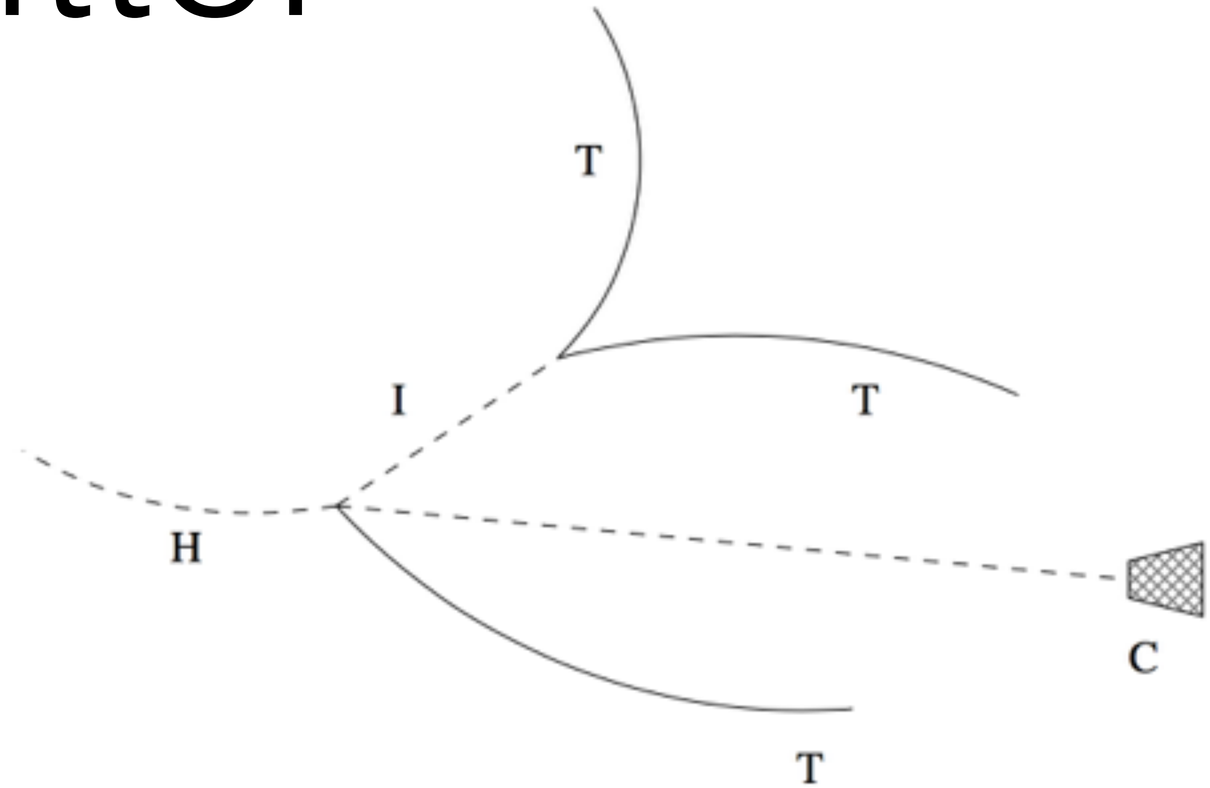
TreeFitter in basf2

Francesco Tenchini
11/01/2016

THE UNIVERSITY OF
MELBOURNE

TreeFitter



- **Progressive fit** of the decay tree to extract:

  — Particle and Vertex parameters

  — Covariance matrix for the full tree

- **Kalman approach required** because of the large amount of parameters

  — Useful for (re)fitting neutral particles

  — Analyses using full covariance matrix (i.e. Time-dependent Dalitz)

  — Can deal with missing particles

Algorithm implementation (when I started)

- Developed for BaBar* and later adapted for use in LHCb
- LHCb TreeFitter code public, but very dependant on Gaudi framework

Two options:

1. Disentangle the code from Gaudi and port it to basf2
2. Follow the same principles but use already available tools

**(W. Hulsbergen, arXiv:physics/0503191)*

Is something available?

Idea 1: RAVE

- RAVE **can perform progressive fits**, but…
- Currently **doesn't allow for multiple constraints**, though it's possible in principle
- Development inside RAVE is necessary, **not suitable** right now

Is something available?

Idea 2: ExKFitter

- ExKFitter (Extended Kinematic Fitter) is a KFitter extension that allows for **multiple constraints and simultaneous minimisation**
- **Global, not progressive**: uses Lagrange multipliers
- **Computationally intensive**, requires inversion of very large matrices
- **Not adequate** to our needs → **No alternative to porting/rewriting**

Meanwhile, at BaBar…

- At the end of the year, the BaBar collaboration approved a motion to **allow access to software** for use in other experiments
- I requested access to the original BaBar TreeFitter
- Meanwhile, I’m **converting the LHCb version** from Gaudi to basf2

Conversion work

Currently porting key classes (in particular, data objects):

- **DecayChain** — decay tree container, taken as argument by TreeFitter

- **ParticleBase** — class wrapper for the Particle data object + extra information for fitting (pointer to mother, type of particle and indexing within the decay chain, …)

- **FitParams** — treatment of fit parameters and cov. matrix

- **Constraints** — generalized constraint class

In many cases, a basf2 equivalent already exist (e.g. **Particle**) but requires checking for missing features

Future Plans

- Complete porting of the data structure (use or update already available classes when possible)
- Convert and adapt the Kalman module to basf2
- Implement secondary classes for specific particle types (resonances, missing particles, V0s…)
- Testing and validation