

WMH602 标准版惯性卫星组合导航开发板

版本 20250518

1. 概述

1.1. 功能

开发板的主要功能是演示惯性卫星组合导航。

开发板包含 2 块电路板，层叠堆栈连接。第一块电路板（绿色）包括：处理器 STM32F765VIH6，惯性传感器 ADIS16505-2。第二块电路板（蓝色）为卫星导航接收机。开发板赠送的附件有：卫星导航天线、串口-USB 转换器。

开发板提供示例代码。

主要特点：

- (1) 输出惯性传感器原始角速度、加速度，输出卫星导航位置，输出组合导航欧拉角、速度、位置。
- (2) 扩展卡尔曼滤波 EKF、ESKF 方法。
- (3) 采用 double 型双精度浮点数 FPU 计算。
- (4) RS485 接口。
- (5) 支持 UsartGPU 显示屏（另行购买）直接显示。

代码包适用于：组合导航的学习、研究、教学、科研。商业用途请联系本店另行定作。



1.2. 部分代码截图

（由于版本迭代，实际代码可能与截图有轻微差别）

```

474  /* USER CODE BEGIN WHILE */
475  while (1)
476  {
477      /* USER CODE END WHILE */
478
479      /* USER CODE BEGIN 3 */
480      if(adis_dr)
481      {
482          adis_dr=0;
483          read_adis();
484
485          if (gnss.update > 0)
486          {
487              gnsstrans();
488              gnsrecord();
489          }
490
491          send_ul();
492          calc();
493          if(navistate==2)
494          {
495              HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_SET);
496          }
497      }
498  }
499  }
500  }
501  /* USER CODE END 3 */
502  }

```

```

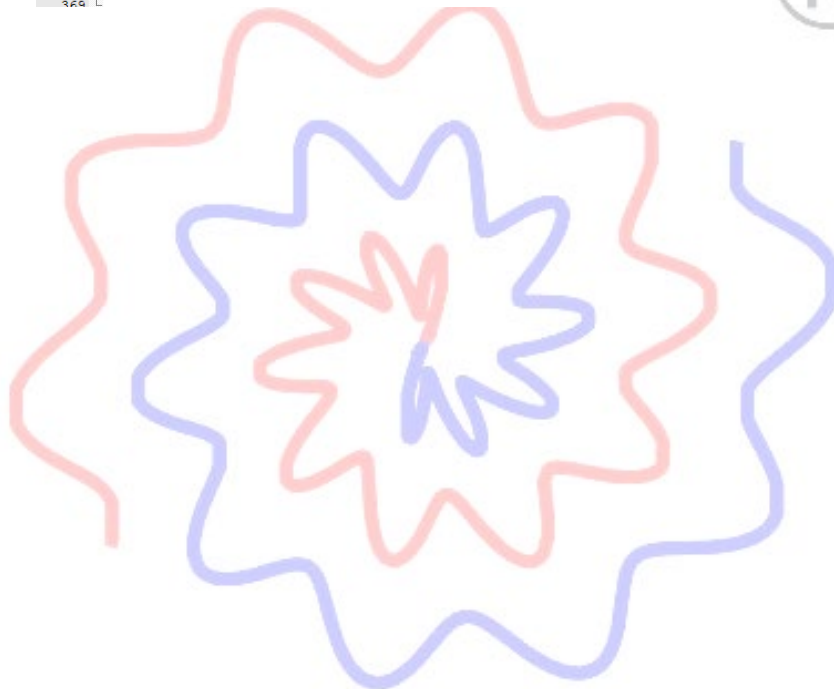
230  //根据位置更新地球模型
231  earthmodelupdate();
232
233  //准备，补偿传感器误差，注意方向
234  accel=op_AaddB(accel,biasacc);
235  gyro=op_AaddB(gyro,biasgyro);
236
237
238  //一、计算姿态
239  wien=wien();
240  wenn=wenn();
241  cbn=cbn(qa);
242  wmbb=op_AsubB(gyro,op_AB(op_AT(cbn),op_AaddB(wien,wenn))); //扣除地
243  qa=quatupdate(qa,op_kA(dTins,wmbb)); //更新姿态
244
245
246  //二、计算速度
247  acon=op_AB(cbn,accel); //更新这个数，以便于卡尔曼滤波的部分使用
248  gn=matinit(3,1,0);
249  gn.num[2][0]=(-ge);
250  an=op_AaddB(op_AsubB(acon,op_AcrossB(op_AaddB(op_AaddB(wien,wien),ve
251  tspeed=op_AaddB(tspeed,op_kA(dTins,an)); //更新速度
252
253  //三、计算位置
254  dpos=matinit(3,1,0);
255  H=tpos.num[2][0];
256  dpos.num[0][0]=tspeed.num[1][0]/(Rmeri+H); //北向速度得到纬度
257  dpos.num[1][0]=tspeed.num[0][0]/((Rprim+H)*cos(tpos.num[0][0])); //东
258  dpos.num[2][0]=tspeed.num[2][0];
259  tpos=op_AaddB(tpos,op_kA(dTins,dpos)); //更新位置
260
261  pushposrecord();
262
263
264

```

```

335
336 MAT solve(MAT Z)
337 {
338     MAT Pkk=op_AaddB(op_AB(op_AB(Phi,Pk),op_AT(Phi)),Q);
339     MAT K=op_ArightB(op_AB(Pkk,op_AT(H)),op_AaddB(op_AB(op_AB(H,Pkk),op_AT(H)),R));
340     MAT X=op_AB(K,Z); //因为每次滤波之后补偿了误差, 所以状态预测总是0.
341     MAT IKH=op_AsubB(Eye,op_AB(K,H));
342     Pk=op_AaddB(op_AB(op_AB(IKH,Pkk),op_AT(IKH)),op_AB(op_AB(K,R),op_AT(K)));
343
344     return X;
345 }
346
347
348
349 void sat(double lati,double longi,double height)
350 {
351     MAT Z=matinit(3,1,0);
352     MAT X;
353     Z.num[0][0] = posrecord[0][0] - lati;
354     Z.num[1][0] = posrecord[0][1] - longi;
355     Z.num[2][0] = posrecord[0][2] - height;
356
357
358
359     X=solve(Z);
360     Phi = matinit(15, 15, 1);
361     Q=matinit(15,15,0);
362
363     tpos=op_AsubB(tpos,submat(X,0,0,3,1));
364     tspeed=op_AsubB(tspeed,submat(X,3,0,3,1));
365     qa=quatupdate(qa,op_AB(op_AT(Cbn(qa)),submat(X,6,0,3,1)));
366     biasgyro=op_AsubB(biasgyro,submat(X,9,0,3,1));
367     biasacc=op_AsubB(biasacc,submat(X,12,0,3,1));
368 }
369

