Combining secondary code correlations for fast GNSS signal acquisition

Jérôme Leclère
René Jr Landry
1. Introduction

2. How to extend the coherent integration time?

3. Solution: Combining secondary code correlations

4. Application to the GPS L5 signal

5. Conclusion
A GNSS signal contains three essential elements:

- Data (ephemeris, corrections, …)
- Pseudorandom code (distance measurement, spectrum sharing)
- Carrier

Modern signals include new features:

- Pilot channel (allows long coherent integration time)
- Subcarrier (improves multipath mitigation)
- Secondary pseudorandom code
1. Introduction
1.2 GNSS signals

Model of a GNSS signal transmitted:

\[ s_t(t) = a_{t,d} c_d(t) d(t) \cos(2\pi f_L t + \varphi_{t,d}) + a_{t,p} c_p(t) \sin(2\pi f_L t + \varphi_{t,p}) \]

Model of a GNSS signal received:

\[ s_r(t) = \frac{1}{L} s_t(1 + \alpha(t - \tau)) + \eta(t) \]

\[ = a_{r,d} c_d(1 + \alpha(t - \tau)) d(1 + \alpha(t - \tau)) \cos(2\pi (f_L + f_D) t + \varphi_{r,d}) + a_{r,p} c_p(1 + \alpha(t - \tau)) \sin(2\pi (f_L + f_D) t + \varphi_{r,p}) \]
1. Introduction
1.2 Goal of the acquisition

The receiver wants to:
- decode the data
- synchronize with the code

For this, the receiver needs to:
- estimate the Doppler frequency $f_D$ to remove the carrier
- estimate the code delay $\tau$ to synchronize with the code and remove it

This is the goal of the acquisition.
1. Introduction

1.3 How do we do acquisition?

Acquisition is basically a correlation + Doppler search.

Actual implementations use high parallelization or fast Fourier transforms (FFTs).

![Diagram showing acquisition process]

Incoming signal

- Local carrier replica of frequency $\hat{f}_D$
- Local code replica of delay $\hat{\tau}$

Result:

- Cross ambiguity function (CAF)

Repeated for different $\hat{\tau}$ and $\hat{f}_D$
1. Introduction
1.3 How do we do acquisition?

If \( \hat{f}_D = f_D \) and \( \hat{\tau} = \tau \) (both replicas match the input) \( \Rightarrow \) high CAF

If \( \hat{f}_D \neq f_D \) or \( \hat{\tau} \neq \tau \) (at least 1 replica does not match) \( \Rightarrow \) low CAF

Example:
L1 C/A signal
\( f_D = 2000 \text{ Hz} \)
\( \tau = 200 \text{ chips} \)
1 ms integration
1. Introduction
1.4 Why the integration time is important?

The other challenge of the acquisition is the noise.

To reduce the noise as much as possible the integration time should be as long as possible.
1. Introduction

1.4 Why the integration time is important?

But there are limitations: data, dynamics, XO, freq. resolution…

One way to overcome these drawbacks is to continue integrating after taking the magnitude:

\[
\begin{align*}
\text{incoming signal} & \xrightarrow{\text{local carrier replica of frequency } \hat{f}_D} \text{local code replica of delay } \hat{\tau} \\
& \xrightarrow{\text{coherent integration}} | \cdot | \xrightarrow{\text{non-coherent integration}} \text{cross ambiguity function (CAF)}
\end{align*}
\]
1. Introduction

1.4 Why the integration time is important?

However, in terms of detection, coherent integration is more efficient than non-coherent integration.
Modern GNSS signals include a secondary code.

Advantages:
- Easier synchronization with the data
- Better cross-correlation between satellites/constellations
- Better interferences mitigation

Drawbacks:
- Complicates the primary code correlation for FFT-based implementations (sign transition possible anytime)
- Complicates the extension of the coherent integration time
1. Introduction

2. How to extend the coherent integration time?

3. Solution: Combining secondary code correlations

4. Application to the GPS L5 signal

5. Conclusion
2. How to extend the coherent integration time?

2.1 Possibilities

Mainly two possibilities to extend the coherent integration time in presence of a secondary code:

- Use short coherent integration time, $T_C < T_{SC}$
- Use long coherent integration time, $T_C \geq T_{SC}$

$T_C$: coherent integration time (s)

$T_{SC}$: period of the secondary code (s)
2. How to extend the coherent integration time?

