Overview of post-processing with GLOBK

T. A. Herring M. A. Floyd R. W. King

Massachusetts Institute of Technology, Cambridge, MA, USA

UNAVCO Headquarters, Boulder, Colorado, USA
19–23 June 2017

http://web.mit.edu/mfloyd/www/courses/gg/201706_UNAVCO/

Material from R. W. King, T. A. Herring, M. A. Floyd (MIT) and S. C. McClusky (now at ANU)

GLOBK overview

- Here we review the main features of `globk` and `glred`
  - Program flow
  - Kalman filtering
  - GLOBK files and estimation rules
  - `glorg` program/function: Define origin plus other things
  - Output options
  - Flow of programs

GLOBK purpose

- GLOBK is a suite of programs designed to combine geodetic results together. GPS phase processing can take a considerable time and GLOBK provides a fast method for make large network solutions, combining many days to years of data together and studying alternative parameterization and reference frames for the velocities of sites.
- GLOBK uses as data input, quasi-observation files called binary h-files which contain geodetic solutions with loosely constrained full covariance information.   These files can generated from GAMIT solutions or SINEX files.
- GLOBK is a smoothing Kalman filter and can incorporate random walk process noise in its estimation (method for accounting for temporally correlated noise in time series).
- Its two main uses are to generate velocity field estimates and time series in a well-defined and often different reference frames. (It can also be used to merge large networks of GPS sites).

Common applications of GLOBK

- Repeatability analysis (`glred`)
  - individual sessions
  - combine regional and global files for orbit control and reference frame (orbit control is not so important anymore; IGS orbits are very good apriori)
- Combine sessions to get average position over survey
  - connects stations observed separately
  - reduces number of h-files to be used for velocities
- Combine averaged positions to estimate velocities
  - and/or earthquake offsets and post-seismic motion
- When `globk` is run in parallel in the same directory, care should be used in scratch file names (discussed later)