```



```

414
415 //位置对位置影响的子矩阵
416 Fpp.num[0][2]=RmH1*RmH1*(-vN); //高度对纬度的影响, 与北向速度有关
417 Fpp.num[1][0]=RpH1*vE*secphi*tanphi; //纬度对经度的影响, 与纬度有关, 与东向
418 Fpp.num[1][2]=RpH1*RpH1*(-vE)*secphi; //高度对经度的影响, 与东向速度有关,
419 fillsubmat(&F,0,0,Fpp);
420
421 //速度对位置影响的子矩阵
422 Fvp.num[0][1]=RmH1; //北向速度对纬度影响
423 Fvp.num[1][0]=RpH1*secphi; //东向速度对经度的影响
424 Fvp.num[2][2]=1; //天向速度对高度的影响
425 fillsubmat(&F,0,3,Fvp);
426
427
428 //位置对速度影响的子矩阵
429 Fpv.num[0][0]=2*we*cosphi*vN+2*we*sinphi*vU+vN*vE*RpH1*secphi*secphi; //纬
430 Fpv.num[0][2]=RpH1*RpH1*(vE*vU-vN*vE*tanphi); //高度对东向速度的影响
431 Fpv.num[1][0]=(-2*vE*we*cosphi+vE*vE*RpH1*secphi*secphi); //纬度对北向速
432 Fpv.num[1][2]=RmH1*RmH1*vN*vU+RpH1*RpH1*vE*vE*tanphi; //高度对北向速度的影
433 Fpv.num[2][0]=(-2.0)*vE*we*sinphi; //纬度对天向速度的影响
434 Fpv.num[2][2]=(-RmH1*RmH1*vN*vN-RpH1*RpH1*vE*vE); //高度对天向速度的影响
435 fillsubmat(&F,3,0,Fpv);
436
437 //速度对速度影响的子矩阵
438 Fvv.num[0][0]=(vN*tanphi-vU)*RpH1; //东向速度对东向速度的影响
439 Fvv.num[0][1]=2.0*we*sinphi+vE*RpH1*tanphi; //北向速度对东向速度的影响
440 Fvv.num[0][2]=(-2.0)*we*cosphi-vE*RpH1; //天向速度对东向速度的影响
441 Fvv.num[1][0]=(-2.0)*(we*sinphi+vE*RpH1*tanphi); //东向速度对北向速度的影响
442 Fvv.num[1][1]=(-RmH1)*vU; //北向速度对北向速度的影响
443 Fvv.num[1][2]=(-RmH1)*vN; //天向速度对北向速度的影响
444 Fvv.num[2][0]=2.0*(we*cosphi+vE*RpH1); //东向速度对天向速度的影响//有的书公
445 Fvv.num[2][1]=2*vN*RmH1; //北向速度对天向速度的影响
446 fillsubmat(&F,3,3,Fvv);
447
448 //姿态对速度影响的子矩阵
449 fE=accn.num[0][0];
450 fN=accn.num[1][0];
451 fU=accn.num[2][0];
452
453 Fav.num[0][1]=(-fU); //北向角度对东向速度的影响
454 Fav.num[0][2]=fN; //天向角度对东向速度的影响
455 Fav.num[1][0]=fU; //东向角度对北向速度的影响
456 Fav.num[1][2]=(-fE); //天向角度对北向速度的影响
457 Fav.num[2][0]=(-fN); //东向角度对天向速度的影响
458 Fav.num[2][1]=fE; //北向角度对天向速度的影响
459 fillsubmat(&F,3,6,Fav);
460
461
462 //位置对姿态影响的子矩阵
463 Fpa.num[0][2]=vN*RmH1*RmH1; //高度对东向姿态的影响
464 Fpa.num[1][0]=(-we)*sinphi; //纬度对北向姿态的影响
465 Fpa.num[1][2]=(-vE)*RpH1*RpH1; //高度对北向姿态的影响
466 Fpa.num[2][0]=we*cosphi+vE*RpH1*secphi*secphi; //纬度对天向姿态的影响
467 Fpa.num[2][2]=(-vE)*tanphi*RpH1*RpH1; //高度对天向姿态的影响
468 fillsubmat(&F,6,0,Fpa);
469
470
471 //速度对姿态影响的子矩阵
472 Fva.num[0][1]=(-RmH1); //北向速度对东向姿态的影响
473 Fva.num[1][0]=RpH1; //东向速度对北向姿态的影响
474 Fva.num[2][0]=RpH1*tanphi; //东向速度对天向姿态的影响
475 fillsubmat(&F,6,3,Fva);
476
477 //姿态对姿态影响的子矩阵
478 Faa.num[0][1]=we*sinphi+vE*RpH1*tanphi; //北向姿态对东向姿态的影响
479 Faa.num[2][0]=we*cosphi+vE*RpH1; //东向姿态对天向姿态的影响
480 Faa.num[2][1]=vN*RmH1; //北向姿态对天向姿态的影响
481 Faa.num[0][2]=(-Faa.num[2][0]); //天向姿态对东向姿态的影响
482 Faa.num[1][0]=(-Faa.num[0][1]); //东向姿态对北向姿态的影响
483 Faa.num[1][2]=(-Faa.num[2][1]); //天向姿态对北向姿态的影响
484 fillsubmat(&F,6,6,Faa);
485
486
487 fillsubmat(&F,6,9,op_kA(-1,cbn)); //陀螺仪零偏对姿态的影响。有的书把这个公
488 fillsubmat(&F,3,12,cbn); //加速度计零偏对速度的影响。
489
490

```

1.3. 矩阵计算库

为了矩阵计算, C 代码中有结构体 MAT。其内容为:

```

int m; //行数
int n; //列数
double num[MAT_MAX][MAT_MAX]; //矩阵数据内容

```

可以根据需要直接修改矩阵的数值。特别注意, C 或 C++ 中数组元素的下标从 0 开始; 而 MATLAB 的下标从 1 开始。

矩阵计算的功能已经包含在代码包中, 通常情况下不需要修改。

本代码包的矩阵运算是专门简化过的、另起炉灶的计算库, 与 Eigen、OpenCV 等常见

矩阵库不兼容。

```

16
17
18 typedef struct{
19     int m; //行数
20     int n; //列数
21     double num[MAT_MAX][MAT_MAX]; //矩阵数据内容
22 } MAT;
23
24 MAT matinit(int setm, int setn, int kind); //kind=1单位阵, kind=0零矩阵
25 MAT submat(MAT w, int a, int b, int lm, int ln);
26 void fillsubmat(MAT* w, int a, int b, MAT s);
27 MAT op_kA(double k, MAT a);
28 MAT op_AB(MAT a, MAT b);
29 MAT op_AaddB(MAT a, MAT b);
30 MAT op_AsubB(MAT a, MAT b);
31 MAT op_AT(MAT a);
32 MAT op_ArightB(MAT a, MAT b);
33 MAT op_AleftB(MAT a, MAT b);
34
35 double square(MAT w);
36 double absvec(MAT w);
37 MAT op_AcrossB(MAT a, MAT b);
38
39

```

```

272
273 MAT op_ArightB(MAT a, MAT b)
274 {
275     //高斯消元法
276
277     int x, xb;
278     double k;
279     double s;
280     int p;
281     double sxb;
282
283     for(x=0; x<b.n; x++)
284     {
285         //首先找到最佳的列。让起始元素最大
286         s=0;
287         p=x;
288         for(xb=x; xb<b.n; xb++)
289         {
290             sxb=fabs(b.num[x][xb]);
291             if(sxb>s)
292             {
293                 p=xb;
294                 s=sxb;
295             }
296         }
297         //同时变换两侧矩阵
298         if(x!=p)
299         {
300             columnexchange(&a, x, p);
301             columnexchange(&b, x, p);
302         }
303
304         //这一列归一
305         k=1/b.num[x][x]; //这一句不要嵌套到下面两行中。否则会因更新不同步导致计算错误。
306         columnmulti(&a, x, k);
307         columnmulti(&b, x, k);
308
309         //把其它列归零
310         for(xb=0; xb<b.n; xb++)
311         {
312             if(xb!=x)
313             {
314                 k=(-b.num[x][xb]);
315                 columnadd(&a, xb, x, k);
316                 columnadd(&b, xb, x, k);
317             }
318         }
319     }
320
321     return a;
322 }
323
324

