

MARIO GRANDI

HLS DEVELOPMENTS AND PLANS AT SUSSEX



- Summary of my goals framed within the WP4 plan of activities
- The status of tracking by ATLAS for the HL-LHC environment and the areas of overlap with SWIFT-HEP
 - Heard from Liv a quick summary of the general status of CMS tracking and the ongoing work in Imperial on the use of GNN for particle tracking
- The ongoing and planned work carried out at Sussex
- Plans for future

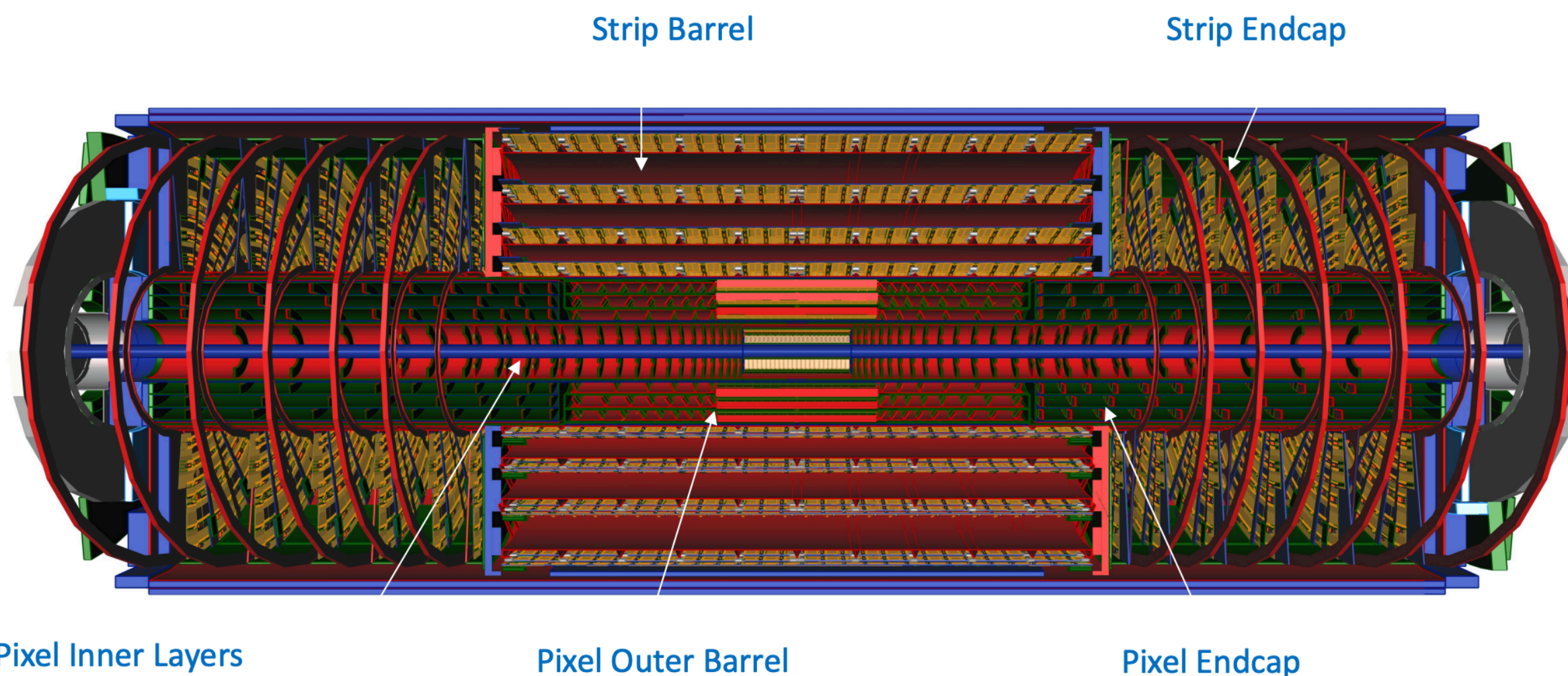
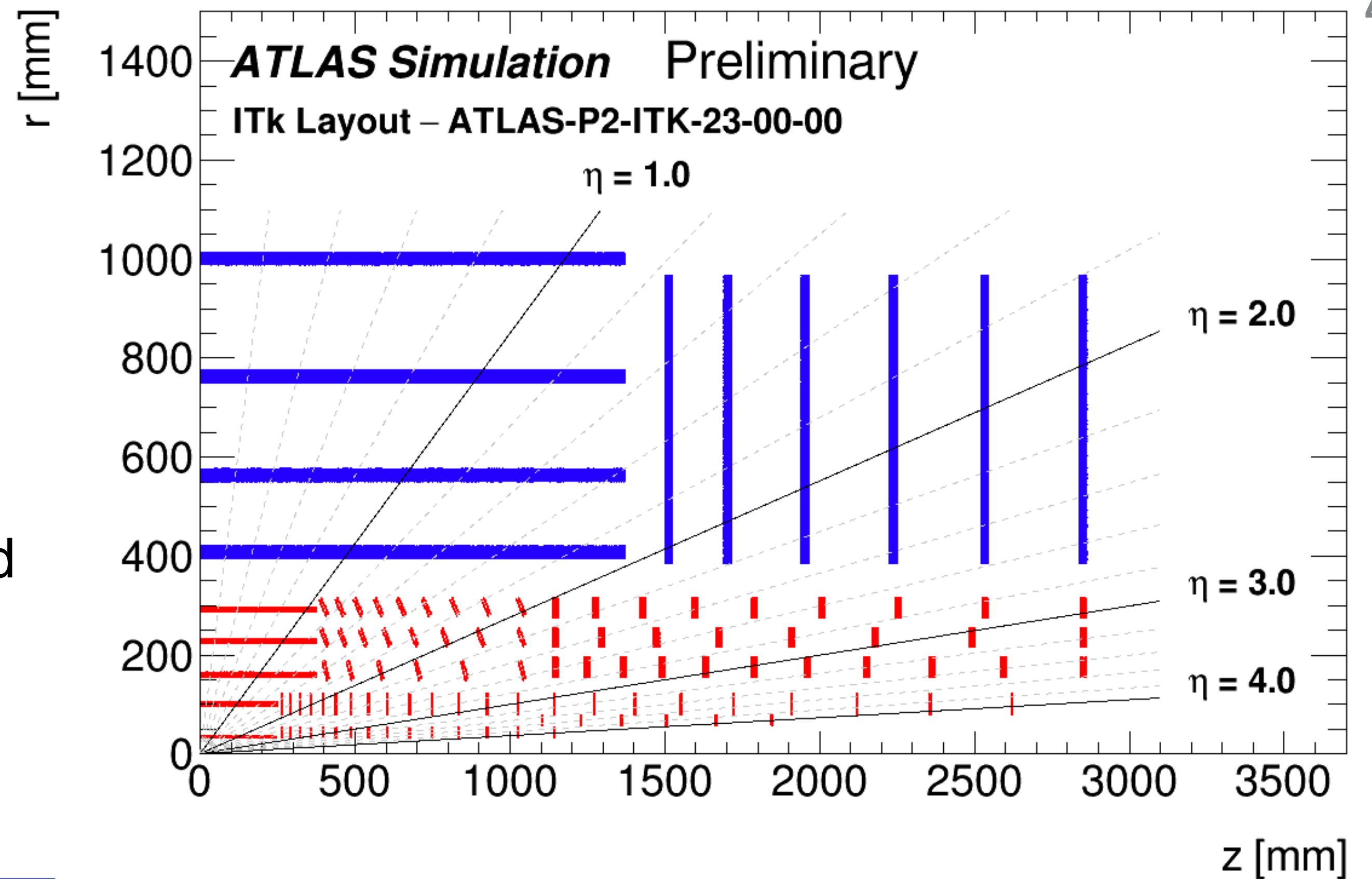
- Deployment and benchmarking of tracking and vertexing algorithms on FPGA accelerators
- Prototype of tracking and vertexing algorithms on heterogeneous compute farms

