PI#8 Feature SP-1260 Delay Models Generation

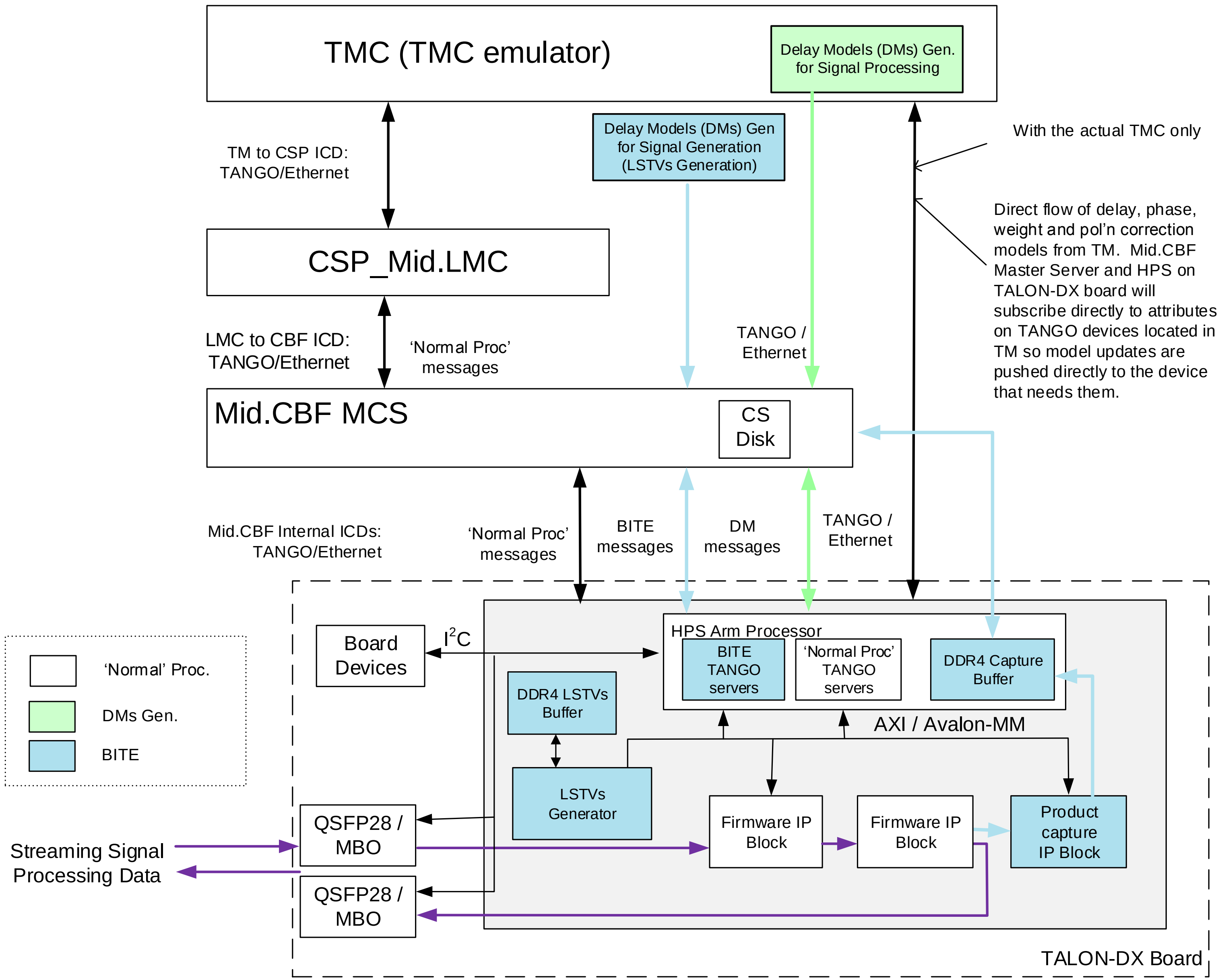
NRC/MDA

19 Nov 2020

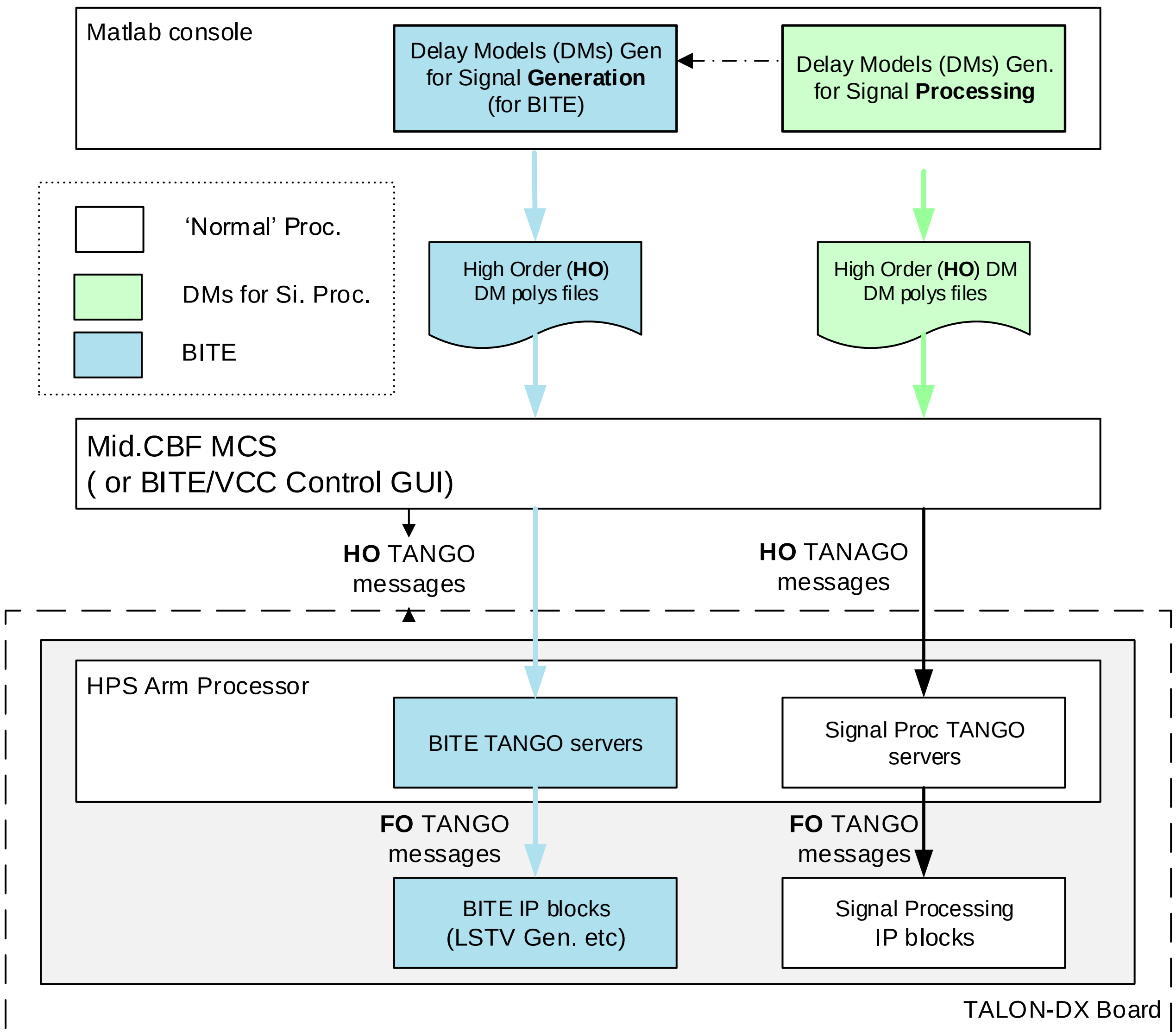
SPO-753 Objective:

Objective for CIPA Team: Extend the TDC BITE functionality to include RFI Tone Generation as well as more flexibility in defining the statistics of the receiver and sky source noise. Also **implement tools (in Matlab) (based on existing Signal Model Matlab prototype) to generate delay models (DM) and delay correction models (DMC) to be used by BITE (for LSTV generation) and the Signal Processing chain.**

This work in anticipation of future BITE and VCC MVP Builds which will incorporate application of the delays to Long Sequence Test Vectors (LSTVs) via resampling and application of matching delay corrections to the signal processing chain (also via resampling).