```

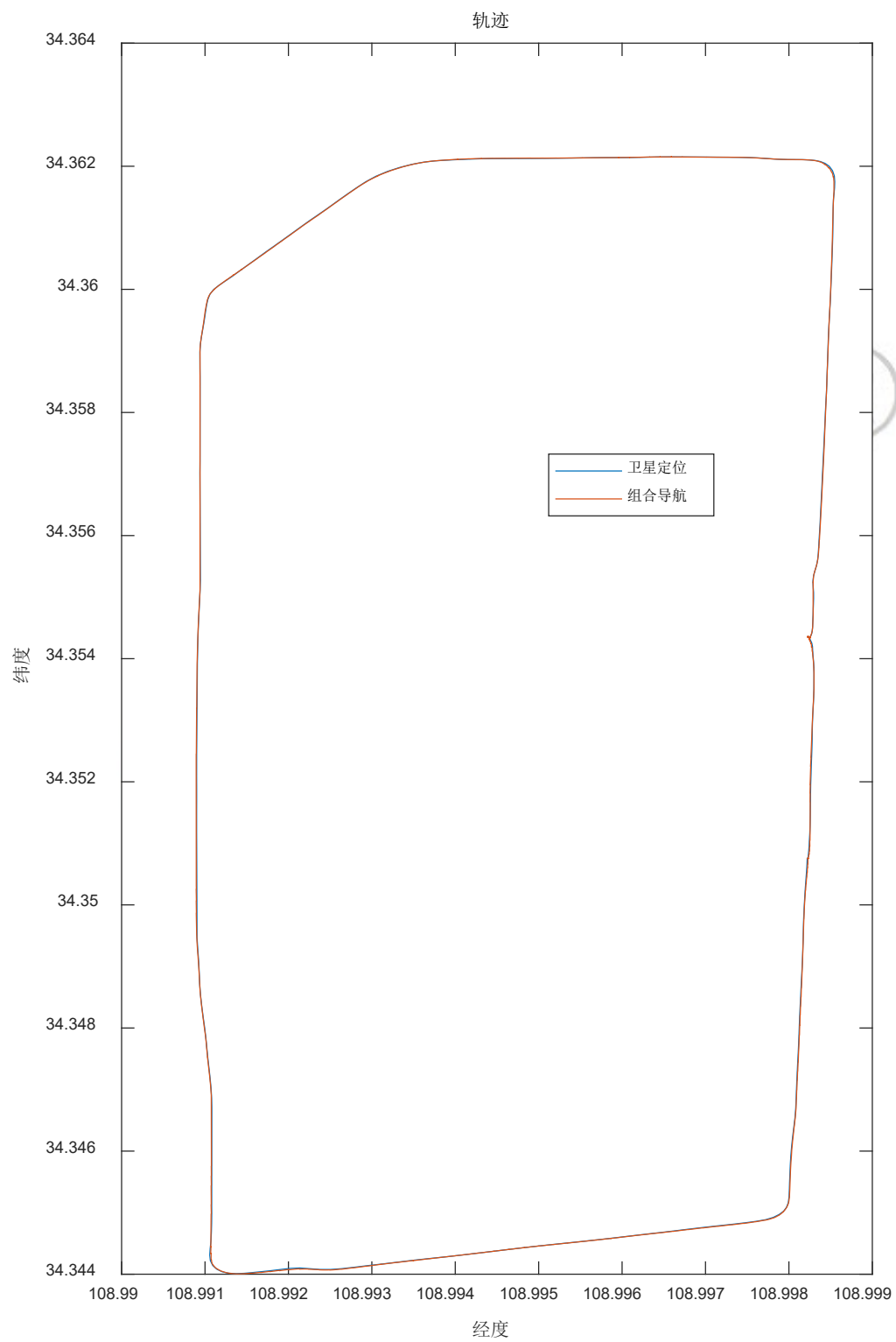
1.4. 实验数据曲线

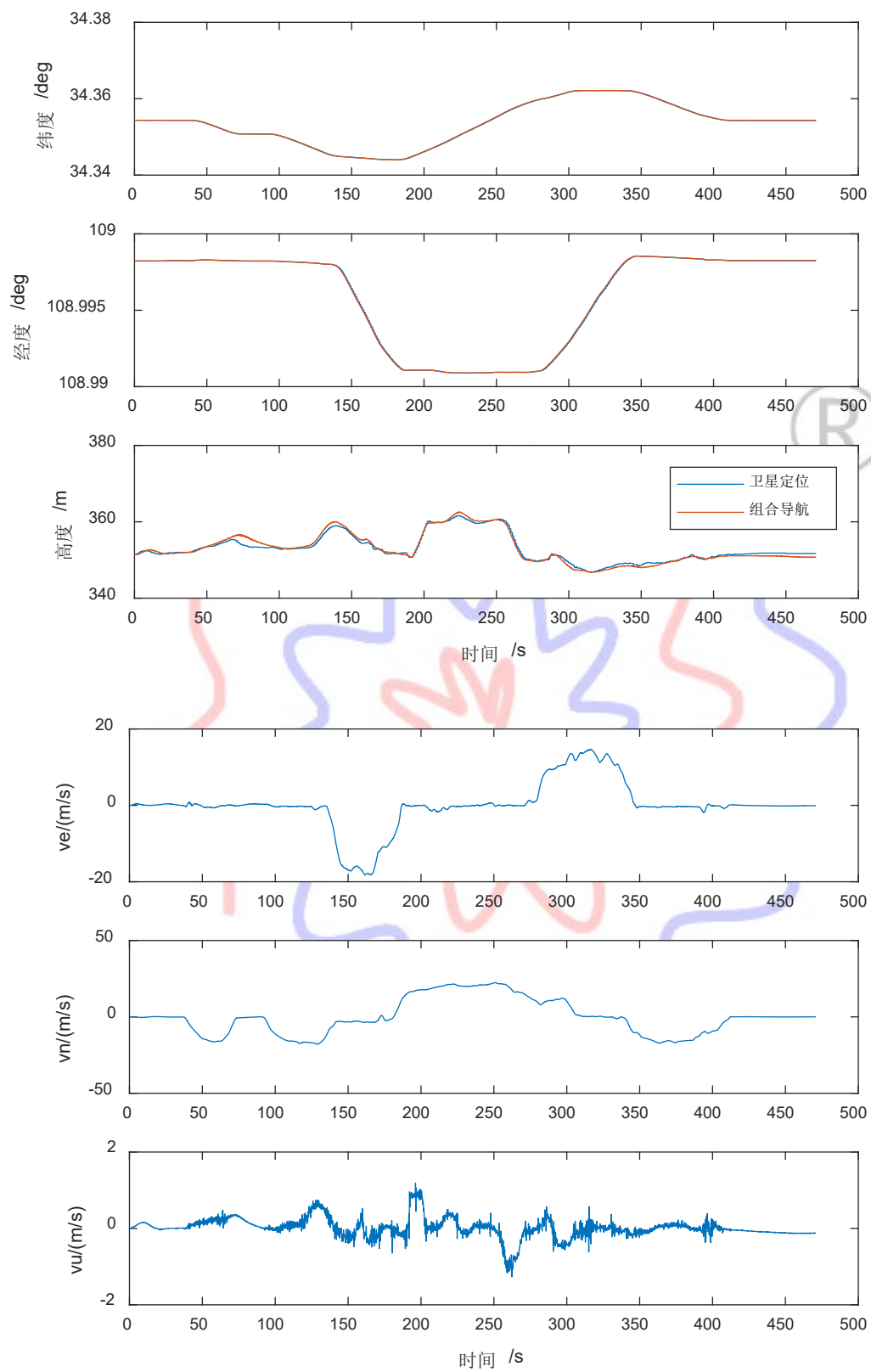
典型车载实验数据如下，实验路线包含城市道路、郊区道路、高架快速路、立交桥遮挡等实验条件。

