НИУ «МЭИ»
Институт Радиотехники и электроники
Им. В.А. Котельникова

Аппаратура потребителей спутниковых радионавигационных систем

Курсовой проект

«Расчет траектории движения спутника GPS по данным с демодулятора его сигнала»

Студент: Коробков А.Ю.
Группа: ЭР-15-17
Преподаватель: Корогодин И.В.

Москва
2022

Оглавление

Цель работы 3
Исходные данные 3
Решение 3
Пункт 1 3
Пункт 2 3
Приложение 3

Цель работы

Изучение особенностей сигналов спутников GPS для определения положения спутника по данным с демодулятора его сигнала L1 C/A. На первом этапе происходит моделирование модуля разбора навигационного сообщения до структуры эфемерид.

Исходные данные

Файл “in.txt” с записанными в нем данными, зафиксированными навигационным приемником по сигналу GPS L1C/A. Файл содержит наблюдения псевдодальностей и прочих радионавигационных параметров, демодулированные и разобранные данные навигационного сообщения.

Решение

Пункт 1

На первом этапе необходимо создать программу в среде C++, выполняющую функции аналогичные модулю разбора навигационного сообщения. Листинг созданной части программы приведен в приложении.

Пункт 2

Приложение

```
#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <iostream>

struct Ephemeris {
	double Crs;
	double Dn;
```

```
        double M0;
        double Cuc;
        double e;
        double Cus;
        double sqrtA;
        uint32_t toe;
        double Cic;
        double Omega0;
        double Cis;
        double i0;
        double Crc;
        double omega;
        double OmegaDot;
        double iDot;
        double Tgd;
        uint32_t toc;
        double af2;
        double af1;
        double af0;
        uint32_t Wn;
        uint16_t IODC;
        uint8_t URA;
        uint8_t Health;
        uint16_t IODE2;
        uint16_t IODE3;
        bool codeL2;
        bool L2P;
        uint32_t slot;
};
```

```
using namespace std;
setlocale(0, "");

const int32_t subFrameLength = 300;
struct SF1_3 {
	char sf1[subFrameLength+1];
	char sf2[subFrameLength+1];
	char sf3[subFrameLength+1];
};

void printEph(Ephemeris* ep);
int32_t file2subFrames(SF1_3* sf, FILE* fid, uint8_t svNum);


 int main(void) {

	uint8_t svNum = 12;
	FILE* fid = fopen("in.txt", "r");

	if (fid != nullptr) {
		SF1_3 subFrames;
		if (!file2subFrames(&subFrames, fid, svNum)) {
			printf("SF1: %s\n", subFrames.sf1);
			printf("SF2: %s\n", subFrames.sf2);
			printf("SF3: %s\n", subFrames.sf3);

			// Ephemeris ep;
			Ephemeris *ep = (Ephemeris*) calloc(1, sizeof(Ephemeris));

			if (!subFrames2Eph(ep, &subFrames)) {
```

```
                printEph(ep);
            } else {
                printf("Cannot decode subframes");
            }

            free(ep);
            fclose(fid);
        } else {
            printf("Subframes not found\n");
        }
    } else {
        printf("Cannot open in.txt\n");
    }

    return 0;
}

uint32_t str2uint(char *sf, int32_t start, int32_t stop) {
    uint32_t ans = 0;
    for (int i = start; i < stop; i++) {
        bool bit = (sf[i-1] == '1');
        ans = ans | (bit << (stop - i - 1));
    }
    return ans;
}

int32_t subFrames2Eph(Ephemeris* ep, SF1_3* subFrames) {
    ep->slot = subFrames->slot;
    ep->Wn = str2uint(subFrames->sf1, 61, 71);
    return 0;
```

```
}

int32_t file2subFrames(SF1_3* sf, FILE* fid, uint8_t svNum) {

	int32_t sth1, sth2, sth3, sth4, sth5;
	char str_0R[8];
	char str_GpsL1CA[12];
	char str_reh[8];
	char str[1000];
	uint32_t svStr;
	uint32_t slot;
	uint32_t  subFrameNum;

	uint32_t slot_SF1 = 0;
	uint32_t slot_SF2 = 0;
	uint32_t slot_SF3 = 0;

	uint32_t readres = 0;
	while (readres != EOF) {

		svStr = 0;
		readres = fscanf(fid, "%d %d %d %s %s %s %u\t %u %d %d %d %s", &sth1, &sth2, &sth3, str_0R, str_GpsL1CA, str_reh, &svStr, &slot, &sth4, &sth5, &subFrameNum, str);
		if ((svStr == svNum) && (slot >= (604800/6))){
			if(subFrameNum == 1) {
				slot_SF1 = slot;
				strncpy(sf->sf1, str, sizeof(sf->sf1));
			} else if(subFrameNum == 2) {
				slot_SF2 = slot;
				strncpy(sf->sf2, str, sizeof(sf->sf2));
```

```
                } else if(subFrameNum == 3) {
                        slot_SF1 = slot;
                        strncpy(sf->sf3, str, sizeof(sf->sf3));
                }
                if ((slot_SF1 + 1 == slot_SF2) && (slot_SF2 + 1 == slot_SF3)) {
                        sf->slot = slot_SF1;
                        return 0;
                }
        }
    }

    return 1;
}

void printEph(Ephemeris* ep) {
    printf ("LNAV Ephemeris (slot = %u) = \n", ep->slot    );
    printf ("Crs    = %f              \n",   ep->Crs   );
    printf ("Dn     = %f  \t[deg/s]      \n",   ep->Dn          );
    printf ("M0     = %f  \t[deg]        \n",   ep->M0          );
    printf ("Cuc    = %f              \n",   ep->Cuc   );
    printf ("e      = %f                  \n",   ep->e    );
    printf ("Cus    = %f              \n",   ep->Cus   );
    printf ("sqrtA          = %f              \n",   ep->sqrtA );
    printf ("toe    = %u              \n",   ep->toe   );
    printf ("Cic    = %f              \n",   ep->Cic   );
    printf ("Omega0     = %f  \t[deg]      \n",   ep->Omega0);
    printf ("Cis    = %f              \n",   ep->Cis   );
    printf ("i0     = %f  \t[deg]        \n",   ep->i0          );
    printf ("Crc    = %f              \n",   ep->Crc   );
    printf ("omega          = %f  \t[deg]      \n",   ep->omega );
```

```
        printf ("OmegaDot  = %f  \t[deg]     \n",   ep->OmegaDot);
        printf ("iDot  = %f             \n",   ep->iDot  );
        printf ("Tgd   = %f             \n",   ep->Tgd   );
        printf ("toc    = %u            \n",   ep->toc   );
        printf ("af2    = %f            \n",   ep->af2   );
        printf ("af1    = %f            \n",   ep->af1   );
        printf ("af0    = %f            \n",   ep->af0   );
        printf ("Wn    = %f                \n",   ep->Wn        );
        printf ("IODC         = %u              \n",   ep->IODC  );
        printf ("URA          = %u              \n",   ep->URA   );
        printf ("Health       = %u              \n",   ep->Health);
        printf ("IODE2        = %u              \n",   ep->IODE2 );
        printf ("IODE3        = %u              \n",   ep->IODE3 );
        printf ("codeL2    = %u             \n",   ep->codeL2);
        printf ("L2P   = %u             \n",   ep->L2P   );
}
```