Simulink DLL based Interface

SimulinkDLL1

Simulink DLL Interface

Si	mulink DLL	- based Interface	1
1	Descript	ilon	1 1
2		ments and compatibility	וו ר
ა	2.1 Initi	el actua in MATLAP	∠ک
	3.1 IIIII	al Setup III MATLAD	∠۲ د
		PS 101 Clealing a DLL	۲ ۲
	3.2.1	Sample Simulink diagram	
	3.2.2	Long generation steps	
	3.2.3	Importing the model into EMTP works	ð
	3.2.4	Rules and limitations	
	3.2.5	Searching for tunable parameter problems on Simulink side	
4	Simulink	condel options and configurations	12
	4.1 Mod	del Calling Rule	13
	4.1.1	Simulink time-step (default option)	
	4.1.2	User-defined	
	4.1.3	On EMTP time-mesh	
	4.1.4	At any time-step	
	4.2 Oth	er Settings	
	4.2.1	DLL version	
	422	Interpolation and Extrapolation options	14
	4.3 Beh	haviour when $\Lambda tEMTP \neq \Lambda tCallSimulink$	11
5	Compati	ibility with EMTP simulation options	

César Martin (RTE-France), Jean Mahseredjian, 5/11/2023 1:37:00 AM

1 Description

This device named "Simulink DLL Interface" (SDI) is used to interface EMT with DLLs created using Simulink[®] (MATLAB[®]). The SDI is an EMTP Toolbox.

Once a DLL is created using Simulink and a compiler, this device is used to create all the necessary connections (pins) and provide access to some modifiable parameters on the Simulink side.

The interfacing pins (ports) on the Simulink side must be created by the user. The interfacing pins in EMTPWorks are automatically created with the SDI device. The user must then finalize all the desired connections in EMTPWorks to use the Simulink code in the simulations on EMTP side.

2 Requirements and Compatibility

The user must have a valid MATLAB license with Simulink. The following MATLAB Toolboxes are minimal requirements, but this may change according to MATLAB versions. Required Toolboxes:

- 1. Matlab Coder
- 2. Simulink Coder
- 3. Embedded Coder

Some other Toolboxes may be required according to Simulink designs.

This version of the SDI has been tested with the following MATLAB versions: R2013b to R2021b. It may also work with older Matlab versions.

It is noticed that under some circumstances, especially when S-Functions and MATLAB functions are used in Simulink diagrams, the simulation results may be corrupted or the DLL generation process may fail. It is the user's responsibility to verify coherency in simulation results.

In addition to the above MATLAB requirements, the user must install a C++ compiler. The compatible compilers are Microsoft Visual Studio C++ version 2010, 2013, 2015, 2017 and 2019. The Visual Studio C ++ version must be compatible with the Matlab version used. See "supported compilers" for each Matlab version here https://fr.mathworks.com/support/requirements/previous-releases.html

3 DLL generation and integration steps

In the following sections, the name <u>ApplicationDir</u> designates the application folder (path):

- 1. On a 64 bit Windows system the default value is C:\Program Files (x86)\EMTPWorks.
- 2. On a 32 bit Windows system the default value is C:\Program Files\EMTPWorks.

The EMTP Toolboxes are located in the folder <u>ApplicationDir\Toolboxes</u>. The SDI Toolbox is located in the folder <u>ApplicationDir\Toolboxes\SimulinkDLL</u>.

The name <u>UserDataFolder</u> designates the EMTP Documents folder in Windows. On a typical computer this is given by:

C:\Users\<User Name>\Documents\EMTP

The name <u>ModelName.dll</u> designates the DLL file generated from Simulink.

3.1 Initial setup in MATLAB

The following commands must be executed in MATLAB to prepare for automatic generation of DLLs from Simulink.

