Document ID
P-WOP-NOT-00009-RSE

Date Released 2012-03-24	Issue 4
Classification Unclassified	Doc.Status Released

Distribution	Alt. Document ID	Alt. Issue	Page
Acc.to distribution list in DOC			1(33)

PROJECT

Radio Occultation Study

TITLE

CSM and IRM User Manual

Issued by Magnus Bonnedal	Function Systems Engineer	
Approved by Magnus Bonnedal	Function	Date 2012-03-24

RUAG Space AB

Postal address	Telephone	Telefax	Registered number	VAT number
SE-405 15 Göteborg Sweden	+46 (0)31 735 00 00	+46 (0)31 735 40 00	556134-2204	SE556134220401

DOCUMENT CHANGE RECORD

Changes between issues are marked with an outside-bar.

Issue	Date	Paragraphs affected	Change information
1	2011-07-07	All	New document
2	2011-09-07	2.3, 3.2, 3.5.2, 4.2, 4.4, 4.8.1, 4.8.4, 5	Comments by EUMETSAT 2011-07-28 accounted for.
3	2012-03-24		Minor corrections
4	See header		Proprietary text removed

TABLE OF CONTENTS PAGE

Released

1 INTRODUCTION

1.1 Purpose

This document describes the design of the coded signal module, CSM. It also provides the user's manual for the simulator.

1.2 Overall System Description

The WOP and Radio Occultations Program Package, ROPP, modules have previously been used to simulate atmospheric profiles with a single carrier in two frequency bands, Figure 1.

The CSM will simulate the properties of a GNSS signal modulated by the atmosphere, using the WOP propagated signals for multiple frequencies within the signal bandwidth. The WOP IFM (InterFace Module) combines the WOP output data into a format containing phase and amplitude for a number of frequencies as required by CSM. An Ideal Receiver Model, IRM, will extract the carrier amplitude and phase for a tracked modulated signal, see Figure 1.

The IRM results for two signals (and hence two CSM runs) in different bands will in the ROPP IFM, be packed in one file to be used by ROPP.

The CSM provides a baseband coded signal, sampled at ~30 MHz, that can be used as a well controlled input to a versatile receiver module to develop and test acquisition and tracking schemes. For the purpose of verification, an Ideal Receiver Module, IRM, is used to extract the carrier phase and amplitude used as input to ROPP. In order to save computer time, the generation of the baseband signal output from CSM can be switched off, and CSM provides code and carrier phase and the complex correlator function around the proper code phase, as from a large number of receiver correlators. This latter output can also serve as input to a receiver model that focuses on open and closed loop tracking.

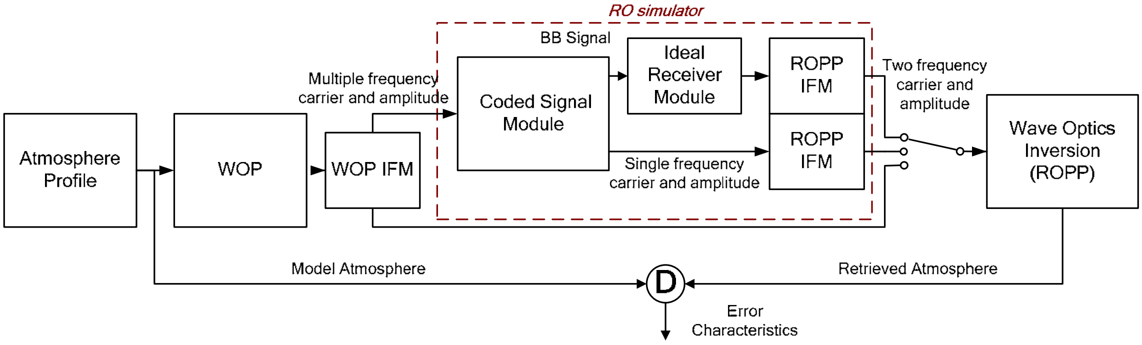


Figure 1 High level block diagram for atmospheric simulator, showing the position of the Coded Signal Module, CSM and the Ideal Receiver Module, IRM.

2 DOCUMENTS

2.1 Applicable Documents

Acronym	Description
[SoW]	Statement of Work for a Study on Optimisation of Tracking Strategies for Radio Occultation, EUM/PEPS/SOW/10/0024, Issue v2B, 8 June 2010
[Prop]	Optimisation of Tracking Strategies for Radio Occultation, Part one – Technical and Management Proposal, PRO-EUM-ITT-2010-part-1, Version 1.0

2.2 Reference Documents

None.

2.3 Abbreviations

Abbreviation	Meaning
ASCII	the American Standard Code for Information Interchange
BB	Baseband (signal, in-phase (I) and quadrature-phase (Q) component)
CSM	Coded Signal Module
E	Early (correlator value)
EE	Early Early (correlator value)
FFT	Fast Fourier Transform
i/f	Interface
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
IFM	InterFace Module
IMT	Instrument Measurement Time
IRM	Ideal Receiver Module
L	Late (correlator value)
LL	Late Late (correlator value)
LSB	Least Significant Bit
NCO	Numerically Controlled Oscillator
P	Punctual (correlator value)
PVT	Position, Velocity and Time (navigation solution)
ROPP	Radio Occultation Processing Package
S/W	Software
WOP	Wave Optics Propagation (module)

3 TEST BENCH ENVIRONMENT

3.1 Overview

In order to verify the individual modules, a test bench environment is created, see Figure 2. The Ideal Receiver Module, IRM, will eventually be substituted for a more realistic receiver module including receiver and external errors, where acquisition and tracking concepts can be tested. Results from Wave Optics Propagation simulations of realistic atmospheres will be stored in files, which can be processed with a number of different GNSS codes. These results are also stored in files, which are used as input for the receiver model processing. An interface to enable subsequent processing by ROPP is also provided for.

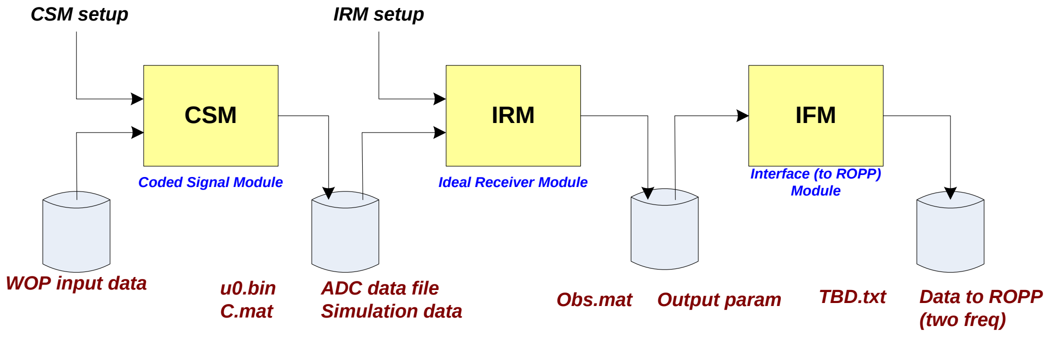


Figure 2 CSM/IRM test bench overview.

3.2 Coded Signal Module (CSM)

The Coded Signal Module (CSM) uses the WOP simulations for a set of frequencies to generate a baseband signal representing the full modulation of a selected GNSS signal after propagating through the atmosphere. The basic principle is to treat the atmosphere simulations from WOP as the frequency response of a filter. The selected GNSS signal, represented by its spectrum, is multiplied with the filter spectrum and inverse Fourier transformed. This will generate the modulated signal at the receiver. The signal will include the impact from the travelled range and the atmosphere on amplitude code phase and carrier phase.

In the CSM, the signal modulation is processed in batches of the code epoch or a fractional part of the code epoch. The signal is transmitted at time intervals of 1 ms, and the processing is for every ms performed together with the previous and the following ms. This in order to account for the delay of the signal and make sure the simulated signal covers the time interval we simulate.

A high level block diagram is shown in Figure 3. For every ms WOP sample we do the following processing; The WOP complex amplitude is upsampled in frequency by interpolation. The delay due to the geometry is added to the phase and this spectrum of the atmosphere is multiplied with the spectrum of the modulated signal received from the GNSS signal generator. After an inverse FFT and interpolation from one ms sample to the next, the 30 MHz signal is output on file.

In a second branch, the interpolated complex amplitude for the excess phase is multiplied with the GNSS signal spectrum. In the Ideal Reference Receiver block (not to be confused with the IRM), the Fourier spectrum of the signal modulated by the atmosphere is again multiplied with the complex conjugate of the original signal and IFFT to form the correlation function. Detecting the peak of this correlation function provides the signal delay, while evaluating the complex signal at the peak will provide the received amplitude and phase. All this information can be used as a secondary truth data to be compared with the output from the IRM.

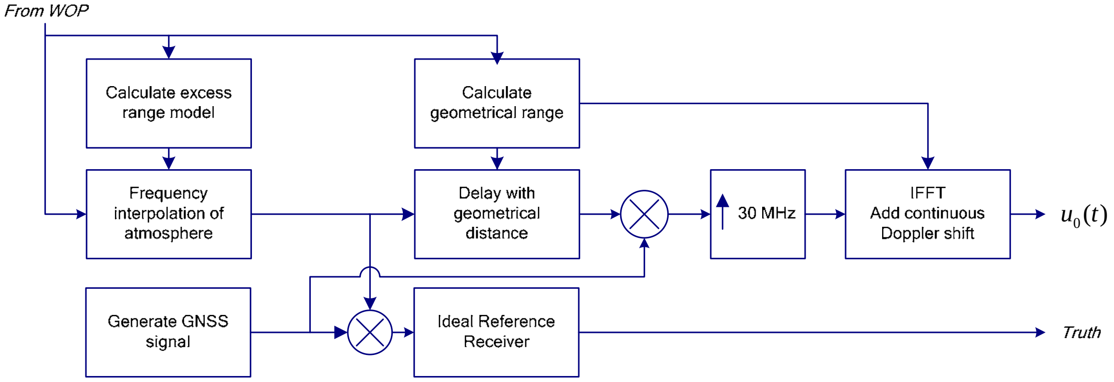


Figure 3 Coded Signal Module, CSM, overview.

Referring to Figure 4, we describe in some more detail the processing steps, repeated for every time sample. The numbering refers to Figure 4

WOP Data Processing:

1. Amplitude and excess phase is read from file and interpolated to a denser frequency spacing in order to properly match the code spectral representation. An excess range model is subtracted before and added after the interpolation in order to eliminate phase cycle ambiguities.
2. Range phase is added to the excess phase obtained from WOP, with this representing the entire transmission from transmitter to receiver.

