Data Concentrator for Belle II DEPFET Pixel Detector

Michael Schnell
schnell@physik.uni-bonn.de

supervised by
Jochen Dingfelder and Carlos Mariñas

University of Bonn

December, 3rd 2012

universitätbonn

Content

1. Introduction to SuperB factories
2. Overview of the Data Concentrator System
3. Data Acquisition
4. FPGA-based Tracking

Content

1 Introduction to SuperB factories

2 Overview of the Data Concentrator System

3 Data Acquisition

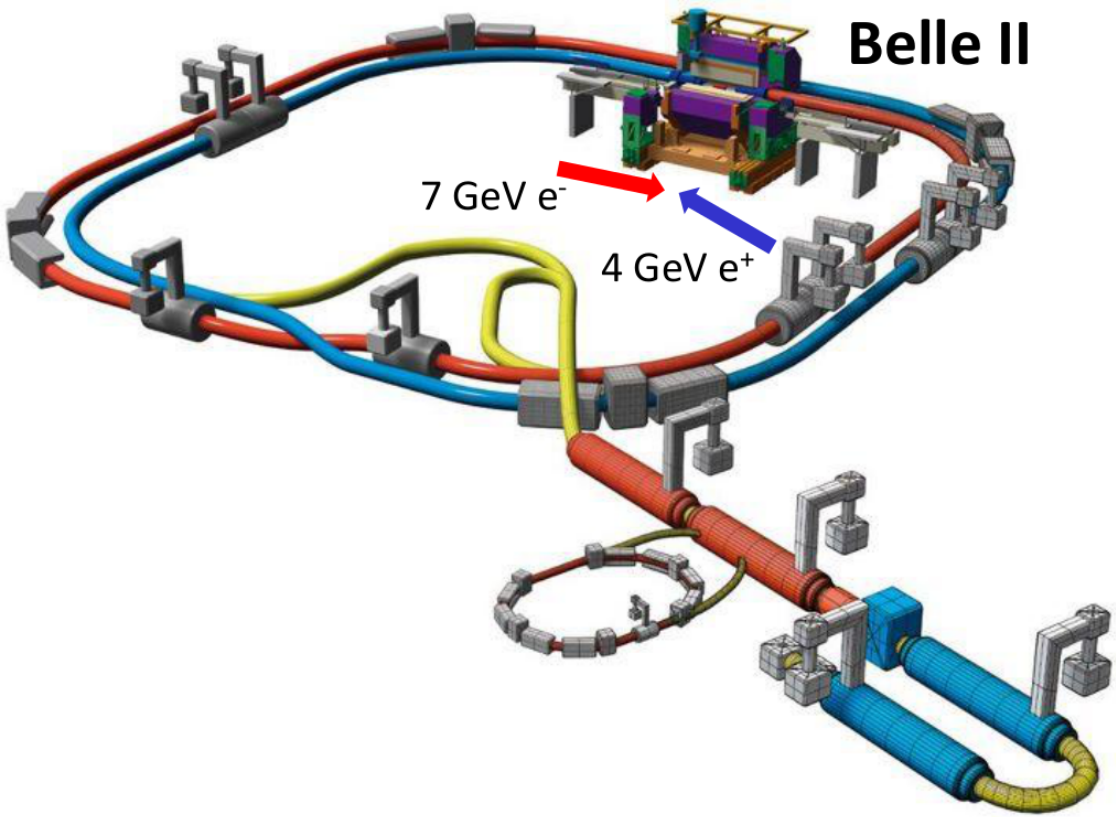
4 FPGA-based Tracking

KEKB

KEKB accelerator facility, located in Tsukuba, Japan:

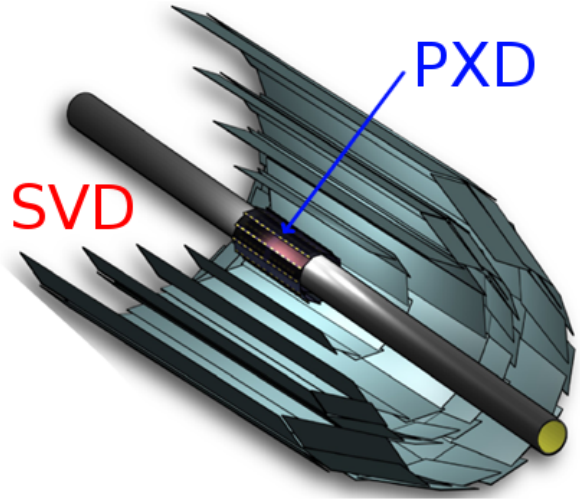
Overview of SuperKEKB

- Upgrade of KEKB
- Asymmetric energy collider with 7 *GeV* for e^- and 4 *GeV* for e^+
- 40 times more data
- Search for new physics in B-meson decays

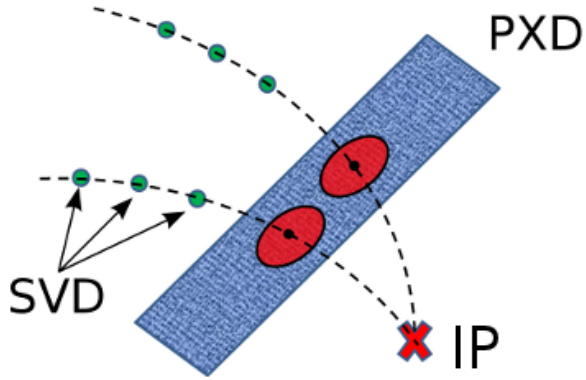


Introduction to the Vertex Detector

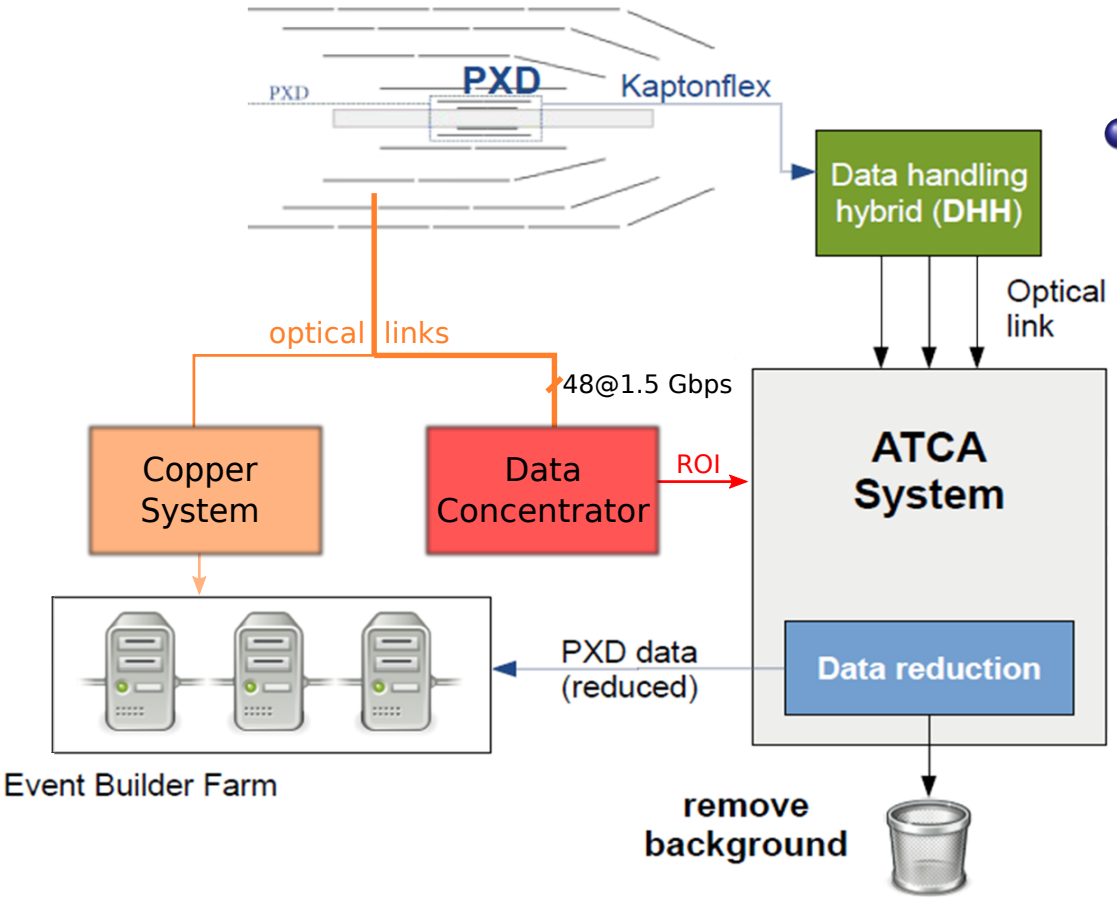
- Reduce the high data rate (60 Gbps) from 8 million pixels of the DEPFET Pixel Detector (PXD)
  → Goal: factor of 10 in reduction



- Idea for Data reduction:
  - Reconstruct tracks in Silicon Vertex Detector (SVD) - extrapolate to PXD - create so called Region of Interests (ROI)
  - Saving information contained on the pixels lying in the ROI



Layout



- Major tasks of the Data Concentrator (DatCon):
  1. Acquire the data from the SVD
  2. Reconstruct the track segments and create the ROIs

Content

1 Introduction to SuperB factories

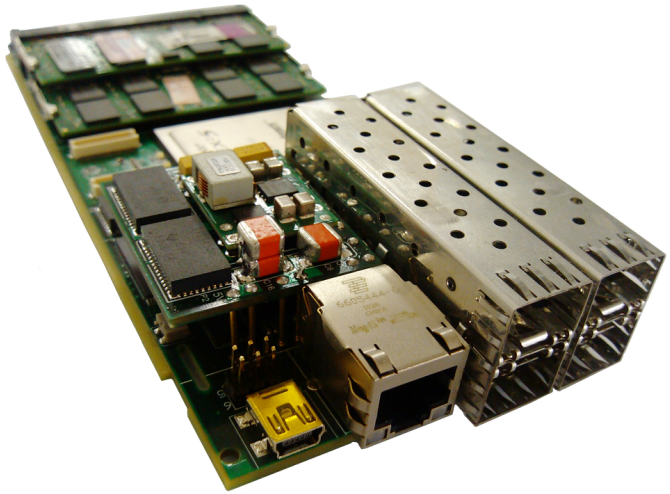
2 Overview of the Data Concentrator System

3 Data Acquisition

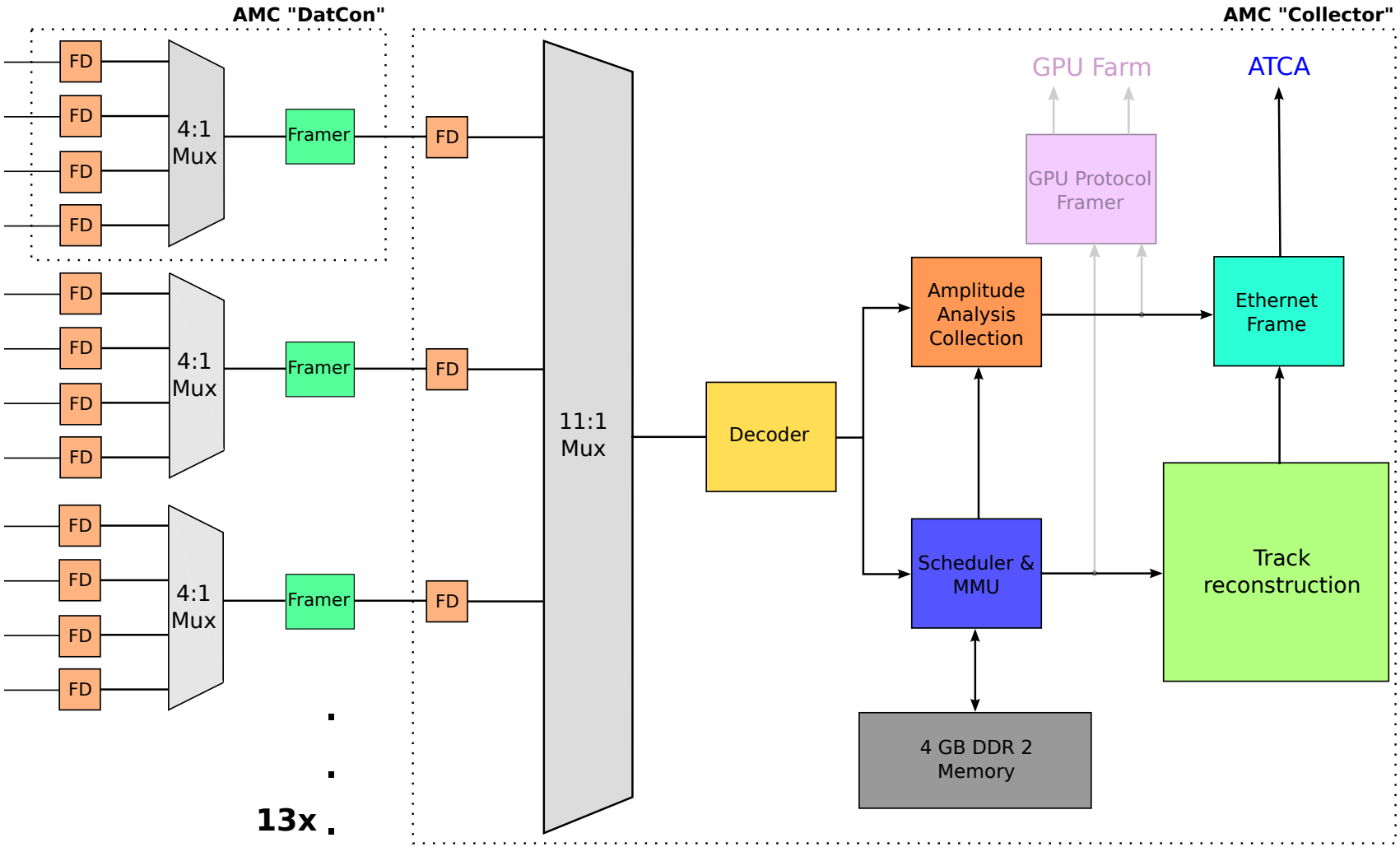
4 FPGA-based Tracking

Hardware Solution for Data Concentrator

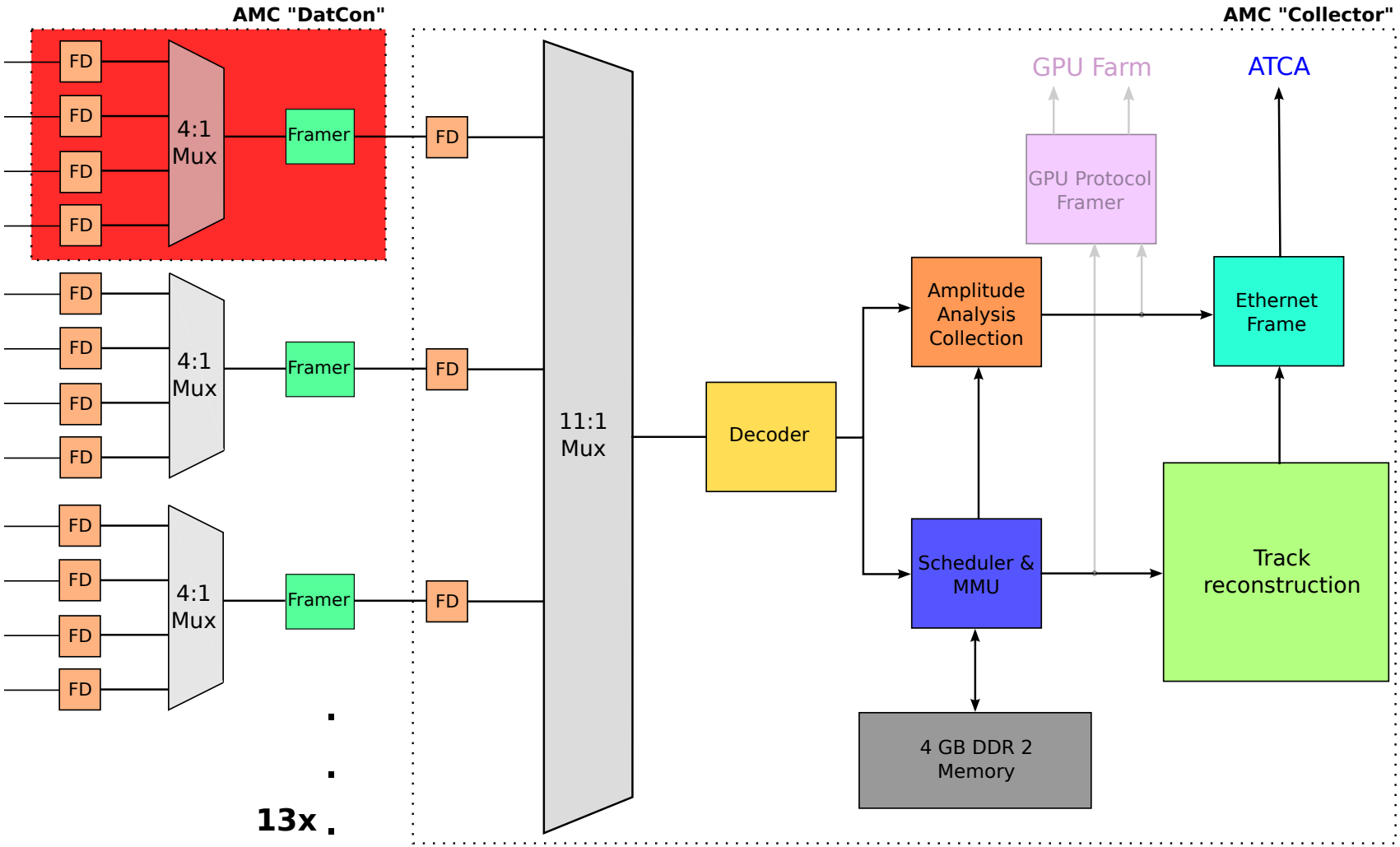
- Advanced Mezzanine Card (AMC)
  - 13 AMCs with 4 SFP+ connectors and Virtex 5
  - 1 AMC with 2 SFP+ for track reconstruction with a high performance Virtex 5
  - 4 Carrier Board with backplane communication