Institute	Role		Start	End	PM
RAL	PDRA	WP4.1	1	24	12
RAL	PDRA	WP4.2	9	24	8
Sussex	PDRA	WP4.1	1	24	12
Warwick	RSE	WP4.1/4.2	10	36	14

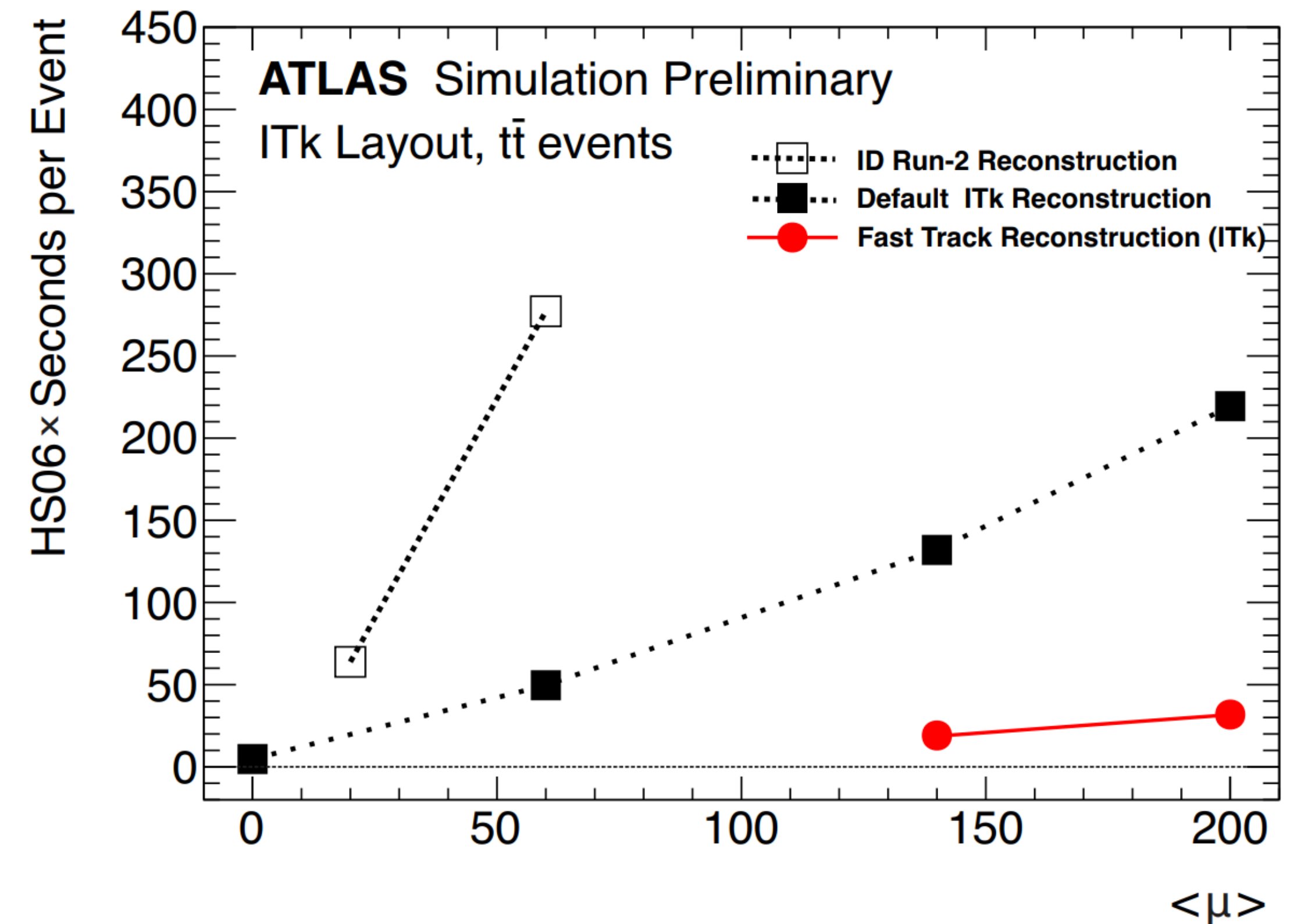
- At Sussex we have the goal to identify the techniques required to enable algorithms to be developed cost effectively both online and offline in computing environments that employ FPGA accelerators, in collaboration with RAL.
 - Find existing algorithms implemented in CPUs and FPGAs with VHDL and compare their performance with the same algorithms deployed in FPGA making use of HLS
 - Hough transform, NN,
 - Document best practices for synthesis and experiences with vendor-specific development and practices
 - Focus on HLS implementations that offer generic solutions for next generation experiments

