GRAMMAR	

README document for the TUT Simulink Galileo E1 signal simulator

Editor:	Simona Lohan
Author(s):	*Zahidul Bhuiyan (TUT), Xuan Hu (TUT), Zhang Jie (TUT), Simona Lohan (TUT)*
Total number of pages:	13

Abstract:

This document describes and the Simulink environment and parameters regarding the TUT Simulink Galileo E1 signal generator.

Disclaimer:

none

Authors

Company	Name	Contact information
TUT		
	Simona Lohan	Phone: +358 408490669 Fax: +358 3 3115 3808 e-mail: elena-simona.lohan@tut.fi
	Xuan Hu	e-mail: xuan.hu@tut.fi
	Zahidul Bhuiyan	e-mail: mohammad.bhuiyan@tut.fi
	Zhang Jie	e-mail : zhang.jie@tut.fi

Table of Contents

List of Acronyms and Abbreviations **4**

1. Simulink model **5**

1.1 Description of the Simulink model 5
1.1.1 Transmission 5
1.1.1.1 E1 Transmitter 5
1.1.2 Channel 7
1.1.3 Reception 8
1.1.3.1 Acquisition block 8
1.1.3.2 Tracking Unit block 9
1.2 Description of the Simulink model interfaces 13

2. References **13**

List of Acronyms and Abbreviations

Term	Description
AWGN	Additive White Gaussian Noise
CFAR	Constant False Alarm Rate
DLL	Delay Lock Loop
FLL	Frequency Locked loop
FPGA	Field Programmable Gate Array
ICD	Interface Control Document
LoS	Line-of-sight
MEE	Multipath Error Envelopes
OS	Open Service (Galileo) / Operating System (Computers)
PDF	Probability Density Function
PLL	Phase Locked Loop
PVT	Position, Velocity and Time
RMSE	Root Mean Square error

1. Simulink model

In the basic Simulink model (*Main_block_E1_tx_rx.mdl*), the whole E1 chain is implemented, including the E1 transmitter block, the multipath channel, the acquisition block and the tracking unit block. Both E1B and E1C channel are tracked in the Simulink model. The simulator works in an infinite front end bandwidth scenario.

The key functions, such as DLL discriminator and PLL discriminator, are implemented using m-language based S-function. It facilitates the algorithm development. The parameters used in the Simulink are realistic. For example, 3.42MHz intermediate frequency and 13MHz sampling rate. They can be modified according to the designer needs.

Almost all the signals propagated inside the Simulink have the same data type, 'single', except for the 'nco phase' which is record in the 'Record trigger time' block to avoid the truncating error.

1.1 Description of the Simulink model

The Simulink block is implemented on Matlab 2007a. The following is the detail information about the version.

Table 1-1 Matlab version at TUT

>> ver

MATLAB Version 7.4.0.287 (R2007a)
Operating System: Microsoft Windows XP Version 5.1 (Build 2600: Service Pack 3)

MATLAB Version 7.4 (R2007a)
Simulink Version 6.6 (R2007a)
Signal Processing Blockset Version 6.5 (R2007a)

1.1.1 Transmission

1.1.1.1 E1 Transmitter

The E1 transmitter block is implemented with CBOC modulation, which exactly matches the latest Galileo OS ICD from 2008 [GSA08]. In the transmitter block, E1B and E1C channels, in which the OS signals are carrier on, are modelled as the following equation, [GSA08]. The detail subcarrier parameter and weighting factor are documented in the ICD.

$$S_{E1}(t) = \frac{1}{\sqrt{2}}\left(e_{E1-B}(t)\left(\alpha sc_{E1-B,a}(t) + \beta sc_{E1-B,b}(t)\right) - e_{E1-C}(t)\left(\alpha sc_{E1-C,a}(t) - \beta sc_{E1-C,b}(t)\right)\right)$$

The snapshot of the transmitter is shown in the Figure 1-1.