1. The TLC_EMTP folder

As a first step it is necessary to access the TLC_EMTP folder. This folder is normally located in the UserDataFolder. An initial version of this folder is also found in ApplicationDir\Toolboxes\SimulinkDLL. The user must run the following command in the MATLAB window: >> addpath(genpath('UserDataFolder\SimulinkDLL\TLC_EMTP')) >> savepath

2. Selecting the compiler in MATLAB

If the C++ compiler is not already selected in MATLAB, the user must run the following MATLAB command:

>> mex -setup or mex -setup c++

3.2 Steps for creating a DLL

The Simulink diagram may contain the following:

- 1. One or several Input ports (Inports) for interfacing on EMTPWorks side
- 2. One or several Output ports (Outports) for interfacing on
- 3. The ports may be real or complexe of dimension 1
- 4. The ports may be real or complexe of dimension greater than 1
- 5. One or several Subsystems
- 6. Different sampling rates
- 7. Tunable parameters: parameters than can be modified in EMTPWorks side
- 8. MATLAB functions
- 9. S-Functions

Under some circumstances the MATLAB functions and S-Functions may not perform as expected and the user must verify that the simulation results are coherent.

3.2.1 Sample Simulink diagram

The Simulink example shown in Figure 1 is used here to demonstrate the DLL generation steps. The top level model (design) is shown in Figure 1a and the contents of the sample Subsystem are shown un Figure 1b.

. The Simulink model (mdl) file corresponding to Figure 1 can be found in the folder ApplicationDir\Toolboxes\SimulinkDLL\Example\SimpleDemo.

The individual components on Simulink side must be correctly specified using Simulink input data tabs. In this example:

- 1. The ports In_complex_1 and In_complex_2 have "Port Dimensions" set to 1 and "Signal type" set to complex. The "Sample time" is -1 for inherited.
- 2. For the In_real port, "Port dimensions" is set to -1, "Sample time" is set to Ts, the "Signal type" is auto.
- 3. The Out_complex_vector is set to inherit its "Port dimensions" and "Signal type" is auto which makes it of dimension 2 and of complex type.
- 4. The Out_vector_real port inherits its dimension of 2 and its "Signal type" type is also inherited as realtype.
- 5. For the Subsystem:
 - a. the gain block Gain_K is set to the variable value K
 - b. the gain block Gain_L is set to the variable value L
 - c. the Discrete-Time Integrator has an inherited sample time
 - d. the Discrete-Time Integrator_Tsx100 has a sample time of 100*Ts

Since this example contains variables in some of its blocks, it is necessary to initialize those variables in the MATLAB work space. In this case it is necessary to enter:

>> K = 2;

>> L = 3;

>> Ts = 10e-6;

Or to run the provided m-file init_data.m.



a) Top level model



b) Subsystem

Figure 1 Sample Simulink diagram

3.2.2 Code generation steps

The steps are as follows:

- 1. Select the configuration parameters using the Simulink menu "Simulation>Model Configuration Parameters" (or by clicking on CTRL+E) and click on Code Generation.
 - a. It is necessary to locate and select the "System target file" EMTP.tlc using the Browse button as shown in Figure 2.
 - i. In the shown example, there is no referenced model, but it is allowed to use such models by setting exactly the same target file (EMTP.tlc) and the same configuration parameters in all references models. It is not allowed to use tunable parameters when referenced models are used.
 - b. Under the "Code Generation" submenu, select "EMTP code generation options" (see Figure 3):
 - i. Delete source files after code compilation: this option can be used to delete source code for confidentiality reasons.
- 2. All files related to this model (design) must be located in the same folder as the model file.
- 3. Select Solver (see Figure 4) and verify that the Fixed-step size is set to the variable Ts.
- 4. In this optional step it is possible to select <u>tunable parameters</u>. The tunable parameters are parameters that can be modified from EMTPWorks without recompiling (regenerating) the Simulink DLL. Two different methods depending on the Matlab version used.
 - a. For Matlab version R2018b or older:
 - i. In the menu Optimization>Signals and Parameters, click on the Configure button (when "Inline parameters" is checked).
 - ii. Select the "Source list" being "Matlab workspace variables".
 - iii. Select the tunable parameters, and click on "Add to table", as shown in Figure 5. The tunable parameters can be of type real, logical or integer. Complex numbers, vectors and matrices are not allowed. Some parameters (like Ts) cannot be tunable.
 - iv. Click OK to proceed.
 - b. For Matlab version R2019a or newer:
 - i. From the Simulink circuit, open « Model Explorer » under the Modeling tab or click CTRL+H
 - ii. In the Model Hierarchy on the left, select your Model workspace and click on Add then Matlab variable (or move the variable from the Matlab workspace to the Model workspace if the variable was previously created in the Matlab workspace)
 - iii. Choose a name and set a value to the parameter if it hasn't been done already
 - iv. Check the "argument" box
 - v. Right click on the variable, then "Convert to parameter object"
 - vi. Under "StorageClass" click on Configure
 - vii. Select "Model default » for the storage class