ATLAS DETECTOR UPGRADE FOR HL-LHC

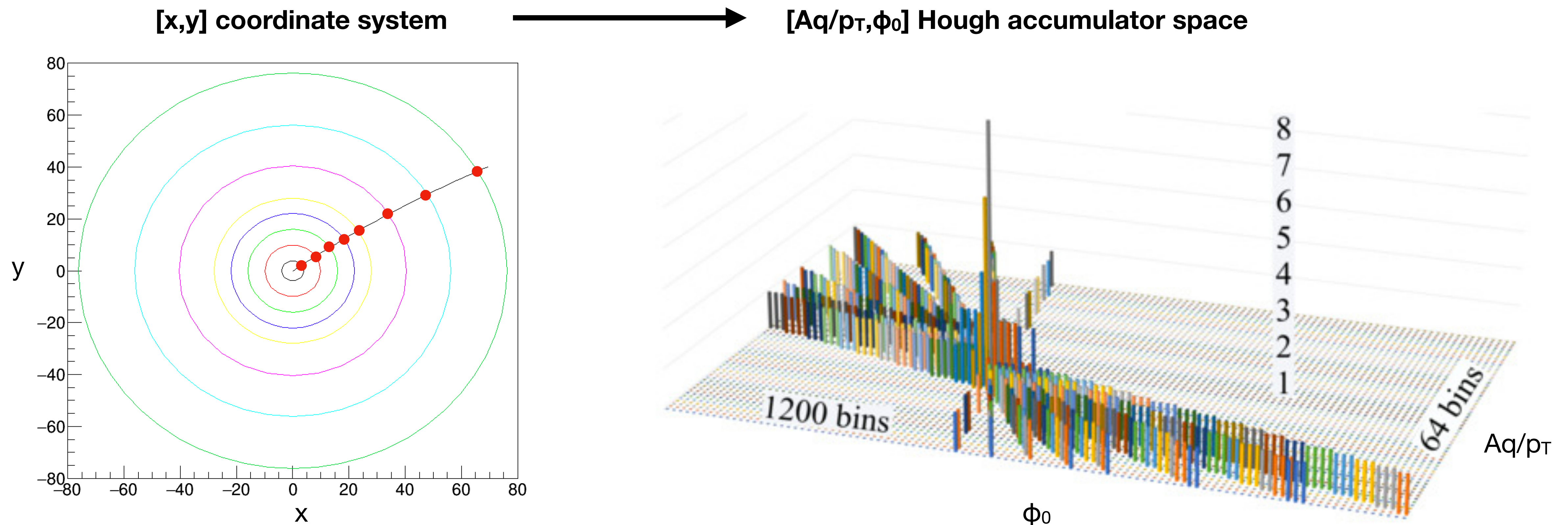
- New Inner Tracker (ITk) detector will provide better coverage for track fitting
- Silicon strip and pixel detector in 2 T magnetic field
 - 4-central strip layers and two end cap with six disks each
 - 5-pixel layer in the central and forward sections
 - Improved coverage up to $|\eta| < 4$



- Software tracking is the baseline solution, with improvements expected from the integration with commercial accelerators
 - Using a fast tracking algorithm based on a combinatorial Kalman Filter seeded on combinations of pixel hits
- In order to optimise the EF farm, several alternative hardware accelerator based solutions are being explored:
 - Pattern recognition via Hough Transform in FPGAs (VHDL and HLS)
 - GPU based regional tracking
 - GNN based tracking on FPGAs
 - ...