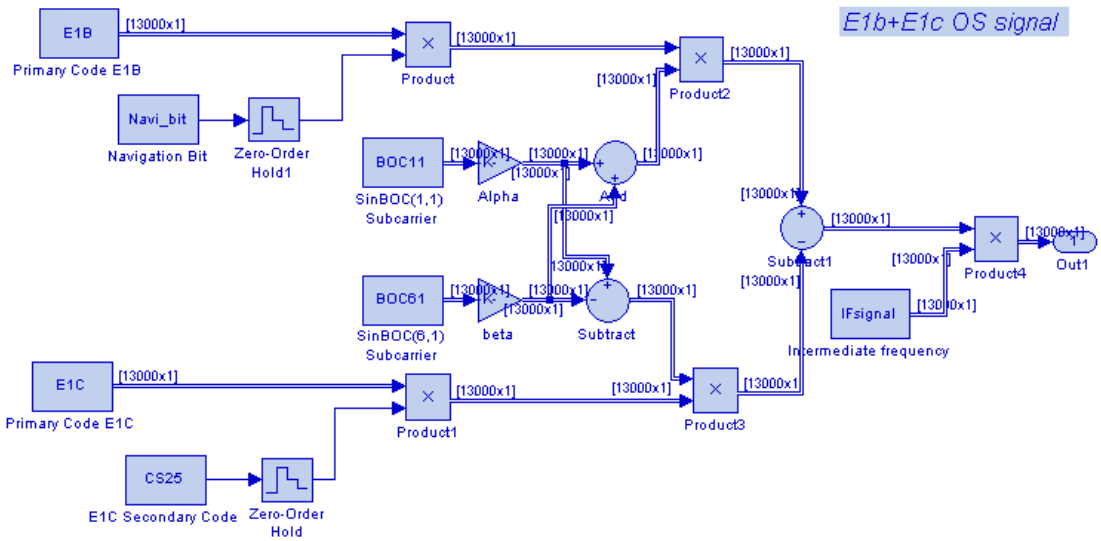

Figure 1-1 The snapshot of the E1 transmitter Simulink block

The block utilizes the frame data with frame duration 1 millisecond, in order to speed up the simulation.

We remark that the spreading factor (that is the number of chips per code symbol) is the same for Galileo and GPS signals, that is 1023 chips or 1ms. In Galileo however, the code epoch length is 4ms, which means that the codes are not the same from one code symbol to another (this is similar with the long codes concept used in WCDMA). When we do the processing over 1 ms (here referred to as frame length), we do partial code correlation in Galileo. However, since the spreading factor is still 1023 chips, this processing does not affect the processing gain.

In each frame, $f_s * 10^{-3}$ samples are included. For example, each frame contains 13,000 samples when the sampling rate is 13 MHz.

Five variables ought to be generated in the Simulink initial callback function. They are 'E1B', 'E1C', 'BOC11', 'BOC61' and 'IFsignal'.

Detailed parameters' properties are listed in the following table:

Table 1-2 Receiver model parameters

Required Data file	E1primary.mat	Stored all the primary code for E1B and E1C channel			
	E1CScode.mat	Secondary code for E1C channel			
	Variable Name	**Description**	**Unit**	**Dimension**	**Typical value**
Input parameters	sv	Satellite index for selecting the corresponding primary code		1	1~50
	fs	Sampling frequency	Hz	1	13e6, 26e6
	fIF	Intermediate frequency	Hz	1	3.42e6
	Navi_bit	Navigation bit		>=1	1, -1
Intermediate parameters	E1B	Sampled E1B primary code		[fs*0.004, 1]	1,-1
	E1C	Sampled E1C primary code		[fs*0.004, 1]	1,-1
	BOC11	Sampled subcarrier BOC(1,1) waveform		[fs*0.004, 1]	1,-1
	BOC61	Sampled subcarrier BOC(6,1) waveform		[fs*0.004, 1]	1,-1
	IFsignal	Sampled intermediate carrier waveform		[fs*0.004, 1]	

Notes:

- the navigation data has a symbol rate of 250 symbols/second in Galileo. The code epoch rate is 1000 code symbols/second (from 1ms spreading interval) => the navigation data has 4 ms length (not 20ms as for GPS before).
- In the transmitter block, E1B is the CBOC('+')-modulated signal with navigation data (upper part) and E1C is the CBOC('-')-modulated signal without data (pilot channel). E1 signal is formed as the difference between the two signals abovementioned (Eq. 11 from the SIS ICD Draft 1 [GSA08])
- In E1C channel, there is no navigation data, but secondary code. That is a 25-bit sequence (0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 1 0), not all one's.
- The sampling frequency f_s is a variable of the model (currently, it is set to 13 MHz).