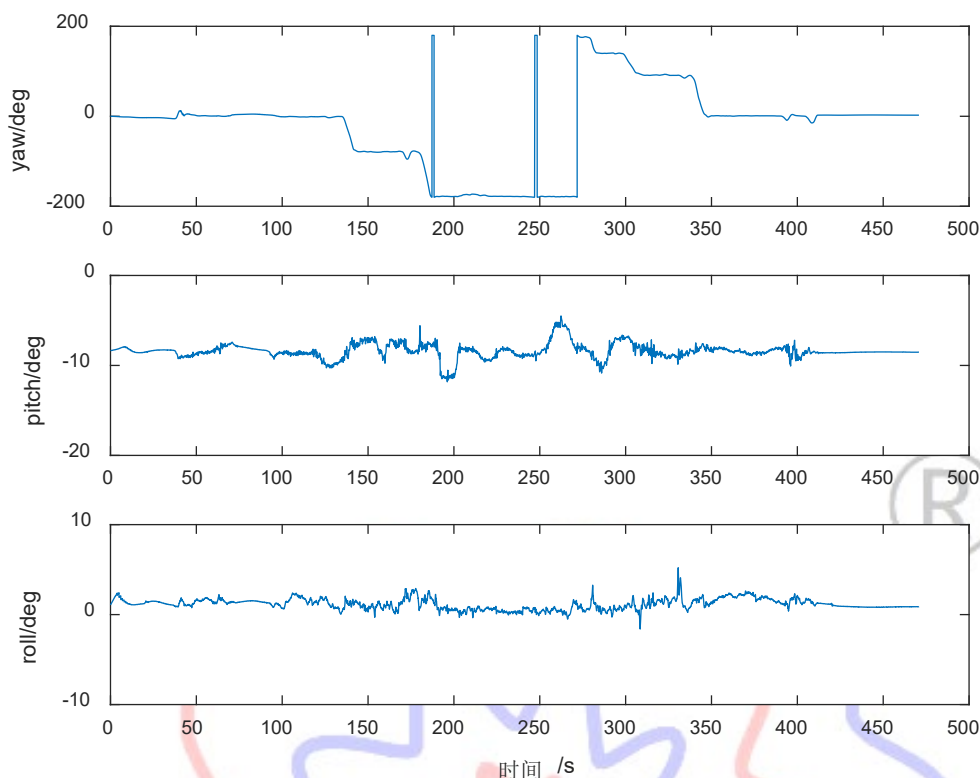


实验数据曲线如下若干图。





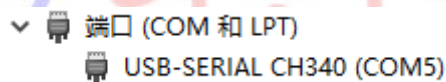




2. 操作方法

2.1. 准备

串口 USB 转换器直接连接计算机。注意，有些机型需要安装串口 USB 转换器的驱动。如果驱动正常，在硬件管理器中应显示如下标志：



配置串口助手。注意串口号应当与硬件管理器一致；默认波特率 230400；无校验位、8 数据位、1 停止位。

务必勾选“十六进制显示”。

实验前，建议勾选“接收转向文件”，以保存数据。

点击“打开”，即可接收串口数据。

2.2. 工作过程

卫星接收机天线要与开发板第二块电路板连接。在户外开阔地域静止。注意，天线与开发板本体距离要尽可能接近，否则会因为杆臂效应导致额外误差。

开发板没有自对准航向的功能，默认航向为 0。初始条件下，必须将 y 轴指向北方。在静态条件下通电，等待对准。对准期间要保持静止。

对准结束后，进入导航状态。此时第一块电路板亮绿灯。可以开始运动。运动方向不限，上下左右前后运动均可。注意天线与开发板本体应一同运动。

2.3. 数据处理

2.3.1. 数据解压

串口助手采集的文件，形式如下图，重命名为 d.txt。数据解压程序只识别这个文件名，务必一致。


```

49 00 00 6E E5 FF FF 94 E4 FF FF D7 E5 FE FF 1B 57 FE FF 7B D1 0E 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 5B 00 D9 FC CE 00 DA FF 2C FA FF FF 06 00 7B 14 78 DA F7 4
5 FE FF 1D 1C 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 5A 00 DB FC CC 00 DA FF 2C FA FF FF E4 FF 7A 14 77 DA F7 40 9C 0D 88 1D EB 90 3A 57 2E 00 00 F2 15 00 00
40 A4 0D 02 5A 00 DD FC CB 00 DA FF 2D FA FD FF C3 FF 7A 14 76 DA F7 40 9C 0D 34 21 EB 90 3F 0B 22 00 00 8B FE FF FC DE FF FF 66 C0 FF FF 10 F1 FD FF 5E B1 0F 00 00
1 FF 7A 14 75 DA F7 40 9C 0D 27 1F EB 90 44 94 28 00 00 F0 0C 00 00 DE DE FF FF C3 EA FF FF BE 34 FE FF 9D 3F 0E 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 58 00 E0 FC
00 00 AE 46 00 00 72 E5 FF FF 2B 00 FF FF 04 C7 FD FF 1A 31 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 57 00 E1 FC C8 00 DA FF 2D FA FF FF 79 FF 7A 14 74 DA F7 40
D FF 58 E1 0D 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 57 00 E2 FC C8 00 DA FF 2D FA FF FF 57 FF 7A 14 73 DA F7 40 9C 0D 4C 1E EB 90 4F 9A 1F 00 00 CE F5 FF FF 20 I
A4 0D 02 56 00 E2 FC C9 00 D9 FF 2D FA F8 FF 35 FF 7A 14 72 DA F7 40 9B 0D 3A 1D EB 90 54 76 F3 FF FF 1E 17 00 00 AE F3 FF FF 58 22 FF FF 46 22 FE FF 8E E7 0F 00 00 00 0
F 7A 14 71 DA F7 40 9B 0D 5F 1C EB 90 59 13 1C 00 00 BC E6 FF FF C3 EF FF FF CF 4C FF FF 3A 8A FE FF C2 07 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 55 00 E2 FC C
FF 11 EC FF FF 2D DE FF FF B1 D9 FF FF 83 08 FE FF 37 3C 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 55 00 E2 FC C7 00 D8 FF 2D FA F6 FF EC FE 7A 14 6F DA F7 40 9B
F 20 87 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 54 00 E1 FC C5 00 D8 FF 2D FA F6 FF CA FE 7A 14 6E DA F7 40 9B 0D 53 20 EB 90 64 8C 07 00 00 67 5F 00 00 BE E1 F
0D 02 53 00 E1 FC C4 00 D8 FF 2D FA F6 FF A8 FE 7A 14 6D DA F7 40 9B 0D AB 1D EB 90 69 D2 F2 FF FF C6 0F 00 00 73 D2 FF FF EE 20 FF FF 5A D0 FD FF EB 4E 0F 00 01 00 01
A 14 6C DA F7 40 9B 0D B9 1A EB 90 6E 3E 15 00 00 4C 3F 00 00 86 E1 FF FF D0 64 FF FF CC 6F FD FF 81 B1 10 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 52 00 E0 FC C3 0
00 E4 FF FF 7F E3 FF FF C9 6E FF FF 99 6E FD FF 52 DF 0E 00 00 00 01 F6 02 7B 14 A9 DA F7 40 A4 0D 02 51 00 E0 FC C3 00 D8 FF 2D FA F7 FF 5E FE 7A 14 6B DA F7 40 9B 0D

```

运行 trans27.exe，即可解压文件。注意，d.txt 和 fr.txt 必须与 trans27.exe 放在一个文件夹中。解压后的文件形式如下图。

37099	0	-0.13471395	0.47159940	-0.12371689	-0.22151507	-1.40763819	9.74296592	0	0	0
37099	1	-0.38429350	-0.12639910	-0.08314848	-0.21777116	-1.34072267	9.68397595	0	0	0
37099	2	-0.40615350	0.20746887	-0.08462369	-0.08710660	-1.52800232	9.80635174	0	0	0
37099	3	0.11003762	0.68926066	-0.16904622	-0.16785227	-1.50349922	9.55576077	0	0	0
37099	4	0.11989474	0.24387985	-0.11600554	-0.25970861	-1.35466456	9.78878699	0	0	0
37099	5	-0.15167892	0.16515702	-0.13625622	-0.13441779	-1.42926350	9.78931785	0	0	0
37099	6	-0.03728271	0.51645935	-0.17246604	-0.15997328	-1.48974359	9.66699794	0	0	0
37099	7	0.15905499	0.40541589	-0.15831739	-0.22325665	-1.40124001	9.73262824	0	0	0
37099	8	-0.03895909	0.25159121	-0.13283640	-0.19382685	-1.39753334	9.80388373	0	0	0
37099	9	-0.02990663	0.43042749	-0.07013977	-0.12498349	-1.48858875	9.71837901	0	0	0
37099	10	0.17675757	0.56172162	-0.07845461	-0.23944303	-1.43290497	9.67388973	0	0	0
37099	11	0.05813688	0.29229373	-0.13558567	-0.18238090	-1.38186850	9.80727375	0	0	0
37099	12	-0.05176663	0.43304265	-0.11855364	-0.17653219	-1.44810416	9.72340815	0	0	0
37099	13	0.03211945	0.39133429	-0.19218028	-0.18435530	-1.45773403	9.65734012	0	0	0
37099	14	0.08676946	0.33681840	-0.13712794	-0.21526590	-1.40854158	9.73578542	0	0	0
37099	15	-0.02125651	0.28149784	-0.10373443	-0.16669743	-1.41422264	9.78908502	0	0	0

用户也可以根据后文的通信协议自行制作数据解析软件。

2.3.2. 数据绘图

将 data.txt 与 viewdata.m 放在一个文件夹中。用 Matlab 运行 viewdata.m 即可得到数据曲线。

2.4. 烧写程序

电路板发货时已经烧写默认程序，可以不烧写程序、直接使用。

需要烧写程序时，要准备的软件开发环境为：Keil uVision5；推荐同时使用 STM32CubeMX。需要准备硬件的仿真器，如 ST-LINK V2（需要自行购买，本店发货时不包含硬件仿真器）。

仿真器连接到开发板的 P2 口，注意引线对应关系。

源代码最好解压到纯英文路径中，使用 Keil uVision5 打开\MDK-ARM\imu.uvprojx。然后点击 LOAD 即可烧写程序。



注意，烧写程序时，切勿选择擦除全部 FLASH；否则会导致配置失效、工作异常。

3. 性能

项目	数值	单位	备注
角速度量程	500	deg/s	
角速度零偏全温重复性	0.14	deg/s	X
	0.14		Y
	1.4		Z
角速度零偏稳定性	2.2	deg/h	X
	2.7		Y
	1.6		Z
加速度量程	78.4	m/s/s	
加速度零偏全温重复性	19.6×10^{-3}	m/s/s	
加速度零偏稳定性	26.5×10^{-6}	m/s/s	X Y
	43.1×10^{-6}		Z

卫星导航刷新率	1	Hz	最大 10
输出数据刷新率	200	Hz	可自行修改
位置精度	3	m	水平*
位置精度	6	m	天向*
速度精度	0.05	m/s	水平*
速度精度	0.1	m/s	天向*
姿态精度	0.05	deg	pitch roll *
姿态精度	0.1	deg	yaw*
纯惯性发散	3	m	纯惯性 20 秒, 典型值*
纯惯性发散	30	m	纯惯性 50 秒, 典型值*

