Simulink To SpaceEx Converter tool (SL2SX)

(User Guide)

Stefano Minopoli and Goran Frehse

December 17, 2014

Example: DCMotor.mdl (provided)

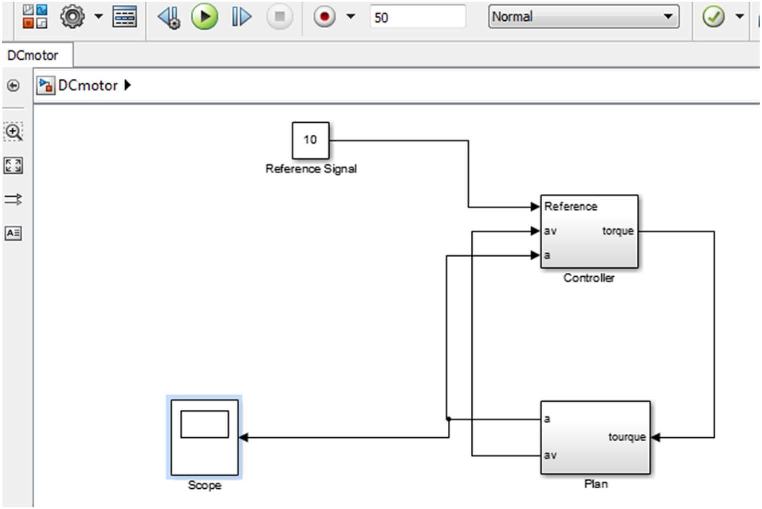


Figure: DCMotor.mdl Simulink Diagram

Example: DCMotor.mdl (provided)

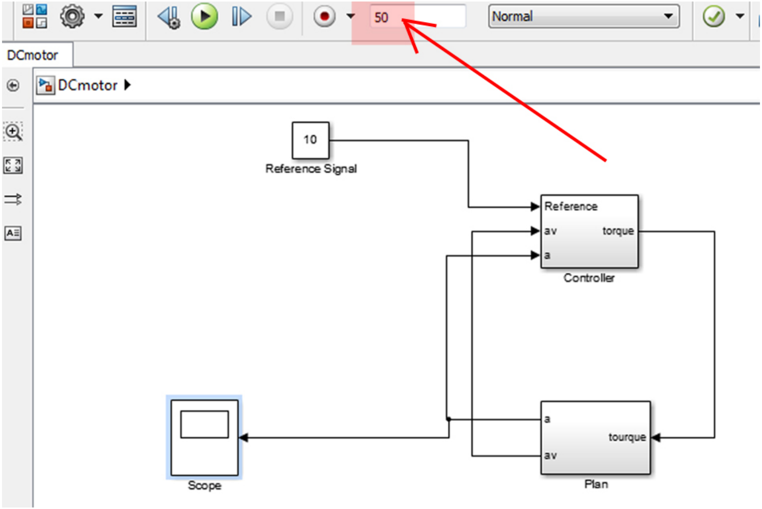


Figure: Simulation Time

The tool SL2SX: Usage Example

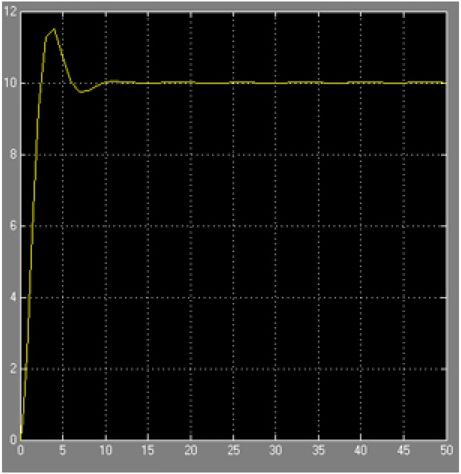
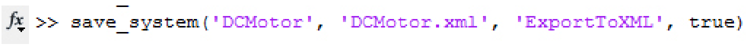


Figure: The Result of the Simulation (over 50 seconds)

The tool SL2SX: Usage Example

Step 1: Convert to .xml

- We need the .xml version of the Diagram



```
>> save_system('DCMotor', 'DCMotor.xml', 'ExportToXML', true)
```

