

ソフトウェア無線機を利用した GNSS信号シミュレータの実装

中部大学 海老沼拓史

1

GNSS信号シミュレータとは？

- ▶ 衛星から送信されるGNSS信号を模擬するものではない。
 - ▶ このような信号発生器はPseudolite(擬似衛星)と呼ばれる。
- ▶ ある特定のユーザの受信機アンテナで受信されるGNSS信号を模擬する信号発生器である。
 - ▶ 符号分割多重化(CDMA)がされた複数衛星からのGNSS信号を同時に生成する。

GNSS信号シミュレータの用途

- ▶ 電波の届かない屋内にしながら、GNSS受信機の測位機能を確認できる。
- ▶ 衛星配置や信号強度など、同じ条件で均質な試験を繰り返し実施できる。
- ▶ ロケットや人工衛星など、特殊な移動体での測位機能を確認できる。
- ▶ ただし、マルチパスのような複雑な受信環境を模擬するのは難しい。
- ▶ 本来は、工場などでGNSS受信機製品の動作をチェックするための検査治具として用いられる。
 - ▶ 研究用途としては、コストパフォーマンスが悪い。

▶ 3

ソフトウェア無線とは？

- ▶ 従来、電気電子回路のハードウェアで実現していた変復調などの信号処理を、ソフトウェアによるデジタル信号処理で行う。
- ▶ 数式で表される変復調の処理をそのままプログラムとして記述できるために、直感的な実装が可能となる。
- ▶ アイディアとしては古くからあったが、近年のプロセッサの性能向上により、現実的となった。
- ▶ ソフトウェア無線のための無線ICも数多く市販され、それらを搭載したデバイスも安価に入手できる。
 - ▶ GNSS信号を生成するソフトウェアさえあれば、GNSS信号シミュレータが実現できる。

▶ 4

代表的なソフトウェア無線機

	LimeSDR Mini 2.0	HackRF One	USRP B210	BladeRF x40	ADALM-PLUTO
RF Transceiver	LMS7002M	MAX5864	AD9361	LMS6002M	AD9364
Frequency Range	10 MHz – 3.5 GHz	1 MHz – 6 GHz	70 MHz – 6 GHz	300 MHz – 3.8 GHz	325 MHz – 3.8 GHz
RF Bandwidth	40 MHz	20 MHz	61.44 MHz	40 MHz	20 MHz
Sample Depth	12 bits	8 bits	12 bits	12 bits	12 bits
Sample Rate	30.72 MSPS	20 MSPS	61.44 MSPS	40 MSPS	61.44 MSPS
RF Channels	1	1	2	1	1
Interface	USB 3.0	USB 2.0	USB 3.0	USB 3.0	USB 2.0
Oscillator Precision	1 ppm	20 ppm	2 ppm	1 ppm	25 ppm
Price	\$399	\$350	\$2,100	\$520	\$230

▶ 5

ソフトウェア無線機の購入

- ▶ LimeSDR Mini 2.0
 - ▶ <https://www.crowdsupply.com/lime-micro/limesdr-mini-2>
- ▶ HackRF One
 - ▶ <https://www.nooelec.com/store/hackrf-one.html>
 - ▶ <https://www.nooelec.com/store/tiny-tcxo.html>
- ▶ USRP B210
 - ▶ <https://www.ettus.com/all-products/ub210-kit/>
- ▶ BladeRF x40
 - ▶ <https://www.nuand.com/product/bladerf-x40/>
- ▶ ADALM-PLUTO
 - ▶ Digi-Key / Mouser / chip1stop