Coded Signal:

3. A 3 ms sequence of the desired upsampled GNSS coded signal is loaded from file at a high sampling rate.
4. Navigation bit modulation is introduced as a phase shift in the data sequence and the signal is down-sampled to 20.46 MHz.
5. The code sequence is Fourier transformed.
6. The Fourier spectrum is band limited (filtered) to the actual GNSS signal bandwidth.

Multiplication and Up-sampling (this part can be switched on or off)

7. The atmosphere spectral representation and the signal spectral representation are multiplied, now representing the signal at the receiver including the range and atmospheric delays.
8. The spectrum is zero-padded to reach the desired 30 MHz sampling rate.
9. The spectrum is inverse Fourier transformed to the time representation.

10. The desired time fraction of the signal is selected. The mid 1 ms sequence is cut out for transmission.
11. The phase is linearly interpolated from one original WOP sample point to the next.
12. The signal is stored on file in binary format.

Ideal Reference Receiver Producing Correlator Functions and Truth Data Generation (this part can be switched on or off)

13. The excess atmosphere spectrum is multiplied by the coded signal spectrum.
14. The received signal spectrum is multiplied by the conjugate of the coded signal spectrum.
15. The resulting spectrum is inverse Fourier transformed, representing the correlation function.
16. The peak of the correlation function is identified, representing the time delay of the modulated signal.
17. Amplitude and phase is extracted for this peak value and is together the central part of the correlation function stored on file
18. The peak value is together the central part of the correlation function stored on file.

Generation of Approximate Truth Data

19. The centre frequency amplitude and phase are stored on file together with the signal (code) delay, to be used as approximate truth data.

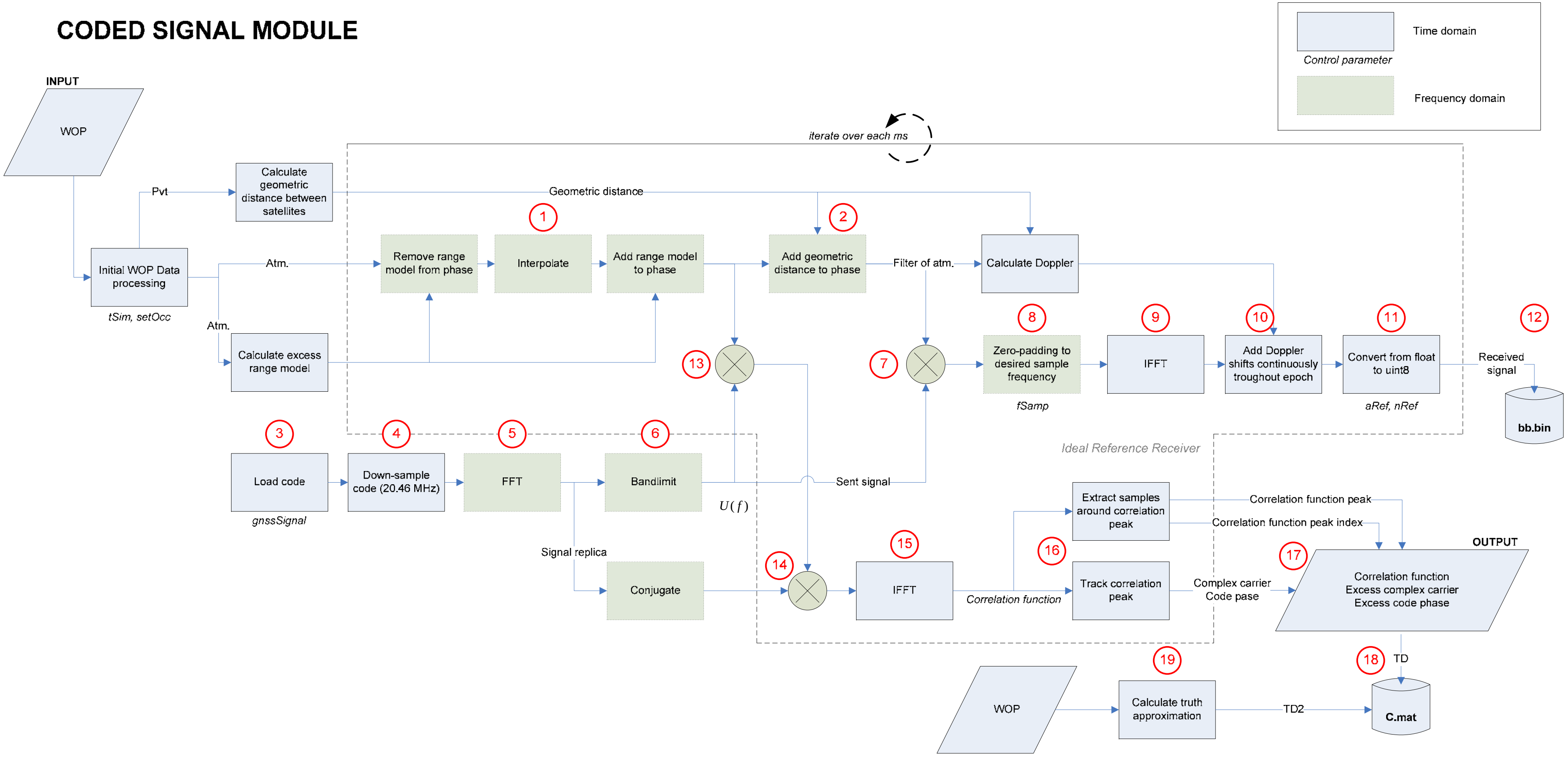


Figure 4 CSM module overview. Circled numbers refer to the list in §3.2

The centre frequency and bandwidths of the different signals supported by CSM are shown in table below. It is a pre-requisite that the frequencies provided by the WOP data covers each signal band.

Signal	Centre frequency [MHz]	Bandwidth [MHz]	Remark
GPS L1CW	1575.42	20.46 MHz	
GPS L1CA	1575.42	20.46 MHz	
GAL E1CW	1575.42	20.46 MHz	
GAL E1-CASM-B	1575.42	20.46 MHz	
GAL E1-CASM-C	1575.42	20.46 MHz	
GAL E1-CBOC-B	1575.42	20.46 MHz	
GAL E1-CBOC-C	1575.42	20.46 MHz	
GAL E5a-I	1176.45	20.46 MHz	
GAL E5a-Q	1176.45	20.46 MHz	

Table 1 Centre frequency and bandwidths for the different signals.

3.3 Ideal Receiver Module (IRM)

The IRM processes the 30 MHz signal very much as a “real” receiver, see Figure 5. The signal is down-converted and the code replicas, punctual, P, as well as early early, EE, early, E, late, L, and late late, LL, are multiplied onto the signal and integrated. The receiver operates in open loop mode, using truth data from CSM to guide the Numerically Controlled Oscillators, NCO, in the code tracking and the carrier tracking. If the input signal represents the truth data properly, the I/Q phase should be close to zero and the Early and Late correlator values should balance in amplitude. The output signal is the punctual correlator value integrated over one code epoch, with the NCO phase added to the phase. The actual length of a code epoch, i.e. the output sampling rate, will vary with geometric and atmospheric range (approximately ±23 ppm for rise/set occultations).

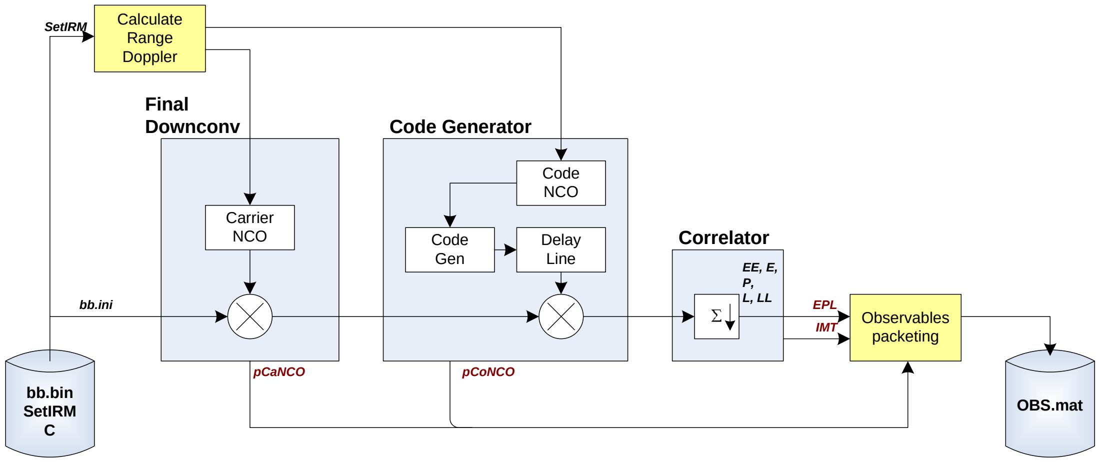


Figure 5 IRM functional block diagram.

3.4 Interface To ROPP Module (IFM)

ROPP requires as input the signal amplitude and carrier phase at two frequencies. Two CSM/IRM runs, using the same atmosphere, but simulating two signals in two different bands, shall therefore be combined into one ROPP input file. The straight line range shall be removed from the total carrier phase range as provided by IRM. An ASCII format will be used as described in §4.8.

3.5 CSM and IRM Signal Processing

3.5.1 Overview

Referring to Figure 6 we describe the occultation geometry. The provided time spacing is 1 ms and represent the signal at the received moment. This implies that the transmit time for the corresponding samples are non equidistant since the range as well as the atmospheric delay varies during the occultation.

The signal modulation is in the CSM processed in batches of the code epoch or a fractional part of the code epoch. The signal is transmitted at time intervals of 1 ms, matching a single or a fraction of a code epoch. The propagation delay is in short treated as follows:

1. The range phase is added to the signals for all frequencies and upsampled to denser frequency spacing, representing an “atmosphere filter”.
2. For every 1 ms sample, the Fourier representation of the modulated signal is multiplied with the atmosphere filter.
3. The inverse Fourier transform of the 1 ms signal package will contain the delay and stretching in time caused by the atmosphere and by the range.
4. A 1 ms section of the signal will be extracted and added to the output data sequence.
5. The signal is for every ms epoch processed in batches of epochs, including the previous and following epochs. At the receiver a 1 ms segment is cut out from the delayed and stretched received signal.

The total received Doppler shift is obtained from a linear interpolation of the phase difference between the sampled points.

The start of a code epoch is aligned with the first received sample. The following epochs are successively delayed relative to the ms intervals as illustrated in Figure 7. Sufficient overlap between the code epochs and the ms intervals is ensured by processing also the previous and following epochs.