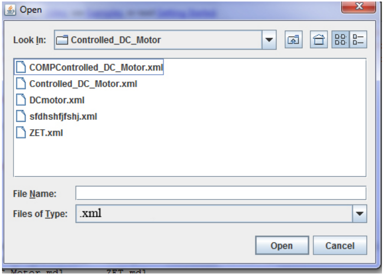
Figure: MATLAB command to export in .xml

- From the file DCMotor.mdl ⇒ to the file DCMotor.xml

The tool SL2SX: Usage Example

Step 2: Run the tool

- Double click on the file *“Tool SimuLinkToSpaceEx (SL2SX).jar”*: a dialog box will be appear to open a file



Look In: Controlled_DC_Motor

COMPControlled_DC_Motor.xml
Controlled_DC_Motor.xml
DCmotor.xml
sfdhshfjfshj.xml
ZET.xml

File Name:

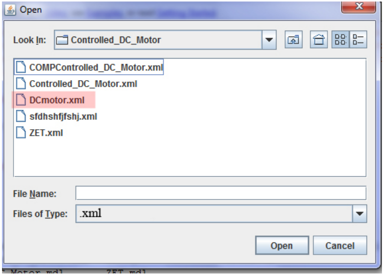
Files of Type: .xml

Figure: Dialog Box to choose the .xml file of the Simulink Model

The tool SL2SX: Usage Example

Step 3: Select the file

- Choose the file: we select *DCmotor.xml* (it is the file obtained after Step1)



Look In: Controlled_DC_Motor

COMPControlled_DC_Motor.xml
Controlled_DC_Motor.xml
DCmotor.xml
sfdhshfjfshj.xml
ZET.xml

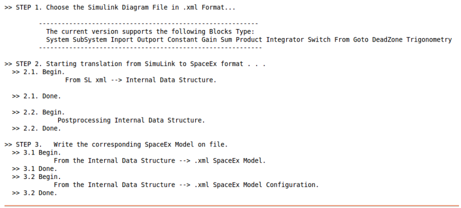
File Name:

Files of Type: .xml

Figure: Open DCMotor.xml

The tool SL2SX: Usage Example

Step 4: Wait for the translation



```
>> STEP 1. Choose the Simulink Diagram File in .xml Format...

        -----------------------------------------------------
          The current version supports the following Blocks Type:
          System SubSystem Inport Outport Constant Gain Sum Product Integrator Switch From Goto DeadZone Trigonometry
        ------------------------------------------------------

>> STEP 2. Starting translation from SimuLink to SpaceEx format . . .
  >> 2.1. Begin.
               From SL xml --> Internal Data Structure.

  >> 2.1. Done.

  >> 2.2. Begin.
             Postprocessing Internal Data Structure.
  >> 2.2. Done.

>> STEP 3.  Write the corresponding SpaceEx Model on file.
  >> 3.1 Begin.
            From the Internal Data Structure --> .xml SpaceEx Model.
  >> 3.1 Done.
  >> 3.2 Begin.
            From the Internal Data Structure --> .xml SpaceEx Model Configuration.
  >> 3.2 Done.
```

Figure: Messages from the tool during the translation phase.

- We obtain:
  1. The SpaceEx model: *SX_DCmotor.xml*
  2. The (not-complete) configuration file: *SX_DCmotor.cfg*