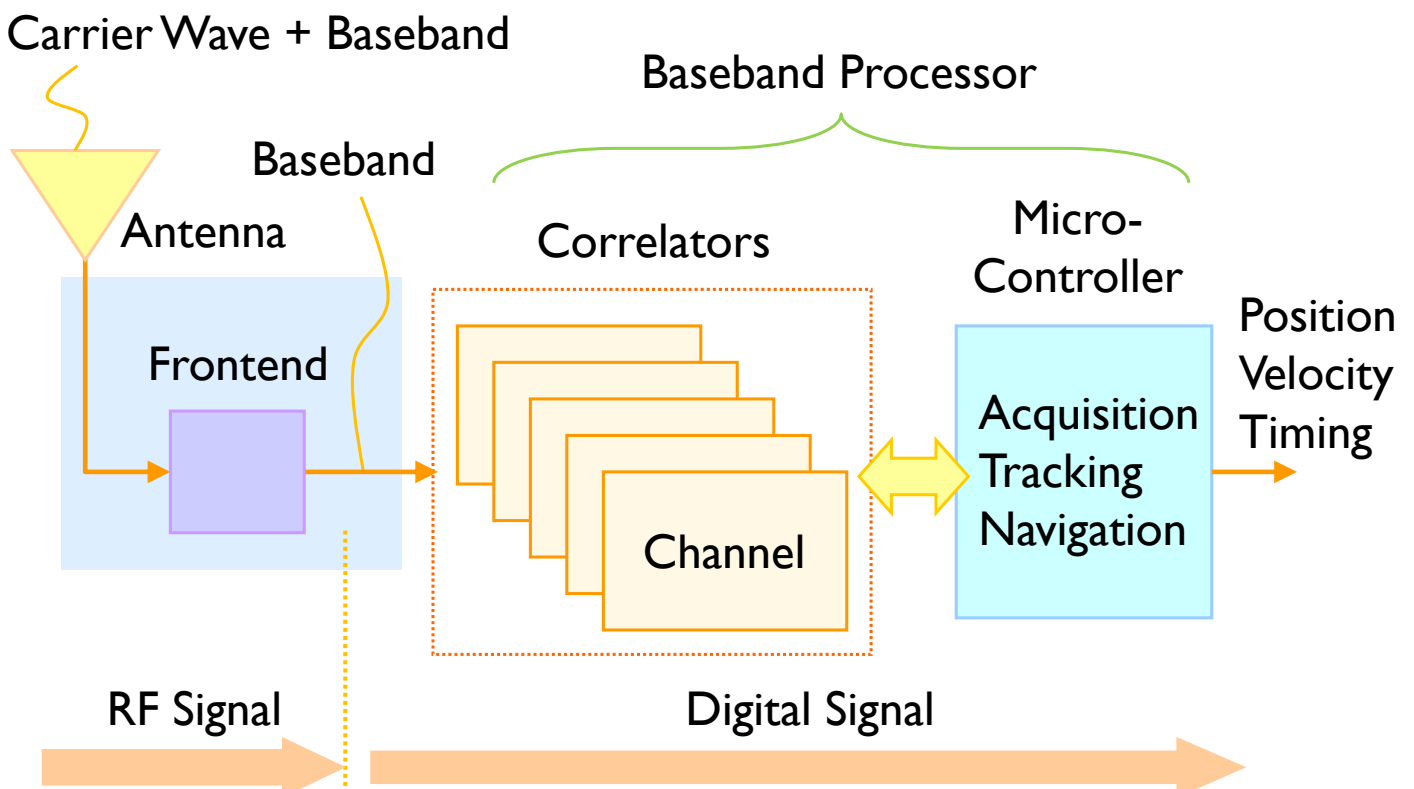
▶ 6

無線信号とベースバンド

- ▶ 無線通信で送りたい情報信号をベースバンド (baseband) と呼ぶ。
 - ▶ 一般的に、情報信号は低周波数の信号である。
- ▶ ベースバンドを電波で効率的に送信するために、高周波数の無線信号である搬送波 (carrier wave) に情報信号を埋め込む。
 - ▶ ベースバンドを無線信号に埋め込む処理を変調 (modulation) と呼ぶ。
- ▶ 受信機では、搬送波に埋め込まれたベースバンドを取り出し、さらにそこから情報を読み出す。
 - ▶ 無線信号からベースバンドを取り出す処理を復調 (demodulation) と呼ぶ。

▶ 7

GNSS受信機の構成



▶ 8

搬送波の変調

- ▶ 三角関数で記述できる搬送波に情報を埋め込もうとすると、振幅(amplitude)および位相(phase)を情報信号に応じて変化させることになる。

$$\begin{aligned} \text{Carrier Wave} \\ s(t) &= A \cos(\omega t - \phi) \\ &\quad \text{Amplitude} \qquad \text{Phase} \\ &= A \cos(\phi) \cos(\omega t) + A \sin(\phi) \sin(\omega t) \\ &= I(t) \cos(\omega t) + Q(t) \sin(\omega t) \\ &\quad \text{In-phase} \qquad \text{Quadrature} \end{aligned}$$

▶ 9

周波数変調と位相変調

- ▶ 搬送波の周波数を変化させることで情報を埋め込むこともできるが、ここでは位相変調と等価として扱う。

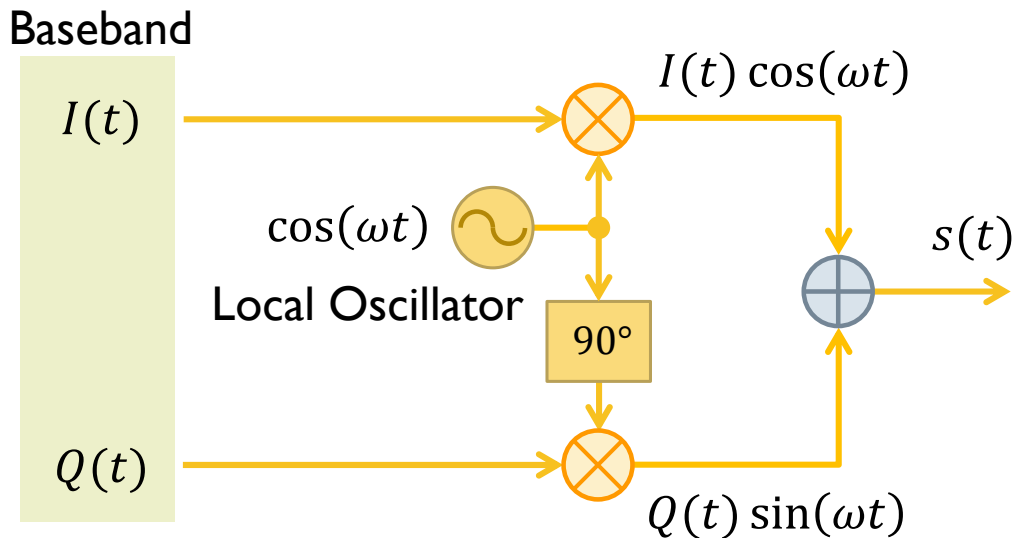
$$\begin{aligned} s(t) &= A \cos((\omega - \Delta\omega)t - \phi) \\ &= A \cos(\omega t - \Delta\omega t - \phi) \\ &= A \cos(\omega t - \underbrace{(\Delta\omega t + \phi)}_{\varphi(t)}) \end{aligned}$$

- ▶ ベースバンドに含まれるドップラーなども $\Delta\omega$ として扱う。

▶ 10

Quadrature Amplitude Modulation

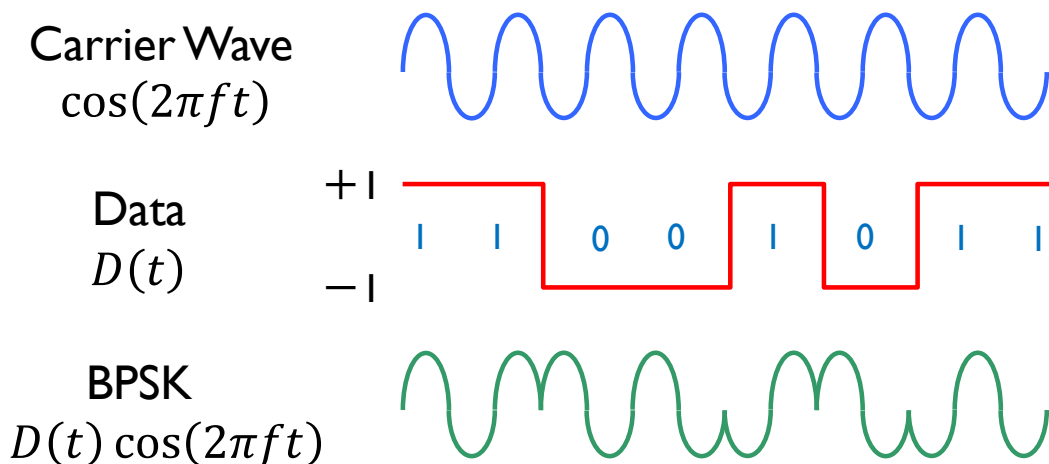
- ▶ 互いに直交した2つの搬送波の振幅をベースバンドに応じて変更することで情報を伝送する変調方式を直交位相振幅変調 (quadrature amplitude modulation) と呼ぶ。