1.1.2 Channel

In the channel block, only the multipath signals and complex noise are generated. The basic function of the channel block can be modelled as,

$$r_{E1}(t) = \sum_{i=1}^{l} \alpha_i(t) S_{E1}(t-\tau_i) + n$$

Here $r_{E1}(t)$ is the received E1 signal, which is the output of the channel block; and $\alpha_i(t)$ and τ_i are the complex path coefficient and the path delay for *i*-th path; n is the AWGN.

The following figure shows the basic diagram of the channel block,

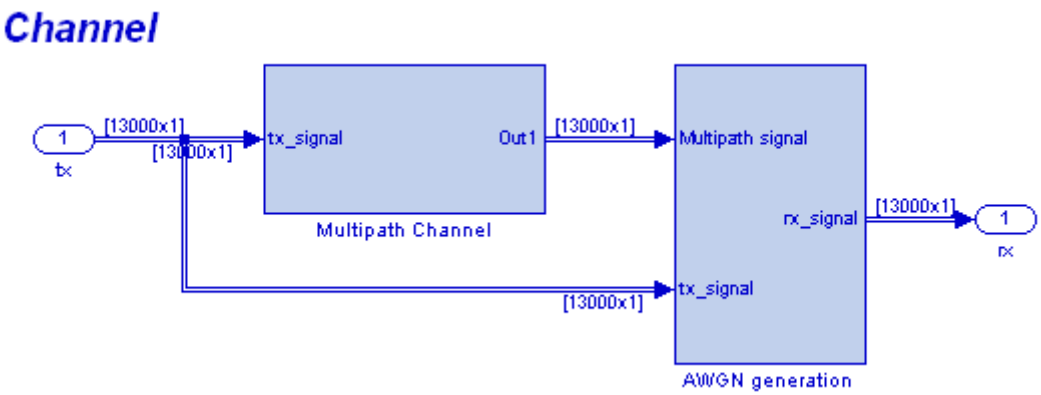

Figure 1-2 Channel subblocks; multipath and noise

The channel parameter should be defined before the simulation. The generation of the parameters can be embedded in the Simulink initial callback function. The following shows the parameters used in channel

Table 1-3 Channel model parameters

	Variable name	Description	Unit	Dimension	Example of values
Input parameters	Mpath_gain	The average path gain vector	dB	[5,1]	0 ,-3
	Mpath_delay	The average path delay vector	s	[5,1]	0, 29.3e-7
	Dop_vec	The Doppler spread vector	Hz	[5,1]	4
	CNR	The Carrier-to-Noise Ratio	dB/Hz	1	35~50
	fad_type	The fading type		1	'static' or 'fading'
Intermediate parameters	CIR_vec	The channel impulse response for the multipath channel, each row for a path.		[5, N]*	
	delay_vec	The delay vector for the path	samples	[5,1]	

	std_noise	The standard deviation ratio between the noise and carrier		1	

* N depends on the simulation time

Note:

- The path amplitude remains the same within 1 ms, which is the frame duration of the signal in the transmitter and channel. This is because the Doppler spread is usually small.
- In the Simulink initial callback functions, the function, 'chan_init_function', are used to generate the CIR_vec, delay_vec and std_noise. Two kinds of fading type, 'static' or 'fading', can be used. The sub-function 'CIR_gen_Rayleigh' is used to produce the response in the Rayleigh distributed channel.
- The 'CIR_gen_Rayleigh' can be replaced by other functions in the simulation. The key task of the function is to generate the response.
- The CIR_gen_Rayleigh should be removed when the real channel response is used. The variable 'CIR_vec' should contain the measured response.
- The whole chain is in IF frequency. Therefore, the carrier phase is also effected by the path delay. Therefore, for MEE simulation, it is better to use a baseband chain by setting the fIF to be 0.

