ARC - a novel RICH detector for a future e^+e^- collider
ARC detector implementation discussion

Martin Tat[1] Roger Forty[2] Guy Wilkinson[1]

[1]University of Oxford

[2]CERN

24th October 2023

Outline

1. **A**rray of **R**ICH **C**ells
2. Optimisation of ARC layout
3. Technical details of optimisation
4. Idea: Analytical optimisation
5. Summary

Array of RICH Cells

Quick recap

Array of RICH Cells

- **A**rray of **RI**CH **C**ells (ARC): A novel RICH detector concept
  - Compact, low-mass solution for particle ID for FCC-ee
  - A large number of small RICH cells
- Adapted to fit into the CLD experiment concept, taking 10% from the tracker volume
  - Radial depth of 20 cm, radius of 2.1 m and a length of 4.4 m
  - Aim to keep material budget below $0.1X_0$
- Aerogel and gas radiators with a spherical mirror
  - Aerogel also acts as thermal insulation between gas and detector

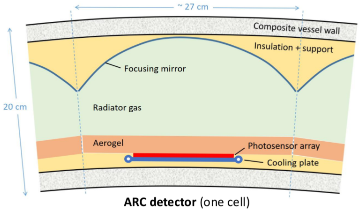

Figure 1: ARC has a cellular structure, similar to an insect's compound eyes

Array of RICH Cells

- All cells are the same size, organised on a hexagonal grid
  - Barrel (endcap) has 945 (402) cells in total, where 18 (23) are unique
  - Some cells are partial cells, but all photon sensors are the same size