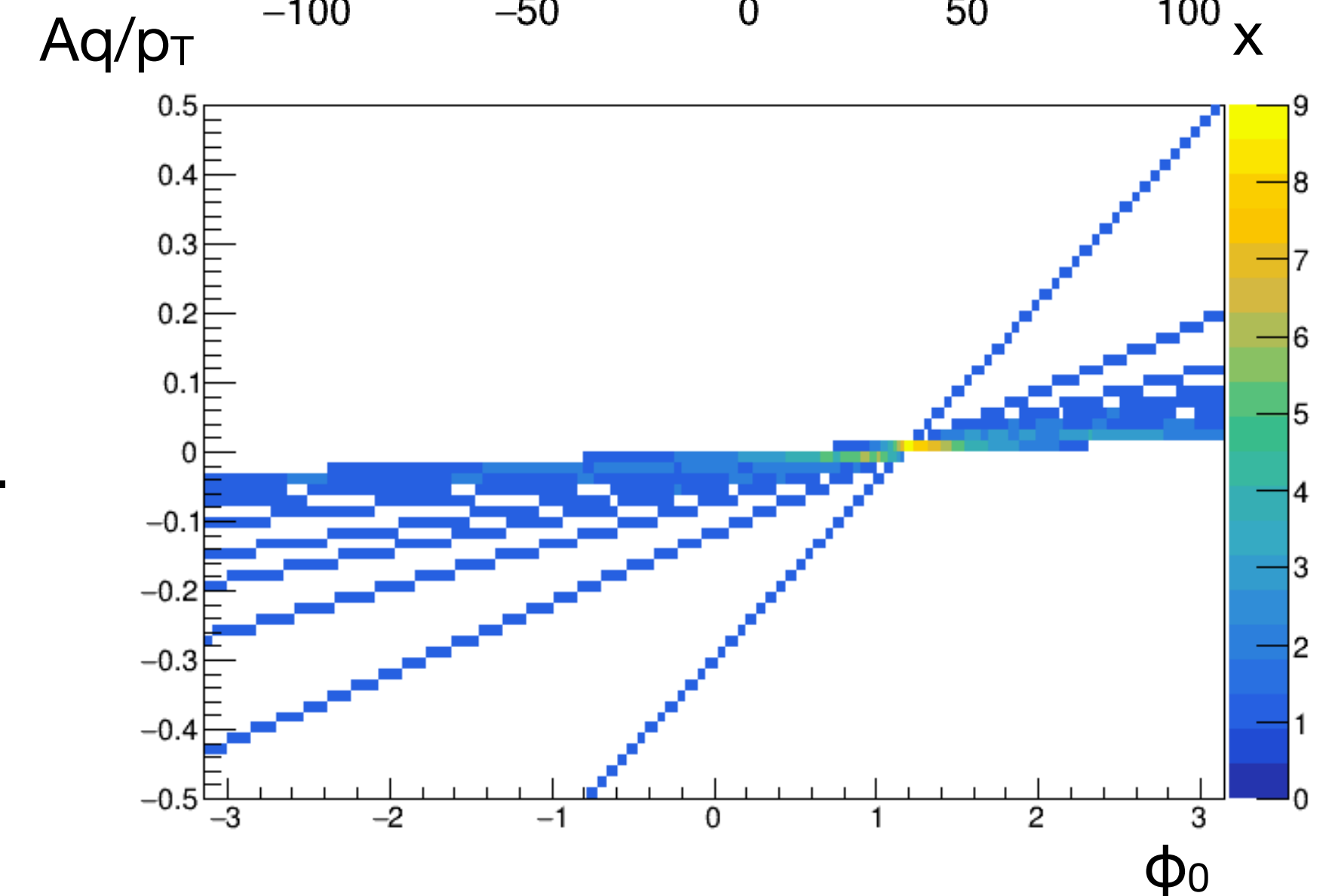
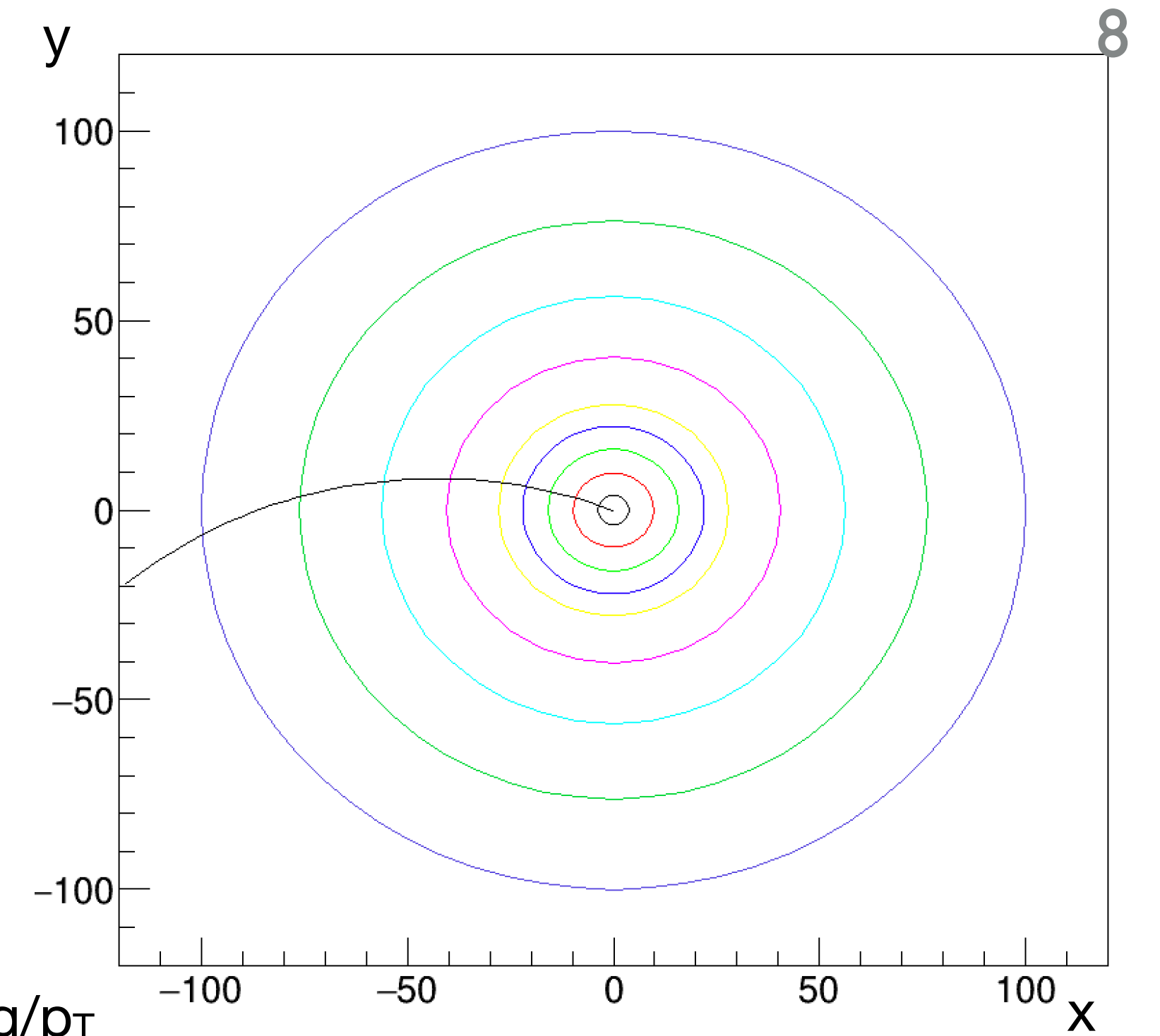
- A VHDL implementation of the Hough transform has been developed by ATLAS
 - Converts detector hit coordinates $[x,y]$ to Hough space $[Aq/p_T, \phi_0]$
 - “Points of accumulation” in the Hough space are used to identify candidate patterns
- A pipelined parallelised HDL implementation optimises the use of the FPGA architecture