▶ 11

Binary Phase Shift Keying (BPSK)

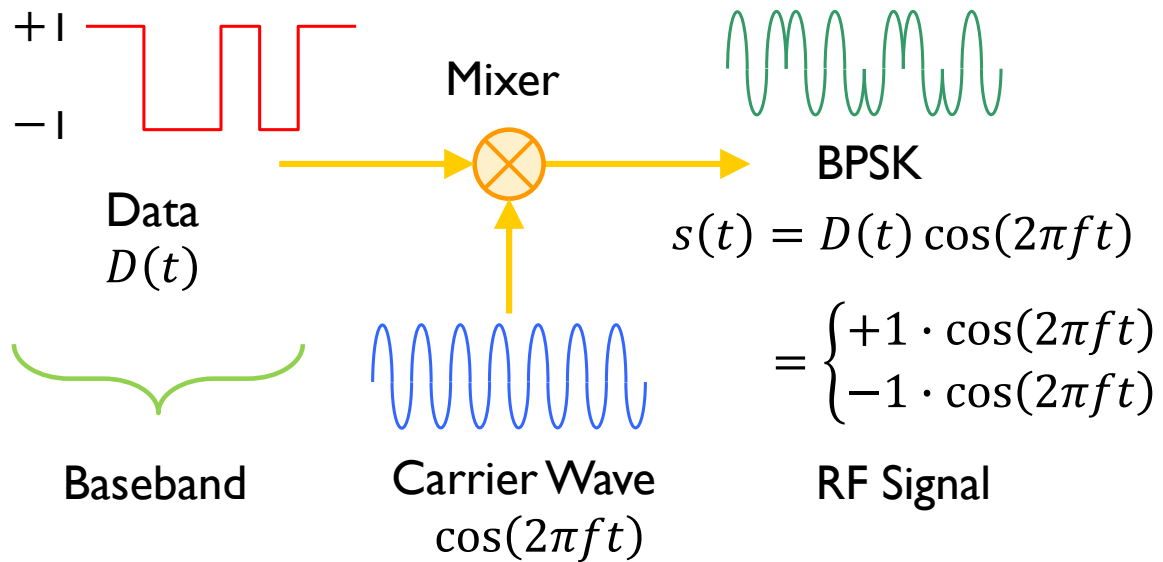
- ▶ GNSS信号の変調方式として、代表的なGPS信号のBPSKを取り上げる。
- ▶ BPSKはデジタル変調方式のひとつであり、データ信号のビットの変化に応じて、搬送波の位相を 180° 変化させる。



▶ 12

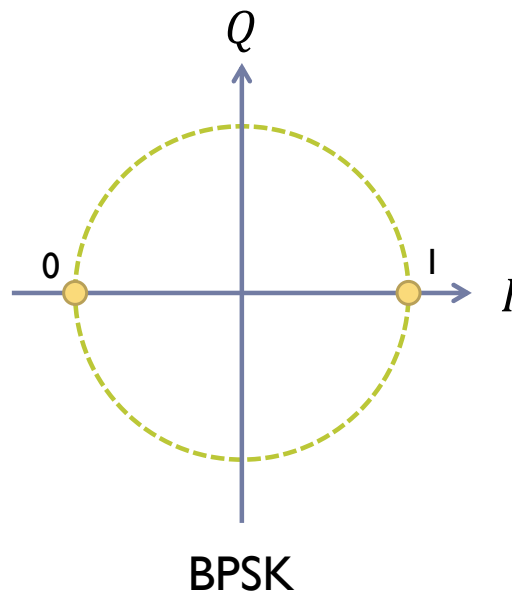
BPSK信号の生成

- ▶ BPSK変調は、+1と-1の2値で表される情報信号と、三角関数で記述される搬送波の掛け算で生成できる。
 - ▶ 直交位相振幅変調のひとつとして実装できる。



Constellation Diagram

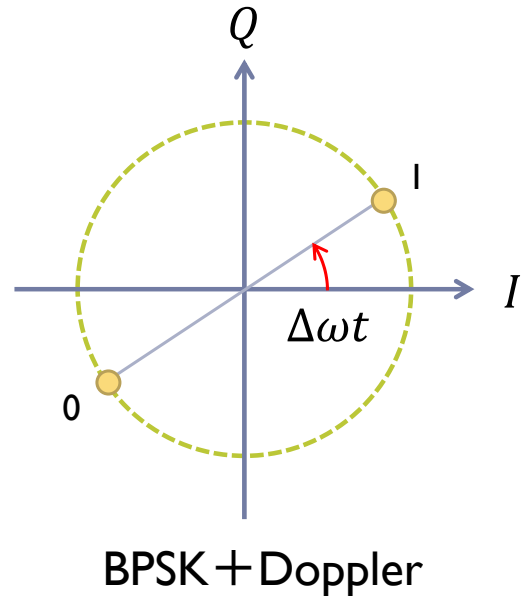
- ▶ 直交位相振幅変調における直交した2つの搬送波の振幅を直交座標で描いたものをconstellation diagramと呼ぶ。



