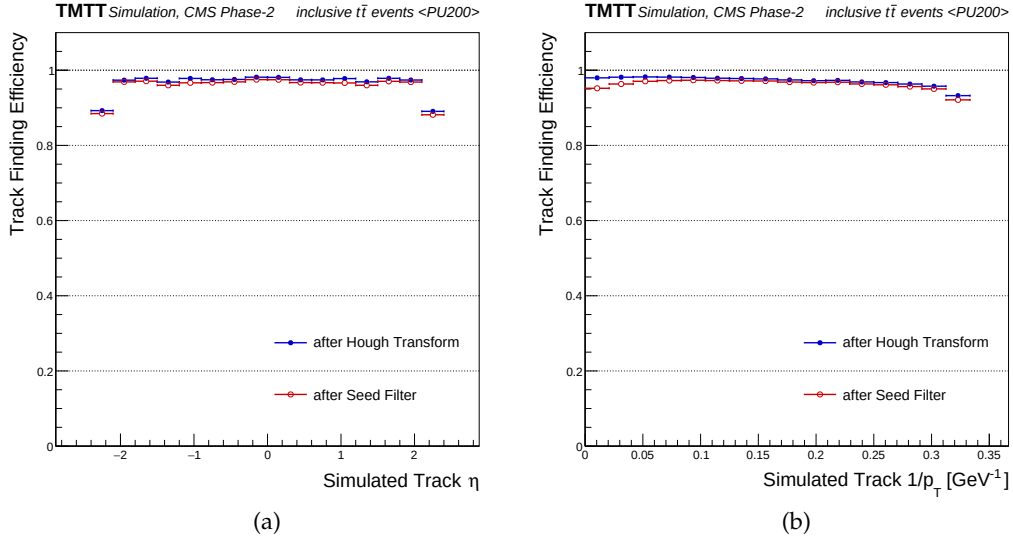


Figure 3.20: Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with **PU200**, before (red) and after (blue) the Seed Filter stage.

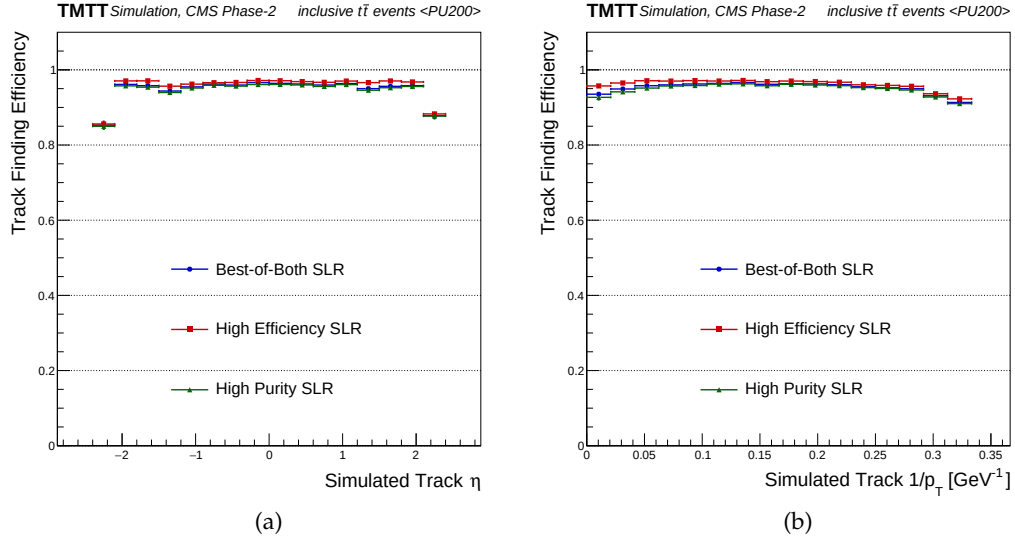


Figure 3.21: Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU200, using a High Efficiency SLR (blue line), a High Purity SLR (black line) or a Best-of-Both SLR fitter.

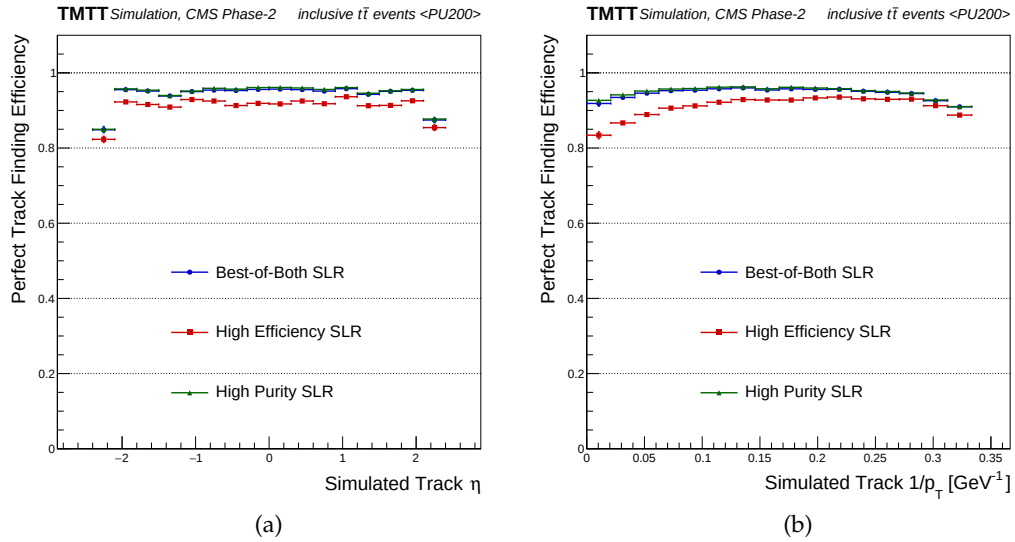


Figure 3.22: Track finding *perfect* efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU200, using a High Efficiency SLR (blue line), a High Purity SLR (black line) or a Best-of-Both SLR fitter.

Table 3.1: Simulated track finding performance in top pair production events at a PU of 200, for the three investigated linear regression fitter methods.

Track Fitting Algorithm	Efficiency	Perfect Efficiency	No. Stubs Per Track
High-Efficiency SLR	96.24%	91.59%	5.34
High-Purity SLR	95.09%	95.09%	4.00
Best-of-Both SLR	95.43%	94.76%	4.70

In order to fully compare the three approaches it is important to measure their performance in terms of track fit parameter resolution. Figures 3.23 and 3.24 present the resolution of the four track parameters as a function of the inverse of transverse momentum and the pseudorapidity of the simulated truth particle, respectively. The resolution of a fit parameter, in this case, is simply calculated as the mean difference between the measured and the generated values. The pseudorapidity η is estimated directly from the measured $\tan \lambda$,

$$\eta = -\log(\tan(0.5 \cdot \arctan(1/\tan \lambda))). \quad (51)$$

The High-Efficiency SLR is the one that shows the worst performance on both r - ϕ and r - z planes. This implies that a further cleaning step is necessary before the final track fitting. The High-Purity and the Best-of-Both approaches present similar resolution in the r - ϕ plane, while the latter shows better performance for the r - z parameters. This result can be explained by the fact that, since z_0 and $\tan \lambda$ are calculated using only stubs from PS modules, sometimes only two stubs are used to calculate the fit parameters by the High-Purity SLR, hence resulting in worse resolution.

In conclusion, the Best-of-Both SLR has been selected as the default track fitting algorithm and it has been used to obtain the results that will be shown in the next section. In order to reduce further the fake rate, it is possible to apply a cut based on track fit χ^2 . Figure 3.25 shows the χ^2 distribution for tracks that match and that do not match with a truth particle. Based on this distribution a cut at $\chi^2/\text{d.o.f.} < 20$ has found to be a good compromise between track finding performance and fake rate reduction.

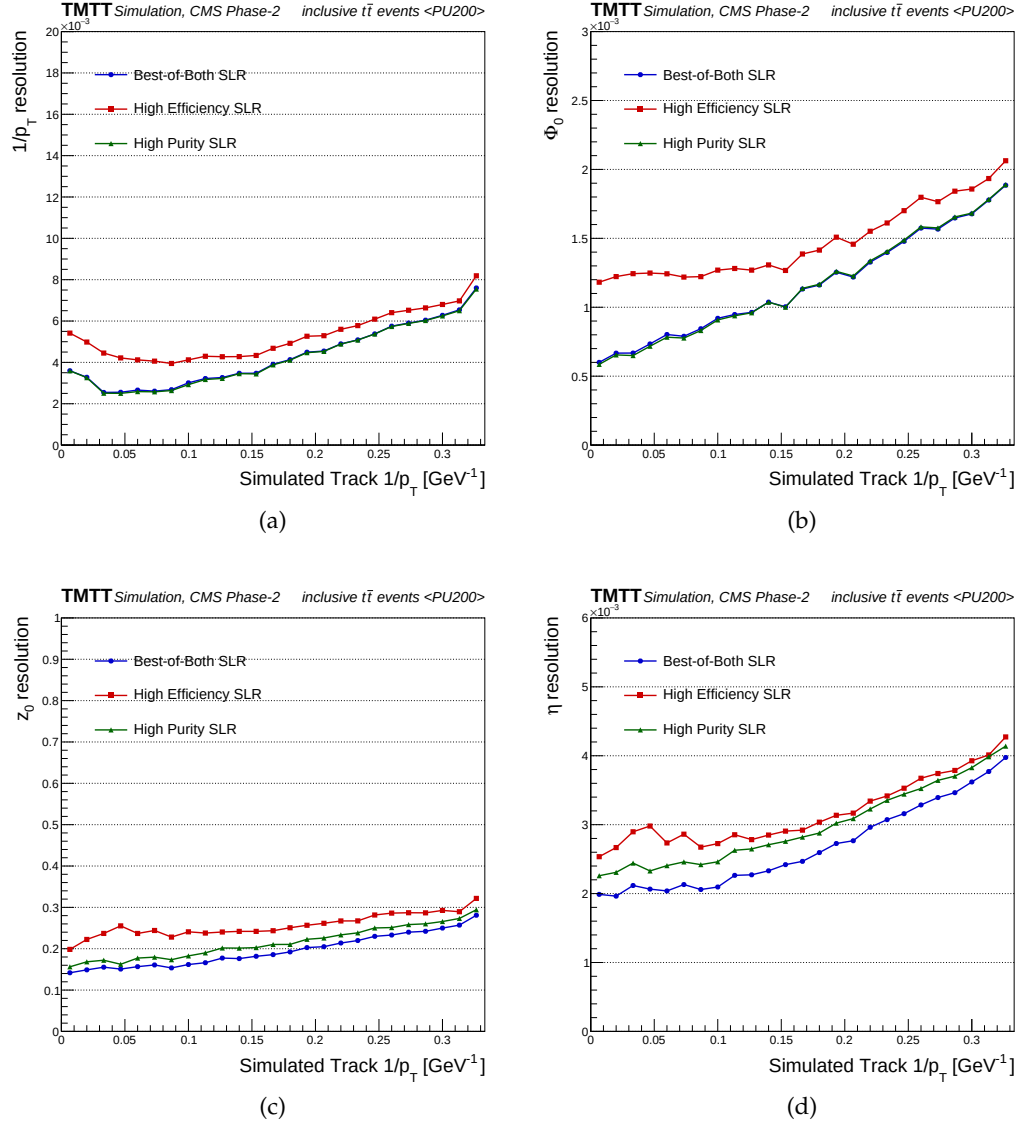


Figure 3.23: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. Results obtained with the three different SLR fit versions are compared.

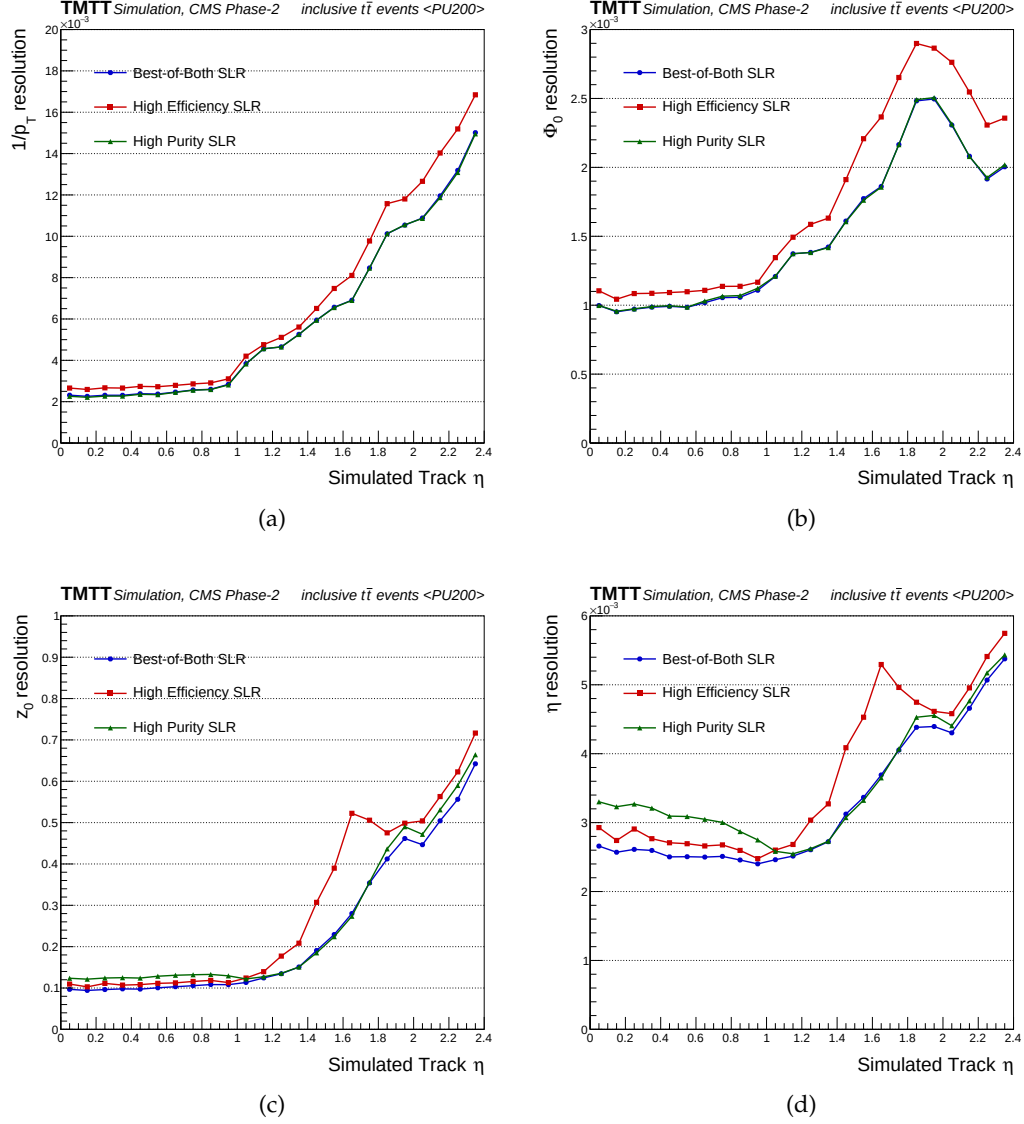


Figure 3.24: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with the three different SLR fitter versions are compared.

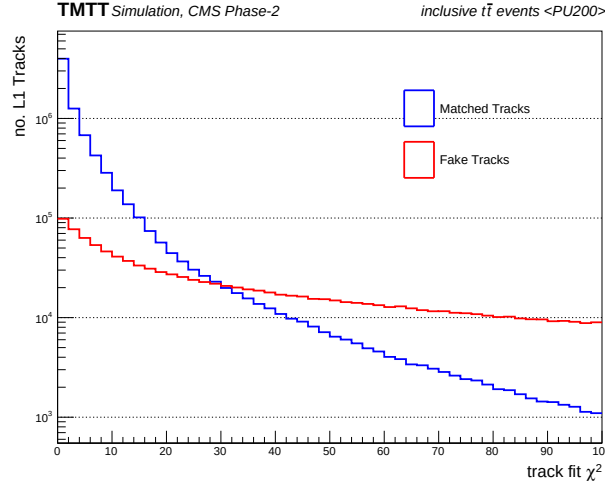


Figure 3.25: χ^2 distributions of reconstructed tracks matched (unmatched) with a truth particle, in top pair production events with PU200.

3.4 TMTT TRACK FINDING PERFORMANCE

In this section, the final track finding performance for a TFP with the Seed Filter plus Linear Regression approach are presented. Results have been obtained at the end of the chain, after the duplicate removal stage. A comparison with the results obtained utilising the Kalman Filter is also provided.

3.4.1 Track Reconstruction Efficiency

The track reconstruction efficiency has been measured as in eq. 10 for both approaches and is shown in Figure 3.26, for charged tracks from the primary interaction in $t\bar{t}$ events with 200 PU. Similar results have been obtained in the two cases, with the SF+LR showing a slighter higher efficiency on average. From Figure 3.26b it is possible also to observe how the KF behaves better for high- p_T tracks, while the SF+LR presents better performance at low- p_T .

Figure 3.27 shows the distribution of the perfect track finding efficiency, where only reconstructed tracks with no incorrect stubs are selected as matched tracks. In the KF case, the perfect efficiency coincides with the standard efficiency definition, since it has been designed to keep only four stubs per tracks, as in the High-Purity SLR fitter. The SF+LR chain, instead, makes use of the Best-of-Both SLR and, therefore, does not have a predefined number of stubs per track. Hence, the perfect

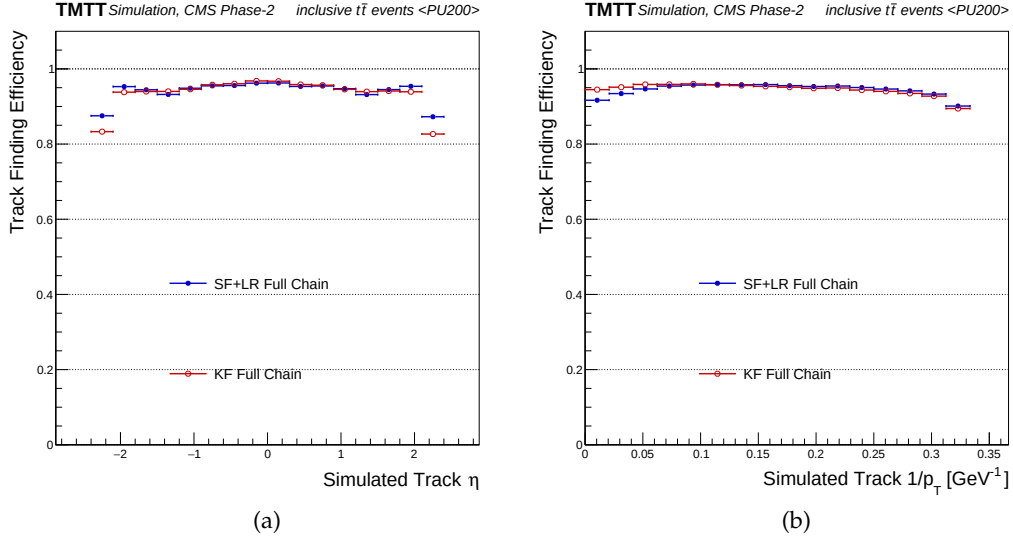


Figure 3.26: Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with PU200. Results obtained with Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).

efficiency, in this case, is slightly lower than the standard tracking reconstruction efficiency, nevertheless exceeding 93% over the entire acceptance.

PILEUP DEPENDENCY. Figure 3.28 shows the measured track reconstruction efficiency in $t\bar{t}$ events with different pileup content using the SF+LR fitter. A weak dependency on the number of pileup interactions can be observed, especially for low p_T particles where a drop in efficiency of about 4% is observed when passing from PU0 to PU200. This loss is mainly attributable to the SF. Low p_T tracks are more sensitive to multiple scattering effects, which modify their trajectory. It is possible for a stub belonging to a pileup track to be associated with a real track candidate by the HT. If this stub happens to be closer to the seeding line than the truth particle stub, it will be kept by the Seed Filter, resulting in a wrongly reconstructed track.

As shown in Figure 3.29 the number of reconstructed tracks per event increases with increasing pileup, as expected. On average around 80 tracks per event are reconstructed in $t\bar{t}$ events with 200 PU.

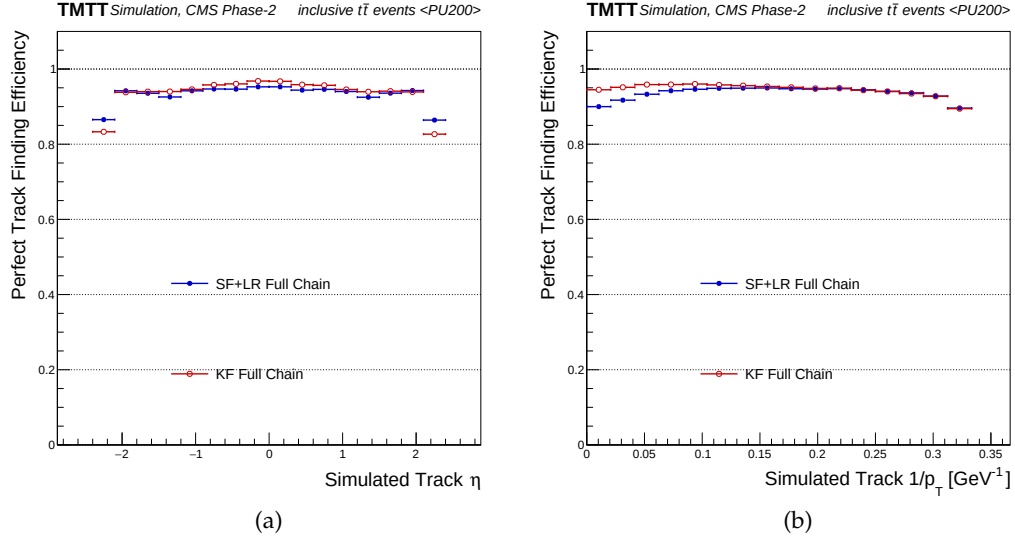


Figure 3.27: Track finding *perfect* efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with $\langle PU200 \rangle$. Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).

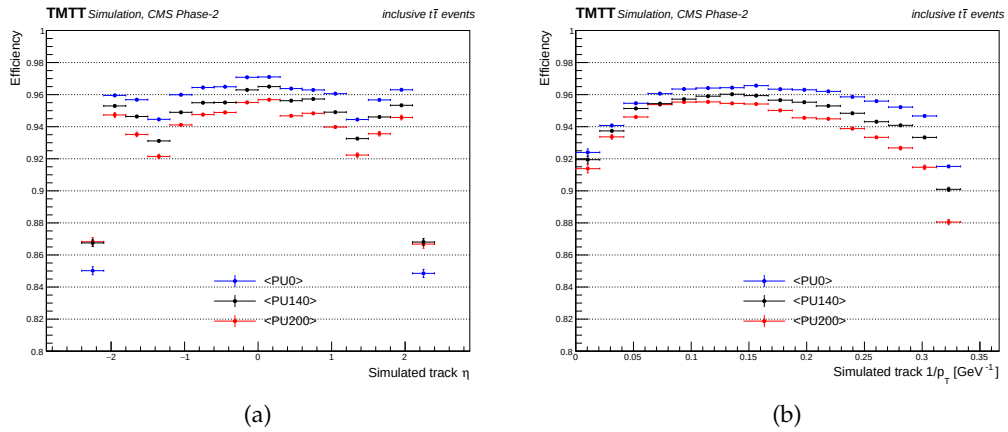


Figure 3.28: Track finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in top pair production events with different pileup contents (0, 140, 200). These results have been obtained using the SF+LR track fitter.

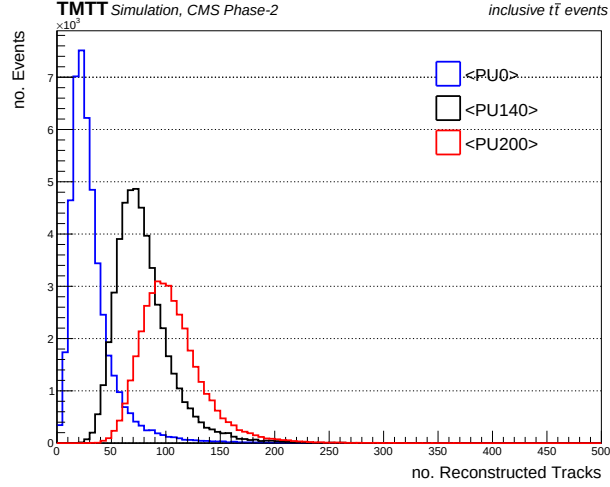


Figure 3.29: Distribution of the number of reconstructed tracks using a track finding algorithm with a [SF+LR](#) fitter, for top pair production events with different pileup contents (0, 140, 200).

LEPTON RECONSTRUCTION EFFICIENCY. Of crucial importance for the future [L1](#) track finding system is the capability to correctly reconstruct the high energy leptons in the collision. Figure [3.30](#) shows the reconstruction efficiency for muons and electrons originating from the primary interaction in $t\bar{t}$ events at pileup 200. The efficiency to reconstruct the muons in the event is about 98% over the entire pseudorapidity range. Worse performance is obtained in the electron case ($\sim 82\%$), mainly due to the bremsstrahlung effect, which results in deviations of the electron trajectories from the helix trajectory assumed.

Several possibilities have been investigated in order to retrieve the loss in efficiency for the electron case. One option exploits the possibility of using different bin sizes for the Hough Transform array, with larger bins at low p_T and smaller at high p_T . This helps the task of reconstructing the electrons in the event, but nevertheless, it increases the number of tracks and the number of fakes out of the track finding processor of about a factor 2. Therefore it has been decided to utilise a standard [HT](#) array, with a unique bin size. Further studies are currently on-going with the aim of increasing the electron reconstruction efficiency.

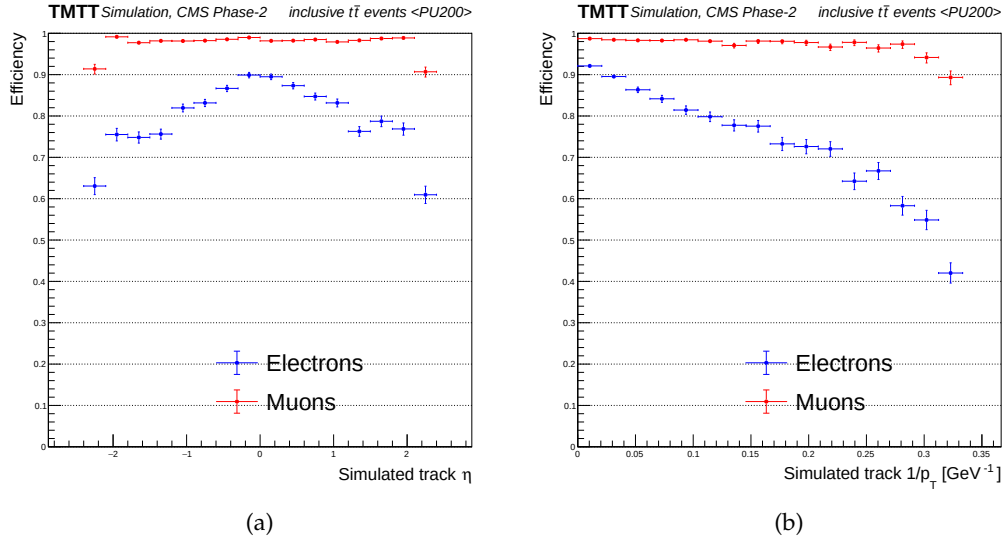


Figure 3.30: Track finding efficiency to reconstruct electrons (blue line) and muons (red lines) in top pair production events with pileup 200, as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b).

3.4.2 Track Parameter Resolution

Figures 3.31 and 3.32 show the resolutions of the four track parameters (p_T , Φ_0 , η and z_0) for reconstructed charged tracks from the primary interaction in $t\bar{t}$ events at pileup of 200, for both SF+LR and KF full chains, as a function of the truth particle inverse transverse momentum and pseudorapidity.

Tracks reconstructed by the SF+LR algorithm have been found to have a better parameter resolution in the r - z plane, than in the KF case. This can be explained by the fact that the KF has to cope with more stubs than the SLR and it sometimes selects the wrong state, because of the limited number of allowed iterations.

On the contrary, the KF shows better performance for the r - ϕ parameters, demonstrating that a linear approximation does not fit perfectly for low p_T particles, whose trajectory differs more from a straight line because of the high magnetic field.

Both approaches show an expected degradation in resolution with increasing pseudorapidity, due to a combination of the reduced effective precision for hits in the endcaps and the increased multiple scattering probabilities, because of the more material traversed by the particles. The r - z parameter resolutions are also

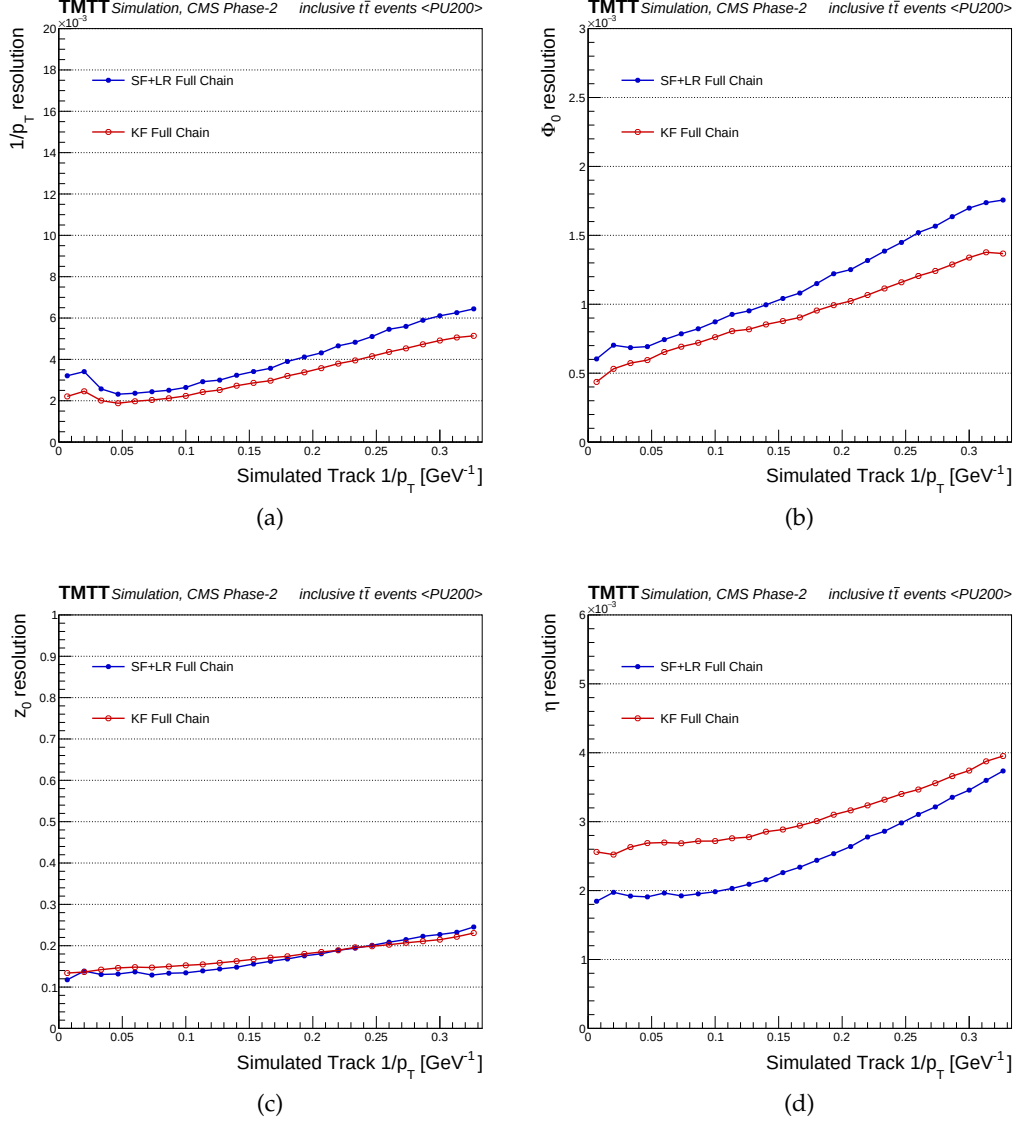


Figure 3.31: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).

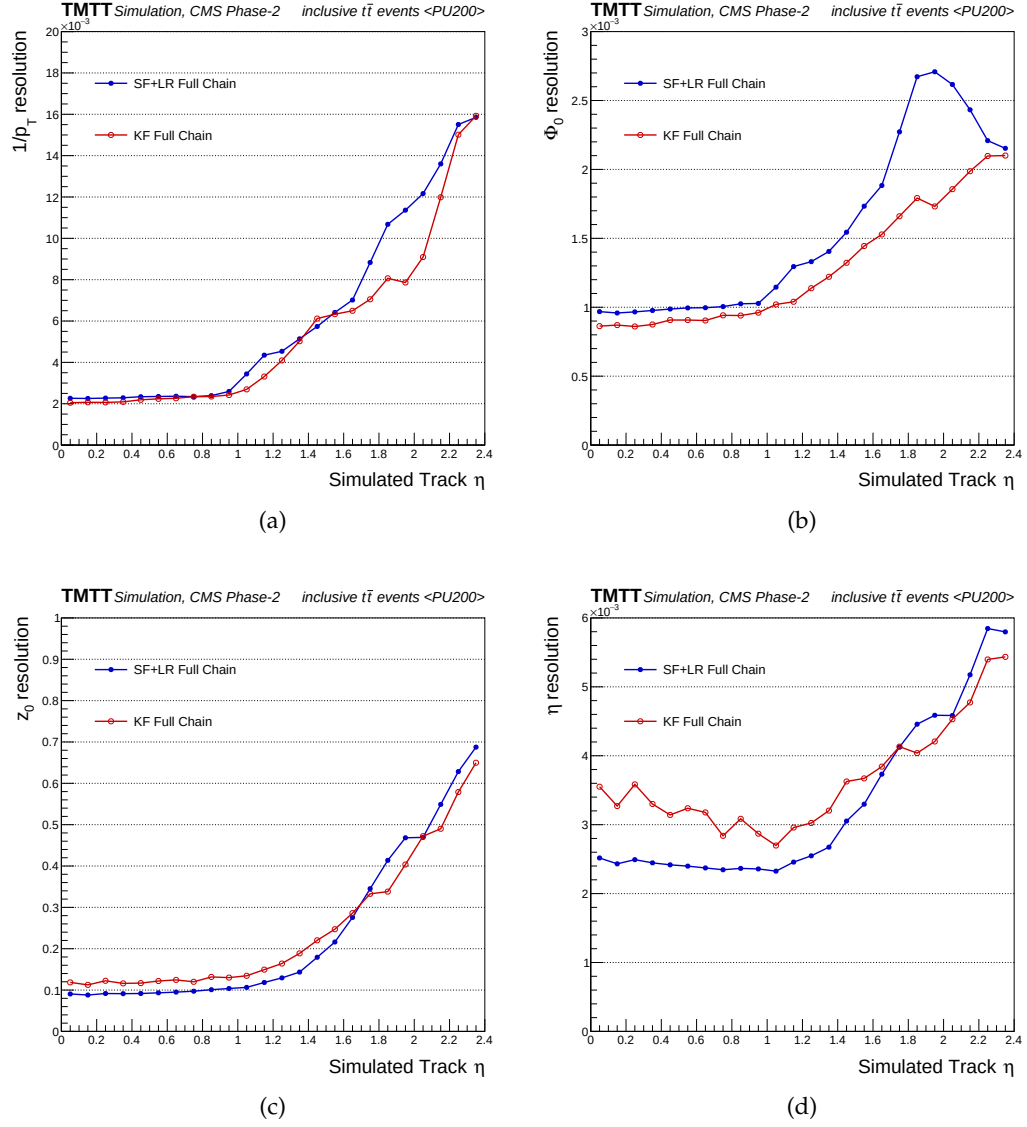


Figure 3.32: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with the Seed Filter plus Linear Regression fitter (blue line) are compared with Kalman Filter results (red line).

degraded by the fact that, at high rapidities, uncertainties on the stub position play a major role.

Table 3.2: Simulated track finding performance in top pair production events at a PU of 200, after each stage of the algorithm. The average number of reconstructed tracks per event in the entire tracker is also shown, together with the average number of fake and duplicate tracks that survived the different filtering processes. For benchmark purposes, also results obtained with a KF full chain are reported.

Stage	Efficiency [%]	no. of Tracks	no. of Fakes	no. of Duplicates
HT	97.16	337	141	129
SF	96.37	188	31	91
LR	95.63	188	36	86
DR	95.3	100	25	10
SF+LR Full Chain	94.72	83	10	9
KF Full Chain	94.72	96	17	18

3.4.3 Summary

Table 3.2 summarises the track finding performance obtained in $t\bar{t}$ events at a PU of 200, at each stage of the track finding algorithm. The results obtained are compared with the KF track finding alternative.

Both approaches show similar track finding performance in terms of efficiency. However, the SF+LR algorithm has been found to be capable of rejecting more fake tracks and of producing tracks with better track parameters in the r - z plane. This is important for the downstream trigger stages, especially for the primary vertex reconstruction task, as it will be shown in Chapter 5.

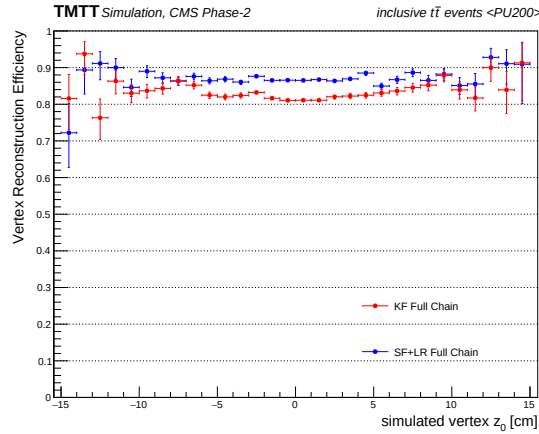


Figure 3.33: Efficiency to reconstruct the primary vertex of the hard interaction in top pair production events with 200 PU , using tracks produced with a SF+LR (blue) and a KF based track finding algorithm.

Figure 3.33 anticipates some results by showing the reconstruction efficiency for the primary interaction vertex in $t\bar{t}$ events with PU200, utilising tracks produced by the SF+LR and the KF algorithms. The higher fake rate out of the Kalman Filter affects the vertex reconstruction performance, resulting in an efficiency 4% lower than in the SF+LR case.

Figure 3.34 shows the fraction of fake tracks as a function of the reconstructed track pseudorapidity η , produced using the SF+LR and the KF algorithms in $t\bar{t}$ events with 200 PU. An average lower fake rate can be observed for the SF+LR track fitter, except in the high pseudorapidity region, where both algorithms make use only of stubs belonging to PS modules.

