*COSIM

Purpose: The keyword *COSIM allows LS-DYNA to co-simulate with other software. Currently, co-simulation only works through the functional mock-up interface (FMI).

*COSIM_FMI_CONTROL

*COSIM_FMI_INTERFACE

*COSIM_FMI_CONTROL

Purpose: Define the functional mock-up interface (FMI) co-simulation settings. With this keyword, you will define the co-simulation role of LS-DYNA, such as primary or secondary. You can also specify additional settings, such as IP address/port, functional mock-up unit (FMU) location, and co-simulation time step.

This keyword should be included in the input deck along with *COSIM_FMI_INTERFACE. You also need to download a free plugin called "FMU Manager" to enable this feature. The latest zip file that includes this plugin is located at https://ftp.lstc.com/anonymous/outgoing/isheng/deliver. The zip file also contains detailed documentation and illustrative sample decks.

Card 1	1	2	3	4	5	6	7	8
Variable	APPID		OPT	MODE	FMI			
Type	A20		A10	A10	I			
Default	none		none	none	0			

Settings card. Define additional settings. Repeat this card if necessary. The next keyword ("*") card terminates this input.

Card 2	1	2	3	4	5	6	7	8
Variable	SETTING							
Type	A80							
Default	none							

VARIABLE	DESCRIPTION
APPID	FMU identification. Each FMU must have a unique APPID.
OPT	LS-DYNA's role (see Remark 1): EQ.G: Generation mode. LS-DYNA will generate a new FMU. EQ.C: Co-simulation mode. LS-DYNA will co-simulate with an existing FMU.

VARIABLE	DESCRIPTION
MODE	LS-DYNA's mode (see Remark 1): EQ.P: LS-DYNA is primary, and another software is secondary. EQ.S: LS-DYNA is secondary, and another software is primary.
FMI	Flag to select FMI standard: EQ.1: The generated or co-simulated FMU is based on FMI1.0 standard. EQ.0.OR.2: The generated or co-simulated FMU is based on FMI2.0 standard. This is the default.
SETTINGS	Settings for the generating or using the FMU. Up to 80 characters per line. Multiple lines are allowed for each setting and multiple settings can be defined for each FMU. See Remark 2.

Remarks:

1. **How Co-Simulation Works.** The secondary software generates an FMU, which is imported into the primary software for co-simulation. The primary controls the co-simulation time step, Δt_1, which is different from the explicit time step, Δt_2, in LS-DYNA. Generally, the former should be greater than the latter by several magnitudes, depending on your application ($\Delta t_1 \leq \Delta t_2$ is also allowed). If LS-DYNA is secondary, Δt_1 is set in the primary software. If LS-DYNA is the primary, Δt_1 is set in LS-DYNA by "<time> fmudt=*xxx*" through Card 2 of this keyword as discussed in Remark 2.

   The process of setting up the FMU and running the co-simulation depends on whether LS-DYNA is the primary or secondary:

   a) *LS-DYNA is the secondary.* In prior versions of LS-DYNA this would correspond to LS-DYNA being the *slave*. If you want to run LS-DYNA as the secondary during a co-simulation, usually you need to set OPT = G and run LS-DYNA to generate an FMU. After importing the FMU into another primary software, switch OPT from G to C to run LS-DYNA in the co-simulation mode to co-simulate with the 3rd party software. In this case, you run LS-DYNA twice.

   b) *LS-DYNA is the primary.* In prior versions of LS-DYNA this would correspond to LS-DYNA being the *master*. If you want to run LS-DYNA as the primary during a co-simulation, you usually need to generate an FMU with another software and import the FMU into LS-DYNA for co-simulation. In

this case, set OPT = C and run LS-DYNA only once to connect to the other software for co-simulation.

2. **SETTINGS.** This field is used to setup general settings for the FMU, such as the FMU directory or the co-simulation time. Each setting must start on a new line and begins with <*setting name here*> followed by parameters for the setting. A setting can span multiple lines with each line having a maximum of 80 characters.

   See the table below for a full list of settings in alphabetical order. A description of each setting is below in the Overview of Settings section which is followed by some examples.

