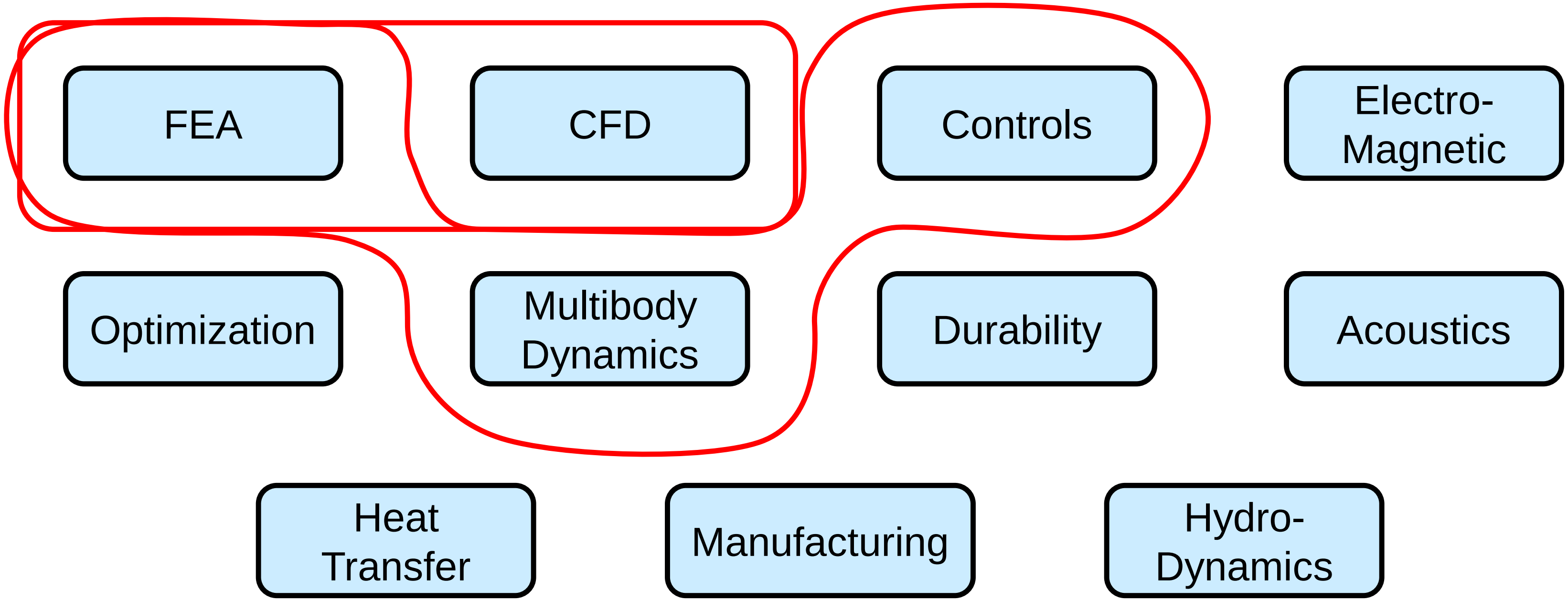
Multi-Physics RecurDyn Style – Interoperability

Brant Ross

1. System-Level Simulation

1. Multi-disciplinary Simulation

- Multiphysics = Any combination of CAE
- Multiphysics for Fluid – Structure Interaction
- Multiphysics for Assemblies in Motion

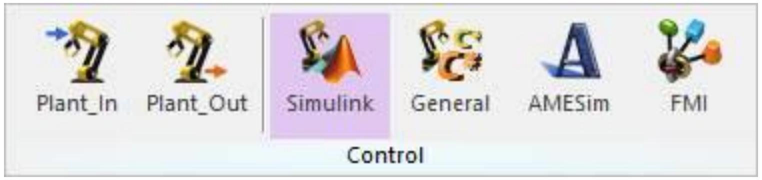


2. Co-Simulation Options: Controls/Hydraulics

- Capabilities found in RecurDyn Help: Communicator, Control, Co-simulation
- 4 Categories:
  - Matlab/Simulink – Control system modeling
  - User Program – General programming interface
  - AMESim - Hydraulics
  - FMI – Functional Mock-up Interface

Note: only 1 control interface per RecurDyn model!

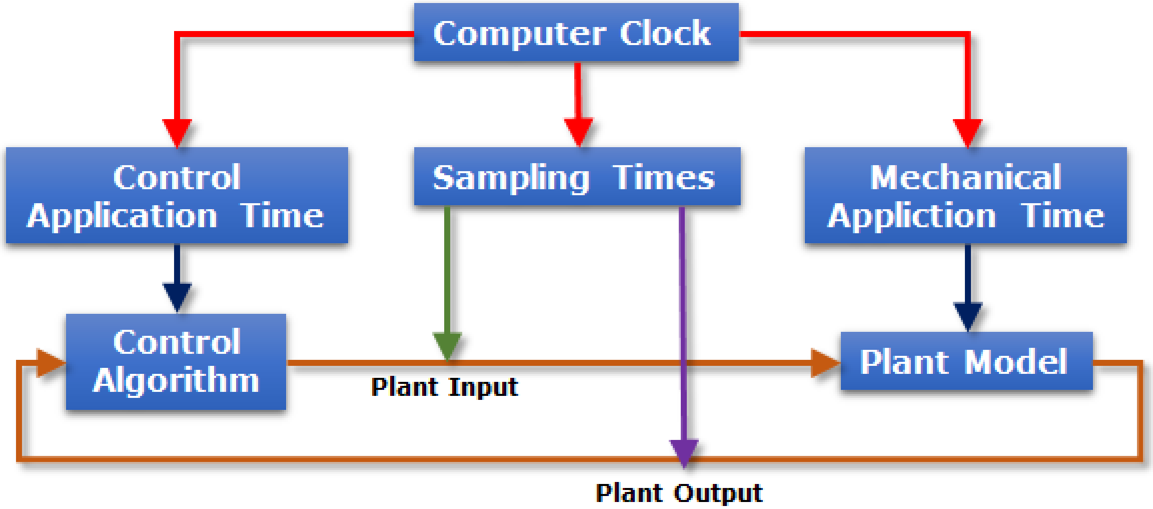
2. Co-Simulation Options: Common Controls Concepts