Simplified Block Diagram



Simplified Block Diagram



Content

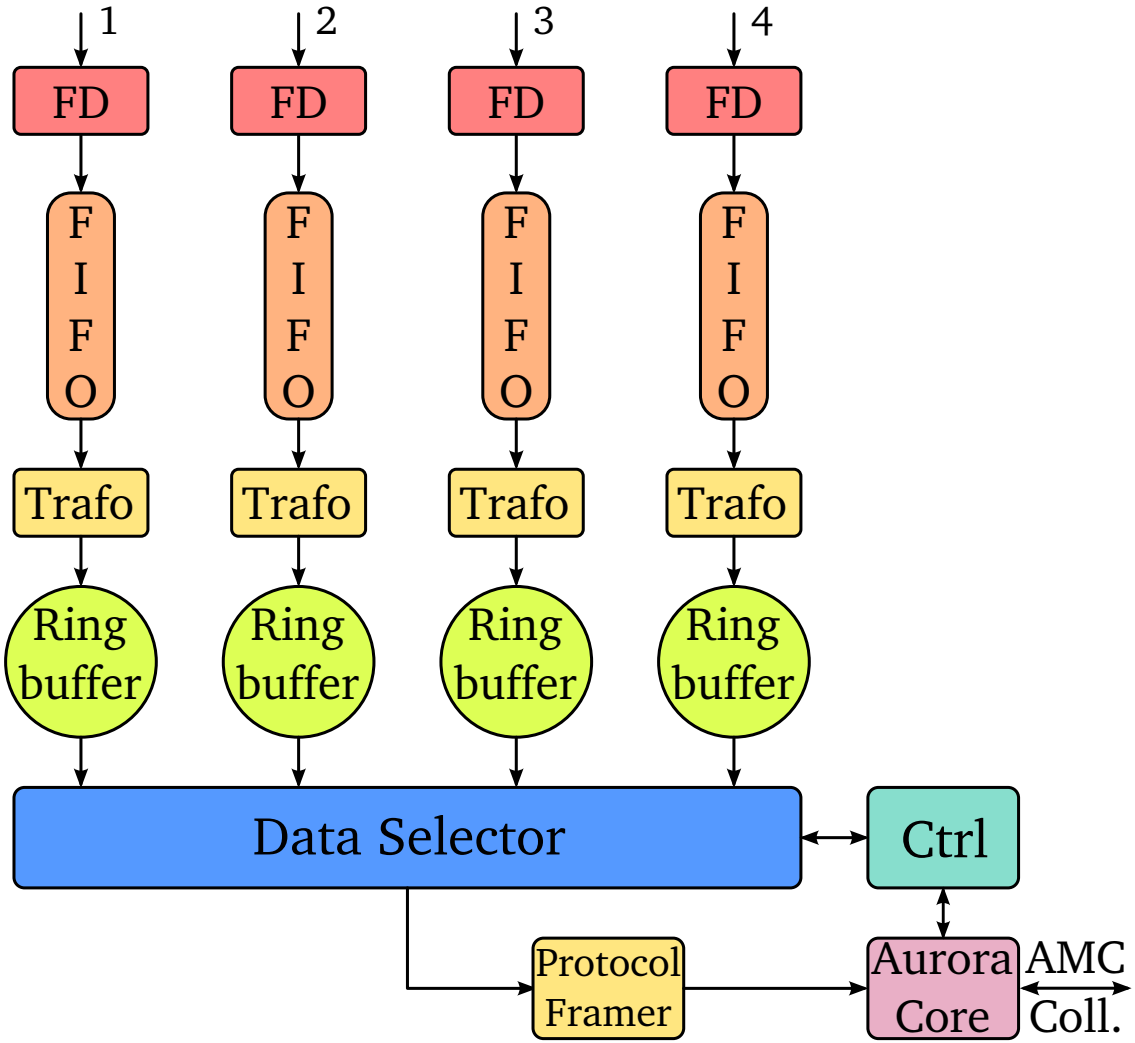
1 Introduction to SuperB factories

2 Overview of the Data Concentrator System

3 Data Acquisition

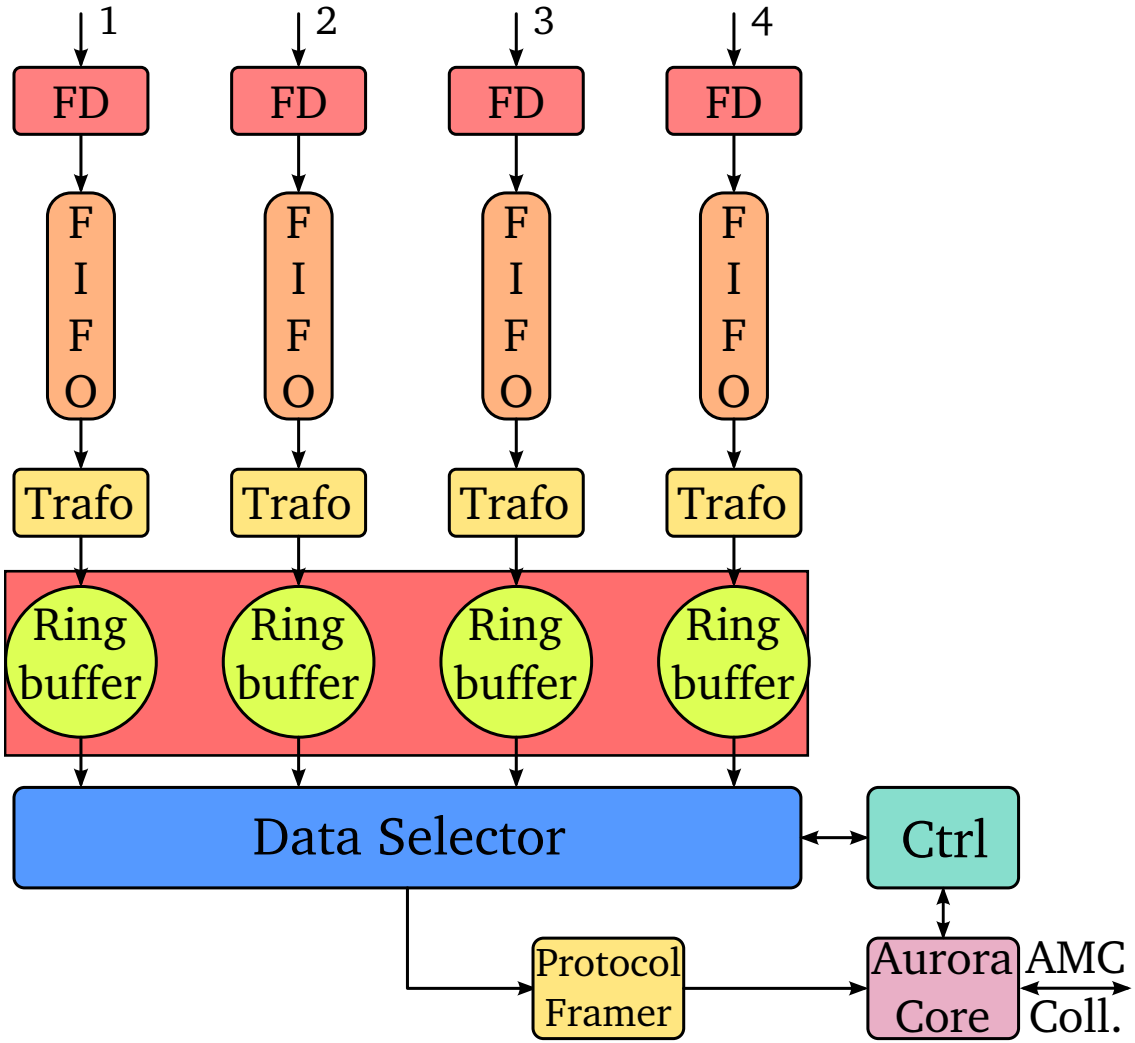
4 FPGA-based Tracking

AMC "DatCon" Layout



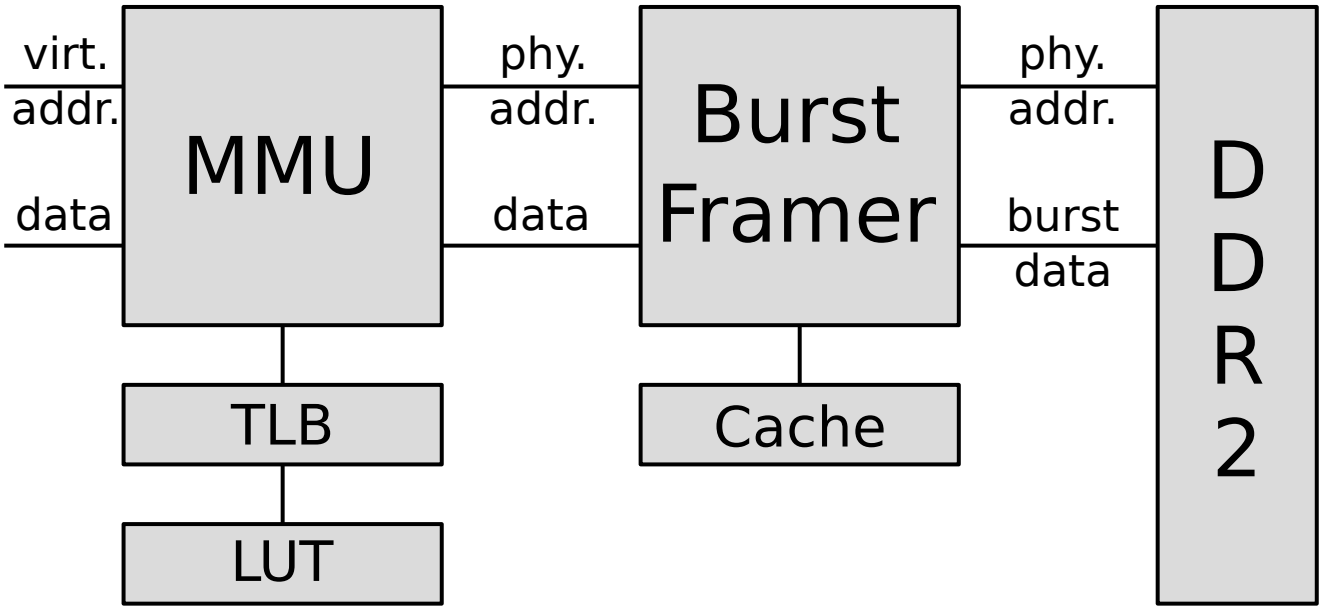
- FD: Frame Detection Unit
- Trafo: Transformation on incoming data, like coordinate translation...
- Ring buffer: Implemented in DDR2 memory.
- Ctrl: Control unit, sends data when next event is requested by collecting FPGA

AMC "DatCon" Layout



- FD: Frame Detection Unit
- Trafo: Transformation on incoming data, like coordinate translation...
- Ring buffer: Implemented in DDR2 memory.
- Ctrl: Control unit, sends data when next event is requested by collecting FPGA

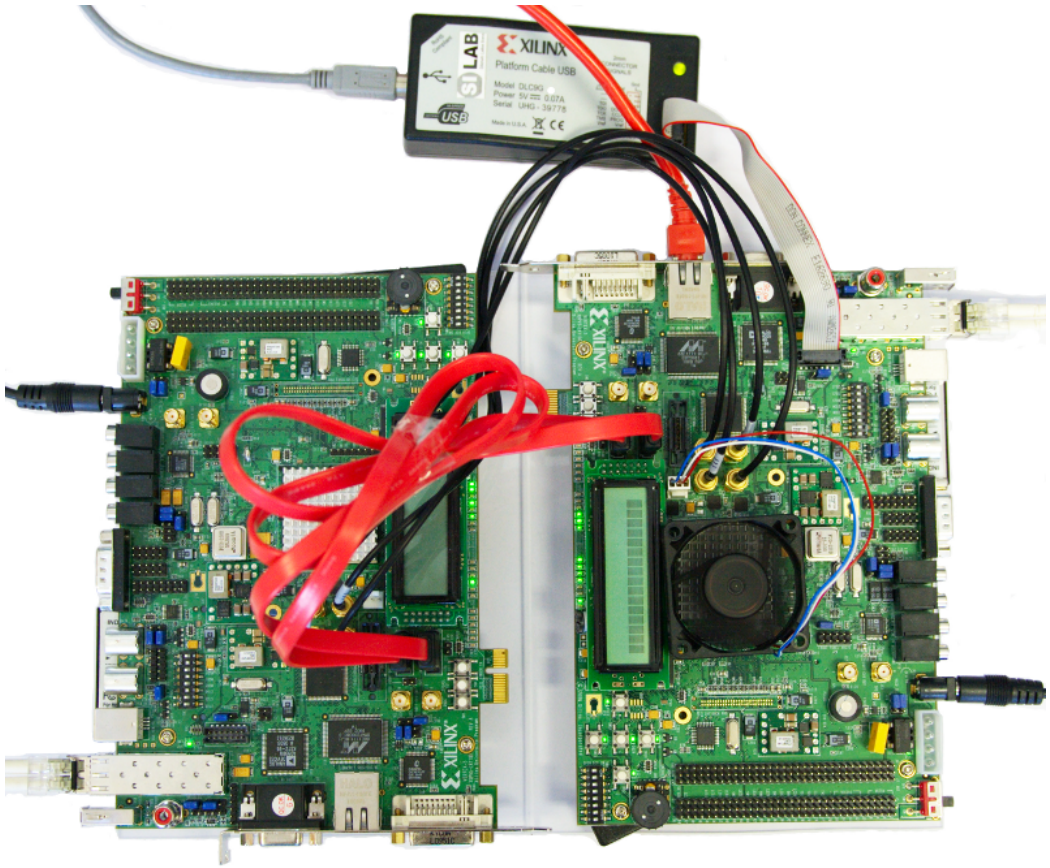
Memory Management Unit (MMU)



- Memory Management Unit for easy access to DDR2 memory
- Simplifies implementation for ring buffer via memory mapping and handle concurrent writes
- TLB, Cache not necessarily required

Test Setup for the AMC DatCon Cards

- ML505 equipped with Virtex 5 used to test multiplexer, protocol and data handling
- Data Generator (left)
- Receiver (right): Running the protocol decoder, multiplexer...
- Algorithms and Design verified and tested through pattern and long term bit error test



→ Ran stable over 37 days!

Content

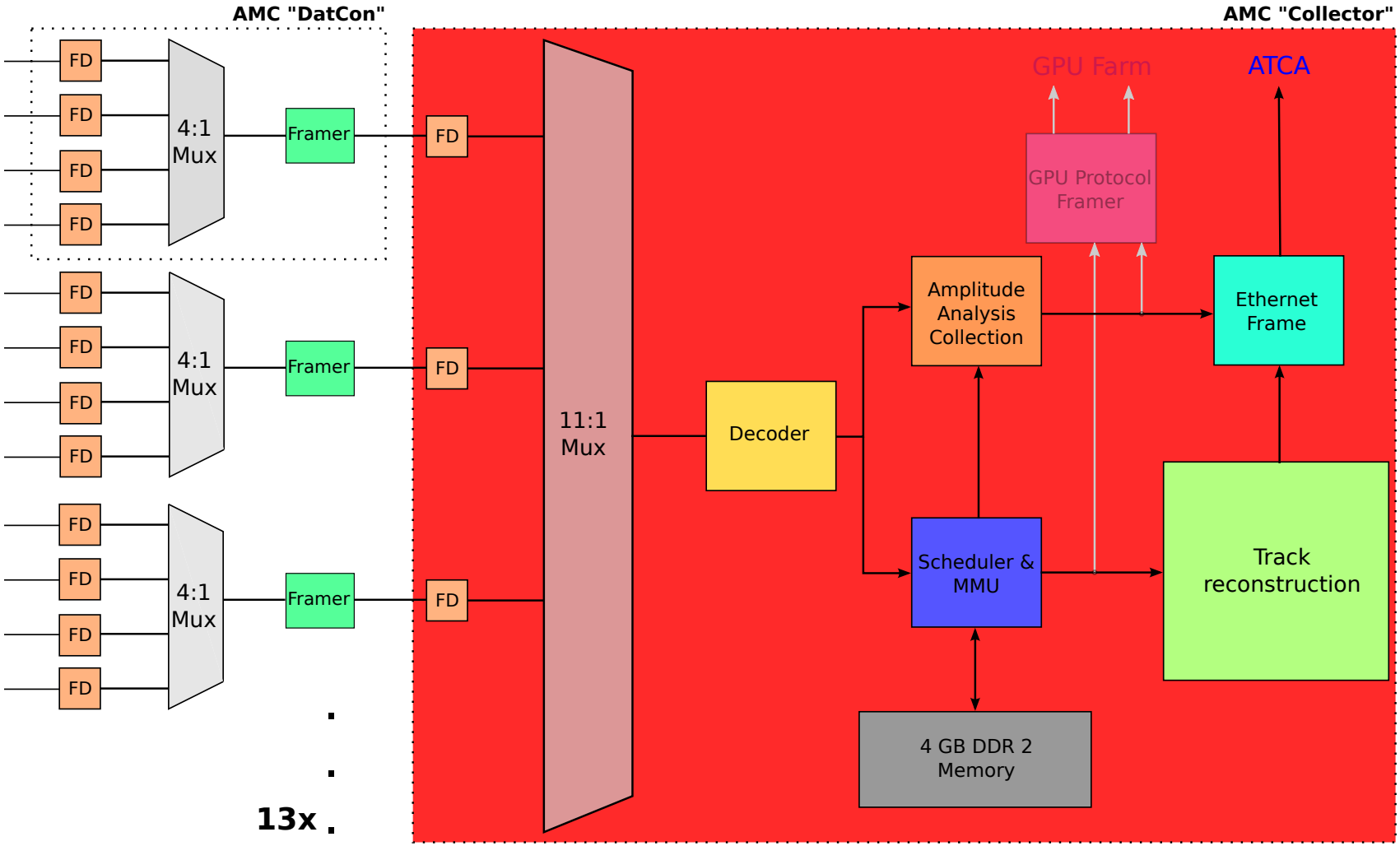
1 Introduction to SuperB factories

2 Overview of the Data Concentrator System

3 Data Acquisition

4 FPGA-based Tracking

Simplified Block Diagram



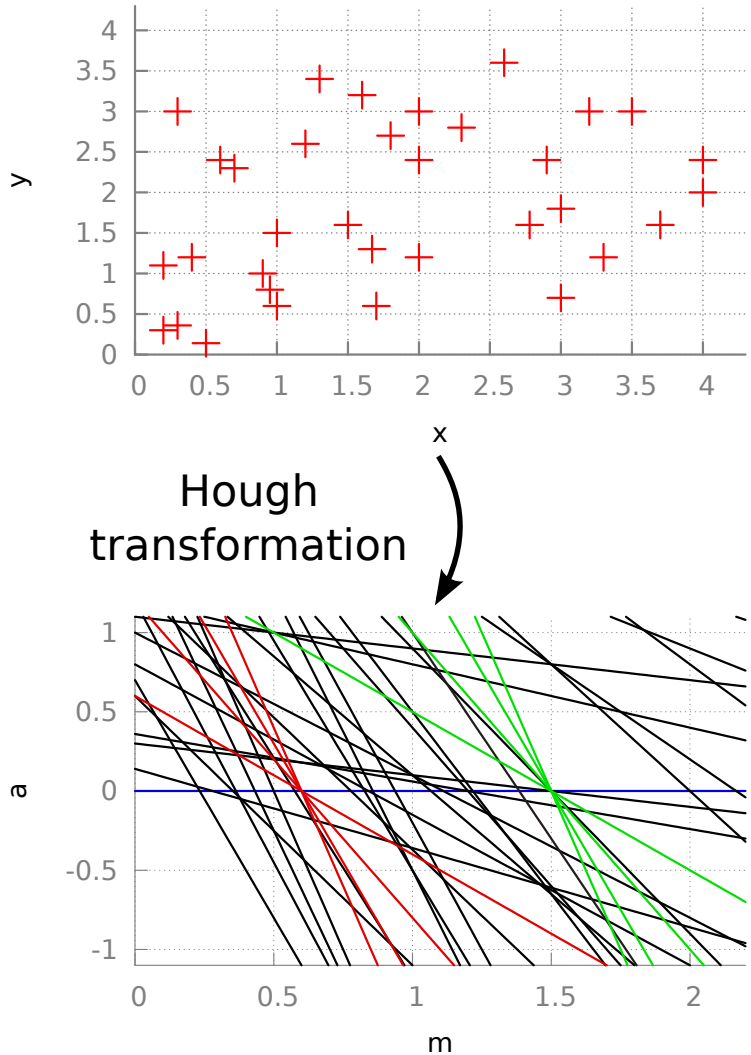
Hough Transformation Basics

- Data Concentrator: Tracking in FPGA using Hough Transformation
- Hough Transformation to find straight and arc tracks (after conformal transformation)