The tool SL2SX: Usage Example

Step 5: Compare the two models

- Open the obtained file by SpaceEx Editor

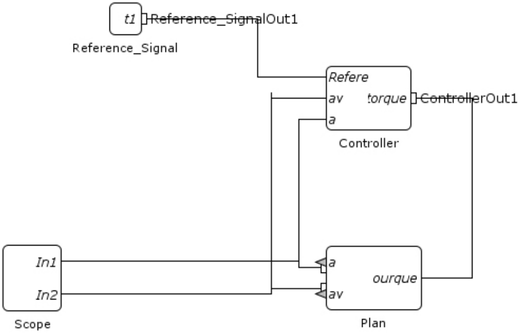
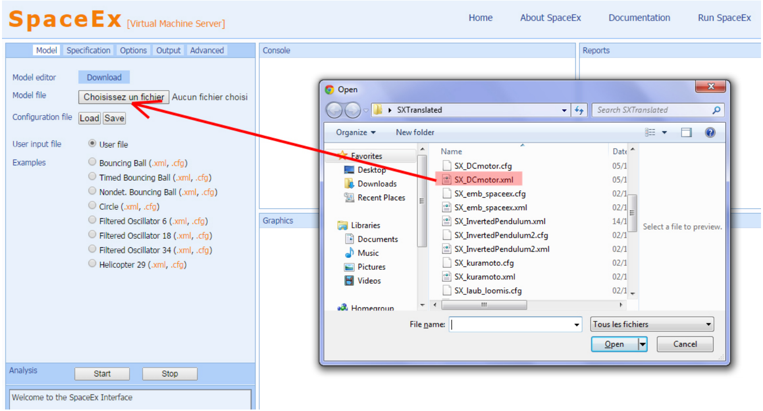


Figure: The resulting SpaceEx Model.

The tool SL2SX: Usage Example

Step 6: Run SpaceEx (Open model file)



Model file: Choisissez un fichier Aucun fichier choisi

User input file: ● User file

Examples:
- ○ Bouncing Ball (.xml, .cfg)
- ○ Timed Bouncing Ball (.xml, .cfg)
- ○ Nondet. Bouncing Ball (.xml, .cfg)
- ○ Circle (.xml, .cfg)
- ○ Filtered Oscillator 6 (.xml, .cfg)
- ○ Filtered Oscillator 18 (.xml, .cfg)
- ○ Filtered Oscillator 34 (.xml, .cfg)
- ○ Helicopter 29 (.xml, .cfg)

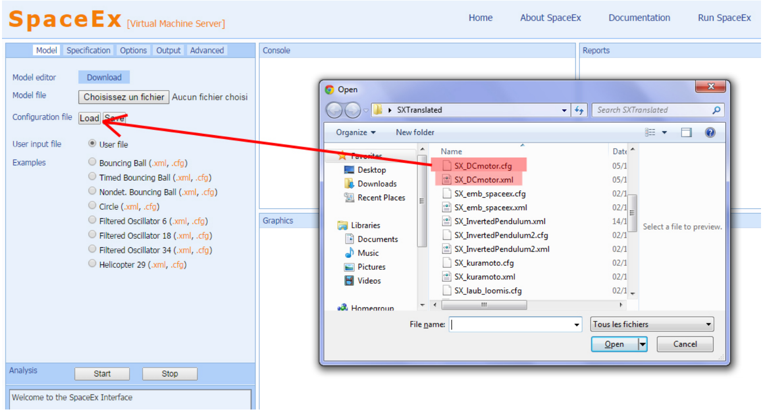
Welcome to the SpaceEx Interface

Name	Date
SX_DCmotor.cfg	05/1
SX_DCmotor.xml	05/1
SX_emb_spaceex.cfg	02/1
SX_emb_spaceex.xml	02/1
SX_InvertedPendulum.xml	14/1
SX_InvertedPendulum2.cfg	02/1
SX_InvertedPendulum2.xml	02/1
SX_kuramoto.cfg	02/1
SX_kuramoto.xml	02/1
SX_laub_loomis.cfg	02/1

Figure: Open the file *SX_DCmotor.xml* by the SpaceEx Web Interface.

The tool SL2SX: Usage Example

Step 7: Run SpaceEx (Open config file)



