Topics in Particle Physics and Astroparticles II

Guilherme Soares

Laboratory of Instrumentation and Experimental Particle Physics

10 March 2022

Index

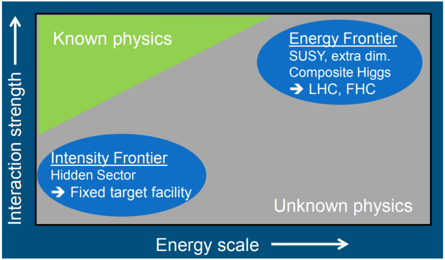
Current State of Particle Physics

The Standard Model is a very successful model, that describes all known elementary particles and their interactions up to the TeV scale. However, it is not able to explain some outstanding phenomena such as :

- Baryonic Assymetry of the Universe (BAU)
- Dark Matter
- Neutrino Oscillations
- Departure from Leptonic Flavour Universality

Can we solve all of this summarily ?

Beyond the Standard Model Physics

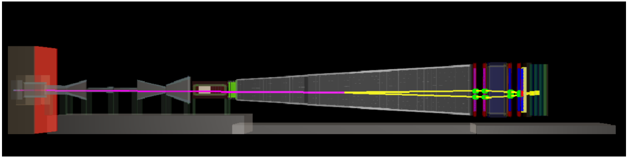


Requirements of Hidden Sector searches :

- High Luminosity
- Long Expected Lifetime
- Controlled Background Levels

The Search for Hidden Particles Experiment

The SHiP Experiment



SHiP is a discovery experiment designed to find particles whose production is heavily suppressed ($O(10^{-10})$), with masses of $< \mathcal{O}(10)\text{GeV/c}^2$.

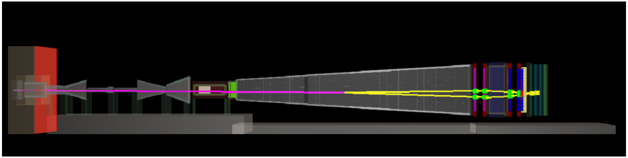
Experiment details :

- 400 GeV/c protons
- 2×10^{20} p.o.t.
- 5 years running
- Discoveries through > 2 decays

Therefore, the background must be totally under control to ensure a zero background environment.

The expected relevant sources of background are the following :

The SHiP Experiment



SHiP is a discovery experiment designed to find particles whose production is heavily suppressed ($O(10^{-10})$), with masses of $< \mathcal{O}(10)\mathrm{GeV/c}^2$.

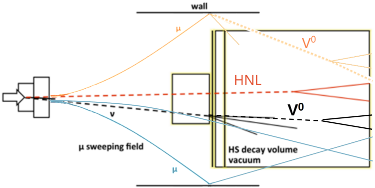
Experiment details :

- 400 GeV/c protons
- 2×10^{20} p.o.t.
- 5 years running
- Discoveries through $>$ 2 decays

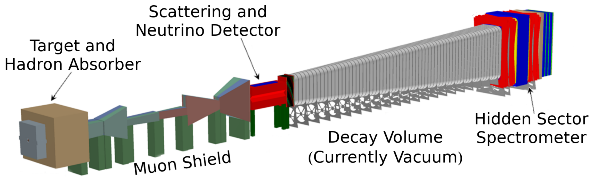
Therefore, the background must be totally under control to ensure a zero background environment.

The expected relevant sources of background are the following :

- Neutrino Deep Inelastic Scattering
- Muon Deep Inelastic Scattering
- Muon Combinatorial

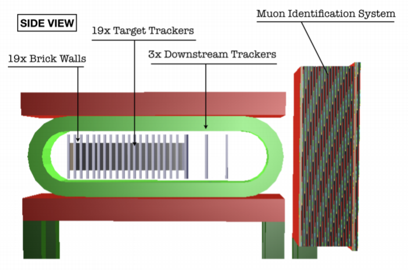


Target, Shields and the Scattering and Neutrino Detector



Titanium-Tungsten Target
Hadron Absorber
Muon Shield
Scattering and Neutrino Detector :

- Magnet
- Emulsion Cloud Chamber
- Compact Emulsion Spectrometer
- Muon Identification System (Upstream Background Tagger)

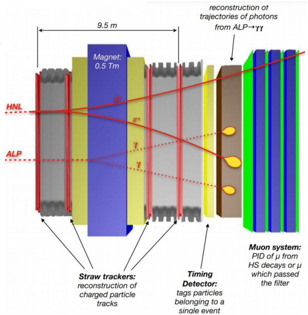


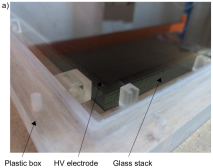
Hidden Sector Apparatus

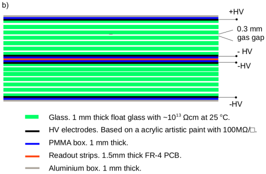
Decay Vessel
Spectrometer Straw Tracker
Timing Detector
Calorimeter (*SplitCal*)
Muon Identification System

MRPC based Timing Detector
Sealed Glass Stack (SGS)
Two identical 6-gap MRPCs with dimensions of $1600 \times 1200 mm^2$.
Final detector is a 7×5 matrix of modules.

