AUBURN UNIVERSITY

Overview of GPS Pseudorange Derivation

MECH 5970/6970
Fundamentals of GPS

Tracking GPS Signal

- We have seen how to calculate PVT from pseudorange (and doppler) measurements
- We are now developing tracking signals to match carrier and code phase
  - PLL
  - DLL
- The costas PLL will provide the navigation data bits
  - Provides the satellite ephemeris and almanac
  - Also used with the outputs of the PLL and DLL to form doppler and pseudoranges

Costas Loop

- The costas PLL loop has two branches to make it imune to data bit transitions
  - I (in-phase, i.e. the sine signal)
  - Q (quadrature, i.e, the cosine signal)
  - Discriminator (i.e. error calculation) is the 2-quadrant arctangent
  - $\theta_{err} = tan^{-1}\left(\frac{\sum Q}{\sum I}\right)$
  - Data Bit Estimate = $\pm \sum I$
    - Note that the costas PLL loop has an ambiguous solution
    - This means the data bits can be “inverted”

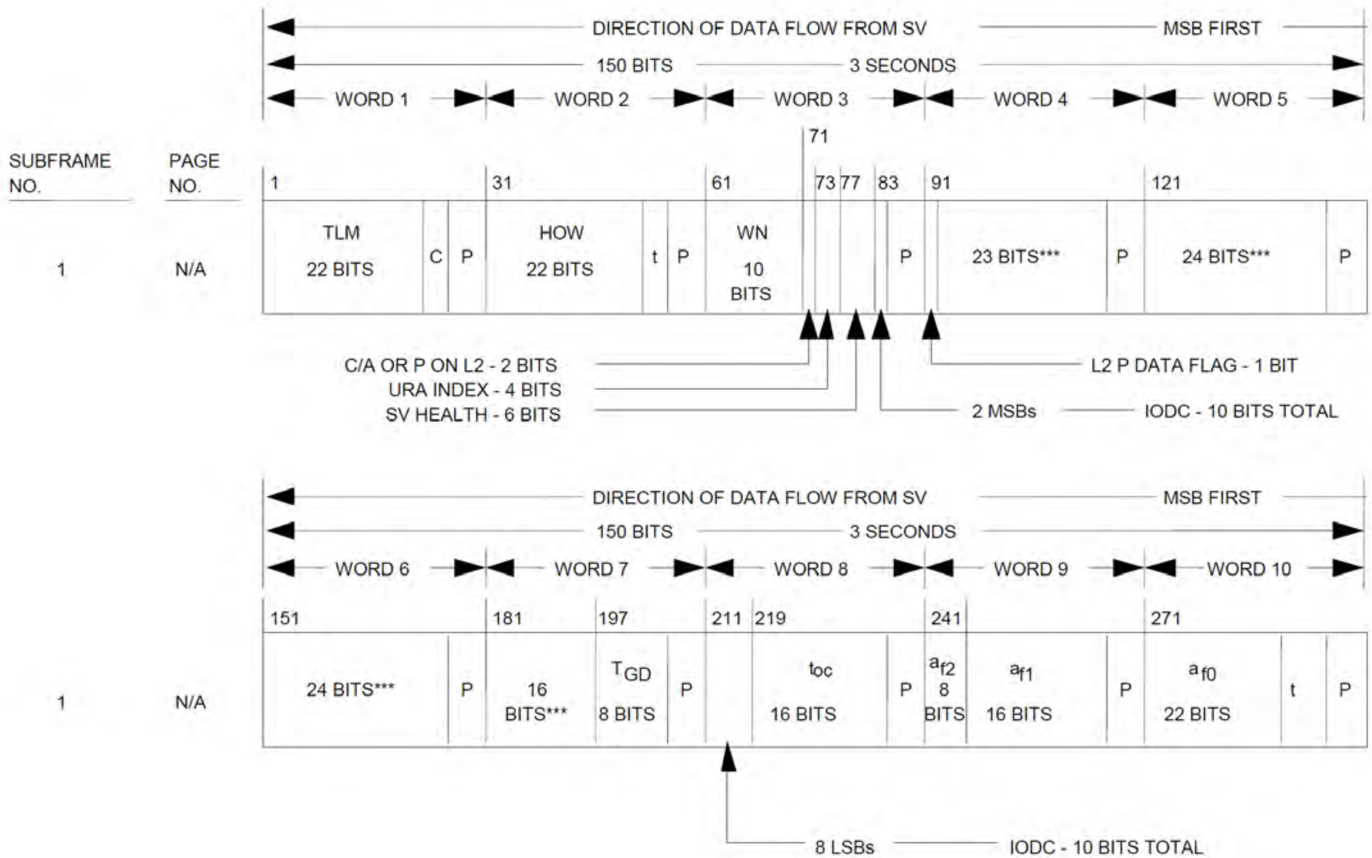
Navigation Bits

- Recall the navigation data bits are divided into 5 subframes.
  - Each subframe is 6 seconds
    - Each navigation bit is 20 ms
  - Each subframe is divided into 30 bit “words”
    - 10 words per subframe
  - 5 subframes provides all the data (ephemeris) for a single satellite
    - So all data is available in 30 seconds
  - 25 frames required for all of the navigation data (i.e. all of the almanac)
    - 12.5 minutes