- Several alternative Hough Transform FPGA implementations are being explored by ATLAS, broadly falling into two categories:
 - Both targeting Alveo U250-U280 FPGAs using VHDL for a clock speed 250-500 MHz depending on the implementation
- Optimised for parallelisation
 - Minimising latency
- Implementing optimised clustering to preserve efficiency
- Target clock speed of 250 MHz
- Modifies the implementation of Hough Transform via a geometrical approximation for a simplification of the algorithm
 - Involves intrinsic loss in resolution in r
 - Compensated by more detailed z information due to smaller FPGA footprint
 - Pipelined for efficient use FPGA resources with potential for large areas of improvement.
 - Target clock speed of 500 MHz

WORK AT SUSSEX - HOUGH TRANSFORM IMPLEMENTATION ON FPGA WITH HLS

- Current initial focus is “baseline” Hough Transform, possibly re-optimised based on ATLAS studies, implemented on Xilinx U280 FPGAs with HLS
 - Using vendor-specific application tool (Vitis_HLS) provided by Xilinx
 - Allows for documentation of best practices
 - To be used as baseline for comparison with available implementations
- Additional studies of real time data driven machine learning algorithms are ongoing
 - Focused on the determination of pattern recognition operating parameters based on raw detector data
 - e.g. real-time data driven determination of alignment-dependent information (fit and pattern recognition parameters etc.)
 - Primarily carried out by Master students (supervised by Alessandro Cerri and me)



- Have begun working on a twiki page where information can be gathered and centralised
- Allows for:
 - Easy access to general info and detail of different ongoing/completed tasks
 - Clear layout of task distribution and contributions
 - Centralised location for summary of available resources
- Available information:
 - Summary of available machinery
 - Collaborators and corresponding tasks
 - General description of ongoing projects
- Missing information:
 - Detailed description of available algorithms and accessible resources
 - Performance studies

SwiftHep

- ↓ [Introduction](#)
- ↓ [Available Resources](#)
- ↓ [Projects and Status](#)
 - ↓ [WP4. Reconstruction and Trigger](#)
 - ↓ [Deployment and Benchmarking of tracking and vertexing algorithms in FPGA accelerators](#)
 - ↓ [Prototype of tracking and vertexing algorithms on heterogeneous compute platforms](#)
- ↓ [Machinery](#)
- ↓ [Collaborators](#)

MORE CONTRIBUTIONS NEEDED!!!!

Introduction

The Software Infrastructure and Technology for High Energy Physics (SWIFT-HEP) is a UK based project involving several institutes around the UK. The computing requirements and challenges for the next-generation HEP experiments, such as ATLAS, [CMS](#), Alice, and LHCb in the will become increasingly more complex and important with the higher event rates and data volumes expected in the higher luminosity environment of the HL-LHC. The SWIFT-HEP project goal is to study and develop new software and algorithm techniques to develop in parallel to the new hardware being developed for these, and other, next-generation experiments.

Available Resources

- SWIFT-HEP/ExcaliburHep workshop
 - Liv slides : Accelerating Particle tracking at [CMS](#) for the HL-LHC
 - Mario Slides : HLS developments and plans at Sussex