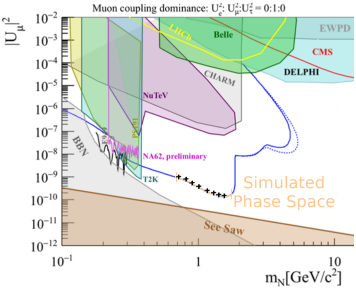
- Average timing accuracy of 54 ps
- 98% detection efficiency

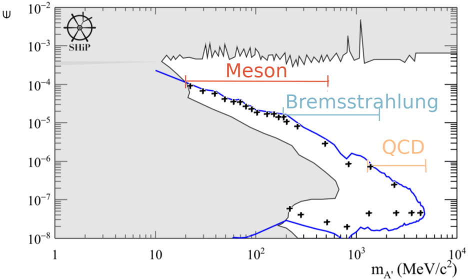






Heavy Neutral Leptons / Dark Photons





Heavy Neutral Leptons

Production Modes :

- D decay
  2×10^{18} mesons
- B decay
  2×10^{14} mesons

Decay Modes :

- $N \rightarrow \ell^{\pm}\, \pi^{\mp}$
- $N \rightarrow \ell^{\pm}\, \rho^{\mp}$
- $N \rightarrow \rho^{0}\, \nu_{\ell}$
- $N \rightarrow \mu^{+}\, \mu^{-}\, \nu_{\ell}$

Dark Photons

Production Modes :
(Photon Mixing)

- Meson Decay
- Proton Bremmstrahlung
- QCD Fusion

Decay Modes :

- $A' \rightarrow \ell^{+}\, \ell^{-}$
- $A' \rightarrow q\, \overline{q}$

Kinematic Properties

The required background levels at SHiP are of < 0.1 events over the course of the experiment.

Resources :

- PID
- **Kinematic Features**

Total Momentum
Transverse Momentum
Fraction of Transverse Momentum
Opening Angle
Impact Parameter
Coordinates of the Decay Vertex

Tutorial 3 - Data Analysis with Pandas

The 3rd tutorial can be found on the usual places, with the name Template_Data_Analysis.ipynb

This template will show you the basics on handling dataframes with pandas.

The topics covered will be the following :

- Printing the DataFrame
- Reading Entries one at a time
- Concatenating DataFrames
- Adding and Removing Columns
- Bulk Dataset Cuts
- Plotting Variables
- Normalizing Data

Tutorial 3 - Data Analysis with Pandas

In order to start messing around with Pandas dataframes, first we need some data to import. Available on the "Data" folder are 2 files with the names Signal.dat and Background.dat. These will be the files that we will look at with this template.

Both .dat files are saved in a .csv format. This means that each line corresponds to an event, and variables are separated by commas ",", with the first line having the name of each variable. This format can be imported to Pandas dataframes directly through pandas : :readcsv("path/to/file").

The first two blocks of code are very simple commands : **pandas : :head()** and **pandas : :tail()**. This commands function in a similar way to a print(), but only show the first and last 5 entries of the dataframe, respectively, alongside the variable names.

Although these commands are not very helpful while working the data, they are great ways to verify the structure of the dataframe, and quick check to know that everything is going well when altering the dataframe in bulk.

In order to read the dataframe, usual iteration methods work fine. Notice that to iterate along a column you need to write a variable explicitly in an analogous way to .root files $\rightarrow$ *for entry in Signal["variable"] :*

Tutorial 3 - Data Analysis with Pandas

Merging dataframes is very useful. Pandas does this comfortably with the pandas : :concat([dataframe1,dataframe2],ignore_index=True).

If we want to add a column with a flag that distinguishes 2 different dataframes before we merge them, we can also add it with another simple command dataframe['NewFlag'] = 1. This creates a new column where every entry has the value 1.

If for some reason you want to remove a column from your dataset, you can also do it in bulk by utilizing the pandas : :dataframe : :drop() function. Although the example shown in the code only refers to eliminating columns, the dataframe : :drop() function is much more powerfull and will be utilized in our future Machine Learning classes.

One feature to take into account when working with pandas dataframes is that most functions have an *inplace* option. This refers to alterations "inplace", which means in the dataframe itself. This option is False by default, as most of the time you want to create a new dataframe without altering the previous one, to avoid overwriting crucial data.

The last great function that pandas provides is to perform bulk cuts on the dataframes, based on specific variables. Utilizing the following operator *newdataframe = dataframe[dataframe["variables"] > 2]* makes it so that newdataframe will be a copy of dataframe, but only events where our variable is bigger than 2 will be copied.