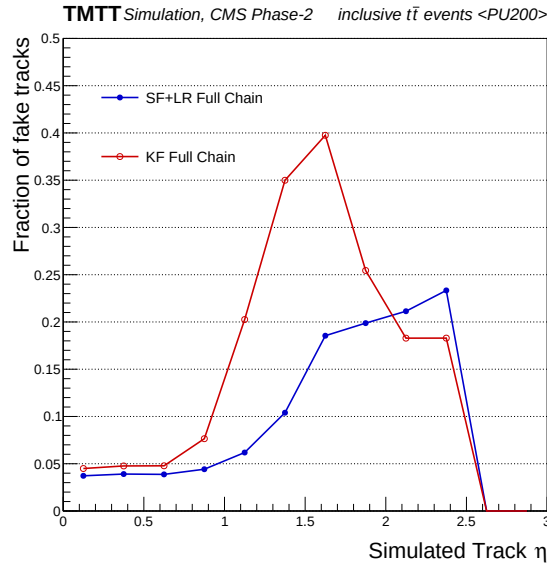


Figure 3.34: Fraction of fake tracks produced in top pair production events at a pileup of 200 by the SF+LR (blue) and the KF (red) track finding algorithms.

3.4.4 Finding tracks below 3 GeV

The downstream stages of the L1 trigger may require the availability of information about tracks below the 3 GeV threshold. To find tracks with $2 \text{ GeV} > p_T > 3 \text{ GeV}$, it is necessary to modify the Hough Transform array, adding 16 more bins along the q/p_T axis, in order to extend the range up to $\pm 1/2 \text{ GeV}^{-1}$.

In the low p_T region cells are merged in 2x2 macro-cells. This is required to prevent any loss of efficiency due to multiple-scattering effects dominant at low

transverse momenta. Figure 3.35a shows the distribution of the number of reconstructed tracks at the end of the chain with a $p_T^{\min} = 2$ GeV threshold, compared with the standard cut at 3 GeV. As expected the total number of tracks increases significantly, with the track finding algorithm producing on average about a factor 2.5 more tracks. The distribution of tracks per event is also much wider, having some events with more than 500 produced tracks. This number can constitute a problem for a final system implementation, which would require more resources or processing time, increasing subsequently the overall cost.

Figure 3.35b shows the track finding efficiency as a function of the inverse transverse momentum of the generated track. A clear drop in efficiency is visible at low p_T ($p_T < 2$ GeV), revealing that multiple-scattering effects in this range cannot be neglected. Indeed, despite the increase in the total number of output tracks, worse tracking performance are achieved, with a rising overall fake rate of about 20%.

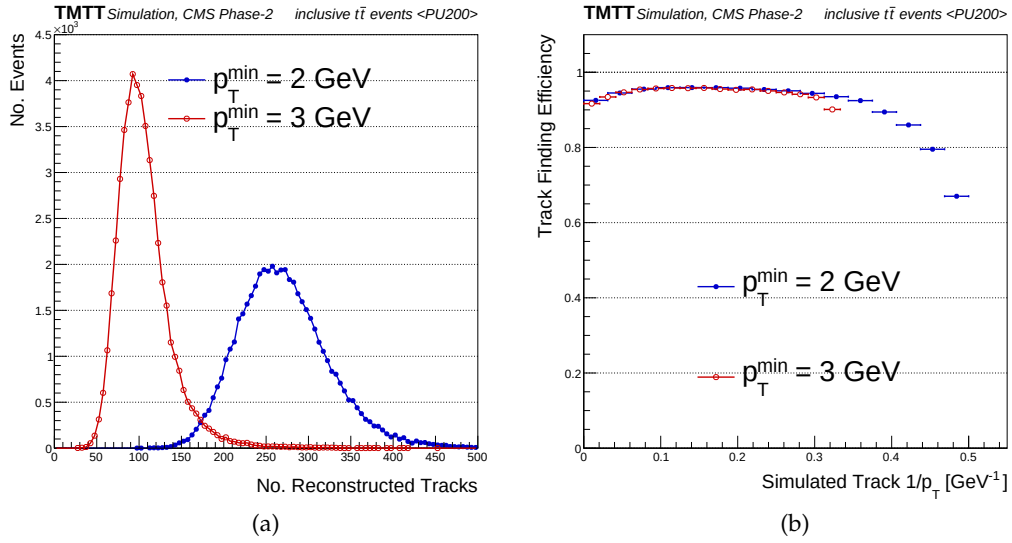


Figure 3.35: (a) Number of reconstructed tracks in $t\bar{t}$ events with 200 pileup. (b) Track finding efficiency as a function of the inverse transverse momentum $1/p_T$ of the simulated track. The blue line shows the obtained distributions using a modified version of the track finding algorithm, designed to reconstruct also tracks with $2 \text{ GeV} < p_T < 3 \text{ GeV}$. As a reference, also results obtained with the baseline p_T cut are shown (red line).

The track parameter resolutions are illustrated in Figure 3.36, as a function of the inverse transverse momentum, for the two $p_T^{\min}$ thresholds. As expected worse resolutions are obtained for low p_T tracks. In fact, low p_T particles multiple scatter more often, leading to a deviation of their trajectories that prevents particles

activating stubs in all the tracking layers. This reduction of stubs per track affects the track fitter performance, which as a consequence produces worse track parameters.

3.4.5 Preliminary Tilted Geometry results

Despite optimising the Track Finding algorithm for a flat barrel geometry, it is possible to use it also to reconstruct tracks in the new Phase 2 dataset samples, which adopt the novel Outer Tracker geometry with tilted PS modules in the barrel region.

The new design significantly reduces the number of produced stubs in each event. On average a reduction of a factor 1.5 has been observed, with the new tilted geometry producing an average of 10,000 stubs per BX compared to 15,000 with the old flat barrel design. This reduction translates in a reduced occupancy of the (η, ϕ) sub-sectors, which now contain an average of 59 stubs each (Figure 3.37a), and in fewer activated cells in the HT arrays (Figure 3.37b).

Despite the lower number of stubs available to perform the track reconstruction, no significant loss of efficiency has been observed. Figure 3.38 shows the track finding efficiency for $t\bar{t}$ events with PU200. On average the algorithm is able to reconstruct correctly 94.57% of the tracks belonging to charged particles with a $p_T > 3$ GeV.

Table 3.3 shows the track finding performance obtained with novel tilted geometry OT, compared with the results obtained with a flat barrel design. The new geometry further reduces the fake track rate at the end of the chain, bringing it down to about 9.7% on average.

Table 3.3: Simulated track Finding performance in top pair production events with 200 PU, using a novel outer tracker geometry with tilted modules in the barrel. Results obtained with a flat geometry are also presented as a benchmark.

Stage	Efficiency [%]	no. of Tracks	no. of Fakes	no. of Duplicates
Flat Barrel	94.7	84	10	9
Tilted Barrel	94.57	78	8	8

The resolution of the track parameters is presented in Figure 3.39. The obtained resolutions are comparable with what has been observed with the flat geometry.

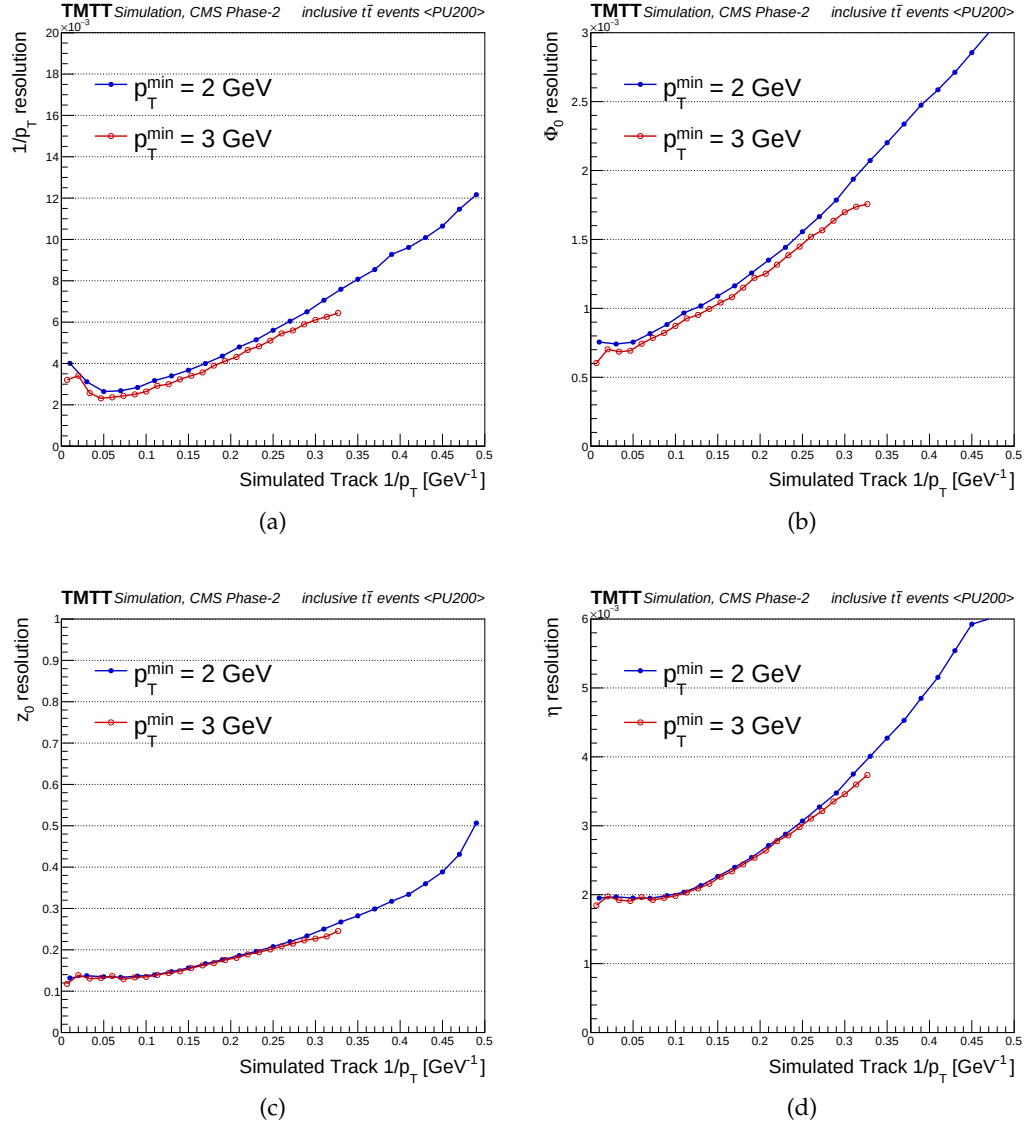


Figure 3.36: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track inverse transverse momentum $1/p_T$. The blue line shows the obtained distributions using a modified version of the track finding algorithm, designed to reconstruct also tracks with $2 \text{ GeV} < p_T < 3 \text{ GeV}$. As a reference, also results obtained with the baseline p_T cut are shown (red line).

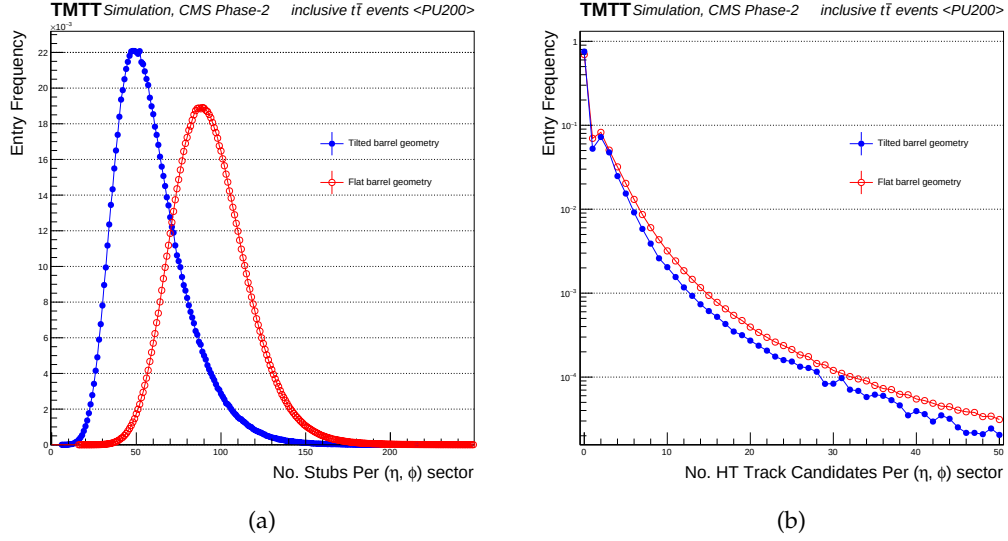


Figure 3.37: (a) Number of stubs per (η, ϕ) sub-sector and (b) number of track candidates per HT array, in $t\bar{t}$ events at a pileup of 200. The blue line represents the results obtained with novel tracker geometry with tilted module in the barrel region, which have been compared with the old flat barrel geometry (red line).

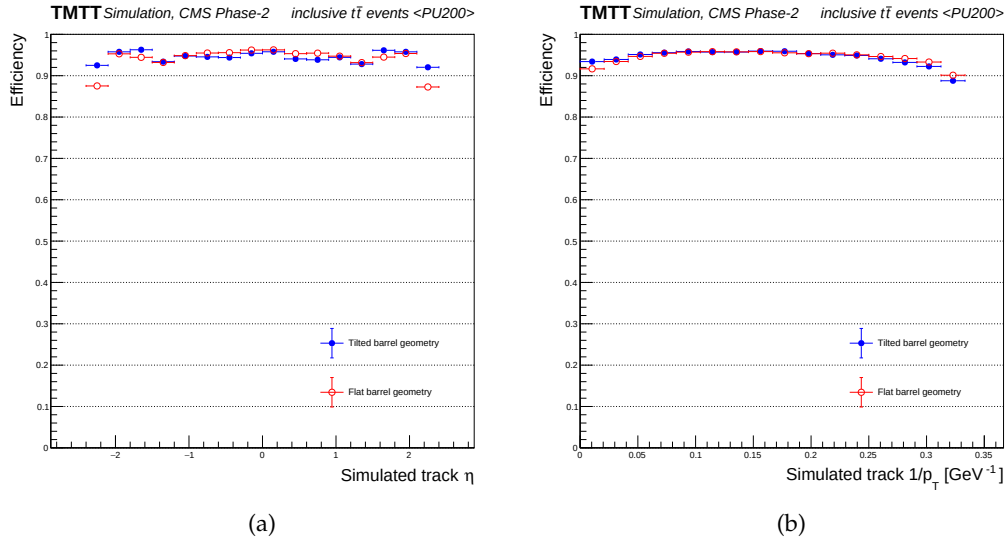


Figure 3.38: Track Finding efficiency as a function of the simulated track pseudorapidity η (a) and inverse transverse momentum $1/p_T$ (b) in $t\bar{t}$ events at a pileup of 200. The blue line represents the results obtained with novel tracker geometry with tilted module in the barrel region, which have been compared with the old flat barrel geometry (red line).

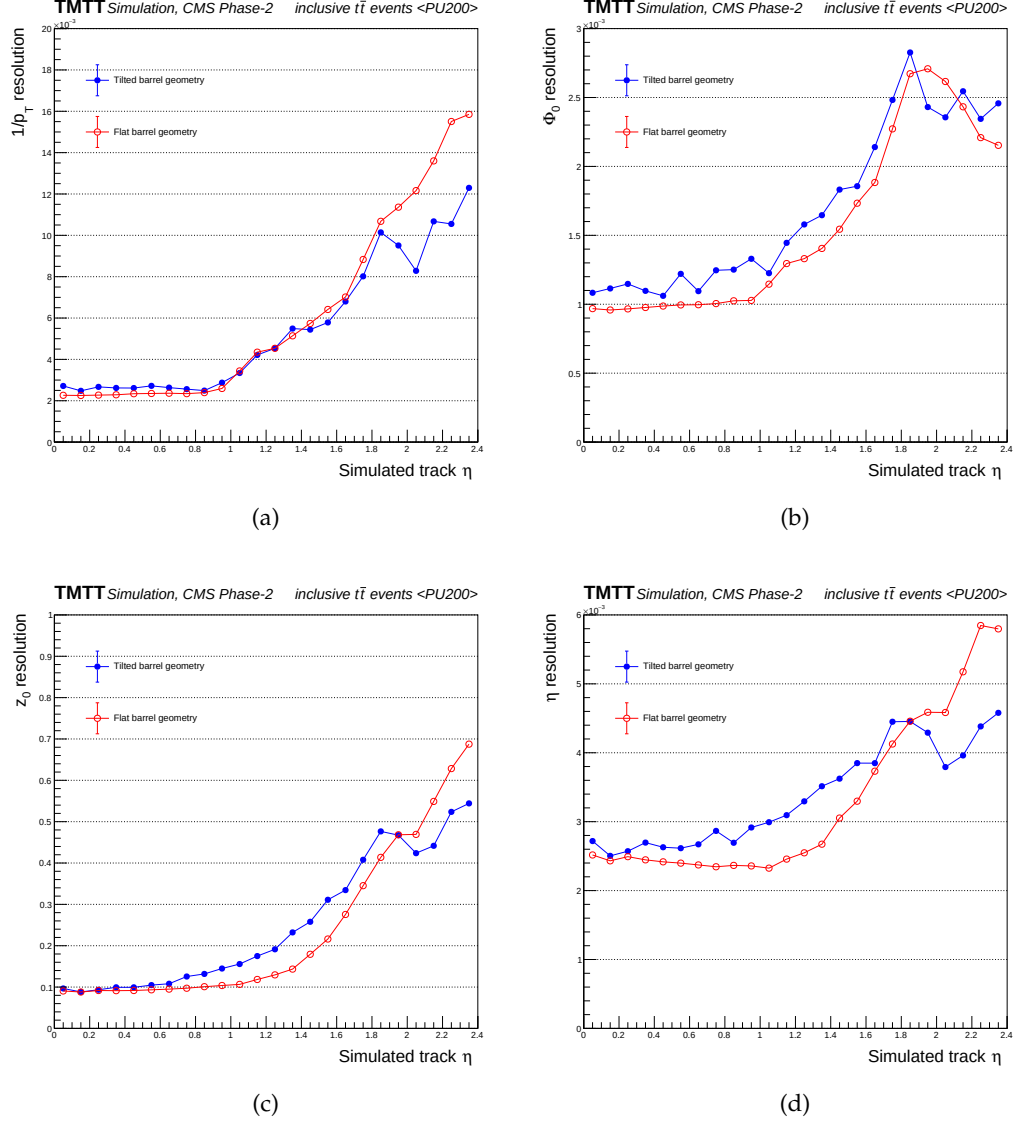


Figure 3.39: Track parameter resolutions for reconstructed tracks in top pair production events with pileup 200 as functions of the simulated track pseudorapidity η . Results obtained with a tilted barrel geometry (blue line) are compared with that from a flat barrel geometry (red line).

In conclusion, these preliminary results have been found to be encouraging, showing that a track finding algorithm based on the Hough Transform technique can successfully reconstruct tracks with a tilted barrel design. Further optimisations to the algorithm are foreseen in order to exploit fully the novel geometry and to significantly improve the overall track finding performance.

HARDWARE DEMONSTRATION OF THE TMTT L₁ TRACK FINDER PROCESSOR

To fully understand the feasibility and the potential of the proposed L₁ Track Finding system it is essential to demonstrate its functionalities in hardware and to measure its performance within the required latency budget.

A hardware system has been built to prove the performance of a track finder slice, corresponding to a single Track Finder Processor (TFP) board. Since each TFP will be identical to the others, by injecting into the demonstrator system data from different tracker ϕ octants, it is possible to reconstruct entire events in hardware.

The main demonstrator system is located at the CERN Tracker Integration Facility (TIF) and is equipped with FPGA-based MP7 boards [97]. Two auxiliary systems have been built at Imperial College London and at the Rutherford Appleton Laboratory for firmware development and debugging purposes.

The following section will provide a detailed description of the implemented demonstrator setup. Successively the firmware implementation of the track finding algorithm will be covered, showing how each step in the track finding chain has been realised on FPGAs. Here particular attention will be given to the Seed Filter plus Linear Regression (SF+LR) firmware, which will be described in detail. Eventually, the performance obtained with the demonstrator system will be shown, together with a comparison with the results expected from software simulation.

4.1 THE DEMONSTRATOR SETUP

The demonstrator system has been installed in a custom dual-star microTCA [104] shelf, equipped with a commercial NAT microTCA Carrier Hub (MCH) which provides Ethernet communication via the backplane. The crate also contains an AMC13 [98] card, designed by CMS for synchronisation, timing and control. The track finding processor functions are implemented in five Imperial Master Processor, Virtex-7, Extended Edition (MP7-XE) double width AMC cards [97].

4.1.1 *The Master Processor MP7 board*

The Imperial Master Processor MP7 board has been designed for the upgrade of the CMS [L1](#) Calorimeter trigger [\[89\]](#). It is equipped with a powerful Xilinx Virtex-7 XC7VX690T FPGA [\[84\]](#) and 12 Avago Technologies MiniPOD optical transceivers [\[105\]](#), each of them providing 12 optical link pairs capable of 12.5 Gbps for a total I/O bandwidth of 0.9 Tbps. This high optical bandwidth makes the MP7 an excellent stream-processor engine.

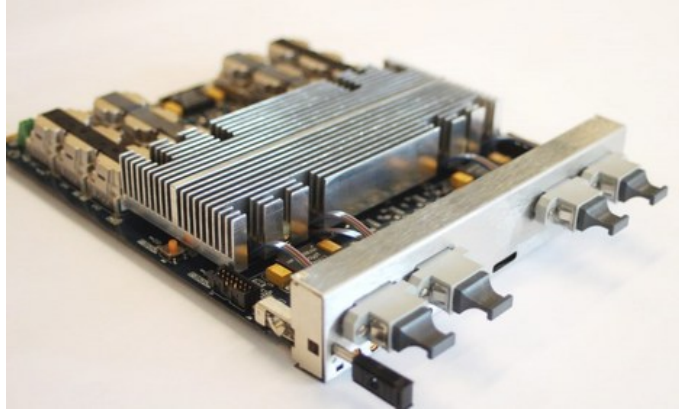


Figure 4.1: The Imperial Master Processor MP7 board [\[97\]](#).

Links are configured to run at 10 Gbps in the demonstrator system with 8b10b encoding for an effective 8 Gbps transfer rate. Therefore the bandwidth is half of that assumed for the final system, which will have a 16.3 Gbps transfer rate with 64b66b encoding. This results in an increased time multiplexed factor of 36 instead of 18.

Since the MP7 has been designed to fulfil different tasks, it provides a sufficiently generic core firmware, which manages the transceivers, the serialisation/de-serialisation, the data buffering, the I/O formatting, the clock configuration and the communication with the Ethernet interface. In this way the end-user is able to implement the desired algorithm, that is completely agnostic with respect to the physical hardware. Figure [4.2](#) shows the model of abstraction used in the MP7 core firmware.

This division between core and algorithm firmware allows the separation of tasks of the track finding processor in different blocks, each implemented into a single MP7 board, daisy-chained together with high-speed optics. This facilitated the development of the demonstrator system, assigning firmware responsibilities

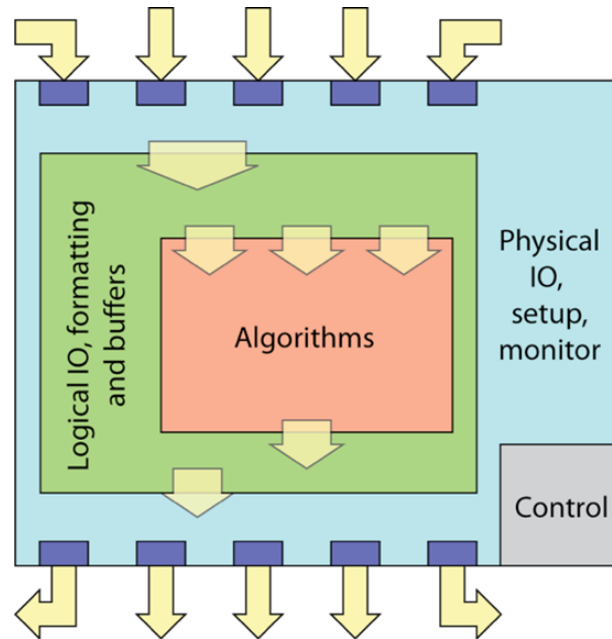


Figure 4.2: The structure of the MP7 firmware. The algorithm block is abstracted from the physical hardware and service tasks. [97]

between the available personnel. In addition, daisy-chaining the algorithms across multiple boards permits the estimation of the final system performance utilising currently available technology.

Ethernet communication with the board is achieved via IPBus [106], which also permits the of data into the data buffers of the MP7 for algorithm testing.

4.1.2 Data Formats, Mappings and Link Specifications

A picture of the CERN demonstrator system is shown in Figure 4.3. A total of eleven MP7 boards have been installed in the microTCA crate. Eight boards are assigned to the demonstrator chain. Two boards (*source*) replicate the task of the DTCs, transmitting data from a set of 36 DTCs boards into the Track Finding Processor. The source board firmware consists simply of a large buffer which contains all stub data belonging to a particular detector octant. Each output link from the source boards represents a separate DTC output. Two source boards are needed in order to correctly emulate the boundary effects between two adjacent detector octants.

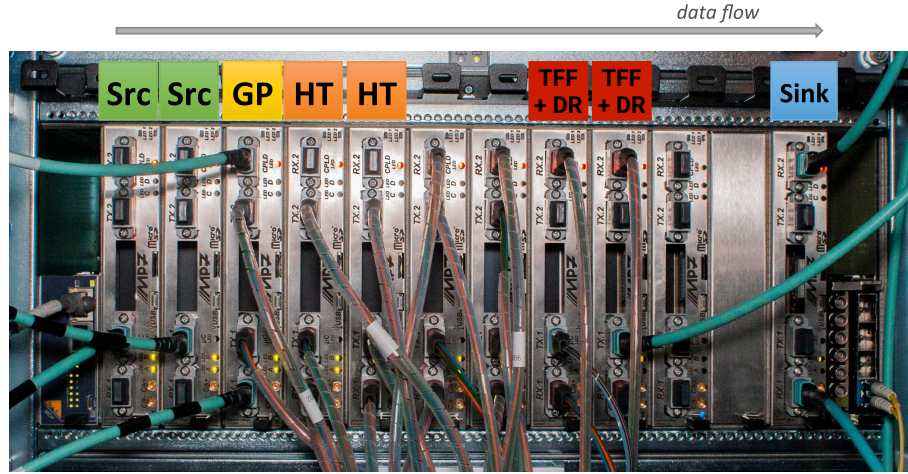


Figure 4.3: The **TMTT** microTCA hardware demonstrator crate installed at CERN TIF. The crate is equipped with 11 MP7-XE boards, an AMC13, MCH and required optics. [99]

Stubs are injected into the **TFP** in a 48-bit data format (Table 4.1). The chosen time multiplexing period (36 **BX**) fixes the maximum amount of payload data that can be injected into a single **TFP** before a new event arrives. Each input link therefore carries a data packet of 36 **BX** in length (216 clock cycles at 240 MHz) and includes all payload (e.g. stub) data plus additional link headers/trailers. The payload data is 210 frames in length and propagate through the several logic blocks composing the Track Finder Processor.

Table 4.1: Stub data-format at the input of the Track Finder Processors. Stubs are formatted and time multiplexed by the DTC boards, which encode them in a 48-bit data format.

Parameter	Unit	Resolution	Range		Bits
ϕ_{O}	rad	0.0000479	-0.78539	0.78539	15
r_{T}	cm	0.1006	-51.52	51.52	10
z	cm	0.15625	-320.0	320.0	12
bend	strip	0.25	-8	7.75	6
mod. type	-	1	0	7	3
stub valid	-	-	-	-	1
EoE	-	-	-	-	1
Tot.	48b				

The **TFP** consists of five MP7 boards, each of them implementing a particular stage of the track finding chain. The output track data produced is eventually

captured by another MP7 board placed at the end of the chain (*sink*). The remaining three boards are used for standalone firmware testing or parallel data taking. Figure 4.4 shows how the different components in the chain are connected.

4.1.3 FPGA resources

The following sections of this chapter will show the different firmware designs that have been developed to implement the track finding algorithm on FPGA. For each logic block it will be shown also the actual resource utilisation in the Xilinx Virtex-7 XC7VX690T FPGA. The main elements that have been considered are:

- Lookup tables (LUTs), which constitute one of the main logic block of an FPGA. A LUT consists of a some number of inputs and outputs. A LUT contains usually a block of RAM indexed by the LUT's input. The output corresponds to the indexed value in the RAM. LUTs in a Virtex-7 FPGA have six independent inputs and two independent outputs;
- Flip-Flops (FFs), simple circuits that can store one bit of data and have two stable states (1 or 0);
- Block RAM (BRAM), blocks of RAM memory that can store up to 36 Kb of data. They can also operate in Simple-Dual Port (SDP) mode, having two independent inputs and two independent outputs with different clocks;
- Digital Signal Processings (DSPs), dedicated logic components that can be used for fast signal processing, as wide dynamic bus shifters or as arithmetic units.

4.2 THE DEMONSTRATOR SOFTWARE FRAMEWORK

In addition to the hardware system, a demonstrator software framework has also been developed in order to convert simulated data into the required MP7 format and to compare the hardware results with the emulation. The demonstrator software framework consists then of three programmes, all based on CMSSW: the Pattern Writer, the Unpacker and the Comparison Software. Figure 4.5 shows how the demonstrator data taking process is achieved, illustrating the various communication between software and hardware.

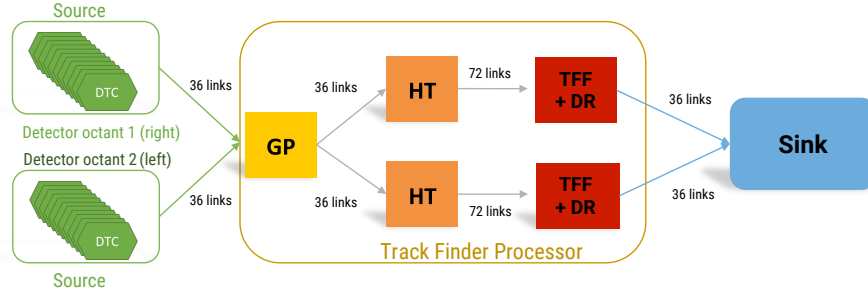


Figure 4.4: Block diagram of the demonstrator slice. Each block in the diagram represents a separate MP7 board, running a separate step in the track finding algorithm. The whole system is made of eight MP7s: two Source, a Geometric Processor (GP), two Hough Transform (HT), two Track Filter and Fitter (TFF) plus Duplicate Removal (DR) and a Sink. [99]

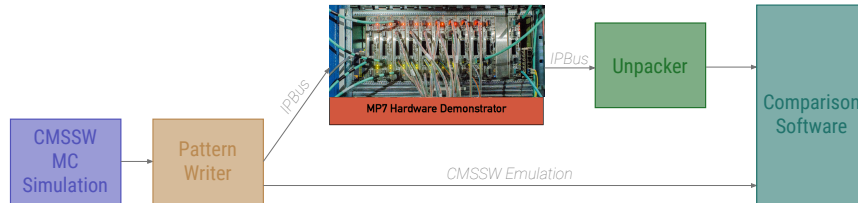


Figure 4.5: The demonstrator data taking procedure. Stub and track data produced with a simulated Phase-2 CMS tracker geometry (CMSSW MC Simulation) are digitised and converted into a specific MP7 txt format by the Pattern Writer (PW), which runs also a digitised version of the track finding algorithm (CMSSW Emulation). Txt files are then injected via IPBus [106] into the MP7 Hardware Demonstrator, running the firmware version of the track finding algorithm. Hardware results are then captured via IPBus and reconverted into a CMSSW format by the Unpacker. Finally, hardware and emulation results are compared by the Comparison Software.

4.2.1 *Pattern Writer*

The first task in the chain is to convert stub simulated data into the required MP7 data format. The *PW* software is responsible for this. The *PW* takes as input simulated physics events produced with *CMSSW* and converts them to a text file format which will be transmitted via IPBus to the MP7. To allow testing of each firmware block in a standalone way, the *PW* also runs a modified version of the simulation software. This emulation processes the same integer formatted stubs used as input into the hardware demonstrator, implementing fixed-point mathematical operations in order to simulate as closely as possible the logic implemented in the FPGAs. In addition to the full chain output, the *PW* is able to access the results at output of each steps of the track finding algorithm. For example, for a standalone testing of the *HT* firmware, the *PW* will run first the *GP* algorithm, producing an input text file containing the sorted stub information out of the *GP*, and then it will run also the *HT* step, in order to produce the simulated *HT* track candidates that will be compared with the hardware results.

4.2.2 *Unpacker*

The Unpacker software is responsible for converting data captured from the MP7s into a useful *CMSSW* EDM format. It takes as input the captured text files, which are decoded utilising the relative data format. To allow data taking at any stage of the Track Finding chain, the Unpacker implements the data format at the output of each step. In addition, it is also able to measure the latency for each block by comparing the input and the output text files.

4.2.3 *Comparison Software*

The final task in the demonstrator chain is performed by the Comparison Software. It takes as input the emulated software data, produced by the *PW*, and the unpacked hardware results from the Unpacker. Its role is to make a one-to-one comparison, between the tracks produced by the emulator and by the hardware demonstrator. In addition, it measures important track finding parameters, such

as the efficiency and the track fit parameter resolution, comparing the obtained tracks, from both emulation and hardware, with the generated true tracks.

4.3 TRACK FINDING PROCESSOR HARDWARE IMPLEMENTATIONS

Each Track Finding Processor board will be responsible for finding tracks in a particular processing octant. The Track Finding algorithm has been implemented in firmware in five different MP7 boards, each of them running a separate block of the algorithm. Figure 4.6 shows how the different logic blocks are connected in a single Track Finding Processor board.

The following sections will describe each logic block, showing the hardware demonstrator results at any step of the chain compared with the emulation.

4.3.1 Geometric Processor

Table 4.2: Stub data-format at the input of the HT boards. Stubs are converted by the GP into an useful 64-bit data-format, which ease the load of the downstream processing stages.

Parameter	Unit	Resolution	Range		Bits
ϕ_S	rad	0.0000479	-0.3926	0.3926	14
r_T	cm	0.1006	-51.52	51.52	10
z	cm	0.15625	-320.0	320.0	12
layer iD	-	1	0	7	3
sector iD	-	1	0	17	5
$m_{\min}$	1/cm	0.0000119	-0.00190	0.00190	5
$m_{\max}$	1/cm	0.0000119	-0.00190	0.00190	5
sub- η_L	-	-	-	-	1
sub- η_R	-	-	-	-	1
stub valid	-	-	-	-	1
EoE	-	-	-	-	1
Reserved	-	-	-	-	6
Tot.	64b				

The first task that the Track Finding Processor board has to face is to assign stubs coming from the DTC to their relative angular sub-sectors, as defined in section 3.2.1. The firmware block responsible for this is the Geometric Processor (GP). The GP takes as input the 48-bit DTC stubs and unpacks them into a 64-bit extended

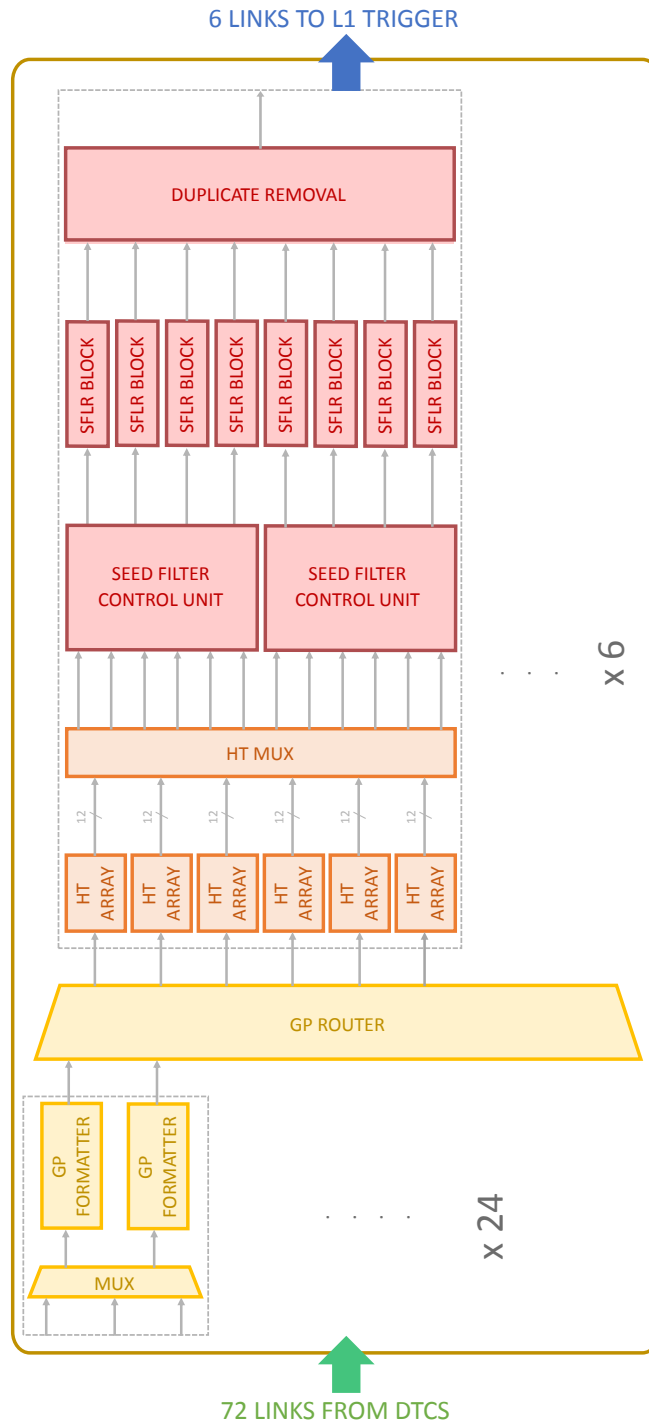


Figure 4.6: Overview of the entire TFP firmware, showing the main logic components and connections. Different colours highlights the various parts of the chain: GP in yellow, HT in orange and TFF+DR in red. [99]

format, which reduces the load on the downstream stages, and assigns them to the correct sub-sector. The GP output data format is shown in Table 4.2. In total the GP routes stubs from 72 inputs (one per DTC) to 36 outputs (one per sector).

The GP algorithm has been implemented as a pre-processing block, which calculates the correct sub-sector for each stub based on its global position, followed by a routing block. The routing stages are sub-divided into three layers. Stubs are firstly routed via a rough η sorting. This is followed by a finer η sorting and finally by a ϕ sector routing (Figure 4.7). Each sorting block in the GP firmware is highly configurable so that alternative sector definitions can be easily be adopted. The stubs associated with each sub-sector are routed to dedicated outputs, such that data from each sub-sector can be processed by an independent HT array.