Projects and Status

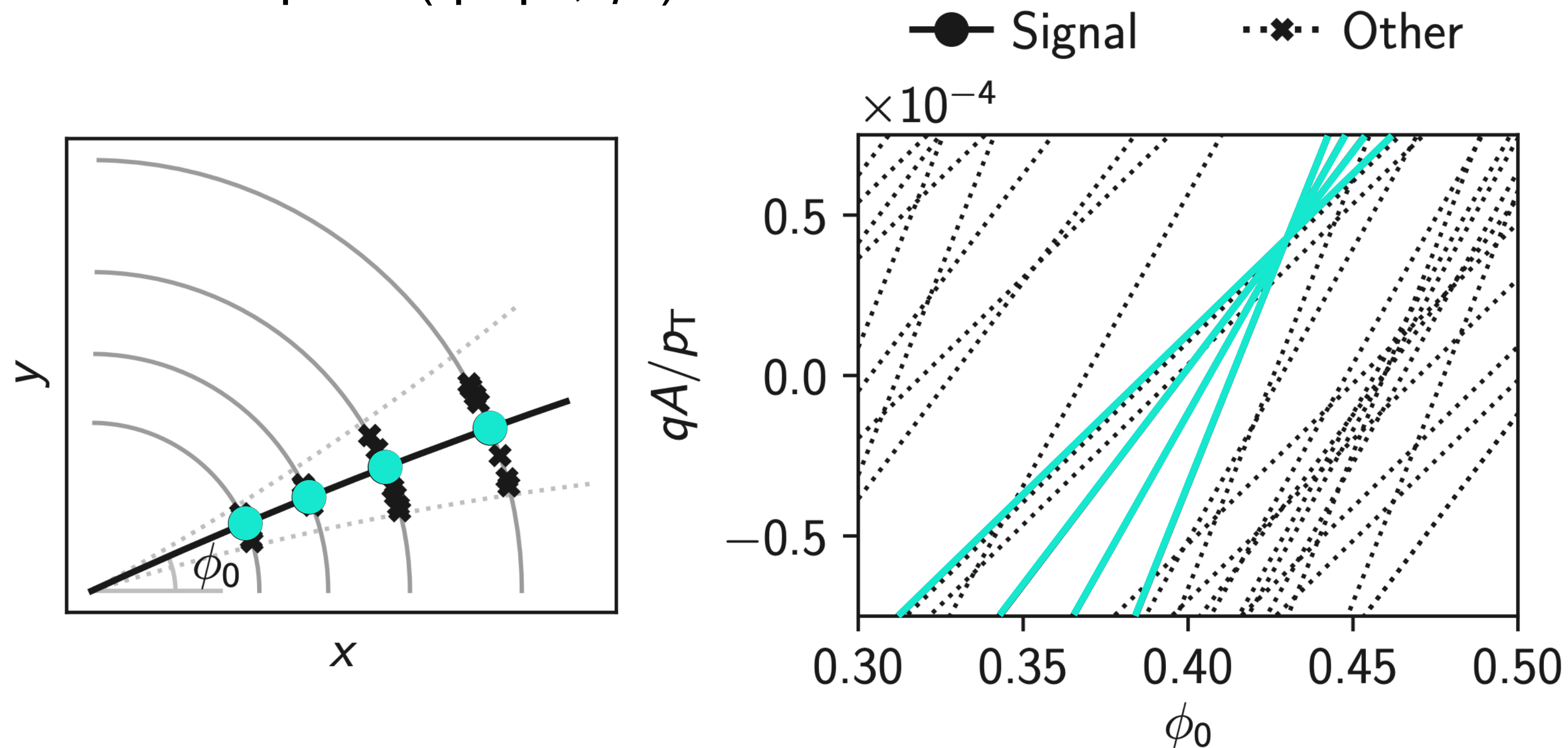
- The existing experience in developing the RTL Hough Transform provides a starting point to compare different solutions and identify/develop benchmarks
 - The existing solution is focused on addressing pattern recognition in high-occupancy and multiplicity environments
- I will work on identifying, documenting and - if needed developing - tools, procedures and benchmarks for tracking and vertexing
 - Started collecting information and documenting ATLAS and CMS contributions.
Please do get in touch so that we can include more details and solutions!
 - This documentation and available benchmarks are a starting point
 - We need your help to integrate and complement as needed!
 - Once the tools, techniques and best practices higher level language programming have been documented, they may be useful to the study of other algorithms performances.

- ATLAS baseline software tracking algorithm can benefit from implementation of hardware accelerator based solutions
 - Several different avenues are being explored to achieve highest possible efficiency with best performance
- Created Twiki page to gather relevant information
 - This platform can be used to centralise resources and results, and guide the project forward
 - ***More information is needed and always welcome!!!***
- Sussex will contribute to the SWIFT-HEP goals for the deployment and benchmarking of tracking and vertexing algorithms in FPGA accelerators
 - Beginning from existing Hough Transform algorithms developed in RTL on FPGAs as benchmark to develop techniques and tools needed to compare and study the achievable performances of available algorithms
 - Started implementing Hough Transform algorithm (based on ATLAS version) on FPGA using HLS
 - Using Xilinx development tools