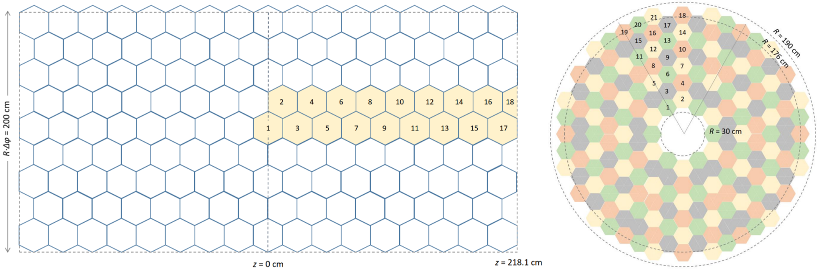

Figure 2: Barrel (left) and endcap (right) cells

Array of RICH Cells

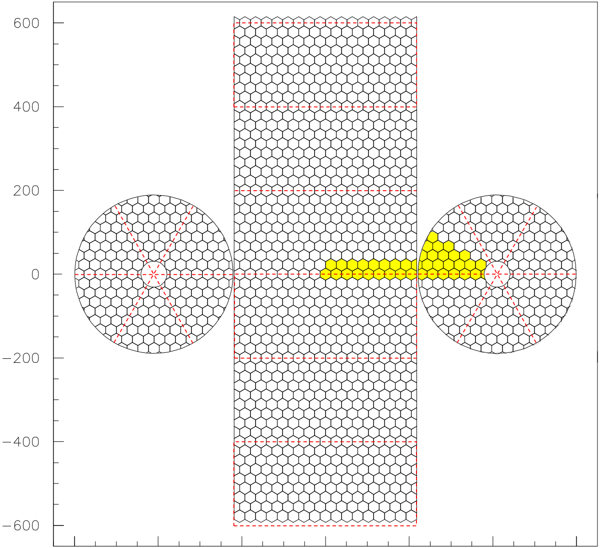

Figure 3: Barrel and endcap cells

Photon hits

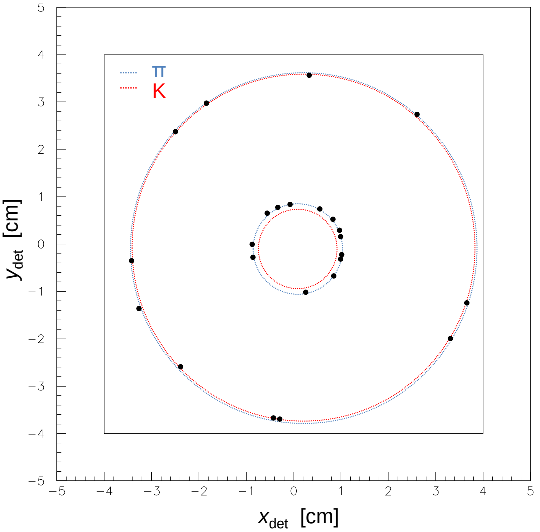

Figure 4: Photon hits on photodetector

Event display

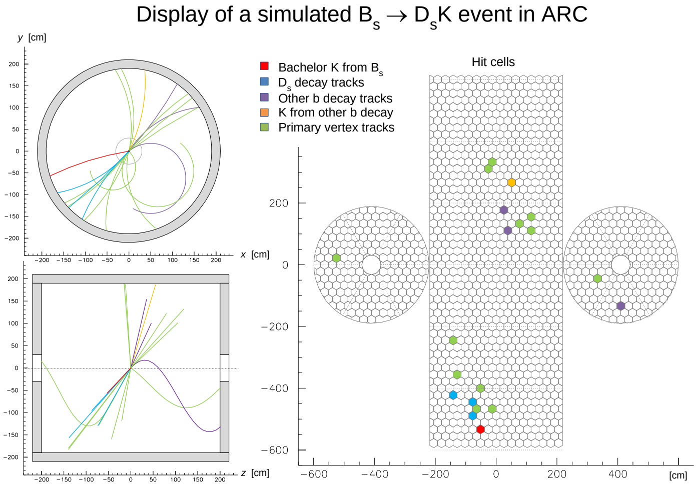

Figure 5: $B_s \to D_s K$

Optimisation of ARC layout

My strategy for optimising ARC

How to calculate ARC performance

1. Generate 2×10^4 charged pions with high momentum (100 GeV)
2. Generate Cherenkov photons from gas
3. Track photons through the optics
4. For each photon, reconstruct the Cherenkov angle θ
5. Find average θ from all detected photons from the charged track
6. For each cell, vary 5 parameters to optimise the performance

"Figure of merit": Cherenkov angle uncertainty

$$\Delta\theta = \frac{1}{\sqrt{N}} \times \frac{1}{N-1} \times \sum_{i=0}^{N-1} (\theta - \bar{\theta})^2$$

Parameters to optimise

1. Mirror curvature
2. Mirror horizontal position
3. Mirror vertical position
4. Detector horizontal position
5. Detector tilt

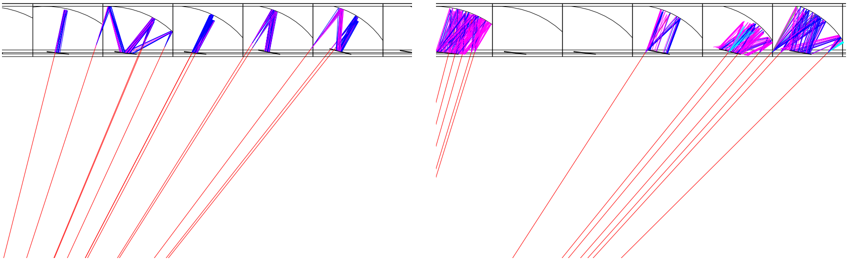

Figure 6: Tracking of photons from gas radiator (left) and aerogel radiator (right) through the ARC optics

Stochastic optimisation

- Issue: Each iteration uses a finite number of photons
  - Cannot use minimisation algorithms based on gradients
- Use stochastic optimisation:
  1. Differential evolution
  2. Start with a population of candidate solutions in parameter space
  3. In each iteration, each solution has a small probability of "evolving" by combining existing solutions
  4. Iterate until it "converges"
- I found an implementation here
  - A bit slow, but it seems to find a sensible minimum
  - Requires some tweaking of parameter bounds

Technical details of optimisation

How to run the code?