Setting Name	Description	OPT	MODE
<fmudir>	Specify FMU directory	C, G	P, S
<fmu_input>	Generate FMU with additional input variables	G	P
<fmu_output>	Generate FMU with additional output variables	G	P
<home>	Specify the direction of the FMU manager	C, G	P, S
<hopsan>	Generate FMU / co-simulate with Hopsan	C, G	S
<name>	Match variables in LS-DYNA and FMU	C, G	P, S
<message>	Specify how often to print out the co-sim messages		
<parameter>	Generate FMU with additional parameter variables	G	P
<socket>	Generate FMU with specified socket properties	G	S
<tcp>	Generate FMU with specified IP and bind LS-DYNA with specified port	C, G	S
<time>	Specify time step and duration of co-simulation	C	P, S

Overview of Settings:

The settings are described in greater detail below. Parameters in italics are for illustration only and indicate the parameters that you will replace for your co-simulation.

<fmudir> ***path***

The setting specifies either where the FMU is located during co-simulation mode or where the FMU is generated in the generation mode. If not included, the directory is assumed to be the folder of the input keyword file. Note the difference of \ and / in Windows or Linux. If the directory is relative to the input file, you should start with ..\ or .\ for Windows, ../ or ./ for Linux. Please avoid any space characters in or after the directory.

<fmu_input> name = ***current*****, type =** ***real*****, value =** ***0***

With this setting, you can define additional FMU input variables during generation mode. Currently only variables of **type=*real*** are supported. The initial value is set with **value**. LS-DYNA must be in the primary mode for this setting to work, meaning MODE = P and OPT = G.

<fmu_output> name=*voltage***, type=***real***, value=***0***

With this setting, you can define additional FMU output variables during generation mode. Currently only variables of **type=*real*** are supported. The initial value is set with **value**. LS-DYNA must be in the primary mode for this setting to work, meaning MODE = P and OPT = G.

<home> ***path***

This setting specifies the directory of the FMU manager kit. If not included, the directory is assumed to be the parent folder of the keyword input file, meaning ..\. You can also specify a directory relative to the input file, starting with ..\ or .\ for Windows and ../ or ./ for Linux. Please avoid any space character in or after the directory.

<hopsan> name = ***port1*****, type =** ***hydraulic_Q*****, segment =** ***1*****, node =** ***1867*****, dir = +***z*

You specify this setting when you want to co-simulate with piston models in Hopsan. Here, **name** is the port number, **type** is the piston type, **segment** is the piston segment set ID to which the hydraulic force is applied, and **node** can be any node on the piston where piston velocity is picked up. The parameter **dir** gives the direction of the piston motion. The positive direction is the extension while the negative direction is the retraction.

<message> step = ***100***

This setting specifies how often co-simulation messages are output to fmu_msg. They are output every ***step*** co-simulation time steps. If not specified, LS-DYNA will print out the following message every 100 co-simulation time steps:

FMU time: 0.000000e+00 cosim-dt: 6.644e-05 cosim-step: 1

FMU time: 6.471417e-03 cosim-dt: 6.258e-05 cosim-step: 101

<name> ***DX500 = In1***

This setting is used to match variables in LS-DYNA and the FMU during the generation or co-simulation mode. The above example tells LS-DYNA that *In1* in FMU is equivalent to the *X* displacement of node 500 (additional cases are documented in the co-simulation examples manual). During co-simulation, LS-DYNA will match the variables on both sides of "=" to send out or receive data from the FMU. The variable on the right side can be found inside the modelDescription.xml of the FMU. The variable on the left side corresponds to the LS-DYNA local variables with its name being the combination of the value of FIELD with the node number; see *COSIM_FMI_INTERFACE.

<parameter> name=*PCOEF1*, type=*real*, value=*1*

This setting is used to define parameter variables for an FMU to be generated. Currently, only variables of **type=*real*** are supported. The parameter **value** defines the initial value for the parameter. LS-DYNA must be in the primary mode for this setting to work, that is, MODE = P and OPT = G. Usually, you need to modify function `getReal()` in template/fmi_udf.c for such application to generate the needed FMU.