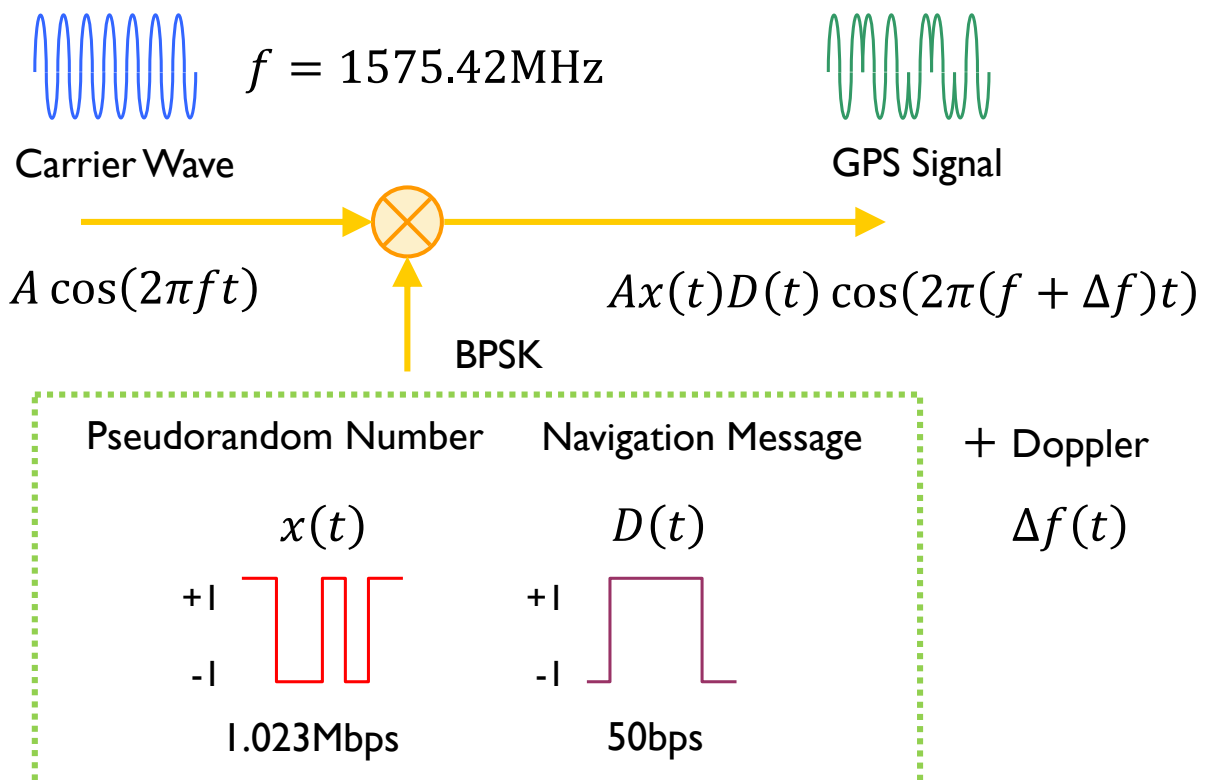
ベースバンドとドップラー

- ▶ ベースバンドにドップラーが含まれる場合、その周波数に応じてIQの座標がconstellation diagram中で回転することになる。

$$\begin{aligned}
 I(t) &= A \cos(\Delta\omega t + \phi) \\
 &= \begin{cases} +1 \cdot A \cos(\Delta\omega t) \\ -1 \cdot A \cos(\Delta\omega t) \end{cases} \\
 Q(t) &= A \sin(\Delta\omega t + \phi) \\
 &= \begin{cases} +1 \cdot A \sin(\Delta\omega t) \\ -1 \cdot A \sin(\Delta\omega t) \end{cases}
 \end{aligned}$$



GPS信号の構成



GPS信号の生成

- ▶ ソフトウェア無線でGPS信号を生成するためには、受信時刻 t における受信機アンテナ位置での擬似雑音符号 $x(t)$ および航法メッセージ $D(t)$ の値と、GPS衛星と受信機の相対運動によって発生するドップラー $\Delta f(t)$ から、直交位相振幅変調の振幅 $I(t)$ および $Q(t)$ を計算すればよい。

$$\begin{cases} I(t) = Ax(t)D(t) \cos(2\pi\Delta ft) \\ Q(t) = Ax(t)D(t) \sin(2\pi\Delta ft) \end{cases}$$

GPS信号シミュレータ

- ▶ GPS信号の $I(t)$ および $Q(t)$ を計算するオープンソースのGPS信号シミュレータとして、gps-sdr-simを公開中。
 - ▶ <https://github.com/osqzss/gps-sdr-sim>
- ▶ 生成されたベースバンドはファイルに書き出されるため、そのデータを読み込ませれば、どのソフトウェア無線機であっても、GPS信号を生成することができる。
 - ▶ ファイルサイズが大きくなるため、長時間のシミュレーションは困難である。
- ▶ 数値演算のみであるため、標準的なC言語のプログラムとなっており、機種依存なく実行ファイルがビルドできる。

gps-sdr-simのビルド

- ▶ Windows
 - ▶ Visual Studio
 - ▶ GCC (MSYS2)
- ▶ Linux
 - ▶ GCC
- ▶ MacOS
 - ▶ GCC (Homebrew)

```
$ gcc gpssim.c -lm -O3 -o gps-sdr-sim
```

gps-sdr-simのオプション

Usage: gps-sdr-sim [options]

Options:

- e <gps_nav> RINEX navigation file for GPS ephemerides (**required**)
- u <user_motion> User motion file in ECEF x, y, z format (dynamic mode)
- x <user_motion> User motion file in lat, lon, height format (dynamic mode)
- g <nmea_gga> NMEA GGA stream (dynamic mode)
- c <location> ECEF X,Y,Z in meters (static mode)
e.g. 3967283.15,1022538.18,4872414.48
- l <location> Lat,Lon,Hgt (static mode) e.g. 30.286502,120.032669,100
- t <date,time> Scenario start time YYYY/MM/DD,hh:mm:ss
- T <date,time> Overwrite TOC and TOE to scenario start time
- d <duration> Duration [sec] (default: 300)
- o <output> I/Q sampling data file (default: gpssim.bin)
- s <frequency> Sampling frequency [Hz] (default: 2600000)
- b <iq_bits> I/Q data format [1/8/16] (default: 16)
- i Disable ionospheric delay for spacecraft scenario
- v Show details about simulated channels