1.1.3 Reception

The Reception mainly consists of the acquisition and tracking unit blocks.

1.1.3.1 Acquisition block

The acquisition unit uses FFT-based correlation. The acquisition detection is implemented according to a Constant False Alarm Rate (CFAR) algorithm [PLR05].

The acquisition is implemented using m-language based S-function, 'acq_E1.m'. The detailed implementation of m-language based S-function is described inside the function. The input and output data and the parameters are listed in the following table.

Table 1-4 Acquisition I/O and parameters

	Description	Unit	Dimension
Input data	Incoming signal within 4 ms		[fs*0.004,1]
Output data	Tracking Unit enabling signal		1
	Estimated frequency from acquisition	Hz	1
	Estimated code delay from acquisition	Chip	1
Parameters	Sampled E1B reference signal (the BOC modulated E1B primary code)		[fs*0.004,1]
	Sampled E1C reference signal (the BOC modulated E1C primary code)		[fs*0.004,1]
	Frequency uncertainty range	Hz	
	Receiver sampling frequency	Hz	1
	Acquisition detection threshold		1

With CFAR algorithm, the decision is made based on the correlation output. Here we use the average of the absolute value of E1B and E1C correlation output. The acquisition decision variable will be according to ratio of peaks , after setting to 0 a neighbourhoud of -3:+3 chips in time and -500:+500 Hz in frequency domain. The detection threshold is currently set to 1.3 (it can be changed).

The output data are stored in the memory, using the library provided block 'Data Store Memory'. Those are 'Tracking_Ena', 'Est_freq' and 'Est_code_delay'. The stored data are read outside the acquisition block, for instance, in the tracking unit.

Currently, the coherent integration is 4 milliseconds. Therefore, the frequency uncertainty range is from -5 kHz to + 5 kHz with step of 125 Hz. The frequency uncertainty range can be redefined according to the

requirements. The time uncertainty range is equal to 4 ms code epoch; the time-bin step is equal to 1 sample.

When the signal is acquired, we go to the tracking stage.

We remark that in the current implementation of the acquisition unit, the reference code at the receiver is the sine BOC(1,1)-modulated code. If we want to use the CBOC-modulated code as a reference signal, modifications should be done in the acquisition block and in the NCO block.

An example of the time-frequency mesh in the acquisition block is shown in Figure 1-3.

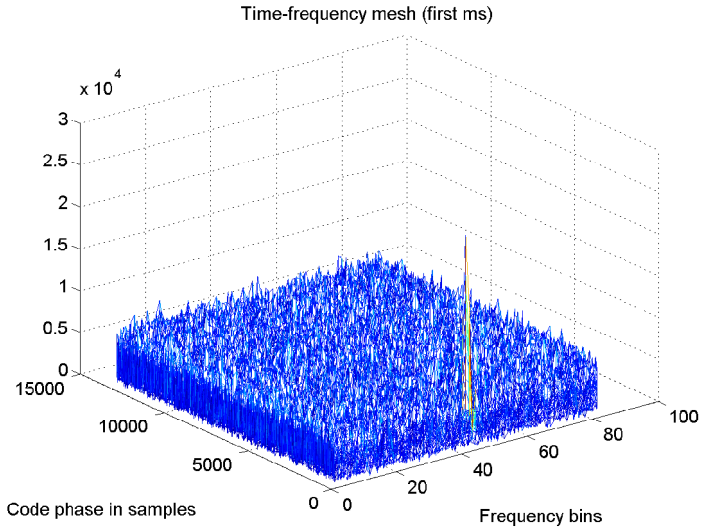

Figure 1-3. Time Frequency mesh in the E1 acquisition process

1.1.3.2 Tracking Unit block

The tracking Unit consists of three major blocks: ‘carrier wipe-off’ block, ‘Code nco’ block, and ‘Dual channel correlation and discriminators’ block.