Example straight tracks

$$y_i = m \cdot x_i + a \xrightarrow[trafo]{Hough} a = -m \cdot x_i + y_i$$

- Implemented in the FPGA over Fast Hough Transformation



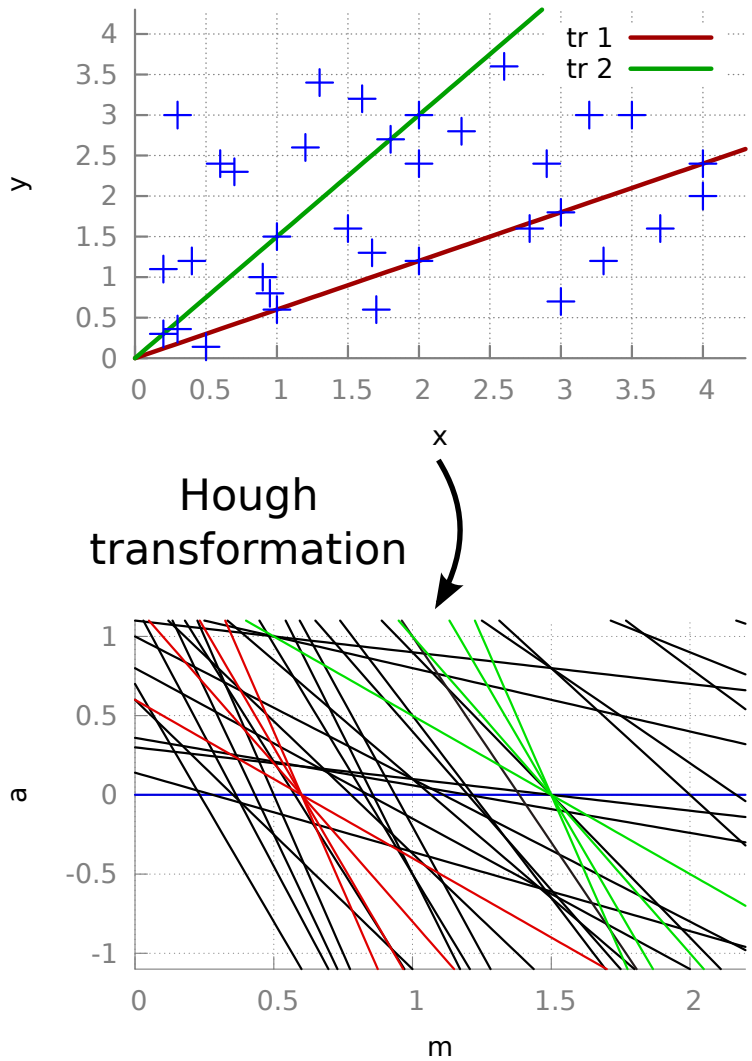
Hough Transformation Basics

- Data Concentrator: Tracking in FPGA using Hough Transformation
- Hough Transformation to find straight and arc tracks (after conformal transformation)

Example straight tracks

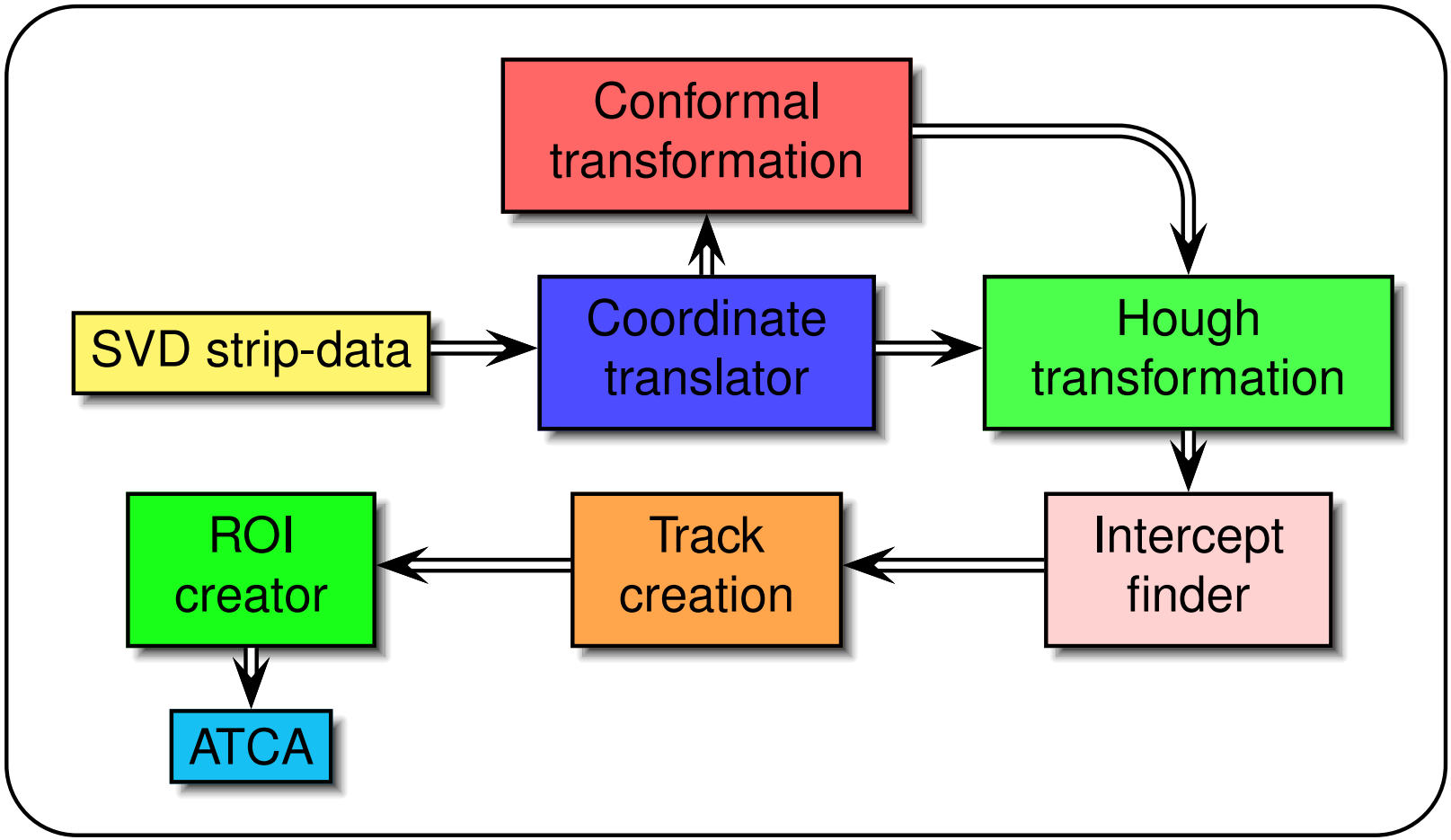
$$y_i = m \cdot x_i + a \xrightarrow[trafo]{Hough} a = -m \cdot x_i + y_i$$

- Implemented in the FPGA over Fast Hough Transformation



Data Flow Diagram Plan

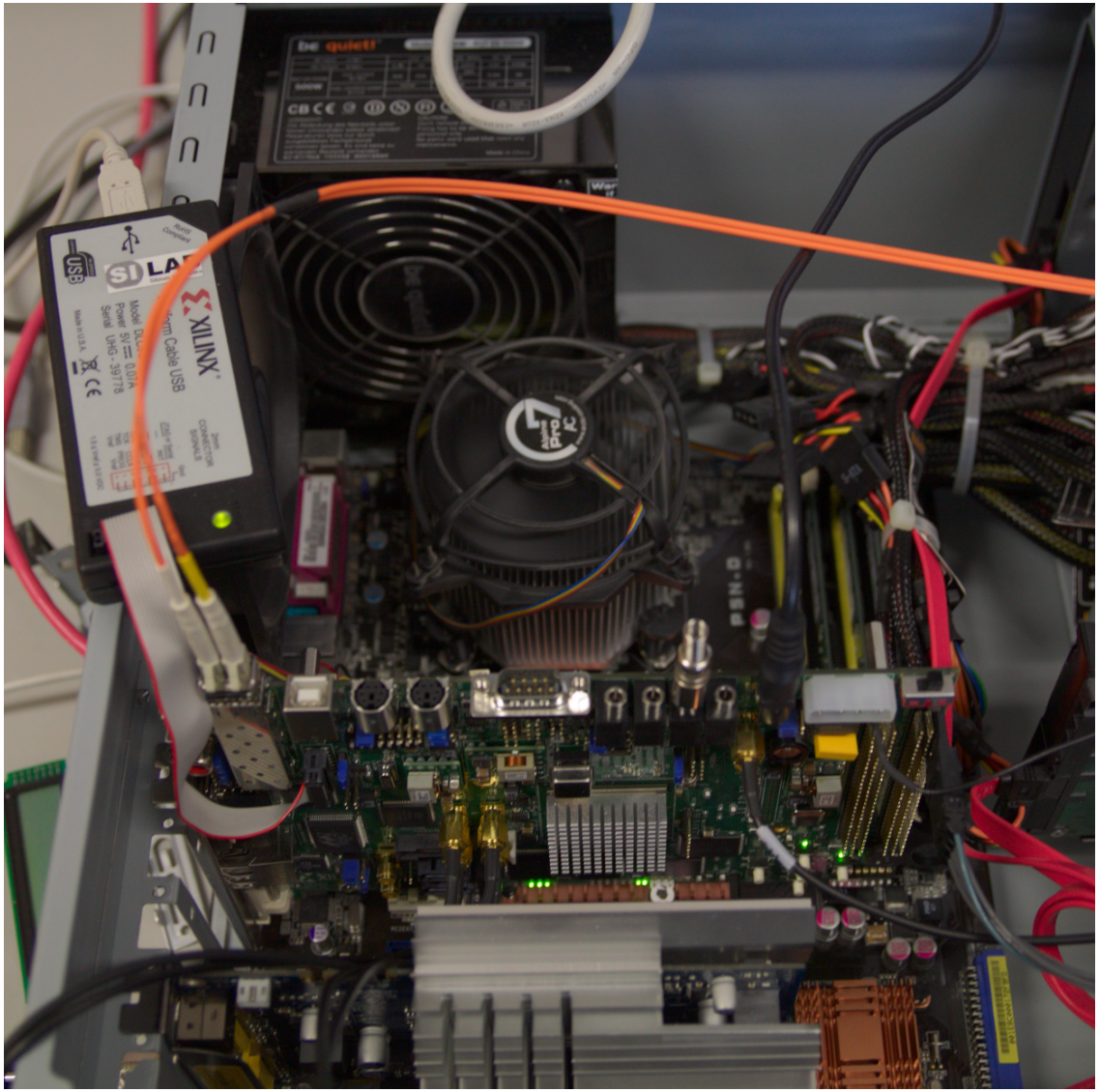
General tracking module interconnection (can also be distributed)

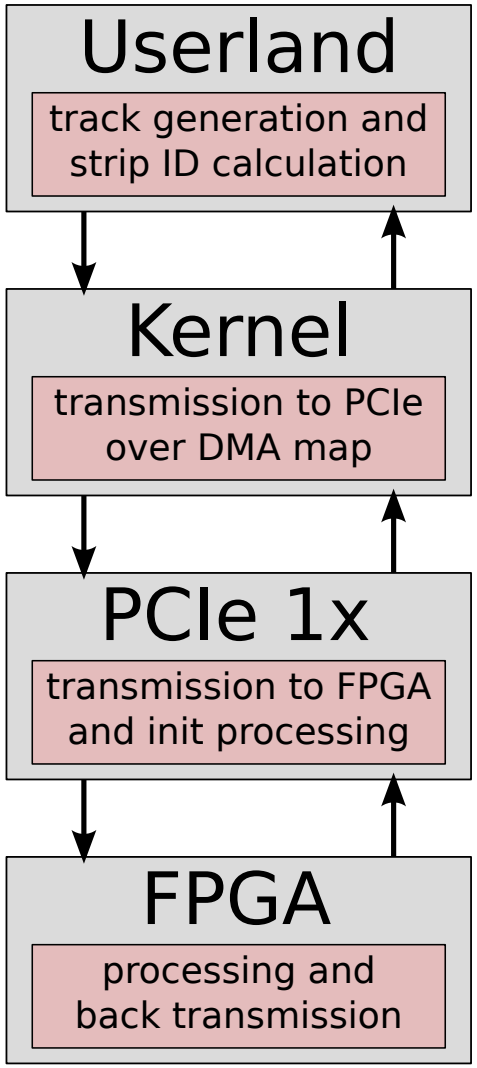


Operations needed

- Need trigonometric functions like sine and cosine e.g. for coordinate transformation
- CORDIC: Basic Lookup table for nodes and a shift-add implementation of approximation function
  Pure LUT: Large Lookup table for every value (high memory requirements)
- FPGAs' DSPs used as Multiplier/Divider unit.
- Block Memory needed to translate strip IDs into coordinates and ROI creation