ベースバンドファイルの生成

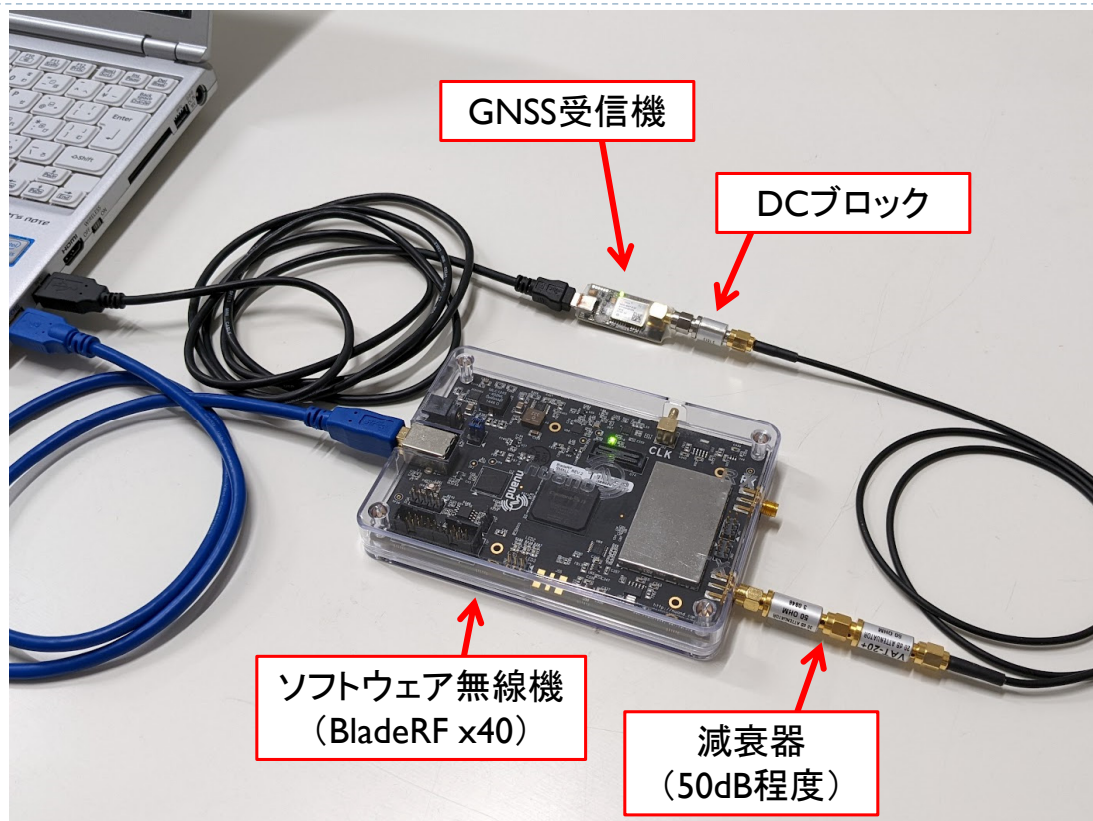
- ▶ 出力されるベースバンドのファイル名は、デフォルトで `gpssim.bin` となる。
 - ▶ `-o` オプションで任意のファイル名を指定できる。

```
gps-sdr-sim -e brdc3540.14n -u circle.csv
```

RINEX Navigation File
(必須)

User Motion File
(オプション)