	🖮 Model Ex	plorer								
	<u>F</u> ile <u>E</u> dit <u>\</u>	<u>/</u> iew <u>T</u>	ools <u>A</u> do	<u>H</u> elp						
	2 🗖 🖌	8 8	₩ 🗄	MATLAB Variable	Ctrl+M					
	Nodel Hierarchy	,	[10] 010	Silhulink Parameter	Ctrl+P	del Workspac	e (only)			Filter Contents
	🗸 🎦 Simulir	nk Root		Simulink <u>L</u> ookupTable		Ohiosha	- 0	Dataila		0 -hi-hi
	🛨 Ba	se Works	pace	Simulink <u>B</u> reakpoint	cul. c	Dijects	• <u>S</u>	low Details		<u>U ODJect(s</u>
	🗸 🎦 tes	ts_R2020)b_10t	Simulink Signal	Ctri+S	staType Dim	ensions C	omplexity Mir	Max Unit Arg	ument StorageClass
	0	Configu	uration 🛄	Simulink AliasType		I				
		Evterna		Simulink Bus		I				
		MATLA	B Fun	Simulink <u>V</u> ariant						
	含	Subsyst	tem 🙆	Configuration		I				
				Configuration Reference	e					
				Add <u>C</u> ustom		I				
				Event	Ctrl+E	I				
				Data	Ctrl+D	I				
			4	Trigger		I				
			JX	Function Call Output						
	Name	Value	e DataT	ype Dimensions	Complexity	Min Ma	x Unit	Argument	StorageClass	_
	<u>н</u> г			- 7 74 43						
			Rename							
			Rename	All						
		X	Cut		Ctrl+X					
		A	Conv		Ctrl+C					
					curre					
			Paste							
			Select A	,u	Ctrl+A					
			Unselec	t All	Ctrl+U					
			_			-				
			Find Wh	iere Used						
			Export 9	selected						
			Compar	e Selected						
			Convert	to parameter object	ct					
de Mappi	ngs - C		K							
ata Defau	ilts Functi	ion Def	aults	Functions Innor	rts Outry	vrts Par	ameters	Data Stor	es Signale/	States
	its runcu	ION DEI	auits	Functions Inpor	is outpu		ameters	Data Stor	ies bigridis/a	biates
				Source						
External F	Parameter Ob	jects (0))				Refr	esh		
Model Par	rameter Argu	ments ((1)							
[10] 010 L								Model de	fault	
Model Par	rameters (0)									

5. At this stage the setup is ready to generate the DLL by clicking on the Build button in the top panel of the Simulink circuit or by clicking on CTRL+B from the Simulink circuit.

<u>If there are no error messages</u> in MATLAB, then the build is successful and two new folders will appear in the same folder as the compiled model file folder: slprj and ModelName_EMTP_rtw. In this example the ModelName_EMTP_rtw folder is tests_R2018b_10us_full_EMTP_rtw.