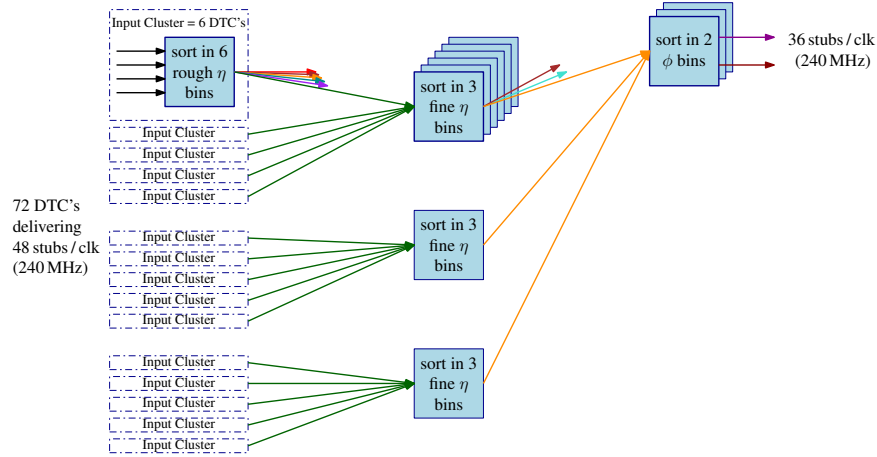


Figure 4.7: The Geometric Processor (GP) routing network. Stubs coming from the DTCs are assigned to 36 finer sub-sectors in (η, ϕ) . The sorting happens through a three-layers router. Stubs are initially grouped in six rough bins in η , then sorted in three finer η bins and finally assigned to the correspondent sector according to their ϕ .

Table 4.3 shows the required resource utilisation for the GP firmware running at 240 MHz as implemented within a Xilinx Virtex-7 XC7VX690T FPGA. The firmware fits well in a single MP7 board and has been found to have a total processing latency, defined as the time difference between the first stub received and the first stub transmitted, of 251 ns. The routing block has been also tested at a higher frequency of 480 MHz, leading to a latency reduction of 60 ns.

The output packet data from the GP presents few gaps in the data stream, resulting in a limit on the maximum number of stubs that can be transmitted to the HT per sub-sector equal to 175.

Table 4.3: Resource utilisation of each pre-processing block (with 48 needed per TFP) and the entire routing block of the GP as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis.

	LUTs	DSPs	FFs	BRAM (36 Kb)
Pre-processing Block	1942 (0.4%)	22 (0.6%)	2416 (0.3%)	1 (0.0%)
Routing Block	27700 (6.4%)	0 (0.0%)	89531 (10.3%)	174 (11.8%)
Tot. GP	29642 (6.8%)	22 (0.6%)	91947 (10.6%)	175 (11.8%)

Figure 4.8 shows the average number of stubs assigned to each (η, ϕ) sector by the GP firmware compared with the results obtained using the emulation. The matching between hardware and software is very good, with about 99.9% of stubs assigned to the same sub-sector and having the same coordinates in both cases.

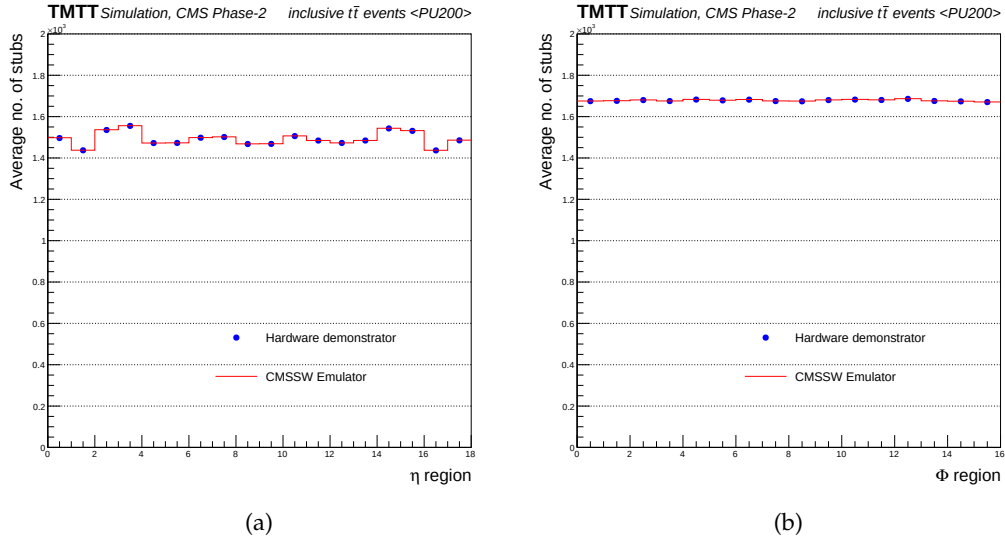


Figure 4.8: Average number of stubs per η (a) and ϕ sectors. Results obtained with the hardware demonstrator (blue dots) are compared with expectations from the software emulation (red lines). An excellent matching has been observed, greater than 99.9%

4.3.2 Hough Transform

Each pair of output links (*channel*) from the GP transports stub data relative to a particular (η, ϕ) sub-sector, for a total of 36. In order to increase the parallelisation of the process and to reduce the latency, a HT array is assigned to each channel. Hence, each TFP employs 36 independent HT arrays implemented in two Virtex 7 FPGAs.

The HT array firmware consists of a fully pipelined design, processing stubs at 240 MHz. Figure 4.9 displays the firmware design of an individual HT array block. Each array consists of 32 ‘Columns’ blocks, corresponding to the q/p_T columns in the HT array, controlled by several ‘Book Keeper’ modules. The current implementation implements 12 Book Keepers, each communicating with two or three daisy-chained Columns.

Each Book Keeper receives as input one stub per clock cycle, which is then immediately stored in 36 Kb Block RAM. Stubs are then propagated to the first Column in the Book Keeper. On each clock, cycle stubs are then transmitted to the following Column in the daisy chain.

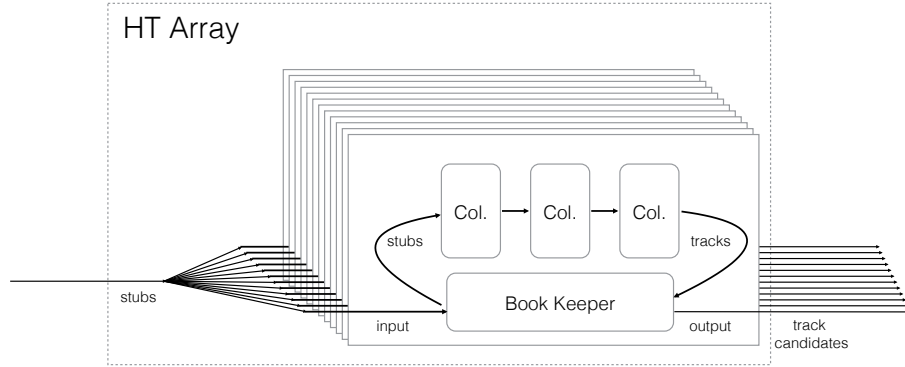


Figure 4.9: Overview of the firmware implementation of one HT array, assigned to an individual sub-sector. Stubs are propagated to 12 HT blocks, each containing a Book Keeper logic connected to a daisy chain of 2 or 3 Columns (Col.). [99]

In Figure 4.10 a logic diagram of the components in a Column block is represented. Each Column calculates for each stub a value of Φ_T at its right-hand boundary, as defined in eq. 29. In order to optimise FPGA resource consumption,

constants appearing in the equation, i.e. the B field, are absorbed into the definition of q/p_T , such that the implemented equation becomes,

$$\Phi_T(n) = \Phi_T(0) + n \cdot \Delta_{q/p_T} \cdot r_T, \quad (52)$$

where n is the Column index in the range $[-16, 15]$ and Δ_{q/p_T} is the fixed width of a q/p_T column. The value of $\Phi_T(0)$ corresponds simply to the ϕ_S coordinate of the stub in the sector. Furthermore, a stub is allowed to enter a particular Column only if it satisfies eq. 22. The allowed q/p_T range is pre-calculated for each stub by the GP, such that the implemented relation is simply,

$$m_{\min} < n < m_{\max}. \quad (53)$$

By construction, stubs can activate one or two Φ_T cells in the same column. To identify stubs compatible with two cells the firmware calculates in each column both values of Φ_T at the left and right edges. If a stub is found to be consistent with two cells, it is duplicated and stored in a “ Φ_T buffer”, whose content will be processed at the first gap in the data stream. Stubs are then propagated to the “Track Builder” block, which implements the 64 Φ_T cells and assigns each stub to the appropriate cell. A maximum of 16 stubs per cell is imposed here, ignoring any additional stub entering the cell.

For each Φ_T cell the Track Builder keeps two records of the tracker stations activated by the stubs stored in the cell, one for each pseudorapidity half of the sub-sector under consideration. Where a stub is compatible with both sub-sectors, it is kept for both sides. If the threshold criteria for the number of activated tracker stations is satisfied in at least one of the two halves, then a track candidate has been found and the cell is marked for read-out.

Stubs consistent with a track candidate are then shifted to the next Column by the “Hand Shake” component, until there are no more stubs in the pipeline. Each track candidate stub carries now information about the track parameters, q/p_T and Φ_T , in terms of the Hough Transform array coordinates (m, c) . The full output stub data format of the HT stage is shown in Table 4.4.

Finally, output track candidates from six Book Keepers are grouped together by means of a multiplexer stage, in order to minimise the number of output channels

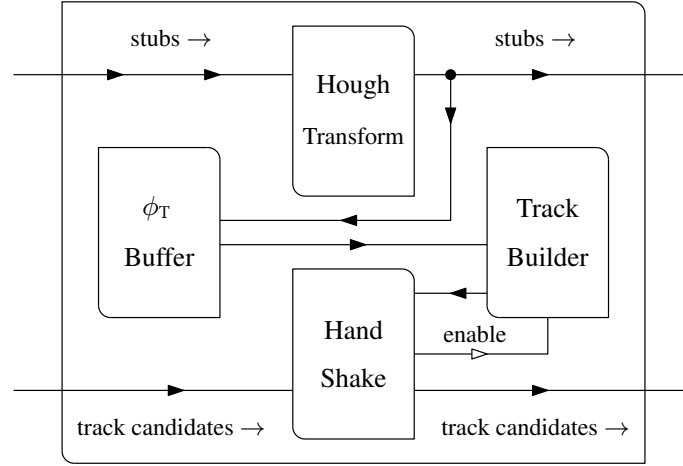


Figure 4.10: Logic components within a HT Column, corresponding to a single q/p_T column in the array. Input stubs (*stubs* in the picture) and stubs consistent with a track candidate (*track candidates* in the figure) are propagated from Column to Column. [99]

Table 4.4: Stub data-format at the input of the TFF+DR boards. Stubs are still encoded in a 64-bit words, while some information has been discarded (i.e. bend), not being of any use for the final filtering tasks. The HT transmits only stubs consistent with a track candidate, appending to each stub the coordinate of the cell in the array.

Parameter	Unit	Resolution	Range		Bits
ϕ_S	rad	0.0000479	−0.3926	0.3926	14
r_T	cm	0.1006	−51.52	51.52	10
z	cm	0.15625	−320.0	320.0	12
layer iD	-	1	0	7	3
sector iD	-	1	0	17	5
m_{bin}	1/cm	0.0000119	−0.00190	0.00190	5
c_{bin}	rad	0.00614	−0.196	0.196	6
stub valid	-	-	-	-	1
Reserved	-	-	-	-	13
Tot.	64b				

and to balance the output load across sub-sectors. In this way, if an excessive number of tracks is found within a single HT array, corresponding typically to a very high energetic jet, track candidates are assigned to different output channels, hence avoiding unwanted truncation effects. Since firmware has been designed to receive and transmit a packet of data of a fixed length (210 clocks), all tracks produced after that limit has been reached will be just discarded.

In total 72 output channels have been implemented from the HT stage to the following Track Filtering and Fitting step. Table 4.5 shows the resource utilisation of one Column and of one HT array, as implemented in a Virtex 7 FPGA.

Table 4.5: Resource utilisation of one Column (in the HT array) and of one entire HT array, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources are shown in parenthesis. The entire TFP needs 36 HT arrays. The resources needed to implement the multiplexer are not included here, but are relatively small in comparison.

	LUTs	DSPs	FFs	BRAM (36 Kb)
One Column	188 (0.0%)	2 (0.1%)	204 (0.0%)	1 (0.1%)
One HT array	6014 (1.4%)	64 (1.8%)	6718 (0.8%)	33 (2.2%)

Figure 4.11 shows the number of track candidates found by the HT firmware as a function of the HT array parameters q/p_T and Φ_T , compared with results obtained by the emulation, in $t\bar{t}$ events with 200 pileup. Very good matching has been found between the two, with about 99.4% of the track candidates produced in software having the same HT parameters of the track candidates reconstructed in hardware. In 97.7% of cases the matched track candidates contain also exactly the same stubs, obtaining a tracking efficiency above 97% in both cases.

The HT firmware has a fixed first in - first out latency equal to 1025 ns. Similar to the GP, the HT firmware produces gaps in the output data stream, limiting the number of stubs that can be transmitted to the downstream track fitter stage to 202 per output channel. Track candidates are here ordered by the absolute value of the q/p_T bins, from the smallest to the largest, prioritising the high- p_T tracks.

4.4 TRACK FITTING WITH SEED FILTER PLUS LINEAR REGRESSION

Both algorithms, developed to execute the track filtering and the fitting stage after the Hough Transform, have been also implemented in firmware, targeting a Virtex-7 FPGA. The Kalman Filter firmware has been designed using the high-

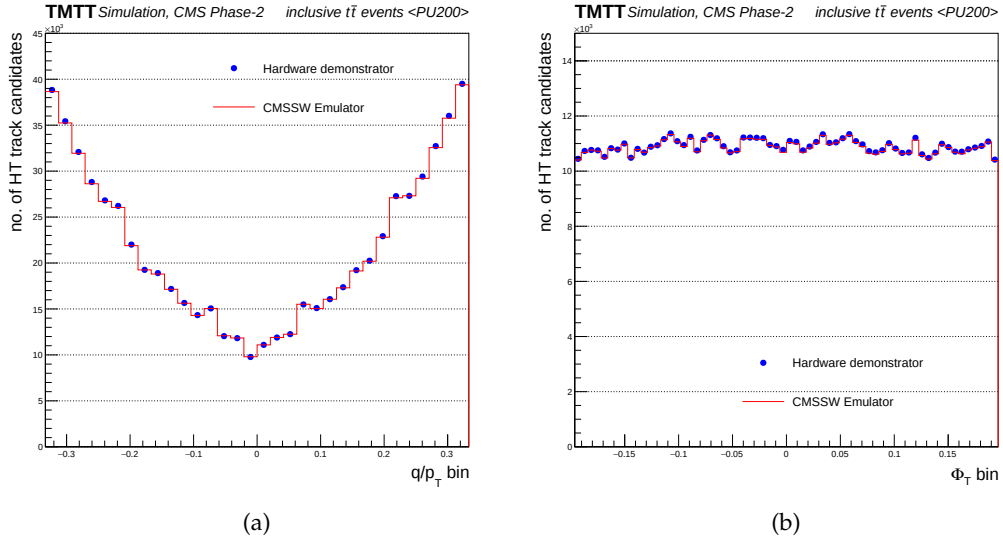


Figure 4.11: Distribution of track candidates as a function of the HT track parameters q/p_T (a) and Φ_T (b). Results obtained with the hardware demonstrator (blue dots) are compared with expectations from the software emulation (red lines). An excellent matching has been observed, greater than 99.4%.

level synthesis language MaxJ [107], while the Seed Filter plus Linear Regression has been completely programmed in VHDL [108]. The following sections will describe in detail the SF+LR firmware implementation, whose performance will be compared with the KF approach. A full description of the KF implementation can be found in [109].

The Duplicate Removal firmware has been developed independently from both fitters, in such a way that both implementations integrate the same DR block.

4.4.1 Design overview

An overview of the overall design of the Seed Filter plus Linear Regression and Duplicate Removal firmware [110] is shown in Figure 4.12. Hough Transform track candidate stubs enter the design across 36 input channels. Each input channel is connected to a Track Locator component, which stores and transmits the stubs to the next firmware block. Read-out from the Track Locators is managed by the Control Unit blocks, which have as role transmitting stubs belonging to individual track candidates to the downstream Seed Filter plus Linear Regression (SF+LR) modules for processing.

Table 4.6: Track data-format at the output of the [TFP](#). Track information is encoded in a 96-bit word, containing the four track fit (Φ_T , q/p_T , z_T and $\tan \lambda$) and quality (χ^2) parameters.

Parameter	Unit	Resolution	Range		Bits
Φ_T	rad	0.0000479	-0.3926	0.3926	14
q/p_T	1/cm	0.0000037	-0.0038	0.0038	11
z_T	cm	0.0195	-160.0	160.0	14
$\tan \lambda$	rad	0.0000037	-24.85	24.85	17
χ^2	-	0.25	0	127	9
sector iD	-	1	0	17	5
m_{bin}^{HT}	1/cm	0.0000119	-0.00190	0.00190	5
c_{bin}^{HT}	rad	0.00614	-0.196	0.196	6
m_{bin}^{fit}	1/cm	0.0000119	-0.00190	0.00190	5
c_{bin}^{fit}	rad	0.00614	-0.196	0.196	6
track valid	-	-	-	-	1
consistent	-	-	-	-	1
out of range	-	-	-	-	1
Reserved	-	-	-	-	1
Tot.	96b				

A total of 12 Control Units have been implemented per [TFP](#), each of them handling six Track Locators and four [SF+LR](#) blocks. Each [SF+LR](#) runs the track filtering/-fitting algorithm discussed in Section [3.3](#), and produces track candidates, which are stored in an internal FIFO until a read-out signal is received from the top module. The time of arrival of this signal is the same for all the [SF+LR](#) blocks and defines the latency of the design, defined as the difference between the time in which the first stub enters the design and the time in which the first track is buffered out.

Tracks are then transmitted into the [DR](#) router block, which, in a similar way to what happens in the [GP](#), sorts the track candidates with respect to their sub-sector and routes them to the respective [DR](#) block. Here, the Duplicate Removal algorithm has been implemented in order to reject all the duplicate tracks produced. Surviving candidates are then transmitted out across 36 output channels, one per sub-sector in the processing octant. Each output channel consists of three MP7 output links, which transmit a track per clock cycle. Table [4.6](#) shows the output track data format. Output candidate information is encoded in a 96-bits word.

The overall design has been implemented successfully in two Virtex-7 XC7VX690T FPGAs, running at a frequency of 240 MHz.

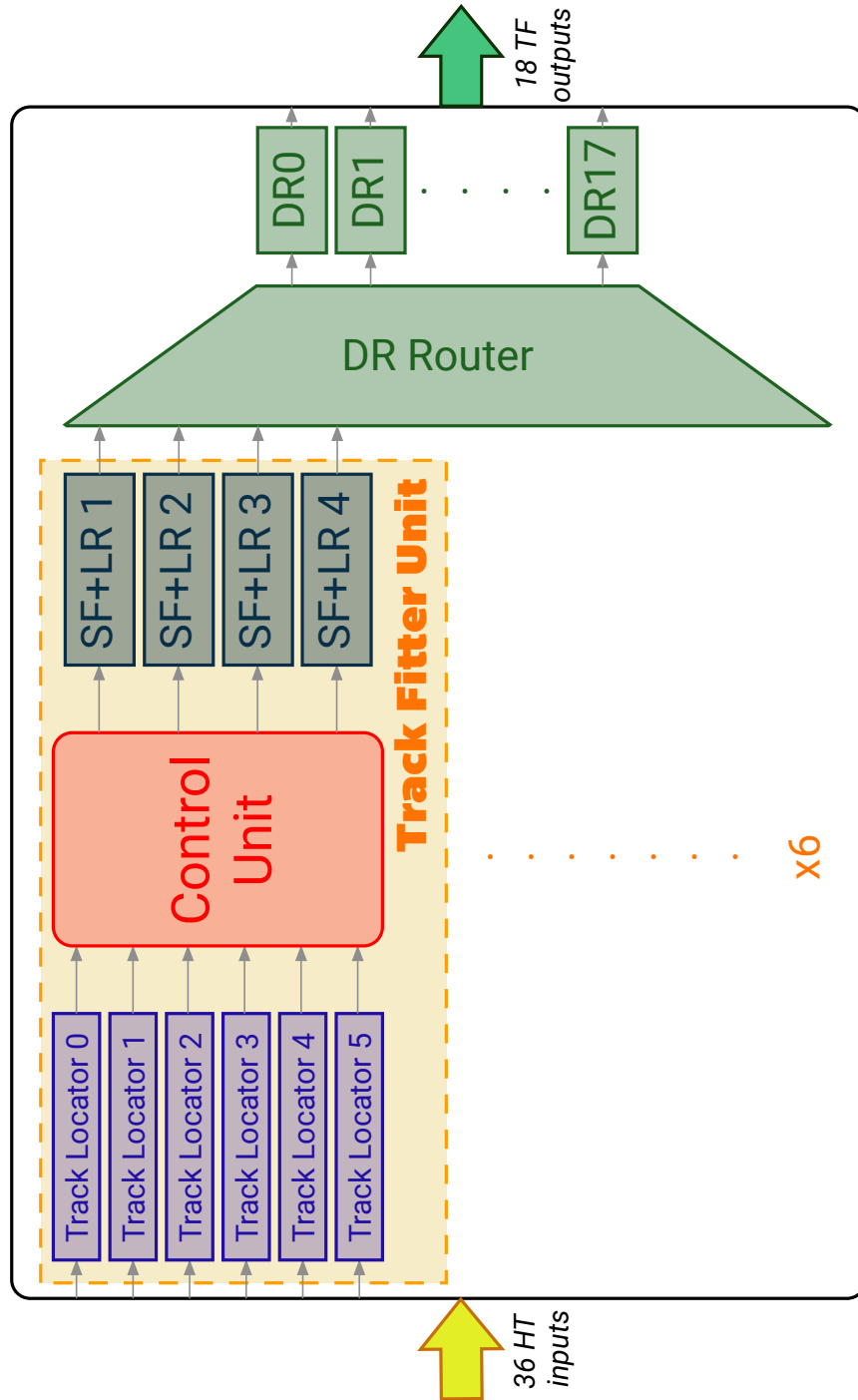


Figure 4.12: Overview of the Seed Filter plus Linear Regression (SF+LR) firmware. Stubs coming from the HT output channels are temporarily stored in input FIFOs (Track Locators). A Control Unit object readouts the tracks from the Track Locators and distributes them to the SF+LR blocks. To optimise the parallelisation of the firmware, each Control Unit manages six Track Locators and four SF+LRs (Track Fitter Unit). A total of six Track Fitter Units have been implemented on a Virtex-7 FPGA. Produced tracks are then transmitted to a Duplicate Removal (DR) Router, which forwards them to 18 DR modules that reject the duplicate candidates. Surviving tracks are finally output through 18 channels.

4.4.2 Track Locators

The Track Locator logic is responsible for storing stubs belonging to HT track candidates temporarily before they are distributed to the SF+LR modules by the Control Unit. Figure 4.13 shows a simplified schematic diagram of the Track Locator design.

Stubs enter the module at each clock cycle and they are stored into a dual-port synchronous 256-bit $\times$ 64-bit Block RAM buffer. Two buffers are present for each Track Locator since they are required to handle multiple events. Stubs are sorted in the two RAMs by the *Event Switcher* logic. The input packet size is fixed to be 210 clock cycles long (about 850 ns), followed by gap of 6 clock cycles (25 ns). Whenever there is a gap in data stream the Event Switcher receives an event switcher signal from the top module (*EventId*), which defines the buffer to be filled.

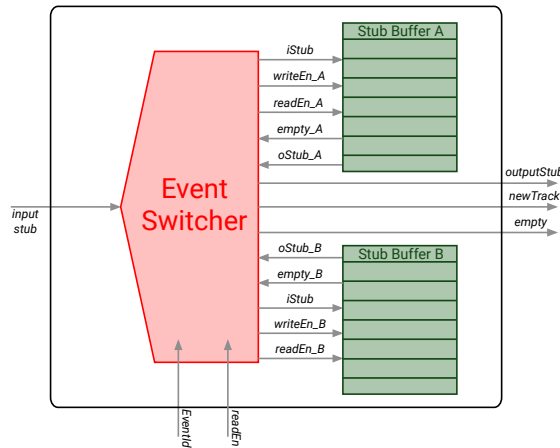


Figure 4.13: A Track Locator logic component. The module is provided with two BRAM buffers (*Stub Buffer A/B*) to store stubs from two consecutive events. The arrival of a not-valid stub into the *Event Switcher* block flags the end of the event. The Event Switcher will, therefore, switch the enabling and disabling of the read/write signals in the two buffers.

Once the Track Locator receives the *readEn* signal from the Control Unit, it begins to readout stubs from the corresponding buffer. Two other signals are transmitted to the Control Unit: the *NewTrack* signal, which notifies the Control Unit that the next stubs will belong to a different track candidate, and the *empty* signal, which informs that the buffer has been read out completely.

Table 4.7 shows the resource utilisation of a Track Locator module as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. A total of 36 Track Locators have been implemented in the current design in each MP7 board.

Table 4.7: Resource utilisation of one Track Locator module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 72 Track Locators, split in the hardware demonstrator across two MP7s.

	LUTs	DSPs	FFs	BRAM (36 Kb)
Track Locator	123 (0.0%)	0 (0.0%)	315 (0.0%)	2 (0.1%)

4.4.3 Control Unit

The Control Unit distributes stubs from the Track Locators to the downstream SF+LR modules for processing. It basically acts as a multiplexer. Figure 4.14 shows a schematic diagram of the Control Unit design.

Each Control Unit receives as input from the Track Locators the *NewTrack* and the *Empty* signals. A *readEn* signal is sent to the first, not empty Track Locator, which will start to transmit its stubs to the Control Unit. Stubs will be then propagated to all SF+LR modules, which store and process them if a positive *FilterEnabler* signal is also received. Stubs from the first track will be processed by the first SF+LR, from the second track by the second SF+LR and so on, in a round robin fashion. Once the last SF+LR module has received a track, the transmission process will start again from the first Seed Filter.

The enabling/disabling process of the Track Locators and Seed Filters has been implemented in firmware by means of a simple FSMs.

Table 4.8 shows the resource utilisation for a single Control Unit logic. Each Control Unit can handle a configurable number of Track Locators and SF+LR modules, which have been currently fixed to respectively six and four, for a total of 12 Control Units, splits across the two dedicated MP7 boards.

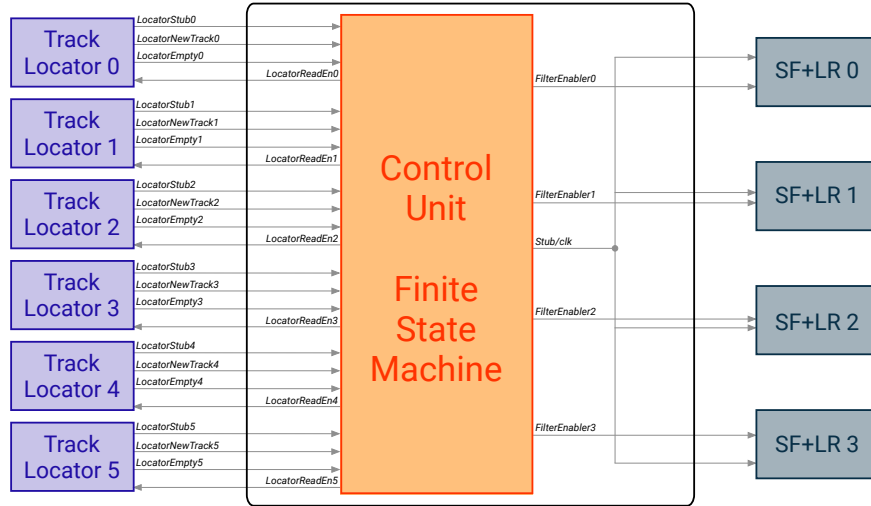


Figure 4.14: Overview of the Control Unit module. It has been designed as a simple FSM, which activates and deactivates the readout from the six Track Locators. Group of stubs with the same HT parameters (m_{bin} , c_{bin}) are transmitted to the four SF+LR modules in a round-robin fashion.

Table 4.8: Resource utilisation of a single Control Unit module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 12 Control Units, split in the hardware demonstrator across two MP7s.

	LUTs	DSPs	FFs	BRAM (36 Kb)
Control Unit	159 (< 0.05%)	0 (< 0.05%)	427 (< 0.05%)	0 (< 0.05%)

4.4.4 The Seed Filter plus Linear Regression

The Seed Filter plus Linear Regression is the core module of the entire firmware. It implements both the track filter and fitter algorithms. Figure 4.15 shows a simplified block diagram of the module workflow.

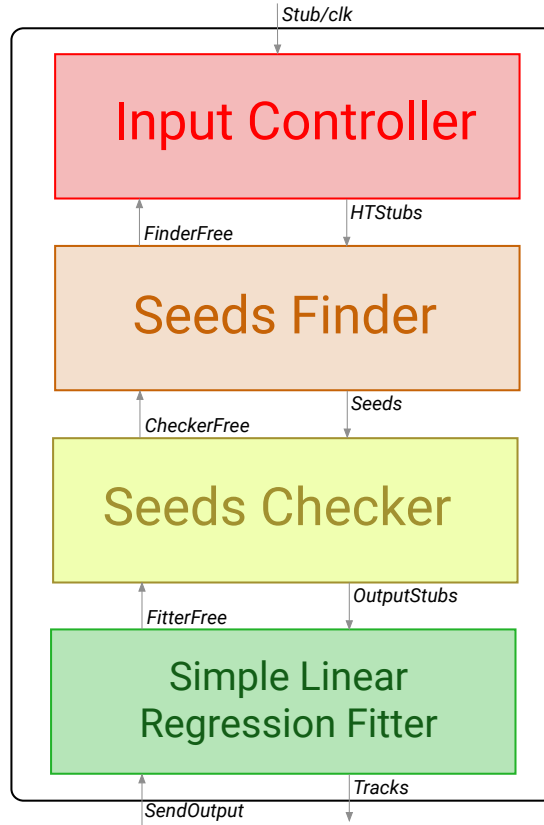


Figure 4.15: Block diagram of the Seed Filter plus Linear Regression logic. The design can be split in four stages, each representing a different step in the algorithm. Each stage has to wait the arrival of the *Free* signal only from the successive block, before transmitting/receiving new data.

The algorithm implementation has been subdivided into four separate phases:

1. **Input Controller.** Track candidate stubs are stored in a Block RAM buffer. Track candidate are read out one at a time and sent for processing to the second phase.
2. **Seeds Finder.** Pairs of PS stubs, belonging to different tracker stations, form here the *Seeds*, which are then transmitted to the downstream stage.

3. **Seeds Checker.** Each seed is beforehand verified to be compatible with the beam spot length and the sub-sector definition. Those that pass both criteria are then extrapolated to the other tracking stations, in order to verify the compatibility of the other stubs in the candidate with the same seeding line. Subsequently only the best seed is kept and its associated stubs are transmitted to the Simple Linear Regression fitter for the final processing.
4. **Simple Linear Regression.** Implementing the Best-of-Both SLR algorithm, it cleans further the track candidate from spurious stubs and calculates the fit parameters. Produced track candidates are then stored in a FIFO, which is read out once the *SendOutput* signal is received.

Seed Filter

Figure 4.16 displays a detailed schematic of the Seed Filter implementation. The overall module can be seen as three separate state machines, communicating with each other, which handle the first three phases described above. The separation of the algorithm into those phases allows a better pipeline, thus reducing the required processing times. The basic idea is to make each phase independent of each other in a way that an upstream phase can start to process the next track candidate, while the downstream phase is still dealing with the previous track.

INPUT CONTROLLER. The first phase (*Phase 0*) consists of storing the stubs received by the Control Unit. Input stubs are saved in a dual port 64-bit $\times$ 64-bit Block RAM buffer (*Input Stub Buffer*), which is controlled by the *Input Controller* state machine. This sends the *InputReadEn* signal as soon as at least one stub is present in the buffer. Reading from the buffer is halted every time a new track arrives, i.e. every time at least one of the stub m_{bin} , c_{bin} or sector iD variables changes its value. A new track candidate can be read out only once the Seed Finder State Machine is ready to process another track (*FinderSMFree*).

SEED FINDERS. Stubs from the Input Stub Buffer (*HTStub*) are then stored in a temporary dual port 16-bit $\times$ 64-bit Block RAM buffer (*HT Stub Buffer*) and transmitted at the same time to the 15 *Seed Finder* modules. Each Seed Finder is responsible for creating pairs of PS stubs, i.e. Seeds, which will be stored in the *Seed Buffer*. Figure 4.17 shows how a Seed Finder module operates.

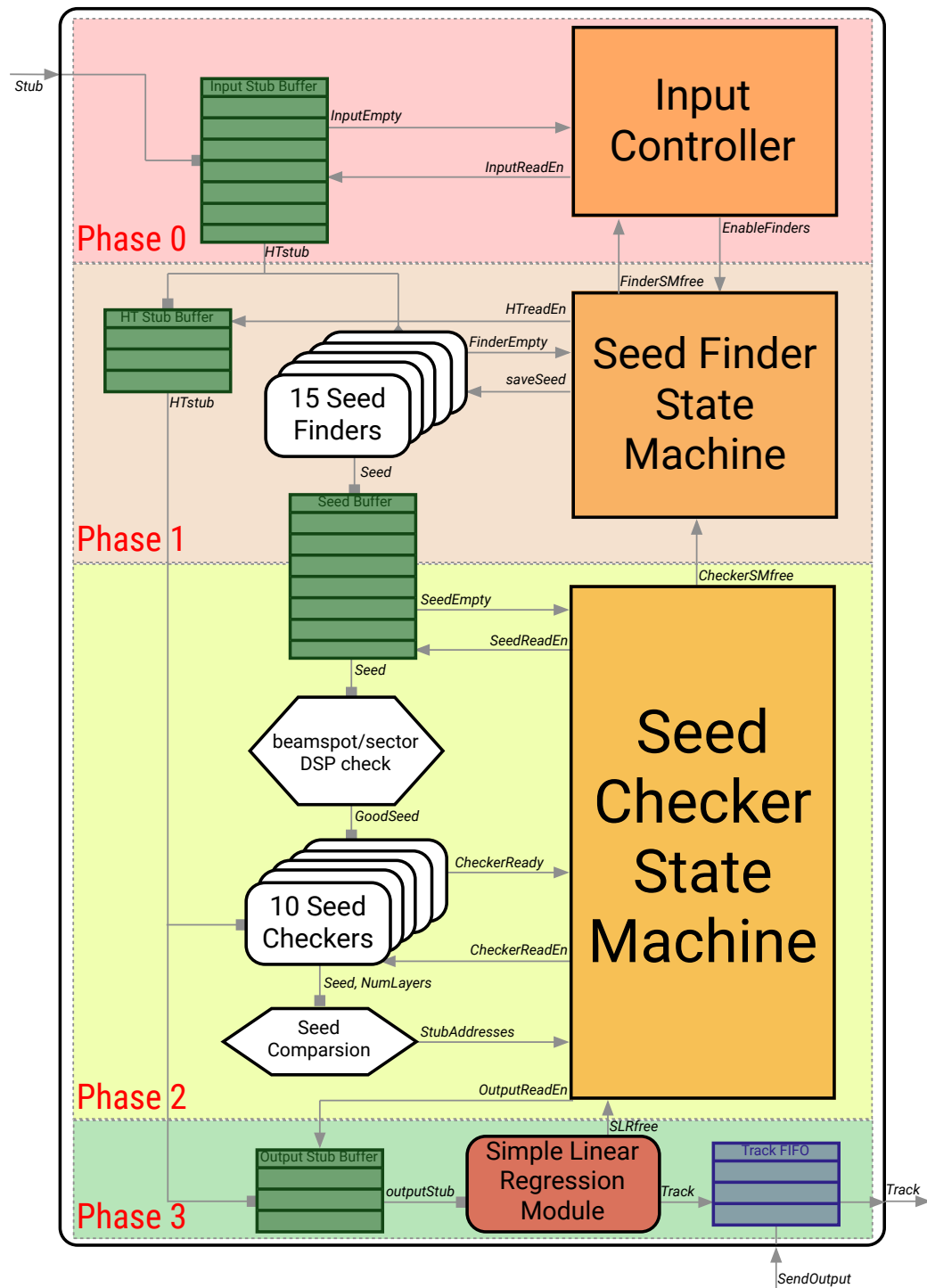


Figure 4.16: Overview of the Seed Filter plus Linear Regression design. Internal components are shown as round boxes, FSMs as squared boxes, FIFOs as blue boxes and BRAM as green boxes. The design can be seen split into four phases, resembling the steps of the algorithm. A detailed description is provided in the text.

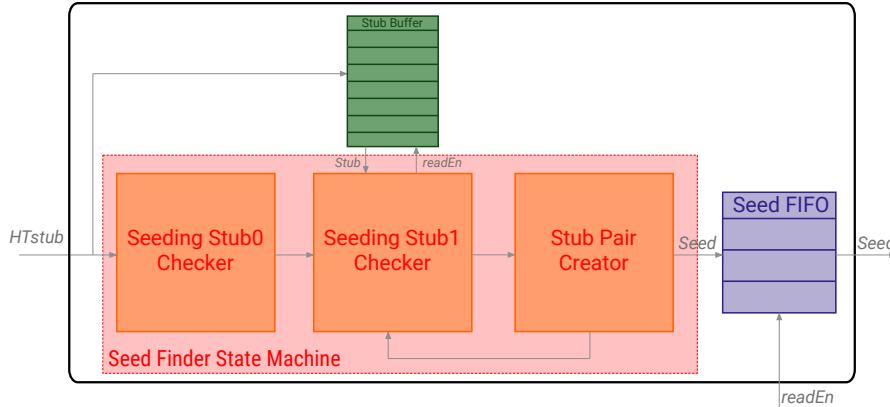


Figure 4.17: Overview of the Seed Finder module. Each Seed Finder receives in input a copy of the stubs which are stored in *Stub Buffer*. By means of a *FSM* it creates pairs of stubs (*seed*) belonging to PS modules in different stations of the Outer Tracker. The first seeding stub (*Stub0*) is always the stub at the address i in the *Stub Buffer*, where i corresponds to the index of the Seed Finder. If *Stub0* is actually a seeding stub, the module loops over the buffer starting from address $i + 1$, looking for other seeding stub candidates (*Stub1*). Pairs of seeding stubs are then stored in a FIFO (*Seed FIFO*), readout by the top module. A maximum of three seeds can be stored in the FIFO per Seed Finder.