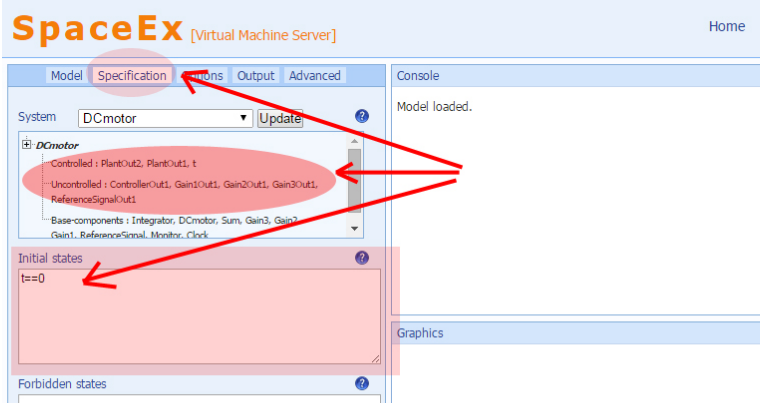
Welcome to the SpaceEx Interface

Name	Date
SX_DCmotor.cfg	05/1
SX_DCmotor.xml	05/1
SX_emb_spaceex.cfg	02/1
SX_emb_spaceex.xml	02/1
SX_InvertedPendulum.xml	14/1
SX_InvertedPendulum2.cfg	02/1
SX_InvertedPendulum2.xml	02/1
SX_kuramoto.cfg	02/1
SX_kuramoto.xml	02/1
SX_laub_loomis.cfg	02/1

Figure: Open the file *SX_DCmotor.cfg* by the SpaceEx Web Interface.

The tool SL2SX: Usage Example

Step 8: complete the initial states



Model loaded.

System DCmotor

DCmotor
- Controlled : PlantOut2, PlantOut1, t
- Uncontrolled : ControllerOut1, Gain1Out1, Gain2Out1, Gain3Out1, ReferenceSignalOut1
- Base-components : Integrator, DCmotor, Sum, Gain3, Gain2, Gain1, ReferenceSignal, Monitor, Clock

Initial states

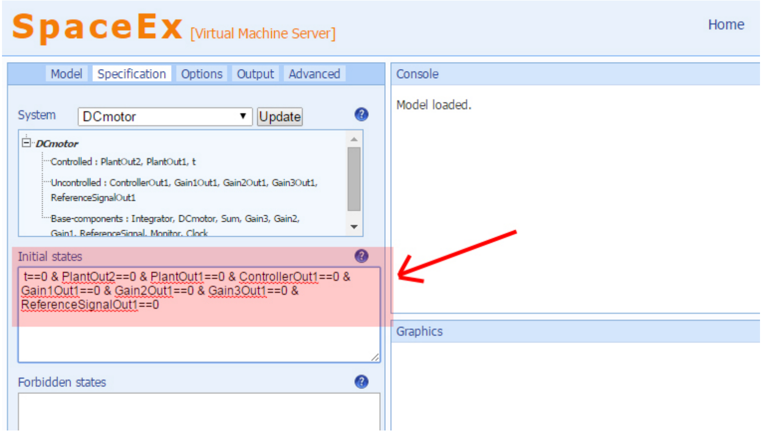
```
t==0
```

Forbidden states

Figure: On SPECIFICATION: set to zero all the Controlled/Uncontrolled Vars.

The tool SL2SX: Usage Example

Step 8: complete the initial states



Model loaded.

System DCmotor

DCmotor
- Controlled : PlantOut2, PlantOut1, t
- Uncontrolled : ControllerOut1, Gain1Out1, Gain2Out1, Gain3Out1, ReferenceSignalOut1
- Base-components : Integrator, DCmotor, Sum, Gain3, Gain2, Gain1, ReferenceSignal, Monitor, Clock

Initial states

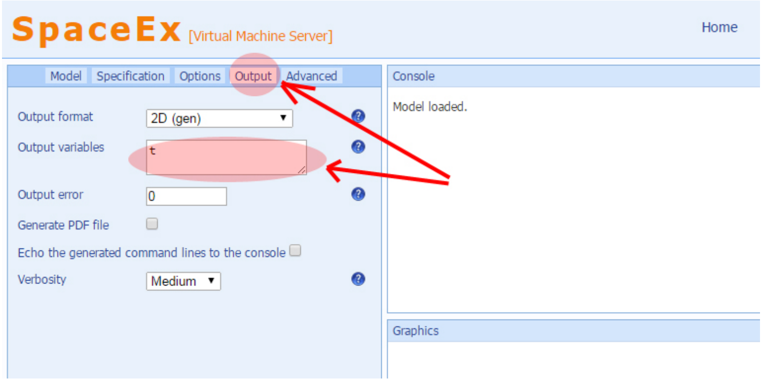
```
t==0 & PlantOut2==0 & PlantOut1==0 & ControllerOut1==0 &
Gain1Out1==0 & Gain2Out1==0 & Gain3Out1==0 &
ReferenceSignalOut1==0
```

Forbidden states

Figure: Initial states: done.