The final DLL file of the model is also located in the same folder as the model file. Its generic name is ModelName.dll (tests_R2018b_10us_full.dll for this example). The two folders slprj and ModelName_EMTP_rtw are not needed and can be deleted, only the DLL file is needed to run in EMTP.

Q Search	
Solver Data Import/Export Math and Data Types Diagnostics Hardware Implementation	Target selection System target file: EMTP.tic Language: C++ Description: v2.0 Create Visual C/C++ 32 bits Solution for EMTP-RV
Model Referencing Simulation Target	Build process
Optimization Report Comments Symbols Custom Code Interface Code Style Verification	Generate code only Package code and artifacts Zip file name: <empty> Makefile configuration Generate makefile Template makefile: EMTP_vc.tmf Make command: make_rtw</empty>
Templates Code Placement Data Type Replacement EMTP code generation op Simscape	Code generation objectives Prioritized objectives: Unspecified Check model before generating code: Off Check Model

Figure 2 Simulink Model Configuration Parameters, Code Generation.

Q Search		
Solver	Delete source files after code compilation	
Data Import/Export		
Math and Data Types		
 Diagnostics 		
Hardware Implementation		
Model Referencing		
Simulation Target		
Code Generation		
Optimization		
Report		
Comments		
Symbols		
Custom Code		
Interface		
Code Style		
Verification		
Templates		
Code Placement		
Data Type Replacement		
EMTP code generation op		
Simscape		

Figure 3 Simulink Model Configuration Parameters, EMTP code generation options.

Q Search									
Solver Data Import/Export Math and Data Types Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation Optimization Report Comments Symbols Custom Code Interface Code Style Verification Templates Code Placement Data Type Replacement EMTP code generation op	Simulation Start tim Solver sel V Solver d Fixed Tasking Period Al Al Hi	n time me: 0.0 Fixed-step details d-step size (func g and sample time reat each discre llow tasks to ex utomatically hai ligher priority va	amental sam ne options constraint: [te rate as a s acute concurr ndle rate trans lue indicates	ple time): T Jnconstrain eparate task ently on targ ittion for dat nigher task p	Sto ▼ Ts ned k get ta tra prio	ansfer	100e-6 discrete (no con	ntinuous states	s) -

Figure 4 Simulink Model Configuration Parameters, Solver.

ource list		Global (tunable)	parameters			
Referenced workspace varia	bles 🔽	Name	Storage class		Storage typ	e qualifier
Name		1 K	Model default	*		×
1 6		2 L	Model default	*		1
21						
3 Ts						
0.00						

Figure 5 Simulink Model Configuration Parameters, Tunable parameters option for Matlab R2018b or older

3.2.3 Importing the model into EMTPWorks

Once the previous DLL code generation steps have been completed, it is possible to import (interface) the DLL into EMTPWorks. The user must drag in the device named "Simulink DLL" into the design and double-click on it for selecting the Simulink model. The "Select Simulink Model" button (see Figure 6) is used to browse and select the DLL file ModelName.dll generated in the above steps. In the example of Figure 1 the ModelName.dll is actually tests_R2018b_10us_full.

Once the DLL selection is completed, some model information is displayed (name, time-step, versions of Matlab and Simulink, etc.) and the tunable parameters (if any) are shown (see Figure 6) in the Data tab. The corresponding EMTP file EMTPSimulinkDLL.ecf can be found in the folder Examples.

The tunable parameters can become named-values and used in upper level subnetwork masks on EMTPWorks side, if needed.