GPS信号シミュレータのセットアップ

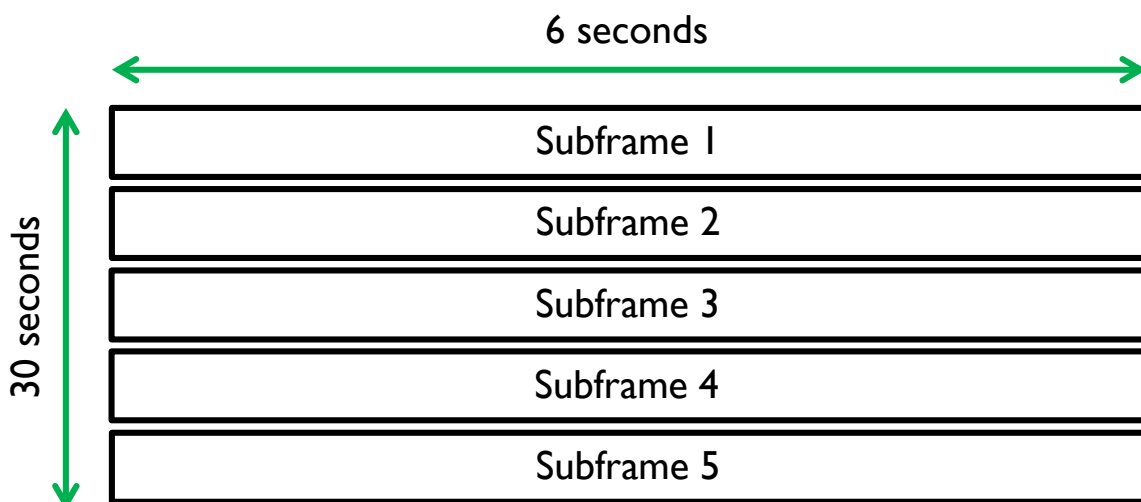


擬似距離と送信時刻

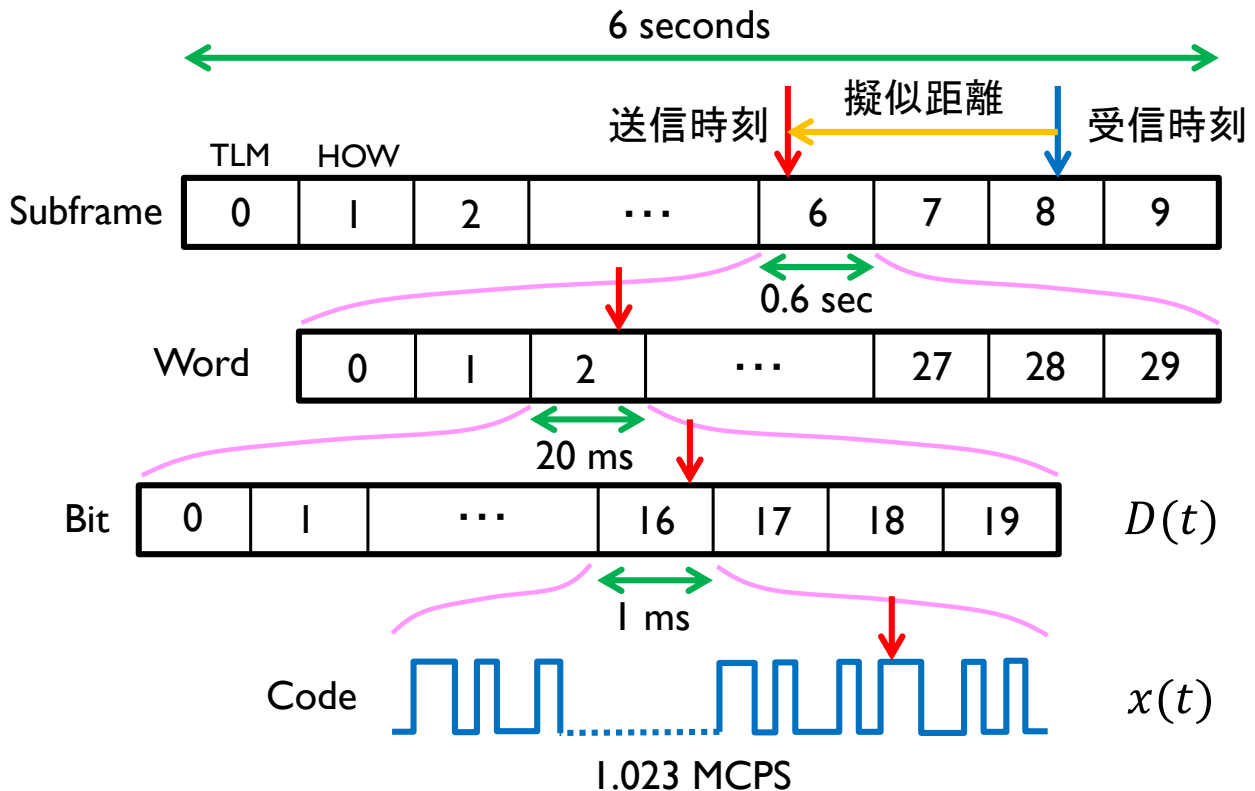
- ▶ GPS信号は、周期的に変化する擬似雑音符号と航法メッセージによって、信号の送信時刻を送信している時計である。
- ▶ 任意の受信時刻において、擬似距離分だけ過去の(遅延した)信号が届く。
- ▶ そのときに観測された擬似距離雑音および航法メッセージの位置(位相)が、その信号の送信時刻そのものである。
- ▶ 任意の時刻におけるGPS衛星および受信機の位置から擬似距離を計算し、それを光の速度で割ることで、受信時刻に観測されるGPS信号の送信時刻を求める。

GPS信号の構造

- ▶ GPS時刻の整数秒に同期した航法メッセージが30秒周期で送信される。これをフレームと呼ぶ。
- ▶ フレームは、6秒周期の5つのサブフレームから成る。



GPS信号の構造



▶ 27

GPS信号の送信時刻

- ▶ Subframe 1の航法メッセージとして、GPS信号の送信時刻に対応したGPS Week Numberが送信される。
- ▶ 各サブフレームのHOWに、次のサブフレームの先頭が送信されるTime of Week (TOW)が含まれている。
 - ▶ 各サブフレームの先頭がGPS時刻の整数6秒に同期している。
- ▶ サブフレームを構成しているビットに同期することで、20ms単位の送信時刻を知ることができる。
- ▶ 各ビットに含まれるコードに同期することで、1ms単位の送信時刻を知ることができる。
- ▶ さらに、受信信号の相関波形の最大値から、1コード未満(1ms未満)の送信時刻を知ることができる。

▶ 28

GPS信号の配列構造

- ▶ フレームの中身は、航法メッセージが更新される2時間ほどは同じであるため、gps-sdr-simでは固定値として事前に計算して、配列構造に展開している。
 - ▶ HOWに含まれるTOWは、送信時刻に応じて周期的にアップデートする。
- ▶ 擬似雑音符号も、衛星ごとに固有のパターンであるので、同様に配列で事前に定義している。
- ▶ 任意の受信時刻における擬似雑音符号 $x(t)$ および航法メッセージ $D(t)$ の値は、フレームの先頭から始まる配列から、受信機時刻に相当するインデックスで読み出せばよい。