The tool SL2SX: Usage Example

Step 9: choose the output to show



SpaceEx [Virtual Machine Server]

Output format: 2D (gen)

Output variables: t

Output error: 0

Generate PDF file □

Echo the generated command lines to the console □

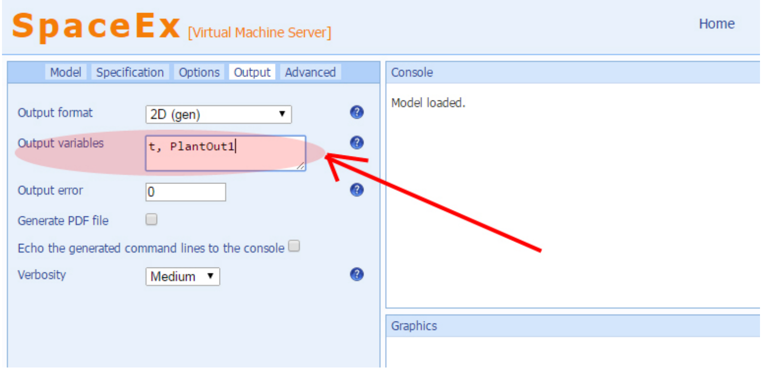
Verbosity: Medium

Model loaded.

Figure: On OUTPUT: We want to see PlantOut1.

The tool SL2SX: Usage Example

Step 9: choose the output to show



Output format: 2D (gen)

Output variables: t, PlantOut1

Output error: 0

Generate PDF file □

Echo the generated command lines to the console □

Verbosity: Medium

Model loaded.

Figure: Output: done.

The tool SL2SX: Usage Example

Step 10: Run the Simulation

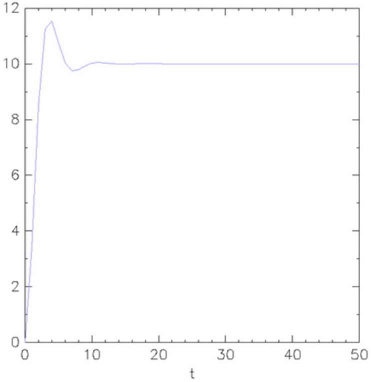


Figure: The resulting SpaceEx Simulation (over 50 seconds).

The tool SL2SX: Usage Example

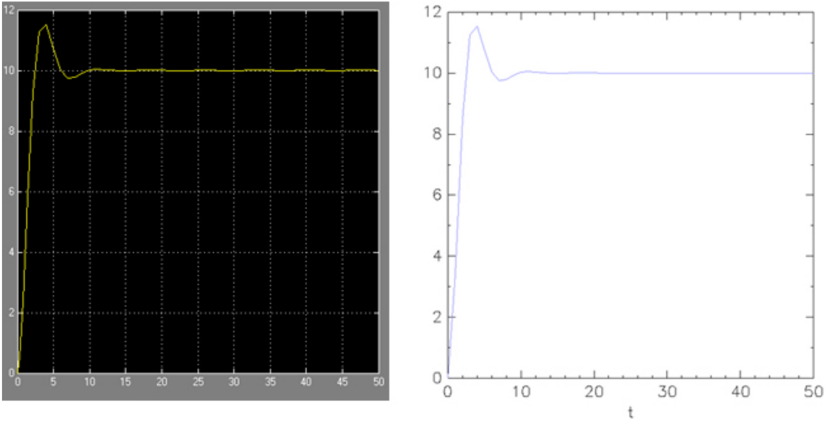
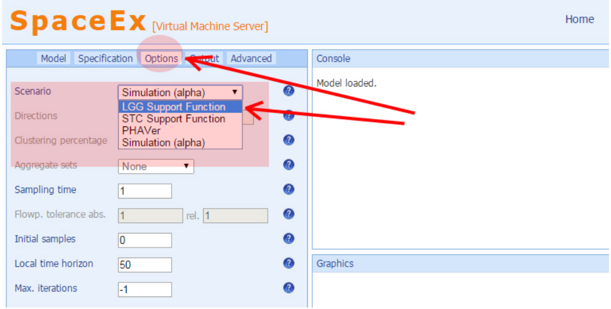