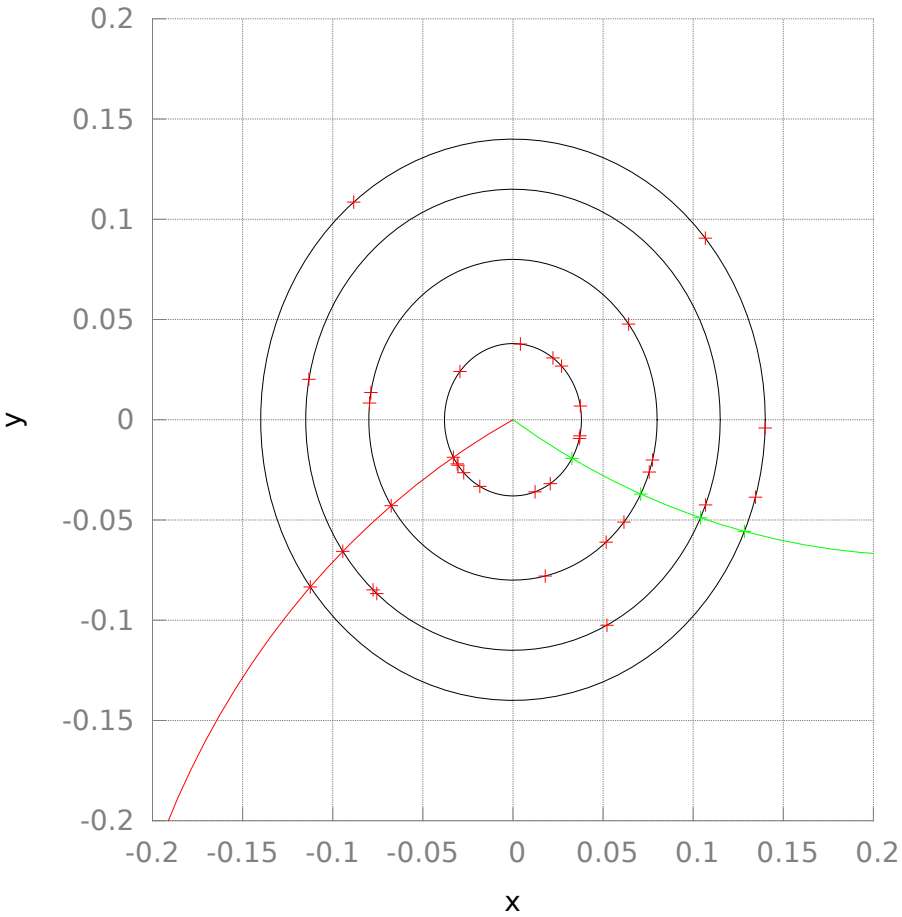
General Testing Platform

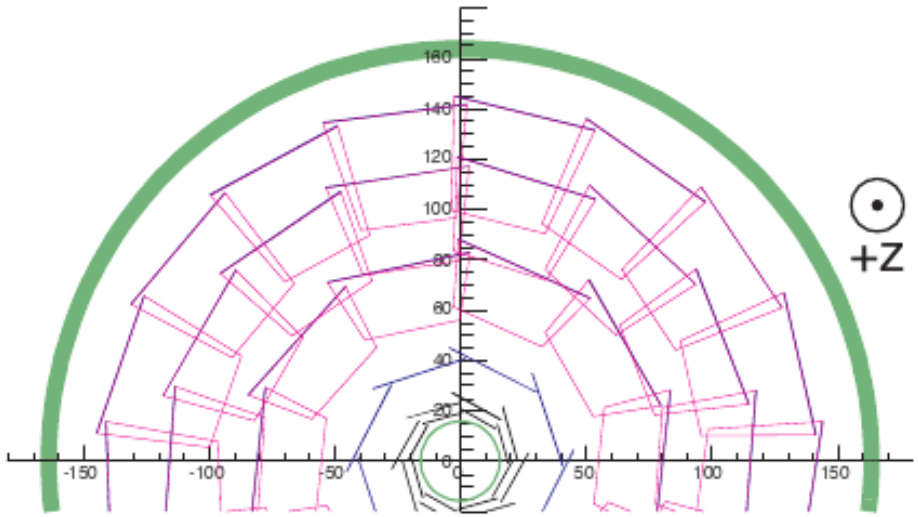




Fast Hough Transformation Example

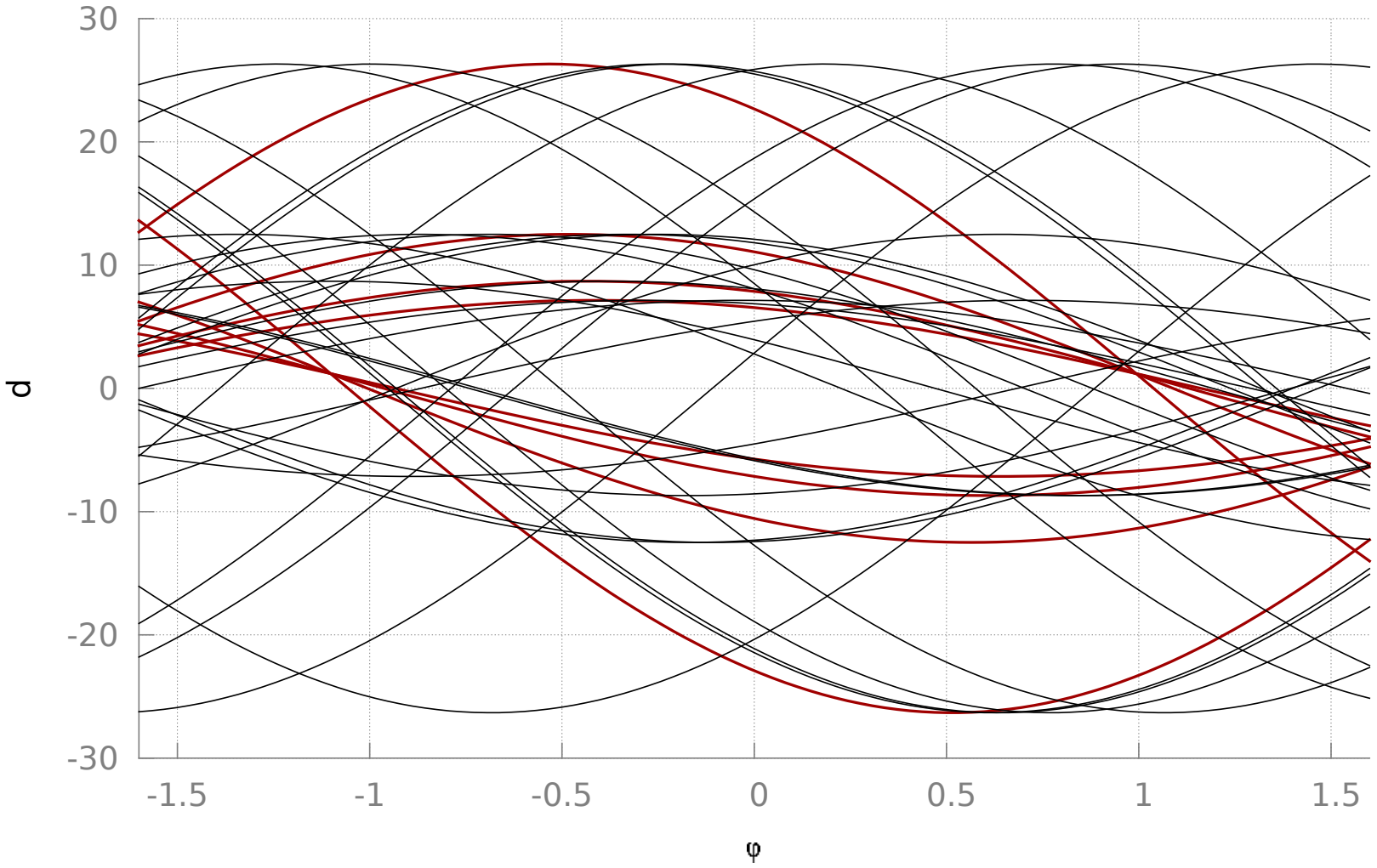
30 background hits and 2 tracks 2D example