2.2 Short coherent integration time

If have $T_C < T_{SC}$, all the possible sign sequences should be tested.

Parallel implementation with coherent integration only: 
($N_C = 3$)

Drawback: number of possibilities grows exponentially

$\Rightarrow$ limited to 4 or 5 primary code periods.
2. How to extend the coherent integration time?

2.2 Short coherent integration time

If non-coherent integration is used too, more possibilities

Parallel implementation with coherent and non-coherent integrations: \((N_C = 2, N_{NC} = 2)\)

Drawback: number of possibilities grows double exponentially
2. How to extend the coherent integration time?

2.2 Short coherent integration time

<table>
<thead>
<tr>
<th>$N_C$</th>
<th>$N_{NC}$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

Number of coherent accumulators

\[ = 2^{N_C - 1} \]

<table>
<thead>
<tr>
<th>$N_{NC}$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>16</td>
<td>64</td>
<td>256</td>
<td>1024</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>64</td>
<td>512</td>
<td>4096</td>
<td>32768</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>256</td>
<td>4096</td>
<td>65536</td>
<td>1048576</td>
</tr>
</tbody>
</table>

Number of non-coherent accumulators

\[ = (2^{N_C - 1})^{N_{NC}} \]

Number of accumulators explodes with non-coherent integration

- Non-coherent integration not usable
- Stuck to low sensitivity
2. How to extend the coherent integration time?

2.3 Long coherent integration time

To have $T_C \geq T_{SC}$, correlation with the secondary code is needed.

Parallel implementation:

Serial implementation:

Drawback:
- Huge computational burden
- Huge resources or long acquisition time.

$N_S$: secondary code length
### 2. How to extend the coherent integration time?

#### 2.4 Solution: intermediate coherent integration time

<table>
<thead>
<tr>
<th>Method</th>
<th>Use of non-coherent integration?</th>
<th>Sensitivity</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>long $T_C$</td>
<td>✓</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>short $T_C$</td>
<td>✗</td>
<td>low</td>
<td>low to moderate</td>
</tr>
<tr>
<td>proposed solution with $T_C$</td>
<td>✓</td>
<td>moderate to high</td>
<td>low to moderate</td>
</tr>
</tbody>
</table>
Content

1. Introduction

2. How to extend the coherent integration time?

3. Solution: Combining secondary code correlations

4. Application to the GPS L5 signal

5. Conclusion
3. Solution: Combining secondary code correlations

3.1 Why combining secondary code correlations?

Combining secondary code correlations reduce the complexity:

- There are less delays to search
  - ⇒ Shorter processing time for the serial implementation
  - ⇒ Less accumulators for the parallel implementation

- Inputs may be combined
  - ⇒ Less primary code correlations to compute
  - ⇒ Shorter local secondary code

- The local modified secondary code may contain zeros
  - ⇒ Less operations for the secondary code correlation
3. Solution: Combining secondary code correlations

3.2 How combining secondary code correlations?

Not every combination gives the same complexity reduction! Let’s see a simple example for a 4-chip code.

Traditional correlation:

\[
\begin{bmatrix}
 y_0 \\
 y_1 \\
 y_2 \\
 y_3 \\
\end{bmatrix} = \begin{bmatrix}
 s_0 & s_1 & s_2 & s_3 \\
 s_3 & s_0 & s_1 & s_2 \\
 s_2 & s_3 & s_0 & s_1 \\
 s_1 & s_2 & s_3 & s_0 \\
\end{bmatrix}
\begin{bmatrix}
 r_0 \\
 r_1 \\
 r_2 \\
 r_3 \\
\end{bmatrix}
\]

computed 4 times
3. Solution: Combining secondary code correlations

3.2 How combining secondary code correlations?

Combination of one result with the next one:

\[
\begin{bmatrix}
\mathbf{y}_0 + \mathbf{y}_1 \\
\mathbf{y}_2 + \mathbf{y}_3
\end{bmatrix} =
\begin{bmatrix}
s_0 + s_3 & s_1 + s_0 & s_2 + s_1 & s_3 + s_2 \\
 s_2 + s_1 & s_3 + s_2 & s_0 + s_3 & s_1 + s_0
\end{bmatrix}
\begin{bmatrix}
r_0 \\
r_1 \\
r_2 \\
r_3
\end{bmatrix}
\]

\[
\begin{align*}
\mathbf{x}_i & \quad \text{Primary code correlation} \\
\mathbf{r}_i & \quad \text{still computed 4 times} \\
\sum_{i=0}^{N_p} s_i + s_{i-1} & \quad i = 0, 1, 2, 3
\end{align*}
\]

\[
\begin{align*}
\sum_{i=0}^{N_S-1} \mathbf{y}_0 + \mathbf{y}_1 & \quad | \cdot | \\
\sum_{i=0}^{N_S-1} \mathbf{y}_2 + \mathbf{y}_3 & \quad | \cdot | \\
\sum_{k=0}^{N_NC} z_0 + z_1 & \quad z_2 + z_3
\end{align*}
\]
3. Solution: Combining secondary code correlations