Each Seed Finder stores a copy of the HT stubs in an internal dual port 16-bit $\times$ 64-bit Block RAM buffer (*Stub Buffer*). The seed pair creation has been implemented as a simple three-state machine. The first state (*Seeding Stub0 Checker*) checks that the first considered stub belongs to a PS module. The first stub is different and fixed for each Seed Finder module, corresponding to the stub at position i in the HT Stub Buffer, where i is the Seed Finder index.

If the first stub satisfies the requirement, the Seed Finder will start to loop over all remaining stubs in the buffer starting from position $i + 1$ (*Seeding Stub1 Checker*). Any time a compatible stub is found a seed pair is created, consisting simply of the addresses of the two stubs in the RAM (*Stub Pair Creator*). The found seed is then stored in an internal 4-bit $\times$ 4-bit FIFO (*Seed FIFO*), which is read out once the *readEn* signal is received from the upstream *Seed Finders State Machine*. Since a limit on the number of seeds per Seed Finder has been set to three, there is no need for a bigger FIFO.

Seeds are readout from each Seed Finder starting from the first not empty unit and stored into a small dual port 16-bit $\times$ 4-bit Block RAM buffer (*Seed Buffer*) since only the first 15 seeds will be considered for each track candidate.

SEED CHECKERS. As soon as at least one seed is present in the Seed Buffer, *Phase 2* can begin. Stubs are immediately copied into another 6-bit $\times$ 64-bit Block RAM buffer (*Output Stub Buffer*) and sent to the *Seed Checker* modules. Once all stubs have been saved, a signal is sent to Seed Finder State Machine (*CheckerSM-free*), which is now allowed to start processing a new track.

In *Phase 2* seeds are readout from the Seed Buffer and transmitted through a DSP calculation chain (*beamspot/sector DSP checker*), which verify the compatibility of the seed considered with the beam spot length and the sector definition as in eq. 34. The use of divisions in FPGA should be avoided, unless it is completely necessary, since their firmware implementation requires a relatively high number of resources and high latency. Therefore, in order to avoid divisions, the implemented equations are,

$$|z_1 \cdot (r_1 - r_2) - (z_1 - z_2) \cdot r_1| < w \cdot |r_1 - r_2|, \quad (54)$$

$$z_S^{\min} \cdot (r_1 - r_2) < z_1 \cdot (r_1 - r_2) - (z_1 - z_2) \cdot (S - r_1) < z_S^{\max} \cdot (r_1 - r_2). \quad (55)$$

Each *GoodSeed* is sent to a different Seed Checker block. Simulation studies have shown that a limit on the number of Good Seeds equals to 10 does not affect the tracking performance. Therefore 10 Seed Checker instances have implemented into the current design.

Each Seed Checker contains a copy of the track candidate stubs, stored in another BRAM buffer. Once it receives the seeding stubs information, stubs are read out from the buffers and sent through a DSP chain, which checks the compatibility of the stubs with seeding lines according to equation 36.

A record of the activated tracking stations is kept here for each track candidate, maintaining only one stub per layer, based on first-come-first-served fashion.

Each Seed Checker will then transmit out information about how many tracking stations have been activated, together with pointers to the stubs that fired

them. Subsequently a comparison between results from the different Seed Checkers takes place, which selects the first seed with the more activated stations. Stubs compatible with the chosen seed are finally retrieved from the Output Stub Buffer and sent to the *Simple Linear Regression* module for the track fitting operations.

The Simple Linear Regression module

The Simple Linear Regression module consists of a finite state machine, which handles the communications between the several components in the design. Figure 4.18 shows a schematic working diagram of the SLR design. Four main components can be identified, resembling the track fitting algorithm steps listed in Section 3.3.2.

1. **R Φ Helix Calculator.** Stubs are sent through a DSP chain, which measures only the Φ_T and q/p_T fit parameters.
2. **R Φ Residual.** Residual distance from the $r\phi$ fitting line is calculated for each stub according to the just measured fit parameters. All stubs with a residual value larger than a predefined threshold are kept or the four stubs with smallest residuals. The module also checks that at least two stubs from PS tracking modules are maintained.
3. **R Φ /RZ Helix Calculator.** Surviving stubs are used to calculate fit parameters on both $r\phi$ and rz planes. For the latter, only PS stubs are utilised. Both calculations have been implemented using a DSP chain.
4. **χ^2 Calculator.** Residuals are measured again on both planes, and the χ^2 is then calculated using DSPs.

The most challenging part of this design is the implementation of the required divisions for the calculation of the track fitting parameters. These have been implemented by means of LUTs, storing the reciprocal values of the two denominators. Two different divider precisions have been adopted for the $r\phi$ and rz cases, equal to respectively 20-bit and 16-bit. Further details on how divisions have been implemented in the SF+LR firmware are presented in Appendix B.

Resource Utilisation

Table 4.9 shows the resource utilisation for a single SF+LR module as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. Also resources used by the individual

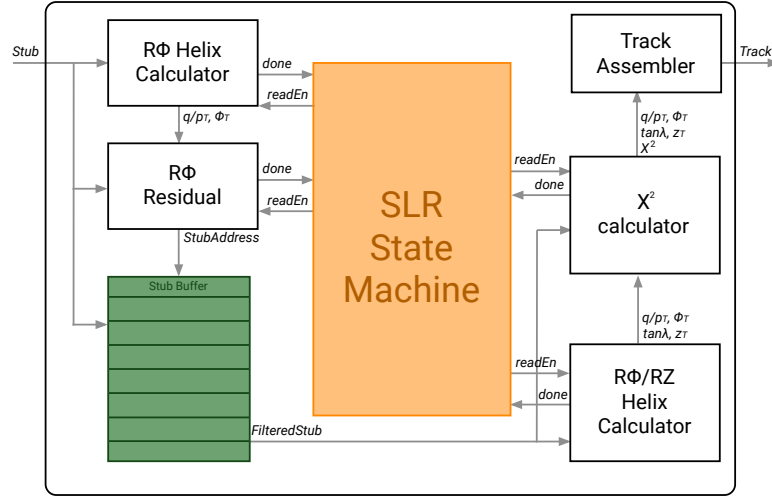


Figure 4.18: Overview of Simple Linear Regression module. The design consists of a state machine that deals the communications between the different components, running each an individual step in the fitting algorithm. More details are provided in the text.

Seed Finder, Seed Checker and Simple Linear Regression modules are presented. The design makes a large use of LUTs. This is due to the fact that in some part of the firmware LUTs are preferred to BRAMs to store stub information, because they guarantee a faster access to the data and therefore lower latencies for the design to meet timing requirements. Future improvements can be made in order to have a more balanced resource utilisation.

Table 4.9: Resource utilisation of a Seed Filter plus Linear Regression module, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 48 Seed Filters, split equally in the hardware demonstrator across the two MP7s. Also, the resources utilised by the single Seed Finder and Seed Checker components are presented.

	LUTs	DSPs	FFs	BRAM (36 Kb)
Seed Finder	107 (0.0%)	0 (0.0%)	112 (0.0%)	0 (0.0%)
Seed Checker	479 (0.1%)	7 (0.2%)	380 (0.0%)	0.5 (0.0%)
SLR	2097 (0.5%)	31 (0.8%)	1259 (0.1%)	0.5 (0.0%)
SF+LR	9267 (2.1%)	107 (3.0%)	7808 (0.9%)	9 (0.6%)

Figure 4.19 shows how the 24 SF+LR modules have been placed across the Virtex-7 FPGA [84]. Because of the relatively small amount of resources needed by each SF+LR block, it would be possible even to implement more than 24 instances. How-

ever, the chip has to accommodate also the DR firmware, which needs a large router to communicate with all the Seed Filters. Implementing more Seed Filters would make very difficult the design to close up the timing, because of the very complicated routing. Moreover, as it will be shown in the results section, it has been found that 24 SF+LR are sufficient to obtain excellent track finding performance, without significant losses due to truncation effects.

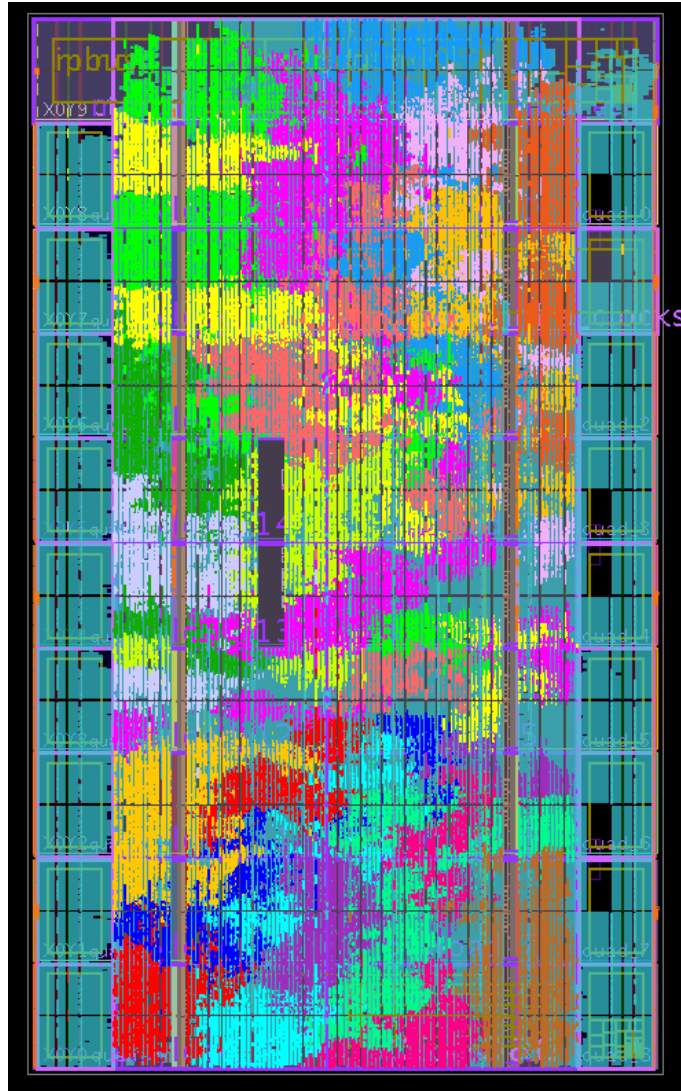


Figure 4.19: Floor planning of the implemented SF+LR+DR design in a Virtex-7 690 FPGA [84]. Each colour represents a separate SF+LR module. The fitters are spread along the entire chip to optimise the routing from the Track Locators.

4.4.5 Duplicate Removal

The Duplicate Removal step has been implemented on the same FPGA chip as the Track Filter/Fitter. As described in Chapter 3, the Duplicate Removal algorithm has to have access to all the track candidates belonging to an individual sub-sector in order to work properly. For this reason, it requires a routing stage at the output of the SF+LR, which assigns the track candidates to their relative sub-sectors.

The DR Router has been implemented in firmware as a simplified version of the GP router. Only two routing stages are here required, sorting the tracks according to their sector iD. Each sub-sector is then processed by an individual DR block. Therefore, according to the current geometrical division of the Tracker, 36 DR blocks are required per TFP.

Figure 4.20 shows the firmware architecture of a single DR block. Tracks input to the DR will fill a 18 Kb memory (*Matrix*), which represents the HT array, at the address corresponding to the $(q/p_T, \Phi_T)$ coordinates within the HT array. The track fitter stage marks as *consistent* the tracks whose track fit parameters are compatible with the HT measurement. Those consistent tracks are immediately forwarded to the output channel and the corresponding Matrix address is marked.

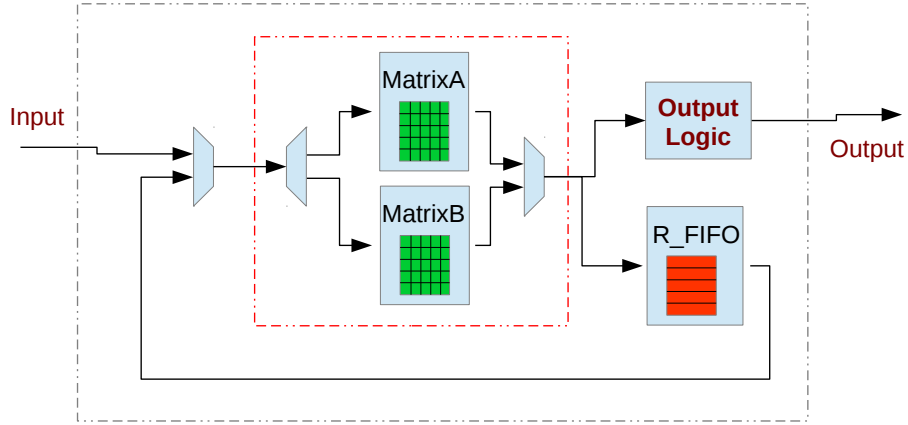


Figure 4.20: Architecture of a single Duplicate Removal (DR) block, processing tracks of a single (η, ϕ) sub-sector (*Input*). Therefore 36 DR modules are needed to process the entire octant.

Inconsistent tracks are instead stored in a FIFO (*R_FIFO*). After a fixed configurable latency, set to 140 clock cycles (580 ns) in the current design, tracks would be rescued from the R_FIFO and, if their track parameters do not correspond to a marked cell in the Matrix, they are forwarded as well to the output channel.

Two matrices are needed in the design in order to allow reset and process alternate LHC events. Table 4.10 reports the resource utilisation for the DR router and for a single DR block. The design itself is relatively light, having also a low latency of only 4 clock cycles at the DR block stage.

Table 4.10: Resource utilisation of DR router and of a single DR block, as implemented in the Xilinx Virtex-7 XC7VX690T FPGA. The usage as a percentage of the device's available resources is shown in parenthesis. The entire TFP needs 36 Duplicate Removal blocks, one per each sub-sector of the processing octant, split equally in the hardware demonstrator across the two MP7s.

	LUTs	DSPs	FFs	BRAM (36 Kb)
DR Router	22246 (5.1%)	0 (0.0%)	53989 (6.2%)	357 (24.2%)
DR Block	370 (0.1%)	0 (0.0%)	481 (0.0%)	4 (0.3%)

4.5 DEMONSTRATOR RESULTS

Simulated $t\bar{t}$ events with 200 PU have been used to test the demonstrator system. The following sections show the results obtained in hardware, compared with the expectations given by the software emulation. The word *emulation* is used in an inappropriate way, since the software used for comparison replicates only the functionalities of the implemented algorithm with the same digitised precision without providing any information about the evolution in time of the actual signals. Hence, any truncation effect due to latency constraints cannot be reproduced by the software. These truncation effects will be discussed further in the following sections.

4.5.1 Output data rate

As previously mentioned, the Track Finder Processor firmware imposes a limit on the number of stubs at the output of the GP and HT stages, equals respectively to 175 and to 202. Figure 4.21a compares the total number of reconstructed tracks, measured using the emulation software, with and without truncation. Figure 4.21b shows the resulting tracking efficiency as a function of the generated track $1/p_T$. For the majority of events truncation effects can be essentially neglected, being the average number of tracks found by track finding algorithm practically the same with or without truncation (102.5 and 103.9 respectively). However, some issues

arise in the case of high- p_T jets, whose tracks end up in a single sub-sector. Here the tracks with the lower p_T are truncated, resulting in a drop in tracking efficiency at low p_T .

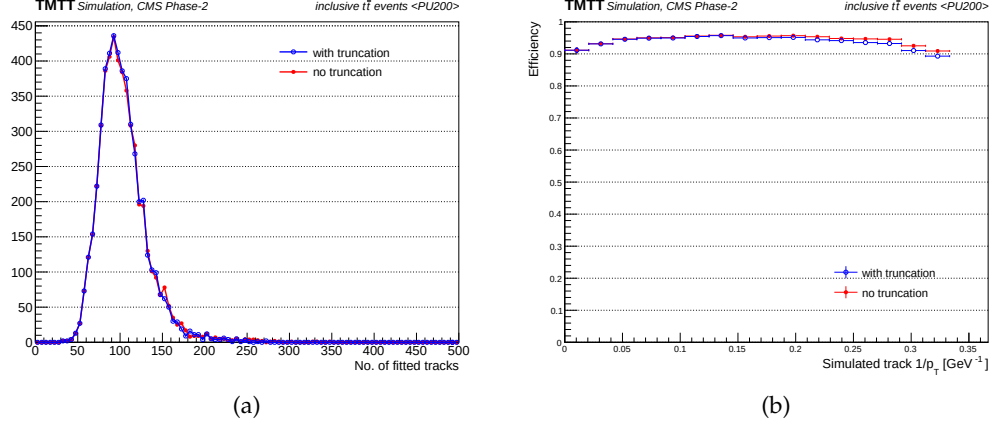


Figure 4.21: (a) Number of fitted tracks and (b) track finding efficiency, in top pair production events with 200 pileup. The plots show the effect of truncation due to the limited size of the input stub packet (blue dots), compared with performance obtained with an limitless packet (red dots). Those results have been obtained in simulation.

Some truncation effects can occur also at the output of the Track Fitter and Duplicate Removal stage. Unfortunately, these cannot be replicated in software, due to the high complexity of the Seed Filter firmware architecture, whose processing time is different for each track candidate. Estimation of this truncation would require an exact emulation (clock-wise and bit-wise) and time constraints did not allow this to be developed. In any case, comparing the total number of tracks out of the demonstrator system with the emulation results, it is possible to get a rough estimation of these truncation effects. It must be noted nevertheless that differences can be also due to other factors, i.e. bugs in the code or different precision in the calculations. Figure 4.22 show the distributions of track candidates in 5,000 events, in hardware and emulation, as a function of the Hough Transform array coordinates (m_{bin}, c_{bin}) and of the estimated track fit parameters ($q/p_T, \Phi_T, \tan \lambda$ and z_T).

The demonstrator system finds on average 100.9 track candidates, against the 102.5 reconstructed by the emulator. About 95% of hardware tracks match with a track found by the software, revealing that in most cases the algorithm is behaving in the same way in both firmware and emulation.

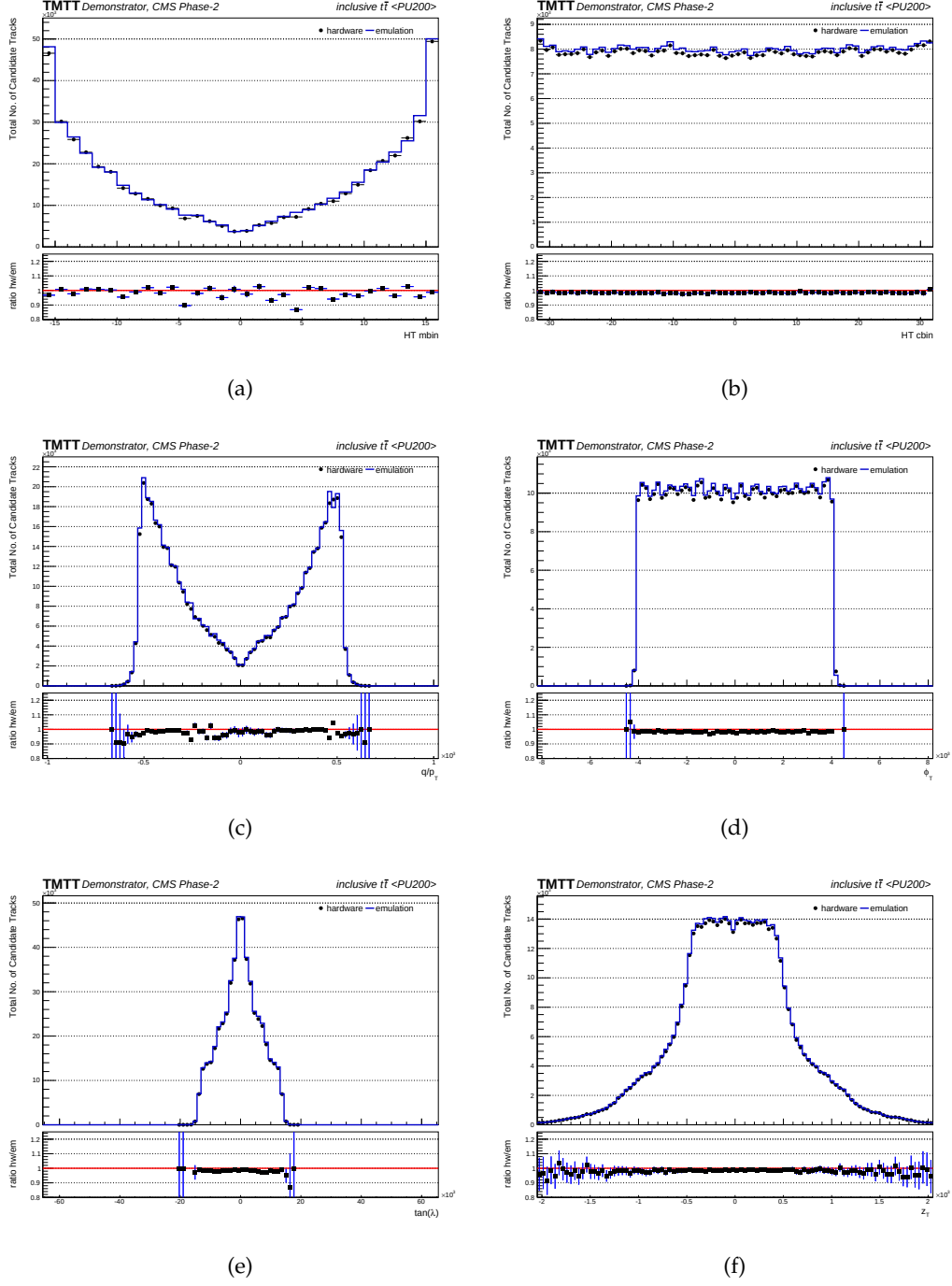


Figure 4.22: Distribution of track candidate as a function of the HT indices (a, b) and the fitted helix parameters (c, d, e, f), at the output of the TFP, for $t\bar{t}$ events with PU200. Results obtained with the hardware demonstrator (dots) are compared with expectations from software emulation (blue line). A 95% matching has been observed.

4.5.2 $L1$ track reconstruction efficiency

Figure 4.23 shows the track finding efficiency distributions as a function of the pseudorapidity and the transverse momentum of the generated tracks in both hardware and emulation. Even if the measured hardware efficiency is still high (about 94.0%), it is slightly lower than in software. The loss in efficiency is due to several factors, the main one being the truncation imposed by the track fitter stage that is not replicated in software. Other differences can arise by different implementation or precision of the algorithms in firmware and emulation.

Despite these dissimilarities, the demonstrator system has been able to prove in hardware that it is possible to reconstruct tracks with $p_T > 3 \text{ GeV}$ in the harsh HL-LHC environment, within the required latency budget.

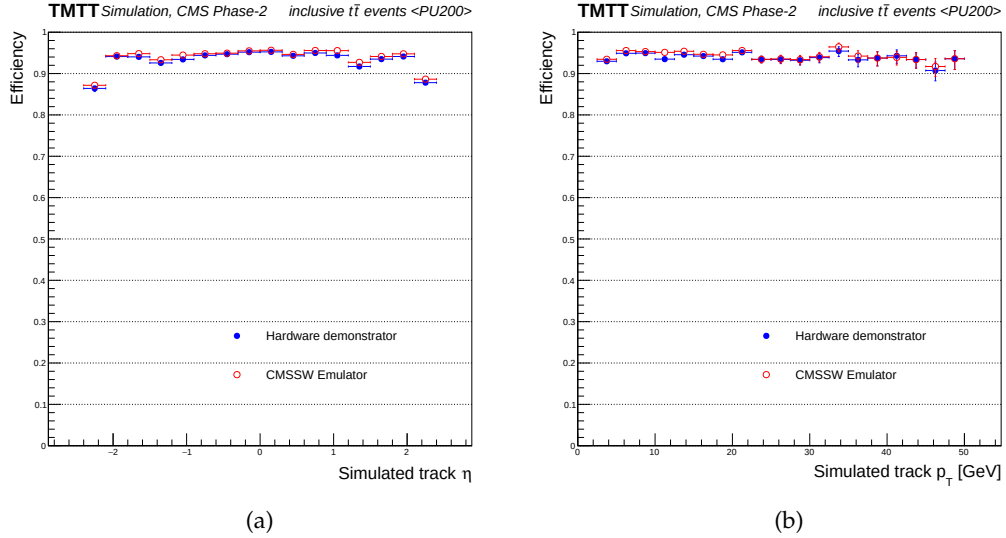


Figure 4.23: Track Finding efficiency in $t\bar{t}$ events at a pileup of 200, as a function of the simulated track pseudorapidity (a) and transverse momentum p_T , measured in hardware (blue dots) and software emulation (red dots). Comparable efficiency have been obtained, with differences due to not-replicable truncation effects and in the use of floating point calculations in some parts of the emulator.

4.5.3 Track Fit parameters resolution

The resolution of the track fit parameters is shown in Figure 4.24 and in Figure 4.25, for both hardware and emulation. The level of agreement between the two is very good in all four distributions, with some remaining differences due to the use of floating point calculation in some parts of the emulator code.

Parameter resolutions on the $r\phi$ plane are comparable to what has been obtained using a full floating point precision simulation, as shown in Section 3.4.2. For the rz fit parameters, slightly worse resolutions have been obtained, due to an inappropriate choice of the number of bits used to encode the stub z coordinate. Figure 4.26 shows how better resolution can be easily achieved by adding one or two extra bits to encode the z and r_T stub coordinates.

4.5.4 Latency

The total latency of the system, defined as the time difference between the moment the first stub enters the TFP and the moment the first track comes out, has been measured independently for each block and also for the total chain and is reported in Table 4.11. Also, the latency due to the optical transmission delays and serialisation/de-serialisation (SERDES) on the links is shown.

Identical results have been obtained when measuring separately the latency for each individual block or in the entire demonstrator chain, from source to sink. The latency of the system is fixed and it is independent of the event characteristics.

It is also important to measure the time difference between the first track and last track coming out of the system, which defines the size of the output track packet. The sum of those two latency measurements must not be greater than the prefixed budget of $4\text{ }\mu\text{s}$, which is the time assigned to the Track Finding system downstream to the DTCs.

The table shows also the latency measured with a TFP demonstrator system, which exploits a Kalman Filter for the track fitting task.

Both approaches are well within the time budget. The SF+LR full chain has been found to require about 150 ns less than the KF chain to process completely a single event. It is possible to observe, on one hand, how the KF output packet size is much smaller than in the SF+LR. That is because the KF firmware accumulates all the reconstructed tracks before transmitting them out.

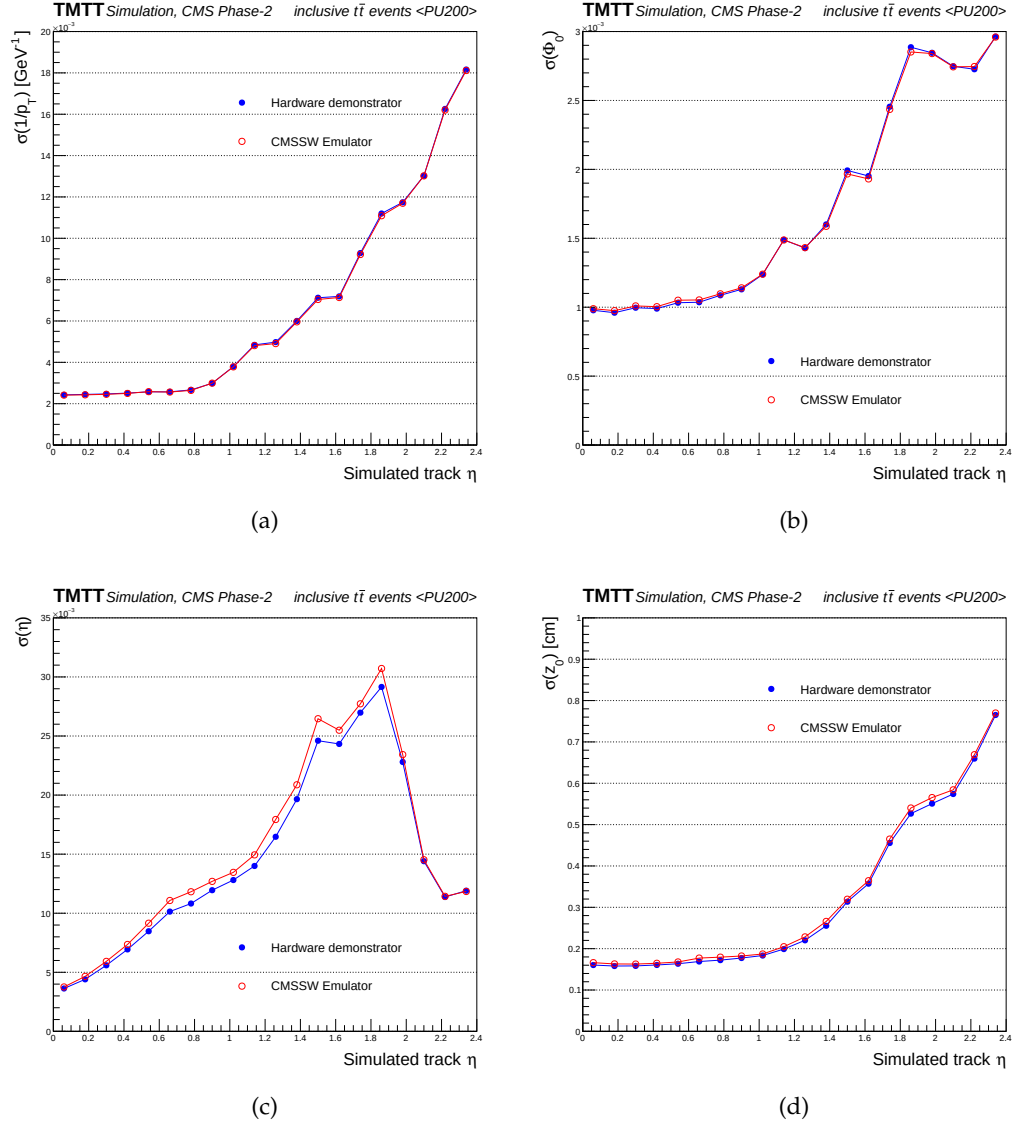


Figure 4.24: Track helix parameter resolutions as a function of the simulated track pseudorapidity η , in $t\bar{t}$ events at a pileup of 200, as measured in hardware (blue dots) and software emulation (red dots).

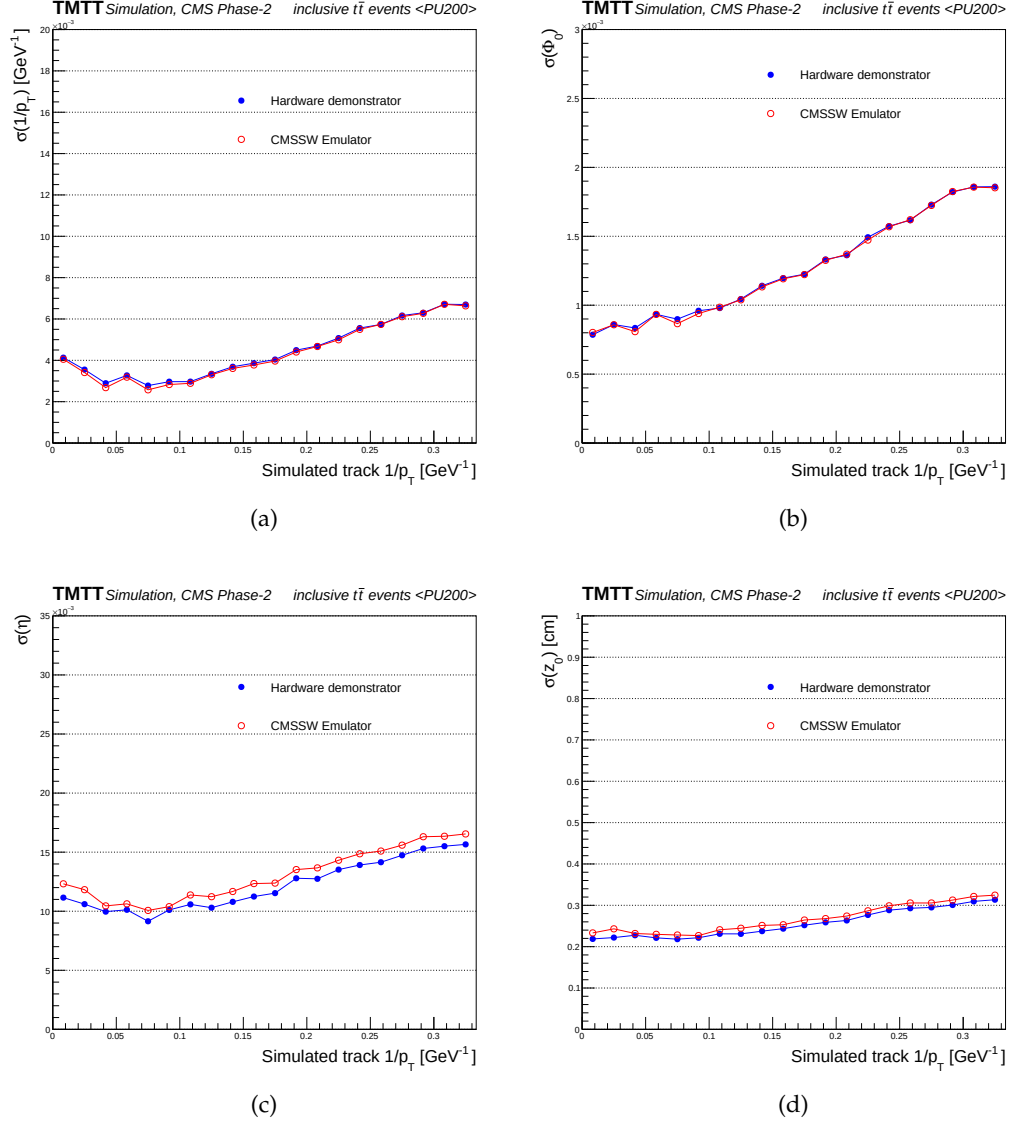


Figure 4.25: Track helix parameter resolutions as a function of the simulated track transverse momentum p_T , in $t\bar{t}$ events at a pileup of 200, as measured in hardware (blue dots) and software emulation (red dots).

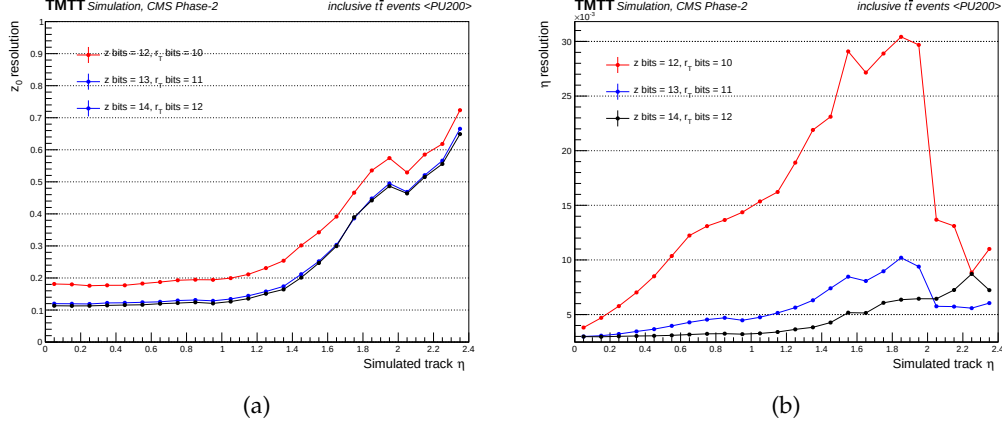


Figure 4.26: Resolution of the r - z helix parameters, for different widths of the stub coordinates (r_T, z), in $t\bar{t}$ events at a pileup of 200. Performance comparable to a full floating point simulation (see Fig. 3.32) can be obtained simply by assigning two extra bits to each variable. Those results have been obtained using the software emulation.

Table 4.11: Measured latency of the demonstrated components of the track reconstruction chain, including the serialisation/de-serialisation (SERDES) and optical transmission delays between each board.

System Latency	Latency [ns]
SERDES + optical length 1	143
GP	251
SERDES + optical length 2	144
HT	1025
SERDES + optical length 3	129
SF+LR + DR	875
KF + DR	1658
SERDES + optical length 4	129
Total SF+LR Chain: First out - First in	2686
Output Packet SF+LR Chain: Last out - First out	875
Total SF+LR Chain: Last out - First in	3571
Total KF Chain: First out - First in	3479
Output Packet KF Chain: Last out - First out	225
Total KF Chain: Last out - First in	3704

On the contrary, the **SF+LR** continues to process track candidates while sending out the already reconstructed ones. The firmware has 210 additional clock cycles to process as many tracks as possible before a new event arrives. Figure 4.27 shows the distribution of the last-out-first-out latency for the **SF+LR** firmware. In most of the cases, the **SF+LR** does not require the entire time to handle all the track in the octant. The average output packet size is indeed of just 79 clock cycles, corresponding to 329 ns. A spike can be seen at about 600 ns, which is the moment the **DR** would start its second loop over the inconsistent tracks, whose fit parameters are not compatible with the Hough Transform coordinates.

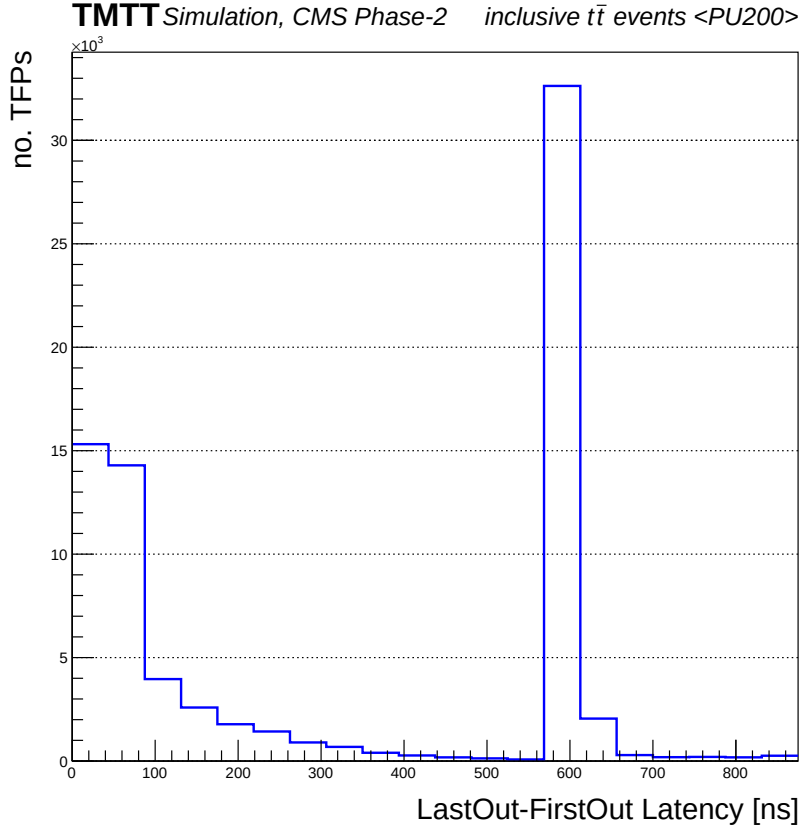


Figure 4.27: Time distribution of tracks along the output packet, in top pair production events with pileup 200. Majority of the tracks are transmitted out at two times. The first block is sent out at the beginning of the packet, corresponding to *consistent* tracks. The second group will arrive at about 600 ns, corresponding to the moment in which the second pass of the **DR** algorithm takes place.