<socket> nconnect=*20*, tdelay=*1000*, recvtimeout=*0*

This setting is used to define the socket properties of an FMU to be generated. LS-DYNA must be in the secondary mode for this item to work, meaning MODE = S and OPT= G. At the beginning of the co-simulation, the other software will try **nconnect** times at most to reach the LS-DYNA computer for connection. If the socket communication to the IP designated with **<tcp>** is not successful, the co-simulation fails. **tdelay** is the time delay in milliseconds for the other software to reconnect to LS-DYNA if the previous connection fails. **recvtimeout** is the max time in milliseconds that FMU waits for data from LS-DYNA, and beyond this time limit, the co-simulation fails due to lost data. If **recvtimeout=0**, the FMU will wait infinitely for LS-DYNA.

<tcp> ip=*127.0.0.1*, port=*39400*

This setting is used to define the FMU properties to be generated. LS-DYNA must be in secondary mode (MODE = S) for this setting to work. The parameter **ip** is the IP address of the LS-DYNA computer in the co-simulation. If the co-simulation is local, you can always set **ip** to be 127.0.0.1. You can also specify a .txt file that includes the IP address in the first line instead of the IP address with **ip=*ip_addr.txt***. If this setting is not included, the default IP address is 127.0.0.1 with port 39400.

<time> tstart = *0*, tend = *1.2*, option = *0*, fmudt = *0.0005*, *tscale = 10*

This setting defines the LS-DYNA properties during co-simulation. When tstart < t < tend, the co-simulation is occurring. The parameter **option** controls the value of exported/imported variables in LS-DYNA when t > tend. When option = 0, the exported/imported variables are set to 0 when t > tend; when option = 1, the exported/imported variables for t > tend are held constant with the value from t = tend. When t < tstart, the imported variables are set to 0 and the exported variables are set to their initial values, which are specified by VINIT in *COSIM_FMI_INTERFACE.

The parameter **fmudt** defines the co-simulation time step between LS-DYNA and the FMU, and it only applies when LS-DYNA is the primary during the co-simulation. The co-simulation time step is different from the explicit time step of LS-DYNA. Generally, the former is usually greater by several magnitudes, however, values of smaller or comparable magnitude are also allowed. Between co-simulation time steps, the FMU variables are held constant. If this setting is not defined, tstart = 0, tend = 10^6, option = 0, fmudt = 100 × tstep, where tstep is the LS-DYNA explicit time step.

For cases when the FMU and LS-DYNA have different time scales, use **tscale** to specify the scale factor between the time scales for FMU and LS-DYNA. The default value is 1 without specification. For instance, if the MATLAB side is based on seconds and LS-DYNA is based on milliseconds, set **tscale** = 1000. In this case, when MATLAB proceeds by 1 s in co-simulation, LS-DYNA will proceed by 1000 ms. Note that the values of tstart, tend and fmudt are based on the LS-DYNA time unit.

Examples:

The following examples only serve as an illustration of syntax.

Example 1

In this example, LS-DYNA is the secondary and generates the FMU. The FMU is then imported into another primary software. You can switch G to C and run LS-DYNA to connect to the primary software.

```
*COSIMULATION_FMI_CONTROL
               MOTOR         G         S
<home> C:\DYNA_bin\FMU_Manager_v4_deliver
<fmudir> C:\DYNA_bin\FMU_Manager_v4_deliver\dyna_key
<tcp> ip=127.0.0.1,port=39400
<socket> nconnect=20,tdelay=1000,recvtimeout=0
<name> DX500=input1
<time> tstart=0, tend=0.2, option=1
```

Example 2

Here LS-DYNA is the primary during co-simulation. You generate an FMU with another secondary software and import it into LS-DYNA for co-simulation. In the settings, you set the co-simulation time step and match variables in LS-DYNA and FMU.

```
*COSIMULATION_FMI_CONTROL
                PID1         C         P
<home> ..\
<fmudir> .\
<time> fmudt=0.005
<name> DX500=disp_in1, VX500=veloc_in2
<name> FX500=force_out1
```

*COSIM_FMI_INTERFACE

Purpose: Define the interface variables during co-simulation. For instance, you may need LS-DYNA to export certain node displacements to another software and to import certain nodal forces.

This keyword should be included in an input deck with *COSIM_FMI_CONTROL. *You also need to download a free plugin called "FMU Manager" to enable this feature.*

This feature is available as of R12.

ID card. Specify the FMU, where the interface variables are imported or exported.