Figure: Comparision. Left: Simulink. Right: SpaceEx

The tool SL2SX: Usage Example

Step 11: Make **Verification**



Scenario: Simulation (alpha)

Directions

Clustering percentage

Aggregate sets: None

Sampling time: 1

Flowp. tolerance abs.: 1 rel.: 1

Initial samples: 0

Local time horizon: 50

Max. iterations: -1

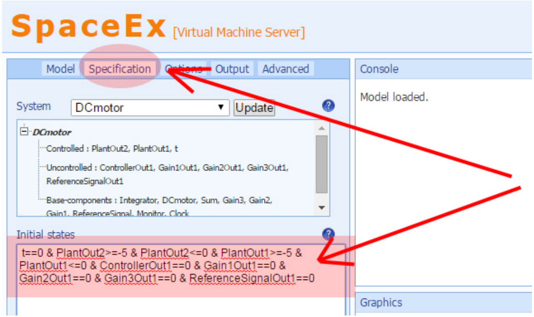
Model loaded.

Figure: On OPTION: choose STC Scenario.

The tool SL2SX: Usage Example

Step 11: Make **Verification**

- Instead of starting from $PlantOut1 == 0$ & $PlantOut2 == 0$ (single point),
- we start from $-5 \leq PlantOut1 \leq 0$ & $-5 \leq PlantOut2 \leq 0$



DCmotor
Controlled : PlantOut2, PlantOut1, t
Uncontrolled : ControllerOut1, Gain1Out1, Gain2Out1, Gain3Out1, ReferenceSignalOut1
Base-components : Integrator, DCmotor, Sum, Gain3, Gain2, Gain1, ReferenceSignal, Monitor, Clock

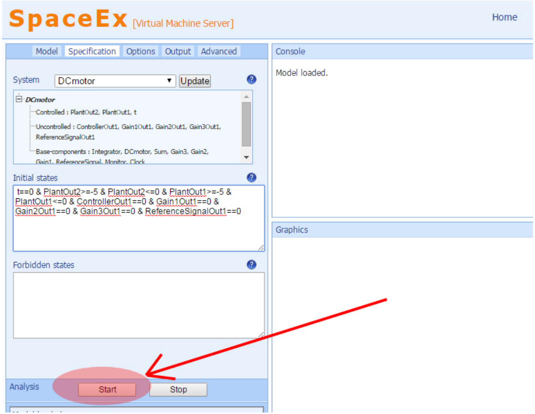
```
t==0 & PlantOut2>=-5 & PlantOut2<=0 & PlantOut1>=-5 &
PlantOut1<=0 & ControllerOut1==0 & Gain1Out1==0 &
Gain2Out1==0 & Gain3Out1==0 & ReferenceSignalOut1==0
```

```
Model loaded.
```

Figure: On SPECIFICATION: choose init states not just as a single point.

The tool SL2SX: Usage Example

Step 12: Run **Verification**



System DCmotor Update

DCmotor
- Controlled : PlantOut2, PlantOut1, t
- Uncontrolled : ControllerOut1, Gain1Out1, Gain2Out1, Gain3Out1, ReferenceSignalOut1
- Base-components : Integrator, DCmotor, Sum, Gain3, Gain2, Gain1, ReferenceSignal, Monitor, Clock

Initial states

```
t==0 & PlantOut2>=-5 & PlantOut2<=0 & PlantOut1>=-5 &
PlantOut1<=0 & ControllerOut1==0 & Gain1Out1==0 &
Gain2Out1==0 & Gain3Out1==0 & ReferenceSignalOut1==0
```

Forbidden states

```
Model loaded.
```

Figure: Press “Start”.

The tool SL2SX: Usage Example

Step 12: Run **Verification**

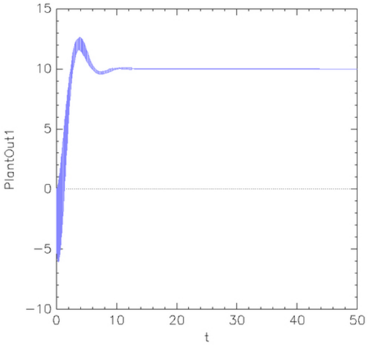


Figure: The result of the Verification.