GitHub repository

My optimisation code can be found here

```
git clone git@github.com:MartinDuyTat/
    ARC_Simulation_Reconstruction.git
cd ARC_Simulation_Reconstruction
mkdir build
cd build
cmake ..
make install
cd ../options
```

I have used ROOT 6.22/2 and C++17

GitHub repository

To run the optimisation, we need 5 options files:

```
General.txt
ARCGeometry.txt
RadiatorCell.txt
Particle.txt
Optimisation.txt
```

General: Number of tracks, chromatic dispersion, seed, etc

GitHub repository

To run the optimisation, we need 5 options files:

```
General.txt
ARCGeometry.txt
RadiatorCell.txt
Particle.txt
Optimisation.txt
```

ARCGeometry: ARC length, radius, number of cells, B-field strength

GitHub repository

To run the optimisation, we need 5 options files:

```
General.txt
ARCGeometry.txt
RadiatorCell.txt
Particle.txt
Optimisation.txt
```

RadiatorCell: Radiator size, aerogel thickness, etc, **optimised parameters**

GitHub repository

To run the optimisation, we need 5 options files:

```
General.txt
ARCGeometry.txt
RadiatorCell.txt
Particle.txt
Optimisation.txt
```

Particle: Momentum, particle ID, direction, etc

GitHub repository

To run the optimisation, we need 5 options files:

```
General.txt
ARCGeometry.txt
RadiatorCell.txt
Particle.txt
Optimisation.txt
```

Optimisation: Number of iterations, population size, bounds, etc

Running the optimisation

To optimise cell labelled "column 3, row 1", run:

```
OptimiseARC 3 1 General General.txt Particle Particle.txt \
ARCGeometry ARCGeometry.txt RadiatorCell RadiatorCell.txt Optimisation Optimisation.txt
```

1. This will create a file named "FitResults.txt"
2. Copy the contents into the file "RadiatorCell.txt"
3. Optimise the next cell
4. Note: The cells must be optimised in order!

Idea: Analytical optimisation

A possible solution for a faster optimisation

Idea: Analytical optimisation

For a given track going through ARC, three variables fully describe each emitted photon:

1. The true Cherenkov angle θ_c^{true}
2. The azimuthal angel ϕ_c of emission
3. Position along the charged track $s \in [0, 1]$

The Probability Distribution Function (PDF) separates into:

$$P(\theta_c^{\mathrm{true}}, \phi_c, s) = P(\theta_c^{\mathrm{true}}) \times P(\phi_c) \times P(s) \times \Theta(\theta_c^{\mathrm{true}}, \phi_c, s)$$

Idea: Analytical optimisation

By tracing photons through the ARC optics, we can map $\vec{v} \equiv (\theta_c^{\text{true}}, \phi_c, s)$ into $\vec{w} \equiv (\theta_c^{\text{rec}}, \phi_c, s)$, and we call the transformation $\vec{w} = \vec{f}(\vec{v})$

If we define the Jacobian

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_c^{\text{true}}} & \frac{\partial f_1}{\partial \phi_c} & \frac{\partial f_1}{\partial s} \\ \frac{\partial f_2}{\partial \theta_c^{\text{true}}} & \frac{\partial f_2}{\partial \phi_c} & \frac{\partial f_2}{\partial s} \\ \frac{\partial f_3}{\partial \theta_c^{\text{true}}} & \frac{\partial f_3}{\partial \phi_c} & \frac{\partial f_3}{\partial s} \end{bmatrix},$$

then the reconstructed Cherenkov angle has the following PDF:

$$P(\theta_c^{\text{rec}}, \phi_c, s) = P(\theta_c^{\text{true}}, \phi_c, s)/|J|$$

Note: Each derivative in J can be numerically calculated by only tracing two photons!

Idea: Analytical optimisation

Conclusion: The PDF $P(\theta_c^{\mathrm{rec}}, \phi_c, s)$ can be analytically calculated with by tracking and reconstructing only 18 photons

The PDF can be (numerically) integrated to obtain the standard deviation, which is directly related to the ARC resolution/performance

"Figure of merit": Cherenkov angle uncertainty

$$\Delta\theta = \frac{1}{\sqrt{N}} \times \frac{1}{N-1} \times \sum_{i=0}^{N-1} (\theta - \bar{\theta})^2$$

Summary

These slides contain the (very boring details of):

1. How the ARC optimisation works
2. How to run the ARC optimisation
3. An idea for an analytical calculation of the ARC performance

Thanks for your attention!