The tracking unit is not implemented using an enabled subsystem in Simulink. This is because the ‘buffer’ and ‘unbuffer’ block does not work properly in the enabled subsystem. Therefore, the tracking unit is always running, no matter the acquisition is done or not. But when the processing is in acquisition stage, there is no feedback to change the code phase and carrier phase.

In the ‘Dual channel correlation and discriminators’ block, E1B and E1C channels are implemented separately. However, these two channels are approximately the same, except the replica used for integration and dump.

1.1.3.2.1 Carrier wipe-off block

In this block, the incoming signal is down converted to the baseband. The carrier wipe-off following the following equation:

$$r_{E1_BB}(t) = r_{E1}(t)\, e^{-j\left(2\pi \hat{f} t + \hat{\varphi}\right)}$$

Here $\hat{f}$ is the estimated frequency from the acquisition; $\hat{\varphi}$ is the estimated phase from the PLL or FLL. The carrier nco block, which is implemented using C-language based S-function is used to generated quantified carrier waveform, as shown in following table.

Table 1-5 Carrier NCO block mappings

Degree	Holding register	SIN map	COS map
0	0 0 0 …	0	3
45	0 0 1 …	-2	2
90	0 1 0 …	-3	0

135	0 1 1 …	-2	-2
180	1 0 0 …	0	-3
225	1 0 1 …	2	-2
270	1 1 0 …	3	0
315	1 1 1 …	2	2

The mapping comes from $\begin{matrix} fix(\cos(2\pi\alpha)*N), \alpha = [0:2^N - 1]/2^N \\ fix(\sin(2\pi\alpha)*N), \alpha = [0:2^N - 1]/2^N \end{matrix}$ and then mapping 0.7071 to 2 and 1 to 3 (due to N=3 bit mapping). That is: (cos([0:7]/8*2*pi)*3) and fix(sin([0:7]/8*2*pi)*3).

After the carrier wipe-off, the real part and the imaginary part of the complex signal are separated, and output as the in-phase channel and quadrature channel signal in baseband.

Note: after the 'unbuffer' block, the incoming signal is implemented in the sample level, not frame level any more. The sample duration is the determined by the sampling frequency. Therefore, 'sample time color' option in the format of the Simulink, different color should be observed between the input and output of the 'unbuffer' block.

1.1.3.2.2 Code NCO block

The 'Code NCO' block is implemented using C-language based S-function. The estimated code phase error, which is propagated from the DLL loop filter, is taken as the input to shift the code phase. The block outputs four signals: E1B replica signal, E1C replica signal, trigger enabling signal and the nco phase.