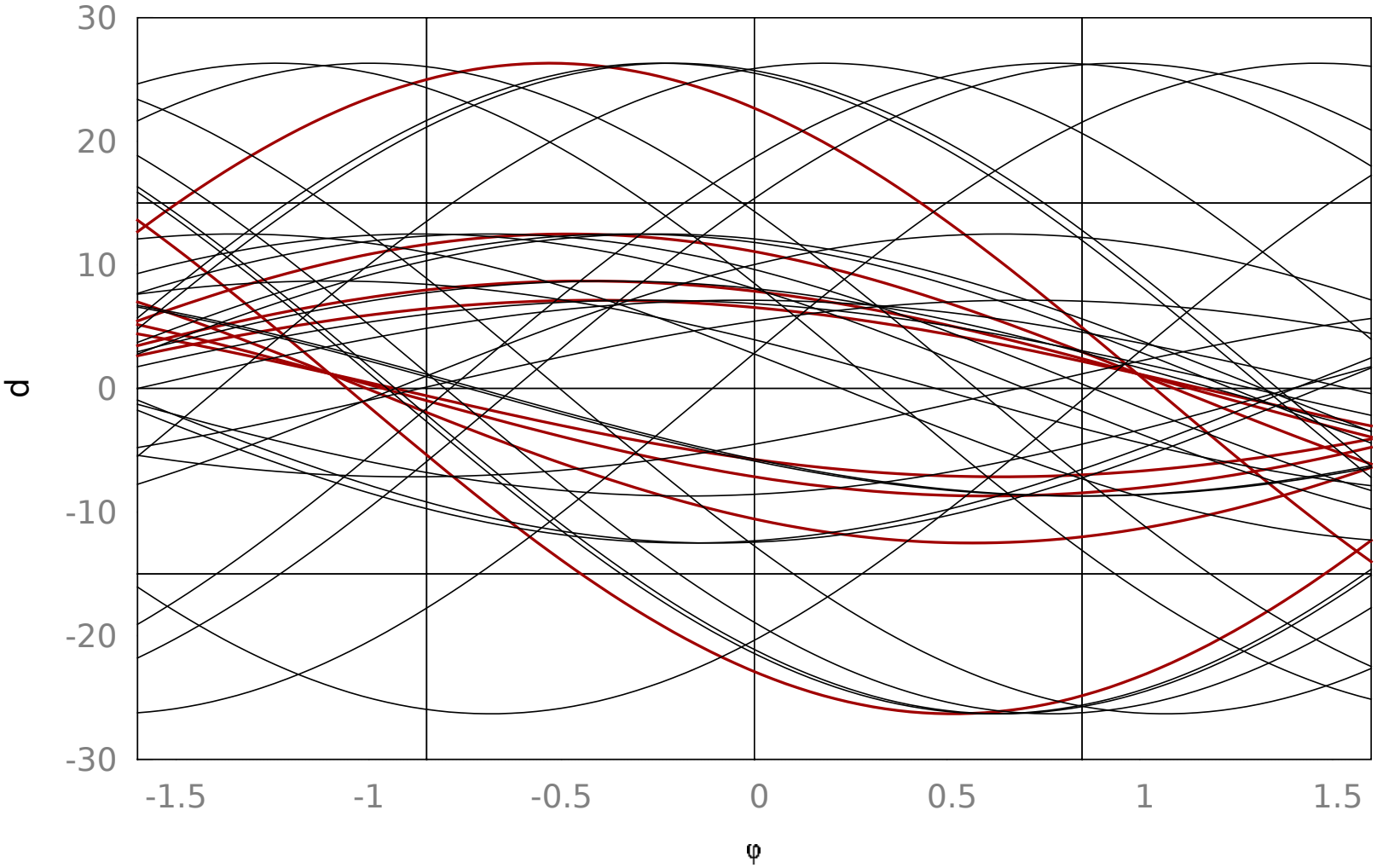
Fast Hough Transformation Example

30 background hits and 2 tracks in Hough space



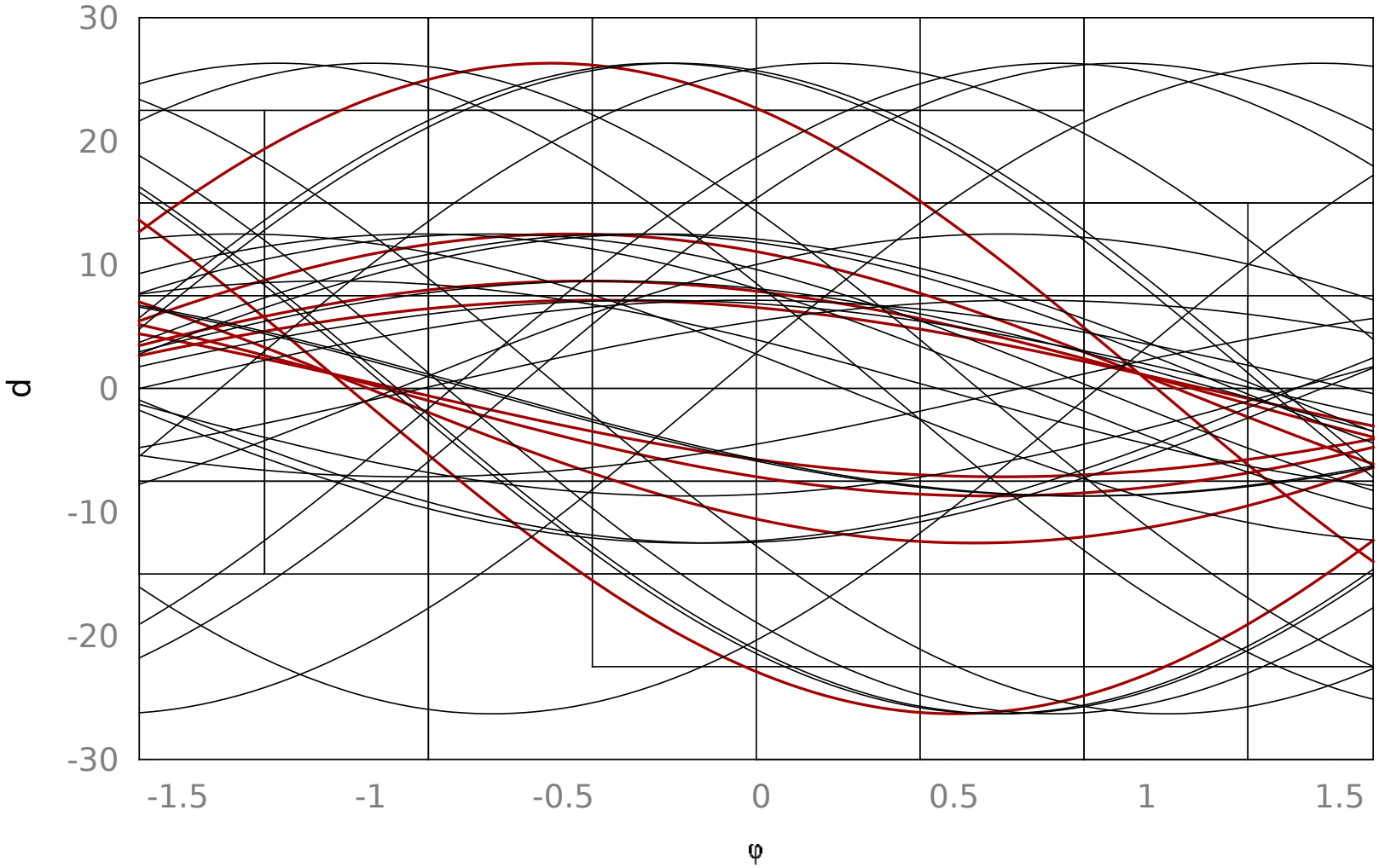
Fast Hough Transformation Example

First iteration



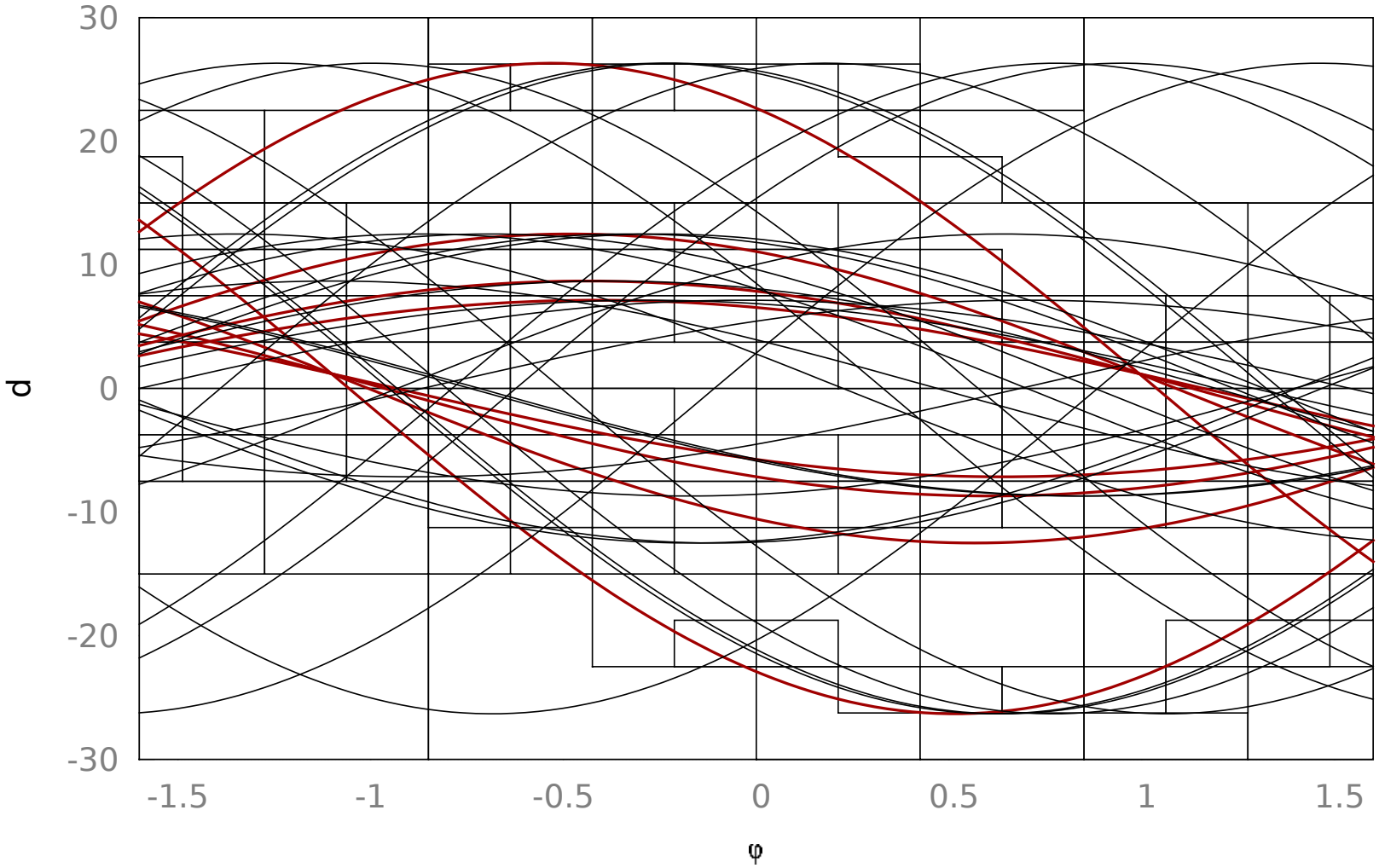
Fast Hough Transformation Example

Two iterations



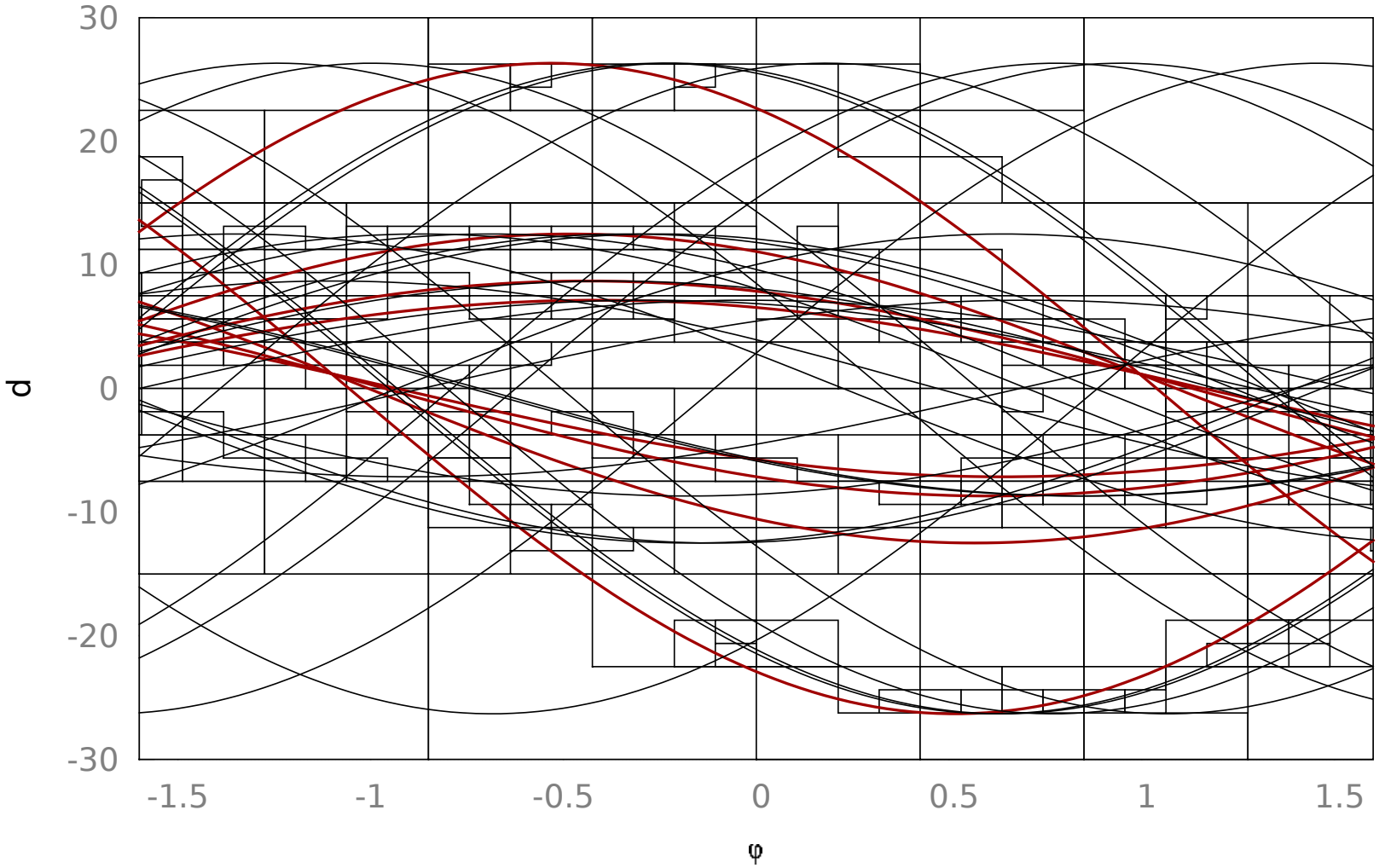
Fast Hough Transformation Example

Three iterations



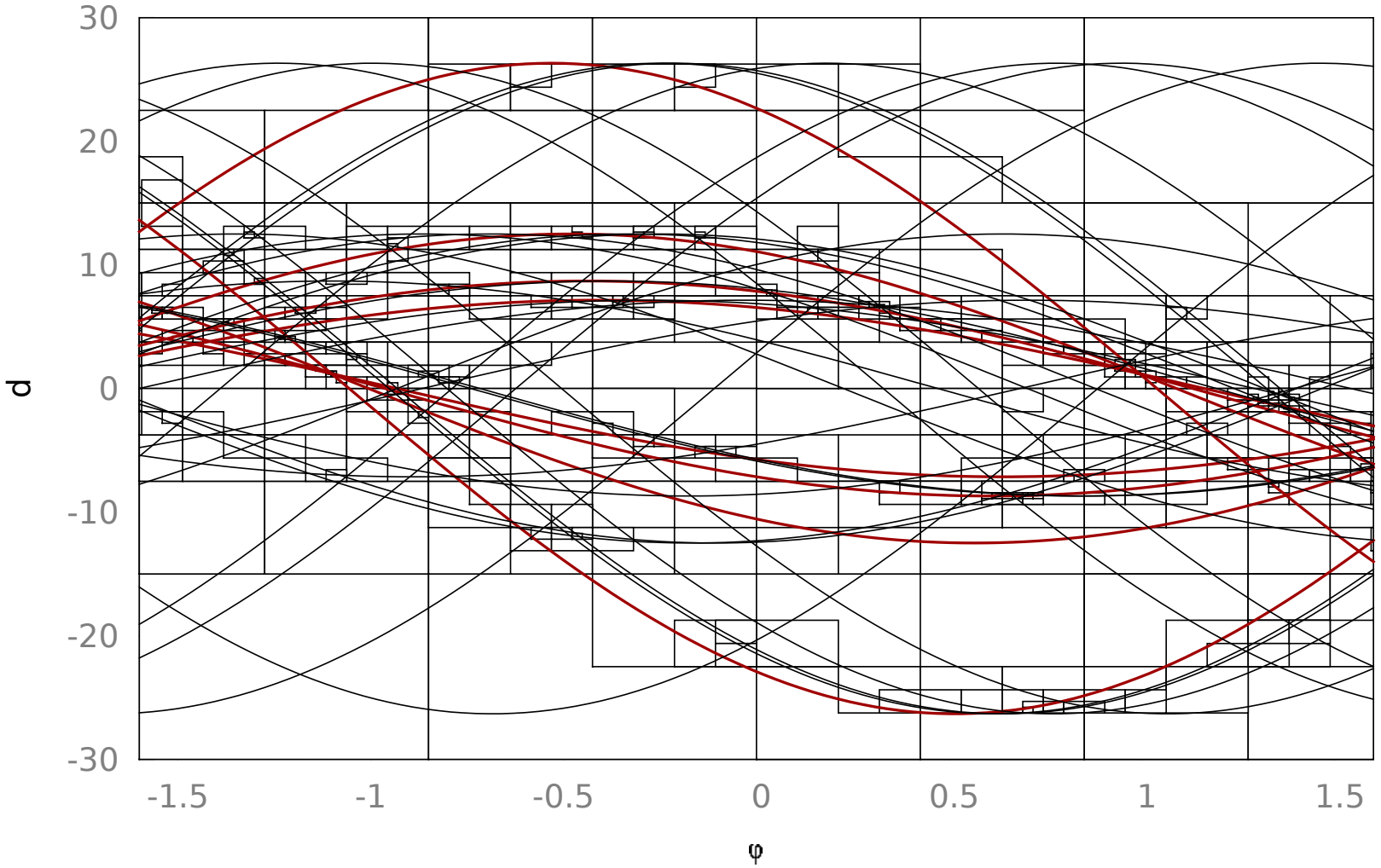
Fast Hough Transformation Example

4 iterations



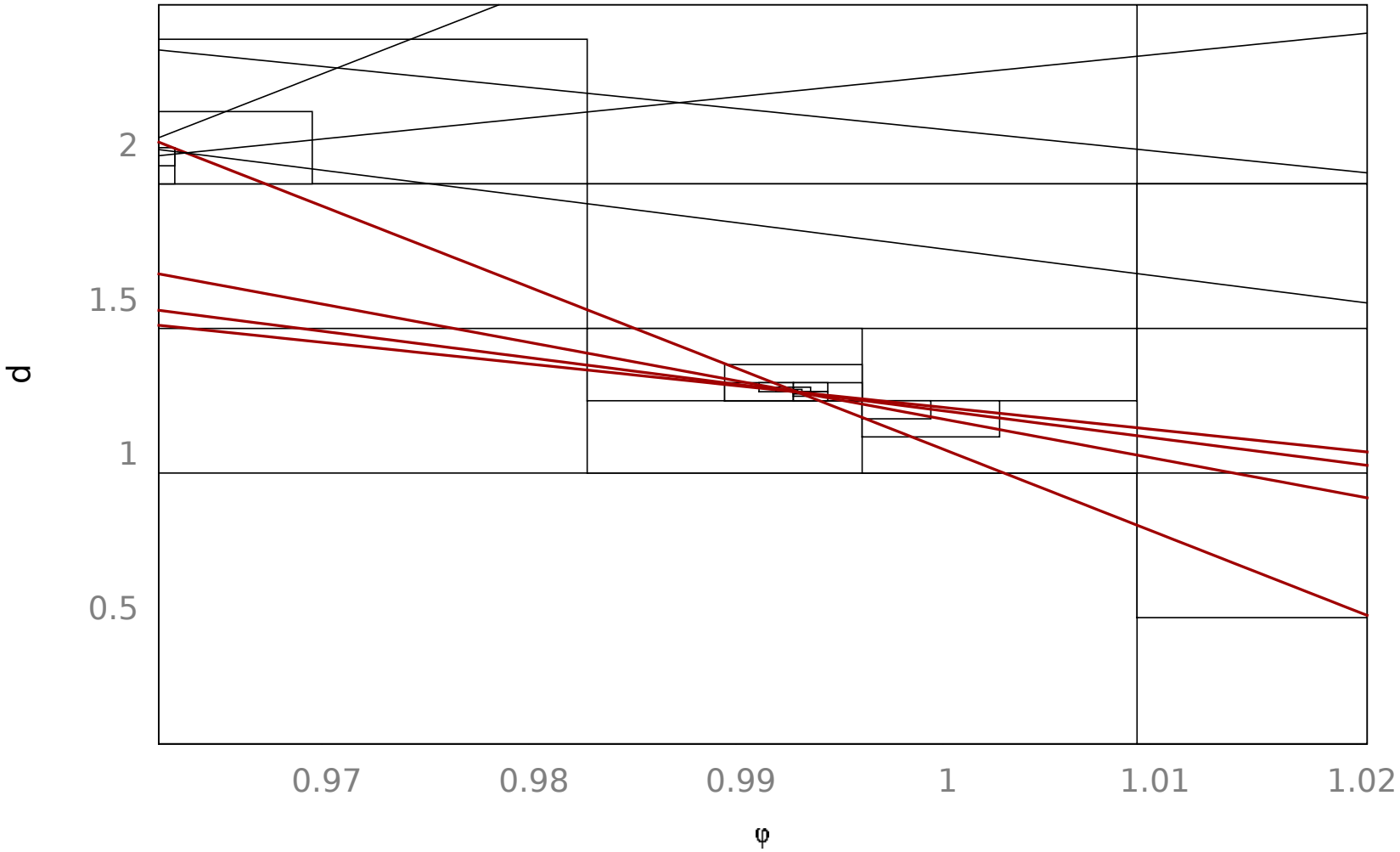
Fast Hough Transformation Example

14 iterations



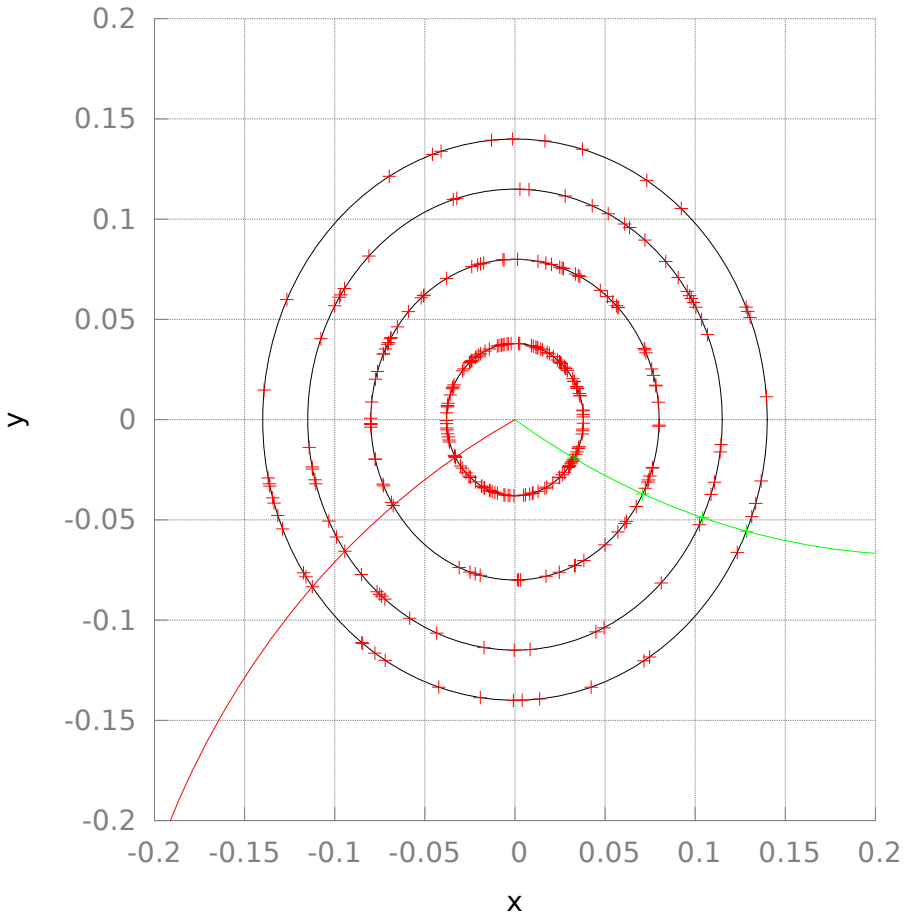
Fast Hough Transformation Example

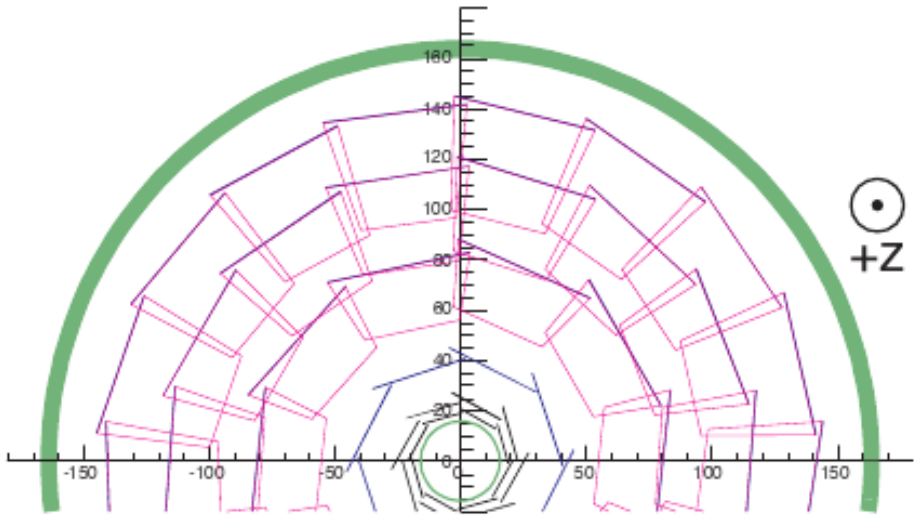
Zoom



2D Realistic Example

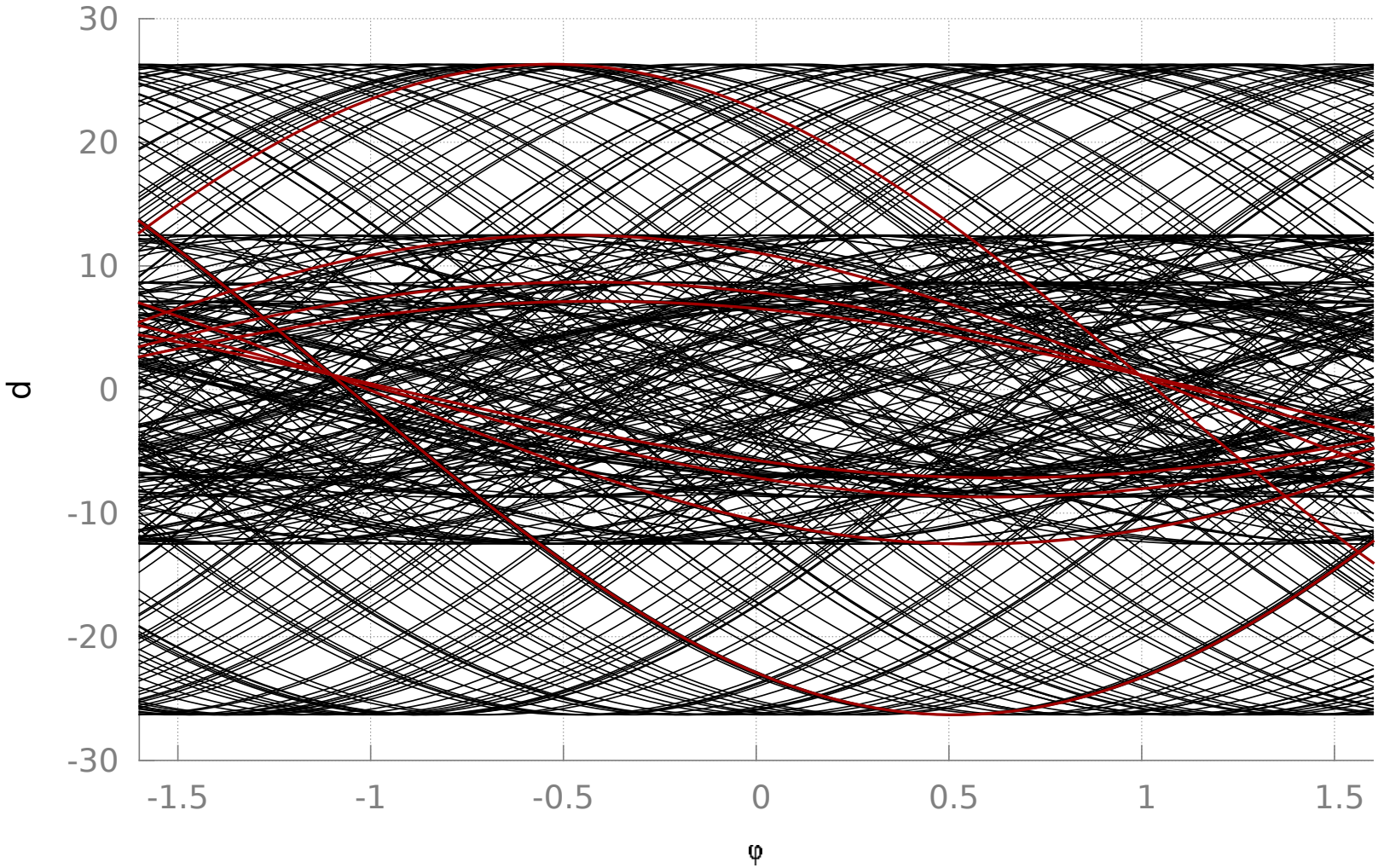
- 300 background hits ($\sim$ 1 percent occupancy scaled with $\frac{1}{r^2}$)
- 2 generated tracks