- Plant Inputs

- Plant Outputs

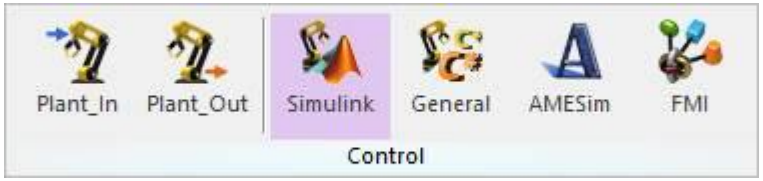
- Host

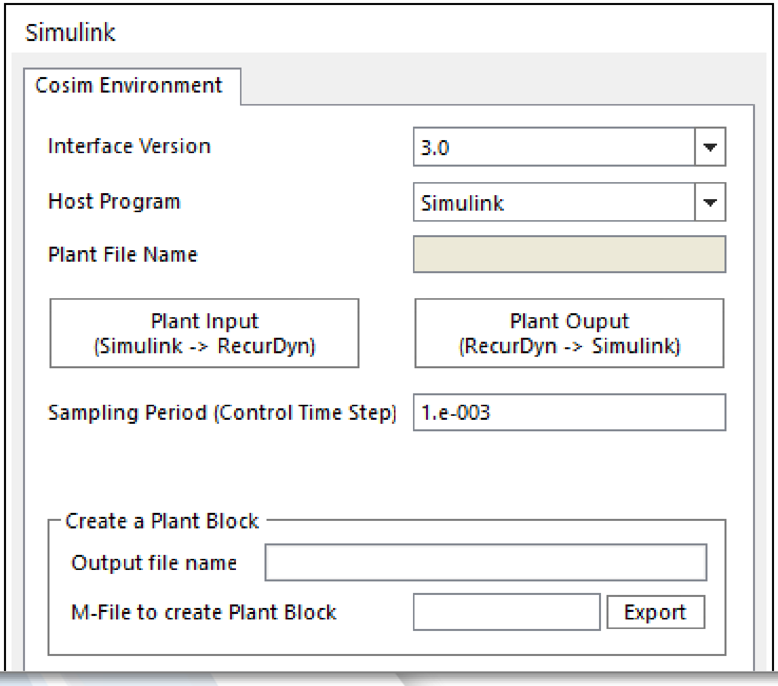


Note: only 1 control interface per RecurDyn model!

2. Co-Simulation: Matlab/Simulink Controls Simulation

- Capabilities found in RecurDyn Help: Communicator, Control, Co-simulation, Co-simulation with Simulink
- Access to the module under the RecurDyn Communicator ribbon:





Interface Version: 3.0

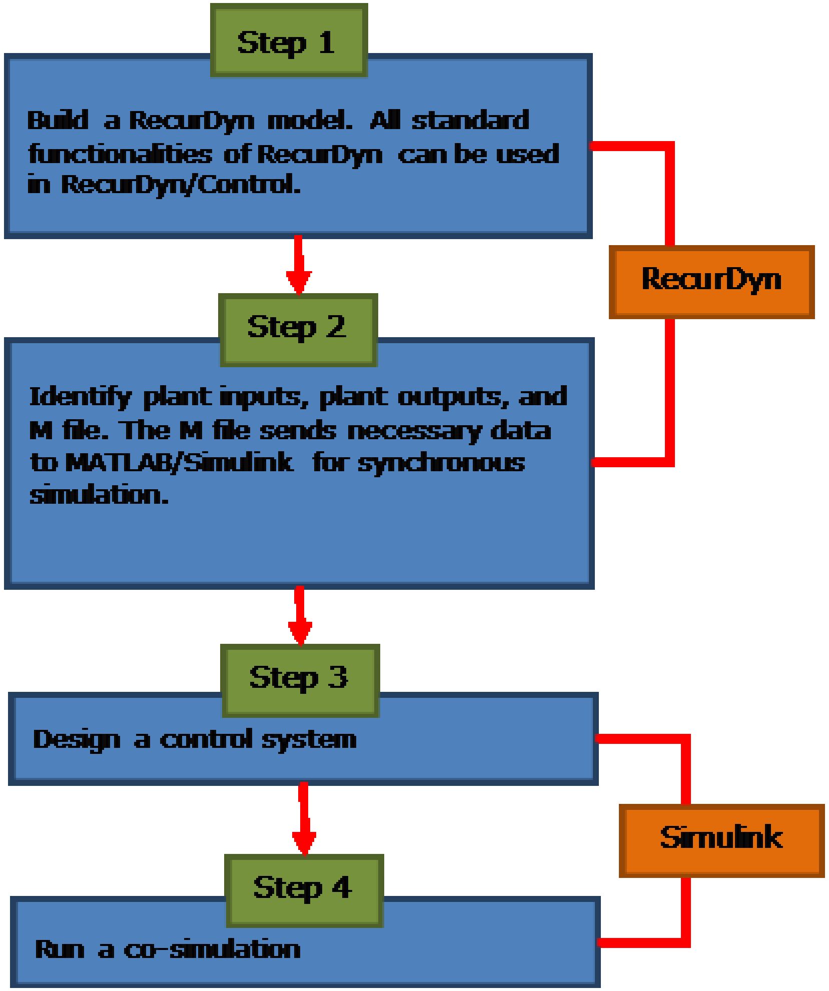
Host Program: Simulink

Plant File Name:

Sampling Period (Control Time Step): 1.e-003

Output file name:

M-File to create Plant Block:



Step 1: Build a RecurDyn model. All standard functionalities of RecurDyn can be used in RecurDyn/Control.

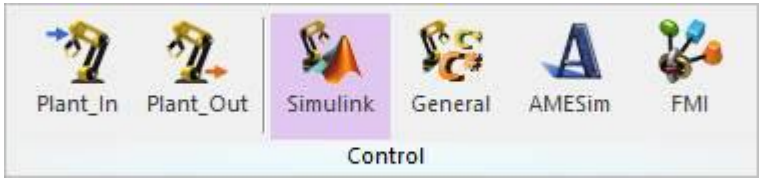
Step 2: Identify plant inputs, plant outputs, and M file. The M file sends necessary data to MATLAB/Simulink for synchronous simulation.