Navigation Bits

- Word 1 is called the TLM (telemetry)
  - First 8 bits are a preamble for synchronization
- Word 2 is the HOW (handover word)
  - First 17 bits (bits 31-47) are the GPS Time of Week (TOW) which corresponds to the time at the end of the subframe
- Word 3 of subframe #1 is GPS Week # (WN)
  - Bits 61-70 (of subframe #1).
  - This is the 10 MSB of TOW.

Subframe 1 (from GPS ICD)



*** RESERVED
P = 6 PARITY BITS
t = 2 NONINFORMATION BEARING BITS USED FOR PARITY COMPUTATION (SEE PARAGRAPH 20.3.5)
C = TLM BITS 23 AND 24. BIT 23 IS THE INTEGRITY STATUS FLAG AND BIT 24 IS RESERVED

Preamble

- The start of each subframe is an 8 bit preamble for synchronization
- Preamble: [1 -1 -1 -1 1 -1 1 1]
  - Must do 2 checks:
    1) Check for inverted bits: [-1 1 1 1 -1 1 -1 -1]
    2) Check that it is repeated in 6 seconds (300 bits).
       - This is to ensure that it was the preamble and not a coincidence of those 8 bits.

Z-count

- The preamble is followed by the z-count.
- GPS time is stored as 29 bit z-count
  – The 10 MSB is week number (WN)
  – The 19 LSB is time of week (TOW) in 1.5 second increments
  – The 17 MSB of the 19 LSB TOW is in 6 second increments (i.e. length of the subframe)
  – This time corresponds to the end of the subframe (which is the start of the next subframe)
    - So the start of the subframe is the z-count TOW minus 6 seconds

GPS time

- Current time within in the week (i.e. TOW) is:

  tow_z_count*1.5 (from last subframe)

  + number of bits in current subframe * 20 ms

  + code_cycles*1 ms

  + code_chips * 1/1023 ms

  + fraction code_chip * 1/1023 ms
- Remember the navigation data only has the 17 MSB of the z_count TOW
  - Handle in one of two ways:

    a) [nav_tow_z_count 0 0]*1.5

    b) (nav_tow_z_count*4)*1.5=nav_tow_z_count*6

Pseudorange computation

- Because of the clock uncertainty, exact received time is not (initially) known.
- However, it's the relative ranges that are critical to determine user position
- Therefore, we can pick one SV as a starting point
  - Use closest SV (closest to zenith/overhead) or the one with the smallest code phase from the DLL
  - Assume a nominal time of flight of 70 ms
    - Beware of wrap across subframes

Pseudorange Estimates

- Order the pseudoranges based on arrival time
  - Arrival time estimate comes from comparing the times for each SV from the DLL

$$\rho_0 = c \times 0.7$$
$$\rho_1 = \rho_0 + c \times (t_1 - t_0)$$
$$\rho_2 = \rho_1 + c \times (t_2 - t_1)$$

- Recall that within a data bit are 20 CA codes!

Pseudorange Estimates

- Getting the pseudoranges is a bit of a book-keeping exercise.
- You must either:
  - Determine the sample number for the start of the same subframe across all channels (SVs)
    - Longer pseudoranges means the start of the same subframe will be later in the sampled data
  - At the same sample point determine how far into the subframe each channel (SV) is
    - This is done by counting data bits into the subframe (from the start of the subframe) plus the number of code cycles plus the number of code chips

DLL Book-keeping

- Additionally, you must be careful that your integrate and dump occurs over full code cycles
  - This is to ensure there is no data bit transition in the middle of a DLL integrate.
  - So generally DLL integrate periods are in intervals of 1 ms increments up to a max of 20 ms
    - So 1 ms, 5 ms, 10 ms, or 20 ms are common.
    - Higher integration periods can be used (i.e. additional averaging) if or when the data bits are known.
- Therefore, you must monitor the code error and ensure your integration period occurs over samples of data that guarantee no data bit transition.
  - This is easiest if using 1 ms integration period (at least initially)