TDC Data Flow from TMC to FPGA IP Block Registers (based on Mid.CBF DDD, and updated for BITE)


Matlab console
Delay Models (DMs) Gen for Signal Generation (for BITE)
Delay Models (DMs) Gen. for Signal Processing
'Normal' Proc.
DMs for Si. Proc.
BITE
High Order (HO) DM polys files
High Order (HO) DM DM polys files
Mid.CBF MCS
(or BITE/VCC Control GUI)
HO TANGO messages
HO TANAGO messages
HPS Arm Processor
BITE TANGO servers
Signal Proc TANGO servers
FO TANGO messages
FO TANGO messages
BITE IP blocks (LSTV Gen. etc)
Signal Processing IP blocks
TALON-DX Board

GDM HO Polynomial File Format:

t_start[s]	t_stop[s]	c6[ns/s^6]	c5[ns/s^5]	c4[ns/s^4]	c3[ns/s^3]	c2[ns/s^2]	c1[ns/s^1]	c0[ns]
0.016	10.016	-1.2E-24	1.81E-15	3.31E-13	-1.1E-09	7.68E-05	1.216196332	-28887.5
9.016	19.016	-2.1E-23	6.26E-13	-7E-12	-1E-09	7.68E-05	1.217578551	-28876.5
18.016	28.016	-3.9E-23	1.22E-12	-1.4E-11	-1E-09	7.67E-05	1.218960245	-28865.6
27.016	37.016	9.79E-24	-3.1E-13	4.03E-12	-1.1E-09	7.67E-05	1.220341415	-28854.6
36.016	46.016	1.6E-23	-5.3E-13	6.72E-12	-1.1E-09	7.67E-05	1.221722058	-28843.6
45.016	55.016	1.85E-23	-5.8E-13	7.04E-12	-1.1E-09	7.67E-05	1.223102176	-28832.6

`t_start` - the start time, with respect dc_utc, of the validity period of the HO polynomial [s]
`t_stop` - the stop time, with respect dc_utc, of the validity of the HO polynomial [s]
$C_n, C_{n-1}, \ldots, C_0$ - The polynomial coefficients in units of nanoseconds/s^n (for readability purposes)

With these notations, the Time Varying Delay (TVD) at a given time, t, is calculated as:

$$TVD(t) = C_n*(t-t\_start)^n + C_{n-1}*(t-t\_start)^{(n-1)} + \ldots + C_1*(t-t\_start) + C_0$$

DEMO

Terminology and Abbreviations:

DC	Delay Center
BD	Beam Direction
GDM	Geometrical Delay Model(s)
FO	First Order (GDM polynomials)
HO	High Order (GDM polynomials)

- When the context calls for a generic meaning ‘*Delay Models*’ or ‘*Geometrical Delay Models*’ (GDM) (a name inherited from the Signal Model Matlab prototype) is used.
- When referring to the Delay Models generated by TMC, ‘*TMC Delay Models*’ or simply ‘*Delay Models*’ is used, as per the SKA ICDs (and other SKA documents) .
- When the context requires specificity, the following explicit naming is used:
  - ‘*GDM for Signal **Processing***’ (abbreviated for ex. as gdm_sig_proc)
  - ‘*GDM for Signal **Generation***’ (abbreviated for ex. as gdm_sig_gen) (or ‘*GDM for LSTV Generation*’)

Work Decomposition:

- Extract and refactor relevant code form the Signal Processing Matlab Model prototype (described in SKA document 311-000000-007_4) (AT5-488)
- Regression testing against original (AT5-489, AT5-490)
- Implement/test the two GDM generators (AT5-491, AT5-492)
- Reviews, Consulting, Final version, Demo (AT5-493, AT5-494)

Regression Testing Approach:

1. Implement test driver `GDM_Test_Main` to:
   - Reproduce the 'Original' parameter initialization stage
   - Reproduce the signal processing loop of the 'Original' Mid.CBF Matlab Signal Processing model (but disregarding all non-GDM signal processing steps)
   - Support both 'original' and 'refactored' GDM with a `'switch'` statement.
   - Save GDM polynomials to .csv files
2. Run both cases (using the script `GDM_Test_Run`)
3. Use comparison tools to compare the results from the two cases.

GDMs Generation:

1. A ‘main’ function for each of the GDM generators creates HO polynomials:
   - `GDM_Sig_Gen_HO_Main.m`
   - `GDM_Sig_Proc_HO_Main.m`
2. For each, save results to .csv and json files.
3. Run both (using the script `GDM_Run`)
4. Use comparison tools to compare results.