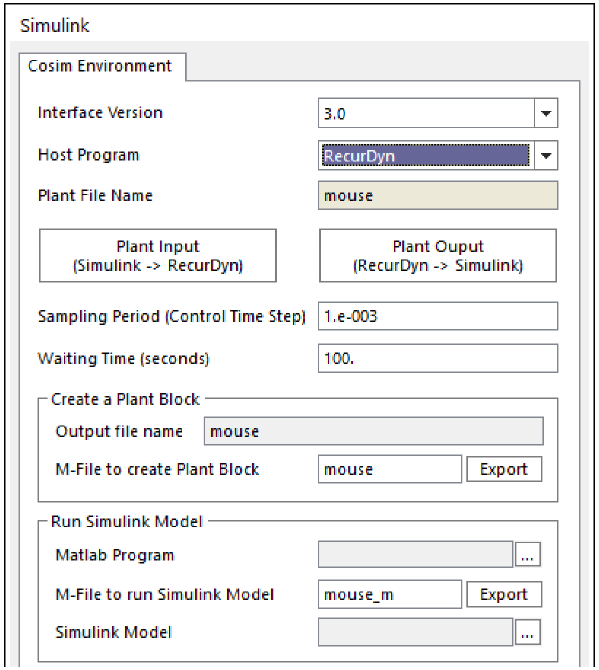
Step 3: Design a control system

Step 4: Run a co-simulation

2. Co-Simulation: Matlab/Simulink Controls Simulation

- Capabilities found in RecurDyn Help: Communicator, Control, Co-simulation, Co-simulation with Simulink
- Access to the module under the RecurDyn Communicator ribbon:




Simulink

Cosim Environment

Interface Version	3.0
Host Program	RecurDyn
Plant File Name	mouse

Plant Input (Simulink -> RecurDyn)

Plant Ouput (RecurDyn -> Simulink)

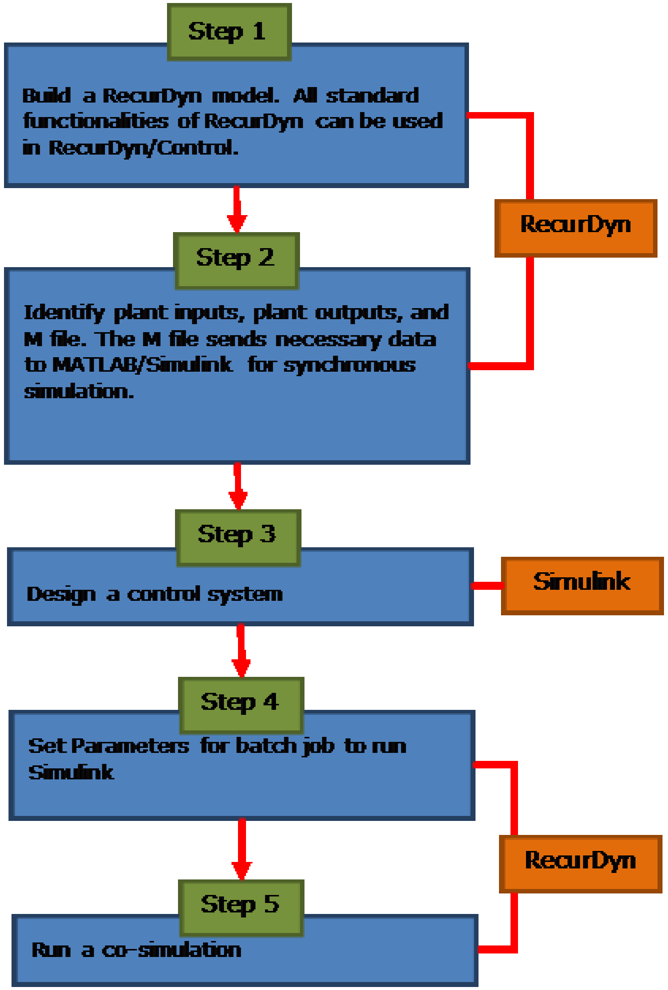
Sampling Period (Control Time Step)	1.e-003
Waiting Time (seconds)	100.

Create a Plant Block

Output file name	mouse
M-File to create Plant Block	mouse Export

Run Simulink Model

Matlab Program	...
M-File to run Simulink Model	mouse_m Export
Simulink Model	...


Step 1

Build a RecurDyn model. All standard functionalities of RecurDyn can be used in RecurDyn/Control.

Step 2

Identify plant inputs, plant outputs, and M file. The M file sends necessary data to MATLAB/Simulink for synchronous simulation.

RecurDyn

Step 3

Design a control system

Simulink

Step 4

Set Parameters for batch job to run Simulink

Step 5

Run a co-simulation

RecurDyn

2. Co-Simulation: Matlab/Simulink Controls Simulation

1. Build and Verify the RecurDyn mechanical system model
   a) Any RecurDyn Model can be used in this control toolkit interface.
   b) If plant inputs (PIN) and plant outputs (POUT) are not defined, only the mechanical system model is simulated in RecurDyn.
   c) Prior to applying a control system, make sure that the model runs without errors. Errors not corrected early are difficult to find/fix in co-simulation.

2a. Identify Plant Inputs

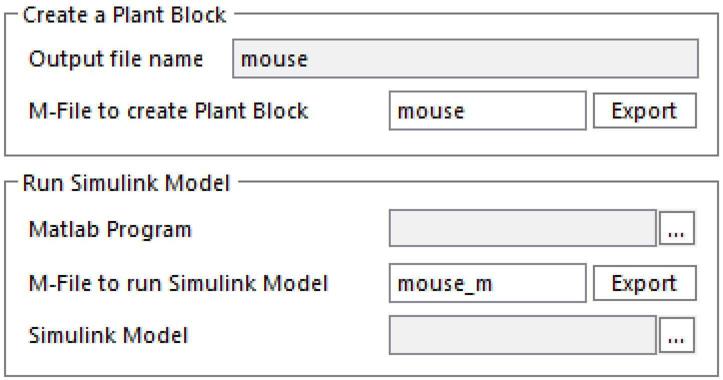
   a) Define plant inputs and their variable names in the given list. For more information, click here.