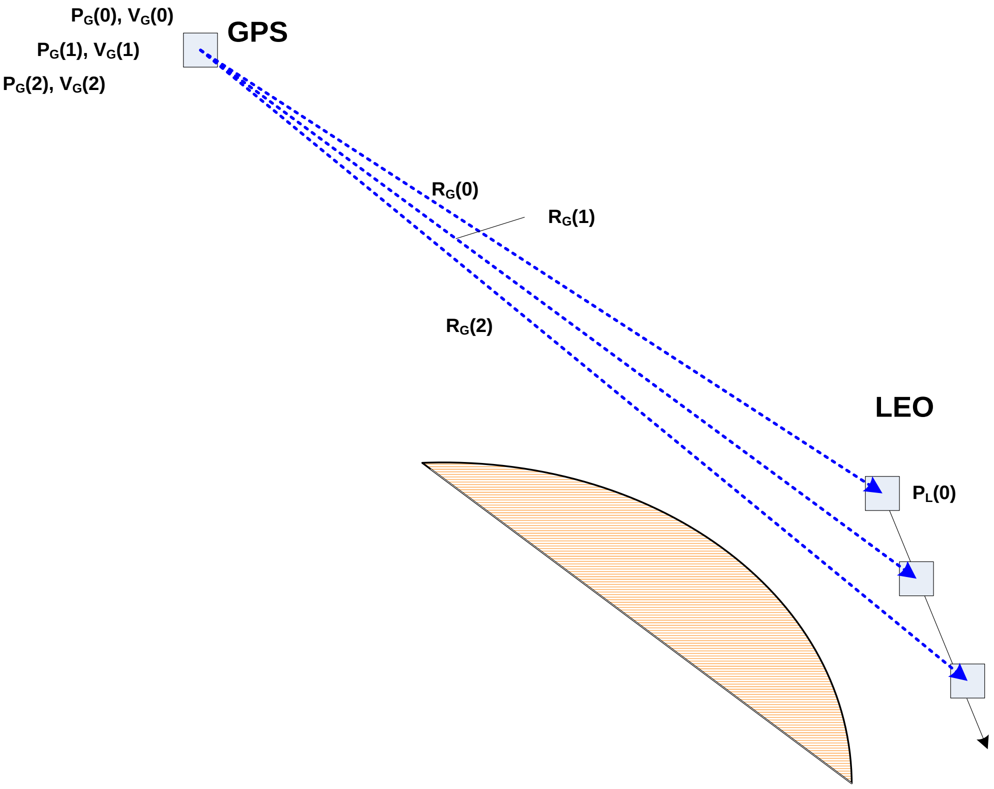


Figure 6 Occultation geometry and definition.

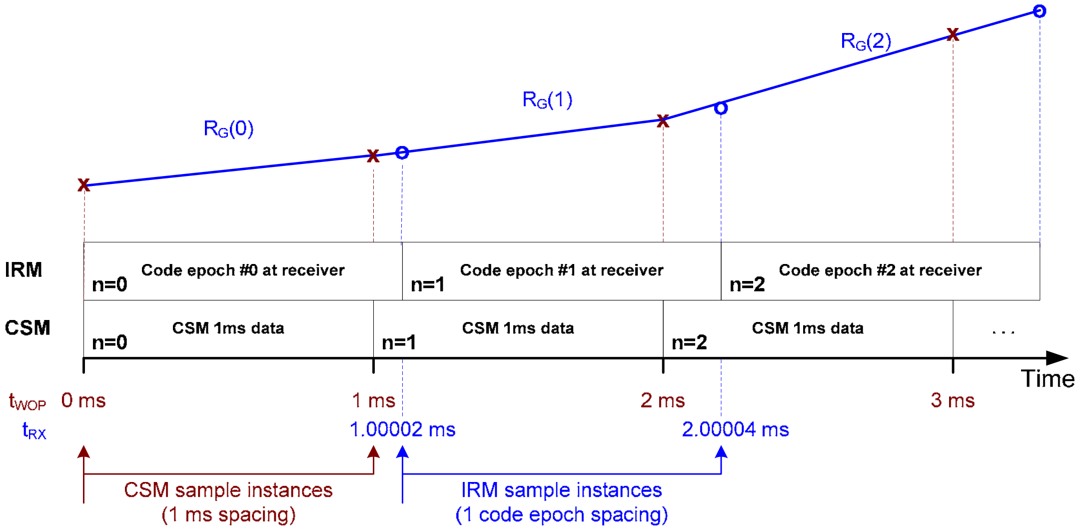


Figure 7 CSM and IRM time vectors at the receiver.

3.5.2 CSM Processing

The geometric range relative to start time, ΔR_G, is calculated according to:

$R_G = |P_G – P_L|$ GPS velocity equals 0 **Eq 1**

$\Delta R_G = R_G - R_G(0)$ **Eq 2**

The operator |*| denotes the geometric distance $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, where as an example; $\Delta x = x_{Rx} - x_{Tx}$ and so forth.

The code and carrier delay from atmosphere (also referred to excess phase) are calculated as the group delay and offset phase by the filter, where the transfer function is defined by the amplitude and phase profiles versus frequency in the input WOP data.

The frequency spread GNSS signal is referred to as *Ex(t)*. The signal at the receiver, E*y(t)*, is calculated according to:

$$Ey(t) = Ex(t - \Delta t) \cdot e^{-j2\pi f_{Doppler} \cdot t} \qquad \textbf{Eq 3}$$

$$\Delta t = \Delta R_G / c_0 + \Delta R_A / c_0 \qquad \textbf{Eq 4}$$

Where the delay Δt is the sum of the relative geometric delay and atmospheric delay.

The Doppler frequency, $f_{Doppler}$, is calculated according to:

$$f_{Doppler}(n) = \frac{V_D(n)}{c_0} f_0 + f_{OFFS} \qquad \textbf{Eq 5}$$

$$V_D(n) = \frac{R_D(n+1) - R_D(n)}{T_S} \qquad \textbf{Eq 6}$$

$R_D(n) = R_G + R_{A,Carr}(n)$ **Eq 7**

T_S is the time between samples

f_{OFFS} Offset frequency at input to ADC

The CSM calculates batches of data with 1ms length. To speed up the processing time, both the code delay and $f_{Doppler}$ are held constant during each 1 ms of generated BB data.

The 30 MHz signal is stored in single precision in unsigned 8-bit format to save storage space. In order not to lose precision, the amplitude is scaled from unity by a factor aRef, and white noise with a standard deviation nRef is added to introduce dithering to avoid introducing quantisation effects. The noise introduced by this dithering is estimated in the following.

The integrated coherent amplitude from *N* samples is calculated as (we assume the samples all have the same amplitude):

$$a = \sum_{i=1}^{N} a_i \approx N \cdot \text{aRef} \qquad \textbf{Eq 8}$$

The amplitude is decreasing from unity at the beginning of the scenario why the actual amplitude will be somewhat lower.

The integrated incoherent noise power over *N* samples is calculated as (we assume the samples all have the same std)::

$$n^2 = \sum_{i=1}^{N} n_i^2 \approx N \cdot \text{nRef}^2 \ \Rightarrow \ n \approx \sqrt{N} \cdot \text{nRef} \qquad \textbf{Eq 9}$$

and we can derive the carrier to noise ratio C/No as:

$$C/No \approx \left(\frac{a}{n}\right)^2 \cdot f_s = \left(\frac{N \cdot \text{aRef}}{\sqrt{N} \cdot \text{nRef}}\right)^2 \cdot f_s = N \cdot \left(\frac{\text{aRef}}{\text{nRef}}\right)^2 \cdot f_s \qquad \textbf{Eq 10}$$

With aRef = 70, nRef = 5, N = 30 000 samples and f_s = 1 kHz, we obtain *C/No* = 98 dB Hz, and including some margin a suitable pass criterion is 90 dBHz which is considerably higher than any live signal value.

The user may change these values to get a more realistic (lower) C/N_0 value. It is important to ensure that the sum of signal and noise is less than the maximum amplitude that can be represented by the 8-bit value, in order to avoid saturation effects. As a rule of thumb:

$$\text{aRef} + 3\cdot\text{nRef} < 127$$ **Eq 11**

3.5.3 IRM Processing

IRM processing is based on the received code epochs. The geometric range is calculated according to:

$$R_G = |P_G - P_L|$$ **Eq 12**

$$\Delta R_G = R_G - R_G(0)$$ geometric range at 1 ms spacing normalised to t=0 **Eq 13**

$$t_{RX} = t + \Delta R_G / c_0$$ Receive time for code epochs **Eq 14**

The code velocity, V, at start for each code epoch at receiver is calculated according to:

$$R(n) = R_G(n) + R_{A,Code}(n)$$ **Eq 15**

$$V(n) = \frac{R(n+1) - R(n)}{t_{RX}(n+1) - t_{RX}(n)}$$ **Eq 16**

Where n refers to the received code epoch number, $R_{A,Code}(n)$ is the thruth code delay (in meters) from CSM and referred to received code epochs. R_G is the geometric range time stamped to start of received code epoch according to Eq 14 above.

The carrier velocity, V_D, at start for each code epoch at receiver is calculated according to:

$$R_D(n) = R_G(n) + R_{A,Carr}(n)$$ **Eq 17**

$$V_D(n) = \frac{R_D(n+1) - R_D(n)}{t_{RX}(n+1) - t_{RX}(n)}$$ **Eq 18**

Where n refers to the received code epoch number, $R_{A,Carr}(n)$ is the truth carrier delay (in meters) from CSM and referred to received code epochs.

The frequency of the code NCO is calculated according to:

$$f_{CodeNCO}(n) = f_{Code} \cdot \left[1 - \frac{V(n)}{c_0}\right]$$ e.g. f_{Code} = 1.023 MHz **Eq 19**

The frequency of the receiver carrier NCO for downconversion is calculated according to:

$$f_{CarrNCO}(n) = \frac{V_D(n)}{c_0} f_0 + f_{OFFS}$$ **Eq 20**

where f_0 is the signal centre frequency, e.g. 1575.42 MHz for GPS L1 and Galileo L1, and f_{OFFS} is the offset frequency at input to ADC.

4 USER's MANUAL

4.1 Matlab Environment

The CSM is written in Matlab and requires the following Matlab resources:

- Matlab R2010a (v 7.10.0)

Above resources need to be accessible via the Matlab path before CSM can be executed. Use the Matlab "addpath" command to update the path if needed.