Note: The `GDM_Sig_Gen` generator depends (weakly) on parameters specific to `GDM_Sig_Gen` for two reasons:

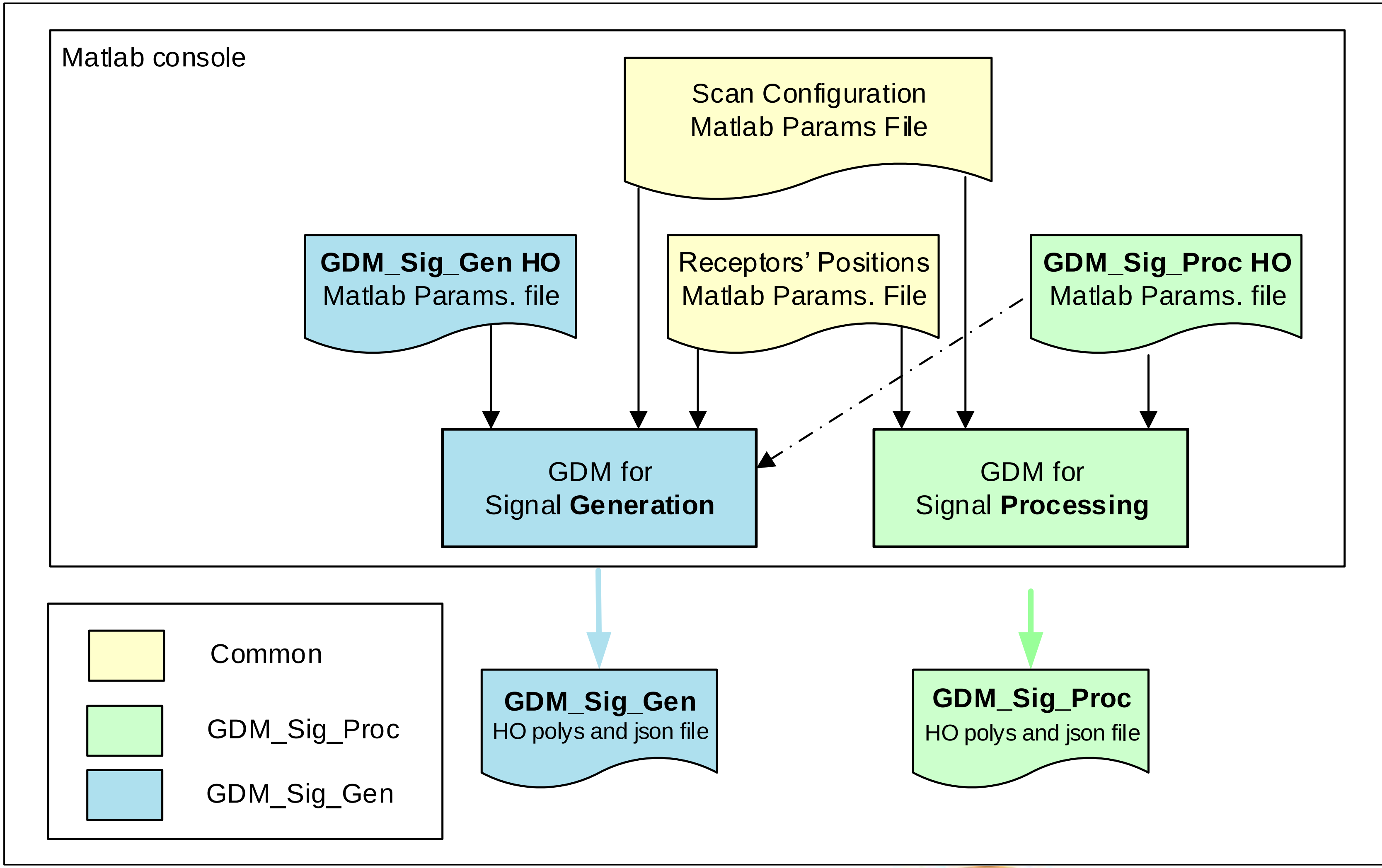
i. The Source position has to be specified relative to the Delay Center (DC) position

ii. The start time of the simulation, `t_sim_start`, for each receptor is an arbitrary number which can be set to:

```
t_sim_start  = t_sim_ref_start - approx_receptor_delay
```

where `approx_receptor_delay` can be (and currently is) set to the delay for the current receptor w.r.t. DC at the GDM reference time. This delay can be derived from the HO delay polynomials w.r.t. DC (which depends on the DC position).

