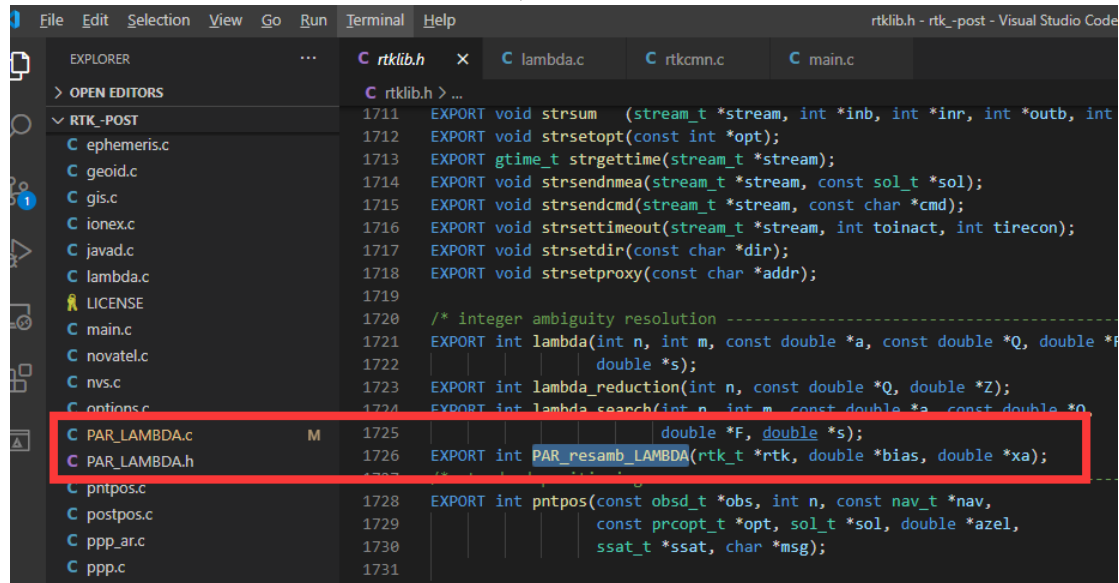


Implementation of Partial ambiguity resolution (PAR) based on decorrelation transformation (Wang and Feng 2013). It is programmed by C and Matlab language. The C version is based on [RTKLIB](#) and compatible with it. While Matlab version is based on [LAMBDA](#), which is for debugging and comprehending.

An example is provided in **mytest.m**.

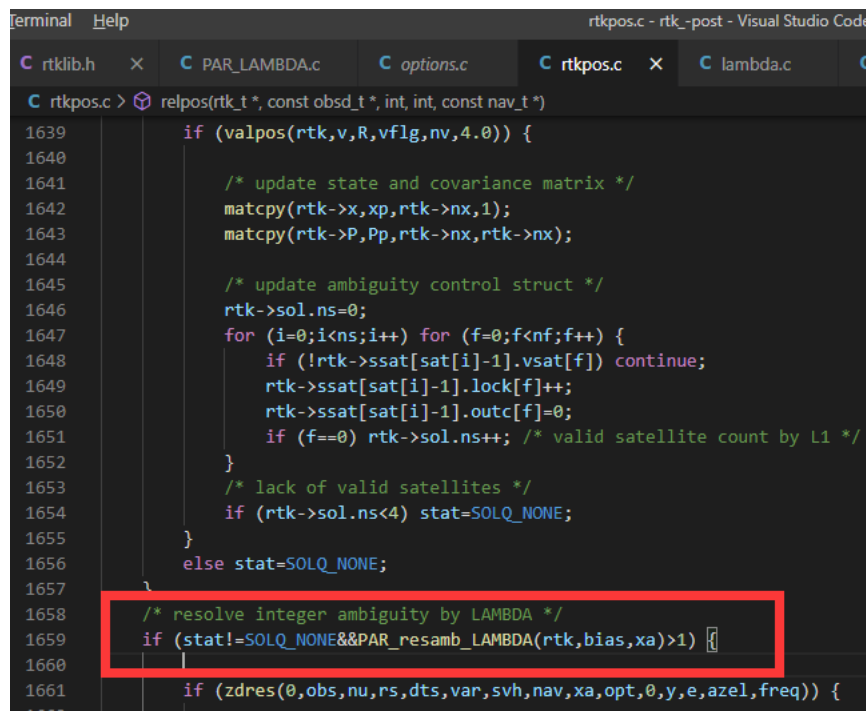
1. If you want to use the C version, you should put **PAR_LAMBDA.c** and **PAR_LAMBDA.h** to RTKLIB. Then, add a function declaration in **rtklib.h**



```

1711 EXPORT void strsum (stream_t *stream, int *inb, int *inr, int *outb, int
1712 EXPORT void strsetopt(const int *opt);
1713 EXPORT gtime_t strgettime(stream_t *stream);
1714 EXPORT void strsendmea(stream_t *stream, const sol_t *sol);
1715 EXPORT void strsendcmd(stream_t *stream, const char *cmd);
1716 EXPORT void strsettimeout(stream_t *stream, int toinact, int tirecon);
1717 EXPORT void strsetdir(const char *dir);
1718 EXPORT void strsetproxy(const char *addr);
1719
1720 /* integer ambiguity resolution -----
1721 EXPORT int lambda(int n, int m, const double *a, const double *Q, double *F, double *s);
1722 EXPORT int lambda_reduction(int n, const double *Q, double *Z);
1723 EXPORT int lambda_search(int n, int m, const double *a, const double *Q,
1724                          double *F, double *s);
1725 EXPORT int PAR_resamb_LAMBDA(rtk_t *rtk, double *bias, double *xa);
1726
1728 EXPORT int pntpos(const obsd_t *obs, int n, const nav_t *nav,
1729                  const prcopt_t *opt, sol_t *sol, double *azel,
1730                  ssat_t *ssat, char *msg);
1731
  