Processing stages

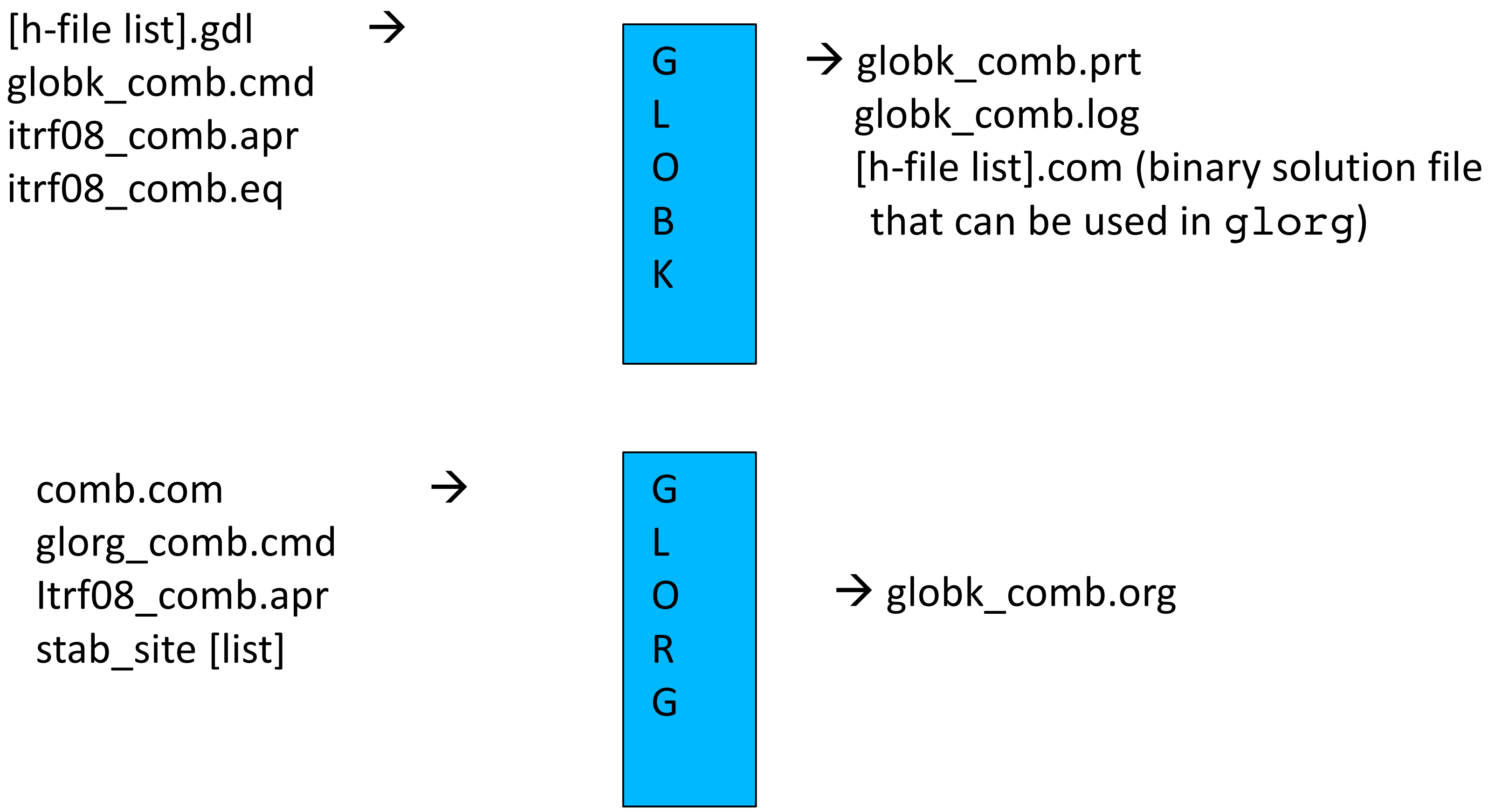
- `htoglb` generates input binary h-files.
- GLOBK has distinct modules that are used:
  - Initialization: Header information from the binary h-files are read to determine the nature of the solutions i.e., parameters in the state vector, all site names accounting for offsets and earthquakes etc.
  - Forward Kalman filter: Binary h-files are combined to generate solution. Normally most parameters are loosely constrained.
  - Possible backwards smoothing filter (not that common)
  - Simple output of the solution (program `glout`: generates .prt file)
  - Reference frame realized solution and post-solution constraints. Generates .org file. Program `glorg` can be used separately for multiple realizations and constraints.
  - Saving binary version of solution for additional processing ("out_glb" option in command file and `glsave` program).
- Modules in GLOBK can be called within GLOBK as subroutines or run externally as stand-alone programs (program names are lower case)

Using non-GAMIT files in `htoglb`

- `htoglb` can convert other types of solution/covariance matrix files into GLOBK binary h-files.
- There are caveats when this is done because these other file formats don't contain the same meta data as the GAMIT h-files.
- For GAGE/PBO Frame resolved SINEX files:
  - Use the -d=TR option to apply rotation and translation loosening
  - -m=512 (allocate 512 Mb memory) is needed from more recent files
  - There are ”loose” SINEX files for which -d=TR is not needed except for CWU files which are not loose (these are the submitted AC files)
- IGS SINEX files
  - Use -s option for name translation with point codes
  - Variance-covariance matrix will need scaling depending on AC (scaling value supplied in .gdl file). COD solution of more difference from unity (~10^{-4})

GLOBK function and file flow

`htoglb`: Translate GAMIT h-files to (e.g., hemeda.10256) to GLOBK
h-files (e.g. h1009131200_emed.glx



[h-file list].gdl →
globk_comb.cmd
itrf08_comb.apr
itrf08_comb.eq

GLOBK

→ globk_comb.prt
globk_comb.log
[h-file list].com (binary solution file that can be used in `glorg`)

comb.com →
glorg_comb.cmd
Itrf08_comb.apr
stab_site [list]

GLORG

→ globk_comb.org

Names of files here can be chosen arbitrarily.