4.2 System Requirements

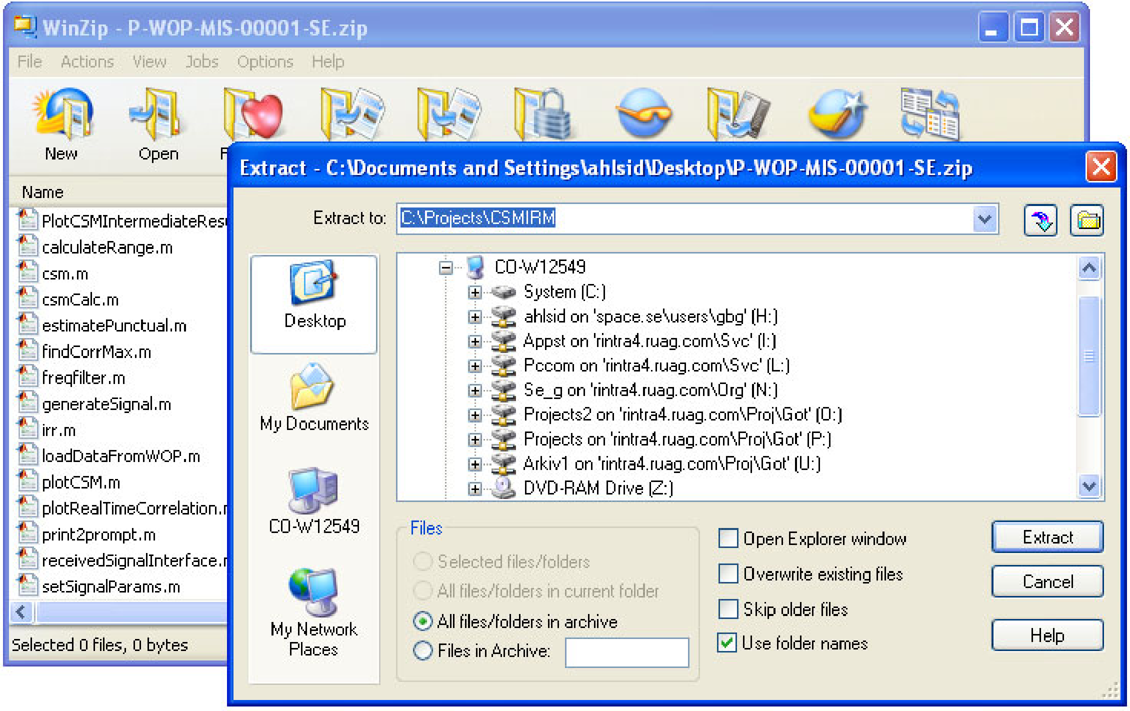
The following system requirements are recommended for a PC environment:

- Intel® Core2 Duo CPU or equivalent
- 3 GHz processor, 2 GB of RAM

4.3 Installation

Follow the instructions listed below to install the CSM/IRM:

- The CSM/IRM simulator is provided as a zip file named: P-WOP-MIS-00001-SE, *Matlab Code for CSM/IRM*
- Unzip the provided file to the desired root directory (ROOT) for CSM/IRM Example: ROOT = C:\Projects\CSMIRM\



Name
PlotCSMIntermediateResu
calculateRange.m
csm.m
csmCalc.m
estimatePunctual.m
findCorrMax.m
freqfilter.m
generateSignal.m
irr.m
loadDataFromWOP.m
plotCSM.m
plotRealTimeCorrelation.r
print2prompt.m
receivedSignalInterface.r
setSignalParams.m

Selected 0 files, 0 bytes

Extract to: C:\Projects\CSMIRM

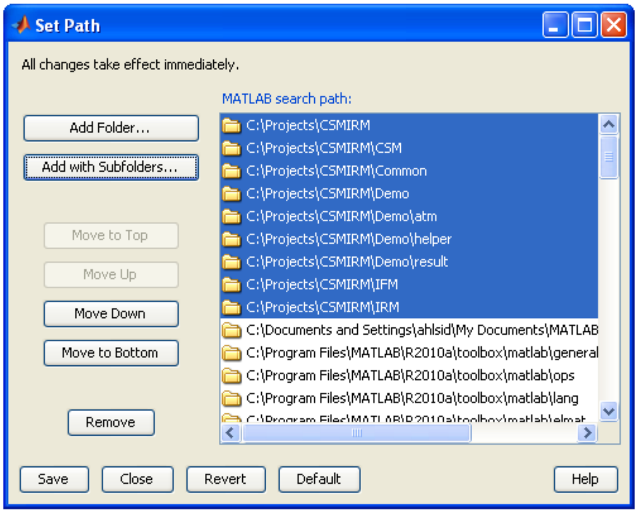
CO-W12549
System (C:)
ahlsid on 'space.se\users\gbg' (H:)
Appst on 'rintra4.ruag.com\Svc' (I:)
Pccom on 'rintra4.ruag.com\Svc' (L:)
Se_g on 'rintra4.ruag.com\Org' (N:)
Projects2 on 'rintra4.ruag.com\Proj\Got' (O:)
Projects on 'rintra4.ruag.com\Proj\Got' (P:)
Arkiv1 on 'rintra4.ruag.com\Proj\Got' (U:)
DVD-RAM Drive (Z:)

Files
○ Selected files/folders
○ All files/folders in current folder
● All files/folders in archive
○ Files in Archive:

☐ Open Explorer window
☐ Overwrite existing files
☐ Skip older files
☑ Use folder names

- Start Matlab

- Add the selected root directory path to the Matlab path using the *'setpath'* command:
  Example: ***File menu, Set Path, click button 'Add with Subfolders', choose your ROOT directory, save, close***



All changes take effect immediately.

MATLAB search path:

```
C:\Projects\CSMIRM
C:\Projects\CSMIRM\CSM
C:\Projects\CSMIRM\Common
C:\Projects\CSMIRM\Demo
C:\Projects\CSMIRM\Demo\atm
C:\Projects\CSMIRM\Demo\helper
C:\Projects\CSMIRM\Demo\result
C:\Projects\CSMIRM\IFM
C:\Projects\CSMIRM\IRM
C:\Documents and Settings\ahlsid\My Documents\MATLAB
C:\Program Files\MATLAB\R2010a\toolbox\matlab\general
C:\Program Files\MATLAB\R2010a\toolbox\matlab\ops
C:\Program Files\MATLAB\R2010a\toolbox\matlab\lang
```

- The CSM/IRM simulator is ready to run. To process a demo scenario, navigate to the Demo folder in your ROOT directory and type CSMIRMdemo into the Matlab terminal window. Two single-frequency simulations will be run through the CSM/IRM simulation chain and the results will be combined into a single textfile by IFM. The demo scenario is further explained in Chapter 4.4.

4.4 Demo Scenario

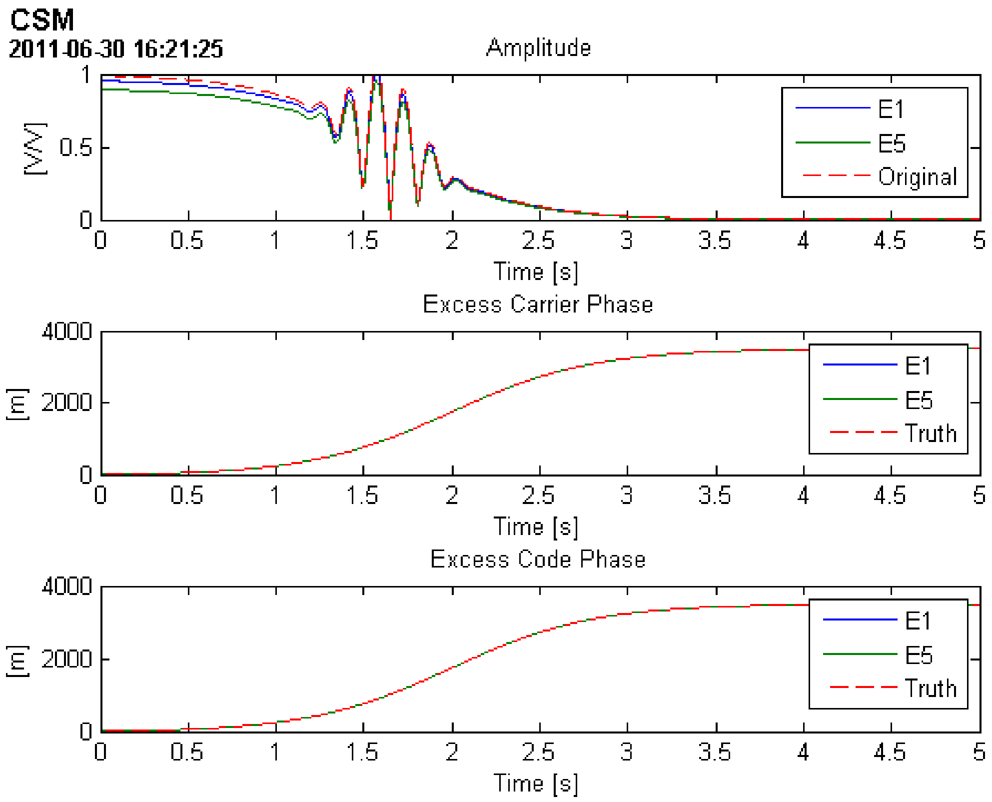
The demo scenario is named *CSMIRMdemo.m* and is located in the Demo\ subfolder.