Properties for Simulink DLL SimulinkDLL1		
Data Options		
Simulink DLL v2.0 (Documentation)		~
Select Simulink Model (.dll)		
tests_r2018b_10us_full.dll		
Model Name: tests_r2018b_10us_full		
Model Time-Step: 0.00001 s		
Model Created On: Tue Aug 30 14:51:28 2022		
Model Version: 2.0 Matlah Version: 9.0 (B2018b) 24 May 2018		
Simulink Version: 9.2		
2 tunable parameters used		
к 2		
3		
L P		
		\sim
0 🔽 Display Scale	OK	Cancel

a) Main data tab

Properties for Simulink DLL SimulinkDLL1

Choose at which time-step the model is called: Simulink time-step of 0.00001 s (recommended) User-defined: 1 s On EMTP time-mesh At any time-step	,
Other Settings Contended on the settings Options below apply only if the "no time-step delay" version of the DLL is used and if EMTP time-step is different from that of the Simulink model. Contended to the setting of the setting	

b) Options

Figure 6 EMTPWorks device "Simulink DLL" data tabs.

When the OK button of Figure 6 is clicked, the EMTPWorks interface block (see Figure 7) is automatically drawn with all the appropriate input and output pins and pin types. These are all control (Input or Output) pins.

×



Figure 7 EMTPWorks interface block with Simulink DLL.

The interface block of Figure 7 is related to the selected version of the Simulink DLL. If the DLL is recompiled (regenerated) the interface will not be automatically updated with changes in interfacing pins, time-step and tunable parameters on Simulink side.

It is apparent from Figure 7 that vector type ports (pins) on Simulink side become bundle pins on EMTPWorks side. The number of pins in each bundle is show on the bundle pin.

It is noticed also that a complex signal on Simulink size is interpreted as a bundle on EMTWorks side. This is illustrated for the Out_complex_vector pin in Figure 8.



Figure 8 Transformation of a complex Simulink vector (see Figure 1) into EMTPWorks bundle.

3.2.4 Rules and limitations

The complexity of Simulink may create various incompatibilities and issues for code generation and interfacing with EMTP. Here is a set of known rules that the user must follow.

- 1. The top level model (design) in Simulink must contain all Input and Output ports for interfacing with EMTP. Simulink export is not allowed for a model without input.
- 2. Tunable parameters
 - a. A tunable parameter is a parameter that can be modified in Simulink during simulation.
 - b. Tunable parameter settings are ignored when referenced models are used in Simulink.
 - c. In some cases the tunable parameters may cause problems by losing tunability. In such cases MATLAB may not necessarily send an error message or the error message may be generic. See the following section for debugging such conditions.
- 3. S-Function
 - a. There are two types of S-Functions in Simulink: standard S-functions based on C or C++ and inlined S-functions based on Simulink TLC files.
 - b. To be able to use two or more Simulink models (DLLs) on EMTP side, it is necessary to use only inlined S-functions. See also MATLAB documentation "Write S-Functions that Support Code Reuse" for other issues, such as non-static variables in S-function codes.
 - c. The multi-instance compatibility problems can be avoided by using two independent DLLs (different names) that can be generated using two different Simulink model file names for the exact same model.
- 4. Algebraic loops
 - a. Simulink uses an iterative method for solving algebraic loops. A typical algebraic loop is shown in Figure 9.
 - b. It is not possible in Simulink to generate code that can solve algebraic loops. In such cases, the user must artificially break the loops by inserting delays and validate that the block diagram maintains desired functionalities.



Figure 9 Algebraic loop condition in Simulink

3.2.5 Searching for tunable parameter problems on Simulink side

The safe approach for avoiding tunable parameter problems is to test them on Simulink side before establishing the interface with EMTP.

The user can search for all blocks using tunable parameters and test the tunable functionality by changing the parameter to observe its impact on simulation results. The "Inline parameters" option in "Optimization of Signals and Parameters" (see Figure 3) must be unchecked for performing the following tests.

If L is a tunable parameter, it can located in a model using the CTRL+F keyboard command. The window of Figure 10 with the selected configurations is used in the example of this document. The search results are listed at the bottom of the window. The user can open the dialog box of a listed block, make a change and validate that the parameter is tunable during the simulation by pressing Apply.