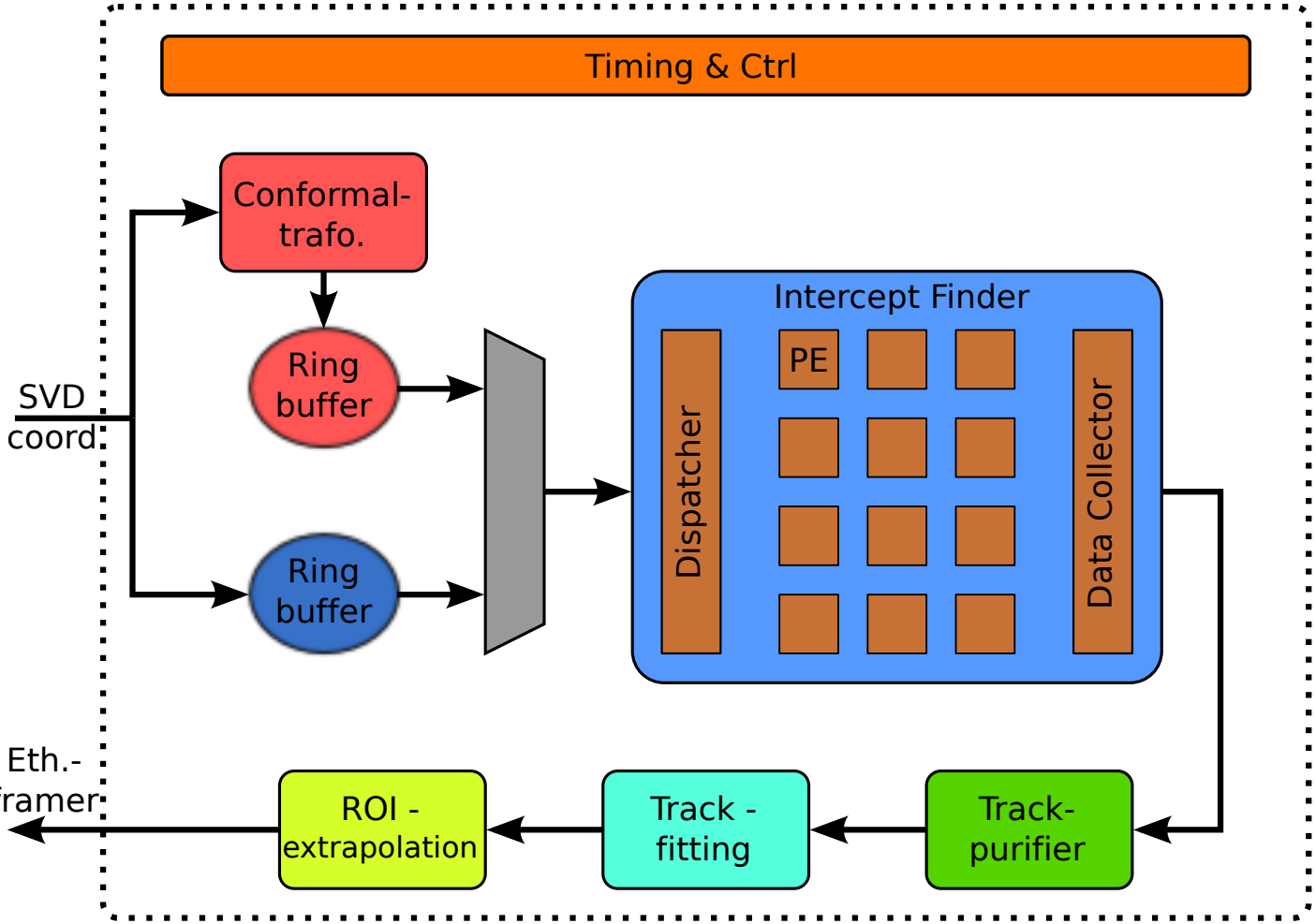


Hough Space

Corresponding Hough Space:

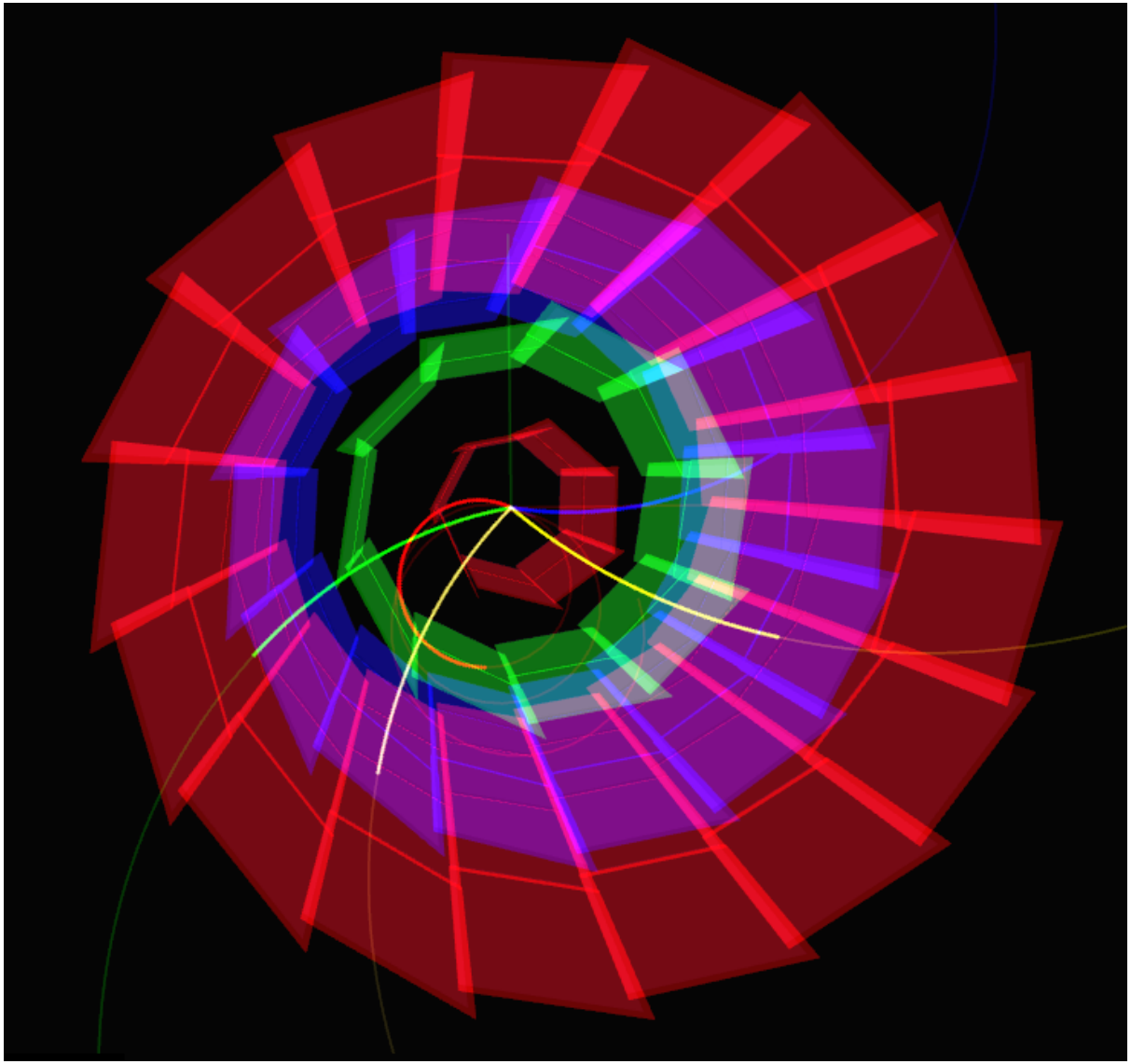


Track Reconstruction Architecture



- PE: Processing Element, basic comparator to detect lines in cell
- Track Purifier: Remove duplicated found tracks and combine two sets

Detector Simulation and Visualization



Evaluation of New Synthesis Tool

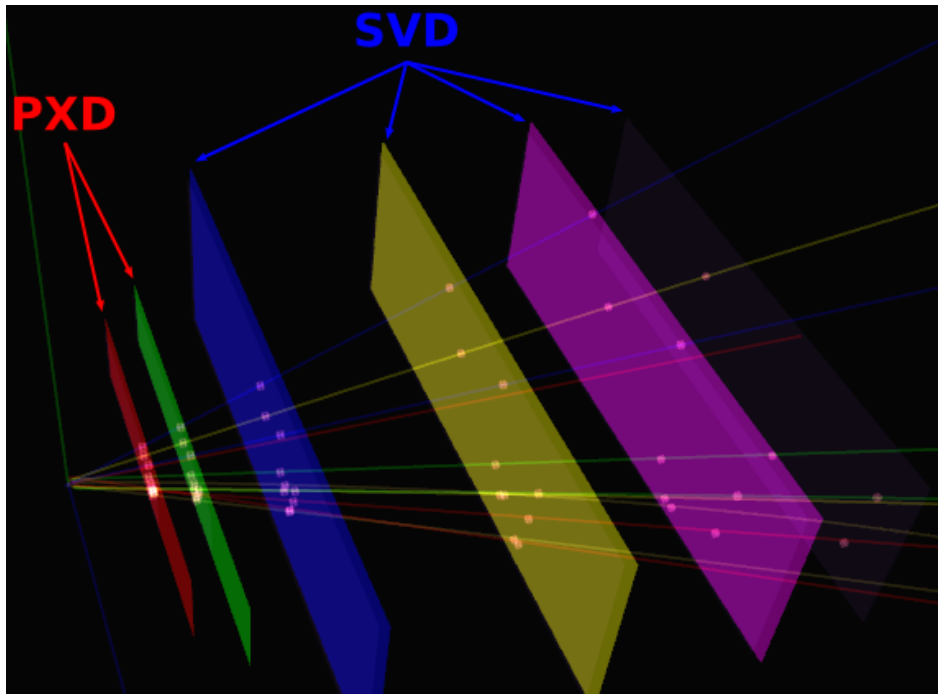
- Using High Level Synthesis with Xilinx HLS for the track reconstruction module
- Directly translate C/C++ code from computer simulation into Verilog
- First tests shows promising results
- Useful for complex operations need to be proved

Summary and Outlook

- Prototype implementation of an distributed FPGA tracking algorithm
- 3D track reconstruction algorithm based on Fast Hough Transformation for a complex geometry detector

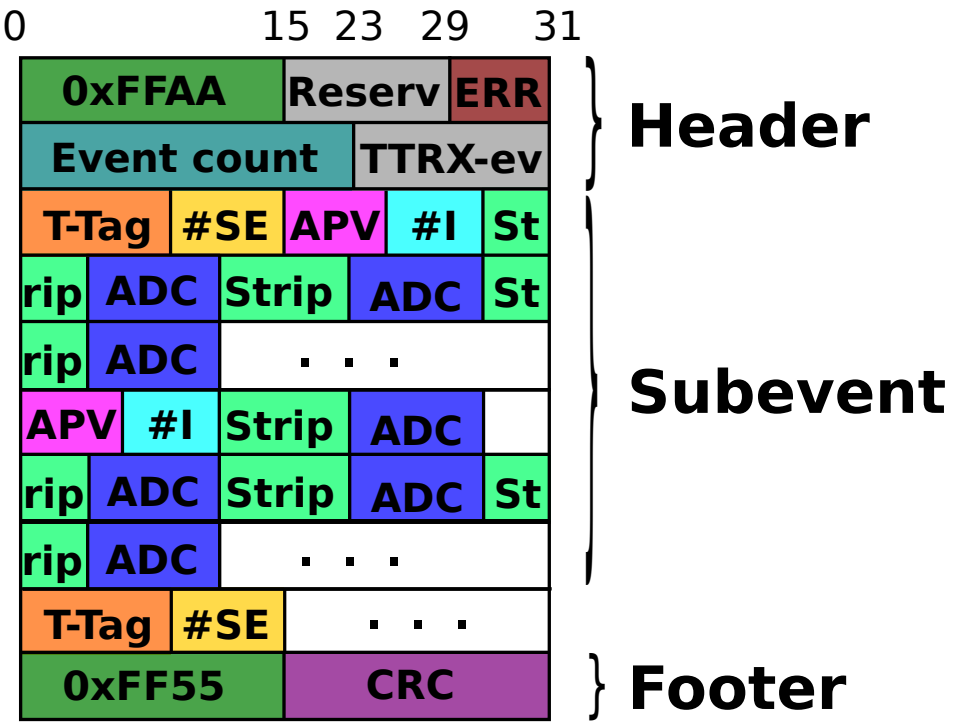
Outlook:

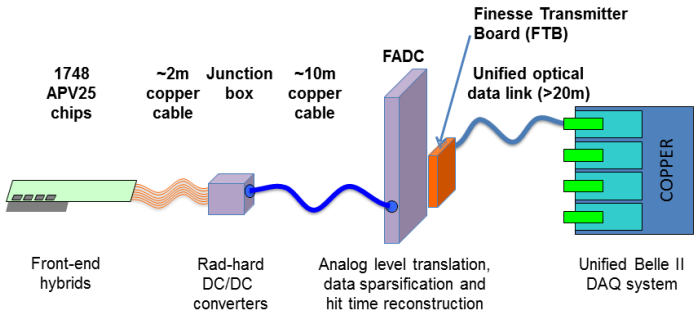
- Tuning and optimization of the FPGA firmware and algorithm
- First test of the system during beam test campaign at DESY in fall 2013



Thank you for your attention!

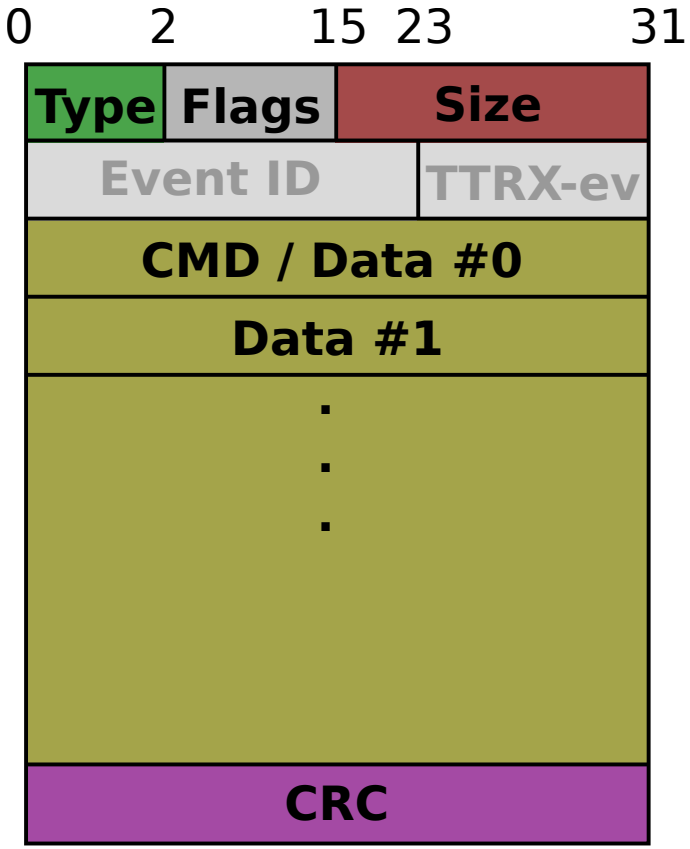
(Proposal for the) Belle II Link with SVD flavor





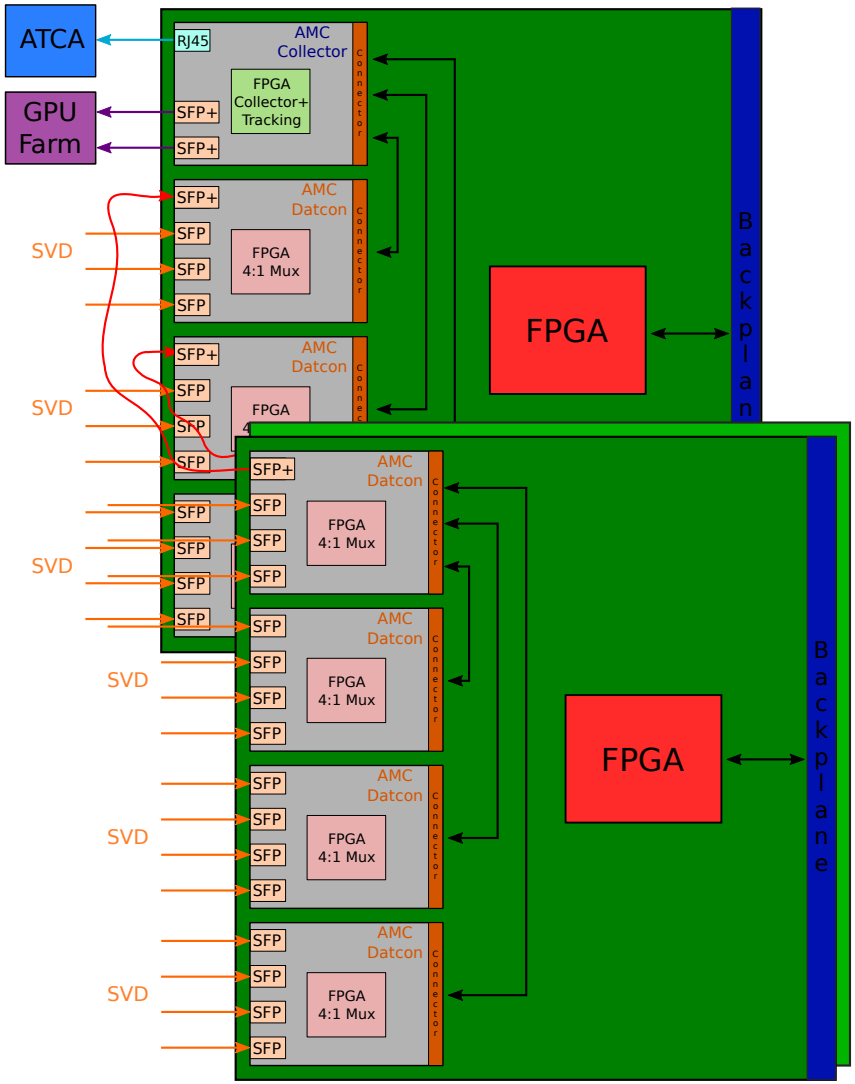
- Layout of header and footer as proposed in the TDR
  - 10 bit Time Tag (T-Tag): 1024 time slots within on event
  - 6 bit #SE: Number of the subevents
  - 6 bit APV: APV identification connected to the FADC
  - 6 bit #I: Number of Strip/ADC values for the FADC-APC
  - 7 bit Strip: Fired Strip ID for the FADC-APC
  - 10 bit ADC: Value of the ADC