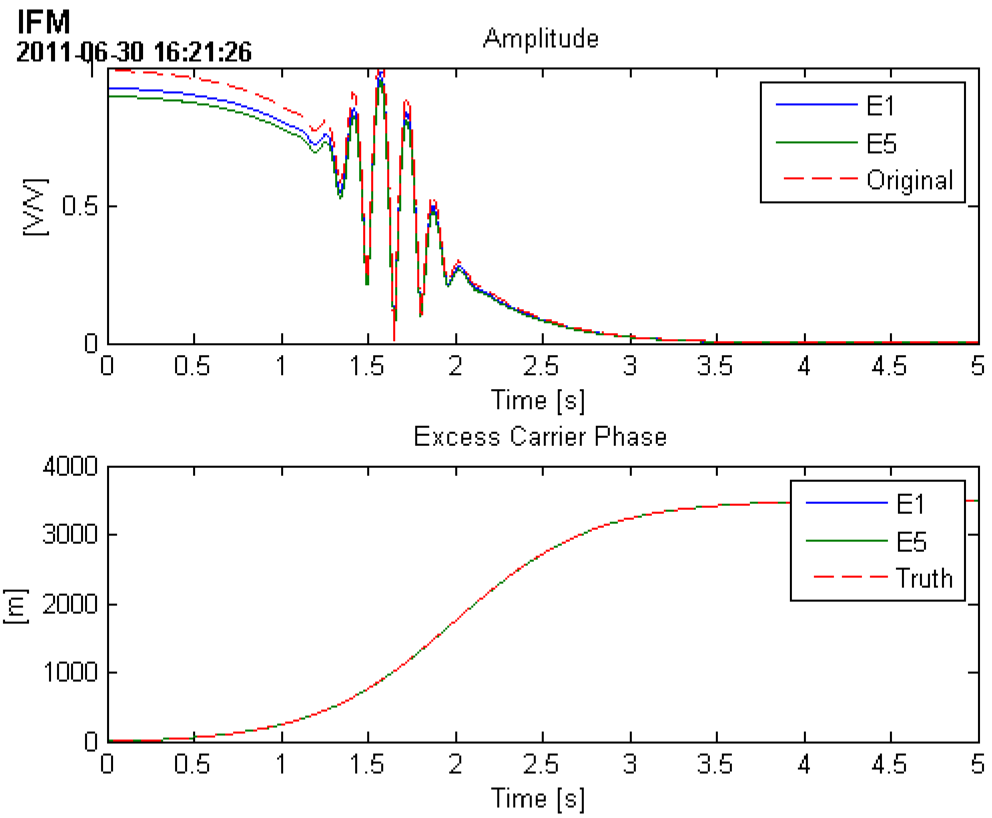
In the initial phase of the demo scenario CSM produces a Galileo E1CBOC-C signal as it propagates through a mathematically defined atmosphere located in the atm\ folder. The atmosphere is not a realistic profile, but includes severe atmosphere qualities like multipath, high dynamics, and large range variations.

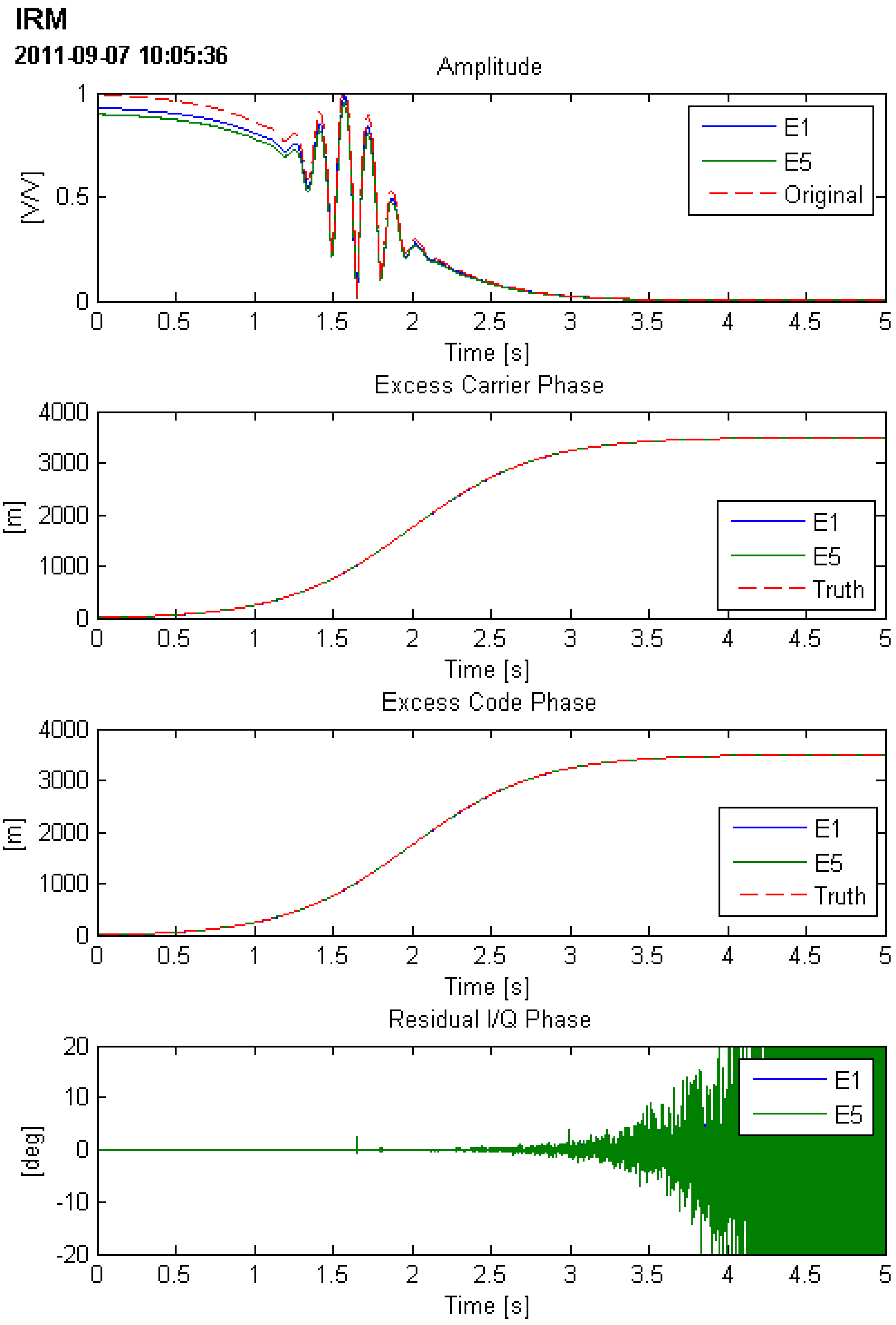
A Matlab figure shows the progress. When the signal has been produced IRM tracks it. The procedure is then repeated for Galileo E5a-I.

After the CSM/IRM processing has been performed for the two frequencies and the results have been combined into a text file by IFM a number of resulting figures will be plotted on the screen, presenting the amplitude and code and carrier phase for CSM, amplitude, and code, carrier and residual I/Q phase for IRM, as well as amplitude and carrier phase for IFM. The true code and carrier phases can be mathematically derived from the demo atmosphere, and the results have been plotted in the figures. We can note the drop in amplitude from the original amplitude, which is due to the effect band-limiting the signal has on signal energy.

The figures that are plotted are presented on the following pages.


CSM
2011-06-30 16:21:25
Amplitude
[V/V]
1
0.5
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time [s]
E1
E5
Original
Excess Carrier Phase
[m]
4000
2000
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time [s]
E1
E5
Truth
Excess Code Phase
[m]
4000
2000
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time [s]
E1
E5
Truth


IFM
2011-06-30 16:21:26
Amplitude
[V/V]
1
0.5
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time [s]
E1
E5
Original
Excess Carrier Phase
[m]
4000
3000
2000
1000
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time [s]
E1
E5
Truth


IRM
2011-09-07 10:05:36
Amplitude
[V/V]
1
0.5
0
E1
E5
Original
Time [s]
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Excess Carrier Phase
[m]
4000
3000
2000
1000
0
E1
E5
Truth
Time [s]
Excess Code Phase
[m]
4000
3000
2000
1000
0
E1
E5
Truth
Time [s]
Residual I/Q Phase
[deg]
20
10
0
-10
-20
E1
E5
Time [s]

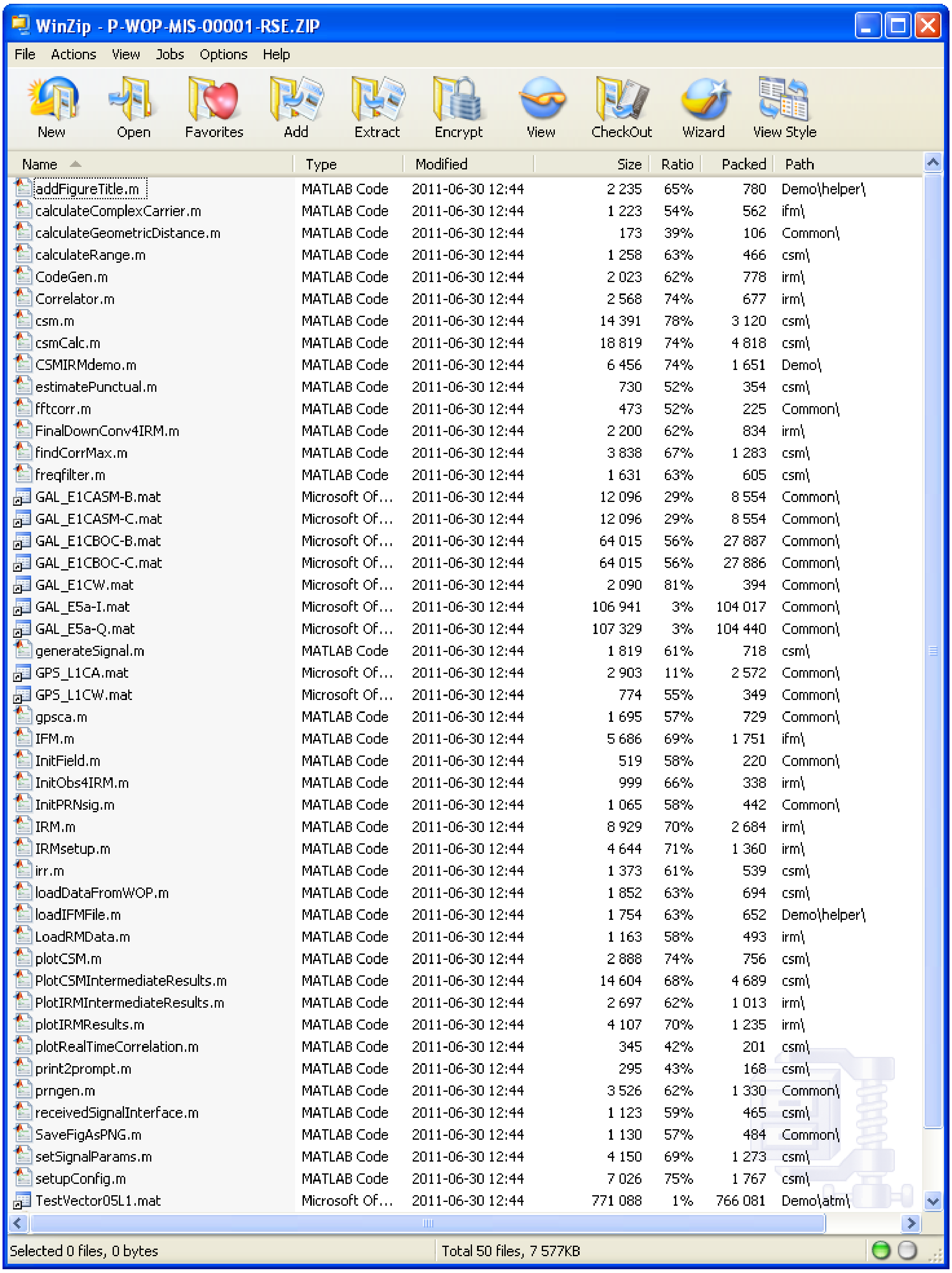
4.5 File Structure

The zip-file extracts to five subfolders: Common\, CSM\, Demo\, IRM\ and IFM\. The Common library contains helper functions related to all modules of the CSM/IRM simulator. CSM, IRM and IFM contain the respective modules, while Demo contains an example for an easy starting point.