GDM Generation Data Flow:

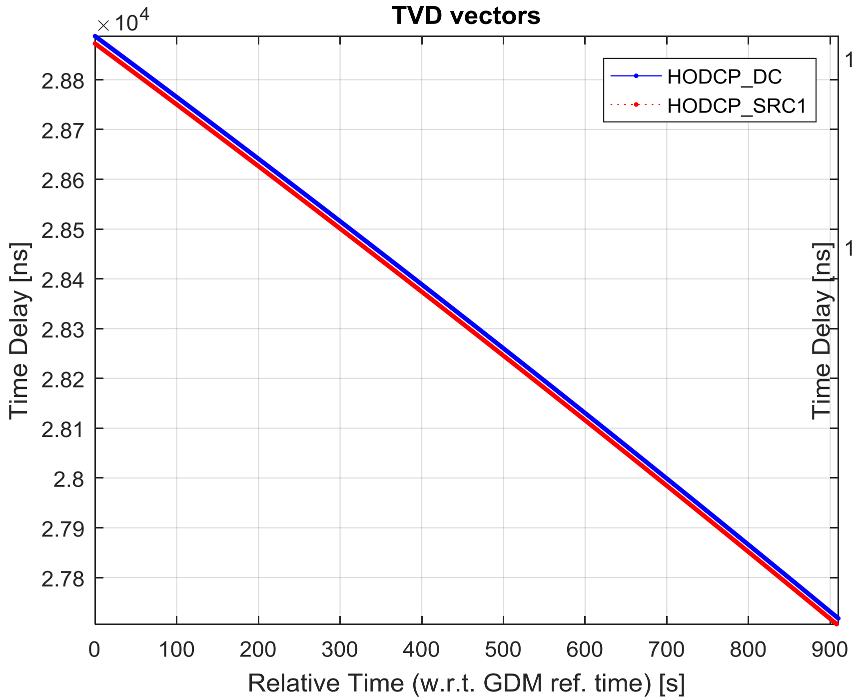


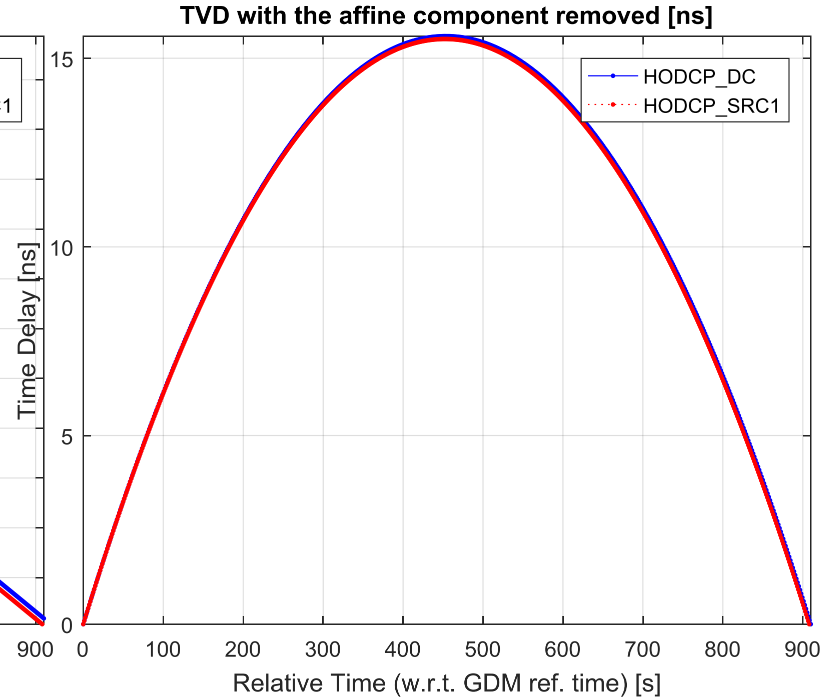
GDMs Generation in Batch Mode:

Input Parameter file to GDM: GDM_Params.xlsx

gdm_common			gdm_sig_proc				gdm_sig_gen							
test_id	run_test	procMode	dc_ecs_pos		bd_ecs_pos_offset		src1_ecs_pos_offset		src2_ecs_pos_offset		src_poly_info			t_sim_ref_start
			right_asc	declination	right_asc	declination	right_asc	declination	right_asc	declination	n_poly	duration	n_points	
T1	1	PST-BF	60	0	0	1.84	0	1.84	-1	1.5	5	8	500	0.015
T2	1	PST-BF	-60.2	0	-0.2	1.84	0	1.84	-1	-1.5	4	9	500	0.11
T3	1	CORR	60	30			0	1.5	0	-1.5	4	10	1000	0.11

Examples (1/2): Compare DC and Source GDM





Examples (2/2): Compare DC GDMs for 2 Receptors