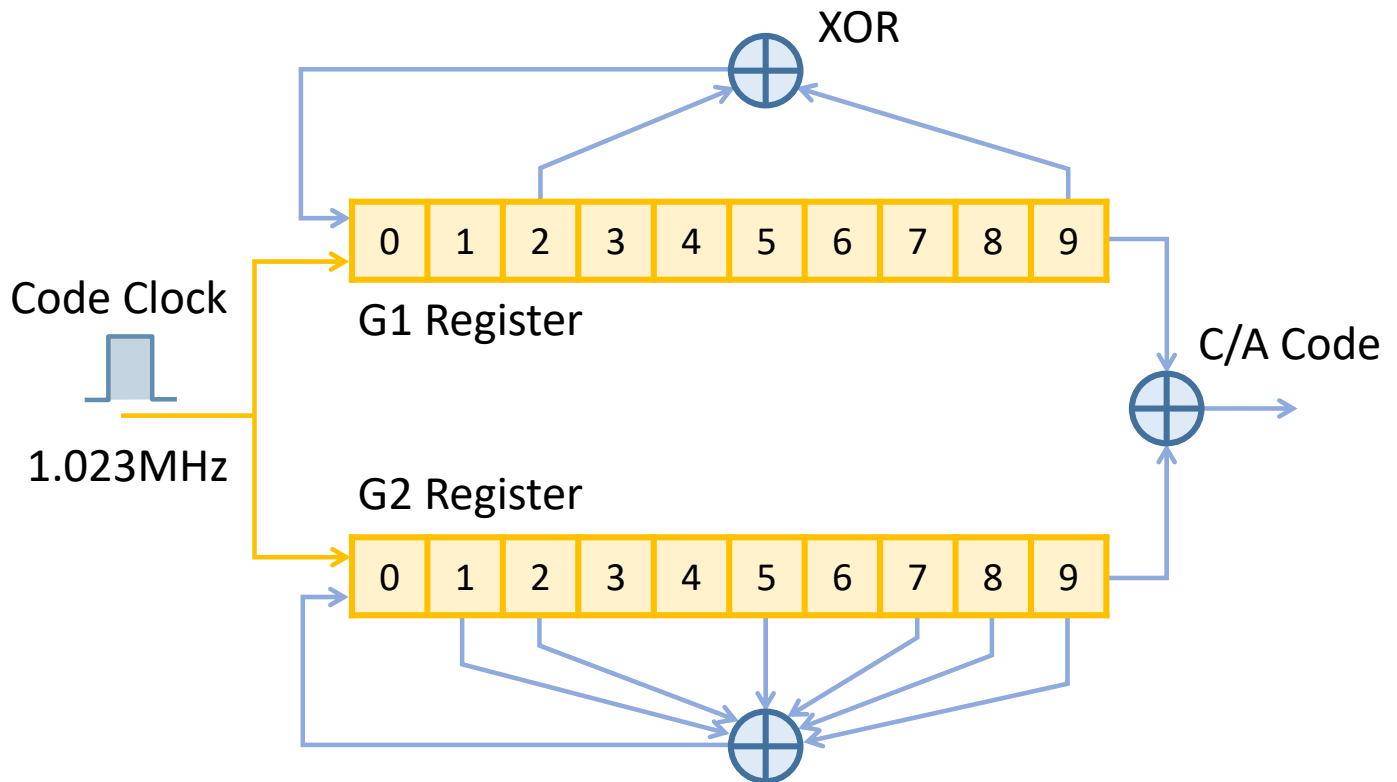
擬似雑音符号の生成

- ▶ `codegen`関数でGPS L1 C/Aコードを生成する。
- ▶ IS-GPS-200には、衛星番号に対応するタップ位置と遅延量の両方が定義されている。
 - ▶ G2レジスタのタップ位置の選択は、G2レジスタが出力するチップの遅延と等価である。
- ▶ XORの論理演算は、+1と-1の掛け算に置き換えている。
 - ▶ 論理値の(0, 1)が、数値の(+1, -1)に対応する。

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

X	Y	×
+1	+1	+1
+1	-1	-1
-1	+1	-1
-1	-1	+1

C/Aコードの生成



▶ 31

エフェメリスの読み込み

- ▶ `readRinexNavAll`関数で、RINEX Navigation Fileからエフェメリスを読み込む。
- ▶ `eph2sbf`関数で、エフェメリスの値を航法メッセージのSubframeのビット列に変換して、Word単位で保存する。
(`chan.sbf[5][10]`)
 - ▶ ビット列には、フレームの時刻を表すGPS WeekやTOWの情報は含まれていない。

▶ 32

航法メッセージの生成

- ▶ `generateNavMsg`関数で、30秒毎に1フレーム分+1サブフレームの航法メッセージをWordの配列に書き出す。
(`chan.dwrd[(5+1)*10]`)
- ▶ 最初の10ワードは、直前のSubframe 5の値をコピーする。
- ▶ その後に、現時刻以降のSubframe 1から5のワードを、対応する`chan.sbf[5][10]`からコピーする。
- ▶ さらに、GPS Week、TOW、check sumを計算して書き込む。

インデックスの生成 (1/2)

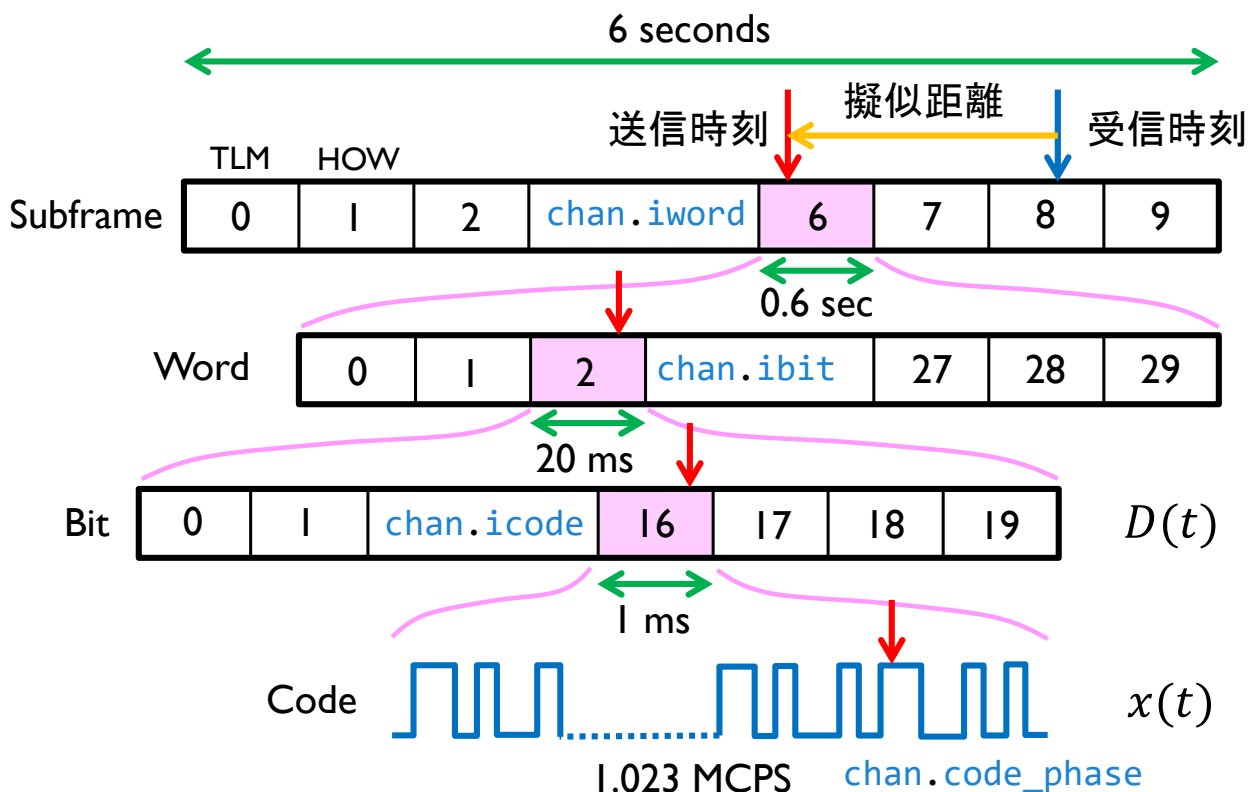
- ▶ `computeCodePhase`関数で、受信時刻における擬似雑音符号および航法メッセージの配列のインデックスを求める。
- ▶ 受信時刻からフレーム配列の先頭に対応する時刻を引くことで、受信時刻におけるフレーム位置が計算される。
 - ▶ 各サブフレームのHOWに含まれる送信時刻は、次のサブフレームの先頭の送信時刻を示している。
 - ▶ そのため、受信時刻に1サブフレーム分の時間に相当する6秒を加え、フレーム位置を進める。
- ▶ さらに、擬似距離に相当する時間を遅延させることで、送信時刻におけるフレームおよび擬似雑音符号の位置(位相)が求められる。