Internal Protocol



- Type: 2 Bit
  - 00: Command (CMD)
  - 01: User Data (DATA)
- Flags: 14 Bit
  - $(0000)_{16}$: Normal operation
  - $(0001)_{16}$: Test the connection, waiting for reply (only command)
  - $(0002)_{16}$: Test the connection, reply (only command)
  - $(1XXX)_{16}$: Error, kind of error indicated by XXX

Real Topology Layout



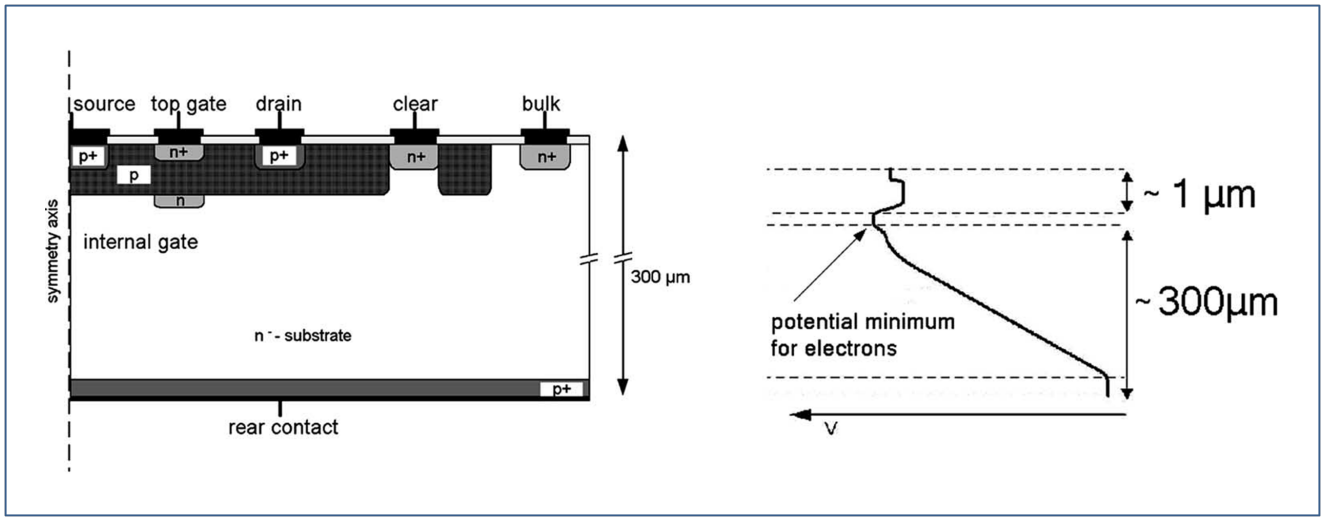
- 3 carrier boards with 4 AMCs
  - → 4:1 / 3:1 Multiplexer in each AMC card
  - → Use one SFP+ transceiver for inter-carrier board transmission
  - → AMC card interconnection over high-speed RocketIOs
- 1 AMC as data collector
  - → 3:1 Multiplexer with track reconstruction algorithm
  - → Ethernet connection for transmission of tracks to ATCA

Aurora overview

- Xilinx Aurora protocol as physical link layer over GTP
- Takes care of every “low level” operation like clock correction, symbol decoding and 8B/10B Encoding
- Easy data transmission/framing over locallink
- Error detection and initialization
- Status wires for channel and lanes
- Lane: One communication line
  Channel: Concentration of one or more lanes

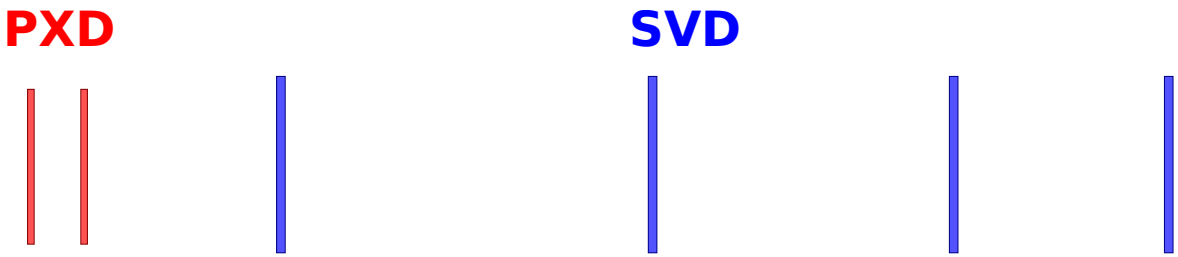
DEPFET Pixel Sensor for Belle II

DEpleted P-channel Field Effect Transistor (DEPFET):



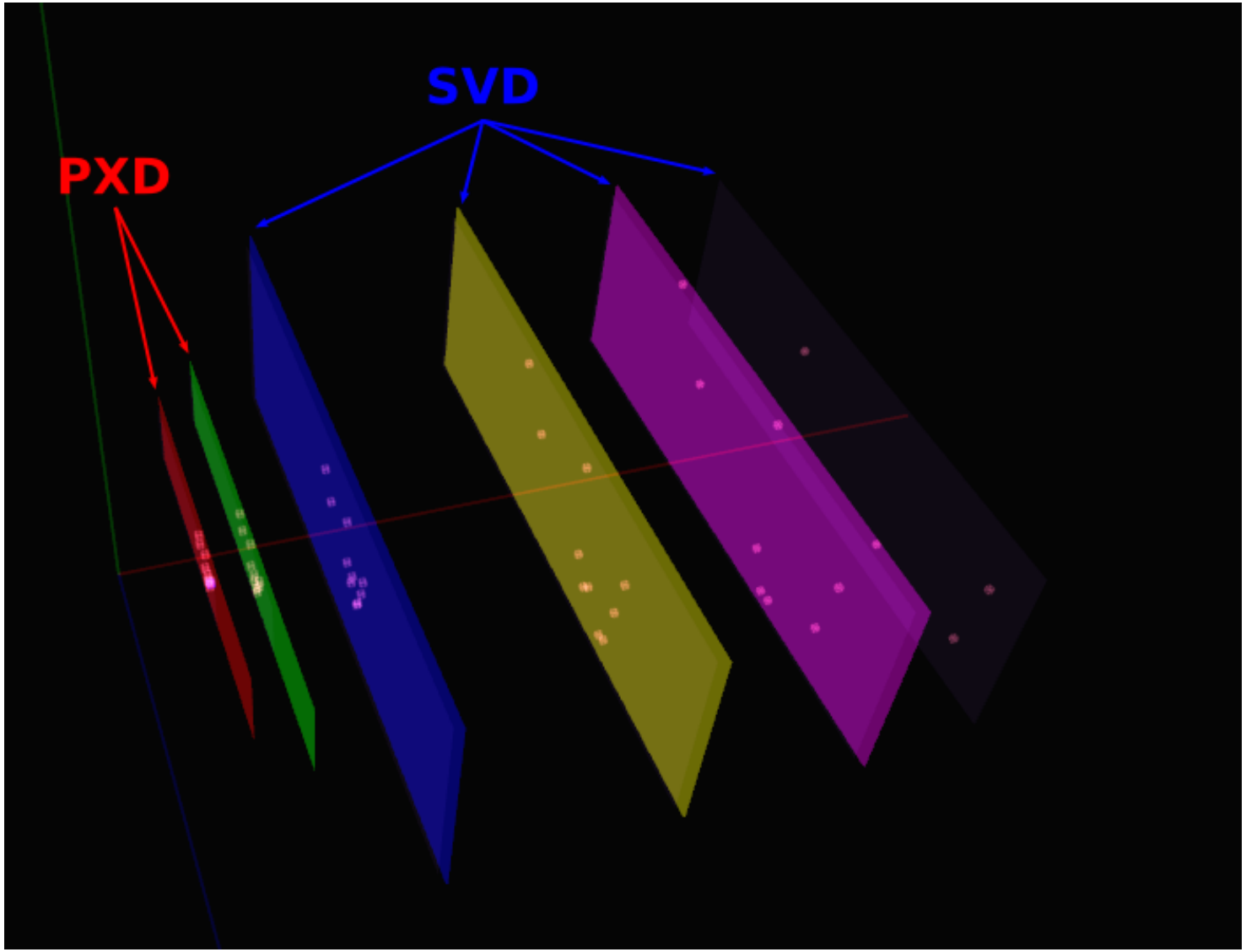
- Two layers at radii 14 *mm* and 22 *mm*
- 8 million readout channels
- 58 $Gbit/s$ bandwidth with 3 percent occupancy

Testbeam Simulation Setup

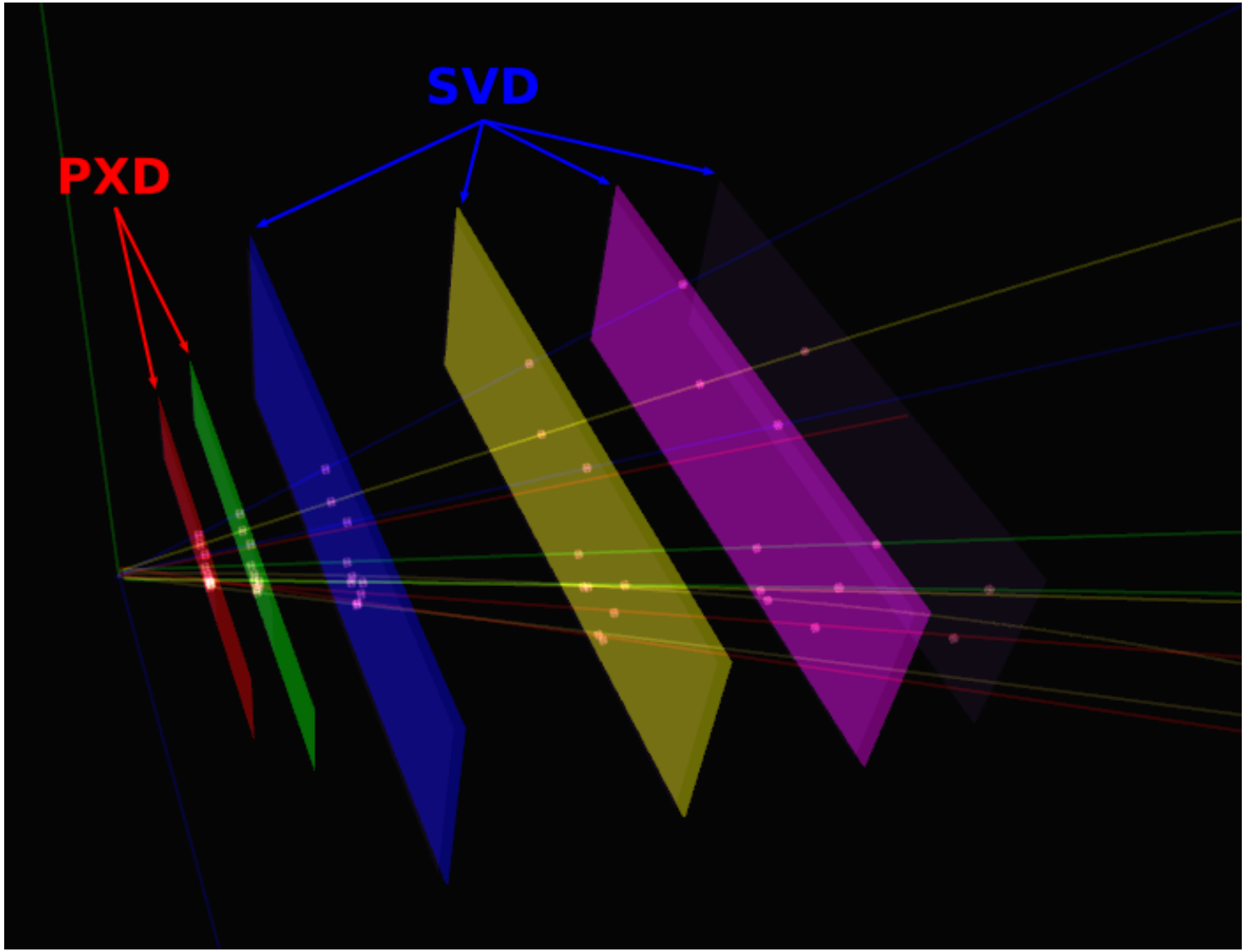


- Set up two layers with one PXD half ladder each.
- And four layers of the SVD in the suggested distance as in the final design
- Run simulation with electrons with energies between 0.5 and 20 *MeV*, 1 T magnet field

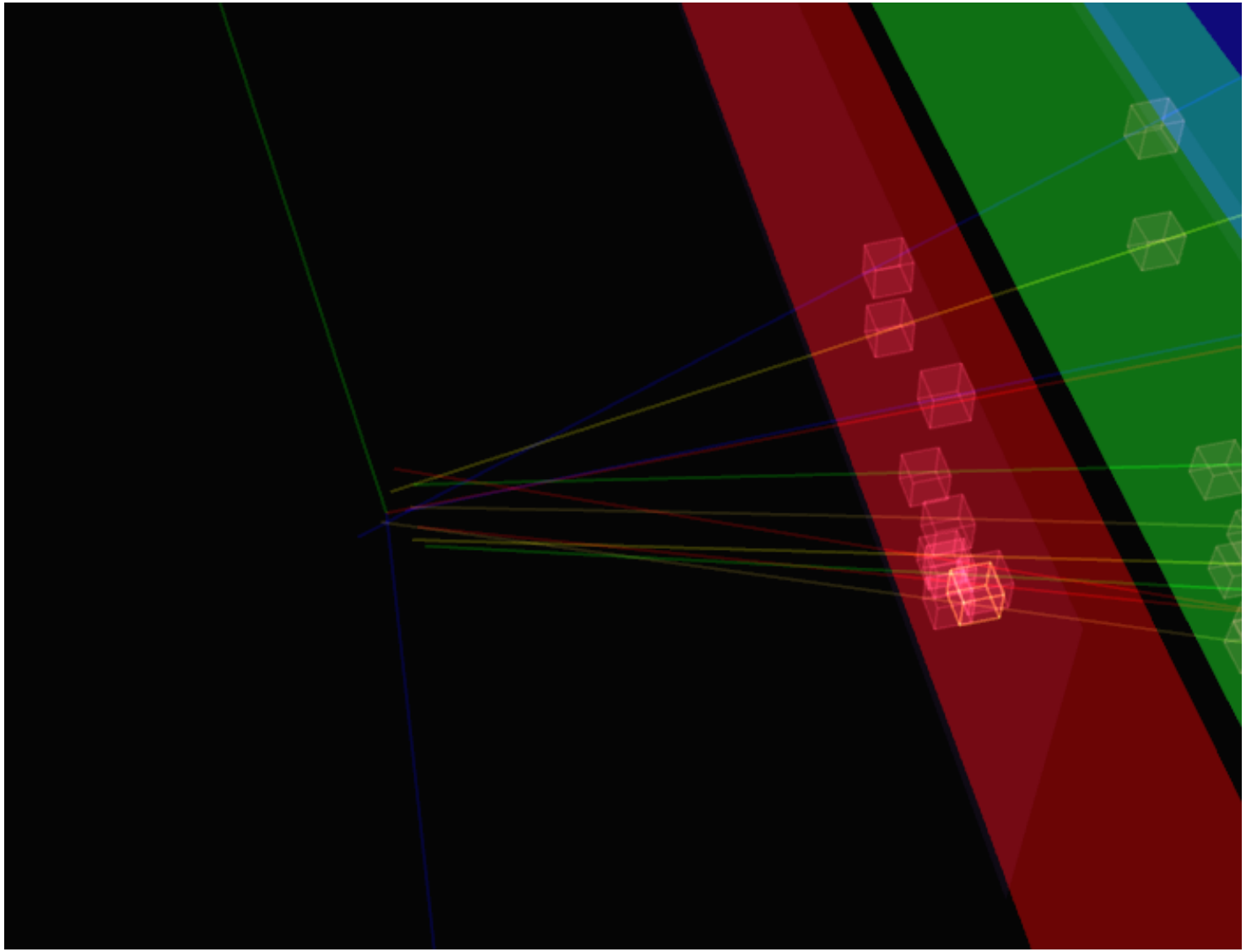
Testbeam Visualization



Testbeam Visualization



Testbeam Visualization

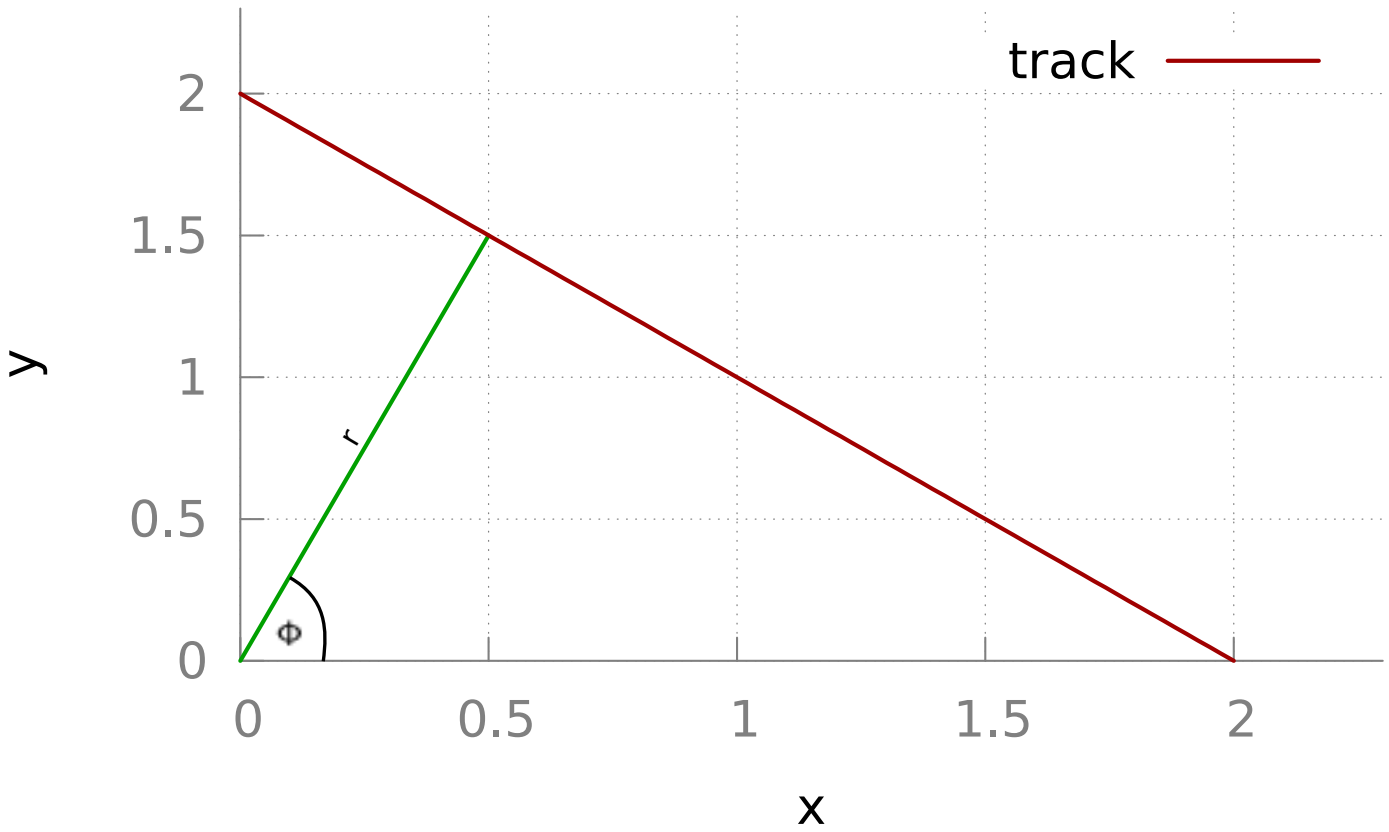


Multiplexing

- Three general types of multiplexing:
  - Frequency Division Multiplexing (FDM)
  - Time Division Multiplexing (TDM)
  - Statistical Multiplexing
- Because of dynamic bandwidth “Statistical Multiplexing” would be most suitable
- Demultiplexing is trivial, if only multiplexing of 32 bit words is ensured