4.5.5 FPGA resource usage

Since the hardware demonstrator is constituted of many MP7 boards, the logic constraints of a single FPGA processing board are avoided. However, the overall resource utilisation must remain realistic, aiming to have a final firmware architecture that will be implemented on one or two future FPGA chips.

Table 4.12 shows the total FPGA resource usage for each demonstrator component (where the numbers given for the HT and Track Fitter implementations are summed across the two boards used for each component). Resource utilisation for both SF+LR and KF approaches are reported.

The sum of the three components gives an estimation of the required resources for a single TFP board with a time-multiplexed factor of 36. In order to meet timing and routing constraints in the Virtex-7, the designs often prioritised the use of LUT based distributed memory over Block RAM, or vice-versa. This balance could be readdressed in the future, as the design is adapted to newer FPGAs.

Table 4.12: Total resource usage for the demonstrator TFP (with time-multiplexed factor $n=36$), as implemented in the Xilinx Virtex-7 XC7VX690T FPGA [84]. The resources needed to construct a complete TFP correspond to the sum of the numbers from the three rows labelled GP, HT, and KF and DR.

	LUTs [10^3]	DSPs	FFs [10^3]	BRAM (36 Kb)
GP	121	1056	205	222
HT	244	2304	299	1188
SF+LR and DR	512	5184	526	1434
KF and DR	398	5112	316	1776
Infrastructure per MP7	90	0	91	291
SF+LR TFP Total (excl. infra)	877	8554	1030	2844
KF TFP Total (excl. infra)	763	8472	820	3186
Virtex-7 690	433	3600	866	1470

4.6 FINAL SYSTEM SCALING

The built demonstrator system has proved that it is possible to perform track finding reconstruction at HL-LHC with available technology. A future system will furthermore benefit from the many improvements the future FPGA chips will carry.

A new demonstrator system is therefore planned to be built, based this time on a modern Xilinx Ultrascale FPGA [111], with transceivers capable of data transmission speeds of at least 16.3 Gbps. The increased link bandwidth would allow the demonstrator to work with a time multiplexed factor of 18, which is the target in the baseline design.

Each TFP board in this design covers the same region as in the current demonstrator system (i.e. one processing octant), thus the total system would be formed by $8 \times 18 = 144$ TFP boards.

Depending on the chosen firmware and its future optimisation, each board would be equipped with a number of FPGAs, varying from 1 to 3. This number has also to take into account the required latency and cost of the final system.

4.6.1 FPGA resources

Having a time multiplexed period half of the one used in the demonstrator system, each TFP would have to process data at twice the bandwidth, in order to guarantee the same described performance.

An upper limit on the number of logic resources needed to handle with the increased rate can be made by simply doubling the resourced used in a demonstrator slice. Table 4.13 reports the resource required, compared with the available logic in three Xilinx Ultrascale FPGAs [111]. According to these numbers, at least three Kintex Ultrascale 115 FPGAs are required per each TFP.

Table 4.13: Upper limit on the required FPGA resources for a Track Finder Processor board running at 480 MHz. Also the available resources for a number of Xilinx Ultrascale FPGAs [111] are shown.

	LUTs [10^3]	DSPs	FFs [10^3]	BRAM (36 Kb)
SF+LR TFP Total (excl. infra)	1754	17108	2060	5688
Kintex Ultrascale 115	633	5520	1266	2160
Virtex Ultrascale+ 9P	1182	6840	2364	2160
Virtex Ultrascale+ 11P	1296	9216	2592	2016

Nevertheless, as shown in Section 3.4.5, the new Outer Tracker tilted geometry reduces considerably the number of produced stubs, input to each TFP and, therefore, the number of reconstructed tracks out of the HT drops by about 30%.

This leads to a reduction on the number of required **SF+LR** modules per **TFP** from 48 to 34. Moreover, following the example of the **GP** firmware, the overall chain is aimed to run at a higher processing firmware. Therefore, by targeting a clock speed of 480 MHz for most parts of the system, it should be possible to double the processing bandwidth of the design and halve the resources required per **TFP**, with respect to Table 4.13.

In conclusion, a final Track Finder Processor board is expected to be constructed with no more than two Kintex Ultrascale 115 FPGAs or with one Virtex Ultrascale+ 11P.

4.6.2 Latency

The use of 16.3 Gbps links to communicate with the **TFP** boards will reduce the latency required to accumulate data in all the different blocks of the chain. In particular, the data accumulation period in the **HT** and the Seed Filter latency can be both reduced by 450 ns.

Table 4.14: Latency table for the baseline system, extrapolated from the existing demonstrator. Note that extrapolation to 480 MHz assumes that additional registers are necessary, meaning that rather than a 50% reduction in latency, a 30% reduction is assumed throughout. This number is based on what was achieved with the **GP** router at 480 MHz. Two SERDES stages internal to the **TFP** are also assumed, to cover the worst case scenario (for latency) of three daisy-chained FPGAs.

Latency [ns]	240 MHz	480 MHz
DTC	250	250
DTC → TFP (SERDES & fibre)	150	150
GP	251	176
HT	575	403
SF+LR+DR	425	425
Two internal SERDES stages	240	240
TFP → L1 (SERDES & fibre)	150	150
TFP First out → Last Out	875	613
Total: First in DTC → First in L1	2041	1794
Total: First in DTC → Last in L1	2916	2407

Table 4.14 reports a conservative estimation of the latency of a final system. Here the latency required for the **DTC** boards to unpack, format and sort stubs has been

estimated to be 250 ns. An additional latency of 150 ns is assumed to transfer data from the [DTCs](#) to the [TFP](#) boards.

Also estimations for a system running at 480 MHz are shown. A latency limit, intended as the time difference between the first stub coming in and first track coming out, can be estimated as at 1.8 μ s.

4.7 SUMMARY

The assembled hardware demonstrator has successfully demonstrated the feasibility to reconstruct tracks at the [L1](#) trigger stage of the Phase-II [CMS](#) experiment at the [HL-LHC](#) conditions, within the required latency budget.

The Track Finder Processor functionalities have been implemented into five MP7 cards, installed in a microTCA crate located at CERN. Both track fitting approaches have been successfully implemented in firmware and validated against the software predictions. Good matching between hardware and emulation has been observed, with the demonstrator capable to reconstruct correctly more than 94.0% of tracks with $p_T > 3$ GeV in $t\bar{t}$ events at a [PU](#) of 200.

Both [KF](#) and [SF+LR](#) track finding chain process events with a fixed latency, which is well below the 4 μ s requirement. In particular the [SF+LR](#) has been found to be the fastest, having a first-in-last-out latency of about 3.57 μ s.

The resource usage is similar in both cases. The entire track finding algorithm could fit potentially in just two Kintex Ultrascale 115 [FPGAs](#). A new demonstrator system is planned to be built in the near future, exploiting larger [FPGAs](#) and faster optical links.

PRIMARY VERTEX FINDING AT L₁ TRIGGER LEVEL

The availability of central track information at the Level-1 (L₁) trigger level will constitute a new powerful tool, necessary to form the correct trigger decision. At the L₁ Correlator Trigger (CT) track data will be combined with information from muon and calorimeter trigger systems. This will enable [46]:

- improvements in the identification of charged leptons (electrons, muons and taus);
- improvements in the measurement of the transverse momenta of lepton candidates, particularly muons;
- the determination of the isolation factor for leptons and photons with respect to other tracks.

In addition to being used in conjunction with the other sub-systems, the Track Trigger primitives can also be used in standalone mode. In particular, one of the most interesting applications is the use of reconstructed L₁ Tracks to determine the primary vertices in an event. The correct identification of the hard interaction primary vertex in the collision will permit the implementation of a new set of trigger algorithms in the online Global Trigger (GT), similar to those currently used for offline reconstruction and at the High-Level Trigger (HLT), leading to better event reconstruction.

Primary vertex information can, in fact, be used to improve the event jet reconstruction or the pileup subtraction. It will also provide additional event-specific parameters, like a tracker-based missing transverse energy, which can be exploited in several trigger algorithms.

Preliminary studies were conducted in the past to estimate the achievable event vertexing performance. Those studies have been based on the use of a simple fast vertex reconstruction algorithm and have shown promising results. Benefiting from more accurate L₁ track reconstruction simulation, the potential of an L₁ Vertex Finder at the CT has been further explored and it is reported in this chapter.

Several clustering algorithms have been investigated, to identify the best option for a final implementation. The chosen algorithm should provide excellent vertex reconstruction performance and be fast, taking into account that it must be designed to target a future FPGA implementation.

5.1 SIMULATED DATA AND DEFINITIONS

Each of the vertex reconstruction algorithms presented make use of L1 Tracks produced with the standard Time Multiplexed Track Trigger (TMTT) track finding simulation [99]. Track candidates are firstly identified in the $r\phi$ plane using a Hough Transform technique and then filtered and fitted by means of the Seed Filter plus Linear Regression fitter.

Only tracks that satisfy some minimal quality requirements are used in this analysis. They must have a track fit $\chi^2 < 50$ and, in order to reduce effects from high- p_T fake tracks, L1 tracks with a $p_T > 100$ GeV are taken into consideration only if they have stubs assigned in at least six tracking stations.

The primary vertex finding efficiency is defined as:

$$\epsilon = \frac{N_{L1Vx}}{N_{GenVx}}, \quad (56)$$

where N_{GenVx} is the total number of generated primary vertices from hard interactions, which contain at least two charged tracks that can be reconstructed by the track finding algorithm (see Table 2.2). N_{L1Vx} is the total number of reconstructed vertices, where a vertex is considered correctly reconstructed if it is within a distance δ from the generated hard interaction vertex along the beam axis z ,

$$|z_{L1} - z_{Gen}| < \delta. \quad (57)$$

Most of the examined algorithms reconstruct usually more than one vertex. The hard interaction primary vertex is therefore identified as the vertex having the largest scalar sum of track transverse momenta.

The vertex reconstruction performance has been tested using simulated data sets with both the old flat and the new tilted barrel geometries. Top pair production events at a PU of 200 have been utilised as the reference, while further tests have been conducted also in other samples, as will be shown in Section 5.3.6.

5.2 VERTEX RECONSTRUCTION ALGORITHMS

Several different kinds of clustering algorithms have been tried and compared to find an optimal solution. All the algorithms have been designed in CMSSW simulation and designed to find vertices only along the beam axis z .

The algorithms can be classified into four categories [112]:

- **Grid-based methods (FastHisto).** Tracks fill a histogram and clusters are identified by looking at local minima;
- **Agglomerative Hierarchical Algorithms (Single-link, Complete-link, Average-link).** Each track starts in its own cluster, and pairs of clusters are merged at each step of the algorithm;
- **Divisive Hierarchical Algorithms (PVR).** At the beginning all tracks are part of the same cluster, which is split at successive iterations;
- **Density based clustering (DBSCAN).** Clusters are first formed in regions of high density of tracks with momenta above a defined threshold. A second step merges into the clusters all tracks within a chosen distance;
- **Partitioning Methods (Kmeans).** Parameter space is initially partitioned in a fixed number of clusters. Tracks are relocated between clusters at each iteration.

The following paragraphs will illustrate the operation of the algorithms, each of which has been tested using a simulated data set of $t\bar{t}$ events with PU200. In this context, a found vertex is defined as correctly reconstructed if it is within 1 mm of the generated vertex.

5.2.1 *FastHisto algorithm*

The *FastHisto* clustering algorithm was first presented in the Technical Proposal of the CMS Phase-2 Upgrade [46, 113]. The basic idea is to discretise the z axis in 1 mm-wide bins. Then a histogram is filled with the track information. A track is inserted in a particular bin n if,

$$|z_n - z_{0,\text{trk}}| < \varepsilon, \quad (58)$$

where z_n is the position of the centre of the bin, $z_{0,\text{trk}}$ is the longitudinal impact parameter of the track, defined as the distance of the track Δz from the CMS axis origin $(0,0,0)$, and ε represents the assumed maximum width of a cluster. Once all tracks have been processed, the hard interaction primary vertex is identified by looking at the bin with the highest value of $\sum_{\text{trk}} p_T^{\text{trk}}$.

Figure 5.1a shows the efficiency of reconstructing the hard interaction primary vertex as a function of the z coordinate of the generated vertex. A value of $\varepsilon = 0.15$ cm has been found to give the highest efficiency (about 86.4%). This is also confirmed by looking at reconstructed vertex z resolution distribution (Figure 5.1b), where an average resolution of 0.046 cm can be achieved.

The *FastHisto* algorithm does not require any pre-sorting of the tracks and therefore constitutes a good candidate for a possible FPGA implementation. The number of iterations required by the algorithm to find the main event primary vertex is estimated to be $(n_{\text{bins}}/2 + n_{\text{trk}})$.

5.2.2 *Agglomerative Hierarchical Algorithm*

Agglomerative hierarchical clustering algorithms construct clusters by merging observables in a bottom-up fashion. Initially, each object forms a single cluster. At each iteration, clusters are then merged until a certain criterion is satisfied.

This method has been applied to *L1Tracks* in order to find the primary vertices in the interaction. At the first iteration, each track form a single vertex. At any

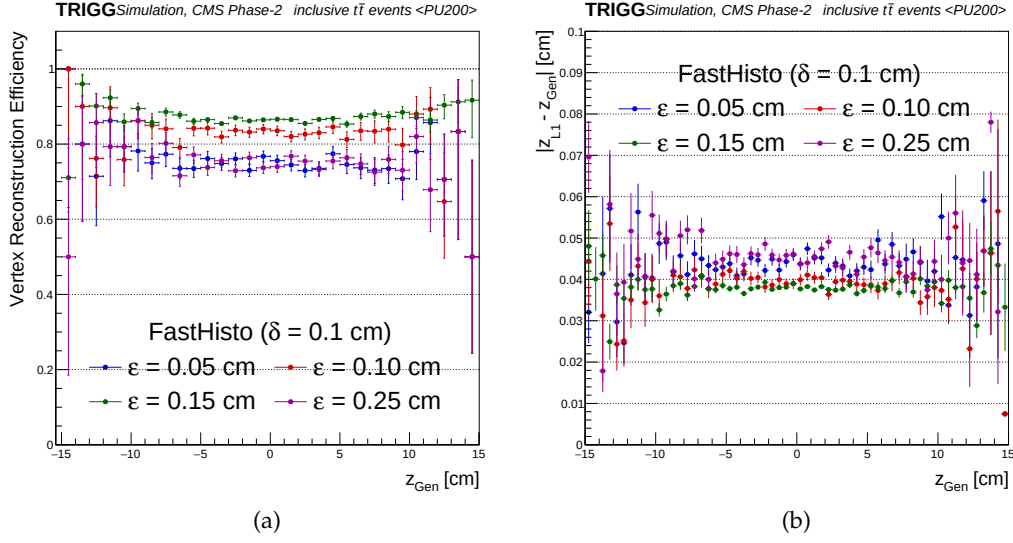


Figure 5.1: Vertex reconstruction performance with a FastHisto clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed vertex width d . [46, 113]

following iteration, two vertices $\mathcal{A}$ and $\mathcal{B}$ are merged into a single one if their distance is less than a predefined parameter ϵ ,

$$D(\mathcal{A}, \mathcal{B}) < \epsilon. \quad (59)$$

The distance between two clusters can be computed in different ways, which sub-divide further the agglomerative hierarchical clustering algorithms as follow.

- **Single-link clustering.** Distance is defined as the shortest distance from any member of the first cluster $\mathcal{A}$ to any member of the second cluster $\mathcal{B}$. [114]

$$D(\mathcal{A}, \mathcal{B}) = \min\{d(x, y) : x \in \mathcal{A}, y \in \mathcal{B}\}; \quad (60)$$

- **Complete-link clustering.** Distance is defined as the longest distance from any member of the first cluster $\mathcal{A}$ to any member of the second cluster $\mathcal{B}$. [115]

$$D(\mathcal{A}, \mathcal{B}) = \max\{d(x, y) : x \in \mathcal{A}, y \in \mathcal{B}\}; \quad (61)$$

- **Average-link clustering.** Distance is defined as the average distance between any member of the first cluster $\mathcal{A}$ to any member of the second cluster $\mathcal{B}$. [116]

$$D(\mathcal{A}, \mathcal{B}) = \frac{1}{n_{\mathcal{A}} \cdot n_{\mathcal{B}}} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y), \quad (62)$$

where $n_{\mathcal{A}, \mathcal{B}}$ are the sizes of the clusters and $d(x, y)$ is the distance between the two considered members x and y ;

Each of the metrics has been tested independently, with the goal of finding the best gap distance parameter for each of them.

SINGLE-LINK CLUSTERING. Figure 5.2a shows the primary event reconstruction efficiency, using a single-link clustering method, for different values of gap parameter, as a function of the generated vertex position z_{Gen} . The highest efficiency has been obtained with a gap value of 0.1 cm, confirmed also by the vertex resolution distribution (Figure 5.2b).

The main disadvantage of the single-link clustering method is represented by the so-called “chaining effect”, which happens when two independent clusters are wrongly merged because of the presence of uncorrelated points between them. The algorithm has been found to work well in most cases, with an average efficiency of 87%. However, the drawback for its simplicity is represented by its long processing time. The simplest implementation of the algorithm requires, in fact, $\mathcal{O}(n^3)$ iterations to converge and it is not linear with the number of clusters [112]. A faster version of the algorithm has been also developed, named SLINK, which reduces the number of required iteration to $\mathcal{O}(n^2)$ [117].

COMPLETE-LINK CLUSTERING. The complete-link clustering method has been designed in order to avoid the chaining effect. In fact, merging clusters according to the longest distance between their members leads to the creation of more compact clusters than the single-link method (see Figure 5.3).

Figure 5.4a shows the vertex reconstruction efficiency as a function of the generated vertex z coordinate, for different gap values. The best performance has been obtained for a gap value of 0.65 cm, leading to an average efficiency of 81% and a vertex resolution of 0.036 cm.

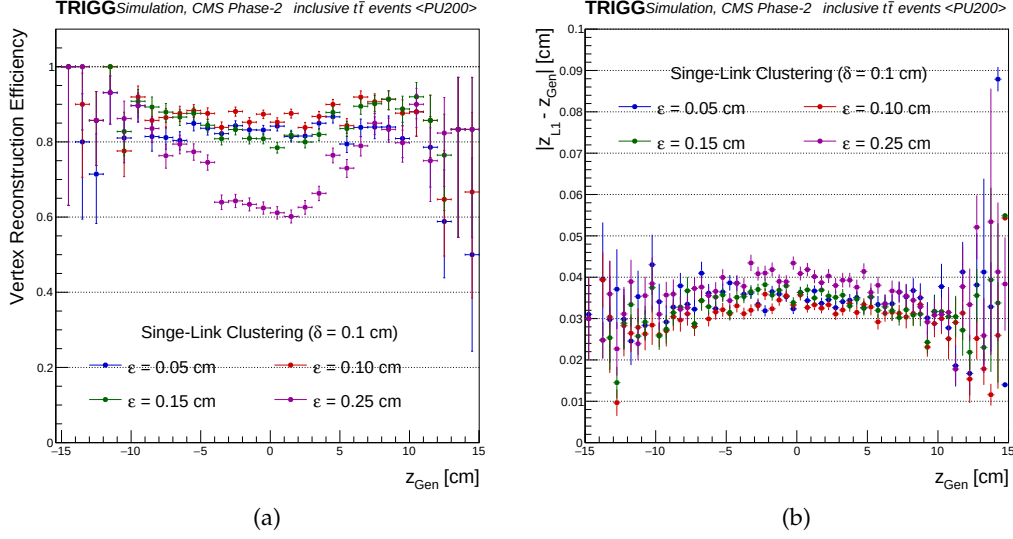


Figure 5.2: Vertex reconstruction performance with a Single-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum gap between tracks in the same cluster.

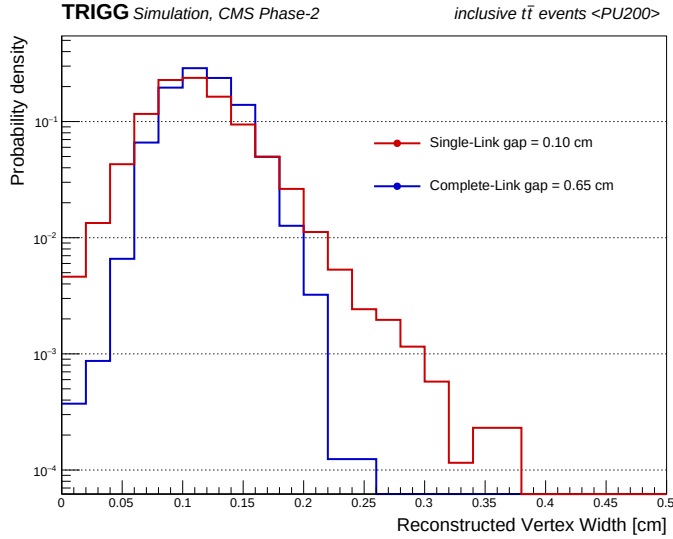


Figure 5.3: Width of track clusters produced with the Single-Link (red line) and the Complete-Link (blue line) algorithms. The Complete-Link tends to create more compact clusters, getting rid of the so-called *chaining-effect*.

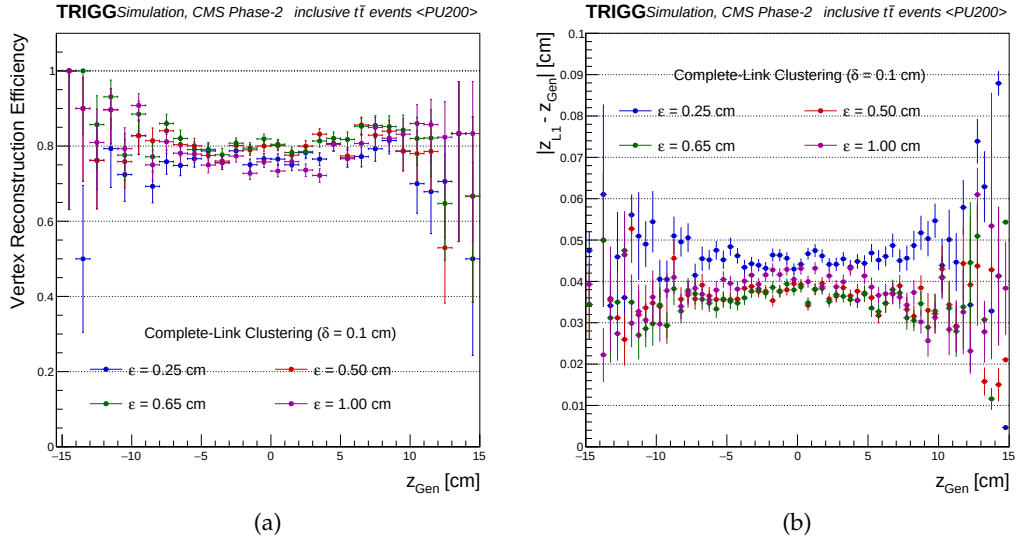


Figure 5.4: Vertex reconstruction performance with a Complete-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum gap between tracks in the two neighbour clusters.

The Complete-link method is an iterative process and, as for the Single-Link approach, requires $\mathcal{O}(n^3)$ iterations to converge. Also, in this case, a faster version of the algorithm has been designed, lowering the time complexity to $\mathcal{O}(n^2)$ [118].

AVERAGE-LINK CLUSTERING. The average-link clustering method merges in each iteration the pair of clusters whose average distance between members is lower. Figure 5.5 shows the distributions of efficiency and resolution as a function of the generated vertex z position.

The highest vertex reconstruction efficiency (89%) was obtained with a gap value of 0.35 cm, while the best resolution (0.03 cm) is achieved with a gap value of 0.25 cm, indicating that with high probability a value of 0.30 cm represents the best working point. The overall time complexity of the average-link clustering algorithm is $\mathcal{O}(n^2 \log n)$ [119].

In summary, the Agglomerative Hierarchical algorithms have shown good performance in reconstructing the hard interaction primary vertex in the event, achieving efficiencies close to 90%. Despite this, their inability to scale well with a large number of tracks would cost a huge amount of time and resources in hardware.

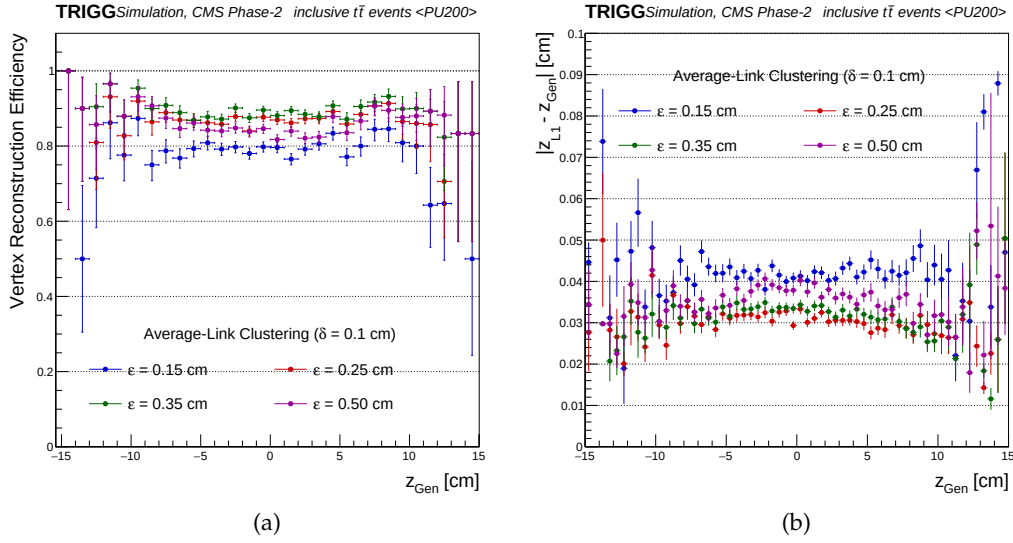


Figure 5.5: Vertex reconstruction performance with a Average-Link clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum gap between centres of two neighbour clusters.

Therefore it is possible to conclude, that they do not represent a viable solution for L_1 trigger applications, even if they constitute a good benchmark tool.

5.2.3 Divisive Hierarchical Algorithms

Another approach in the hierarchical clustering methods adopts top-bottom hierarchy. Initially, tracks are considered all to belong to the same cluster and, at each subsequent iteration, the least compatible track is rejected.

A possible implementation of this algorithm is represented by the so-called Principal Vertex Reconstructor (PVR) [120]. At the beginning, all the tracks are considered to belong to the same vertex. At each iteration, the average z coordinate of the vertex is computed and the least-compatible track (the furthest) is discarded. This procedure goes on until no track is at a distance larger than a predefined ϵ from the centre of the vertex. After that, the process is repeated with the set of rejected tracks, until it contains less than two tracks.

Figure 5.6a shows the efficiency to find the hard interaction primary vertex in the event as a function of the generated vertex z coordinate. The PVR has been found

to achieve excellent performance, with an average efficiency above 92%. Also, the good vertex resolution achieved (Figure 5.6b) demonstrates the great potential of the algorithm.

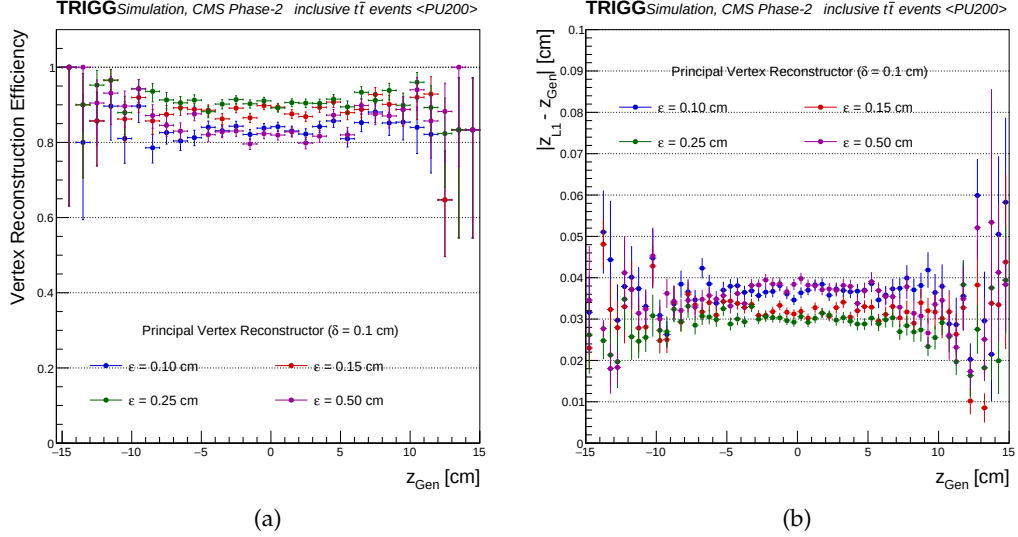


Figure 5.6: Vertex reconstruction performance with a Principal Vertex Reconstructor (PVR) clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum gap between a track and the centre of a cluster.

Nevertheless, those results are obtained in spite of significant cost in terms of processing time. In fact, it is estimated that the PVR requires at least $(n!/m!)$ iterations per cluster, where n is the number of tracks in the vertex at the first iteration and m is the number of tracks to be rejected. Such time complexity excludes the PVR from the list of algorithms suitable for a L1 Trigger application.

5.2.4 Density-based Clustering Algorithms

Density-based clustering algorithms [121] consider clusters as areas of high-density of the a selected density $p(x)$. They do not require any input assumptions about the number of clusters or on the density distribution $p(x)$.

A first density-based algorithm [122] has been designed in order to solve the chaining-effect in the single-link clustering algorithm, rejecting outliers from the

cluster by looking at their density as a function of the distance from the centre of the cluster.

One of the most famous density-based algorithms is the so-called Density-Based Spatial Clustering with Noise (DBSCAN) [123]. The DBSCAN looks in areas with high density of *interesting* points to form the clusters, where an interesting point has to satisfy certain criteria X .

The DBSCAN takes as input two parameters, the distance threshold r and the density threshold k . In addition to these, the algorithm requires a predefined criterion to select points. The density of an interesting point x_i is defined as the number of interesting points n_i that are within r from x_i . If $n_i > k$, x_i is a core point and a cluster is created. Two points are density connected if they are connected to the same core point.

The DBSCAN is a good candidate for the vertex reconstruction task. In this case, intuitively interesting tracks can be selected as that having a transverse momentum larger than a chosen threshold $p_T^{\min}$. The distance threshold is simply the maximum distance allowed between two tracks in a vertex ε , while the density threshold is interpreted as the minimum number of interesting tracks in the vertex M .

To find the optimal configuration of the DBSCAN, tests have been conducted varying one of the three parameters at a time, keeping the two other fixed. Figure 5.7 shows the vertex reconstruction performance obtained for different values of ε , requiring only $M = 2$ tracks with a $p_T^{\min} = 3$ GeV to form a vertex. The highest reconstruction efficiency and the best resolution have been observed with a $\varepsilon = 0.15$ cm.

The next step is to change the minimum required transverse momentum $p_T^{\min}$ of the M density tracks around the core track. Figure 5.8 shows the efficiency and the resolution to reconstruct the primary vertices for different values of $p_T^{\min}$, as a function of the generated vertex z position. It has been observed that the best performance is obtained by requiring at least two tracks with a minimum p_T of 4 GeV.

Finally, it is also possible to modify the requirement on the number of tracks in the cluster above a certain p_T threshold. Figure 5.9 shows the performance obtained for various M values.

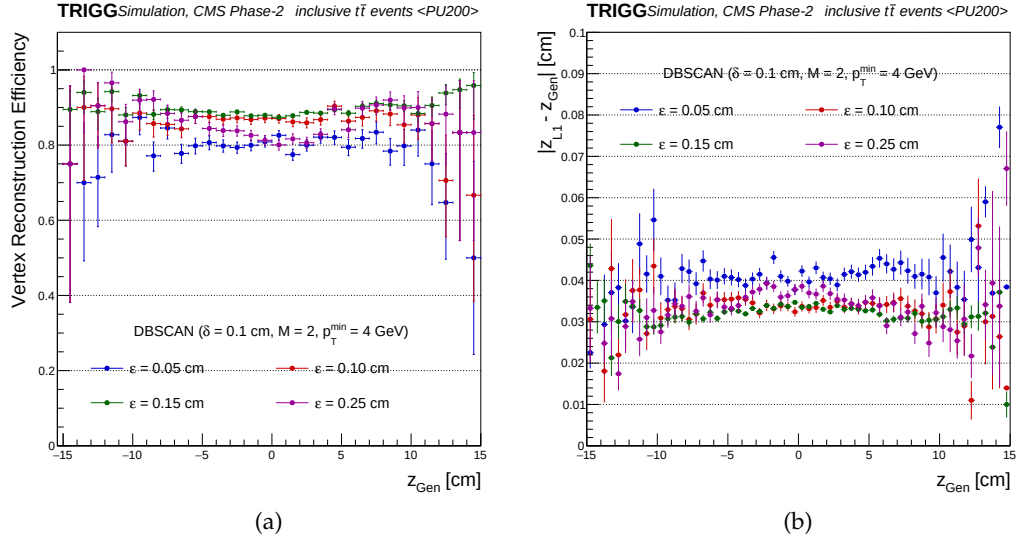


Figure 5.7: Vertex reconstruction performance with a DBSCAN clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum distance ϵ of a track from the core track in a cluster.

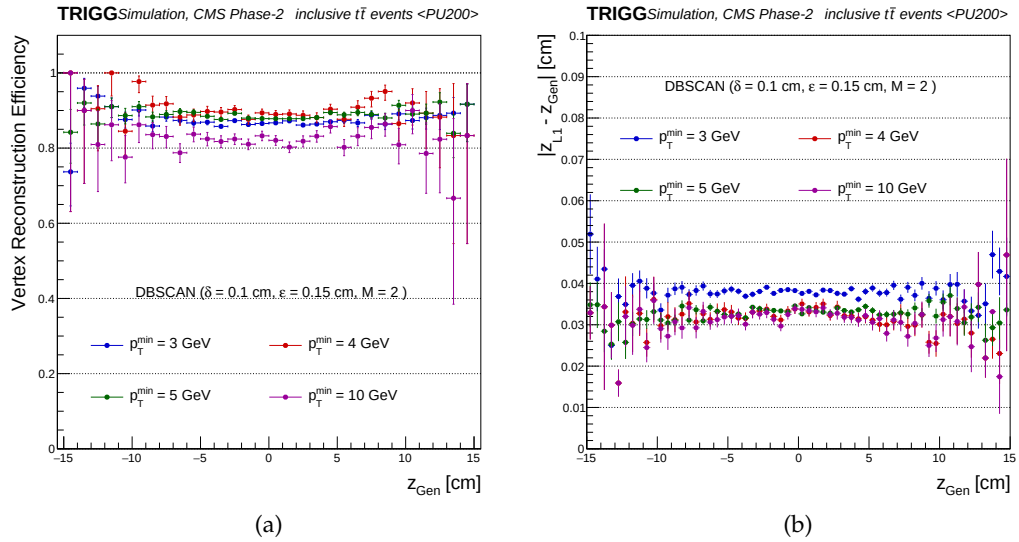


Figure 5.8: Vertex reconstruction performance with a DBSCAN clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum transverse momentum p_T^{min} of density tracks M .

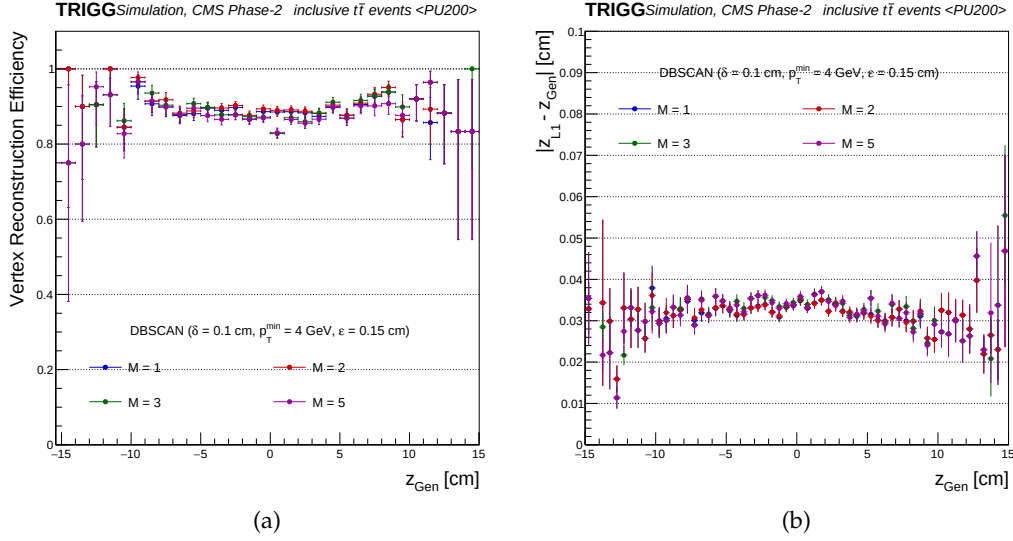


Figure 5.9: Vertex reconstruction performance with a DBSCAN clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed minimum number of density tracks M .

The time complexity of the DBSCAN algorithm is $\mathcal{O}(n^2)$ in the worst case. This can be reduced to $\mathcal{O}(n \log n)$ using an appropriate preliminary indexing [121], which for the vertexing task consists simply of sorting the tracks by increasing p_T .

5.2.5 Partitioning Clustering Methods

The *Partitioning Methods* are another popular fast clustering technique. Here parameter space is divided initially in a predefined number of clusters and, at each iteration, objects are moved from one cluster to another [112].

The simplest partitioning algorithm is the so-called *K-means* [124]. Here, the data space is partitioned into K clusters, whose centres are simply the mean of all the objects that belong to a cluster.

At first instance, the centre of a cluster $\mu_{k,0}$ is simply defined by dividing the z axis into K segments.

$$\mu_{k,0} = z_{\min} + \frac{z_{\max} + z_{\min}}{K} \cdot (k + 0.5), \quad (63)$$

where $z_{\max,\min} = \pm 15$ cm is fixed by the beam-spot length definition.