The file structure is as follows:

- Common\
    - calculateGeometricDistance.m
    - fftcorr.m
    - GAL_E1CASM-B
    - GAL_E1CASM-C
    - GAL_E1CBOC-B
    - GAL_E1CBOC-C
    - GAL_E1CW
    - GAL_E5a-I
    - GAL_E5a-Q
    - GPS_L1CA
    - GPS_L1CW
    - Gpsca.m
    - InitField.m
    - InitPRNsig.m
    - prngen.m
    - SaveFigAsPNG.m
- CSM\
    - calculateRange.m
    - csm.m
    - csmCalc.m
    - estimatePunctual.m
    - findCorrMax.m
    - freqfilter.m
    - generateSignal.m
    - irr.m
    - loadDataFromWOP.m
    - plotCSM.m
    - PlotCSMIntermediateResults.m
    - plotRealTimeCorrelation.m
    - print2prompt.m
    - setSignalParams.m
    - setupConfig.m
- Demo\
    - atm\
        - TestVector05L1
        - TestVector05L1.txt
        - TestVector05L5
        - TestVector05L5.txt
    - helper\
        - addFigureTitle.m

        - loadIFMFile.m
    - CSMIRMdemo.m
- IFM\
    - calculateComplexCarrier.m
    - IFM.m
- IRM\
    - CodeGen.m
    - Correlator.m
    - FinalDownConv4IRM.m
    - InitObs4IRM.m
    - IRM.m
    - IRMsetup.m
    - LoadRMData.m
    - PlotIRMIntermediateResults.m
    - plotIRMResults.m



Name	Type	Modified	Size	Ratio	Packed	Path
addFigureTitle.m	MATLAB Code	2011-06-30 12:44	2 235	65%	780	Demo\helper\
calculateComplexCarrier.m	MATLAB Code	2011-06-30 12:44	1 223	54%	562	ifm\
calculateGeometricDistance.m	MATLAB Code	2011-06-30 12:44	173	39%	106	Common\
calculateRange.m	MATLAB Code	2011-06-30 12:44	1 258	63%	466	csm\
CodeGen.m	MATLAB Code	2011-06-30 12:44	2 023	62%	778	irm\
Correlator.m	MATLAB Code	2011-06-30 12:44	2 568	74%	677	irm\
csm.m	MATLAB Code	2011-06-30 12:44	14 391	78%	3 120	csm\
csmCalc.m	MATLAB Code	2011-06-30 12:44	18 819	74%	4 818	csm\
CSMIRMdemo.m	MATLAB Code	2011-06-30 12:44	6 456	74%	1 651	Demo\
estimatePunctual.m	MATLAB Code	2011-06-30 12:44	730	52%	354	csm\
fftcorr.m	MATLAB Code	2011-06-30 12:44	473	52%	225	Common\
FinalDownConv4IRM.m	MATLAB Code	2011-06-30 12:44	2 200	62%	834	irm\
findCorrMax.m	MATLAB Code	2011-06-30 12:44	3 838	67%	1 283	csm\
freqfilter.m	MATLAB Code	2011-06-30 12:44	1 631	63%	605	csm\
GAL_E1CASM-B.mat	Microsoft Of...	2011-06-30 12:44	12 096	29%	8 554	Common\
GAL_E1CASM-C.mat	Microsoft Of...	2011-06-30 12:44	12 096	29%	8 554	Common\
GAL_E1CBOC-B.mat	Microsoft Of...	2011-06-30 12:44	64 015	56%	27 887	Common\
GAL_E1CBOC-C.mat	Microsoft Of...	2011-06-30 12:44	64 015	56%	27 886	Common\
GAL_E1CW.mat	Microsoft Of...	2011-06-30 12:44	2 090	81%	394	Common\
GAL_E5a-I.mat	Microsoft Of...	2011-06-30 12:44	106 941	3%	104 017	Common\
GAL_E5a-Q.mat	Microsoft Of...	2011-06-30 12:44	107 329	3%	104 440	Common\
generateSignal.m	MATLAB Code	2011-06-30 12:44	1 819	61%	718	csm\
GPS_L1CA.mat	Microsoft Of...	2011-06-30 12:44	2 903	11%	2 572	Common\
GPS_L1CW.mat	Microsoft Of...	2011-06-30 12:44	774	55%	349	Common\
gpsca.m	MATLAB Code	2011-06-30 12:44	1 695	57%	729	Common\
IFM.m	MATLAB Code	2011-06-30 12:44	5 686	69%	1 751	ifm\
InitField.m	MATLAB Code	2011-06-30 12:44	519	58%	220	Common\
InitObs4IRM.m	MATLAB Code	2011-06-30 12:44	999	66%	338	irm\
InitPRNsig.m	MATLAB Code	2011-06-30 12:44	1 065	58%	442	Common\
IRM.m	MATLAB Code	2011-06-30 12:44	8 929	70%	2 684	irm\
IRMsetup.m	MATLAB Code	2011-06-30 12:44	4 644	71%	1 360	irm\
irr.m	MATLAB Code	2011-06-30 12:44	1 373	61%	539	csm\
loadDataFromWOP.m	MATLAB Code	2011-06-30 12:44	1 852	63%	694	csm\
loadIFMFile.m	MATLAB Code	2011-06-30 12:44	1 754	63%	652	Demo\helper\
LoadRMData.m	MATLAB Code	2011-06-30 12:44	1 163	58%	493	irm\
plotCSM.m	MATLAB Code	2011-06-30 12:44	2 888	74%	756	csm\
PlotCSMIntermediateResults.m	MATLAB Code	2011-06-30 12:44	14 604	68%	4 689	csm\
PlotIRMIntermediateResults.m	MATLAB Code	2011-06-30 12:44	2 697	62%	1 013	irm\
plotIRMResults.m	MATLAB Code	2011-06-30 12:44	4 107	70%	1 235	irm\
plotRealTimeCorrelation.m	MATLAB Code	2011-06-30 12:44	345	42%	201	csm\
print2prompt.m	MATLAB Code	2011-06-30 12:44	295	43%	168	csm\
prngen.m	MATLAB Code	2011-06-30 12:44	3 526	62%	1 330	Common\
receivedSignalInterface.m	MATLAB Code	2011-06-30 12:44	1 123	59%	465	csm\
SaveFigAsPNG.m	MATLAB Code	2011-06-30 12:44	1 130	57%	484	Common\
setSignalParams.m	MATLAB Code	2011-06-30 12:44	4 150	69%	1 273	csm\
setupConfig.m	MATLAB Code	2011-06-30 12:44	7 026	75%	1 767	csm\
TestVector05L1.mat	Microsoft Of...	2011-06-30 12:44	771 088	1%	766 081	Demo\atm\

Selected 0 files, 0 bytes

Total 50 files, 7 577KB

Figure 8 File organisation of the CSM/IRM zip file. Part of the files shown.

4.6 Setup Files

4.6.1 CSM Setup File

Field in 'SetCSM' struct	Size	VAL	Default Value	Description
tSim	1x1	float >0	Length of WOP file	Simulation time. Unit: second
inFile		string	NA	Path to WOP file to run. No default value.
outPath		string	NA	Path to output data to be saved. Output folder is created if not exists. If folder exists, existing data will be overwritten. No default value.
gnssSignal		string	GAL_E5a-I	Used GNSS signal. Supported modulated signals are: 'GPS_L1CA', 'GAL_E5a-I', 'GAL_E5a-Q', 'GAL_E1CBOC-B', 'GAL_E1CBOC-C', 'GAL_E1CASM-B', 'GAL_E1CASM-C'. Supported non-modulated signal is: 'GPS_L1CW' PRN number is =1.
setOcc	1x1	'set' or 'rise'	'set'	Defines the simulation to be a setting occultation (VAL='set') or rising occultation (VAL='rise'). It is a pre-requisite that input WOP data is a setting occultation.
generateBBdata	1x1	Boolean	TRUE	Enables (VAL=TRUE) or disables (VAL=FALSE) the generation of BB data.
fSamp	1x1	float	30e6	Sample frequency of the output baseband signal and it must be an integer multiple of 1 kHz. Unit: Hz This parameter is NA if generateBBdata is set to FALSE.
navData	1x1 1xN	integer (-1, +1)	[ ] (empty)	Navigation (and secondary code modulation) data bits. If input vector is empty ([ ]), then the navData stream is a random sequence of -1 and +1. If VAL = +1 (or -1) then constant data will be applied to all epochs. If an input vector is specified, length in time shall match tSim. The navigation data rate is 50 Hz for GPS L1 C/A, 250 Hz for GAL E1, and 1 kHz otherwise. This parameter is NA if generateBBdata is set to FALSE.
aRef	1x1	float	70	Reference BB signal amplitude and high altitude, see §3.5.2 and §4.8.4. This parameter is NA if generateBBdata is set to FALSE.
nRef	1x1	float	5	BB signal RMS noise, see §3.5.2 and §4.8.4. This parameter is NA if generateBBdata is set to FALSE.
saveFigures	1x1	Boolean	FALSE	Saves figures to file (VAL=TRUE) or not (VAL=FALSE).
DEBUG				See Table 3 below.

Table 2 Fields in input variable 'SetCSM' to CSM.

In addition to the fields defined in Table 2 above, it is possible to set the CSM in debug mode by using the fields defined in Table 3 below.

Field in SetCSM.DEBUG struct	Size	VAL	Default Value	Description
stationary	1x1	Boolean	0	Stationary satellite geometry. If VAL = TRUE (1) then is the GPS and LEO satellite positions set to their first value (at time = 0) and velocity vectors set to 0.
excessRangeOn	1x1	Boolean	1	Atmospheric range Is set to zeroes if VAL = FALSE (0).
ampOn	1x1	Boolean	1	Atmospheric amplitude modulation is set to constant one if VAL = FALSE (0).
saveFormat	1x1	string	'uint8'	Saved format of the baseband data signal. VAL can be 'uint8' (1 byte per sample) or 'double'
tSkip	1x1	float	0	Time to be skipped at beginning of an occultation, to be used when the signal from WOP has not stabilised at the beginning of the occultation.