Card 1	1	2	3	4	5	6	7	8
Variable	APPID							
Type	A20							
Default	none							

Variable card. Specify the interface variables to be imported and exported. Repeat this card if necessary. This input terminates with the next keyword ("*") card.

Card 2	1	2	3	4	5	6	7	8
Variable	IMPEXP	REGTYP	REGID	FIELD	VINIT	RATIO	CID	REF
Type	A	A	I	A	F	F	I	I
Default	none	none	none	none	0	1.0	0	0

VARIABLE	DESCRIPTION
APPID	FMU (functional mock-up unit) identification. Each FMU must have a unique APPID.
IMPEXP	Import/export flag: EQ.IMP: Variables are to be imported into LS-DYNA. EQ.EXP: Variables are to be exported from LS-DYNA.

VARIABLE	DESCRIPTION
REGTYP	Type of interface region: EQ.NODE: Single node EQ.NSET: Node set EQ.SSET: Segment set EQ.PART: Rigid part EQ.FUNC: User defined curve function to be exported only. See Remark 1 and Example 2. EQ.CURV: User defined curve to be imported only. If the curve is to be imported for *ICFD (e.g., boundary condition), its field must start with 'ICFD_'. See Remark 2 and Example 3. EQ.SESW: Sense switch to be imported only. REGID is neglected with the sense switch specified in FIELD. See Remark 3 and Example 4. EQ.BAG: Control-volume airbag.
REGID	ID of the corresponding region type in LS-DYNA. If region ID is negative for a node set and FIELD = FX, FY or FZ, all nodes within this set share the same imported force value from the FMU. Otherwise, a distinct force value will be imported for each node.
FIELD	Field data to be imported or exported, depending on the region type. See Remark 4 for a full list of fields.
VINIT	Initial value. For variables to be exported, VINIT is the initial value for the FMU. Therefore, its units must be the same as the FMU's units.
RATIO	Scale factor applied during co-simulation (not FMU generation). When LS-DYNA exports variables, the actual value to be sent is RATIO × LS-DYNA value. When LS-DYNA imports data from FMU, the actual value received is the FMU value / RATIO.
CID	Coordinate system ID (see *DEFINE_COORDINATE). EQ.0: Global (default)
REF	Control how the coordinate system is used for the variable output when CID > 0 (see Remark 5):

VARIABLE	DESCRIPTION
	EQ.0: Variable output is in the local system fixed from the beginning. Note that you should set FLAG = 0 in *DEFINE_COORDINATE_NODES.
	EQ.1: Variable output is projected onto the moving local system.
	EQ.2: Variable output is the projection of the nodal translation motion relative to node N1 of the local coordinate system, COOR. Double precision LS-DYNA is recommended with REF = 2.

Remarks:

1. **REGTYP = "FUNC".** You can export variables from *DEFINE_CURVE_FUNCTION with the corresponding curve function ID. Note that it is for export only. For example, if you want to export the value of SENSOR(*n*) through the FMU, you must do the following three steps (see Example 2):

   a) Set up sensors through *SENSOR.

   b) Include the sensor in *DEFINE_CURVE_FUNCTION.

   c) Add the curve function to *COSIM_FMI_INTERFACE.

2. **REGTYP = "CURV".** You can import general variables through *DEFINE_CURVE. During co-simulation, the curve value in LS-DYNA will be dynamically updated. Note that it is for import only. For instance, if you want to import velocity from the controller in MATLAB and impose it onto a node in LS-DYNA, you must do the following three steps (see Example 3):

   a) Define a curve with initial values of all zero and the time range greater than the termination time of the co-simulation

   b) Set up *BOUNDARY with the defined curve.

   c) Add the curve to *COSIM_FMI_INTERFACE.

3. **REGTYP = "SESW".** If you want to import a sense switch (see Sense Switch Controls in Getting Started) ranging from "SW1~SW4" and "SWA~SWD" to control the status of LS-DYNA, you must specify the switch name in FIELD. All other fields, namely, REGID, VINIT, RATIO, CID, and REF, are neglected. The rising edge of the imported signal (the transition from a low to high value) will trigger the corresponding sense switch in LS-DYNA. If multiple switches are defined, they cannot be triggered at the same time. See Example 4.

4. **FIELD.** See the table below for a detailed list of possible values for FIELD. Not that PART must be for a rigid part.