At each iteration i , each cluster centre is re-calculated as the mean of all the tracks z_t assigned to it,

$$\mu_{k,i} = \frac{1}{N_{k,i}} \sum_{t \in k} z_t, \quad (64)$$

where $N_{k,i}$ is the number of tracks in the cluster k at the iteration i . At each step, tracks are then assigned to the closest cluster according to their distance from the cluster centre that has just been computed. At the last iteration, a cut-off is implemented, by placing a threshold on the maximum distance ε of the tracks from the centre of the cluster.

The K-means algorithm as implemented therefore requires three parameters as input: K, ε, N_{it} . Figure 5.10 shows the efficiency and the resolution to reconstruct the hard interaction vertices for different values of ε , fixing the number of clusters to $K = 20$ and the number of iterations to $N_{it} = 999$. The best performance was obtained using $\varepsilon = 0.25$ cm, reaching a maximum efficiency of about 88%.

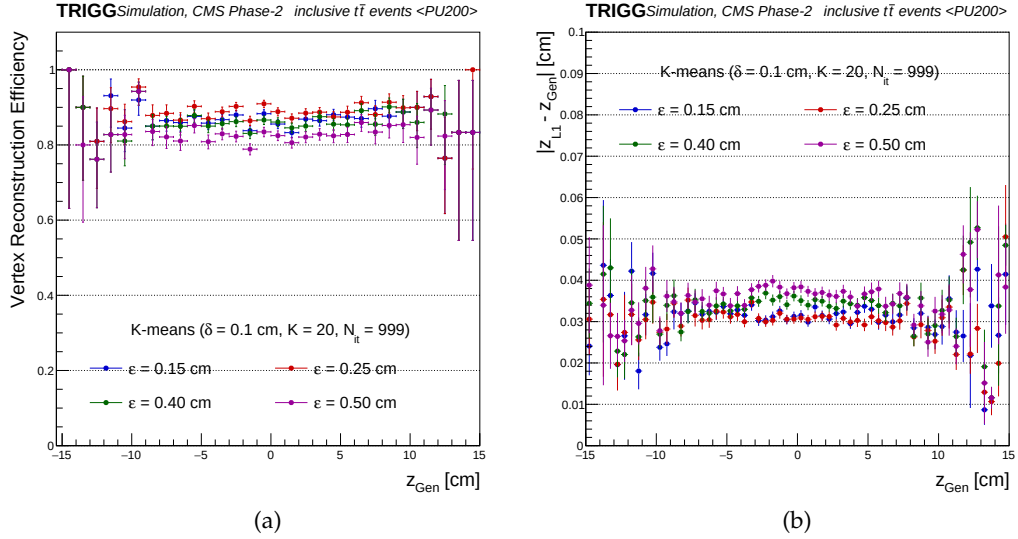


Figure 5.10: Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the assumed maximum distance of a track from the centre of the cluster ε .

The second step consists of finding the optimal number of partitions or input clusters K . Figure 5.11 shows the vertex reconstruction efficiency and the reconstructed vertex resolution for different values of K . The best performance is achieved for a $K = 18$, with an average efficiency of about 89%.

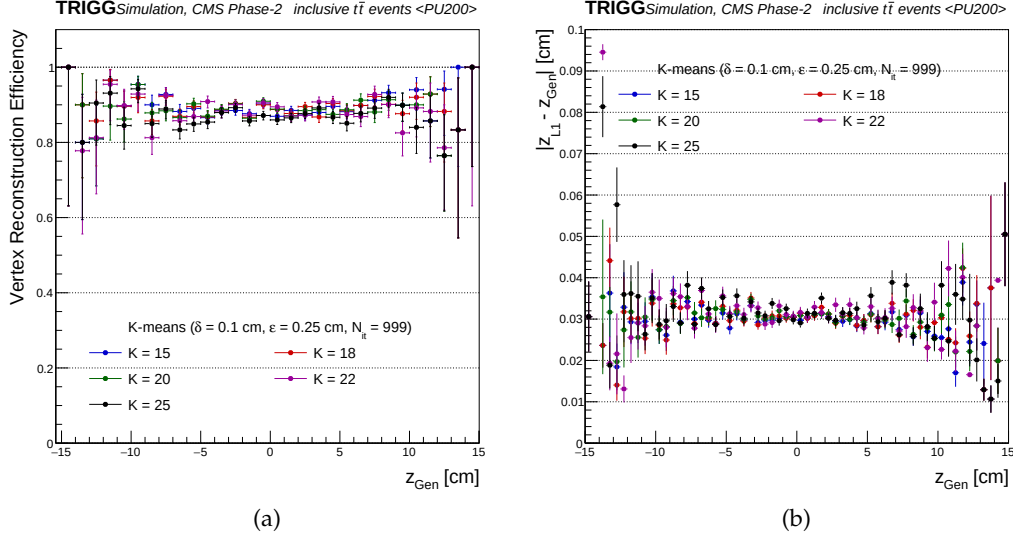


Figure 5.11: Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the pre-defined number of partitions K .

The last configurable parameter to determine is the number of iterations N_{it} required by the algorithm in order to converge to a solution. The results obtained are presented in Figure 5.12 and show that performance becomes constant after 15 iterations.

The time complexity of the K-means algorithm [125] is $\mathcal{O}(N_{it} \cdot K \cdot n)$, which can be reduced to $\mathcal{O}(N_{it} \cdot n_{\text{max}})$ by parallelising the process in K instances, where n_{max} is the maximum number of track in a cluster. The simplicity and the relatively low latency of the K-means clustering method, together with its good performance, makes it one of the best candidates for L1 Trigger applications.

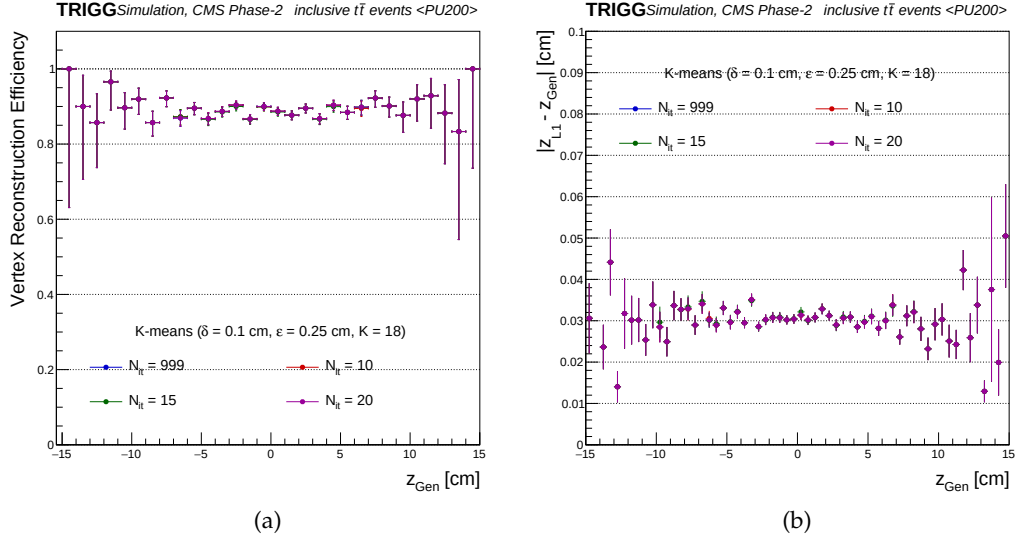


Figure 5.12: Vertex reconstruction performance with a K-means clustering algorithm. (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a PU of 200, for different values of the total number of iterations N_{it} .

5.2.6 Algorithms summary and conclusions

Figure 5.13 compares the results obtained with the tested algorithms. The average performance is summarised in Table 5.1. The PVR method shows the highest vertex reconstruction efficiency and the best vertex resolution. Nevertheless, its great time complexity excludes it from the list of algorithms that are suitable for a FPGA implementation.

The most promising clustering techniques in this respect are the DBSCAN and the K-means algorithms. Indeed, both of them have been already implemented in FPGAs for different applications [126, 127].

At a pileup of 200, the average number of tracks produced is of the order of 100. Therefore, DBSCAN is expected to be fastest, requiring $\mathcal{O}(500)$ iterations, compared with $\mathcal{O}(1,000)$ required by the K-means. In conclusion, DBSCAN has been chosen to be the baseline algorithm to characterise the vertex reconstruction performance at L1 trigger level.

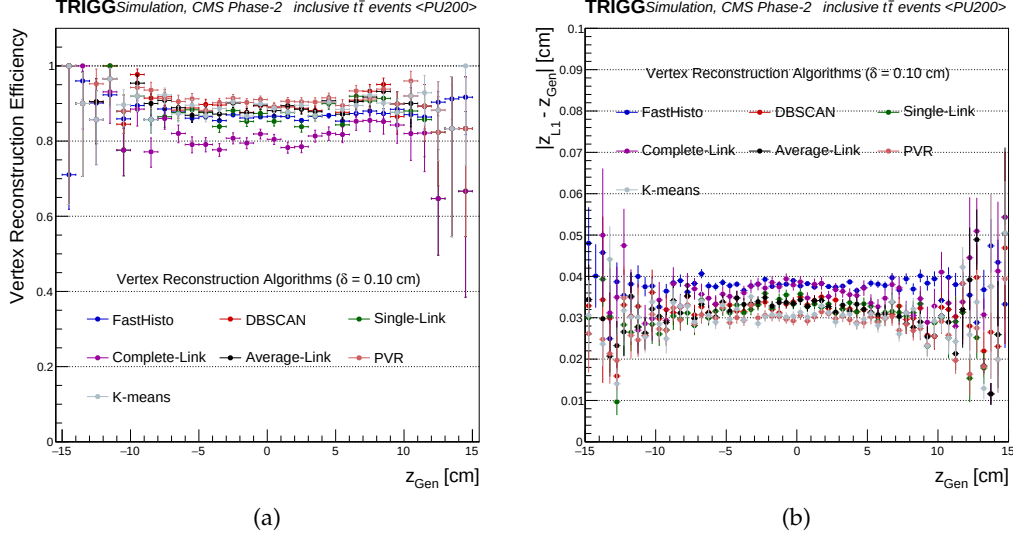


Figure 5.13: (a) Vertex reconstruction efficiency and (b) primary vertex resolution as a function of the generated primary vertex z_{Gen} position, in top pair production events at a **PU** of 200. The results obtained with best configuration of each of the investigated algorithms are compared.

Table 5.1: Vertex reconstruction performance in $t\bar{t}$ events with 200 pileup, for the several tested clustering algorithms. Also the time complexity of each algorithm is reported, as well as the average number of reconstructed vertices (RecoVertices) in the event.

Algorithm	Efficiency [%]	Resolution [cm]	Time Complexity	# RecoVertices
FastHisto	86.4	0.038	$\mathcal{O}(n_{\text{bins}}/2 + n_{\text{trk}})$	1
Single-Link	86.7	0.033	$\mathcal{O}(n^2)$	13
Complete-Link	80.6	0.036	$\mathcal{O}(n^2)$	14
Average-Link	88.8	0.033	$\mathcal{O}(n^2)$	14
PVR	90.5	0.030	$\mathcal{O}(n!/m!)$	15
DBSCAN	89.3	0.032	$\mathcal{O}(n \log n)$	7
K-means	88.7	0.031	$\mathcal{O}(n_{\text{max}} \cdot N_{\text{it}})$	K = 18

5.3 VERTEX RECONSTRUCTION PERFORMANCE WITH DBSCAN

The following section will show the achieved performance with an optimised DBSCAN clustering algorithm to reconstruct the primary vertex from the hard interaction in the event. The performance will be evaluated with respect to the longitudinal impact parameter, the number of tracks associated with the vertex, the number of pileup collisions and the total transverse momentum carried by the charged tracks in the vertex.

Identifying the primary vertex also allows the computation of some its characteristic parameters, i.e. the TrkMET or the sum of the tracks' transverse momenta. These parameters, if found to be reconstructed with an acceptable resolution, will be used by the L1 Correlator and Global Trigger.

5.3.1 Primary vertex reconstruction efficiency

The efficiency to reconstruct the hard interaction vertex in the collision is defined according to eq. 56. In top pair production events, DBSCAN will always reconstruct at least one primary vertex, considering the high multiplicity of tracks that can pass the track finding stage (on average 80 tracks per event are constructed at a pileup of 200, see Figure 3.29). However, the reconstructed vertex might be few centimetres away from the "right" one. At this stage, the only criterion to define a vertex as correctly reconstructed is its distance from the generated vertex, as specified in eq. 57.

Figure 5.14 shows the vertex reconstruction efficiency as a function of the simulated vertex longitudinal impact parameter z_{Gen} and of the number of reconstructable tracks (see Table 2.2) in the vertex. Very good vertex reconstruction efficiency is observed, with about 89.5% of the reconstructed vertices within 1 mm from the simulated vertex and another 8% being within 2.5 mm.

As expected, the efficiency does not present any strong dependency on z_{Gen} , showing a flat distribution. On the other hand, it is clear how the algorithm performs much better with samples containing a high-multiplicity of tracks. The efficiency is lower where fewer tracks from the signal event are available and it increases exponentially till it reaches a saturation plateau when the number of tracks in the simulated vertex is larger than 25.

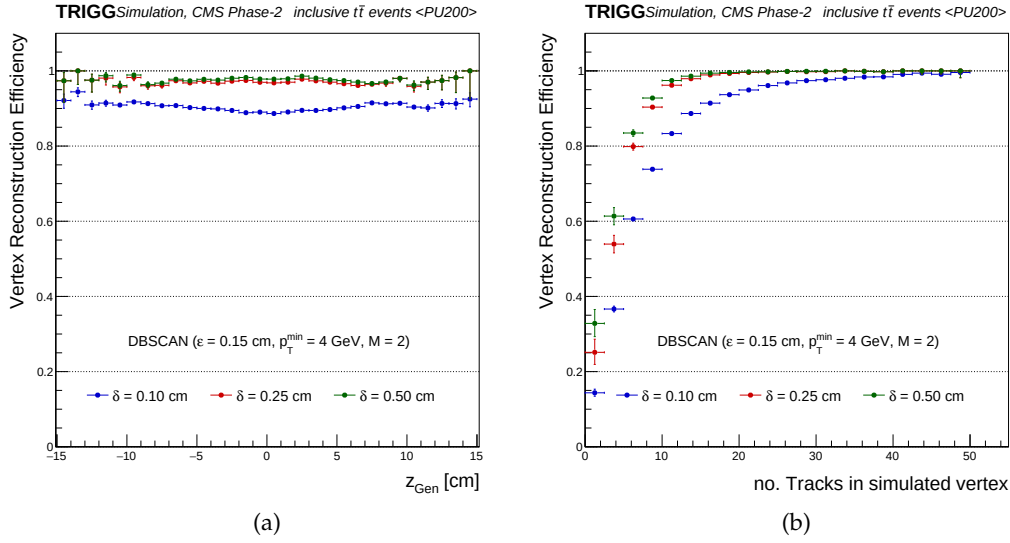


Figure 5.14: (a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events at a PU of 200. The results have been obtained with an optimised DBSCAN algorithm.

5.3.2 Primary vertex resolution

Figure 5.15a illustrates the average longitudinal impact parameter z_{L1} resolution of the reconstructed primary vertex as a function of the generated z_{Gen} , for correctly reconstructed vertices within a maximum distance δ .

Reconstructed vertices within 1 mm of the true vertices have very good resolution, being at an average distance of about 0.3 mm from the generated primary vertex. Nevertheless, including in the analysis the remaining 8% of the reconstructed vertices which are at a maximum distance of 5 mm, a worsening in the average resolution is observed of about 0.15 mm.

Figure 5.15b presents the percentage of tracks in the primary vertex, which did not originate from the hard interaction (*Fake Content*). On average 20% of the tracks assigned to the primary vertex come from pileup collisions or are fake tracks produced by the track finding processors. This fraction is higher in reconstructed vertices with the average content of fake tracks in vertices reconstructed within a distance $1 \text{ mm} < \delta < 2.5 \text{ mm}$ of about 34%. Vertices at larger distances from the truth have an even higher fake rate of about 56%. They represent only a small fraction of the reconstructed vertices (less than 1%), so that it is possible to affirm

that the DBSCAN reconstructs the hard interaction primary vertices with good resolution in the great majority of the cases.

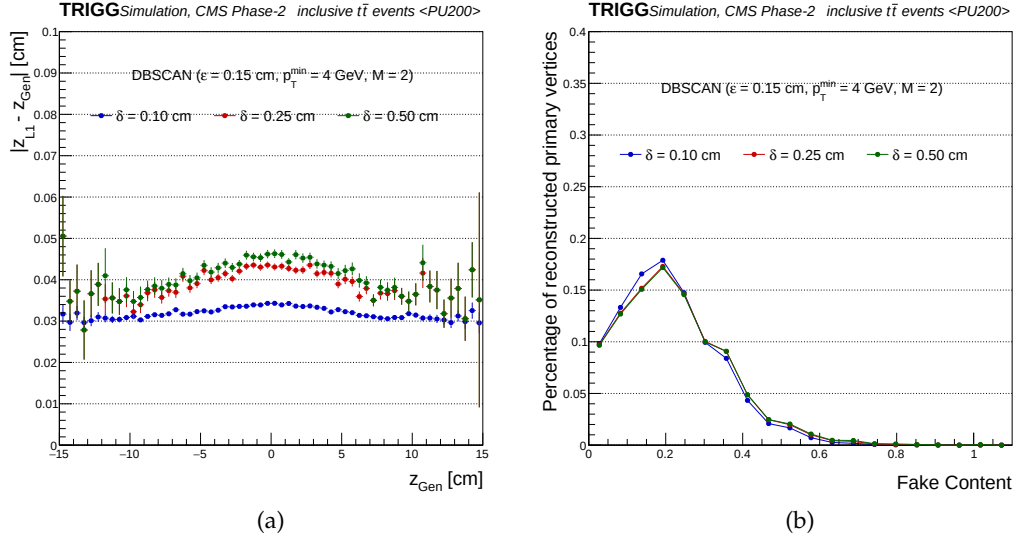


Figure 5.15: (a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events at a PU of 200. The results have been obtained with an optimised DBSCAN algorithm.

5.3.3 Pileup effects

The presence of pileup tracks carrying a relatively high transverse momentum can lead to a misidentification of the primary vertex of the hard interaction. Figure 5.16 shows the effect of the pileup in terms of vertex reconstruction efficiency. The DBSCAN shows excellent vertex reconstruction performance in the data set not contaminated by the pileup, with an average efficiency higher than 95%. The algorithm weakness when acting on low track multiplicity events is still visible even in the sample with no pileup. Although the reconstruction efficiency is higher than in the case of a higher pileup, the performance is still significantly affected by the longitudinal impact parameter resolution of the single tracks.

Effects of pileup can be also observed in the z_{Gen} parameter spectrum (Figure 5.17a). Nevertheless those are not really strong, in fact, the longitudinal impact parameter resolution at PU200 is just 0.03 cm worse than in the pileup-less case.

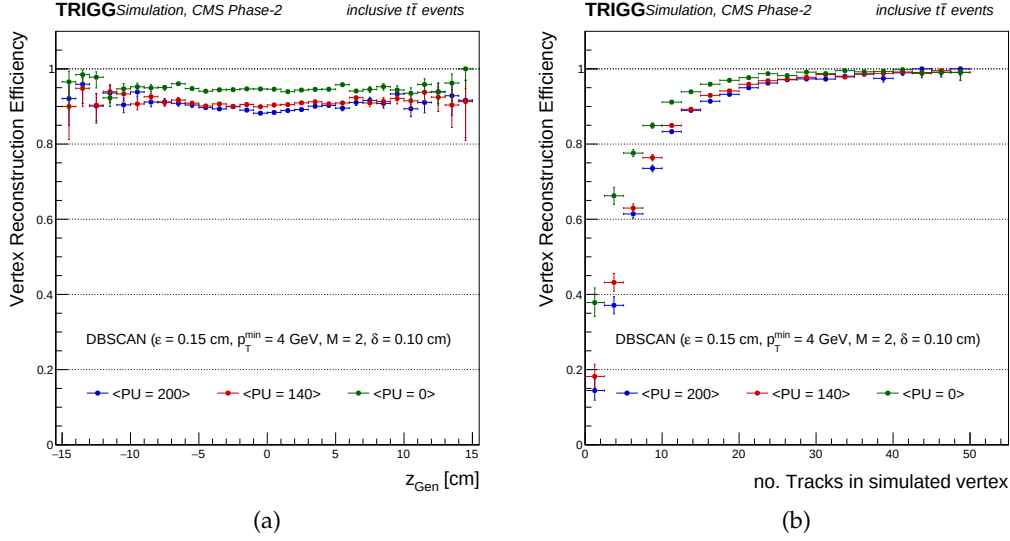


Figure 5.16: (a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events with different PU contents. The results have been obtained with an optimised DBSCAN algorithm.

This is caused by the presence of pileup and fake tracks wrongly assigned to the primary vertex, as shown in Figure 5.17b. It is interesting to point out that also in the PU0 sample, there is still a not negligible fraction of tracks in the primary vertex that is not matched with a truth track from the primary collision.

Both the track finding and the vertex reconstruction algorithms perform a good filtering of the uninteresting pileup tracks. Figure 5.18a shows the number of reconstructed vertices in $t\bar{t}$ events with different PU contents. Even in the highest case of PU200, only a small fraction of the vertices have been reconstructed ($\sim 8\%$). That is thanks to the filtering criteria that the DBSCAN applies in order to form a vertex. Indeed only few pileup interactions produce at least two tracks with a $p_T > 4 \text{ GeV}$, as required. Figure 5.18b shows the distribution of track transverse momenta, for tracks originating from the hard and the pileup interactions. Most of the pileup tracks have a very low momentum (average p_T is lower than 4 GeV),

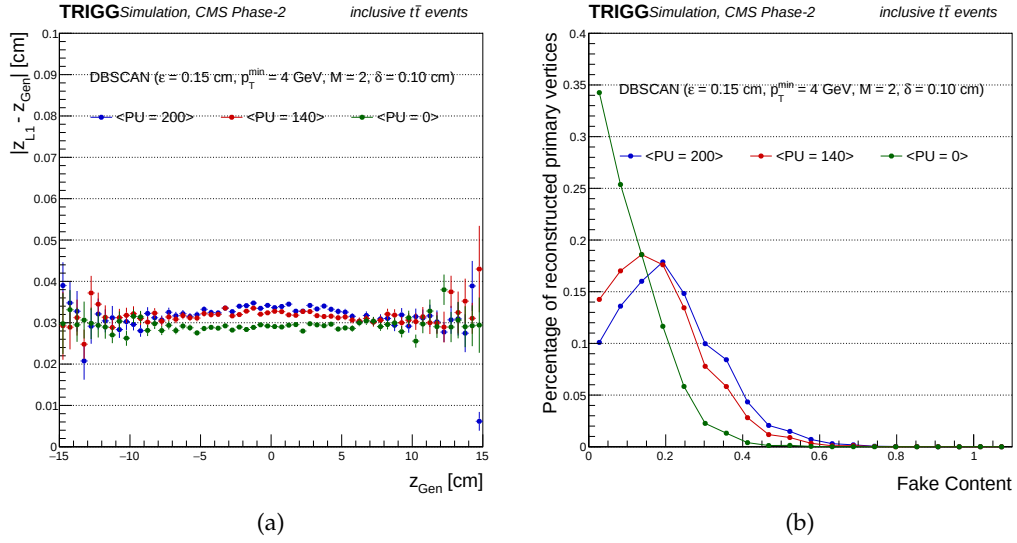


Figure 5.17: (a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events with different pileup contents. The results have been obtained with an optimised DBSCAN algorithm.

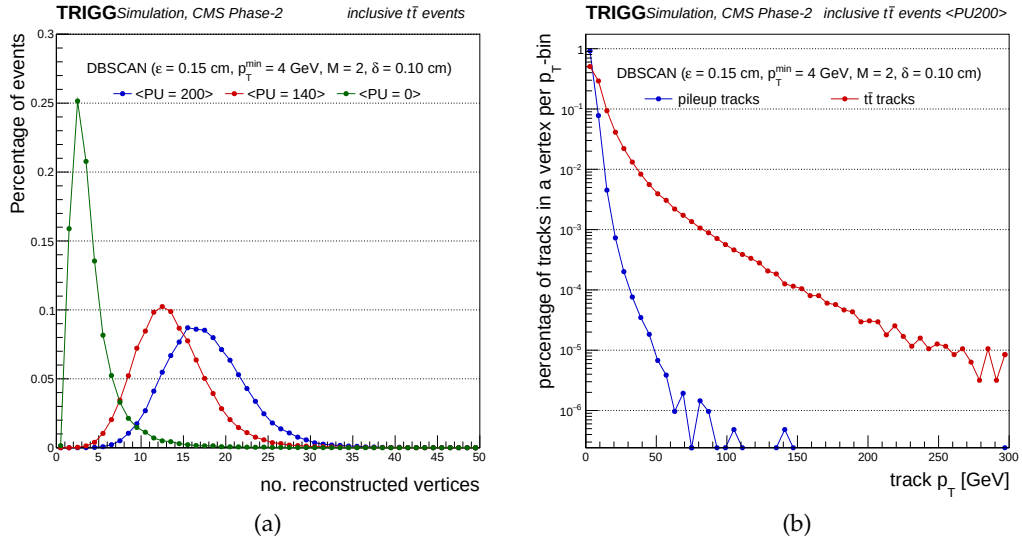


Figure 5.18: (a) Number of reconstructed primary vertices in top pair production events with different pileup contents. (b) Distribution of transverse momentum p_T for generated tracks belonging to the hard $t\bar{t}$ interaction in red or to pileup in blue. The results have been obtained with an optimised DBSCAN algorithm.

5.3.4 Vertex-based triggers

Primary vertex information can be used as a handle in several trigger menus at [L1](#). One of the explored possibilities consists of exploiting the precise measurement of the vertex z coordinate in multi-jet triggers [[113](#)].

Other possible triggers can be based on the measurements of some vertex characteristics, like the total transverse momentum carried by charged particles and the tracker-based missing transverse energy.

5.3.4.1 Total transverse momentum in the vertex

The total transverse momentum carried by charged particles associated with a vertex can be simply computed by summing the estimated p_T of all the reconstructed tracks in the vertex.

$$\sum_{\text{tracks}} |p_T^{\text{track}}|. \quad (65)$$

Figure [5.19a](#) shows the relative resolution of the vertex total transverse momentum, as a function of the generated $\sum_{\text{tracks}} p_T^{\text{track}}$. The resolution increases with increasing p_T , stabilising in a plateau at $p_T > 150$ GeV at a relative resolution of about 15%. A possible trigger, selecting signals with $p_T > 100$ GeV, can be built and its efficiency curve is displayed in Figure [5.19b](#). The curve presents a relatively slow rise, reaching the efficiency plateau at a value of generated $\sum_{\text{tracks}} p_T^{\text{track}} > 130$ GeV.

ROC curves can be therefore drawn for such a trigger. Figure [5.20](#) shows the trigger signal efficiency plotted against the background rejection for three different signal selections. In all the three considered cases an optimal cut on the reconstructed vertex total transverse momentum can be set, which assures good trigger efficiency above 90%, rejecting at the same time more than 90% of the background events.

5.3.4.2 Tracker-based missing transverse energy

The missing transverse energy or [MET](#) is an essential parameter of particle physics events. Typical analyses searching for new physics signals usually perform an

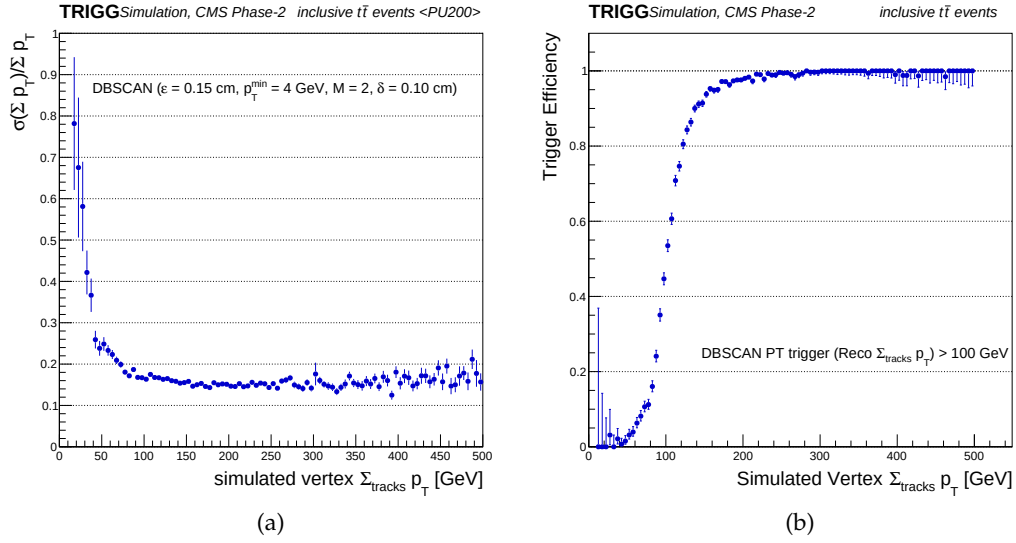


Figure 5.19: (a) Relative resolution of the total vertex transverse momentum as a function of the generated one. (b) Trigger efficiency for selecting signals with $\text{Gen } \Sigma p_T > 100 \text{ GeV}$, as a function of the cut applied on the reconstructed vertex total transverse momentum. The results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200.

event selection based on large MET [128–130]. Therefore the Phase II CMS L1 trigger has to provide a good estimation of this parameter.

The MET is generally computed using the energy information provided by the calorimeters. However, such a measurement is very sensitive to pileup effects, which can lead to poor trigger decisions. On the other hand, the availability of L1Tracks will make possible to estimate the missing transverse energy carried by the charged particles in the interaction, which is referred as Tracker-based missing transverse energy or TrkMET. Knowing which vertex tracks originated from is essential to reject pileup contributions.

Figure 5.21 shows the measured pileup contribution to the missing transverse energy using calorimeter towers (CaloMET) information and L1Tracks, for different pileup scenarios. The TrkMET has been found to be the most robust against pileup, showing an increase of about 60% when passing from PU0 to PU200. On the contrary, the CaloMET has an average increase of about a factor 2.5.

Figure 5.22a shows the relative resolution for the reconstructed TrkMET, as a function of the generated value. A plateau can be observed for $\text{GenTrkMET} >$

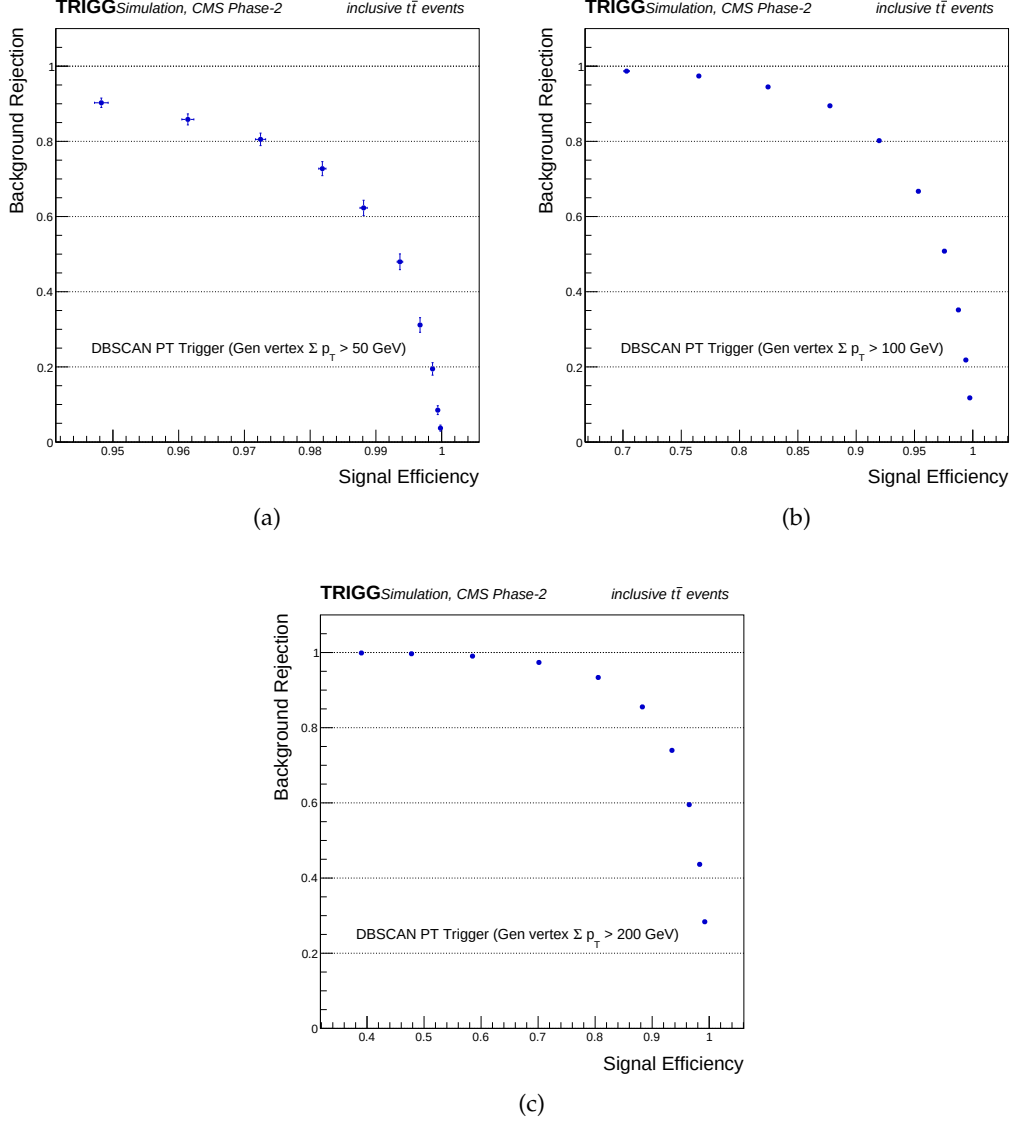


Figure 5.20: The background rejection vs. signal efficiency for an inclusive vertex transverse momentum trigger, in $t\bar{t}$ events with PU_{200} , for three different signal definitions (Gen $\Sigma p_T > 50, 100$ and 200 GeV). For clarity the origin on the x axis has been suppressed.

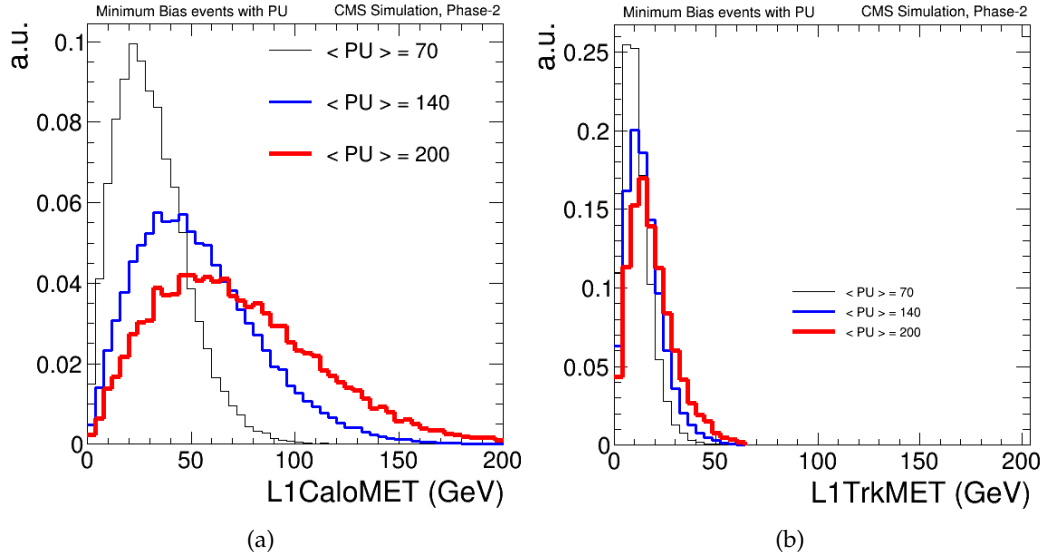


Figure 5.21: Distribution of the missing transverse momentum reconstructed from the calorimeter trigger towers (a) and from the L1 tracks (b), in minimum bias events with an average of 70, 140 or 200 pileup. [113]

50 GeV, resulting in an average relative resolution of about 30%. This poor resolution affects the performance of a possible trigger based on TrkMET.

Analogously to what has been done for the p_T -trigger, it is possible to draw efficiency curves in this case. Figure 5.22b presents the trigger efficiency with a cut on the measured TrkMET set at 50 GeV, as a function of the simulated vertex TrkMET. The poorer resolution translates into a slower efficiency rise, which reaches its plateau only after 150 GeV.

The ROC curves (Figure 5.23) highlights that it is still possible to find an optimal cut on the measured TrkMET able to guarantee acceptable trigger performance, without increasing excessively the output trigger rate because of the accepted backgrounds.

5.3.5 Changing the p_T -threshold

To improve vertexing performance it is possible to change the applied cut on the L1track p_T . Three possible thresholds have been explored, using tracks with $p_T > 2, 3$ or 5 GeV.

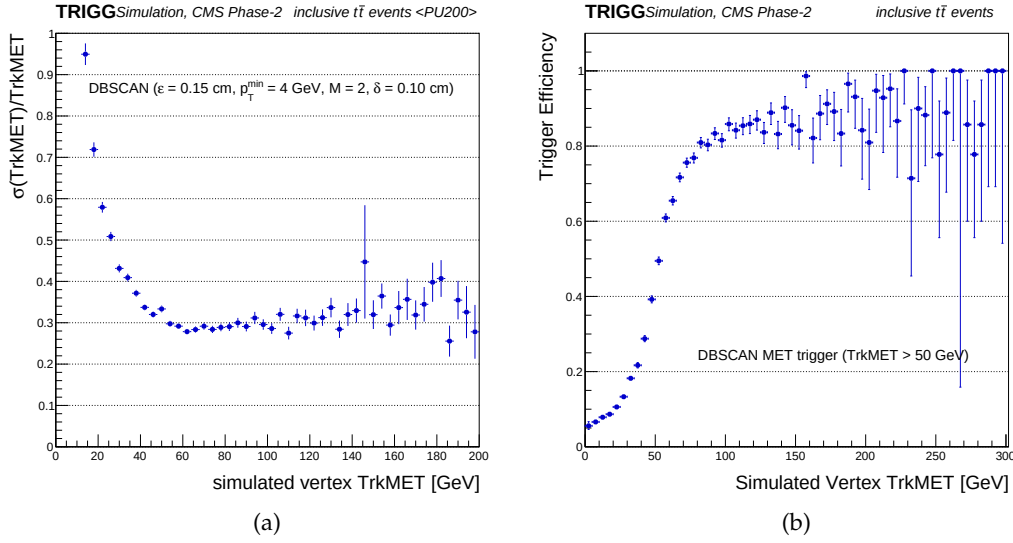


Figure 5.22: (a) Relative resolution of the tracker-based missing transverse energy as a function of the generated one. (b) Efficiency of a TrkMET-based trigger with $\text{TrkMET} > 50$ GeV, as a function of the simulated vertex TrkMET. These results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200.