Table 3 CSM debug fields.

```
setCSM.gnssSignal            = 'GPS_L1CA';
setCSM.inFile                = 'P:\P-WOP\Matlab\TestCases\TestCase03.txt';
setCSM.outPath               = 'C:\Projects\ROSIM\Works\CSM\'
setCSM.tSim                  = 4;           % Simulation time, seconds
setCSM.setOcc                = 'setting';   % Occultation direction
setCSM.navData               = 1;           % Set navigation data stream
setCSM.fSamp                 = 30e6;        % Sample frequency of bb.bin
setCSM.generateBBdata        = 1;           % Boolean -- to save or not to save bb.bin
setCSM.saveFigures           = 1;           % Boolean -- to save or not to save figures
CSM( SetCSM );
```

Table 4 Example of initialisation of 'SetCSM' struct.

4.6.2 IRM Setup File

Field in 'SetRM' struct	Size	VAL	Default Value	Description
tSim	1x1	float >0	length of input file	Simulation time in seconds. If VAL exceeds the length of input data file (*bb.bin*), then simulation time is decreased to match input file length. Unit: second
gnssSignal		string	As def. in C.CON. gnssSignal	Used GNSS signal. Supported modulated signals are: 'GPS_L1CA', 'GAL_E5a-I', 'GAL_E5a-Q', 'GAL_E1CBOC-B', 'GAL_E1CBOC-C', 'GAL_E1CASM-B', 'GAL_E1CASM-C'. Supported non-modulated signal is: 'GPS_L1CW' PRN number is =1.
inPath		string	NA	Path to BB file (*bb.bin*). No default value.
outPath		string	NA	Path to output data to be saved. Output folder is created if not exists. If folder exists, existing data will be overwritten. No default value.
tplotUpd	1x1	float >4e-3	0.1	Update interval of intermediate plot results on screen. A lower value increases the simulation time. Unit: second

Table 5 Fields in input setup variable 'SetRM' to IRM.

```
SetRM.tSim     = 4;                                  % Set simulation length (in sec)
SetRM.inPath   = 'C:\Projects\ROSIM\Works\CSM\K05'; % Path to CSM input files
SetRM.outPath  = 'C:\Projects\ROSIM\Works\IRM';     % Path to IRM output files
IRM(SetRM );                                         % Run IRM
```

Table 6 Example of initialisation of 'SetRM' struct.

4.7 WOP Interface Description

4.7.1 General

The WOP output data file to be used by CSM will contain amplitude and excess phase range at all frequencies, the position and velocity, the time, the frequencies and identification parameters such as; occultation identification number, the occultation date and time and the date and time the WOP simulation is run.

4.7.2 Data Format

The file shall be of ASCII type with the general parameters written in lines starting with %%,, and the bulk of vector data are written in columns as a matrix. The lines are defined as follows:

Line	Parameter	Remark
1	%% ID DATE(yyyy mm dd) TIME(hh mm ss)	Identification number and date and time for the occultation that is analysed
2	%% Values for the parameters of line #1	
3	%% RUNDATE(yyyy mm dd) RUNTIME(hh mm ss)	Date and time the analysis is performed
4	%% Values for the parameters of line #3	
5	%% Modulation frequencies [Hz]	Sequence of frequencies analysed, shall cover the following ranges: TBD
6	%% f1 f2 f3 … fN	See Note1
7	%% List of vectors included, see	
8	1:st value of all vectors of line #5	
9	2:nd value of all vectors of line #5	
10	3:rd value of all vectors of line #5	
•	•	
•	•	
nn	Last value of all vectors of line #5	

Table 7 Content of WOP output data file to be used by CSM.

Note 1: Within each frequency band, 5 frequencies are recommended, spread over the modulation frequency band, e.g. [-15,-7.5,0,7.5,15] MHz offset from the signal centre frequency. Since the observed variation with frequency is continuous and in order to mitigate the short periods of significant numerical noise appearing in some WOP occultation data, we interpolate to more frequencies using a quadratic polynomial matched to the complex amplitude at the provided frequencies. In principle only 3 frequencies per band would hence be required, but 5 frequencies is more robust and recommended, since some of the numerical noise will be averaged. ±15 MHz calculation bandwidth is recommended even if the signal bandwidth is small. This in order to have a robust estimate of the phase slope, which is proportional to the code delay.

The list of vectors in line #5 shall consist of the following parameters:

Vector	Meaning
T [s]	Time stamp of the received signal, shall be in 1 kHz rate, see Note 1
PRX_X [km]	X-position of receiver at time T
PRX_Y [km]	Y-position of receiver at time T
PRX_Z [km]	Z-position of receiver at time T
VRX_X [km/s]	X-velocity of receiver at time T
VRX_Y [km/s]	Y-velocity of receiver at time T
VRX_Z [km/s]	Z-velocity of receiver at time T
PTX_X [km]	X-position of transmitter at time T
PTX_Y [km]	Y-position of transmitter at time T
PTX_Z [km]	Z-position of transmitter at time T
VTX_X [km/s]	X-velocity of transmitter at time T, shall be negligible.
VTX_Y [km/s]	Y-velocity of transmitter at time T, shall be negligible.
VTX_Z [km/s]	Z-velocity of transmitter at time T, shall be negligible.
AMP_f1 [V/V]	Normalised amplitude for the received frequency 1 at time T.
PHASE_f1 [m]	Excess phase range for the received frequency 1 at time T. Straight line phase is subtracted.
AMP_f2 [V/V]	Normalised amplitude for the received frequency 2 at time T.
PHASE_f2 [m]	Excess phase range for the received frequency 2 at time T. Straight line phase is subtracted.
AMP_f3 [V/V]	Normalised amplitude for the received frequency 3 at time T.
PHASE_f3 [m]	Excess phase range for the received frequency 3 at time T. Straight line phase is subtracted.
•	•
•	•
AMP_fN [V/V]	Normalised amplitude for the received frequency N at time T.
PHASE_fN [m]	Excess phase range for the received frequency N at time T. Straight line phase is subtracted.

Table 8 List of vectors used in the WOP data.

Note 1: The CSM module is built on the fact that the WOP data are synchronous with the code epochs. The time vector is not used, but replaced with a time vector sampled at exactly 1 kHz.

The amplitude unit marked as V, is a voltage normalised to unity at the beginning of the occultation.

4.7.3 Example

An example of a data file is shown below. Only the beginning of long rows is shown.

```
%% ID DATE(yyyy mm dd) TIME(mm ss)
%%   681 1997 02 02 23 42  0.240000E+02
%% RUNDATE(yyyy mm dd) RUNTIME(hh mm ss)
%% 2011 03 24 14 52 39
%% Modulation frequencies (Hz)
%% 1.560420e+009 1.565420e+009 1.570420e+009 1.222600e+009 ...
%%  T(s) PR_X(km) PR_Y(km) PR_Z(km) VR_X(km/s) VR_Y(km/s) VR_Z(km/s) .
  0.0000000000000e+000   6.3764344120000e+003  -1.7727122467240e+003
  1.0000000000000e-003   6.3764508261960e+003  -1.7726754431690e+003
  2.0000000000000e-003   6.3764672401550e+003  -1.7726386395580e+003
```

4.7.4 Data Interpretation

It is essential that the WOP data are generated with a time vector sampled at one kHz to almost the numerical accuracy. This is due to the fact that the CSM architecture is built around the principle that the code epochs are aligned with the sampled atmosphere.

The WOP data is for each time generated with the GNSS and LEO positions fixed. This means that the positions are valid for the provided receive time, but also that the GNSS position is valid for the instant of transmission, and the receive time is valid for the instant of reception. The Coded Signal Module is adapted to this circumstance.

In short, the WOP data is generated as follows, see Figure 9:

1. The GNSS transmitter is stationary.
2. The atmosphere is stationary
3. The LEO satellite is probing the signal at positions in equidistant time interval along the orbit.
4. At every LEO location the amplitude and excess phase is sampled as a stationary case.
5. The phase of each frequency is set to zero at the beginning of the (setting) occultation. This is in accordance with a measurement above the atmosphere and ionosphere where the excess phase in theory vanishes.

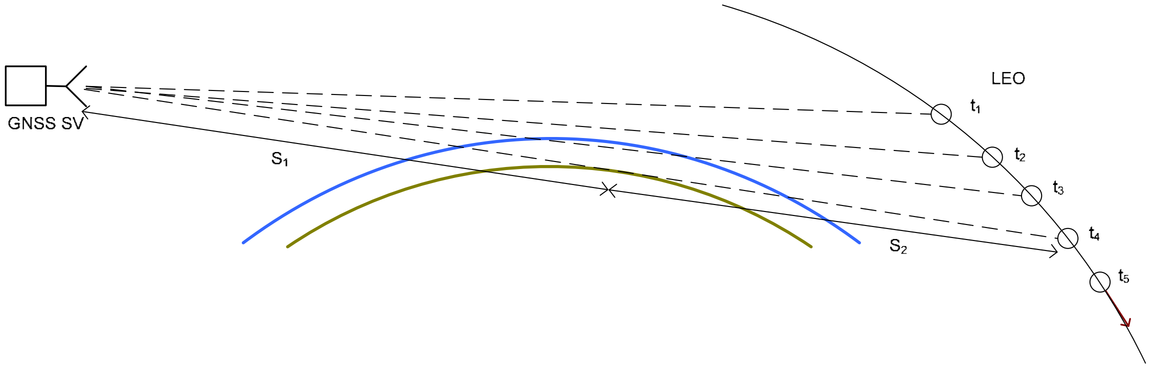


Figure 9 Geometry of WOP atmospheric propagation.

The total received Doppler shift is obtained from the straight line variation and from the phase difference between the sampled points.

In a live, dynamic, measurement, the signal will propagate from the GNSS SV, pass through the atmosphere and being received by the receiver on board the LEO satellite. Since the travel times are not negligible, the GNSS will be in a previous (retarded) position when the signal is transmitted compared to the time of reception. The retarded position has, however, no impact since the GNSS SV is stationary for the WOP data.

4.8 CSM Output Data Files

4.8.1 General

There are two CSM modes; Coded signal mode providing baseband data can be switched on or off to save processing time, but the Correlator function mode, providing the correlation function and its peak value, is always processed, see §3.2. The modes will provide separate data files, but several parameters are common and these data will be produced in all cases. We can hence define the following two data sets.

1. Common data; containing e.g. identifiers, GNSS and LEO PVT (Position, Velocity and Time), code and carrier range. These data are always provided and saved to disk as a Matlab structure. Also control parameters required to run the CSM are saved here.
2. Coded signal data; containing the high data rate baseband signal stored on disk in binary form.