* 不保证性能, 仅供参考; 对性能有严苛要求的用户应当自行验证。

本产品仅用于教学实验、演示工作原理, 不保证恶劣环境下的精度和商业应用的精度。
注意, 运动过程中不能出现超量程情况, 否则会导致精度下降。

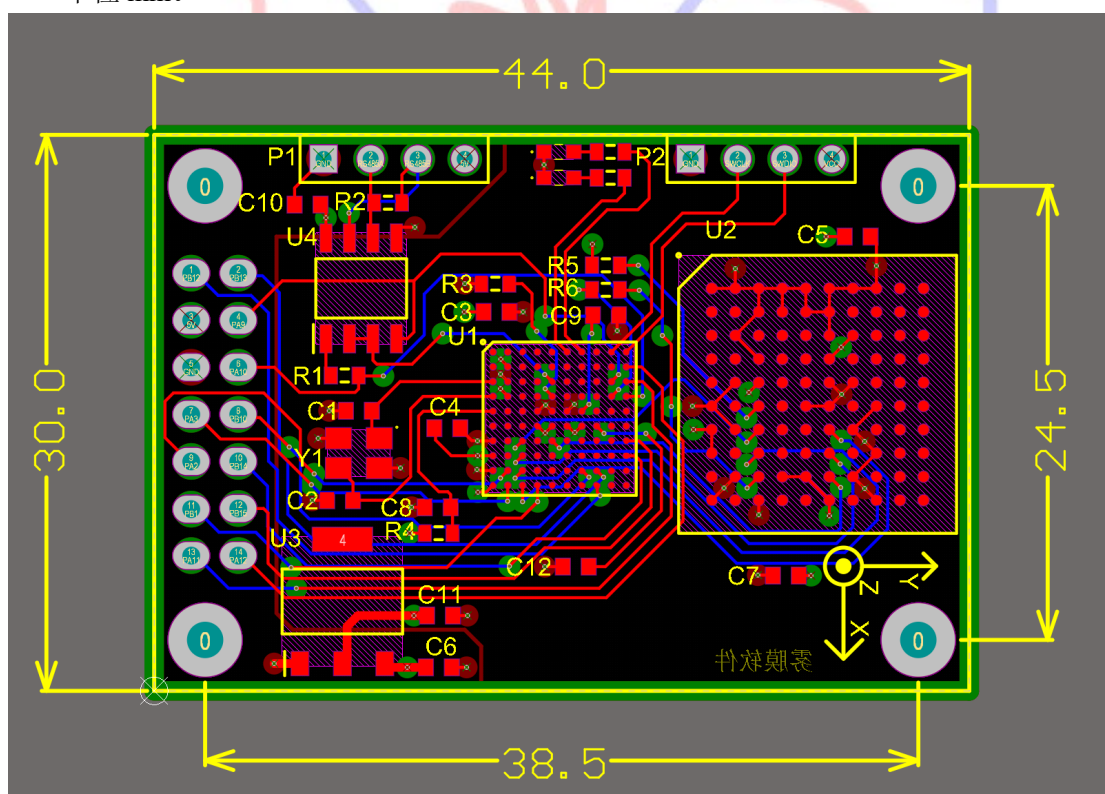
卫星信号正常情况下, 组合导航位置跟随卫星位置。在卫星信号丢失的条件下, 导航误差会在 20 秒后显著发散。实验时应当尽可能保证卫星导航可用。

本产品限中国境内使用。(含港澳台)

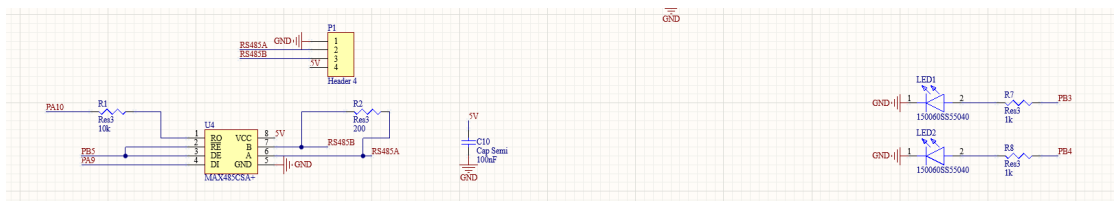
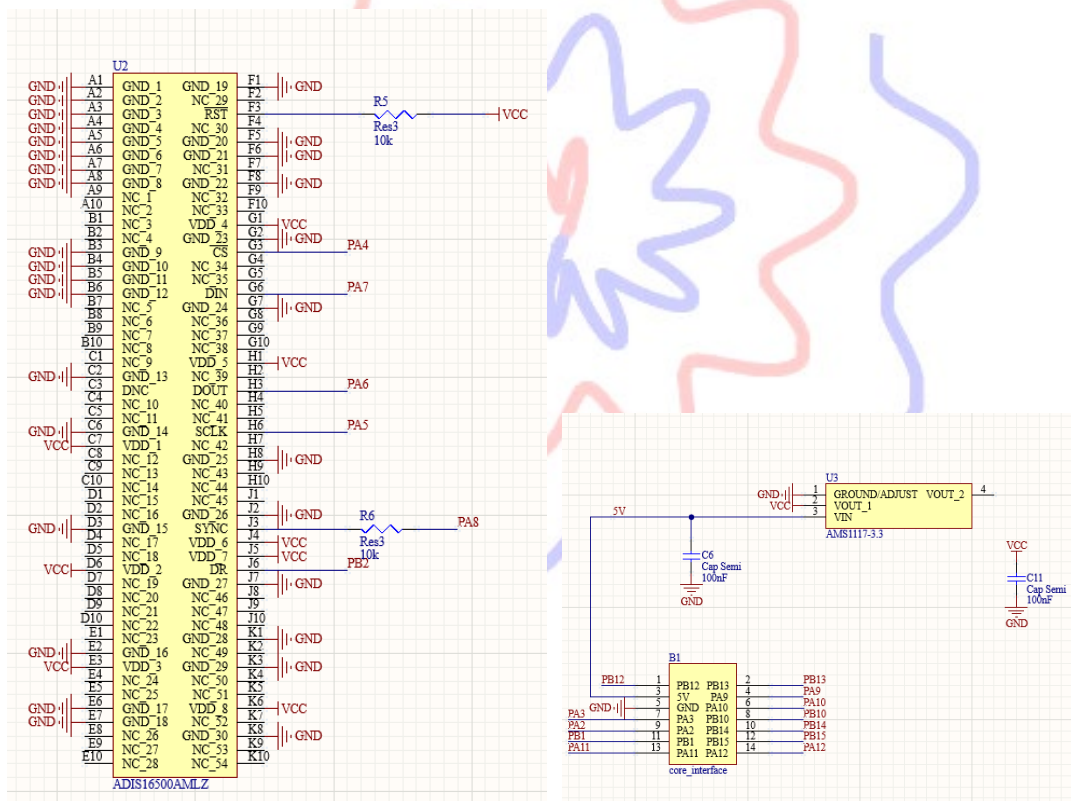
4. 外形和接口

4.1. 机械外形

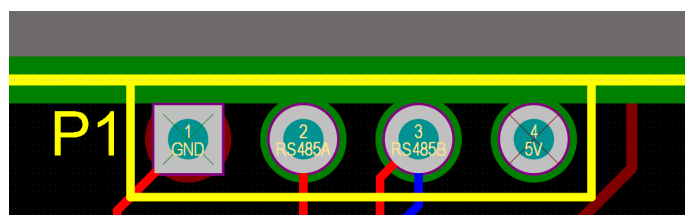
单位 mm。



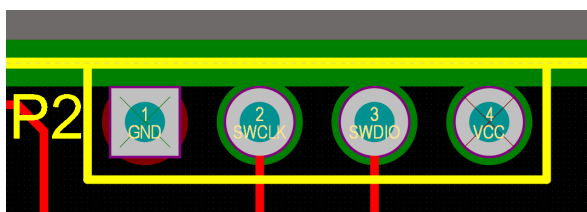
4.2. 原理图



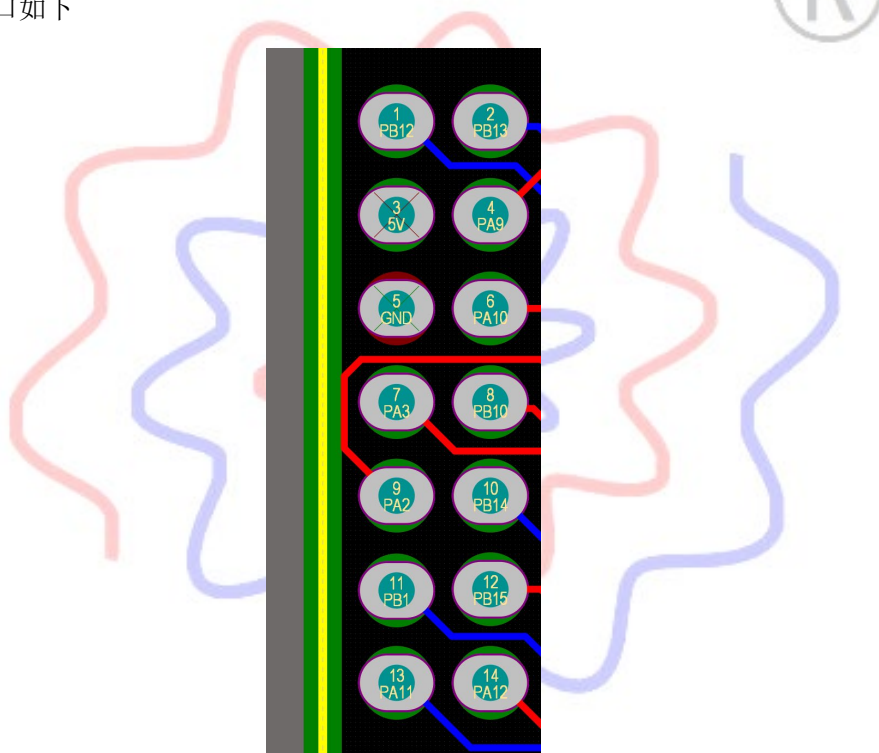
RS485 接口如下:



程序烧写口如下：



扩展接口如下



4.4. 通信协议

注意，使用字节流时，串口助手需要勾选“十六进制显示”。

基本规则：（1）长整数（比如 32 位整数或 16 位整数），拆分为字节，低字节在前、高字节在后；但是文本或帧头按照表中的顺序排列，不再调换顺序。（2）数据除以比例系数为物理量的值；物理量的值乘以比例系数为数据。（3）一般数据为有符号数；但是帧头、状态指示、标志、校验码等为无符号数。

字节号	内容	比例系数	单位/备注
0~1	0xeb 0x90		帧头
2	计数		0~255 循环
3~6	角增量 x	$2^{31}/720$ 或 $2^{31}/720/200$	deg 角增量 或 deg/s 角速度
7~10	角增量 y	同上	同上

11~14	角增量 z	同上	同上
15~18	速度增量 x	$2^{31}/100$ 或 $2^{31}/100/200$	m/s 速度增量 或 m/s/s 加速度
19~22	速度增量 y	同上	同上
23~26	速度增量 z	同上	同上
27	PPS 脉冲标志		
28	GGA 数据标志		
29	卫星导航状态		1 有效, 0 无效
30~33	纬度	10000000	deg
34~37	经度	10000000	deg
38~39	高度	10	m
40	导航状态标志		1 对准, 2 导航
41~42	yaw	100	deg
43~44	pitch	100	deg
45~46	roll	100	deg
47~48	速度东	100	m/s
49~50	速度北	100	m/s
51~52	速度天	100	m/s
53~56	纬度	10000000	deg
57~60	经度	10000000	deg
61~62	高度	10	m
63~64	校验码		2~62 字节求和

5. 算法原理

5.1. 概念定义

惯性测量单元为 3 轴陀螺仪和 3 轴加速度计。定义 x 向东、y 向北、z 向天为姿态 0 位置。旋转方向和角速度方向满足右手法则，即右手握住坐标轴，大拇指位于坐标轴正向，则其余四个手指指向旋转正向。姿态的欧拉角旋转顺序定义为依次绕 z、x、y 旋转。

组合导航中，扩展卡尔曼滤波的状态变量定义为 15 维度的误差量：经纬度高度位置，东北天速度，东北天姿态，三轴陀螺仪零偏，三轴加速度计零偏。

如果无特殊说明，一般采用国际单位制。角度单位为 rad，角速度单位为 rad/s，速度单位 m/s，加速度单位 m/s/s。

5.2. 主程序

程序主要工作流程：（仅供原理说明，不表示实际代码的结构）

```

设定初值
while(1)
{
    读取数据
    if(对准阶段)
    {
        对准
        if(卫星定位稳定)
        {
            转入导航阶段
        }
    }
}

```



```

    }
  }
  else 导航阶段
  {
    惯性导航计算
    更新状态方程
    if (收到卫星数据)
    {
      计算卡尔曼滤波
      补偿惯性导航的误差
    }
  }
  保存数据
}

```

5.3. 坐标系

载体系 b 定义为与载体固定连接的坐标系，不妨取 xyz 轴为右前上。

地理系 t 定义为与载体处地面重合的坐标系，不妨取 xyz 轴为东北天。

导航坐标系 n 是表示导航结果的坐标系。在航海、航空领域中，为了避免船只、飞机通过南北极附近时 n 系快速旋转导致导航结果异常， n 系会与 t 系有一定的夹角。在普通导航系统中，可以不虑载体通过南北极的情况，因此选取 n 系与 t 系重合以使导航算法简化。

平台坐标系 p ，是平台式导航系统中传感器的指向，或者是捷联式导航系统中数学换算后的传感器的指向。理想情况下 p 系与 n 系重合；但是由于陀螺仪误差等因素，真实的 p 系与 n 系有误差角。捷联式导航系统希望把加速度换算到 n 系中，但是实际上是换算到了 p 系中。在一般的导航计算中，不必刻意区分 p 系和 n 系，但是在分析误差时需要引入 p 系。

地球坐标系 e ，是和地球固连的坐标系，不妨规定 z 轴沿着南北极方向指向北， x 轴指向 0 经度方向。

惯性参考系 i 。惯性参考系主要用于描述概念。惯性导航中一般不需要真正地在惯性参考系中投影，所以不必在惯性参考系中规定坐标系。

完整地描述角速率、姿态、加速度、速度、位移等需要 3 个坐标系。坐标系 β 相对于坐标系 α 的变化量 x 在坐标系 γ 的投影表示为 $x_{\alpha\beta}^\gamma$ 。例如，地球自转在地理系的坐标为

$$\omega_{ie}^t = \begin{bmatrix} 0 \\ \omega_e \cos L \\ \omega_e \sin L \end{bmatrix} \quad (5-1)$$

其中 ω_e 是地球自转角速率， L 是纬度。这是地球系 e 相对于惯性系 i 的转动在地理系 t 的投影。在这种表示方法下，一些简单的计算规则如下：

同一个坐标系内表示的变量符合向量加法规则，即

$$x_{AB}^\gamma + x_{BC}^\gamma = x_{AC}^\gamma \quad (5-2)$$

同一个变量在不同坐标系的换算可以用矩阵表示。

$$x_{\alpha\beta}^\mu = C_\gamma^\mu x_{\alpha\beta}^\gamma \quad (5-3)$$

坐标变换矩阵表示旋转关系。例如二维的坐标变换矩阵为

$$C_n^b = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (5-4)$$

三维的坐标旋转有 3 个自由度，可以看作是类似形式矩阵相乘。

坐标变换矩阵是正交矩阵，逆矩阵是原矩阵的转置

$$\mathbf{C}_\mu^\gamma = (\mathbf{C}_\gamma^\mu)^{-1} = (\mathbf{C}_\gamma^\mu)^T \quad (5-5)$$

5.4. 惯性导航

5.4.1. 基本原理

惯性导航的基本原理是：陀螺仪测量角速度，角速度积分得到姿态。加速度计测量加速度，加速度积分得到速度，速度积分得到位置。

实际情况中有一些因素导致上述计算变得复杂。1.需要进行一些坐标系变换。2.需要考虑地球的自转、重力、以及球形形状。

5.4.2. 姿态更新

三维空间有 3 个旋转自由度。类似式(5-4)，依次绕三个坐标轴旋转，则坐标变换矩阵为

$$\mathbf{C}_n^b = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5-6)$$

把坐标变换矩阵表示为绕坐标轴分别旋转三次，三次旋转的角度即为欧拉角。旋转的顺序并不是唯一的，也可以定义旋转顺序不同的欧拉角。同一个坐标变换矩阵，在不同的旋转顺序定义下，有不同的欧拉角角度；同样的旋转角度，按照不同的坐标轴顺序旋转，会得到不同的坐标变换矩阵；这个性质称为姿态角的不可交换性。所以使用欧拉角描述姿态时必须规定清楚旋转顺序。本书中欧拉角定义为：初始状态右前上（xyz）三轴位于东北天方向，依次绕上轴旋转偏航角，绕右轴旋转俯仰角，绕前轴旋转横滚角。

如果每次旋转的角度很小，则坐标变换矩阵近似为

$$d\mathbf{C} = \begin{bmatrix} 1 & 0 & -d\theta_y \\ 0 & 1 & 0 \\ d\theta_y & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d\theta_x \\ 0 & -d\theta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & d\theta_z & 0 \\ -d\theta_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5-7)$$

略去二阶小量，则有

$$d\mathbf{C} = \begin{bmatrix} 1 & d\theta_z & -d\theta_y \\ -d\theta_z & 1 & d\theta_x \\ d\theta_y & -d\theta_x & 1 \end{bmatrix} \quad (5-8)$$

上式表示了坐标旋力矩阵与旋转角度的关系。如果旋转角度很小，则不必考虑旋转顺序。为了表示的方便，引入角增量反对称矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \quad (5-9)$$