The user can search for blocks with masks and validate that tunable parameters are not used in the Initialization tab. A sample case is shown Figure 12. The variable L (see Figure 12a) is used to define the variable A in a

subsystem mask (see Figure 12b) and the variable A is used for initialization in see Figure 12c. In this case the variable B in see Figure 12c is not tunable, which means that changing L will affect A, but not B! In this example, if the initialization tabs are not using the variable A, the user must continue searching by replacing the searched variable L by the variable A since this variable has inherited tunability.

Further details can be found in MATLAB documentation (see Tunable Expression in Subsystem Mask Dialog Box).

Look for Select Simulink objects Find what: Annotations Image: Contains select Signals Image: Contains select Stateflow objects Image: Contains select Masked systems Image: Contains select Linked systems Start in system Referenced models Start in system Type Name	Look for S			Eind
Look inside Masked systems Linked systems Linked systems Referenced models Type Name Parent Source Destin	imulink objects Annotations Blocks Signals itateflow objects	Basic Advanced Find what: L Search block dialog para	Imeters	Cance
Type Name Parent Source Destin	ide ked systems ed systems renced models	Match case Con Start in system test_tunable	tains search s 💌	
	Name	Parent	Source	Destinatio
Block Subsystem1 test_tunable	Subsystem1	able		
Block Gain test_tunable/Subsystem1/Subsystem Block Gain1 test_tunable/Subsystem1/Subsystem	Gain Gain1	able/Subsystem1/Subsystem able/Subsystem1/Subsystem		

Figure 10 Search window for tunable parameter in Simulink.

Main	Signal Attributes	Parameter Attributes	
Gain:			
12			
Aultiplie	cation: Element-wise	e(K.*U)	
ample	time (-1 for inherite	d):	
-1	•		
*			

Figure 11 The Gain_K block, the tunable parameter can be replaced (see Gain field) and Applied.

	Source Block Parameters: Subsystem1	23
	Subsystem (mask)	^
	Parameters	н
	var A	-
	2*L	÷
	OK Cancel Help Apply	/
a)	Subsystem mask definition with tunable parameter	er L
	Mask Editor : Subsystemi	
	Dialog parameters	- 3
	# Prompt Variable Type Evaluate Tunable Tab na 1 var A A edit ▼ ✓ ✓	
	Type-specific options D No type-specific options D	Dialog Dialog Dialo
	Unmask OK Cancel Help Ap	oply
b)	Mask Editor for Subsystem, dialog Parameters ta	b
	Mask Editor : SubsystemI	
	Dialog variables	
	A B = 3*A;	
	Allow library block to modify its contents	
	Unmask OK Cancel Help Ap	ply

c) Mask Editor for Subsystem, Initialization tab

Figure 12 Simulink Mask with tunable parameter

Simulink model options and configurations 4

See Figure 6 for the Simulink model options

The EMTP numerical integration time-step is named hereinafter Δt_{EMTP} . The intrinsic time step of the Simulink model, i.e. the one used during the code generation from Simulink is noted $\Delta t_{InternalSimulink}$. This value is not modifiable, it is hard-coded in the DLL.

The one used to call the Simulink model is referred to as $\Delta t_{CallSimulink}$. This value is modifiable in the Model Calling Rule part of the Options tab.

4.1 Model Calling Rule

This option allows an advanced user to choose when EMTP will call the model. Keep in mind that the internal model time step is not modifiable.

The recommended configuration is $\Delta t_{EMTP} = \Delta t_{InternalSimulink} = \Delta t_{CallSimulink}$, but the setup allows to handle different situations and may provide acceptable results in some cases.

Here are possible options for $\Delta t_{CallSimulink}$

4.1.1 Simulink time-step (default option)

In this case, we have $\Delta t_{InternalSimulink} = \Delta t_{CallSimulink}$. If Δt_{EMTP} is set to the same value, we are in the recommended configuration and the Simulink DLL is called at every full EMTP time step. If Δt_{EMTP} is different from Δt_{EMTP} , calls are adapted (see 4.3)

4.1.2 User-defined

Enter a specific value for $\Delta t_{CallSimulink}$.