4.8.2 File Naming Convention

Output data will be stored in files with predefined names and stored in separate directories defining the run cases. The file names will be:

C.mat	for the common data Matlab file
bb.bin	for the coded signal data binary file.

4.8.3 Common Data

The common data is stored in a Matlab file on disk; (*C.mat*) The variable *C* is a Matlab structure array and consists of a number of sub-structures. It is described in detail in the tables below.

Field in C	Size*	Unit	Description
Control Parameters			
CON			This field is identical to SetCSM, see §4.6.1,apart from the additional field “rundate” which includes the time and date of completion for the CSM run.
WOP Data			
WOP.ID	-	string	Identifier as provided in the WOP input data
WOP.date	-	string	
WOP.fs	-	Hz	Sampling rate in original WOP data.
GNSS Tx Signal			
TX.fc	-	Hz	Signal centre frequency
TX.NAV	1xM	-	Navigation data stream provided for each code epoch.
Occultation Geometry			
GNSS.t	1xN	s	The time vector for Geometry data is down sampled to 1 ms steps, starting with the first WOP time.
GNSS.Px	1xN	m	GNSS x-position
GNSS.Py	1xN	m	GNSS y-position
GNSS.Pz	1xN	m	GNSS z-position
GNSS.Vx	1xN	m/s	GNSS x-velocity
GNSS.Vy	1xN	m/s	GNSS y-velocity
GNSS.Vz	1xN	m/s	GNSS z-velocity
LEO.Px	1xN	m	LEO x-position at GNNS.t
LEO.Py	1xN	m	LEO y-position at GNNS.t
LEO.Pz	1xN	m	LEO z-position at GNNS.t
LEO.Vx	1xN	m/s	LEO x-velocity at GNNS.t
LEO.Vy	1xN	m/s	LEO y- velocity at GNNS.t
LEO.Vz	1xN	m/s	LEO z- velocity at GNNS.t
Truth Data for Amplitude, Carrier and Code			
TD.t	1xM	s	Time vector.
TD.A	1xM	V	Detected carrier excess amplitude, normalised to unity at first sample in setting occultation, last sample in rising occultation.
TD.pCaE	1xM	m	Detected excess carrier range of signal.
TD.pCoE	1xM	m	Detected excess code range of signal.
TD2.t	1xM	m	TD2 is truth data calculated directly from input WOP frequency and amplitude data. Fields identical to TD above.
TD2.A	1xM	m	
TD2.pCaE	1xM	m	
TD2.pCoE	1xM	m	
Correlator Data			
CORR.peakA	KxM	V	Complex correlation function. The vector is ±2µs around amplitude peak value for GPS L1CA.
CORR.peakRange	1xM	m	Range for mid value of vector CORR.peakA.

Field in C	Size*	Unit	Description
CORR.dRange	-	m/sample	Range between each sample in CORR.peakA.
CSM Debug setup			
DEBUG			This field is identical to SetCSM, see Table 3, and is located as a field in the CON substructure.
M = Number of Truth and Correlator Data samples for the entire occultation in code epochs. *N = Number of geometry samples for the entire occultation in 1 Hz* *K = width of correlation function. K is an odd number.*			

Table 9 Content of output '*C*' structure from CSM.

4.8.4 Baseband Data

The baseband data is stored as 8-bit unsigned binary format on file (bb.bin). Every odd order byte in the file is the in-phase component (i_n) and every even order byte is the quadrature component (q_n) according to:

$i_0\ q_0\ i_1\ q_1\ i_2\ q_2$ … etc

The correct offset value of the I and Q values are obtained by performing the following operations:

$I = i - 128;$

$Q = q - 128;$

The reference amplitude, aRef see §3.5.2, of the GNSS signal is defined at the first code epoch (typically 1ms) in the beginning of a setting occultation or at the last code epoch for a rising occultation. aRef is default set to 80 LSB. The coded signal data also contains noise, nRef, with a default RMS value of 7 $\mathrm{LSB}_{\mathrm{RMS}}$. Hence, the input signal to noise ratio (SNR) and carrier to noise density ratio (C/N_0) are, see §3.5.2:

$$\mathrm{SNR} = 20\cdot\log_{10}(\ 80/7\ ) \approx 21.2\ \mathrm{dB}$$

$$\mathrm{C/No} = \mathrm{SNR} + 10\cdot\log_{10}(\ fs\ ) \approx 96.0\ \mathrm{dBHz}$$

Where fs is the sample rate of the coded signal (ADC data) and here assumed to be 30 MHz.

4.9 IRM Output Data Files

The IRM produces output the output file ‘Obs.mat’ as defined in table below.

Field in OBS	Size*	Unit	Description
General Information			
CONCSM			This field is identical to SetCSM, see §4.6.1.
CONRM			This field is identical to ‘SetRM’ struct in §4.6.2
WOP			This field is identical to WOP Data in §4.8.3.
GNSS			This field is identical to Occultation Geometry in §4.8.3.
LEO			This field is identical to Occultation Geometry in §4.8.3.
Code Data			
imtCa	Nx1	1/fSamp	Instrument measurement time for carrier phase data. imtCa is offset +1 code epoch relative to the time given in GNSS.t.
imtCo	Nx1	1/Samp	Instrument measurement time for code phase data. imtCa is offset +1 code epoch relative to the time given in GNSS.t.
iEPL	Nx5	LSB	Real part of correlator values [Ie2 Ie Ip Il Il2].
qEPL	Nx5	LSB	Imaginary part of correlator values [Qe2 Qe Qp Ql Ql2].
pCoNCO	Nx1	meters	Code phase NCO observable time stamped with imtCo
pCaNCO	Nx1	meters	Carrier phase NCO observable time stamped with imtCa
navData	Nx1	-	Navigation data stream for each code epoch.
fc	1x1	Hz	GNSS signal centre frequency.
N is the number of code epochs.			

Table 10 Content of output ’OBS’ structure from IRM.

The total carrier phase is the sum of the NCO phase and the phase of the punctual I/Q correlator value:

$$pCa = pCaNCO - \frac{1}{k} angle(iP + j \cdot qP) \qquad [m]$$

where

k is the propagation constant $2\pi/\lambda$

iP,qP are the in phase and quadrature phase correlator values.

The negative sign is due to the conversion from carrier phase in radians to carrier range in meter.

Note that imtCa timestamps the pCaNCO at the end of the received epoch. imtCo and pCoNCO work correspondingly. iEPL and qEPL are valid at the middle of the code epoch. I.e., let t(n) be a time vector calculated from the IMT. The following table shows the validity of the IRM output samples.

Measurement	Timestamp
pCaNCO	t(n)
pCoNCO	t(n)
iEPL, qEPL	$t(n) – T_{epoch}/2$

Table 11 Timestamp of IRM output data.

4.10 IFM Output Data Files

The ROPP Interface Module, IFM, processes the OBS structures produced by two IRM runs and combines them into a single text file. IFM also has a second mode which can process the C structures produced by CSM runs, and will then combine the truth data in the C.TD structure.

The IFM Matlab function has two mandatory input arguments, and one optional input argument. The first two arguments are structures which have been produced by CSM or IRM (‘C’ or ‘OBS’). The third parameter is a path to a target file. The default file name is called Mod2Freq.txt which will be created in the same folder as the output directory of the first structure (i.e. OBS.CONRM.outPath). IFM returns the path to the created file.

```
IFM(OBS1, OBS2, ‘C:\Projects\Simulations\Mod2Freq.txt’);    % Run IFM
```

Table 12 Example of IFM run. Here it is assumed that two OBS structures according to Table 10 exist in the Matlab workspace.

IRM generates the total phase, while the input to ROPP shall be the excess phase. The geometrical range is therefore for every sample subtracted from the phase range according to:

$$R_G = |P_G - P_L|$$

$$R_{A,Carr} = R_D - R_G$$

where

P_G	is the position of the GNSS transmitter
P_L	is the position of the LEO receiver
R_G	is range between Tx and Rx
$R_{A,Carr}$	is the atmosphere of excess carrier range
R_D	is the total carrier range

The text file structure is shown in Table 13.

Line	Parameter	Description
1	%% ID WOPDATE(yyyy mm dd) WOPTIME(hh mm ss)	Identification for the atmosphere analyzed. ID and date correspond to the WOP output.
2	%% Values for the parameters of line #1	
3	%% RUNDATE(yyyy mm dd) RUNTIME(yyyy mm dd) CSMDATEf1(yyyy mm dd) CSMTIMEf1(hh mm ss) CSMDATEf2(yyyy mm dd) CSMTIMEf2(hh mm ss)	Date and time of IFM run, date and time for the CSM runs for the frequencies f1 and f2.
4	%% Values for the parameters of line #3	
5	%% Modulation frequencies [Hz]	Centre frequencies for the two bands analyzed.
6	%% f1 f2	
7	%% List of vectors included, see Table 14	
8	1st value of all vectors of line #7	
9	2nd value of all vectors of line #7	

Line	Parameter	Description
N	Last value of all vectors of line #7	

Table 13 Content of the IFM output file.

The list of vectors in line #7 consists of the parameters listed in Table 14.

Vector	Meaning
T [s]	Time stamp of the received signal
PRX_X [km]	X-position of receiver at time T
PRX_Y [km]	Y-position of receiver at time T
PRX_Z [km]	Z-position of receiver at time T
VRX_X [km/s]	X-velocity of receiver at time T
VRX_Y [km/s]	Y-velocity of receiver at time T
VRX_Z [km/s]	Z-velocity of receiver at time T
PTX_X [km]	X-position of transmitter at time T
PTX_Y [km]	Y-position of transmitter at time T
PTX_Z [km]	Z-position of transmitter at time T
VTX_X [km/s]	X-velocity of transmitter at time T, shall be negligible.
VTX_Y [km/s]	Y-velocity of transmitter at time T, shall be negligible.
VTX_Z [km/s]	Z-velocity of transmitter at time T, shall be negligible.
AMP_f1 [V/V]	Normalised amplitude for the received frequency 1 at time T.
PHASE_f1 [m]	Excess phase range for the received frequency 1 at time T. Straight line phase is subtracted.
AMP_f2 [V/V]	Normalised amplitude for the received frequency 2 at time T.
PHASE_f2 [m]	Excess phase range for the received frequency 2 at time T. Straight line phase is subtracted.

Table 14 List of vectors included in the IFM output file.

5 APPENDIX: TROUBLESHOOT CHECKLIST

A troubleshoot checklist is included for convenience.

Problem	Solution
WOP data not 1 kHz	The CSM and IRM work, but since the data is assumed to be exactly 1 kHz, the resulting Doppler will be off-set. It is recommended to interpolate the input data to 1 kHz.

Table 15 List of potential problems and possible solutions.