The MCfast Kalman Filter

Robert K. Kutschke
Fermi National Accelerator Laboratory

Presented at:

American Linear Collider Detector
Simulation Study
February 16, 1999

Outline:

- What is a Kalman Filter?
- The Genesis of this Kalman Filter.
- Assorted Details.
- Timetable.
- Summary and Conclusions.

http://www-pat.fnal.gov/personal/kutschke/alcd_990216.ps

What is a Kalman Filter

- A method to "filter" the multiple scattering "noise" out of the tracking "signal".
- In the limit of purely gaussian MS and purely gaussian measurement errors the technique produces estimators of the trajectory which:
  - are unbiased.
  - have minimum variance of all unbiased estimators.
  - have correct covariance matrix (V).
- No large matrices to invert.
- Execution time: **Linear** in n_{hit}.
- Can obtain estimators of the track parameters anywhere along the trajectory.
- For all practial purposes, the same as a "Billoir Fitter".

The Genesis of this Kalman Filter

- This Kalman filter began its life as part of MCfast, an FNAL tool for detector design
- In this environment it is robust code with well characterized results.
- The Kalman filter is being divorced from MCfast:
  - Will not use any fortran commons.
  - User supplies all info via arguments.
  - All information returned into arguments.
  - All changes are in the bookkeeping layer — the guts remain the same.
- Version for a magnetic dipole exists now.
- The version for a solenoid will be ready in days/weeks.

Aside: A plug for MCfast

- A fast, flexible tool for detector design. Original focus: B physics at the collider. Now a multi-purpose tool, including simulation of calorimetry.
- Geometry specified in a simple text file.
- User need only specify:
  - A file of events in stdhep format.
  - An MCfast geometry file.
- Output is a set of fitted tracks.
- Many tracking detector types in its library of devices.
- Comes with an integrated set of mass/vertex fitting utilities and with an Open-Inventor based graphics package.
- `http://www-pat.fnal.gov/mcfast.html` (FNAL — Computing — Physics Applications — MCFAST)

Some Details ... 1

- This Kalman filter is a "final fitter" only.
- That is, No pattern recognition.
- The caller specifies a mass hypothesis, starting values of the track parameters and a list of "nodes":
  - hits
  - scattering surfaces
  - magnetic field changes
- The filter produces:
  - The track parameters and V at the "start" and "end".
  - The same at any node.
  - The χ^2 of the fit.
  - The residual and error at any hit.
  - The residual can have local measurement in or out.
- Starting values have large radius of convergence.

Some Details . . . 2

- One may add hits to an existing track. So this could be used as the propagation step in a Kalman Filter based pattern recognition algorithm.
- One may refit a track excluding a hit by setting a flag in the relevant node and then calling the fitter.
- All memory management of the output is left to the user.
- Magnetic field must be piece-wise constant. (So a node is required at each field boundary).
- All scattering occurs in thin surfaces and only affects angles — it does not generate an offset.

Some Details . . . 3

- Support all tracking devices in:
  - CDF Run II
  - D0 Run II
  - BTeV
  - ATLAS (an early version).
- Some devices have a restricted orientation. For example forward silicon disks, (D0), must be perpendicular to z axis.
- In practice all important variations in orientation are allowed.
- No TPC, but . . .
  - . . . we can fake it on a measurement by measurement basis as a wafer in a 2-sided barrel silicon detector.
- There is no infrastructure to deal with aligning a real detector.

Some Details . . . 4

- Propagates V, not V^{-1}(following Billoir).
- Accumulated correction to trajectory is not applied until the errors are “small”. (Trajectory is linearized around the seed.)
- The bookkeeping layers and the user interface are written in c.
- The guts are written in fortran.
- No fortran common blocks are used.
- Structures are used in both languages; both .h and .inc files exist.
- Runs on OSF, IRIX, SunOs. Port to Linux is underway. Needs some minor work to run under AIX.

First Exercises with the Kalman Filter

- Study realistic resolution functions:
  - Generate hits using a realistic non-gaussian resolution function.
  - Fitter uses gaussian resolution functions.
- Pseudo Pattern Recognition.
  - Look along the trajectory and collect incorrect hits in the neighbourhood of the true hits.
  - Add some incorrect hits to the tracks.
  - Study how these incorrect hits affect vertexing, momentum measurement etc.
- These sorts of studies are a good way to discover where the true hard problems lie.

Summary and Conclusions

- In its original environment, this Kalman filter is mature and robust.
- The bookkeeping layer is now being rewritten to make it independent of MCfast.
- The work can be finished in less than a week.
- I will help new users integrate this Kalman filter into their existing software environment. Worst case estimate: a few days.