FIELD	Description	REGTYP	Export	Import	REF
CX, CY, CZ	Position	NODE, NSET, PART	✓		0, 1
DX, DY, DZ	Translational displacement in *X*, *Y*, and *Z*, respectively	NODE, NSET, PART	✓		0, 1, 2
VX, VY, VZ	Translational velocity in *X*, *Y*, and *Z*, respectively	NODE, NSET, PART	✓		0, 1, 2
ACCX, ACCY, ACCZ	Translational acceleration in *X*, *Y*, and *Z*, respectively	NODE, NSET, PART	✓		0, 1, 2
AX, AY, AZ	Angular displacement in *X*, *Y*, and *Z*, respectively	PART	✓		0, 1
WX, WY, WZ	Angular velocity in *X*, *Y*, and *Z*, respectively	PART	✓		0, 1
WDTX, WDTY, WDTZ	Angular acceleration in *X*, *Y*, and *Z*, respectively	PART	✓		0, 1
BAGP, BAGV, BAGT	Airbag pressure, volume and temperature	BAG	✓		N/A
FX, FY, FZ	Force in *X*, *Y*, and *Z*, respectively	NODE, NSET, PART		✓	0, 1
P	Pressure	SSET		✓	N/A
TX, TY, TZ	Torque	PART		✓	0, 1
Variable name in FMU	Defined curve function	FUNC	✓		N/A
Variable name in FMU	Defined curve	CURV		✓	N/A

FIELD	Description	REGTYP	Export	Import	REF
SW1-SW4, SWA-SWD	Sense switch	SESW		✓	N/A

5. **REF.** The following are general comments about REF:

   a) REF = 1 and 2 differ in that variables with REF = 2 are measured relative to the changing local system.

   b) You cannot set REF to a different value for the same coordinate. For instance, you cannot export DX for node 1 based on CID = 1 with REF = 1, and then export DY for node 2 based on CID = 1 with REF = 2. To avoid this, you can define a new coordinate with a different REF.

   c) If you also define *DATABASE_HISTORY_NODE_LOCAL, please make sure the REF of the corresponding coordinate system is consistent, meaning you cannot set REF = 1 for CID = 1 in *COSIM_FMI_INTERFACE and REF = 2 for the same CID in *DATABASE_HISTORY_NODE_LOCAL.

Examples:

The following examples only serve as an illustration of syntax.

Example 1

This example illustrates exporting the velocity in the *X*-direction for node 100, exporting the displacements in the *Y*-direction for all nodes within set 3, importing the forces along the *X*-direction for all nodes in set 3, importing the same force along the *Y*-direction for all nodes in set 4, and importing the pressure in segment set 1.

```
*COSIMULATION_FMI_INTERFACE
              MOTOR
       EXP      NODE       100        VX         0         1         0         0
       EXP      NSET         3        DY         0         1         0         0
       IMP      NSET         3        FX         0         1         0         0
       IMP      NSET        -4        FY         0         1         0         0
       IMP      SSET         1         P         0         1         0         0
```

Example 2

Export the value of sensor 1 through a defined curve function. The corresponding variable is named "fyball" in FMU during co-simulation.

```
*SENSOR_DEFINE_NODE
         1       681      1107Y                  0COORD
*DEFINE_CURVE_FUNCTION_TITLE
diff_Y
       100         0
SENSORD(1)
*COSIMULATION_FMI_INTERFACE
```

```
                ball
       EXP      FUNC       100    fyball          0          1          0          0
```

Example 3

Import the Y-direction velocity value through a defined curve which is then prescribed to a node in LS-DYNA during co-simulation.

```
*DEFINE_CURVE_TITLE
vy
       200          0       1.0       1.0       0.0       0.0          0          0
                  0.0                 0.0
                100.0                 0.0
*BOUNDARY_PRESCRIBED_MOTION_NODE_ID
         2ball_vy
       681          2          0       200       1.0         01.00000E28       0.0
*COSIMULATION_FMI_INTERFACE
                ball
       IMP      CURV       200    vyball          0          1          0          0
```

Example 4

Import a sense switch during co-simulation.

```
*COSIMULATION_FMI_INTERFACE
$#             appid
              switch
$#  impexp    regtyp     regid     field      init     ratio      coor       ref
       IMP      SESW                 SW1          0          1          0          0
```