The vertex reconstruction efficiency is shown in Figure 5.24 as a function of the generated vertex longitudinal impact parameter z_{Gen} and the number of tracks assigned to the vertex. The highest efficiency is obtained with a p_T threshold set to 3 GeV. A small loss of efficiency is observed applying a higher cut at 5 GeV of about 1%. On the other hand, lowering the threshold to 2 GeV, increases significantly the amount of fake and pileup tracks reconstructed in the track finding stages, which leads to a misidentification of the primary vertex and to a reconstruction efficiency about 6% lower.

Also, the longitudinal impact parameter resolution of the reconstructed vertex is affected by the inclusion of low- p_T track in the process, as visible in Figure 5.25a. On the contrary, vertices reconstructed by means of tracks with $p_T > 5$ GeV present still good resolution, about the same as in the 3 GeV-threshold case. In addition, such a cut reduces drastically the fraction of pileup tracks, leading to a lower average vertex-track fake ratio of about 11% (Figure 5.25b).

Even if including low- p_T tracks can degrade vertexing performance, their presence might be an advantage for vertex-based triggers. Figure 5.26 shows the performance of a trigger based on the sum of the momenta of the tracks assigned to

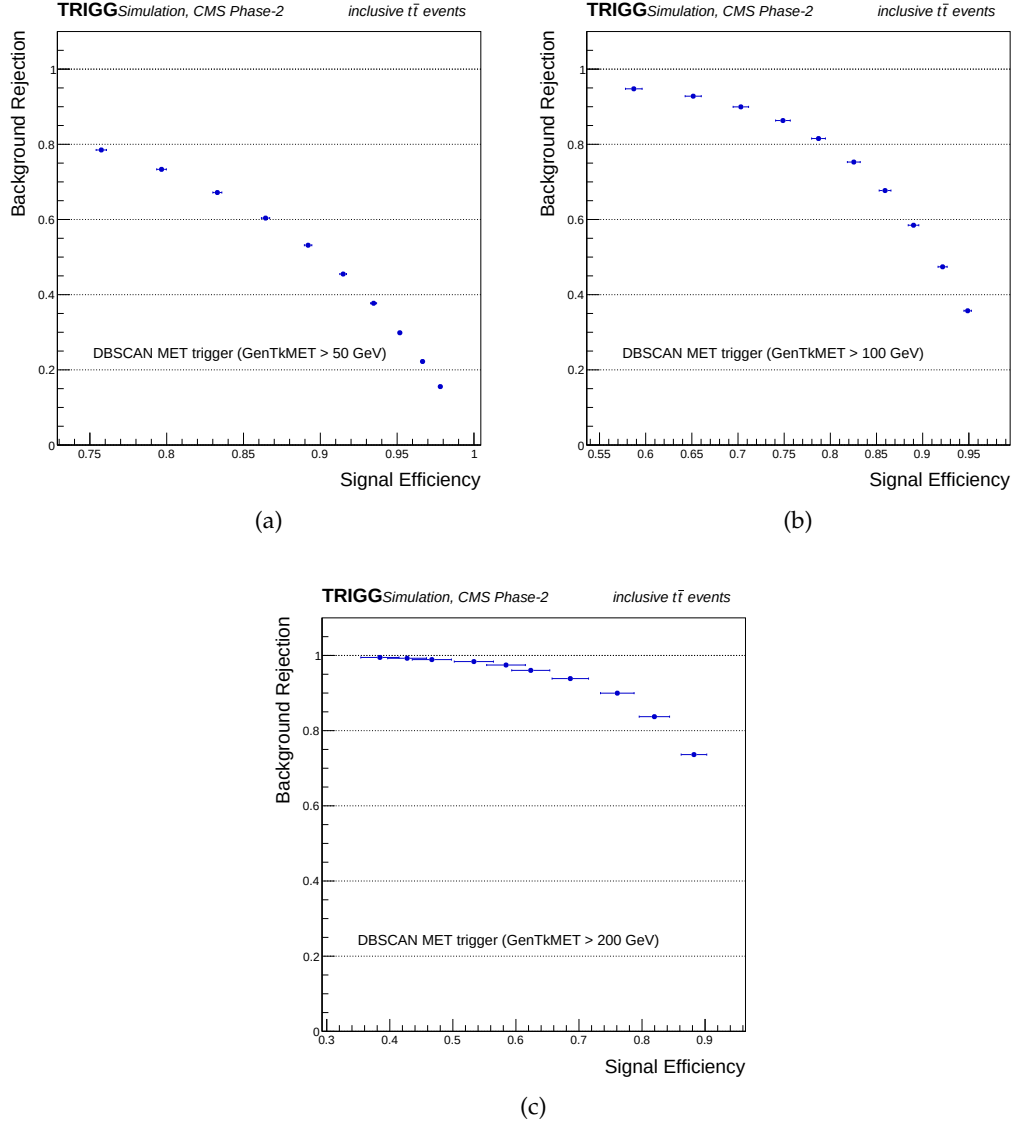


Figure 5.23: The background rejection vs. signal efficiency for an inclusive TkMET trigger, in $t\bar{t}$ events with PU_{200} , for three different signal definitions (GenTkMET > 50, 100 and 200 GeV). For clarity the origin on the x axis has been suppressed.

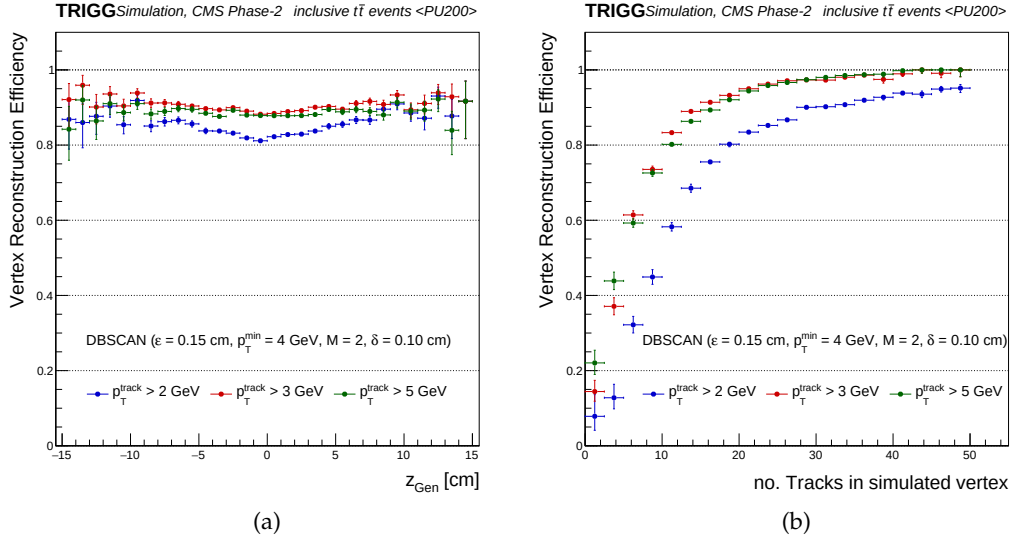


Figure 5.24: (a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.

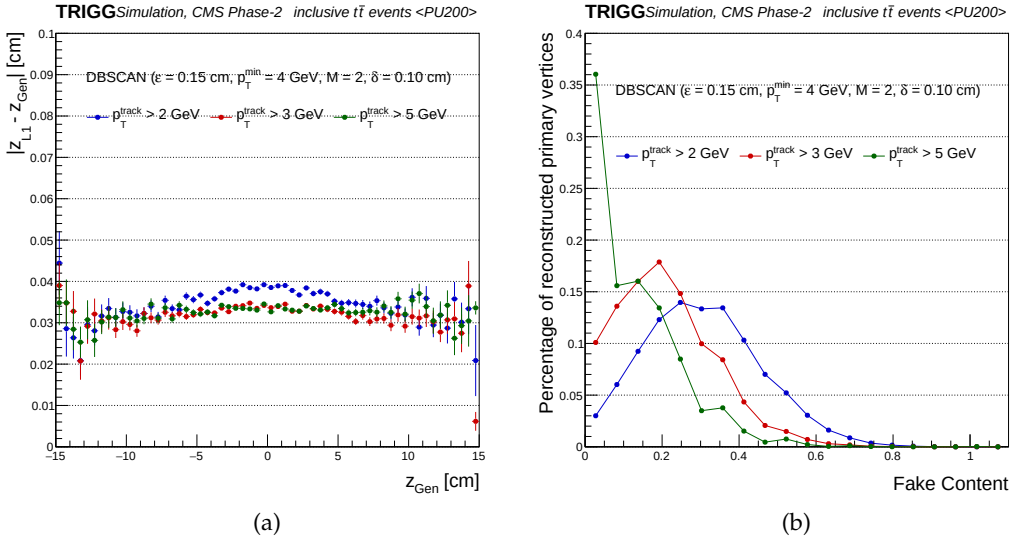


Figure 5.25: (a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.

the primary vertex. It is immediately visible how increasing the p_T -threshold to 5 GeV deteriorates the total transverse momentum relative resolution of about 10%. No remarkable improvements are instead observed using tracks with $p_T > 2$ GeV. The same conclusions can be also drawn by looking at efficiency distribution for a $\sum p_T$ -based trigger, where a slightly steeper curve is seen for a 2 GeV threshold.

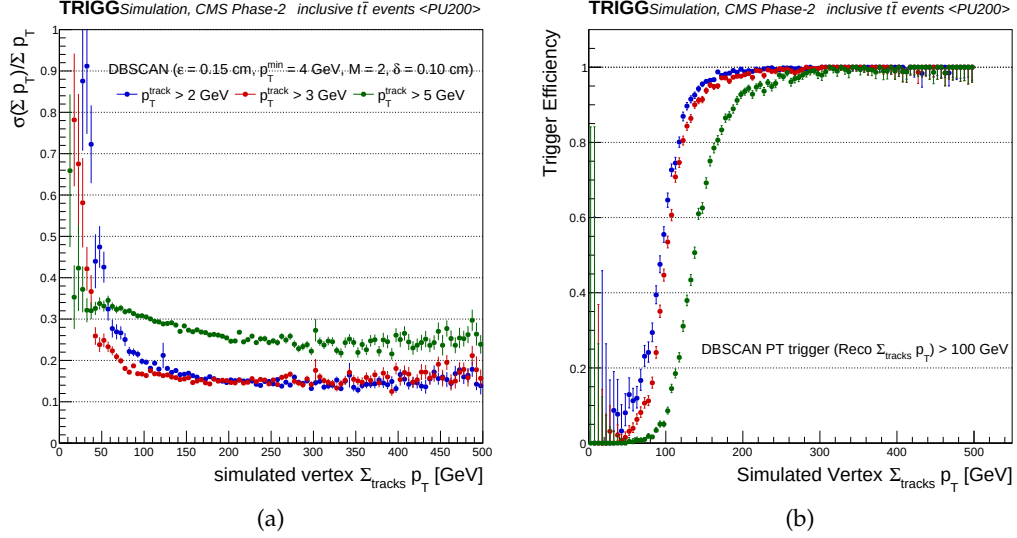


Figure 5.26: (a) Relative resolution of the total vertex transverse momentum as a function of the generated one. (b) Trigger efficiency for selecting signals with Gen $\sum p_T > 100$ GeV, as a function of the cut applied on the reconstructed vertex total transverse momentum. The results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200, using tracks with different p_T thresholds.

Changing the track p_T threshold does not seem to have any particular effect on the measurement of the TrkMET, as shown in Figure 5.27. Similar performances have been observed for the three p_T cuts in terms of TrkMET resolution and trigger efficiency.

To summarise a track p_T threshold set to 3 GeV has been found as an optimal compromise, capable to guarantee excellent vertexing performance with a relatively low fake rate.

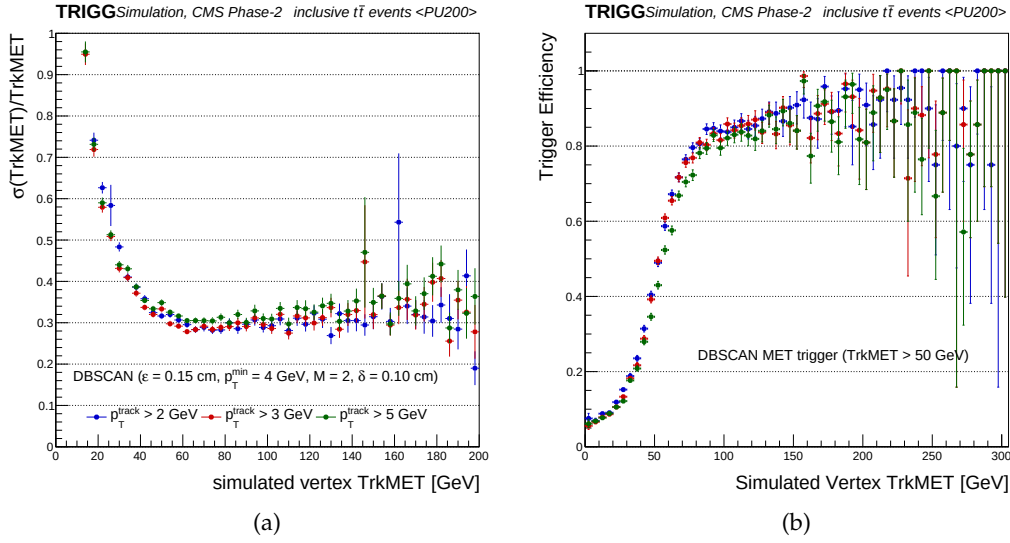


Figure 5.27: (a) Relative resolution of the tracker-based missing transverse energy as a function of the generated one. (b) Efficiency of a TrkMET-based trigger with $\text{TrkMET} > 50$ GeV, as a function of the simulated vertex TrkMET. These results have been obtained with an optimised DBSCAN algorithm in $t\bar{t}$ events with PU200, using tracks with different p_T thresholds.

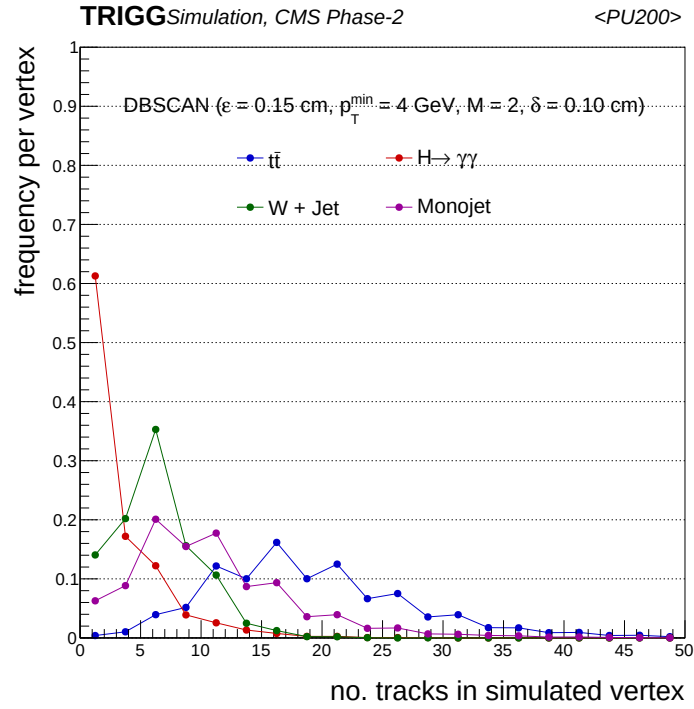
5.3.6 Vertex reconstruction in different final states

It is interesting to measure also vertex reconstruction performance in different final states than $t\bar{t}$. As already shown previously, performance is expected to be much worse for signals with low track multiplicity. Figure 5.28 presents the achieved efficiency and resolution by DBSCAN algorithm applied to:

- events where a W boson, decaying into a pair of leptons, has been produced in association with a high energetic jet with $80 < p_T < 120$ GeV. These signals constitutes an important background for new physics and top quark studies and allows tests of perturbative QCD calculations [131];
- events where a hadronic jet is produced in association with a large transverse energy imbalance (*monojet*). Those signals are predicted in several models of new physics [132] and, in particular, the used samples has been simulated in accordance with an extra dimension model with 3 extra dimensions;
- events where a standard model Higgs boson has been produced and decays in a pair of photons. This is one of the golden channels for Higgs studies

despite its small branching ratio, because of its clean final-state that allows precise measurement of the Higgs invariant mass [133]. Vertexing in such events is really hard, but it is expected to be possible in the case where the Higgs boson carries a significant momentum and recoils against a hard object.

Except that for very low track multiplicity signals, such as inclusive $H \rightarrow \gamma\gamma$, the vertex reconstruction efficiency is still acceptable, being higher than 60% to reconstruct vertices within 0.1 mm. Figure ?? shows the track multiplicity in the generated primary vertex for the considered signal events.



Relaxing the “tolerance” distance requirement, higher efficiencies can be achieved above 80%. Also, several different vertex selection algorithms could be use with different compromises between selectivity and efficiency, and used in different parts of the trigger menu. This permits to compute the missing transverse energy also for other signals and implement consequently a TrkMET trigger.

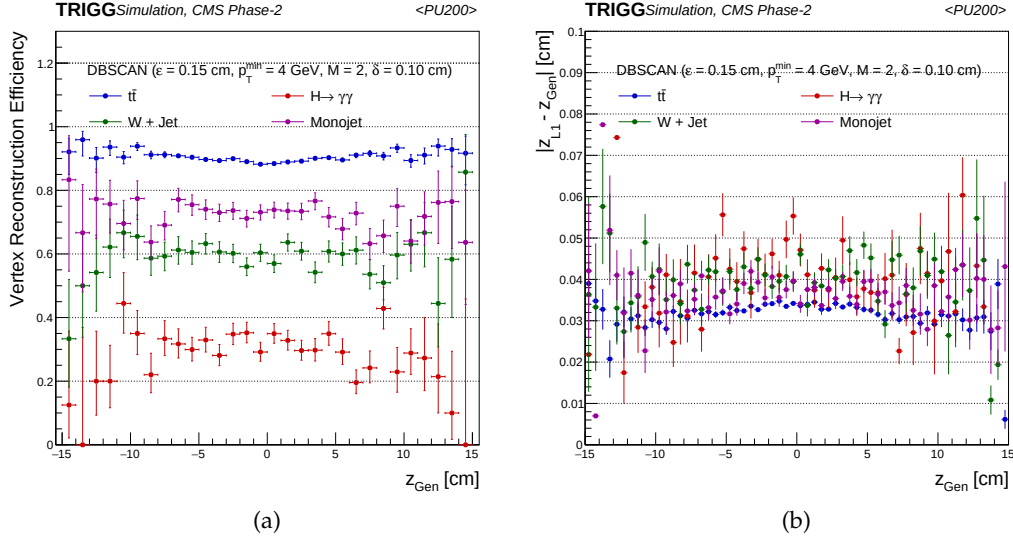


Figure 5.28: (a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} . (b) Resolution of the primary vertex longitudinal impact parameter as a function of z_{Gen} . These results have been obtained with an optimised DBSCAN algorithm applied to different signal events at a PU of 200.

5.4 VERTEXING WITH A TILTED BARREL GEOMETRY

Preliminary results have been obtained also applying the vertex reconstruction framework to the simulated samples with a tilted barrel tracker layout [77]. The utilised samples were known to have a problem to the emulation of the production of stubs, resulting into a lower track finding efficiency (av. 92%) and worse track parameter resolution, affecting also the performance of the downstream vertex reconstruction.

In any case, some preliminary conclusions can be drawn and are illustrated in the following sections.

5.4.1 Primary vertex reconstruction performance

Similarly to what has been done in the previous studies, the vertex reconstruction efficiency is defined with respect to the distance in z between the reconstructed and the simulated vertices. Figure 5.29 shows the measured efficiency as a function of the generated vertex longitudinal impact parameter and of the number of tracks

in the simulated vertex. Worse performance is clearly visible when comparing with the old flat geometry simulation results. A drop in efficiency of about 10% on average is visible when considering vertices correctly reconstructed within 1 mm. In any case, still 96.5% of the hard interaction primary vertices are reconstructed within 5 mm from the generated one.

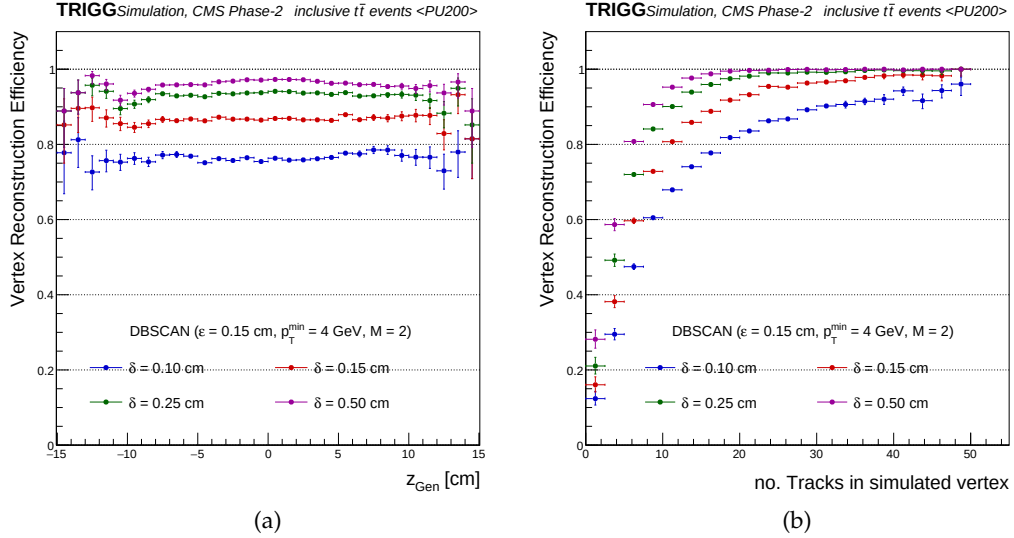


Figure 5.29: (a) Vertex reconstruction efficiency as a function of the generated primary vertex longitudinal impact parameter z_{Gen} (a) and of the number of tracks in the generated vertex (b), in top pair production events at a pileup of 200. These results have been obtained with an optimised DBSCAN algorithm and the latest available tracker geometry with tilted modules in the barrel region.

The worse track z_0 resolution reflects also on the vertex longitudinal impact parameter measurement. As shown in Figure 5.30a, the measured z_{L1} resolution is about 0.1 mm worsen than in the flat geometry case. On the contrary, no increase in the fake rate (Figure 5.30b) has been observed. This might be expected, since fewer tracks, and consequently, fewer fakes, are produced by the track finding processor with a tilted barrel layout. Even for vertices reconstructed within 0.5 mm the measured average fake rate is much lower (43% vs. 56%).

5.5 SUMMARY

The Phase-II upgrade of the CMS experiment scheduled for the next decade will improve significantly the detector trigger capability. In particular, the availability

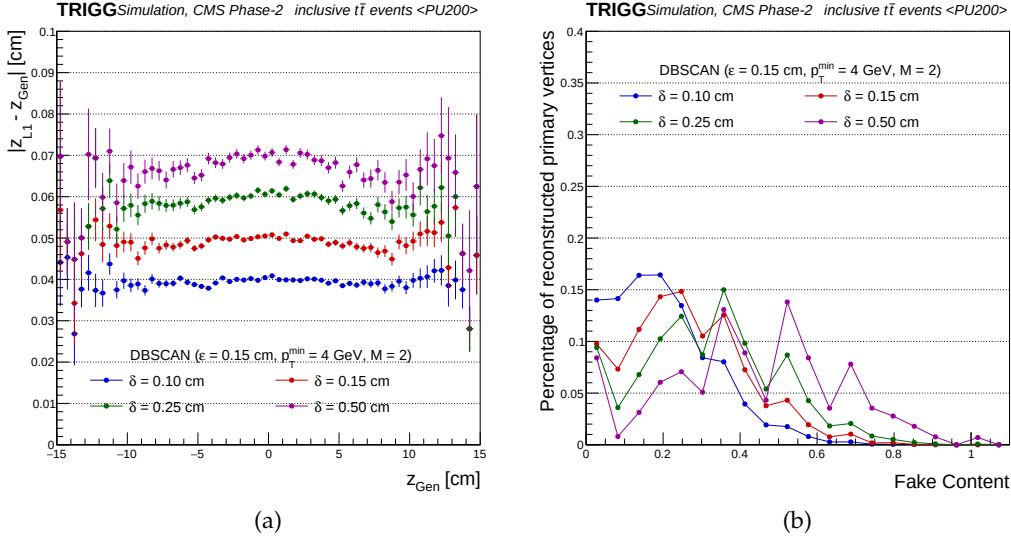


Figure 5.30: (a) Resolution of the longitudinal impact parameter z_{L1} of the reconstructed hard interaction primary vertex, as a function of the generated vertex z_{Gen} , and (b) number of misassigned tracks to the primary vertex in top pair production events using tracks with different p_T thresholds. These results have been obtained with an optimised DBSCAN algorithm.

of central track information at the L1 trigger level will open a wide new range of possibilities. One of the most interesting will be the prospect to reconstruct the primary vertices in the interaction.

The novel L1 vertex finding algorithm will probably run at L1 Correlator Trigger stage, where the track information from the entire tracker acceptance should be available. The chosen algorithm must show high reconstruction efficiency and at the same time must have an easy structure, which allows an FPGA implementation.

Several clustering methods have been therefore tested using simulated data samples with a CMS phase-II geometry, including flat or tilted tracker modules in the barrel region. The most promising algorithm has been identified in the so-called Density-Based Spatial Clustering of Applications with Noise (DBSCAN). It has been found in fact to be able to reconstruct correctly about 90% of the hard interaction vertices within 1.5 mm from the truth in top pair production events at a PU of 200. The algorithm has been also applied to other samples, highlighting its strong dependence from the signal track multiplicity.

The availability of primary vertex information allows also the estimation of some important properties, as the missing transverse momentum carried by charged particles in the interaction. This TrkMET has been found to be very robust against the pileup content and might be used in several parts of the trigger menu.

CONCLUSION

The CMS detector will undergo a major upgrade during the LHC shutdown period scheduled for the second half of the 2020s decade. The experiment will operate at the improved HL-LHC in a very challenging environment.

The physics program of the Phase II CMS upgrade will be still focused on the understanding of the electroweak breaking mechanism, measuring with high precision the properties of the recently discovered Higgs boson. Searches for new physics will be a crucial point of the upgrade program, the higher luminosity will indeed improve the sensitivity of the experiment and extend its discovery mass reach.

Major modifications are foreseen in many parts of the detector in order to maintain the excellent physics performance also under the challenging HL-LHC conditions. The increased number of pileup collisions will require major modifications to the hardware-based L1 Trigger, which should be designed in order to maintain the output data rate within acceptable limits without increasing the trigger thresholds. The upgraded trigger will benefit from a longer latency of up to 12.5 μ s to take an acceptance decision and would allow an increased output trigger rate up to 750 KHz.

This will be possible thanks to the addition of a L1 track finding system, which will reconstruct tracks using data from the Outer Tracker. The silicon tracker system will be therefore completely replaced since it will become inefficient because of the high radiation damage caused by ten years of LHC operations. The new OT will be composed of newly designed silicon stacked modules, capable of rejecting ‘on-detector’ signals not compatible with a high- p_T particle. The produced *stubs* will be input to the Track Trigger system at 40 MHz, which must reconstruct all the tracks from charged particles with a $p_T > 3$ GeV within 4 μ s.

Several proposals have been made for the architecture that will be used to build the Track Trigger system. Chapter 3 has presented a possible system based on the use of commercially available FPGA chips, which will perform the track finding task. The proposed system employs a fully time multiplexed architecture, which al-

allows the implementation of complex algorithms and does not require complicated data sharing between processors. The chosen track finding algorithm is based on the use of a Hough Transform to perform a first coarse pattern recognition on the $r\phi$ plane, followed by a finer filtering and fitting stage. Two different track filtering and fitting methods have been investigated. The first employs a Kalman Filter, similar to the official fitter currently implemented in the off-line reconstruction in CMS, combining both filtering and fitting tasks. A second alternative is based on a two-stage algorithm, called Seed Filter plus Linear Regression. The first step, the Seed Filter, rejects all the reconstructed Hough Transform patterns which do not follow a straight line trajectory on the rz plane. Surviving candidates are then fitted by means of a simple linear regression technique. The two approaches showed similar performance in terms of track finding efficiency and produced data rate. The proposed track finding system is able to reconstruct efficiently about 95% of tracks in top pair production events at a pileup of 200.

The track finding performance has been also demonstrated in real hardware. Chapter 4 illustrated the built hardware demonstrator system, based on the use of MP7 cards, equipped with powerful Virtex-7 FPGAs. The installed demonstrator replicated the tasks of a single Track Finder Processor, which reconstructs tracks in one eighth of the tracker solid angle and one slice in time. A software framework has been designed to operate the demonstrator data taking, injecting into the system simulated Phase-II outer tracker data. The functionalities of a single Track Finder Processor have been divided between five MP7 boards, each running a different stage of the algorithm. The SF+LR algorithm has been implemented in firmware and has been running successfully on two MP7s. Hardware demonstrator results have been measured and compared with expectations provided by a software emulation, showing excellent matching in terms of track finding efficiency and track parameter resolutions. The overall system has operated at a frequency of 240 MHz, employing 10 Gbps optical links for communications between boards. The measured latency is well below the aimed 4 μ s budget. A Track Finder Processor implementing a SF+LR fitter has been found to be much faster than a KF-based system, with a FirstIn-FirstOut latency of 2.7 μ s (KF: 3.5 μ s) and a total FirstIn-LastOut latency of 3.57 μ s (KF: 3.7 μ s).

The system has been recognised by the CMS collaboration as the most promising option for a final implementation. Therefore another hardware demonstrator system is planned to be built in the next years, implementing newer FPGAs and

faster optical links. A final system is expected to be based on Track Finder Processor boards, equipped with 2 or 3 Kintex Ultrascale 115 FPGAs. Further development of the algorithms and a novel tracker layout are expected to reduce significantly the amount of utilised resources and will permit to achieve even lower latencies.

The availability of track information at [L1](#) Trigger level opens a series of all-new possibilities. An interesting idea is to make use of the track information to reconstruct the primary vertex from the hard interaction in the event. [Chapter 5](#) has shown the undertaken studies for an L1 vertex finder. Several clustering algorithms have been investigated, with the [DBSCAN](#) that has been found to be the best compromise in terms of performance and hardware implementability. The algorithm is able to correctly identify the primary vertex in the collision in 90% of the cases within a tolerance of 1 mm in top pair production events at a pileup of 200. The reconstructed primary vertex allows also the measurement of interesting properties, as the sum of transverse momenta carried by charged particles or the tracker-based missing transverse energy, which might be used in downstream trigger menus.

SIMPLE LINEAR REGRESSION

A.1 REGRESSION ANALYSIS

The linear regression [134] analysis is a statistical method widely used in many fields to estimate relationships between variables.

Suppose to have a series of N pairs of measurements (x_i, y_i) , for which exists a linking function,

$$y = f(x_i, \mathbf{a}), \quad (66)$$

where $\mathbf{a}$ is a set of unknown parameters, corresponding to the physical quantities to determine. Using the least square method it is possible to give an estimate of those parameters, by minimising the distance between y and f .

Let us define the residual distance,

$$\varepsilon_i = y_i - f(x_i, \mathbf{a}). \quad (67)$$

The sum of the squared residuals,

$$D^2 = \sum_{i=0}^{N-1} \varepsilon_i^2 = \sum_{i=0}^{N-1} [y_i - f(x_i, \mathbf{a})]^2, \quad (68)$$

depends only on the parameters $\mathbf{a}$. Therefore, the best estimate of $\mathbf{a}$ can be found by minimising the function D^2 . Hence, by solving the equation

$$\frac{dD^2(\mathbf{a})}{d\mathbf{a}} = 0. \quad (69)$$

A.2 SIMPLE LINEAR REGRESSION

The simple linear regression represents a particular application of the linear regression methods, where the relationship between the two variables (x, y) is simply modelled by a straight line equation,

$$y = a + b \cdot x + \varepsilon, \quad (70)$$

where the intercept a and the slope b are unknown constants and ε is a random error component.

The constants a and b can be estimated using the set of N measured data (x_i, y_i) with the method of least squares. For each pair of measurements we can write down,

$$y_i = a + b \cdot x_i + \varepsilon_i, \quad i = 0, \dots, N. \quad (71)$$

The least square criterion is,

$$S(a, b) = \sum_{i=0}^N (y_i - a - b \cdot x_i)^2. \quad (72)$$

The best estimations of a and b , say $\hat{a}$ and $\hat{b}$, are chosen as the values that minimise $S(a, b)$. Therefore,

$$\left. \frac{\partial S}{\partial a} \right|_{\hat{a}, \hat{b}} = -2 \sum_{i=0}^N (y_i - \hat{a} - \hat{b} \cdot x_i) = 0 \quad (73)$$

$$\left. \frac{\partial S}{\partial b} \right|_{\hat{a}, \hat{b}} = -2 \sum_{i=0}^N (y_i - \hat{a} - \hat{b} \cdot x_i) \cdot x_i = 0. \quad (74)$$

The solution to these equations are,

$$\hat{a} = \frac{(\sum_i y_i)(\sum_i x_i^2) - (\sum_i x_i y_i)(\sum_i x_i)}{N(\sum_i x_i^2) - (\sum_i x_i)^2}, \quad (75)$$

$$\hat{b} = \frac{N(\sum_i x_i y_i) - (\sum_i y_i)(\sum_i x_i)}{N(\sum_i x_i^2) - (\sum_i x_i)^2}, \quad (76)$$

which define the fitted linear regression equation,

$$\hat{y} = \hat{a} + \hat{b} \cdot x, \quad (77)$$

that gives an estimation of y for any particular value of x .

It is possible now to write down the equation for the residual, defined as the difference between the observed value y_i and the fitted value $\hat{y}_i$,

$$e_i = y_i - \hat{y}_i = y_i - (\hat{a} + \hat{b} \cdot x_i). \quad (78)$$

The goodness of the fit can be estimated with a χ^2 test.

$$\chi^2 = \sum_{i=0}^N \left(\frac{y_i - \hat{y}_i}{\sigma_y^i} \right)^2, \quad (79)$$

where σ_y^i is the uncertainty on the measure y_i . The obtained χ^2 has N degrees of freedom and evaluates the validity of the fit.

IMPLEMENTING DIVISIONS IN FPGAS

Implementing division on a FPGA circuit is often a difficult task that should be avoided as much as possible. Several division algorithms can be exploited to implement the division, usually requiring a large fraction of resources in a device.

Xilinx provides inside Vivado a useful generator for Divider IP blocks [135], which guarantee excellent results in spite of being demanding in terms of area and timing on the FPGA.

An alternative is represented by the use of a serial fixed point division algorithm, which needs moderate resources but longer processing time [136].

B.1 A SERIAL FIXED POINT DIVISION IMPLEMENTATION

Consider the following division between two signals A and B,

$$\frac{A}{B}. \quad (80)$$

The simplest way to implement it consists on calculating the reciprocal value of B. Multiplying both A and B by a number power of 2,

$$\frac{A}{2^n} \cdot C, \quad (81)$$

where the values of $C = 2^n/B$ can be written in a Read Only Memory (ROM) with 2^n elements, implemented with lookup tables (LUTs) on the FPGA. Accessing the ROM needs only one clock cycle. Later the multiplication $A \cdot C$ can be computed with a DSP and, finally, the division by 2^n translates into a simple register right shifting.

The final result is a bit-word of $N_A + N_B$ bits, where $N_{A,B}$ are the width of the signals A and B. The precision of this method depends on the number of bits used to quantise the divider n, which defines also the resource utilisation. The implemented algorithm has a total latency of just three clock cycles.

ACRONYMS

ALICE	A Large Ion Collider Experiment
AM	Associative Memory
aPDs	Avalanche Photodiodes
ASIC	Application Specific Integrated Circuit
ATCA	Advanced Telecom Computer Architecture
ATLAS	A Toroidal LHC ApparatuS
BCT	Barrel Calorimeter Trigger
BE	Back-End
BMTF	Barrel Muon Track Finder
BRAM	Block RAM
BSM	Beyond the Standard Model
BX	Bunch Crossing
CBC	CMS Binary Chip
CERN	European Organization for Nuclear Research
CIC	Concentrator Integrated Circuit
CLB	Configurable Logic Block
CMS	Compact Muon Solenoid
CMSSW	CMS simulation software framework
CSC	Cathode Strip Chamber
CT	Correlator Trigger

DAQ	Data Acquisition
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DR	Duplicate Removal
DSP	Digital Signal Processing
DT	Drift Tube
DTC	Data, Trigger and Control
EB	ECAL Barrel
EC	Calorimeter endcap
ECAL	Electromagnetic Calorimeter
EE	ECAL Endcap
EMTF	Endcap Muon Track Finder
EVF	Event Filter Farm
FE	Front-End
FF	Flip-Flop
FSM	Finite State Machine
FPGA	Field Programmable Gate Array
FRL	Front-end Readout Link
LEP	Large Electron Positron collider
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty
LR	Linear Regression
LS	Long Shutdown
LUT	Look-Up Table
GEM	Gas Electron Multiplier

GP	Geometric Processor
GT	Global Trigger
HCAL	Hadron calorimeter
HB	Hadron barrel
HDL	Hardware Description Language
HE	Hadron endcap
HF	Hadron forward
HL-LHC	High Luminosity Large Hadron Collider
HLT	High Level Trigger
HO	Hadron outer
HT	Hough Transform
IC	Integrated Circuit
IT	Inner Tracker
KF	Kalman Filter
L1	Level-1
LUT	Lookup table
MCH	microTCA Carrier Hub
MET	Missing Transverse Energy
MPA	Macro-Pixel ASIC
OT	Outer Tracker
PS	Proton Synchrotron
PSB	Proton Synchrotron Booster
PVR	Principal Vertex Reconstructor
PU	Pile-Up

PW	Pattern Writer
QCD	Quantum Chromodynamics
QED	Quantum Electrodynamics
QFD	Quantum Flavourdynamics
ROM	Read Only Memory
RPC	Resistive Plate Chambers
SDP	Simple-Dual Port
SF	Seed Filter
SF+LR	Seed Filter plus Linear Regression
SLR	Simple Linear Regression
SM	Standard Model
SPS	Super Proton Synchrotron
SPPS	Super Proton–Antiproton Synchrotron
SSA	Short Strip ASIC
SUSY	SuperSymmetry
TBPX	Tracker Barrel Pixel Detector
TB2S	Tracker Barrel with 2S modules
TBPS	Tracker Barrel with PS modules
TCDS	Trigger Control Distribution System
TEC	Tracker EndCaps
TEDD	Tracker Endcap Double-Disks
TEPX	Tracker Endcap Pixel Detector
TFF	Track Filter and Fitter
TFP	Track Finder Processor

TFPX	Tracker Forward Pixel Detector
TIB/TID	Tracker Inner Barrel and Disks
TMT	Time Multiplexed Trigger
TMTT	Time Multiplexed Track Trigger
TOB	Tracker Outer Barrel
vPTs	Vacuum Phototriodes

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

- [1] D. Cieri et al. “L1 track finding for a time multiplexed trigger”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016). Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors, pp. 268 –269. ISSN: 0168-9002. DOI: <http://dx.doi.org/10.1016/j.nima.2015.09.117>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900215012206>.
- [2] D. Cieri, CMS Collaboration, and of the Time Multiplexed Track Trigger group. “Hardware Demonstrator of a Level-1 Track Finding Algorithm with FPGAs for the Phase II CMS Experiment”. In: *Journal of Physics: Conference Series* 762.1 (2016), p. 012020. URL: <http://stacks.iop.org/1742-6596/762/i=1/a=012020>.
- [3] C. Amstutz et al. “An FPGA-based track finder for the L1 trigger of the CMS experiment at the high luminosity LHC”. In: *2016 IEEE-NPSS Real Time Conference (RT)*. 2016, pp. 1–9. DOI: [10.1109/RTC.2016.7543102](https://doi.org/10.1109/RTC.2016.7543102).
- [4] Carlos Lourenco. *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*. Tech. rep. CERN-LHCC-2017-013. CMS-TDR-017. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2283192>.
- [5] Robin Cameron Aggleton et al. *An FPGA-Based Track Finder for the L1 Trigger of the CMS Experiment at the High Luminosity LHC*. Tech. rep. CMS-NOTE-2017-009. CERN-CMS-NOTE-2017-009. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2287640>.
- [6] R. Aggleton et al. “A novel FPGA-based track reconstruction approach for the level-1 trigger of the CMS experiment at CERN”. In: *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. 2017, pp. 1–4. DOI: [10.23919/FPL.2017.8056825](https://doi.org/10.23919/FPL.2017.8056825).