2b. Identify Plant Outputs

   a) Create the plant outputs and define them using the general Expression function. For more information, click here.

2c. Set interface information

   a) Select the Host Program as RecurDyn.
   b) Enter the name of the M file to create a Plant Block and click Export.



Create a Plant Block

Output file name: mouse

M-File to create Plant Block: mouse

Run Simulink Model

Matlab Program:

M-File to run Simulink Model: mouse_m

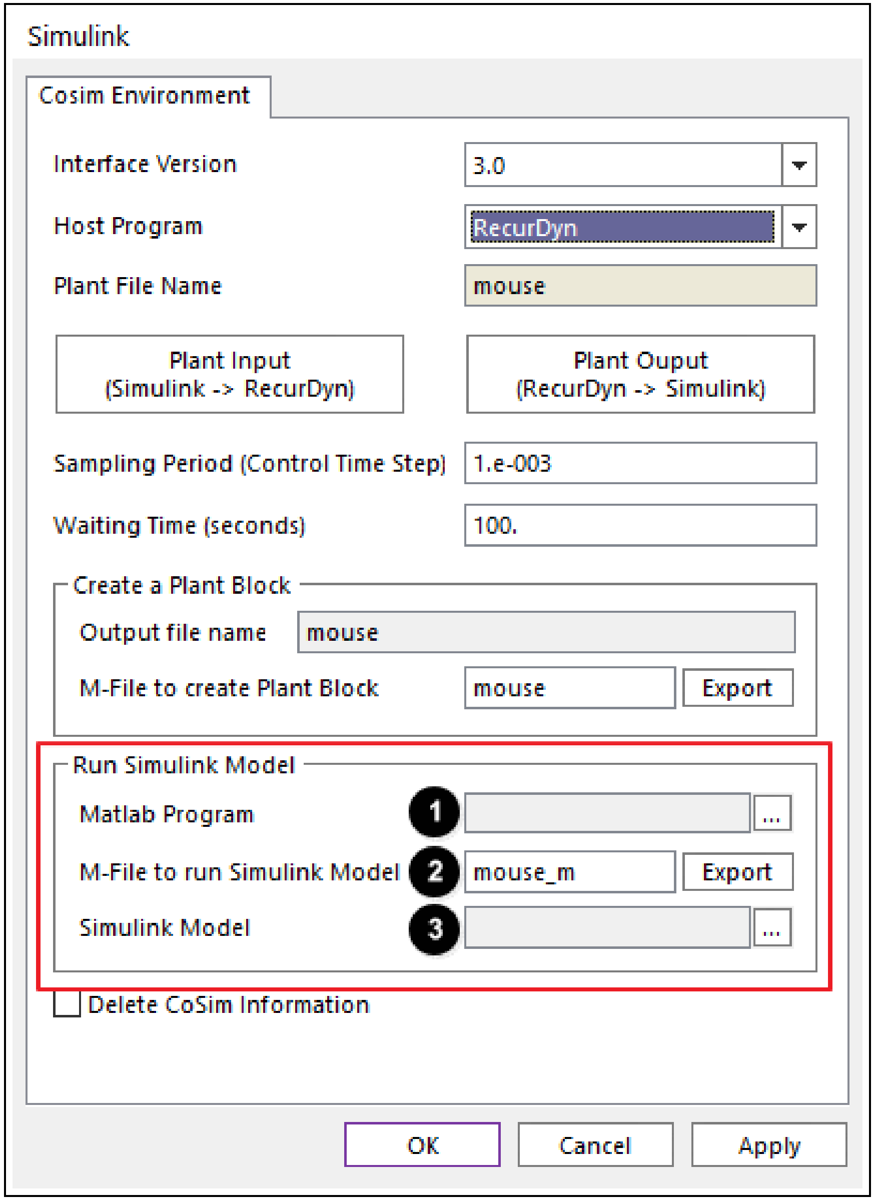
Simulink Model:

2. Co-Simulation: Matlab/Simulink Controls Simulation

3. Build Simulink model with RecurDyn Plant Block
   a) Run the MATLAB program.
   b) Change the working directory to the directory that has the M file and the RecurDyn model.
   c) In the MATLAB command window, enter the name of M file.
   d) In the MATLAB command window, enter “makerd’.
   e) A new Simulink window with RecurDyn Plant Block appears.
   f) Open a new Simulink model window.
   g) Drag and drop RecurDyn Plant Block on the new Simulink model window.
   h) Design the control system with the pasted plant.
   i) Terminate MATLAB program.

2. Co-Simulation: Matlab/Simulink Controls Simulation

4. Set Parameters for Batch Job of Simulink
   1) Enter the executable file name and the installation directory of the MATLAB program.
   2) Enter the name of Simulink model that is built for the co-simulation with RecurDyn.
   3) Enter the name of M file to run Simulink model to be used for the batch job of Simulink model and click Export.



Simulink

Cosim Environment

Interface Version: 3.0

Host Program: RecurDyn

Plant File Name: mouse

Sampling Period (Control Time Step): 1.e-003

Waiting Time (seconds): 100.

Create a Plant Block

Output file name: mouse

M-File to create Plant Block: mouse

Run Simulink Model

Matlab Program:

M-File to run Simulink Model: mouse_m

Simulink Model:

□ Delete CoSim Information

RECURDYN

2. Co-Simulation: Matlab/Simulink Controls Simulation

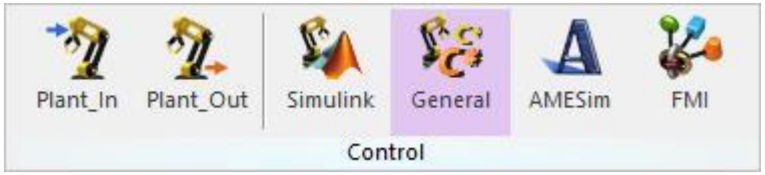
5. Co-Simulation
   - Run the dynamic analysis of RecurDyn. RecurDyn automatically executes MATLAB program.