GLOBK command files

- GLOBK is controlled by a command file that "instructs" the program what to do.
- The command file contain the following classes of commands:
  - Estimation command: Tells `globk` what to estimate and constraints on apriori values and temporal behavior of the parameters. "apr_xxx" and "mar_xxx" commands.
  - a priori information commands: Coordinates, discontinuity times, selection of sites
  - Output (types and files), and control commands (e.g., to run `glorg`)
- GLORG (post-processing program/module) also has its own command file.
- The simplest globk command can have one line:
  apr_neu all 10 10 10 0 0 0
  but in general have several other commons commands (see examples in ~/gg/tables/globk_xxxx.cmd and ~/gg/tables/glorg_xxxx.cmd

GLOBK file name conventions

- GLOBK uses arbitrary file names but there are some conventions used:
  - Binary h-files from htoglb: .glx is bias fixed, .glr is bias free (normally not used)
  - List of binary h-files to process: .gdl extent
  - GLOBK and GLORG command files: globk_<type>.cmd and glorg_<type>.cmd
  - Output files: print file (no glorg reference frame) .prt (often not output); glorg output .org; log file .log
  - Apriori coordinate files: .apr
  - Earthquake and rename file: .eq
  - Lists of stabilization sites (used with source command): .stab

Kalman filtering

- Equivalent to sequential least-squares estimation but allowing for stochastic processes, usually a 1st-order Gauss-Markov process
- GLOBK allows a random walk for coordinates, EOP, network translation and scale, and satellite parameters; variance grows linearly with time
- Because a Kalman filter works with covariance matrices (rather than normal matrices), all parameters must have a priori constraints (usually loose)

- See Herring et al. (1990) and Dong et al. (1998) for a more thorough description as applied to geodetic analysis

GLOBK structural confusions

- `globk` and `glred` are the same program with (slightly) different ways of treating the h-file (.gdl) list:
  - `globk`: all h-files in combined in a single solution
  - `glred`: each h-file generates a separate solution (unless followed by a "+" in .gdl-file). `glred` is a small program that generates sub-set .gdl-files and runs `globk`.
- Two types of solution files:
  - h-files for saving and external exchange (backward compatible)
  - com/sol file is internal, format changes with versions
- `glorg` called by `globk/glred` or run separately to apply generalized constraints to solution and estimate plate rotations. "com_file" command must be used in `globk` command file for `glorg` to run by itself.

GLOBK files

- User supplied
  - command files (may include “source” files)
  - .gdl list of h-files
  - binary h-files (created from SINEX or GAMIT h-files)
  - .apr-file(s) (optional but recommended)
  - EOP (in_pmu file, optional but recommended)
  - eq_file (optional, but must appear at top)
- Generated by `globk`
  - .srt, .com, .sol , .svs (all except .sol must be named and commands need to be top of GLOBK command file)
- Output files
  - screen, log, prt, org and output h-file

GLOBK file handling

- log-, prt-, org-files are concatenated, so should be removed or renamed unless you want them together (e.g. `glred`). The “eras” option can be used in the “prt_opt” and “org_opt” command in the GLOBK command file to erase these files (should not be used with `glred`).
- com-, srt-, sol-files are overwritten; com/sol should not be renamed since the original sol-file name is imbedded in the com-file.
- Automatic naming using wild-cards is available for com, srt, sol, org, and output h-files (i.e., name used depends on name of gdl-file; needed for parallel processing.)