那么姿态矩阵更新公式为

$$\mathbf{C}_b^i(t+T) = \mathbf{C}_b^i(t) \lim_{k \rightarrow +\infty} \left(\mathbf{I} + \frac{[\boldsymbol{\theta}_b^i]}{k} \right)^k = \mathbf{C}_b^i(t) \exp([\boldsymbol{\theta}_b^i]) \quad (5-10)$$

其中 \exp 表示自然常数 e 为底数的指数函数。 $\mathbf{C}_b^i(t)$ 是上一时刻的姿态矩阵， $\mathbf{C}_b^i(t+T)$ 是下一时刻的姿态矩阵。上式即姿态更新公式。

利用麦克劳林公式，能得到更便于计算的如下公式

$$\exp([\boldsymbol{\theta}]) = \mathbf{I} + \frac{\sin|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] + \frac{1 - \cos|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|^2} [\boldsymbol{\theta}]^2 \quad (5-11)$$

如果旋转角度较小，同时为了避免分母为 0，可以采用如下近似公式

$$\exp([\boldsymbol{\theta}]) \approx \mathbf{I} + [\boldsymbol{\theta}] \quad (5-12)$$

根据上述若干公式，使用陀螺仪数据计算得到姿态。

实际导航系统中，为了防止计算误差导致姿态矩阵失去正交性，也为了减少计算量，往往采用四元数代替姿态矩阵进行姿态更新。四元数定义为

$$\mathbf{q} = \begin{bmatrix} \cos \frac{\theta}{2} & u_x \sin \frac{\theta}{2} & u_y \sin \frac{\theta}{2} & u_z \sin \frac{\theta}{2} \end{bmatrix}^T \quad (5-13)$$

其中 θ 是旋转的角度， $[u_x \ u_y \ u_z]^T$ 是旋转轴的单位向量。

四元数也可以表示为

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{A} \sin \frac{\theta}{2} \quad (5-14)$$

其中 \mathbf{A} 是旋转轴的单位向量。

四元数姿态微分方程为

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \mathbf{q} \quad (5-15)$$

引入 4 维的角增量矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_x & -\theta_y & -\theta_z \\ \theta_x & 0 & \theta_z & -\theta_y \\ \theta_y & -\theta_z & 0 & \theta_x \\ \theta_z & \theta_y & -\theta_x & 0 \end{bmatrix} \quad (5-16)$$

四元数更新姿态的公式为

$$\mathbf{q}(t+T) = \left(\cos \frac{|\boldsymbol{\theta}|}{2} \mathbf{I} + \frac{\sin \frac{|\boldsymbol{\theta}|}{2}}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] \right) \mathbf{q}(t) \quad (5-17)$$

姿态可以用 3*3 矩阵或者 4*1 的四元数表示。本代码包采用四元数计算姿态，这是主流方法。但是矩阵对于坐标系变换的计算比较方便，所以坐标变换的地方使用了矩阵表示姿态。根据角度增量更新姿态四元数，为 `quatupdate` 函数。

实际导航中计算姿态时，要扣除地球自转的影响。此外，由于地球是球形的，位置变化时地面会向下弯曲，相对于地面的姿态就变了，所以速度会导致一个额外的角速度。

惯性导航需要依次计算姿态、速度、位置。陀螺仪直接测到 **b** 系相对于 **i** 系的转动，而导航解算中需要计算 **b** 系相对于 **n** 系的姿态。导航系 **n** 系相对于惯性系 **i** 系的相对转动包括两个部分：一是地球自转角速率；二是因为地球表面是曲面，载体位置变化会导致 **n** 系相对 **e** 系的姿态发生变化，角速率 $\boldsymbol{\omega}_{en}^n$ 与速度有关

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} -\frac{v_N}{R_m + H} \\ \frac{v_E}{R_p + H} \\ \frac{v_E \tan L}{R_p + H} \end{bmatrix} \quad (5-18)$$

其中 R_m 和 R_p 分别是子午圈和卯酉圈的半径， L 是纬度。

因为

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{ie}^b - \boldsymbol{\omega}_{en}^b \quad (5-19)$$

所以

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \quad (5-20)$$

根据上述角速率计算 \mathbf{C}_b^n ，即姿态。

5.4.3. 速度更新和位置更新

加速度要从传感器的坐标系换算到东北天。本来速度是加速度积分。但是由于地球自转，需要扣除离心力和科里奥利力。此外，加速度计不能区分重力和一般的加速度，所以还需要扣除重力。

加速度为

$$\dot{\mathbf{V}}_{\text{en}}^n = \mathbf{C}_b^n \mathbf{f}_b - 2\boldsymbol{\omega}_{\text{ie}}^n \times \mathbf{V}_{\text{en}}^n - \boldsymbol{\omega}_{\text{en}}^n \times \mathbf{V}_{\text{en}}^n + \mathbf{g} \quad (5-21)$$

其中 \mathbf{f}_b 是加速度计的数值。根据等效原理，加速度计不能把重力与真正的加速度相区分，所以要扣除地球重力 \mathbf{g} 的影响。因为地球是圆的，所以要补偿离心加速度项 $\boldsymbol{\omega}_{\text{en}}^n \times \mathbf{V}_{\text{en}}^n$ 。因为地球在自转，所以要补偿科氏加速度 $2\boldsymbol{\omega}_{\text{ie}}^n \times \mathbf{V}_{\text{en}}^n$ 。

加速度积分计算得到速度。速度积分计算得到位置。如果用经纬高表示位置，则有

$$\dot{L} = V_N / R_m \quad (5-22)$$

$$\dot{\lambda} = \frac{V_E}{R_p \cos L} \quad (5-23)$$

其中 L 和 λ 分别是纬度和经度， R_m 和 R_p 分别是当前位置的子午圈和卯酉圈半径。

地球是个椭球，函数 `earthmodelupdate` 计算两个方向的半径和重力。

5.5. 组合导航

5.5.1. 原理概述

连续计算惯性导航；当获取卫星数据时，采用扩展卡尔曼滤波修正导航误差。

卡尔曼滤波可以理解为：根据方差求权重，做加权平均。

原始的卡尔曼滤波适用于线性系统。因为导航系统不是线性的，所以采用扩展卡尔曼滤波。扩展卡尔曼滤波的主要方法是，选用误差量，利用一阶微分近似为线性系统。滤波得到误差量估计值后，立刻补偿误差。

有的文献把 EKF 算法进一步细化为 ESKF 算法，严格意义上本代码包的方法属于 ESKF 算法。但是大量的文献没有把 EKF 算法进行如此细致的划分，本代码包的算法完全可以说就是 EKF 算法。

5.5.2. 卡尔曼滤波

比较复杂的系统中，一方面系统具有多个自由度，另一方面被测量随着时间而变化。因此用状态空间方程的形式描述系统的关系，并把加权平均数计算方法用矩阵表示，则得到卡尔曼滤波。

系统表示为：

$$\mathbf{x}_k = \boldsymbol{\Phi} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (5-24)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (5-25)$$

其中是 \mathbf{x} 状态量，是希望获得而又难以准确测量的量。式(5-24)描述了被测量的变化关系，这里是离散形式。 \mathbf{z} 表示量测量，是能测量得到但是包含随机误差的量。式(5-25)描述了量测量与状态量的关系。 \mathbf{w} 和 \mathbf{v} 是随机噪声。有的系统中 \mathbf{w} 和 \mathbf{v} 会乘以系数矩阵，但是大多数惯性导航装置的三轴传感器精度大体相当，因此没必要引入标准卡尔曼滤波的 $\boldsymbol{\Gamma}$ 矩阵。

状态量的变化也可以描述为连续方程

$$\dot{\mathbf{x}}_k = \mathbf{F} \mathbf{x}_{k-1} \quad (5-26)$$

如果采样间隔足够小，离散方程与连续方程的关系为

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F}T \quad (5-27)$$

其中 T 为采样间隔， \mathbf{I} 为单位矩阵。

卡尔曼滤波的解算过程就是根据 \mathbf{z} 估计 \mathbf{x} ，具体方法如下：

如果不考虑误差，前后时刻的 \mathbf{x} 具有关系

$$\hat{\mathbf{x}}_{k|k-1} = \boldsymbol{\Phi} \hat{\mathbf{x}}_{k-1} \quad (5-28)$$

$\hat{\mathbf{x}}_{k-1}$ 是前一时刻 \mathbf{x} 的估计值, $\hat{\mathbf{x}}_{k|k-1}$ 是推算的后一时刻的 \mathbf{x} 。但是因为误差的存在, 这个推算并不准确, 需要根据 \mathbf{z} 修正, 因此取

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \quad (5-29)$$

其中 \mathbf{K}_k 是反映权重的滤波增益。这个增益由如下方法计算

$$\mathbf{P}_{k|k-1} = \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q} \quad (5-30)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R})^{-1} \quad (5-31)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (5-32)$$