BACK-UP

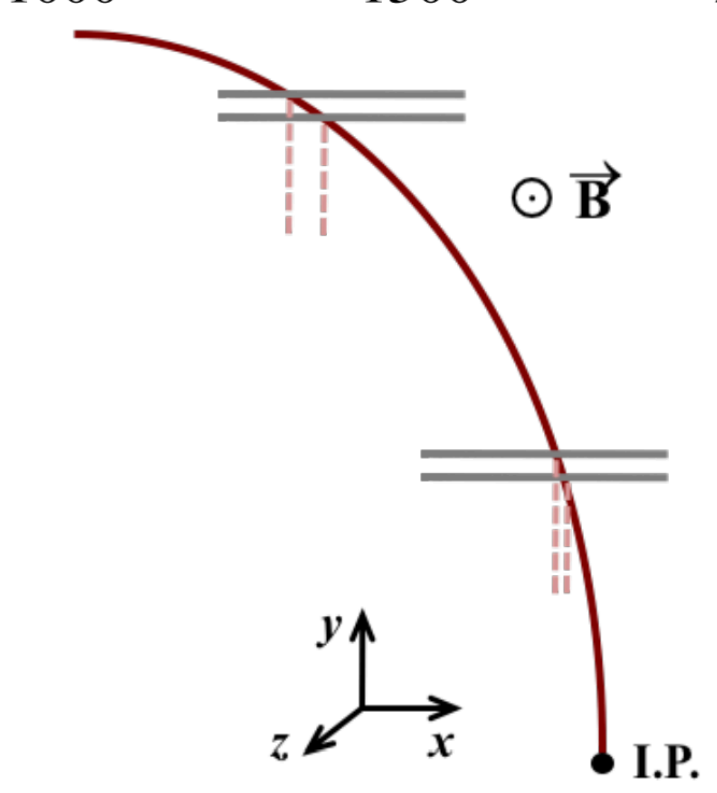
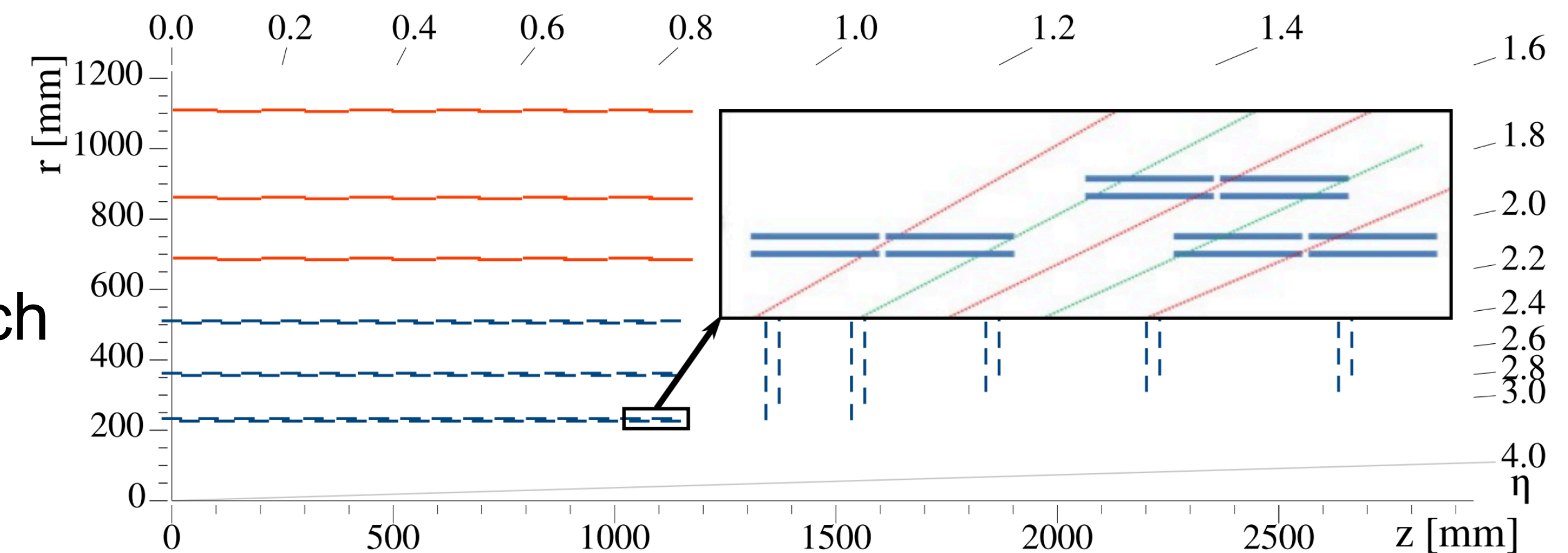
- Will be focusing firstly on Hough transform track reconstruction algorithm
 - Used to project cluster positions in a detector tracker in an (x, y) transverse plane to a curve in the track parameter space $(qA/p_T, \phi_0)$ called **accumulator**