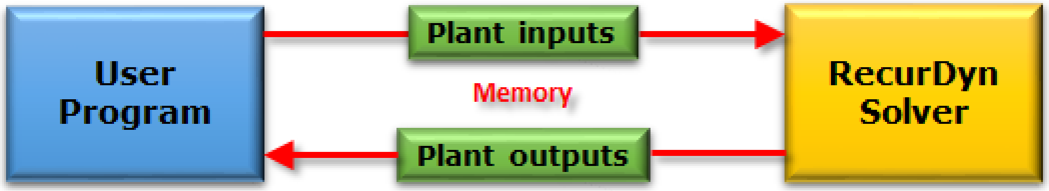
Notes:

1. Only one RecurDyn Plant Block should be used in the Simulink model.
2. Multitasking is not supported.
3. RecurDyn model and Simulink model should be free of errors
4. If MATLAB does not respond for the long time, interrupt the MATLAB program using the Windows Task Manager and then fix the errors in the RecurDyn model.
5. The independent subsystem plant model created by RecurDyn host mode supports the co-simulation of the subsystem plant model.

2. Co-Simulation Options: Controls/Hydraulics

- General programming interface
  - Either RecurDyn or the User Program can be the host

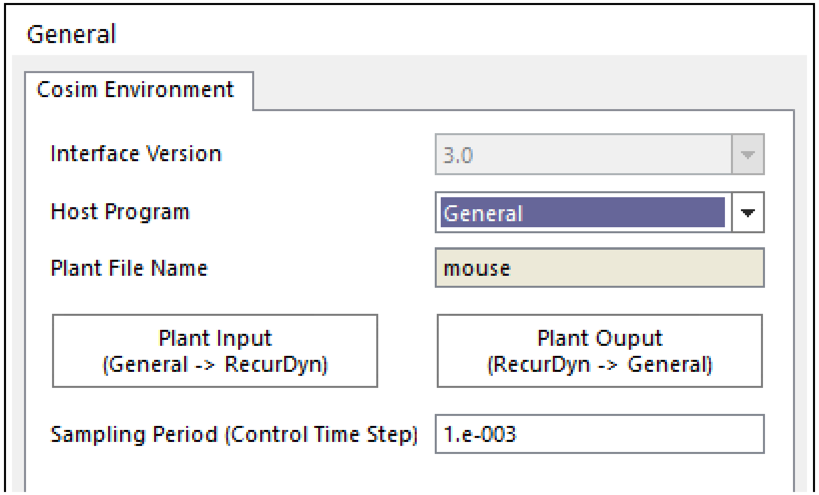




  - Similar to Simulink Interface
    - Plant Inputs
    - Plant Outputs
    - Sequence of steps

2. Co-Simulation Options: Controls/Hydraulics

- General programming interface
  - With the User Program as host
  - If the user wants to open and use User-defined Program and RecurDyn at the same time, RecurDyn recommends that the user uses the following step.



General

Cosim Environment

Interface Version: 3.0

Host Program: General

Plant File Name: mouse

Sampling Period (Control Time Step): 1.e-003

  - Library files:
    - <Install Dir>/Toolkits/Controls/General10/CoSim_Operator.lib
    - CoSim_Operator.dll,
    - CoSimGeneral.h

2. Co-Simulation Options: Controls/Hydraulics

- General programming interface (user program as host)
  - Interface Functions
    - **RunSimulation** (wchar_t *ModelName, double endtime, int plotstep, double interfacestep);   // called only once before interfacing
    - **CommunicateRecurDyn** (double sim_time, int iNPin, int iNPout, double *dPIN, double *dPOUT, double *rd_time, int *error);

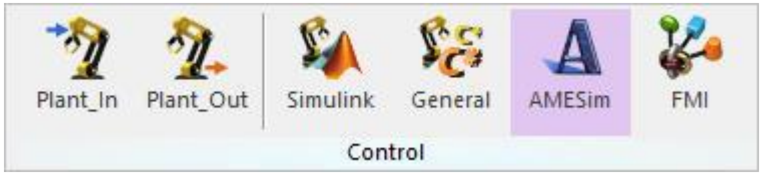
      // called at each interface time step
    - **GetStatusRecurDyn** (StatusRecurDyn* statusRD, int* error);

      // called at each interface time step
    - **GetStringPlantInput** (int seq, char* chDirectory, int* error);
    - **GetStringPlantOutput** (int seq, char* chDirectory, int* error);
    - **CloseRecurDyn**();   // called only once after interfacing.
    - **StopRecurDyn**();   // called only once after interfacing.

Check out the example at Communicator > Control > Sample Co-simulation with General Method!

2. Co-Simulation Options: Controls/Hydraulics

- AMESim – Hydraulics, etc.



- The AMESim interface allows you to link a RecurDyn model with an AMESim model. By coupling the motion simulation and system simulation, this interface improves the accuracy of your full system simulation. This interface is:
  - Useful when you want to build the multibody that interacts with hydraulic or pneumatic fluid power systems or other Amesim systems.
  - Designed so that you can continue to use many of the Amesim facilities while the model is running in RecurDyn.

2. Co-Simulation Options: Controls/Hydraulics

- AMESim – Hydraulics, etc.

Steps to design the hydraulic system

1. Run the AMESim program.
2. Create an Interface block, as following function: “Menu>Modeling>Interface block>Create interface icon…”.
3. Set the Interface icon
   - Select the interface type to be RecurDynCosim.
   - Set the number of inputs and outputs.
4. Build a hydraulic system model.

2. Co-Simulation Options: Controls/Hydraulics

- AMESim – Hydraulics, etc.

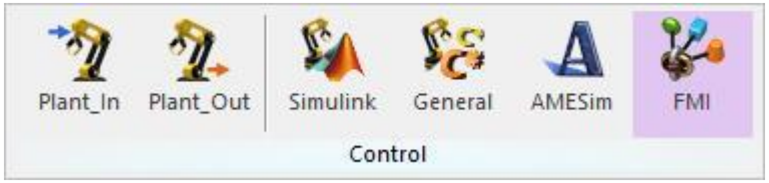
Compile the AMESim model

1. Select “Tools > Options > Amesim Preferences > Compilation“. And then, change the compiler as following dialog box.
   - Note: The compilers should be installed additionally.
   - Note: It is needed to execute “Modeling > Check submodels…” for the Amesim model made in the old version of Amesim.
2. Changes the mode into Parameter mode, the DLL file are generated in the same folder of Amesim model.