The basic behaviour of Code NCO block can be illustrated by following figure,

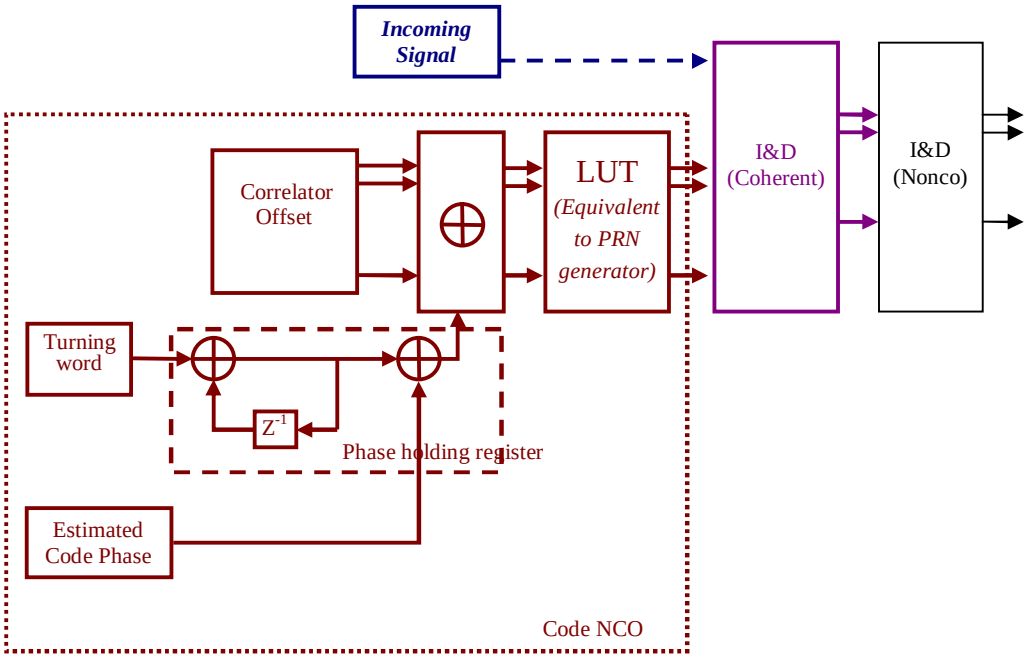

Figure 1-4 Block diagram of the Code NCO block implemented in Simulink

According the correlator offset and the status of the phase holding register, the primary code and the sub-carrier wave can be determined.

Detailed input, output data and parameters are defined in the following table

Table 1-6 Code NCO paramaters

	Description	Unit	Range
Input data	Estimated code phase error	chips	-4091~4091
Output data	BOC modulated E1B primary code replica		-1, 1
	BOC modulated E1C primary code replica		-1,1
	Trigger signal for the dump the integration in the I&D function. This signal is also used to enable the DLL, PLL discriminator and loop filter block.		0,1
	NCO phase. It is used to calculate the starting point of each epoch.		$0\sim2^{NCO_bit}$
Parameters	NCO bit number, which is the length of the nco holding register length		<= 32
	Sampling frequency	Hz	
	E1B primary code in chip level		-1,1
	E1C primary code in chip level		-1,1
	Correlator range	chips	-4091~4091
	The number of coherent integration epoch number		

In the correlator range is a vector of correlator offsets. The correlator range is defined in the code NCO. For example, if the correlator range is [-1: 0.05: 1], the offset range is form -1 chip to 1 chip, and the spacing between each correlator is 0.05 chips. Therefore, the length of the E1B replica and E1C replica is a vector whose length is 41 (that is, the length of [-1: 0.05: 1] vector). This conflicts with the limitation of the hardware. In the hardware, the correlator spacing must be in the sample level. There is a trade-off between the simulation speed and number of the correlator. When the number of correlator is increased, the simulation will be slowed down.

The last parameter of the block is the coherent integration epoch number. For Galileo E1 OS signal, each epoch contains 4092 chips, and lasts 4 milliseconds. Therefore, if the epoch number is 1, the coherent integration time is 4 milliseconds.

Currently, the replica is only modulated via the SinBOC(1,1) sub-carrier. The CBOC sub-carrier can also be implemented.

1.1.3.2.3 PLL discriminator

PLL discriminator is an m-language based S-function It is based on atan-function (see for example page 85 from Parkison & Per Enge book on GPS).

The value of PLL loop filter bandwidth can be changed according to the requirements.

PLL implementation is based on Kaplan's book (chapter 5) [KH06]. PLL loop filter is a second-order filter, whose block diagram is shown in Figure Figure 1-5, a)

—

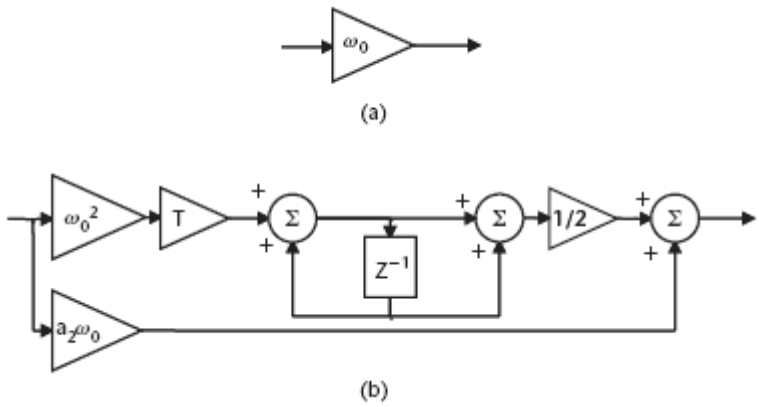

Figure 1-5. Block diagrams of a) first-order and b) second-order loops for carrier phase and frequency tracking [KH06]

1.1.3.2.4 FLL tracking

FLL discriminator is an m-language based S-function. It is based on the atan2 function, since the pull in range is twice as the one using atan function. The theoretical analysis is shown in the book of Kaplan "Understanding GPS", edition 2, chapter 5. The FLL loop filter is a first-order filter, as shown in Figure Figure 1-5, b).