```

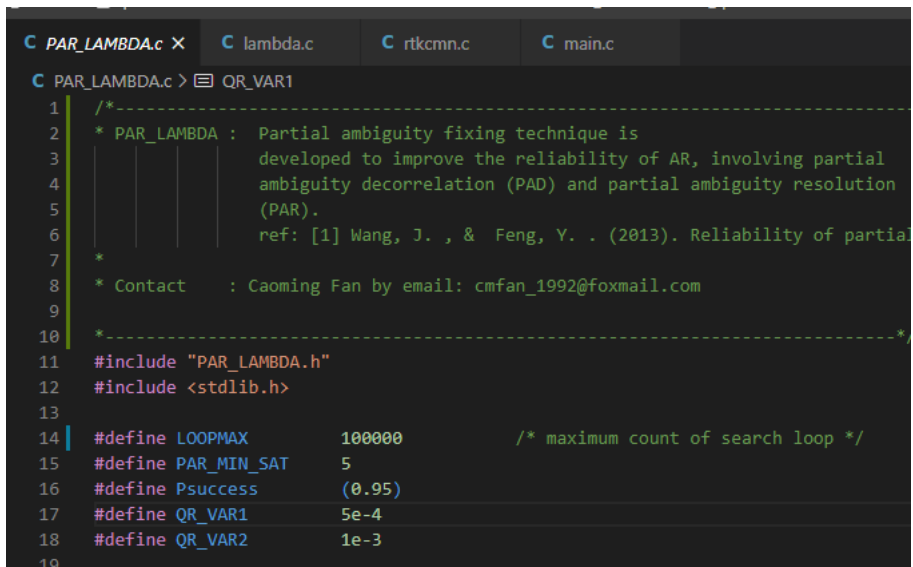
2. Call it in **rtkpos.c**.



```

1639 if (valpos(rtk,v,R,vflg,nv,4.0)) {
1640
1641     /* update state and covariance matrix */
1642     matcpy(rtk->x, xp, rtk->nx, 1);
1643     matcpy(rtk->p, Pp, rtk->nx, rtk->nx);
1644
1645     /* update ambiguity control struct */
1646     rtk->sol.ns=0;
1647     for (i=0; i<ns; i++) for (f=0; f<nf; f++) {
1648         if (!rtk->ssat[sat[i]-1].vsat[f]) continue;
1649         rtk->ssat[sat[i]-1].lock[f]++;
1650         rtk->ssat[sat[i]-1].outc[f]=0;
1651         if (f==0) rtk->sol.ns++; /* valid satellite count by L1 */
1652     }
1653     /* lack of valid satellites */
1654     if (rtk->sol.ns<4) stat=SOLQ_NONE;
1655 }
1656 else stat=SOLQ_NONE;
1657
1658 /* resolve integer ambiguity by LAMBDA */
1659 if (stat!=SOLQ_NONE&&PAR_resamb_LAMBDA(rtk,bias,xa)>1) {
1660
1661     if (zdres(0, obs, nu, rs, dts, var, svh, nav, xa, opt, 0, y, e, azel, freq)) {
1662
  
```

3. You can also adjust some parameters to make it optimal.



```
C PAR_LAMBDA.c X C lambda.c C rtkcmn.c C main.c
C PAR_LAMBDA.c > QR_VAR1
1  /*-----
2  * PAR_LAMBDA : Partial ambiguity fixing technique is
3  | | | | | developed to improve the reliability of AR, involving partial
4  | | | | | ambiguity decorrelation (PAD) and partial ambiguity resolution
5  | | | | | (PAR).
6  | | | | | ref: [1] Wang, J. , & Feng, Y. . (2013). Reliability of partial
7  *
8  * Contact : Caoming Fan by email: cmfan_1992@foxmail.com
9
10 *-----*
11 #include "PAR_LAMBDA.h"
12 #include <stdlib.h>
13
14 #define LOOPMAX 100000 /* maximum count of search loop */
15 #define PAR_MIN_SAT 5
16 #define Psuccess (0.95)
17 #define QR_VAR1 5e-4
18 #define QR_VAR2 1e-3
19
```

Any question? Please Email cmfan_1992@foxmail.com

Wang J, Feng Y (2013) Reliability of partial ambiguity fixing with multiple GNSS constellations. Journal of Geodesy 87: 1-14. doi: 10.1007/s00190-012-0573-4