3.2 How combining secondary code correlations?

Combination of one result with the next but one:

\[
\begin{bmatrix}
  y_0 + y_2 \\
  y_1 + y_3
\end{bmatrix} = \begin{bmatrix}
  s_0 + s_2 & s_1 + s_3 & s_2 + s_0 & s_3 + s_1 \\
  s_3 + s_1 & s_0 + s_2 & s_1 + s_3 & s_2 + s_0
\end{bmatrix} \begin{bmatrix}
  r_0 \\
  r_1 \\
  r_2 \\
  r_3
\end{bmatrix}
\]

computed 2 times!
3. Solution : Combining secondary code correlations

3.2 How combining secondary code correlations?

Not every combination gives the same performance!
Let’s see a simple example for the 6-chip code $[1 1 1 -1 1 -1]$.

Autocorrelation : $[y_0 \ y_1 \ y_2 \ y_3 \ y_4 \ y_5] = [6 \ -2 \ 2 \ -2 \ 2 \ -2]$

Combination 1 : $[y_0 + y_2 + y_4 \ y_1 + y_3 + y_5] = [10 \ -6]$

Combination 2 : $[y_0 + y_1 + y_2 \ y_3 + y_4 + y_5] = [6 \ -2]$ or $[2 \ 2]$

The maximum depends on the delay of the incoming code!
3. Solution: Combining secondary code correlations

3.2 How combining secondary code correlations?

Rules:

1. Combine $N_{RC}$ secondary code correlation results separated by a delay $\frac{N_S}{N_{RC}}$.
2. Sign of the combinations: only $+$, or alternatively $+$ and $-$.

Then:

- Maximum autocorrelation identical regardless of the delay of the incoming secondary code.
- Number of secondary code delays to test is divided by $N_{RC}$.
- $N_{RC}$ inputs can be combined.

$\Rightarrow$ Complexity is approximately divided by $N_{RC}^2$. 
3. Solution: Combining secondary code correlations

3.3 Impact on the SNR

This does not come for free:

- Result modified $\Rightarrow$ loss in the signal-to-noise ratio (SNR)

Using $T_{SC}$ of data, instead of a SNR gain of $N_S$, the gain is

$$G = \frac{\sum_{i=0}^{N_S-1} s_{i-k_1} s_{i-m_S} \pm \sum_{i=0}^{N_S-1} s_{i-k_2} s_{i-m_S} \pm \cdots \pm \sum_{i=0}^{N_S-1} s_{i-k_{N_{RC}}} s_{i-m_S}}{\sum_{i=0}^{N_S-1} \left( s_{i-k_1} \pm s_{i-k_2} \pm \cdots \pm s_{i-k_{N_{RC}}} \right)^2}$$

Easy to express through an equivalent coherent integration time.
1. Introduction

2. How to extend the coherent integration time ?

3. Solution : Combining secondary code correlations

4. Application to the GPS L5 signal

5. Conclusion
4. Application to the L5 signal
4.1 Secondary code autocorrelation

Autocorrelation of the L5 pilot secondary code (20 chips)

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| Auto-correlation | 20 | 0 | 0 | 0 | 0 | 0 | -4 | 0 | 4 | 0 | -4 | 0 | 4 | 0 | -4 | 0 | 0 | 0 | 0 | 0 |

![Graph showing the autocorrelation of the L5 pilot secondary code (20 chips)](attachment:image)
### 4. Application to the L5 signal