Notes:

- When the user closes Amesim, the “model_name_.dll” file disappears. So, do not close Amesim during the RecurDyn simulation.
- If the user wants to extract “model_name_.dll”, use the AMELoad.exe installed in the “win32” directory of the Amesim installation.
- If the user’s system name is “test1.ame”, type “AMELoad test1” in a CMD window in the directory where the Amesim model is located

2. Co-Simulation Options: Controls/Hydraulics



FMI – Functional Mock-up Interface

- The FMI specifications allow any modeling tool to generate the C code or binaries representing a dynamic system model which may then be seamlessly integrated in another modeling and simulation environment.
- The FMI for the Co-Simulation specification deals with models with the built-in solvers and the coupling of simulation tools. The specifications are separated into an execution part and a model description part (XML schema). In summary, an FMU (Functional Mock-up Unit) implementing any of the FMI specifications consists of
  - The XML model description.
  - Implementation of the C function interface in binary and/or source code.
  - Resources such as input data.
  - Image and documentation of the model.

2. Co-Simulation Options: Controls/Hydraulics

FMI – Functional Mock-up Interface

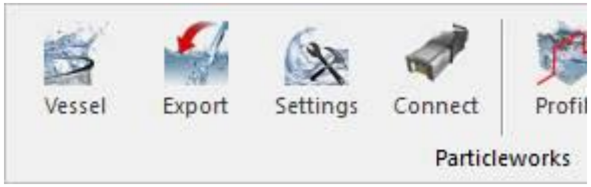
- The FMI entity for RecurDyn integrates the Modelica-based physical modeling into the RecurDyn/Control environment. The FMI entity offers the following main features:
  - Simulation of compiled dynamic models and FMUs in Control. The FMUs can be generated by an FMI-compliant tool such as Dymola, SimulationX or Silver.
  - Simulation of compiled dynamic models and FMUs using the RecurDyn's built-in integrators. This feature is useful to use the FMI entity without accessing Control.
  - The FMI entity supports the FMI1.0 and FMI2.0 for Co-simulation.

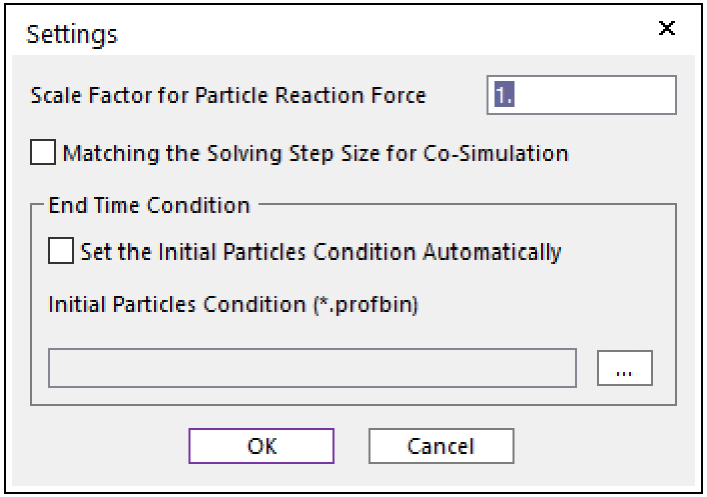
Limitations

- The input and output ports do not support strings, integer and boolean.
- The generated FMU currently does not support the use of several instances.
- The imported FMU does not offer to modify the parameter values and the start values as well as block outputs.

3. Co-Simulation Options: Particle-Based Fluids

- Capabilities found in RecurDyn Help: Communicator, Particleworks
- The Vessel “supports” the fluid.
- Export Vessels to obj files
- Settings:





Scale Factor for Particle Reaction Force 1.

□ Matching the Solving Step Size for Co-Simulation

End Time Condition

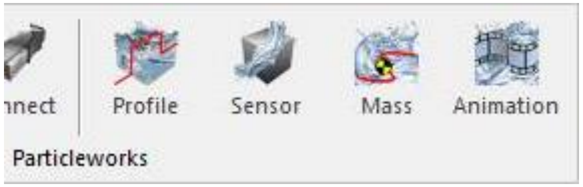
□ Set the Initial Particles Condition Automatically

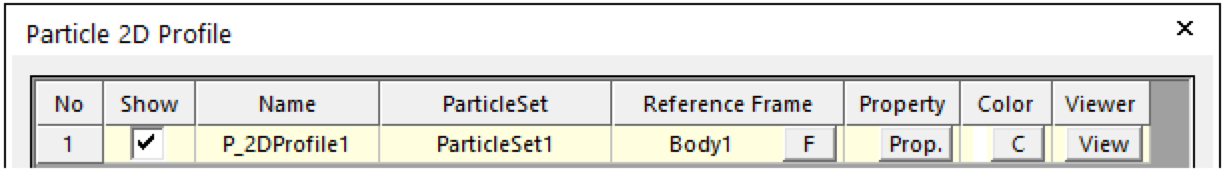
Initial Particles Condition (*.profbin)

- Connect: Indicates if Particleworks will be used in this co-simulation. Note that if no vessels are created, Particleworks will not be used, regardless of the status of the Connect icon.

3. Co-Simulation Options: Particle-Based Fluids

- Profile: This shows the profile of the particles at the outermost of the y-axis direction on the defined xy plane

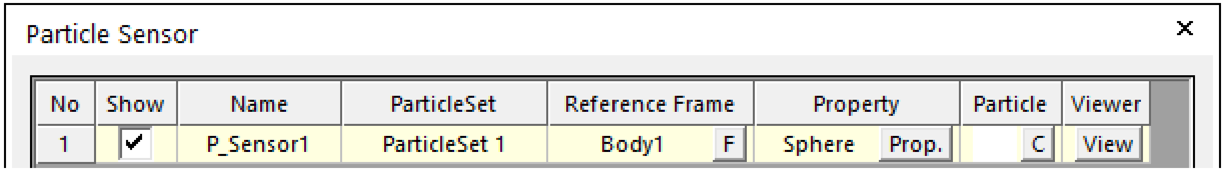




Particle 2D Profile

No	Show	Name	ParticleSet	Reference Frame	Property	Color	Viewer
1	☑	P_2DProfile1	ParticleSet1	Body1 F	Prop.	C	View

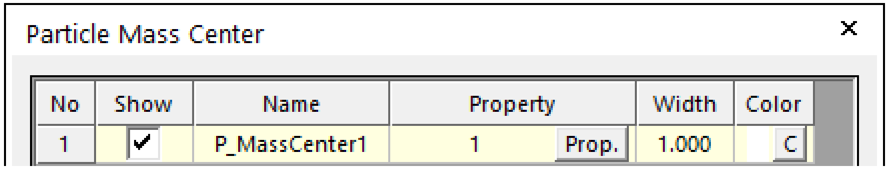
- Sensor: The user can count the number of particles which are in sphere or box at every animation frame.