Coordinate Transformation in Hough Space

- This is bad for tracks close or parallel to y-axes (m = inf.)!
- Make a coordinate transformation like: $r = x \cdot cos(\Phi) + y \cdot sin(\Phi)$

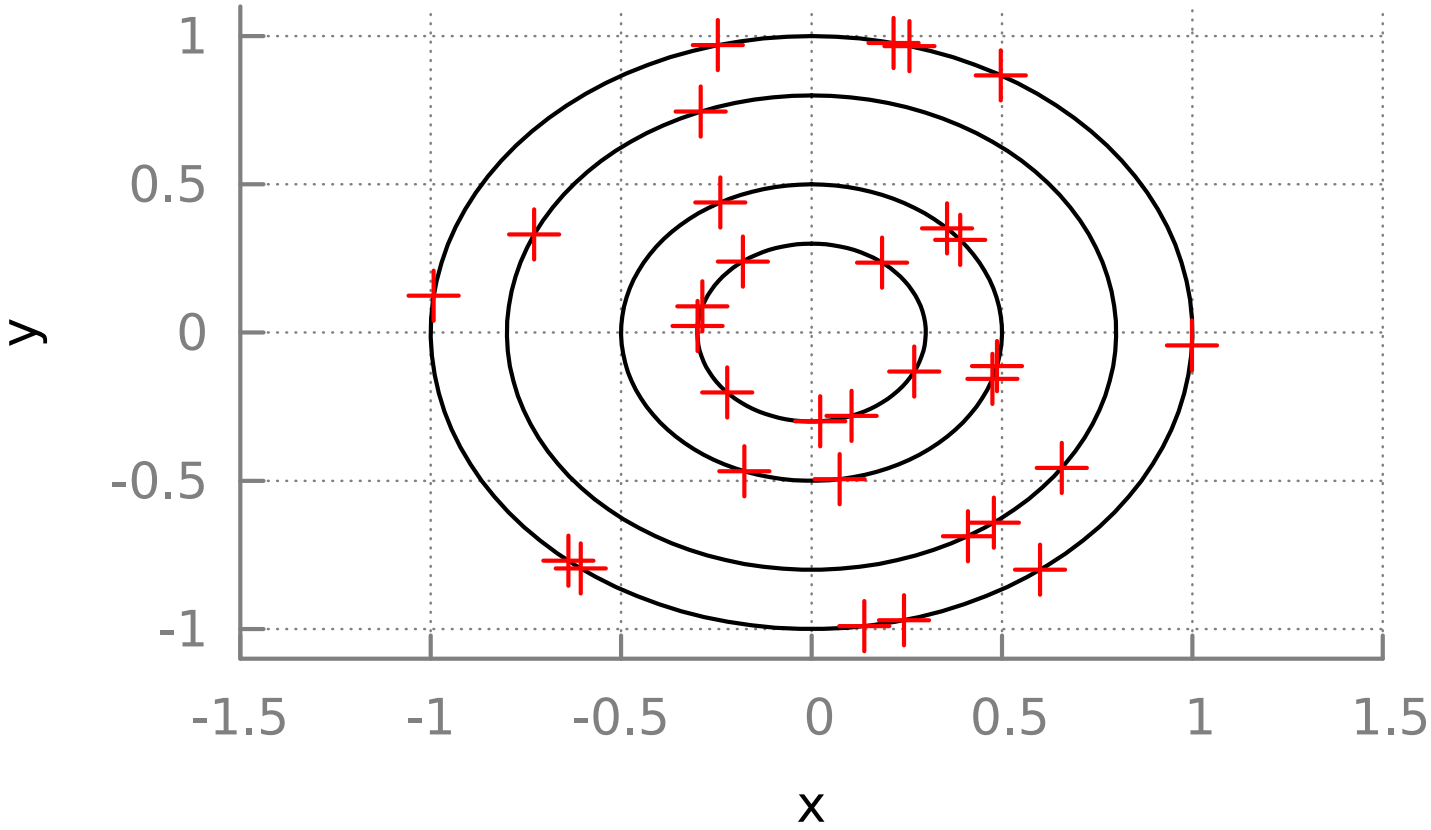


Cellular automaton type

- “Automaton”: In math/theoretical computer science, an abstract machine
- A cellular automaton is a machine, which calculates the state from its cell at $t = t + 1$ with its own state and its surrounding cells at t.
- Create every possible connection between each hits in different layers
- Go reverse and drop connections depending on angle, amplitude, cluster size and other boundary conditions

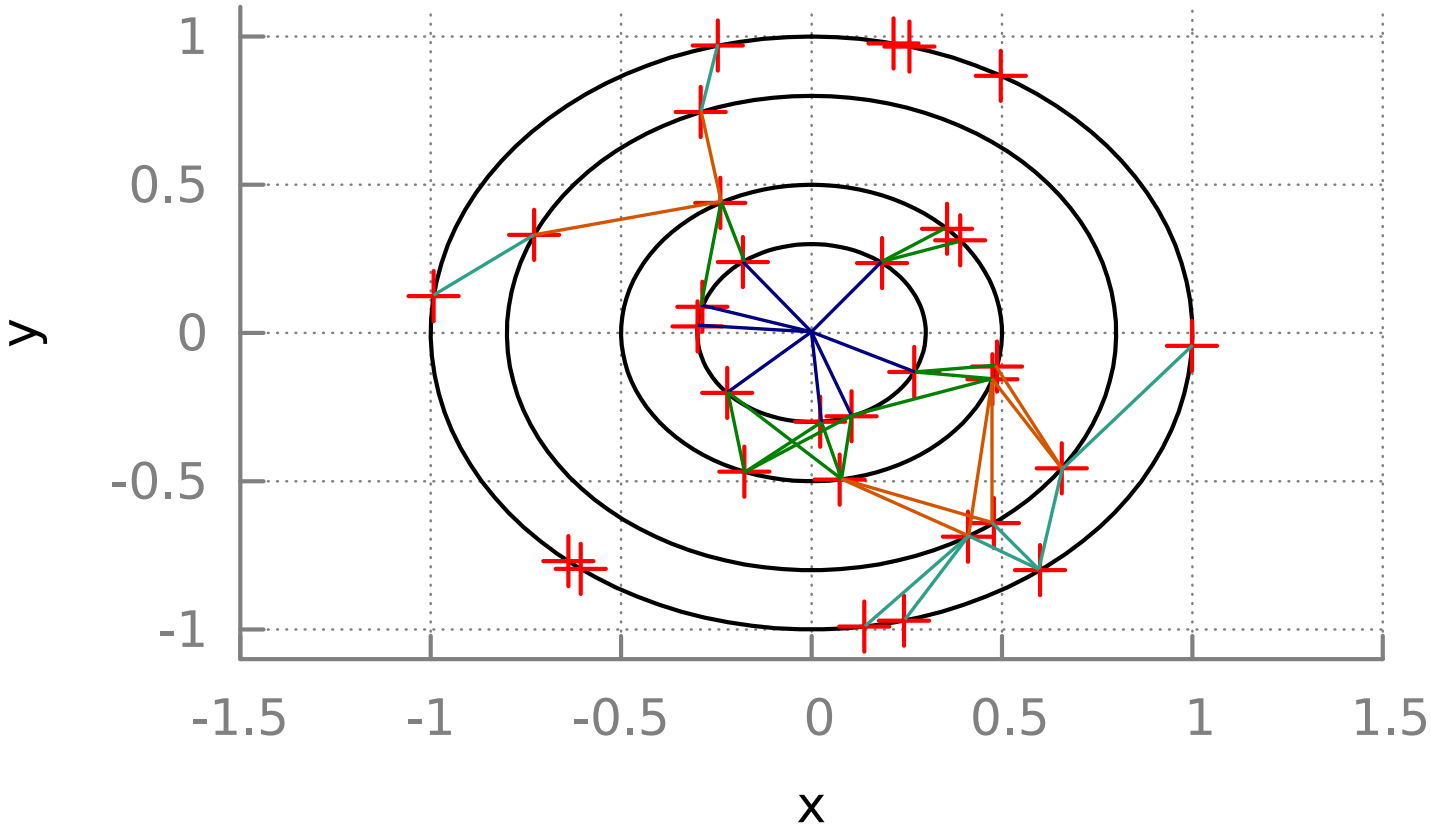
Cellular Automaton Illustration 1

- Example with random data, and one real track



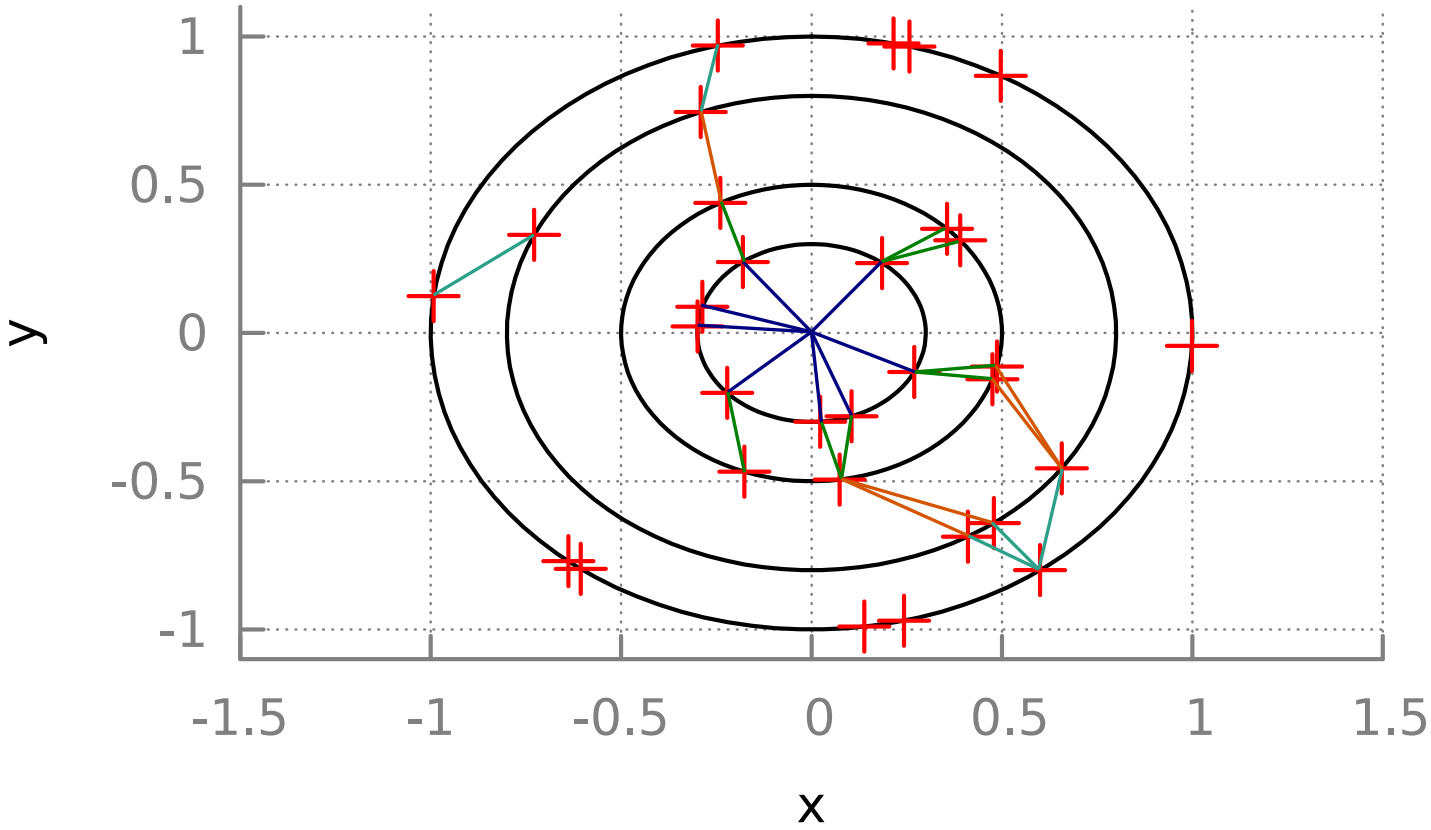
Cellular Automaton Illustration 2

- Connect everything with everything on the next layer



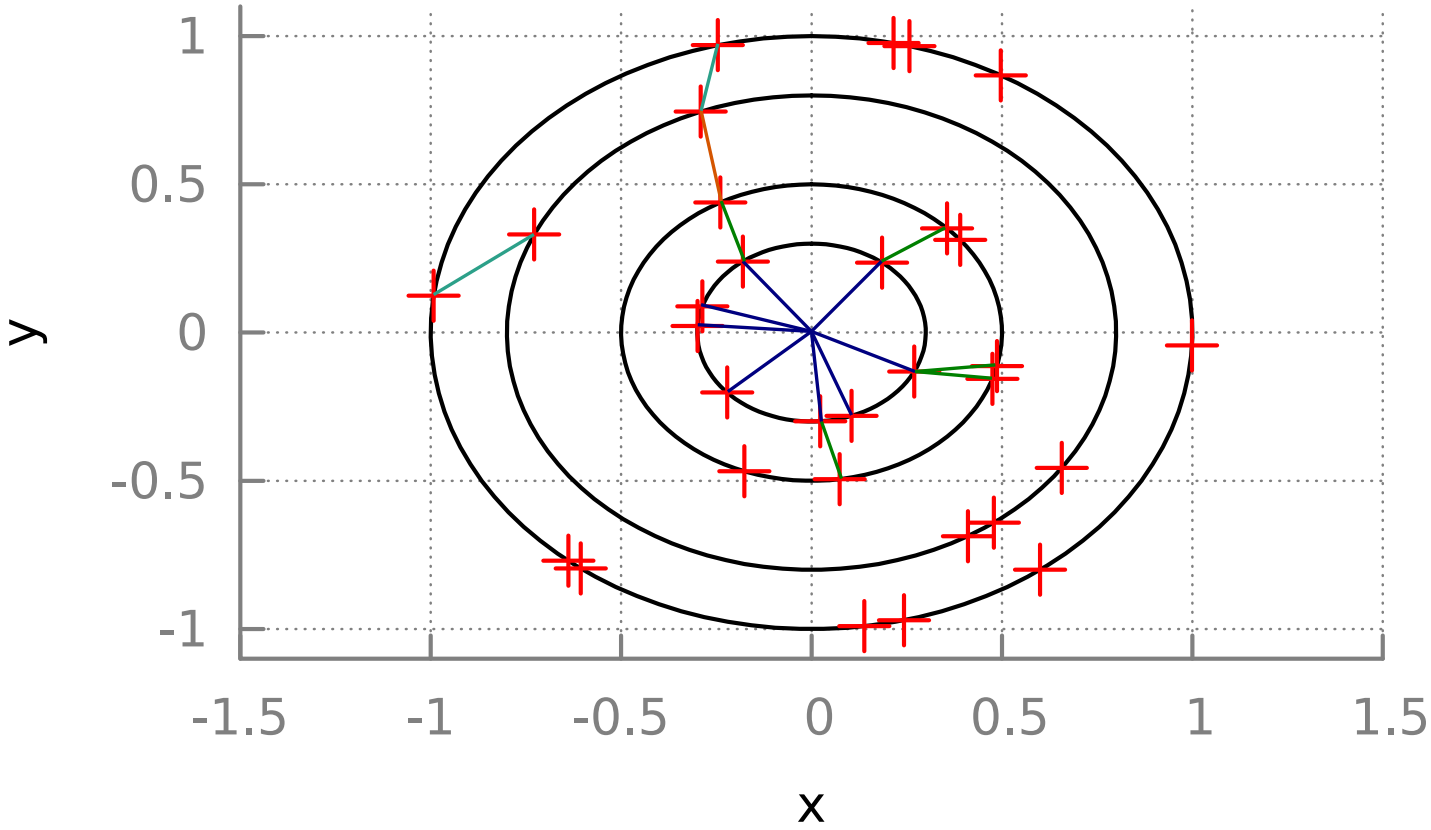
Cellular Automaton Illustration 3

- Lets drop on angle...



Cellular Automaton Illustration 4

- Do some physics...



Cellular Automaton Illustration 5

- Drop non connected...