4.1.3 On EMTP time-mesh

The Simulink DLL is called at every full EMTP time step (Δt_{EMTP}). It does not include calls at intermediate Backward Euler time-points neither calls at all "Simultaneous switching" solutions when re-solving control systems as specified.

4.1.4 At any time-step

The Simulink DLL is called at all solution time-points in EMTP so whenever EMTP requests it.

a. It includes calls at intermediate Backward Euler time-points in EMTP

b. It includes calls at all "Simultaneous switching" solutions when re-solving control systems as specified. Such a selection means the Simulink model is time-step independent and can provide accurate response at every call.

4.2 Other Settings

4.2.1 DLL version

By default, the "no time-step delay" version is selected. That means the SDI is called following the solution of EMTP control equations.

If this option is unchecked:

a) It is expected that simulations results in EMTP will be delayed by the simulation time-step. At startup all Simulink DLL output signals are initialized to zero.

The SDI will be called following the solution of EMTP network equations (as when calling a control diagram in EMTP) and the outputs from an SDI block are used in the network side solution at the following time-point. This means that there is a time-step delay between the network solution on EMTP side and the solution of an SDI block.

b) In some cases, the performance (in terms of simulation time) is improved, especially if the number of outputs of the model is large

Mutli-instance (two or more Simulink models pointing to the same DLL) using different values for this option is not allowed.

4.2.2 Interpolation and Extrapolation options

In case the "no time-step delay" version is selected, it is possible to use the Interpolation and Extrapolation options. These options are enabled by default and relevant only in the case where $\Delta t_{EMTP} \neq \Delta t_{CallSimulink}$. For example, if $\Delta t_{EMTP} = 10 \mu s$, $\Delta t_{CallSimulink} = 7 \mu s$, and the Interpolation and Extrapolation options are enabled :

- $t = 0\mu s$: the Simulink model is called with inputs values a $t = 0\mu s$
- $t = 10\mu s$: inputs values at $t = 7\mu s$ are estimated by interpolation between inputs values a $t = 0\mu s$ and inputs values at $t = 10\mu s$. Then the Simulink model is called. The values of the outputs obtained are estimated values at $t = 7\mu s$ so extrapolation is done to get estimated outputs values at $t = 0\mu s$.
- Etc.

This must be done with caution and sufficiently good accuracy may not be achieved

4.3 Behaviour when $\Delta t_{EMTP} \neq \Delta t_{CallSimulink}$

Here are some examples:

- 1. $\Delta t_{CallSimulink} = 10 \mu s \Delta t_{EMTP} = 20 \mu s$
- $t = 0\mu s$: the Simulink model is called
- $t = 20\mu s$: the Simulink model is called twice
- $t = 40 \mu s$: the Simulink model is called twice
- etc.
- 2. $\Delta t_{CallSimulink} = 20\mu s \Delta t_{EMTP} = 10\mu s$
- $t = 0\mu s$: the Simulink model is called
- $t = 10\mu s$: the Simulink model is not called
- $t = 20\mu s$: the Simulink model is called
- $t = 30\mu s$: the Simulink model is not called
- $t = 40\mu s$: the Simulink model is called
- etc.

Inputs and outputs values differ depending on whether the Interpolation and Extrapolation options are enabled or not.

5 Compatibility with EMTP simulation options

The "Statistical Options" are compatible with the Simulink DLL.

When the "Simultaneous switching" option is used, it may request to "Re-solve control system equations". This means that the control system equations and the Simulink DLL are called more than one times at the same simulation time-point. The Simulink DLL is called only once (first solution time-point) except when the option "call at any time-step" is checked (see 4.1)

There is no steady-state solution with a Simulink DLL.