Particle Sensor

No	Show	Name	ParticleSet	Reference Frame	Property	Particle	Viewer
1	☑	P_Sensor1	ParticleSet 1	Body1 F	Sphere Prop.	C	View

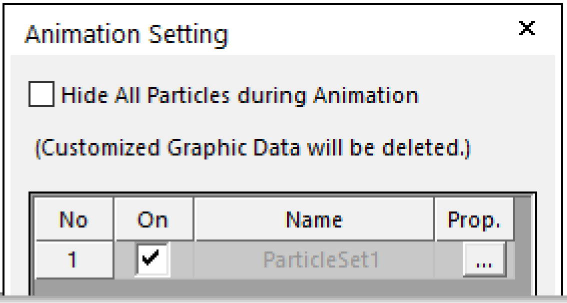
- Mass: trace position of the mass center of defined particle sets



Particle Mass Center

No	Show	Name	Property	Width	Color
1	☑	P_MassCenter1	1 Prop.	1.000	C

- Animation: Controls display of particle sets in the animation



Animation Setting

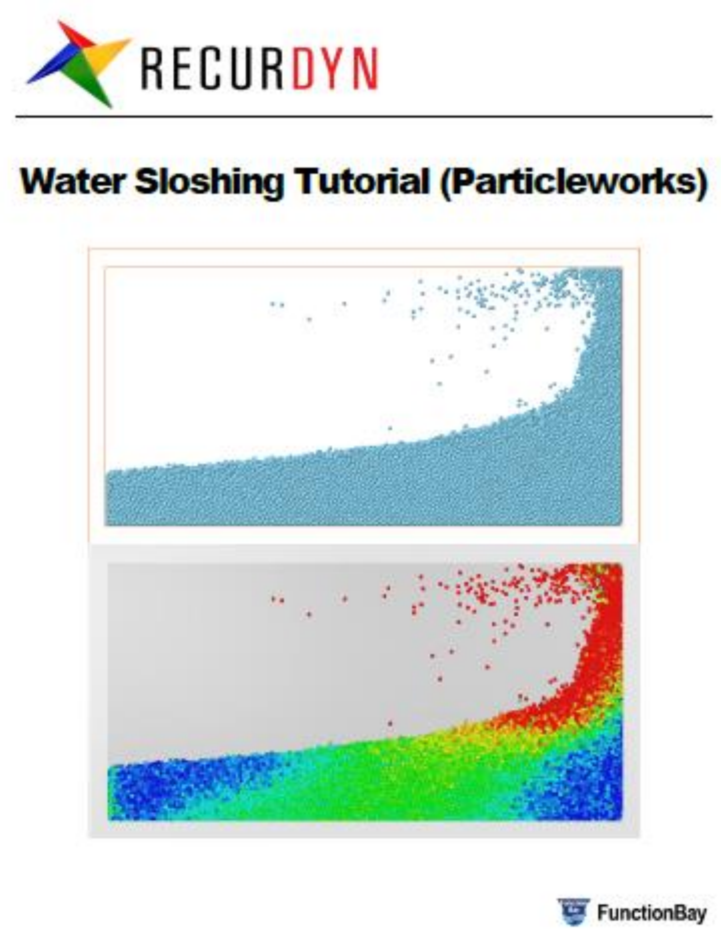
☐ Hide All Particles during Animation

(Customized Graphic Data will be deleted.)

No	On	Name	Prop.
1	☑	ParticleSet1	...

3. Co-Simulation Options: Particle-Based Fluids

- A Particleworks-RecurDyn tutorial can be found in the Help folder of the RecurDyn installation directory.



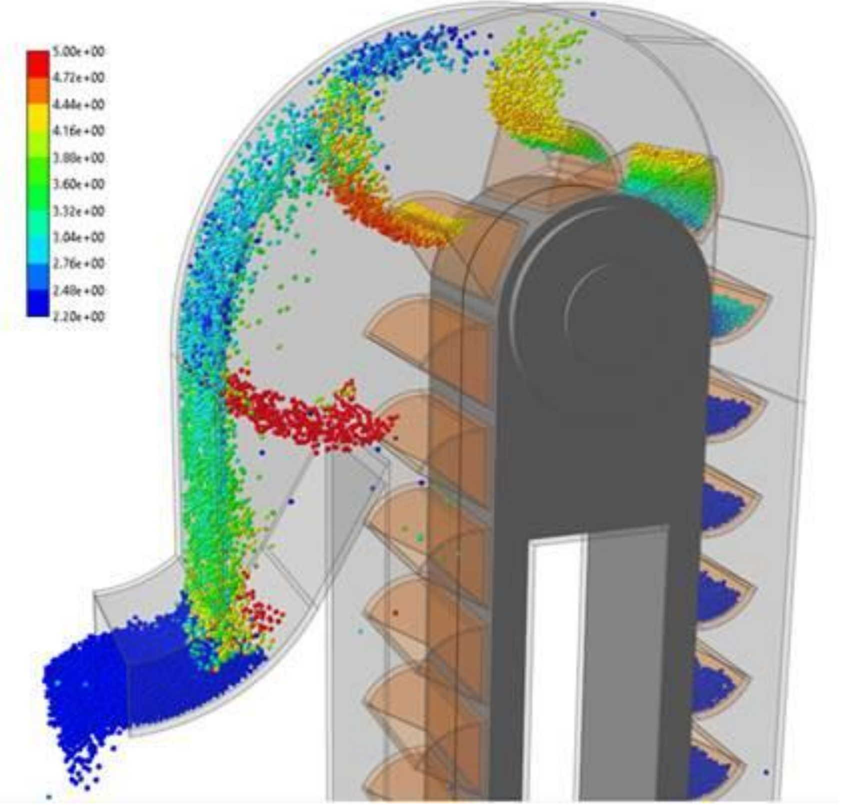
RECURDYN

Water Sloshing Tutorial (Particleworks)