Estimation commands rules

- For a parameter to estimated in GLOBK, apr_xxx command must be used where xxx is a parameter type (e.g., neu, svs, wob, ut1, atm)
- If a parameter is not mentioned, it does not appear in the solution, but if it appears in the h-file (i.e, estimated in GAMIT), its uncertainty is implicit in the GLOBK solution; e.g., if orbits are estimated in GAMIT and you want them constrained in globk, use apr_svs. If apr_svs is not used, orbits are left constrained.
- If zero given as a priori sigma, then parameter is not estimated (effectively left unconstrained)
- To force a parameter to its a priori value, use "F" as the a priori sigma
- Parameters estimated in `glorg` must be kept loose in `globk` ; if rotation or scale is not estimated in glorg, it must be tightly constrained in globk

Earth orientation parameters (EOPs)

- Normally used in two forms:
  - Global network of stations (allows rotation in `glorg`)

    ```
    apr_wob 10 10 1 1
    apr_ut1 10 1
    ```

  - Regional network (constrained). When constrained this way system is not free to rotate so xrot, yrot, zrot should not be used in "pos_org" command (see "pos_org" in `glorg`)

    ```
    apr_wob  .2  .2   .02   .02
    apr_ut1  .2  .02
    ```

- In many analyses, the global form is used even for regional networks in order to allow rotation estimation in `glorg`
  - Care is needed if network is not surrounded by stations with well defined motions

Data editing

- To account for temporal correlations in time series we typically use random-walk (RW) process noise with the "mar_neu" command (units m^2/yr)
- Typical values are 2.5 × 10^{-8} m^2/yr (0.5 mm in 1 yr) to 4 × 10^{-6} m^2/yr (2 mm in 1 yr)

```
mar_neu all   2.5e-8 2.5e-8 2.5e-8  0 0 0
mar_neu chdu  4e-6   4e-6   4e-6    0 0 0
```

- `sh_gen_stats` can used to generate process noise estimates provided sufficiently large number of position estimates are available
- To down-weight noisy segments or equalize continuous and survey-mode data in a combined h-file, can add random noise (units are m)

```
sig_neu all  .001 .001 .003
sig_neu ankr .005 .005 .020  2002 10 01 00 00 2002 11 30 24 00
sig_neu   EMED0504   .010 .010 .1
```

- To remove an outlier, can down-weight severely or rename (in eq_file)

```
sig_neu ankr  .1 .1 .1  2002 10 01 00 00 2002 10 01 24 00
rename ankr_gps ankr_xcl 2002 10 01 00 00 2002 10 01 24 00
```

glorg

- Invoked by globk to apply generalized constraints after h-files are stacked and loose solution performed
  - Can be run as a separate program using the com/sol files from globk
- Also allows linking of parameters and estimation of Euler poles
- Parameters estimated in glorg must be kept loose in globk
  - Site coodinates
  - EOPs (for estimation of rotation)
  - Scale
- glorg is used to define and refine the reference frame for globk solutions

Invoking `glorg` from `globk` command file

- The `globk` command file contains commands that cause `glorg` to run when `globk` completes the solution combination:
  - org_cmd < `glorg` command file name > ---invokes `glorg`
  - org_opt < options for output >
  - org_out < output file name >: Normally not used because name will be generated from prt file name in the `globk` runstring
- If "org_out" is not given then the extent on the print file name is replaced with ".org"

glorg commands

- apr_file – Need not be the same as for globk; needs to contain values only for sites used for stabilization and sites for which coordinates or velocities are equated
- pos_org, rate_org – Control what parameters are estimated in stabilization
  - xtran ytran ztran – allows translation (apr_tran in globk if GAMIT "BASELINE" choice of experiment)
  - xrot yrot zrot – allows rotation
  - scale – allows rescaling of system (if used, estimate scale in globk; apr_scale and possibly mar_scale)
- cnd_hgtv – Control relative weights of heights (variances, nominally 10 but increasing value will reduce heights effecting horizontal position estimates; 1000 is good for this)
- stab_ite – # of iterations and sigma-cutoff to remove a site
- stab_site – List of sites to use in stabilization