It was shown in [KH06] that first and second-order loop filters are unconditionally stable at all noise Bandwidths and they are typically used in carrier tracking loops. The first-order loop is sensitive to velocity stress, while the second-order loop is sensitive to acceleration stress.

FLL loop filter bandwidth can be adjusted according to the needs.

Notes: the DLL and PLL loop bandwidths can be chosen empirically as a good trade-off between fast convergence and low noise. For example, in order to check the converge speed, we can look for example at IDI value in time (recall that IDI is a matrix (IDI(t, d), t=time, d=code phase/delay; IDI(:, d) when d corresponds to the peak shows us how fast the output converges to a stable value; if there are oscillations, then we have to decrease the loop bandwidth. Note that IDI(t,:) is the triangular-shaped correlation function at various times).

1.1.3.2.5 DLL tracking

Currently the DLL discriminator employs the normalized EML function, with discriminator output D given by:

$$D = \frac{I_E{}^2 + Q_E{}^2 - (I_L{}^2 + Q_L{}^2)}{I_P{}^2 + Q_P{}^2}$$

The discriminator is implemented using an m-language based S-function.

Most of the outputs to be stored in Matlab workspace are defined inside the DLL block

The DLL loop filter bandwidth can be changed according to the requirement.

1.1.3.2.6 Record trigger time block

In this block, the time stamp and the nco phase are record.

We can not treat the trigger time as the ending point of the epoch. This is because the resolution will be limited in the sample level, which is too big. For instance, with 13MHz sampling frequency, the resolution is 3e8/13e6, which is approximately 23 meters. Therefore, the nco phase is recorded, and used to calculate the start point of the next epoch.

Currently, the calculation is done in the Simulink stop callback function (from Model parameters). The estimated code epoch starting time stamp is stored in the variable 'tri_time'. It can also be calculated during the simulation.

Notes: the tracking stage will not work when CNR is under 30-35 dB-Hz with only 4ms integration. In order to extend the tracking range below 30-35 dB-Hz (environment dependent), modifications should be made in the model to adapt it to longer coherent and non-coherent integration times.

1.2 Description of the Simulink model outputs

The current outputs are defined in the Model parameters -> callbacks -> StopFcn*. The in-phase and quadrature correlations are plotted for the last simulation instant, and also the delay tracking error.

2. References

[GSA08] European GNSS Supervisory Authority (GSA). Galileo OS SIS ICD. Available at: https://extsites.dlr.de/kn/gsa-grammar/Work%20Packages/WP1%20Market,%20Commerical%20and%20Exploitation/WP1200%20Requirements%20Capture/GalileoOSSISICDDraft1ReformattedPDF.pdf

[LLR06] E.S. Lohan, A. Lakhzouri, and M. Renfors, "Binary-Offset-Carrier modulation techniques with applications in satellite navigation systems", Wirel. Commun. Mob. Comput., Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/wcm.407, 2006

[LR07] E.S. Lohan and M. Renfors, "Correlation properties of Multiplexed Binary Offset Carrier (MBOC) modulation", European Wireless Conference, Apr 2007.

[KH06] E.D. Kaplan and C. Hegarty, "Understanding GPS: Principles and applications", Artech House, 2006.

[PLR05] E. Pajala, E. S. Lohan, and M. Renfors, ``CFAR detectors for hybrid-search acquisition of Galileo signals'', in CDROM Proc. of ENC-GNSS, Munich, Jul 2005 [http://www.cs.tut.fi/~simona/Elina_EGNSS2005.pdf]