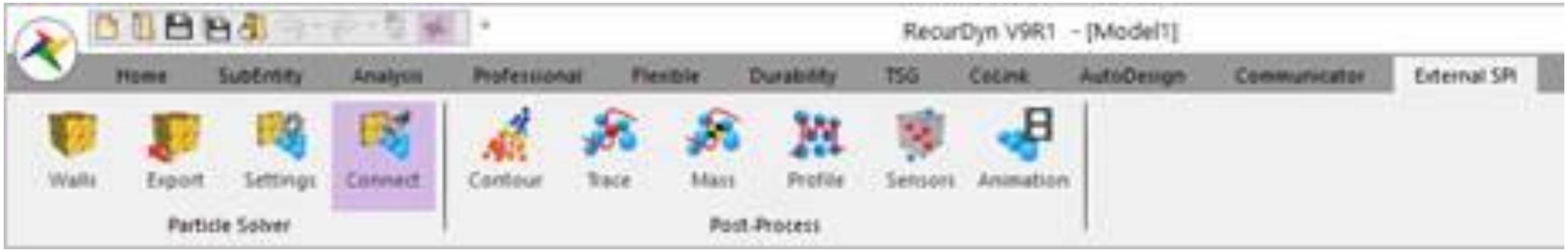
FunctionBay

4. Co-Simulation Options: Standard Particle Interface (SPI)

1. RecurDyn can co-simulate with particle solvers. Particle solvers are dynamics simulation software that simulates the motion of granular solid particles or fluids using a particle-based method.
2. RecurDyn's capability to co-simulate with particle solvers allows for simulations that have a system of rigid and flexible bodies that interact with systems of granular particles and/or fluids.
3. RecurDyn can simultaneously co-simulate with multiple particle solvers in a single RecurDyn model.



4. Co-Simulation Options: Standard Particle Interface (SPI)

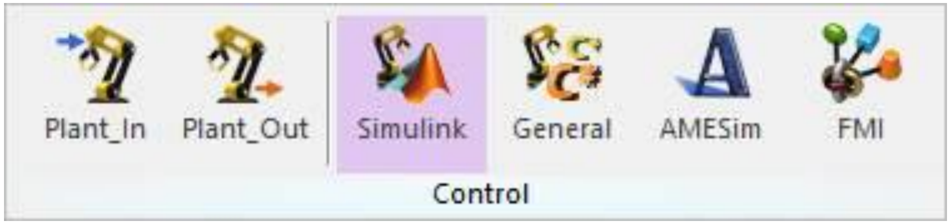


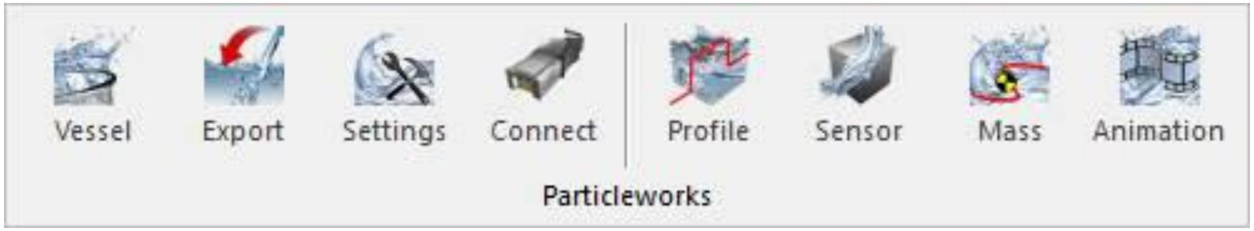
Requirements for a particle solver to co-simulate with RecurDyn

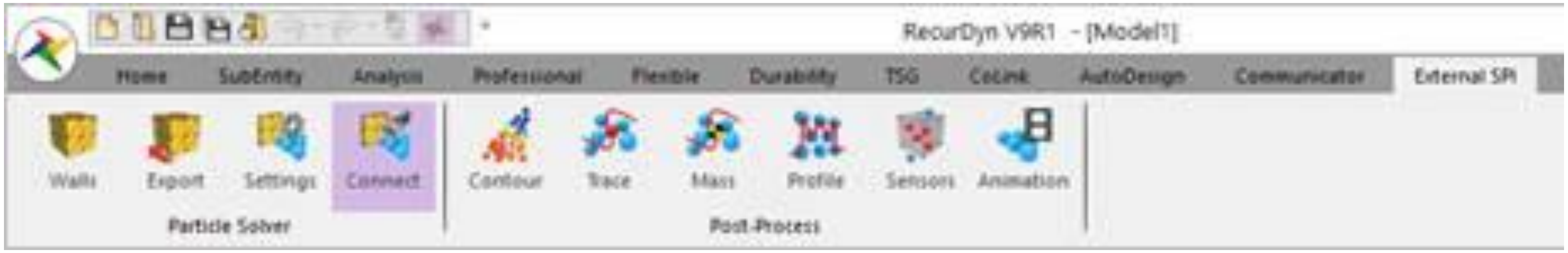
1. XML Configuration File: Provides required information for RecurDyn to access the particle solver, to display the particle solver in the RecurDyn GUI and to find the SPI Coupling DLL file. RecurDyn loads the XML configuration file when RecurDyn starts.
2. SPI Coupling DLL File: Acts as a communication channel between RecurDyn and the particle solver. RecurDyn loads the DLL file when a co-simulation starts. RecurDyn calls the RecurDyn SPI API functions in the DLL to communicate with the particle solver and control the co-simulation.

5. Summary

1. RecurDyn can co-simulate with a wide variety of external physics engines.
2. Certain interfaces are total preconfigured.
3. Other interfaces are API-based and can be used with any tool, including internal, proprietary tools.







Thanks for your attention!