インデックスの生成 (1/2)

- ▶ 擬似距離に相当する時刻遅延において、ミリ秒単位の小数部に1023を掛けた実数値が、擬似距離雑音の位相(チップ単位)に相当する。(chan.code_phase)
- ▶ ミリ秒単位の時刻遅延の整数部を1ワード長(600ms)で割った値の整数部が、航法メッセージのワード位置に相当する。(chan.iword)
- ▶ さらに、その余りを1ビット長(20ms)で割った値の整数部が、受信時刻における航法メッセージのビット位置に相当する。(chan.ibit)
- ▶ さらに、その余りが航法メッセージの1ビット中に含まれるのコードのインデックスとなる。(chan.icode)

▶ 35

GPS信号の配列とインデックス



▶ 36

C/Aコードと航法メッセージの読み出し

- ▶ 受信時刻における時刻遅延から求めたインデックスによって、C/Aコードおよび航法航法メッセージの配列から、受信時刻におけるそれぞれの値を読み出す。
- ▶ さらに、読み出された2進数(0, 1)を整数値(-1, +1)に変換する。

```
chan->codeCA = chan->ca[(int)chan->code_phase]*2-1;  
chan->dataBit = (int)((chan->dwr[dchan->iword]>>(29-chan->ibit)) & 0x1UL)*2-1;
```

- ▶ `chan.codeCA`が受信時刻 t における擬似距離雑音 $x(t)$ 、`chan.dataBit`が航法メッセージ $D(t)$ に相当する。

▶ 37

ドップラーの計算

- ▶ `computeCodePhase`関数において、現時刻と直前の時刻に求められた擬似距離の差分から、レンジレートの近似値を計算する。(`rhorate`)
 - ▶ `gps-sdr-sim`では、計算負荷を下げるために、擬似距離を0.1秒毎に計算している。
- ▶ レンジレートを搬送波の波長で割り、その符号を反転させたものが、搬送波周波数(ソフトウェア無線で設定)のドップラーとなる。(`chan.f_carr`)
- ▶ さらに、擬似雑音符号の周波数に対応するドップラーは、搬送波ドップラーに搬送波とコード周波数の比を掛けたものとなる。(`chan.f_code`)

▶ 38

擬似距離の線形近似

- ▶ `main`関数において、ソフトウェア無線のサンプリング周波数に応じた擬似距離の計算が必要になるが、計算負荷が増大してしまう。
- ▶ 擬似距離は0.1秒毎(10Hz)で計算し、その間の疑似雑音符号および航法メッセージの位置(位相)は、搬送波および疑似雑音符号の周波数による線形近似で求める。

```
// Update code phase
chan[i].code_phase += chan[i].f_code * delt;

// Update carrier phase
chan[i].carr_phase += chan[i].f_carr * delt;
```

▶ 39

インデックスのアップデート

- ▶ `chan.code_phase`が1023チップを超えたら、1ビット中に含まれるコードのインデックスである`chan.icode`をカウントアップする。
- ▶ `chan.icode`が20コードを超えたら、1ワード中に含まれるビットのインデックスである`chan.ibit`をカウントアップする。
- ▶ `chan.ibit`が30ビットを超えたら、1サブフレーム中に含まれるワードのインデックスである`chan.iword`をカウントアップする。
- ▶ アップデートされたインデックスから、サンプリング周期の受信時刻 t における疑似雑音符号 $x(t)$ と航法メッセージ $D(t)$ の値を読み出す。

▶ 40

周波数成分の計算

- ▶ 搬送波のドップラーによる $I(t)$ および $Q(t)$ の周波数成分の計算に、サンプリング周期毎に三角関数を呼び出すのは計算負荷が高い。
- ▶ 1周期分の三角関数を正規化された位相をインデックスとする振幅の配列として定義する。
 - ▶ gps-sdr-simでは、1周期分の位相を512分割している。
- ▶ 現時刻のベースバンド位相である`chan.phase`の対応したインデックスは次のように求めることができる。
 - ▶ 小数部が1周期中の位相に相当する。

```
iTable = (int)floor(chan[i].carr_phase*512.0);
```

▶ 41

振幅の計算

- ▶ 受信信号の振幅は、信号の伝搬距離および受信機アンテナのゲインパターンで決まる。(gain)
 - ▶ 実際の受信信号の電力は、ソフトウェア無線機によって、減衰器を適切に挿入して調整する。
- ▶ 伝搬距離による電力の減衰は、距離の2乗に反比例する。したがって、振幅は距離に反比例する。
 - ▶ 202,000kmを基準距離として、減衰比のみを計算する。

```
path_loss = 20200000.0/rho.d;
```

- ▶ 受信アンテナのゲインパターンは、仰角5度毎にゲインを定義した配列から読み出す。(ant_pat[37])

▶ 42

ベースバンド信号の計算

- ▶ サンプル時刻における擬似雑音符号 `chan.codeCA`、航法メッセージ `chan.dataBit`、ベースバンド位相のインデックス `iTable`、振幅 `gain` から、ベースバンド信号の $I(t)$ および $Q(t)$ が求められる。

$$\begin{cases} I(t) = Ax(t)D(t) \cos(2\pi\Delta ft) \\ Q(t) = Ax(t)D(t) \sin(2\pi\Delta ft) \end{cases}$$

```
ip = chan[i].dataBit * chan[i].codeCA * cosTable512[iTable] * gain[i];  
qp = chan[i].dataBit * chan[i].codeCA * sinTable512[iTable] * gain[i];
```

発展的課題のテーマ例

- ▶ GPS L1 C/A以外の測位信号の生成
 - ▶ 複数のソフトウェア受信を同期させた多周波数シミュレータ
 - ▶ 準天頂衛星のL1 C/B信号などの新しい信号
- ▶ 信号認証サービスのシミュレーション
 - ▶ 航法メッセージに信号認証データ(電子署名)が埋め込まれる
- ▶ “Higher than GPS orbit”信号のシミュレーション
 - ▶ 静止衛星軌道や月周辺の宇宙機で受信されるGPS信号