#### 4.2 Possible combinations

<table>
<thead>
<tr>
<th>Combination</th>
<th>( N_{RC} )</th>
<th>Resulting code and correlation</th>
<th>Numerator of ( G )</th>
<th>Denominator of ( G )</th>
<th>SNR gain ( G )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ( \sum_{k=0}^{1} y_{10k} )</td>
<td>2</td>
<td>2 0 2 0 2 0 0 0 0 -2 0</td>
<td>16(^2) = 256</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 0 4 0 -4 0 -4 0 4 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 ( \sum_{k=0}^{1} (-1)^k y_{10k} )</td>
<td>2</td>
<td>0 2 0 2 0 -2 2 2 0 -2</td>
<td>24(^2) = 576</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>±24 0 ±4 0 ±4 0 ±4 0 ±4 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 ( \sum_{k=0}^{3} y_{5k} )</td>
<td>4</td>
<td>2 0 2 -2 2 2</td>
<td>16(^2) = 256</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 -4 4 4 -4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 ( \sum_{k=0}^{3} (-1)^k y_{5k} )</td>
<td>4</td>
<td>2 0 2 2 2 2</td>
<td>16(^2) = 256</td>
<td>64</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>±16 ±4 ±4 ±4 ±4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 ( \sum_{k=0}^{4} y_{4k} )</td>
<td>5</td>
<td>1 -3 3 3</td>
<td>28(^2) = 784</td>
<td>140</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28 0 -12 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4. Application to the L5 signal

#### 4.2 Possible combinations

<table>
<thead>
<tr>
<th>Combination</th>
<th>$N_{RC}$</th>
<th>Resulting code and correlation</th>
<th>Numerator of $G$</th>
<th>Denominator of $G$</th>
<th>SNR gain $G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 $\sum_{k=0}^{9} y_{2k}$</td>
<td>10</td>
<td>$\begin{array}{c} 4 \ 16 \end{array}$</td>
<td>$16^2 = 256$</td>
<td>160</td>
<td>1.6</td>
</tr>
<tr>
<td>7 $\sum_{k=0}^{9} (-1)^k y_{2k}$</td>
<td>10</td>
<td>$\begin{array}{c} -2 \ \pm40 \end{array}$</td>
<td>$40^2 = 1600$</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>8 $\sum_{k=0}^{19} y_k$</td>
<td>20</td>
<td>$\begin{array}{c} 4 \ 16 \end{array}$</td>
<td>$16^2 = 256$</td>
<td>320</td>
<td>0.8</td>
</tr>
<tr>
<td>9 $\sum_{k=0}^{19} (-1)^k y_k$</td>
<td>20</td>
<td>$\begin{array}{c} 4 \ \pm16 \end{array}$</td>
<td>$16^2 = 256$</td>
<td>320</td>
<td>0.8</td>
</tr>
</tbody>
</table>
## 4. Application to the L5 signal

### 4.3 Proposed method vs short coherent integration time

<table>
<thead>
<tr>
<th>Number of</th>
<th>$N_{RC}$</th>
<th>$T_C$</th>
<th>$\frac{N_S}{N_{RC}}$</th>
<th>$N'_{S,NZ}$</th>
<th>Hardware</th>
<th>Software</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Parallel</td>
<td>Serial</td>
<td></td>
</tr>
<tr>
<td>memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>16</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>primary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>code correlation</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>additions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.6</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>36</td>
</tr>
</tbody>
</table>

Moreover, the proposed solution allows non-coherent integration!
### 4. Application to the L5 signal

#### 4.4 Proposed method vs long coherent integration time

<table>
<thead>
<tr>
<th>Number of</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{RC}$</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Primary code correlation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Additions</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

⇒ significant complexity reduction for smaller $T_C$

 or slight complexity reduction for slightly longer $T_C$ (it’s a win-win)
4. Application to the L5 signal

4.4 Proposed method vs long coherent integration time

Primary code correlations + secondary code correlation

Secondary code correlation only
1. Introduction

2. How to extend the coherent integration time?

3. Solution: Combining secondary code correlations

4. Application to the GPS L5 signal

5. Conclusion
5. Conclusion
5.1 Conclusions

The secondary code complicates the acquisition:
- Short coherent integrations limited to low sensitivities
- Long coherent integrations have high computational load

A solution has been proposed fitting between current solutions:
- Combine secondary code correlation results
- Intermediate coherent integration times + non-coherent integration usable
- Allow moderate to high sensitivity
- Flexible due to the several possible combinations

Drawbacks:
- Use more data than the integration time
- Require several simultaneous accesses to input data
- Still an ambiguity in the secondary code delay
Thank you for your attention!

jerome.leclere@lassena.etsmtl.ca

http://lassena.etsmtl.ca