其中 \mathbf{P} 、 \mathbf{Q} 、 \mathbf{R} 分别是 $\hat{\mathbf{x}}$ 、 \mathbf{w} 、 \mathbf{v} 的方差矩阵。

上述公式给出了线性系统的卡尔曼滤波方法。非线性系统可以局部微分而近似为线性系统, 采用扩展卡尔曼滤波方法解算。扩展卡尔曼滤波中的 \mathbf{x} 是误差量, 扩展卡尔曼滤波获得误差量后, 及时修正, 使得误差量总维持在较小范围内; 在误差量较小时, 局部微分得到的线性系统与原始的非线性系统基本一致, 卡尔曼滤波能取得较好效果。

代码包采用闭环反馈校正的方式, 滤波后修正惯导误差, 所以标准卡尔曼滤波中的 $\hat{\mathbf{x}}_{k-1}$ 取 0, 简化后的计算公式为

$$\hat{\mathbf{x}}_k = \mathbf{K}_k \mathbf{z}_k \quad (5-33)$$

导航系统是非线性系统。取扩展卡尔曼滤波的状态量 \mathbf{x} 为 15 维向量, 包含位置误差、速度误差、姿态误差、陀螺仪零偏、加速度计零偏各 3 各自由度, 即

$$\mathbf{x} = [\delta L \quad \delta \lambda \quad \delta h \quad \delta v_E \quad \delta v_N \quad \delta v_U \quad \delta \phi_E \quad \delta \phi_N \quad \delta \phi_U \quad \varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \nabla_x \quad \nabla_y \quad \nabla_z]^T \quad (5-34)$$

用扩展卡尔曼滤波进行组合导航的步骤是: 1. 进行惯性导航解算。2. 卫星修正时, 比较惯性导航与卫星导航的结果偏差, 即 \mathbf{z} 。3. 用卡尔曼滤波计算 \mathbf{x} 。4. 根据 \mathbf{x} 修正惯性导航的结果, 并返回步骤 1。

5.5.3. 组合导航的状态矩阵

惯性和卫星组合导航系统关键在于具体列出状态矩阵 Φ , 即可实现组合导航的计算。扩展卡尔曼滤波的矩阵 \mathbf{F} 是雅可比矩阵, 即偏微分矩阵。根据惯性导航的计算公式, 可以得到 \mathbf{F} 如下。

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{pp} & \mathbf{F}_{vp} & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \\ \mathbf{F}_{pv} & \mathbf{F}_{vv} & \mathbf{F}_{av} & \mathbf{O}_3 & \mathbf{C}_b^n \\ \mathbf{F}_{pa} & \mathbf{F}_{va} & \mathbf{F}_{aa} & -\mathbf{C}_b^n & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{O}_3 \end{bmatrix} \quad (5-35)$$

其中每个子矩阵都是 3 阶方阵, \mathbf{O}_3 表示 0 矩阵。

反映位置误差对位置误差影响的子矩阵为

$$\mathbf{F}_{pp} = \begin{bmatrix} 0 & 0 & -\frac{v_N}{(R_m + h)^2} \\ \frac{v_E \sec L \tan L}{R_p + h} & 0 & -\frac{v_E \sec L}{(R_p + h)^2} \\ 0 & 0 & 0 \end{bmatrix} \quad (5-36)$$

反映速度误差对位置误差影响的子矩阵为

$$\mathbf{F}_{vp} = \begin{bmatrix} 0 & \frac{1}{R_m + h} & 0 \\ \frac{\sec L}{R_p + h} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5-37)$$

反映位置误差对速度误差影响的子矩阵为

$$\mathbf{F}_{pv} = \begin{bmatrix} 2\omega_e v_N \cos L + 2\omega_e v_U \sin L + \frac{v_N v_E \sec^2 L}{R_p + h} & 0 & \frac{v_U v_E - v_N v_E \tan L}{(R_p + h)^2} \\ -\left(2\omega_e v_E \cos L + \frac{v_E^2 \sec^2 L}{R_p + h}\right) & 0 & \frac{v_N v_U}{(R_m + h)^2} + \frac{v_E^2 \tan L}{(R_p + h)^2} \\ -2v_E \omega_e \sin L & 0 & -\frac{v_N^2}{(R_m + h)^2} - \frac{v_E^2}{(R_p + h)^2} \end{bmatrix} \quad (5-38)$$

反映速度误差对速度误差影响的子矩阵为

$$\mathbf{F}_{vv} = \begin{bmatrix} \frac{v_N \tan L - v_U}{R_p + h} & 2\omega_e \sin L + \frac{v_E \tan L}{R_p + h} & -2\omega_e \cos L - \frac{v_E}{R_p + h} \\ -2\omega_e \sin L - \frac{2v_E \tan L}{R_p + h} & \frac{-v_U}{R_m + h} & \frac{-v_N}{R_m + h} \\ 2\left(\omega_e \cos L + \frac{v_E}{R_p + h}\right) & \frac{2v_N}{R_m + h} & 0 \end{bmatrix} \quad (5-39)$$

反映姿态误差对速度误差影响的子矩阵为

$$\mathbf{F}_{av} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix} \quad (5-40)$$

其中 f_E 、 f_N 、 f_U 是换算到n系的加速度计数值，即不扣除重力的比力信息。

$$\mathbf{f}_n = \begin{bmatrix} f_E \\ f_N \\ f_U \end{bmatrix} = \mathbf{C}_b^n \mathbf{f}_b \quad (5-41)$$

反映位置误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{pa} = \begin{bmatrix} 0 & 0 & \frac{v_N}{(R_m + h)^2} \\ -\omega_e \sin L & 0 & \frac{-v_E}{(R_p + h)^2} \\ \omega_e \cos L + \frac{v_E \sec^2 L}{R_p + h} & 0 & \frac{-v_E \tan L}{(R_p + h)^2} \end{bmatrix} \quad (5-42)$$

反映速度误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{va} = \begin{bmatrix} 0 & -\frac{1}{R_m + h} & 0 \\ \frac{1}{R_p + h} & 0 & 0 \\ \frac{\tan L}{R_p + h} & 0 & 0 \end{bmatrix} \quad (5-43)$$

反映姿态误差对姿态误差影响的子矩阵为

$$\mathbf{F}_{aa} = \begin{bmatrix} 0 & \omega_e \sin L + \frac{v_E \tan L}{R_p + h} & -\omega_e \cos L - \frac{v_E}{R_p + h} \\ -\omega_e \sin L - \frac{v_E \tan L}{R_p + h} & 0 & -\frac{v_N}{R_m + h} \\ \omega_e \cos L + \frac{v_E}{R_p + h} & \frac{v_N}{R_m + h} & 0 \end{bmatrix} \quad (5-44)$$

导航计算机每次收到惯性数据时，要计算 \mathbf{F} 矩阵，并更新 Φ 矩阵。导航计算机收到卫星数据时再进行卡尔曼滤波解算，并根据滤波计算得到的误差量修正导航结果。

5.5.4. 卫星导航数据的延迟补偿

因为数据处理传输过程的耗时，卫星导航数据比惯性导航数据略微滞后。为此，程序中缓存了惯性导航的结果。采用较旧的惯性导航数据与卫星导航结果比对、计算扩展卡尔曼滤波。然后在最新的惯性导航结果上修正误差。

6. 著作权和服务

6.1. 工作原理参考什么资料

为防止非法转卖，代码本身注释较少。但是本店提供人工答疑、纸质资料、讲解视频等，有效帮助用户理解代码。

纸质资料《组合导航应用笔记》，联系本店微信索要。

讲解视频，B 站（哔哩哔哩）搜索 WMSOFT。

6.2. 著作权声明

本店保留著作权。

电路、说明书、全部附属代码（以下简称本代码包）仅限于学习和研究用途的少量使用；包含改编文件、写入嵌入式系统的编译后程序，所有副本总计不得超过 5 份。

本代码包有偿使用。

严禁转卖或公开发布本代码包的全部或一部分。

大规模应用本代码包需要额外取得本店的授权。

对于违反上述要求的用户，本店有权要求停止销售、撤稿、赔偿损失等。

6.3. 服务内容

赠送 30 分钟语音答疑服务，用于解决较为复杂的疑问。赠送长期文字答疑，用于解决简单的、零散的疑问。加微信答疑。

答疑服务仅限直接购买人本人使用。答疑服务不能转让、不能共享。用户需要保留购买凭证截图。

疑问较多的用户，可以付费购买额外的语音答疑服务。

本店可提供数据判读、数据处理、定制化修改代码等服务，但是需要额外收费。

6.4. 联系方式

WMSOFT

组合导航二次开发生态系统
微信扫描二维码添加好友



电子邮箱: braun@wmsoft.wang

网站: <http://wmsoft.xyz>

哔哩哔哩、知乎、闲鱼账号均为 WMSOFT。