Controlling print output

- crt_opt, prt_opt, org_opt specify output options for screen, print and org files
- `globk/glorg` help gives all options, main ones are:
  - ERAS -- erase file before writing (normally files appended)
  - NOPR -- Do not write output (e.g., for globk when invoking glorg)
  - BLEN -- Baseline lengths
  - BRAT -- baseline rates when velocities estimated
  - RNRP -- generates reports on differences in parameter estimates after renames.
  - FIXA -- makes apriori coordinates and velocities consistent when equates are used in glorg (can sometimes fail in complicated rename scenarios--best if apr_file is provided with consistent values)
  - VSUM -- Lat/long summary of velocity (needed to plot velocities)
  - PSUM -- Lat/long position summary
  - GDLF --Include list of hfiles and chi**2 increments from run
  - CMDS -- Echos globk command file into output file

Program flow

- Read all the h-file headers to determine their contents (sites, other parameters, epoch range)
- Apply renames as requested in the eq_file
- Sort the h-file list forward or backward in time (srt_dir)
- Initialize the Kalman filter with the a priori constraints (apr_xxx)
- Read in the h-files, one at a time, a run sequential Kalman Filter. Compute the chi2 increment, coordinate adjustment, and rotation implied by the new data; if within tolerance (max_chii), update the solution and write the χ^2 increment to the log file
- Write the solution to the sol_file and prt file (and optionally to a new h-file)
- Optionally invoke `glorg` to apply generalized constraints
  - Apply the constraints (iterative "stabilization")
  - Apply linkage of parameters (equate, constrain, force), computing the χ^2 increment for each
  - Estimate plate rotations ("plate" command)
  - Write the solution to the org file (`glorg` prt file)

Things GLOBK cannot do

- Repair mistakes in original analysis
  - Cycle slips
  - Wrong antenna phase center models
- Resolve ambiguities
  - Would make files too large
- Overcome non-linear effects
  - As in GAMIT, adjustments must be less than ~ 30 cm
- But GLOBK can delete stations
  - Can help avoid contaminating solution

apr files in GLOBK processing

- GAMIT
  - 10 m accuracy for all sites for cycle-slip repair
  - < 30 cm final adjustment for linearity (1st solution guarantees)
  - ~5 cm accuracy in constrained site(s) for ambiguity resolution
- `globk`
  - If invoking `glorg` for reference frame, apr_file usually optional in `globk`
  - If not invoking `glorg`, need accurate apr_file entries for constrained sites
  - For complicated renames and equates, apr_file may be needed in `globk`
- `glorg`
  - apr_file needs coordinates only for reference sites and equates

What can go wrong?

- `globk`
  - h-files not used: removed automatically for high χ^2, coordinate adjustment, or rotation (max_chii command)
  - High χ^2 increment: inconsistent data. Can be an issue when estimating orbits ("RELAX" mode) if MIT GLX file use different modelling (e.g. albedo, gravity field)
  - Station "missing": not present in h-file or renamed out (use `glist`)
- `glorg`
  - Stabilization fails: too few sites in stabilization
  - Large uncertainties: poor stabilization
  - Uncertainties too small for some stabilization sites: rotation parameters absorbing coordinate adjustment
  - High χ^2 in equate: inconsistent data
  - Wrong velocity for equated sites: unmatched a priori

Associated programs

- `htoglb`: Translates various ascii solution files into GLOBK h-files (GAMIT h-files, SINEX)
- `glbtosnx`: Generates SINEX files from binary h-files
- `glist`: Lists the contents of a series of h-files
- `hfupd`: Updates binary h-files for changes in station.info or sinex header file (distributed by IGS)
- `tssum`, `tsfit`, `tscon`: Time series analysis (batch)
- Matlab-derived programs (interactive):
  - velview: displays and analyzes velocity fields
  - tsview: displays and analyses time series

Summary

- GLOBK has many features and due to its evolution, there are often multiple ways of doing the same or similar things
- There is extensive help in the ~/gg/help/ directory and discussion in the documentation
- GLOBK is where all the major analysis decisions are made and hence can be quite complex for large analyses
- Experimentation and testing your ideas of how different options effect the results is one the best ways to learn the software, e.g.
  - What happens to position/velocity estimates if the "apr_tran" command is added to the `globk` command file?
  - How do my estimates and uncertainties change if the "apr_neu" and "mar_neu" commands are changed?