Trigger Upgrades

2019 4.55 GeV Run

Software Upgrades

Introduction

- Omar wrote a bump hunter code for the 2015 analysis.
  - Sebouh later did a separate analysis of the 2016, but only 10%.
  - Also, RooFit was used, and Omar has identified issues with this.

- We now want to improve upon Omar's code and perform a full analysis of the 2016 data set.
  - We aim to improve the capabilities of the bump hunter.
  - We also need to update it for the 2016 data set.

Technical Difficulties

- There have been a number of technical challenges in getting the bump hunter running.
- Version issues:
  - There are presently four versions of the bump hunter.
  - The first one trialed turned out to be a working version that didn't actually work right.
  - The second turned out to be an old version that still used RooFit.
  - The third version was the correct one, and is now currently in use.
  - The fourth version is the working copy for the upgrades.

Technical Difficulties

- There have been a number of technical challenges in getting the bump hunter running.
- Version issues:
  - There are presently four versions of the bump hunter.
  - The first one trialed turned out to be a working version that didn't actually work right.
  - The second turned out to be an old version that still used RooFit.
  - The third version was the correct one, and is now currently in use.
  - The fourth version is the working copy for the upgrades.

- We should implement better versioning controls moving forward!

Technical Difficulties

- There have been a number of technical challenges in getting the bump hunter running.
- Other issues:
  - Some compile issues with ROOT/compiler versioning.
  - Infinite loop in upper limit calculation method.
    - This is probably working code – should be fixed but can be disabled for now to keep moving forward.

Technical Difficulties

- There have been a number of technical challenges in getting the bump hunter running.
- Other issues:
  - Some compile issues with ROOT/compiler versioning.
  - Infinite loop in upper limit calculation method.
    - This is probably working code – should be fixed but can be disabled for now to keep moving forward.

- These issues have been effectively resolved.
  - The code now runs successfully on 2016 data on JLab.
  - Modifications are underway!

Improvements

- Some improvements have already been made to the code.

- **Code Runs on 2016 Data:** The biggest step forward – the code works for 2016 data.

- **Mass Parameterization:** Sebouh's 2016 data mass parameterization has been implemented.
  - We will want to redo this before the final analysis, but this gives us something from 2016 to use.

Improvements

- **Structural Improvements:** As the code is modified, it is important to ensure that is built for maintainability and so that it can be taken over easily by subsequent students.
  - Documentation is important! Omar was pretty decent about this, but it can always be made better. The code should be clearly understandable to subsequent code managers to avoid unnecessary delays.
  - Design improvements: Some areas of the code design can be improved to remove repeated code segments, overloaded/nested classes, and so on. This makes it easier to modify the code successfully.
  - Switches: Ideally, we can swap improvements in and out easily to test the effects of these changes (or to revert to the original functionality if needed).

Improvements

- **Crystal Ball Fit Function:** It was decided that a crystal ball function is a better choice than a Gaußian for the purpose of fitting the signal.
  - A crystal ball function has been written and integrated into the code base.
  - The fit function code has been granted its own package, abstracted, and generally cleaned.
  - A switch has been added to allow for easy swapping between the original Gaußian function and the crystal ball function.

- This step is on-going – a final switch is needed to enable flipping between the functions at the command line and then the fit needs to be tested to ensure that it works correctly.

Conclusions

- There were some significant delays initially with getting the code up to a state where it could be run and modified.
  - This has been resolved, and modifications are now on-going successfully for 2016 data!

- New and improved functionality is being added, along with structural improvements to enable better future maintainability.

Conclusions

- There's still a lot to do, though!

Task	Subtask	Comment	Document
MC and Data agreement	☐ Check the tracking efficiency (Matt, Omar) ☐ Check single hit efficiency? (Matt, Omar) ☑ MG5 cross section converges? (Bradley) ☐ Any significant discrepancy between MC and data selection cut? (Matt)	Unlike to 2015 data, we see that the Normalized data is about 30% higher than the normalized MC	
Event Selection	☐ Optimize event selection cuts (Rafo) ☐ Optimize energy/momentum sum cut (Rafo? Omar?) ☐ Generate final e+e- invariant mass spectrum (Rafo)		
Determine mass resolution with the pass4 data	☐ Generate A'-beam MC at several different masses along with Moller-beam MC (Bradley). ☐ Check data-MC agreement for Moller data (Matt? Omar?) ☐ Develop a cutflow to isolate Moller peak and fit using a Crystal Ball function to extract the mass resolution (Rafo? Omar?) ☐ Isolate A' invariant mass peaks and fit each using a Crystal Ball function to extract mass resolution (Rafo) ☐ Fit A' mass resolution as a function of A' mass to obtain mass parametrization. (Rafo) ☐ Determine mass scale correction (Kyle)		
Bump Hunting	☑ Run the BumpHunter (Kyle) ☐ Use Crystal-Ball instead of the Gauss for the signal shape (Omar) ☑ Add CLs limit calculation ☐ Import mass resolution parametrization (Kyle) ☐ Optimize fitting function and window size. Requires knowledge of the mass resolution (Kyle) ☐ Incorporate mass resolution and scale systematic. ☐ Add "Pulls to exclusion limits" conversion in the BumpHunter?		
Systematics	☐ fRad (Matt) ☐ Mass resolution (Moeller mass fit, different target positions) (Rafo) ☐ fits (Kyle)		

- With the run complete, we can hopefully focus all efforts on moving forward and filling out this list!