- [7] Cieri D. et al. “An FPGA-Based Track Finder for the L1 Trigger of the CMS Experiment at HL-LHC”. In: *Proceedings of the 2017 Topical Workshop on Electronics for Particle Physics (TWEPP 2017)*. Vol. 131. PoS(TWEPP-17). Proceedings of Science, 2017. URL: <https://pos.sissa.it/313/131/pdf>.

BIBLIOGRAPHY

- [1] Oliver Sim Brüning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. URL: <https://cds.cern.ch/record/782076>.
- [2] *Design Report Tevatron 1 project*. Tech. rep. FERMILAB-DESIGN-1984-01. 1984. URL: <https://cds.cern.ch/record/1478620>.
- [3] *LEP design report*. Copies shelved as reports in LEP, PS and SPS libraries. Geneva: CERN, 1984. URL: <https://cds.cern.ch/record/102083>.
- [4] A. Airapetian et al. "ATLAS: Detector and physics performance technical design report. Volume 1". In: (1999).
- [5] CMS Collaboration. "The CMS experiment at the CERN LHC. The Compact Muon Solenoid experiment". In: *J. Instrum.* 3 (2008). Also published by CERN Geneva in 2010, So8004. 361 p. URL: <http://cds.cern.ch/record/1129810>.
- [6] A. Augusto Alves Jr. et al. "The LHCb Detector at the LHC". In: *JINST* 3 (2008), So8005. DOI: [10.1088/1748-0221/3/08/S08005](https://doi.org/10.1088/1748-0221/3/08/S08005).
- [7] K. Aamodt et al. "The ALICE experiment at the CERN LHC". In: *JINST* 3 (2008), So8002. DOI: [10.1088/1748-0221/3/08/S08002](https://doi.org/10.1088/1748-0221/3/08/S08002).
- [8] Esma Mobs. "The CERN accelerator complex. Complexe des accélérateurs du CERN". In: (2016). General Photo. URL: <https://cds.cern.ch/record/2197559>.
- [9] *Concept of luminosity*. CERN. Geneva: CERN, 2006. URL: <http://cds.cern.ch/record/603056>.
- [10] M Bernardini and K Foraz. "Long Shutdown 2 @ LHC". In: (2015). URL: <https://cds.cern.ch/record/2031225>.
- [11] CMS Collaboration. *CMS Luminosity Public Results webpage*. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults>. 2017.

- [12] Manfred Markytan et al. *Letter of intent: by the CMS Collaboration for a general purpose detector at LHC*. Tech. rep. CERN-LHCC-92-003. CERN-LHCC-92-3. LHCC-I-1. Open presentation to the LHCC 5 November 1992, M. Della Negra/CERN, CMS Spokesman. Geneva: CERN, 1992. URL: <https://cds.cern.ch/record/290808>.
- [13] Wikipedia. *Standard Model — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Standard_Model&oldid=800876100. [Online; accessed 6-October-2017]. 2017.
- [14] S. Tomonaga. “On a Relativistically Invariant Formulation of the Quantum Theory of Wave Fields*”. In: *Progress of Theoretical Physics* 1.2 (1946), p. 27. DOI: [10.1143/PTP.1.27](https://doi.org/10.1143/PTP.1.27).
- [15] R. P. Feynman. “Mathematical Formulation of the Quantum Theory of Electromagnetic Interaction”. In: *Phys. Rev.* 80 (3 1950), pp. 440–457. DOI: [10.1103/PhysRev.80.440](https://link.aps.org/doi/10.1103/PhysRev.80.440). URL: <https://link.aps.org/doi/10.1103/PhysRev.80.440>.
- [16] Julian Schwinger. “Quantum Electrodynamics. I. A Covariant Formulation”. In: *Phys. Rev.* 74 (10 1948), pp. 1439–1461. DOI: [10.1103/PhysRev.74.1439](https://doi.org/10.1103/PhysRev.74.1439). URL: <https://link.aps.org/doi/10.1103/PhysRev.74.1439>.
- [17] Enrico Fermi. “Tentativo di una Teoria Dei Raggi β ”. In: *Il Nuovo Cimento* (1924-1942) 11.1 (2008), p. 1. ISSN: 1827-6121. DOI: [10.1007/BF02959820](https://doi.org/10.1007/BF02959820). URL: <http://dx.doi.org/10.1007/BF02959820>.
- [18] G. Arnison et al. “Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $s^{*}(1/2) = 540\text{-GeV}$ ”. In: *Phys. Lett.* 122B (1983). [611(1983)], pp. 103–116. DOI: [10.1016/0370-2693\(83\)91177-2](https://doi.org/10.1016/0370-2693(83)91177-2).
- [19] G. Arnison et al. “Experimental Observation of Lepton Pairs of Invariant Mass Around $95\text{-GeV}/c^{*2}$ at the CERN SPS Collider”. In: *Phys. Lett.* 126B (1983), pp. 398–410. DOI: [10.1016/0370-2693\(83\)90188-0](https://doi.org/10.1016/0370-2693(83)90188-0).
- [20] C. S. Wu, E. Ambler, R. W. Hayward, D. D. Hoppes, and R. P. Hudson. “Experimental Test of Parity Conservation in Beta Decay”. In: *Phys. Rev.* 105 (4 1957), pp. 1413–1415. DOI: [10.1103/PhysRev.105.1413](https://doi.org/10.1103/PhysRev.105.1413). URL: <https://link.aps.org/doi/10.1103/PhysRev.105.1413>.

- [21] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. “Evidence for the 2π Decay of the K_2^0 Meson”. In: *Phys. Rev. Lett.* 13 (4 1964), pp. 138–140. DOI: [10.1103/PhysRevLett.13.138](https://link.aps.org/doi/10.1103/PhysRevLett.13.138). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.138>.
- [22] Sheldon Lee Glashow. “Towards a unified theory: Threads in a tapestry”. In: *Rev. Mod. Phys.* 52 (3 1980), pp. 539–543. DOI: [10.1103/RevModPhys.52.539](https://link.aps.org/doi/10.1103/RevModPhys.52.539). URL: <https://link.aps.org/doi/10.1103/RevModPhys.52.539>.
- [23] Steven Weinberg. “Conceptual foundations of the unified theory of weak and electromagnetic interactions”. In: *Rev. Mod. Phys.* 52 (3 1980), pp. 515–523. DOI: [10.1103/RevModPhys.52.515](https://link.aps.org/doi/10.1103/RevModPhys.52.515). URL: <https://link.aps.org/doi/10.1103/RevModPhys.52.515>.
- [24] Abdus Salam. “Gauge unification of fundamental forces”. In: *Rev. Mod. Phys.* 52 (3 1980), pp. 525–538. DOI: [10.1103/RevModPhys.52.525](https://link.aps.org/doi/10.1103/RevModPhys.52.525). URL: <https://link.aps.org/doi/10.1103/RevModPhys.52.525>.
- [25] F. Englert and R. Brout. “Broken Symmetry and the Mass of Gauge Vector Mesons”. In: *Phys. Rev. Lett.* 13 (9 1964), pp. 321–323. DOI: [10.1103/PhysRevLett.13.321](https://link.aps.org/doi/10.1103/PhysRevLett.13.321).
- [26] Peter W. Higgs. “Broken Symmetries and the Masses of Gauge Bosons”. In: *Phys. Rev. Lett.* 13 (16 1964), pp. 508–509. DOI: [10.1103/PhysRevLett.13.508](https://link.aps.org/doi/10.1103/PhysRevLett.13.508). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.13.508>.
- [27] Alessandro Bettini. *Introduction to Elementary Particle Physics*. Cambridge University Press, 2008. DOI: [10.1017/CB09780511809019](https://doi.org/10.1017/CB09780511809019).
- [28] S Heinemeyer et al. *Handbook of LHC Higgs Cross Sections: 3. Higgs Properties: Report of the LHC Higgs Cross Section Working Group*. Tech. rep. CERN-2013-004. CERN-2013-004. Geneva, 2013. URL: <https://cds.cern.ch/record/1559921>.
- [29] Francis Halzen and Alan D. Martin. *Quarks and Leptones: An Introductory Course in Modern Particle Physics*. Wiley, 1984.
- [30] Helmut Satz. “The Quark-Gluon Plasma – A Short Introduction”. In: *Nuclear Physics A* 862 (2011). The Sixth International Conference on Physics and Astrophysics of Quark Gluon Plasma (ICPAQGP-2010), pp. 4–12. ISSN: 0375-9474. DOI: <http://dx.doi.org/10.1016/j.nuclphysa.2011.05.014>.

- [31] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: <http://dx.doi.org/10.1016/j.physletb.2012.08.020>. URL: <http://www.sciencedirect.com/science/article/pii/S037026931200857X>.
- [32] S. Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 30–61. ISSN: 0370-2693. DOI: <http://dx.doi.org/10.1016/j.physletb.2012.08.021>. URL: <http://www.sciencedirect.com/science/article/pii/S0370269312008581>.
- [33] Georges Aad et al. "Combined Measurement of the Higgs Boson Mass in pp Collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS Experiments". In: *Phys. Rev. Lett.* 114 (2015), p. 191803. DOI: [10.1103/PhysRevLett.114.191803](https://doi.org/10.1103/PhysRevLett.114.191803). arXiv: [1503.07589](https://arxiv.org/abs/1503.07589) [hep-ex].
- [34] R Aaij et al. "First Evidence for the Decay $B_s^0 \rightarrow \mu^+ \mu^-$ ". In: *Phys. Rev. Lett.* 110. CERN-PH-EP-2012-335. CERN-PH-EP-2012-335. LHCb-PAPER-2012-043 (2012). Comments: 3+6 pages, 2 figures, 021801. 9 p. URL: <https://cds.cern.ch/record/1493302>.
- [35] Serguei Chatrchyan et al. "Measurement of the B(s) to mu+ mu- branching fraction and search for Bo to mu+ mu- with the CMS Experiment". In: *Phys. Rev. Lett.* 111 (2013), p. 101804. DOI: [10.1103/PhysRevLett.111.101804](https://doi.org/10.1103/PhysRevLett.111.101804). arXiv: [1307.5025](https://arxiv.org/abs/1307.5025) [hep-ex].
- [36] F. Zwicky. "Die Rotverschiebung von extragalaktischen Nebeln". In: *Helvetica Physica Acta* 6 (1933), pp. 110–127.
- [37] M.C. Gonzalez-Garcia and Michele Maltoni. "Phenomenology with massive neutrinos". In: *Physics Reports* 460.1 (2008), pp. 1–129. ISSN: 0370-1573. DOI: <http://dx.doi.org/10.1016/j.physrep.2007.12.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0370157308000148>.
- [38] J. Alcaraz et al. "Search for antihelium in cosmic rays". In: *Physics Letters B* 461.4 (1999), pp. 387–396. ISSN: 0370-2693. DOI: [http://dx.doi.org/10.1016/S0370-2693\(99\)00874-6](http://dx.doi.org/10.1016/S0370-2693(99)00874-6). URL: <http://www.sciencedirect.com/science/article/pii/S0370269399008746>.

- [39] A. G. Mayorov et al. “The search for antihelium in cosmic rays using data from the PAMELA experiment”. In: *Bulletin of the Russian Academy of Sciences: Physics* 75.3 (2011), pp. 331–333. ISSN: 1934-9432. DOI: [10.3103/S1062873811030282](https://doi.org/10.3103/S1062873811030282). URL: <http://dx.doi.org/10.3103/S1062873811030282>.
- [40] *Search for supersymmetry using hadronic top quark tagging in 13 TeV pp collisions*. Tech. rep. CMS-PAS-SUS-16-050. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2262651>.
- [41] G. Aad et al. “Summary of the ATLAS experiment’s sensitivity to supersymmetry after LHC Run 1 — interpreted in the phenomenological MSSM”. In: *Journal of High Energy Physics* 2015.10 (2015), p. 134. ISSN: 1029-8479. DOI: [10.1007/JHEP10\(2015\)134](https://doi.org/10.1007/JHEP10(2015)134). URL: [http://dx.doi.org/10.1007/JHEP10\(2015\)134](http://dx.doi.org/10.1007/JHEP10(2015)134).
- [42] Burkhard Schmidt. “The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments”. In: *J. Phys. : Conf. Ser.* 706.2 (2016), 022002. 42 p. URL: <http://cds.cern.ch/record/2263093>.
- [43] L Rossi, S Stavrev, and A Szeberenyi. “3rd Periodic HiLumi LHC Report: Deliverable D1.13”. In: (2016). URL: <http://cds.cern.ch/record/2121246>.
- [44] S. Borowka, N. Greiner, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, U. Schubert, and T. Zirke. “Higgs Boson Pair Production in Gluon Fusion at Next-to-Leading Order with Full Top-Quark Mass Dependence”. In: *Phys. Rev. Lett.* 117 (1 2016), p. 012001. DOI: [10.1103/PhysRevLett.117.012001](https://doi.org/10.1103/PhysRevLett.117.012001). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.117.012001>.
- [45] Peter Krieger. *Standard Model and Higgs Physics at the HL-LHC with ATLAS and CMS*. Tech. rep. ATL-PHYS-PROC-2016-148. Geneva: CERN, 2016. URL: <https://cds.cern.ch/record/2215889>.
- [46] D Contardo, M Klute, J Mans, L Silvestris, and J Butler. *Technical Proposal for the Phase-II Upgrade of the CMS Detector*. Tech. rep. CERN-LHCC-2015-010. LHCC-P-008. CMS-TDR-15-02. Geneva, 2015.
- [47] *Supersymmetry discovery potential in future LHC and HL-LHC running with the CMS detector*. Tech. rep. CMS-PAS-SUS-14-012. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/1981344>.

- [48] *ATLAS Phase-II Upgrade Scoping Document*. Tech. rep. CERN-LHCC-2015-020. LHCC-G-166. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/2055248>.
- [49] LHCb Collaboration. *Expression of Interest for a Phase-II LHCb Upgrade: Opportunities in flavour physics, and beyond, in the HL-LHC era*. Tech. rep. CERN-LHCC-2017-003. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2244311>.
- [50] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter of Intent*. Tech. rep. CERN-LHCC-2012-012. LHCC-I-022. ALICE-UG-002. Geneva: CERN, 2012. URL: <https://cds.cern.ch/record/1475243>.
- [51] C. Patrignani et al. “Review of Particle Physics”. In: *Chin. Phys. C* 40.10 (2016), p. 100001. DOI: [10.1088/1674-1137/40/10/100001](https://doi.org/10.1088/1674-1137/40/10/100001).
- [52] C. Leroy and P.-G. Rancoita. “Particle interaction and displacement damage in silicon devices operated in radiation environments”. In: *Reports on Progress in Physics* 70 (Apr. 2007), pp. 493–625. DOI: [10.1088/0034-4885/70/4/R01](https://doi.org/10.1088/0034-4885/70/4/R01).
- [53] Thomas Mc Cauley. “Collisions recorded by the CMS detector on 14 Oct 2016 during the high pile-up fill”. CMS Collection. 2016. URL: <https://cds.cern.ch/record/2231915>.
- [54] Tai Sakuma and Thomas McCauley. “Detector and Event Visualization with SketchUp at the CMS Experiment”. In: *Journal of Physics: Conference Series* 513.2 (2014), p. 022032. URL: <http://stacks.iop.org/1742-6596/513/i=2/a=022032>.
- [55] V Karimäki et al. *The CMS tracker system project: Technical Design Report*. Technical Design Report CMS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/368412>.
- [56] V.R. Tavolaro. “The Phase1 CMS Pixel detector upgrade”. In: *Journal of Instrumentation* 11.12 (2016), p. C12010. URL: <http://stacks.iop.org/1748-0221/11/i=12/a=C12010>.
- [57] A Dominguez et al. *CMS Technical Design Report for the Pixel Detector Upgrade*. Tech. rep. CERN-LHCC-2012-016. CMS-TDR-11. Additional contacts: Jeffrey Spalding, Fermilab, Jeffrey.Spalding@cern.ch Didier Contardo, Uni-

- versite Claude Bernard-Lyon I, didier.claude.contardo@cern.ch. 2012. URL: <https://cds.cern.ch/record/1481838>.
- [58] M M Angarano. *The Silicon Strip Tracker for CMS*. Tech. rep. CMS-CR-2002-006. Geneva: CERN, 2002. URL: <https://cds.cern.ch/record/687420>.
- [59] A. Affolder. “The CMS Silicon Strip Tracker: Design and Production Status”. In: *Nuclear Physics B - Proceedings Supplements* 150. Supplement C (2006). Proceedings of the 9th Topical Seminar on Innovative Particle and Radiation Detectors, pp. 118 –123. ISSN: 0920-5632. DOI: <https://doi.org/10.1016/j.nuclphysbps.2004.08.037>. URL: <http://www.sciencedirect.com/science/article/pii/S0920563205008236>.
- [60] *The CMS magnet project: Technical Design Report*. Technical Design Report CMS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/331056>.
- [61] *The CMS electromagnetic calorimeter project: Technical Design Report*. Technical Design Report CMS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/349375>.
- [62] *The CMS hadron calorimeter project: Technical Design Report*. Technical Design Report CMS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/357153>.
- [63] CMS Collaboration. “Measurement of differential and integrated fiducial cross sections for Higgs boson production in the four-lepton decay channel in pp collisions at $\sqrt{s} = 7$ and 8 TeV”. In: *JHEP* 04.arXiv:1512.08377. CMS-HIG-14-028. CERN-PH-EP-2015-285 (2015), 005. 42 p. URL: <https://cds.cern.ch/record/2118088>.
- [64] *The CMS muon project: Technical Design Report*. Technical Design Report CMS. Geneva: CERN, 1997. URL: <https://cds.cern.ch/record/343814>.
- [65] CMS Collaboration. *CMS TriDAS project: Technical Design Report, Volume 1: The Trigger Systems*. Technical Design Report CMS. URL: <http://cds.cern.ch/record/706847>.
- [66] A Tapper and Darin Acosta. *CMS Technical Design Report for the Level-1 Trigger Upgrade*. Tech. rep. CERN-LHCC-2013-011. CMS-TDR-12. Additional contacts: Jeffrey Spalding, Fermilab, Jeffrey.Spalding@cern.ch Didier Contardo, Universite Claude Bernard-Lyon I, didier.claude.contardo@cern.ch. 2013. URL: <http://cds.cern.ch/record/1556311>.

- [67] Sergio Cittolin, Attila Rácz, and Paris Sphicas. *CMS The TriDAS Project: Technical Design Report, Volume 2: Data Acquisition and High-Level Trigger*. CMS trigger and data-acquisition project. Technical Design Report CMS. Geneva: CERN, 2002. URL: <https://cds.cern.ch/record/578006>.
- [68] CMS Collaboration. *CMS Software*. <https://github.com/cms-sw>. Accessed: 2017-06-15.
- [69] Andrea Perrotta. “Performance of the CMS High Level Trigger”. In: *Journal of Physics: Conference Series* 664.8 (2015), p. 082044. URL: <http://stacks.iop.org/1742-6596/664/i=8/a=082044>.
- [70] T. Han. *The Dawn of the LHC Era: TASI 2008 : Proceedings of the 2008 Theoretical Advanced Study Institute in Elementary Particle Physics, Boulder, Colorado, USA, 2-27 June 2008*. World Scientific, 2010. ISBN: 9789812838353. URL: <https://books.google.co.uk/books?id=gm6MtG8oo9sC>.
- [71] Stefano Mersi. “Phase-2 Upgrade of the CMS Tracker”. In: *Nucl. Part. Phys. Proc.* 273-275 (2016), pp. 1034–1041. DOI: [10.1016/j.nuclphysbps.2015.09.162](https://doi.org/10.1016/j.nuclphysbps.2015.09.162).
- [72] B Bilki and the CMS Collaboration. “CMS Forward Calorimeters Phase II Upgrade”. In: *Journal of Physics: Conference Series* 587.1 (2015), p. 012014. URL: <http://stacks.iop.org/1742-6596/587/i=1/a=012014>.
- [73] G Bianchi. “tkLayout: a design tool for innovative silicon tracking detectors”. In: *Journal of Instrumentation* 9.03 (2014), p. C03054. URL: <http://stacks.iop.org/1748-0221/9/i=03/a=C03054>.
- [74] CMS Collaboration. *CMS Technical Design Report for the Phase-2 Tracker Upgrade*. Tech. rep. CERN-LHCC-2017-xxx. CMS-TDR-xxx. Geneva, 2017. URL: <https://cds.cern.ch/record/2020886>.
- [75] J (CERN) Chistiansen and M (LBNL) Garcia-Sciveres. *RD Collaboration Proposal: Development of pixel readout integrated circuits for extreme rate and radiation*. Tech. rep. CERN-LHCC-2013-008. LHCC-P-006. The authors are editors on behalf of the participating institutes. the participating institutes are listed in the proposal. Geneva: CERN, 2013. URL: <http://cds.cern.ch/record/1553467>.

- [76] K Klein. *The Phase-2 Upgrade of the CMS Tracker*. Tech. rep. CERN-LHCC-2017-009. CMS-TDR-014. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2272264>.
- [77] Carlos Lourenco. *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*. Tech. rep. CERN-LHCC-2017-013. CMS-TDR-017. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2283192>.
- [78] A.-M. Magnan. “HGCAL: a High-Granularity Calorimeter for the endcaps of CMS at HL-LHC”. In: *Journal of Instrumentation* 12.01 (2017), p. C01042. URL: <http://stacks.iop.org/1748-0221/12/i=01/a=C01042>.
- [79] *Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET*. Tech. rep. CMS-PAS-PFT-09-001. Geneva: CERN, 2009. URL: <https://cds.cern.ch/record/1194487>.
- [80] G.P. Heath et al. “The ZEUS first level tracking trigger”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 315.1 (1992), pp. 431–435. ISSN: 0168-9002. DOI: [http://dx.doi.org/10.1016/0168-9002\(92\)90741-L](http://dx.doi.org/10.1016/0168-9002(92)90741-L). URL: <http://www.sciencedirect.com/science/article/pii/016890029290741L>.
- [81] A. Abulencia et al. “The CDF II level 1 track trigger upgrade”. In: *IEEE Trans. Nucl. Sci.* 55 (2008), pp. 126–132. DOI: [10.1109/RTC.2007.4382856](https://doi.org/10.1109/RTC.2007.4382856), [10.1109/TNS.2007.911618](https://doi.org/10.1109/TNS.2007.911618).
- [82] Fedi, Giacomo. “Associative Memory Pattern Matching for the L1 Track Trigger of CMS at the HL-LHC”. In: *EPJ Web Conf.* 127 (2016), p. 00008. DOI: [10.1051/epjconf/201612700008](https://doi.org/10.1051/epjconf/201612700008). URL: <https://doi.org/10.1051/epjconf/201612700008>.
- [83] Skinnari, Louise. “FPGA-Based Approach to Level-1 Track Finding at CMS for the HL-LHC”. In: *EPJ Web Conf.* 127 (2016), p. 00017. DOI: [10.1051/epjconf/201612700017](https://doi.org/10.1051/epjconf/201612700017).
- [84] Xilinx Incorporated. *7 Series FPGAs Data Sheet: Overview*. https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf. Mar. 2017.

- [85] A. Svetek et al. "The Calorimeter Trigger Processor Card: the next generation of high speed algorithmic data processing at CMS". In: *Journal of Instrumentation* 11.02 (2016), p. C02011. URL: <http://stacks.iop.org/1748-0221/11/i=02/a=C02011>.
- [86] S. M. Trimberger. "Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology". In: *Proceedings of the IEEE* 103.3 (2015), pp. 318–331. ISSN: 0018-9219. DOI: [10.1109/JPROC.2015.2392104](https://doi.org/10.1109/JPROC.2015.2392104).
- [87] Umer Farooq, Zied Marrakchi, and Habib Mehrez. *Tree Based Heterogeneous FPGA Architectures, Application Specific Exploration and Optimization*. Springer, 2012. DOI: [10.1007/978-1-4614-3594-5](https://doi.org/10.1007/978-1-4614-3594-5). URL: <https://hal.archives-ouvertes.fr/hal-01272904>.
- [88] Paul S. Zuchowski, Christopher B. Reynolds, Richard J. Grupp, Shelly G. Davis, Brendan Cremen, and Bill Troxel. "A Hybrid ASIC and FPGA Architecture". In: *Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design. ICCAD '02*. San Jose, California: ACM, 2002, pp. 187–194. ISBN: 0-7803-7607-2. DOI: [10.1145/774572.774600](https://doi.org/10.1145/774572.774600). URL: <http://doi.acm.org/10.1145/774572.774600>.
- [89] B. Kreis et al. "Run 2 upgrades to the CMS Level-1 calorimeter trigger". In: *Journal of Instrumentation* 11.01 (2016), p. C01051. URL: <http://stacks.iop.org/1748-0221/11/i=01/a=C01051>.
- [90] G. Hall. "A time-multiplexed track-trigger for the CMS HL-LHC upgrade". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016). Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors, pp. 292–295. ISSN: 0168-9002. DOI: <http://dx.doi.org/10.1016/j.nima.2015.09.075>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900215011304>.
- [91] G. Hall, D. Newbold, M. Pesaresi, and A. Rose. "A time-multiplexed track-trigger architecture for CMS". In: *Journal of Instrumentation* 9.10 (2014). URL: <http://stacks.iop.org/1748-0221/9/i=10/a=C10034>.
- [92] H.P.V. C. *Method and means for recognizing complex patterns*. US Patent 3,069,654. 1962. URL: <https://www.google.com/patents/US3069654>.

- [93] D. Cieri et al. “L1 track finding for a time multiplexed trigger”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016). Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors, pp. 268 –269. ISSN: 0168-9002. DOI: <http://dx.doi.org/10.1016/j.nima.2015.09.117>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900215012206>.
- [94] D. Cieri, CMS Collaboration, and of the Time Multiplexed Track Trigger group. “Hardware Demonstrator of a Level-1 Track Finding Algorithm with FPGAs for the Phase II CMS Experiment”. In: *Journal of Physics: Conference Series* 762.1 (2016), p. 012020. URL: <http://stacks.iop.org/1742-6596/762/i=1/a=012020>.
- [95] C. Amstutz et al. “An FPGA-based track finder for the L1 trigger of the CMS experiment at the high luminosity LHC”. In: *2016 IEEE-NPSS Real Time Conference (RT)*. 2016, pp. 1–9. DOI: [10.1109/RTC.2016.7543102](https://doi.org/10.1109/RTC.2016.7543102).
- [96] G L Bayatian et al. *CMS Physics: Technical Design Report Volume 1: Detector Performance and Software*. Technical Design Report CMS. There is an error on cover due to a technical problem for some items. Geneva: CERN, 2006. URL: <https://cds.cern.ch/record/922757>.
- [97] K Compton et al. “The MP7 and CTP-6: multi-hundred Gbps processing boards for calorimeter trigger upgrades at CMS”. In: *Journal of Instrumentation* 7.12 (2012), p. C12024. URL: <http://stacks.iop.org/1748-0221/7/i=12/a=C12024>.
- [98] E Hazen, A Heister, C Hill, J Rohlf, S X Wu, and D Zou. “The AMC13XG: a new generation clock/timing/DAQ module for CMS MicroTCA”. In: *Journal of Instrumentation* 8.12 (2013), p. C12036. URL: <http://stacks.iop.org/1748-0221/8/i=12/a=C12036>.
- [99] Robin Cameron Aggleton et al. *An FPGA-Based Track Finder for the L1 Trigger of the CMS Experiment at the High Luminosity LHC*. Tech. rep. CMS-NOTE-2017-009. CERN-CMS-NOTE-2017-009. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2287640>.

- [100] Xilinx Incorporated. *UltraScale Architecture and Product Data Sheet: Overview*. https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf. Nov. 2017.
- [101] Richard O. Duda and Peter E. Hart. "Use of the Hough Transformation to Detect Lines and Curves in Pictures". In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15. ISSN: 0001-0782. DOI: [10.1145/361237.361242](https://doi.acm.org/10.1145/361237.361242). URL: <http://doi.acm.org/10.1145/361237.361242>.
- [102] Rudolph Emil Kalman et al. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.
- [103] R. Faragher. "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes]". In: *IEEE Signal Processing Magazine* 29.5 (2012), pp. 128–132. ISSN: 1053-5888. DOI: [10.1109/MSP.2012.2203621](https://doi.org/10.1109/MSP.2012.2203621).
- [104] PICMG. *Micro Telecommunications Computing Architecture Short Form Specification*. 2006.
- [105] Avago Technologies. *MicroPOD™; and MiniPOD™; 120G transmitters/receivers*. [Online; accessed 19-July-2008]. 2011.
- [106] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, A. Rose, A. Thea, and T. Williams. "IPbus: a flexible Ethernet-based control system for xTCA hardware". In: *Journal of Instrumentation* 10.02 (2015), p. C02019. URL: <http://stacks.iop.org/1748-0221/10/i=02/a=C02019>.
- [107] M. Technologies. *MaxCompiler (White Paper)*. <https://www.maxeler.com/media/documents/MaxelerWhitePaperMaxCompiler.pdf>. 2011.
- [108] "IEEE Standard VHDL Language Reference Manual". In: *IEEE Std 1076-2008 (Revision of IEEE Std 1076-2002)* (2009), pp. c1–626. DOI: [10.1109/IEEESTD.2009.4772740](https://doi.org/10.1109/IEEESTD.2009.4772740).
- [109] S. Summers, A. Rose, and P. Sanders. "Using MaxCompiler for the high level synthesis of trigger algorithms". In: *Journal of Instrumentation* 12.02 (2017), p. C02015. URL: <http://stacks.iop.org/1748-0221/12/i=02/a=C02015>.

- [110] R. Aggleton et al. “A novel FPGA-based track reconstruction approach for the level-1 trigger of the CMS experiment at CERN”. In: *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. 2017, pp. 1–4. DOI: [10.23919/FPL.2017.8056825](https://doi.org/10.23919/FPL.2017.8056825).
- [111] Xilinx Incorporated. *UltraScale Architecture and Product Data Sheet: Overview*. https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf. Feb. 2017.
- [112] Lior Rokach and Oded Maimon. “Clustering Methods”. In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Oded Maimon and Lior Rokach. Boston, MA: Springer US, 2005, pp. 321–352. ISBN: 978-0-387-25465-4. DOI: [10.1007/0-387-25465-X_15](https://doi.org/10.1007/0-387-25465-X_15). URL: https://doi.org/10.1007/0-387-25465-X_15.
- [113] The Track Trigger Integration group. *Use of tracking in the CMS L1 trigger for the phase-2 upgrade*. Tech. rep. CMS DN-14-002. Internal note. Geneva, 2014.
- [114] Louis L. McQuitty. “Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies”. In: *Educational and Psychological Measurement* 17.2 (1957), pp. 207–229. DOI: [10.1177/001316445701700204](https://doi.org/10.1177/001316445701700204). URL: <https://doi.org/10.1177/001316445701700204>.
- [115] P. MacNaughton-Ith, W. T. Willia, M. B. Dale, and L. G. Mockett. “Dis-similarity Analysis: a new Technique of Hierarchical Sub-division”. In: *Nature* 202.4936 (June 1964), pp. 1034–1035. ISSN: 0028-0836. DOI: [10.1038/2021034a0](https://doi.org/10.1038/2021034a0). URL: <http://https://doi.org/10.1038/2021034a0>.
- [116] Joe H. Ward Jr. “Hierarchical Grouping to Optimize an Objective Function”. In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244. DOI: [10.1080/01621459.1963.10500845](https://doi.org/10.1080/01621459.1963.10500845). eprint: <http://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500845>. URL: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845>.
- [117] R Sibson. “SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method”. In: 16.1 (Jan. 1973), pp. 30–34. DOI: [10.1093/comjnl/16.1.30](https://doi.org/10.1093/comjnl/16.1.30).
- [118] D. Defays. “An efficient algorithm for a complete link method”. In: 20.4 (1977), pp. 364–366. DOI: [10.1093/comjnl/20.4.364](https://doi.org/10.1093/comjnl/20.4.364).

- [119] Hinrich Schütze Christopher D. Manning Prabhakar Raghavan. “An Introduction to Information Retrieval”. In: Cambridge, UK: Cambridge University Press, 2009, pp. 377–402. ISBN: 9780521865715. URL: <https://nlp.stanford.edu/IR-book>.
- [120] Wolfgang Waltenberger. “Development of Vertex Finding and Vertex Fitting Algorithms for CMS”. PhD thesis. Vienna, Tech. U., 2004. URL: https://inspirehep.net/record/894094/files/TS2006_012.pdf.
- [121] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. “Density-based clustering”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.3 (2011), pp. 231–240. ISSN: 1942-4795. DOI: [10.1002/widm.30](https://doi.org/10.1002/widm.30). URL: <http://dx.doi.org/10.1002/widm.30>.
- [122] D. Wishart. “Mode Analysis: a generalization of nearest neighbour which reduces chaining effects”. In: *Numerical Taxonomy, London and New York* (1969).
- [123] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD’96*. Portland, Oregon: AAAI Press, 1996, pp. 226–231. URL: <http://dl.acm.org/citation.cfm?id=3001460.3001507>.
- [124] J. MacQueen. *Some methods for classification and analysis of multivariate observations*. English. Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281–297 (1967). 1967.
- [125] S. Z. Selim and M. A. Ismail. “K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6.1* (1984), pp. 81–87. ISSN: 0162-8828. DOI: [10.1109/TPAMI.1984.4767478](https://doi.org/10.1109/TPAMI.1984.4767478).
- [126] Neil Scicluna and Christos-Savvas Bouganis. “FPGA-Based Parallel DB-SCAN Architecture”. In: *Reconfigurable Computing: Architectures, Tools, and Applications: 10th International Symposium, ARC 2014, Vilamoura, Portugal, April 14–16, 2014. Proceedings*. Ed. by Diana Goehringer, Marco Domenico Santambrogio, João M. P. Cardoso, and Koen Bertels. Cham: Springer International Publishing, 2014, pp. 1–12. ISBN: 978-3-319-05960-0. DOI: [10.1007/](https://doi.org/10.1007/)

- 978-3-319-05960-0_1. URL: https://doi.org/10.1007/978-3-319-05960-0_1.
- [127] H. M. Hussain, K. Benkrid, H. Seker, and A. T. Erdogan. “FPGA implementation of K-means algorithm for bioinformatics application: An accelerated approach to clustering Microarray data”. In: *2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. 2011, pp. 248–255. DOI: [10.1109/AHS.2011.5963944](https://doi.org/10.1109/AHS.2011.5963944).
- [128] *Search for dark matter, invisible Higgs boson decays, and large extra dimensions in the $\ell\ell + E_T^{\text{miss}}$ final state using 2016 data*. Tech. rep. CMS-PAS-EXO-16-052. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2264690>.
- [129] Albert M Sirunyan et al. *Search for associated production of dark matter with a Higgs boson decaying to $b\bar{b}$ or $\gamma\gamma$ at $\sqrt{s} = 13$ TeV*. Tech. rep. CMS-EXO-16-012. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2255523>.
- [130] *Search for GMSB supersymmetry in events with at least one photon and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV*. Tech. rep. CMS-PAS-SUS-16-046. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2256754>.
- [131] Tom Cornelis and the CMS collaboration. “Vector boson plus jet physics at CMS”. In: *Journal of Physics: Conference Series* 455.1 (2013), p. 012026. URL: <http://stacks.iop.org/1742-6596/455/i=1/a=012026>.
- [132] *Search for new physics in monojet events in pp collisions at $\sqrt{s} = 8$ TeV*. Tech. rep. CMS-PAS-EXO-12-048. Geneva: CERN, 2013. URL: <https://cds.cern.ch/record/1525585>.
- [133] Vardan Khachatryan et al. “Measurement of differential cross sections for Higgs boson production in the diphoton decay channel in pp collisions at $\sqrt{s} = 8$ TeV”. In: *Eur. Phys. J. C* 76.CMS-HIG-14-016. CMS-HIG-14-016. CERN-PH-EP-2015-195 (2015). Replaced with published version. Added journal reference and DOI, 13. 44 p. URL: <https://cds.cern.ch/record/2047828>.
- [134] G. Geoffrey Vining Douglas C. Montgomery Elizabeth A. Peck. *Introduction to Linear Regression Analysis*. 5th ed. John Wiley and Sons, Inc., 2012. ISBN: 978-0-470-54281-1.

- [135] Xilinx Incorporated. *Divider Generator v5.1, LogiCORE IP Product Guide*. https://www.xilinx.com/support/documentation/ip_documentation/div_gen/v5_1/pg151-div-gen.pdf. 2016.
- [136] Jean-Pierre Deschamps, Gustavo D. Sutter, and Enrique Cant. *Guide to FPGA Implementation of Arithmetic Functions*. Springer Publishing Company, Incorporated, 2014. ISBN: 9401784388, 9789401784382.