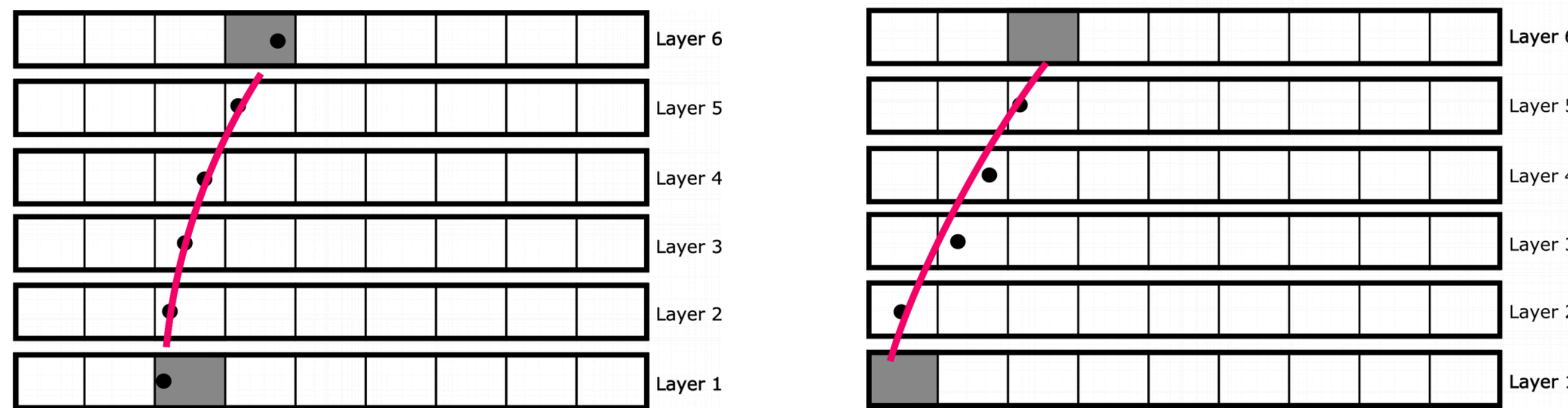
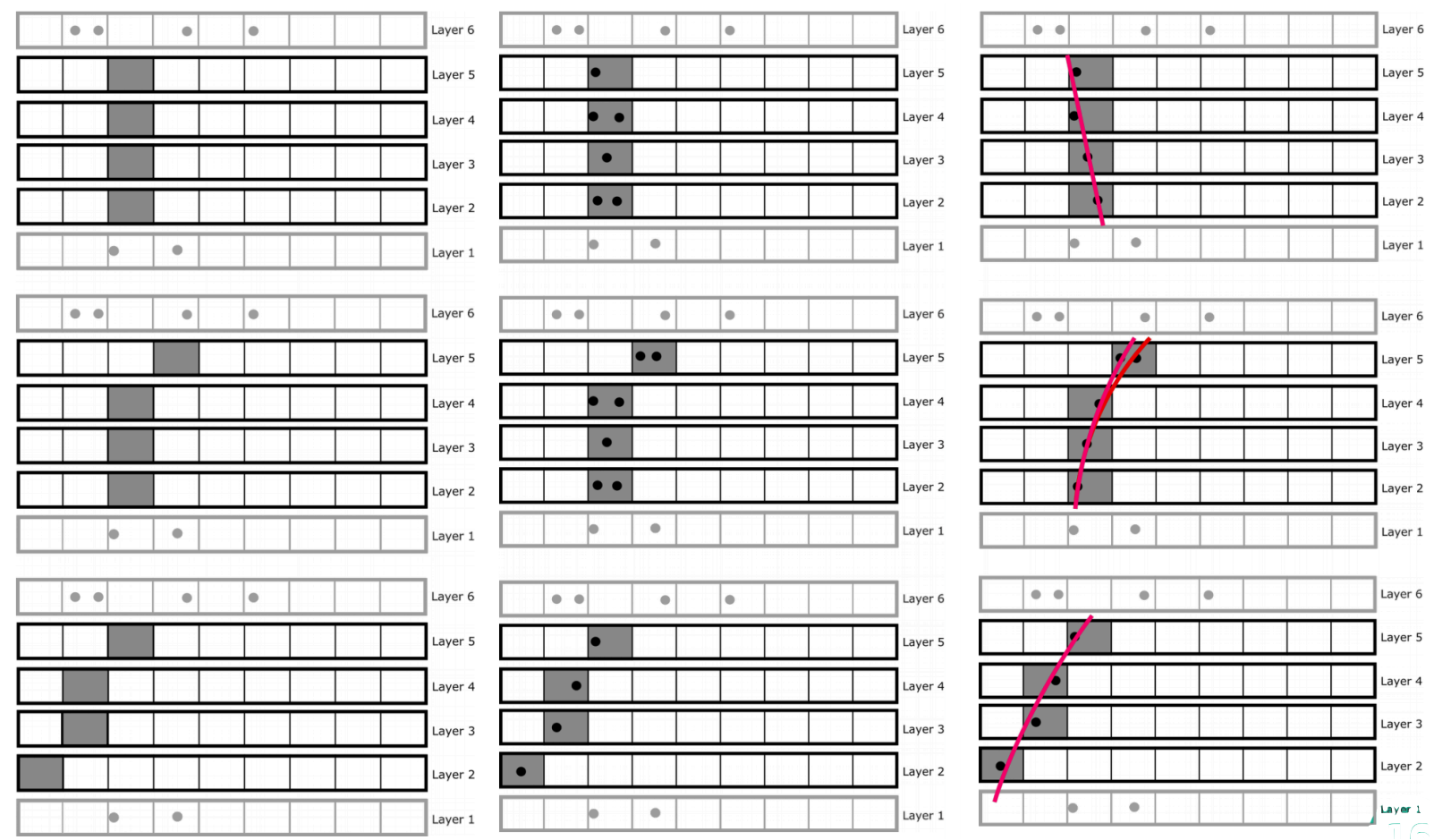
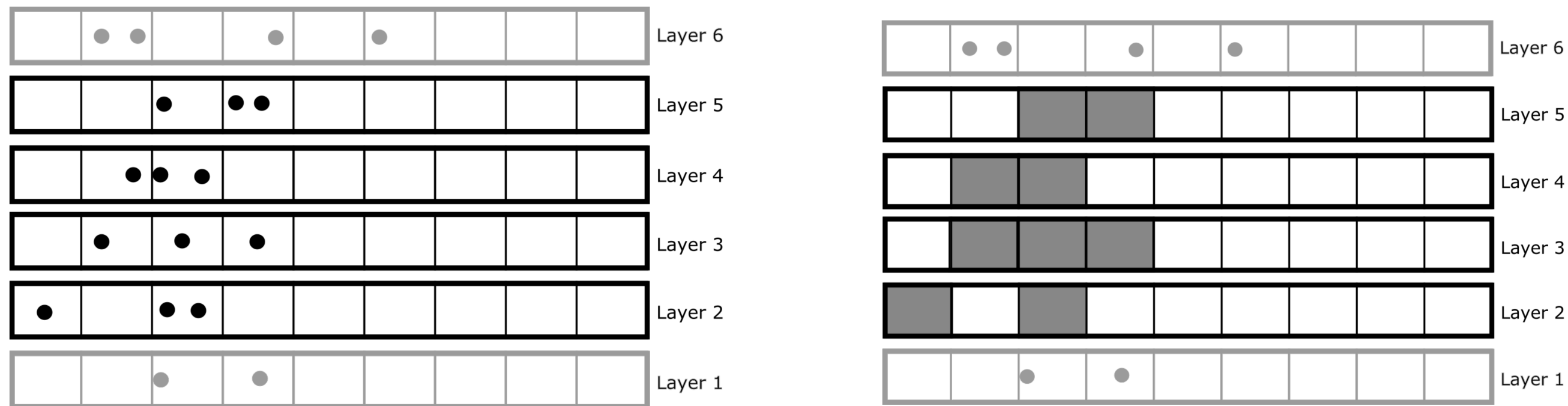


- Will be followed by Kalman filter, and Machine learning algorithm to reconstruct vertices from reconstructed tracks

- Tracker will be replaced with a new tracker made up of modules composed of two layers which will provide improved tracking efficiency
 - Tracking will be performed on vector information rather than individual hits

- Three methods have been developed
 - An ASIC+FPGA approach
 - An all-FPGA hough transform approach
 - An all-FPGA road search algorithm approach
- The three methods have been merged to develop a single system design based on commercial FPGAs
 - Will enable joint exploration of tracking algorithms to optimise the final system performance





- From a cluster of hits in the pixel strips define super-strips (coarser resolution strips) for the inner layers
- Find super-strip for each cluster
- Pattern match super-strips
- Get full resolution clusters within the pattern matched super-strips
- Fit all possible combinations of clusters and remove bad patterns via χ^2 cuts
- Remove duplicates
- Finally extrapolate to super-strips into remaining layers, recalculate χ^2 and remove bad patterns
- Remaining patterns will populate the pattern bank used to identify patterns online