Experimental Benchmark Control Problem for

Multi-axial Real-time Hybrid Simulation

Guideline for the Companion Code

Prepared by: Johnny Condori

Collaborators: Herta Montoya

Supervised by: Shirley J. Dyke

September 5, 2023

Contents

1. Getting started

The vmaRTHS companion package is available in a compressed (ZIP) file and can be downloaded from the MECHS website (https://mechs.designsafe-ci.org/) in the section "Project, Data & Benchmarks" under the Resources tab.

1.1 Software requirements

The vmaRTHS companion package was implemented using the MathWorks products listed in Table 1.

Table 1: Software requirements

Products/Software	Requirement
• MATLAB R2019b or later	Required
• Simulink Toolbox	Required
• Simulink Coder	Required
• MATLAB Control Toolbox	Required
• Simulink Desktop Real-time	Optional: If the vmaRTHS demo is executed in real-time. See Section 3 for more details.

1.2 File organization and description

Once the companion package is downloaded, move the compressed folder to the directory of your preference and extract its content. The companion package is organized into three directory levels, see Fig. 1. Level one directory "`SimTool_maRTHS_BenchM`" (the main directory) contains three files: this guideline, a main script, and a Simulink model. The level two directory "`Support_files`" contains the five scripts corresponding to the five steps to execute the vmaRTHS demo tool. Level three directories include additional files and data such as scripts, functions, ground motion records, experimental data, finite element definition, etc. See Table 2 for a more detailed description of these directories.

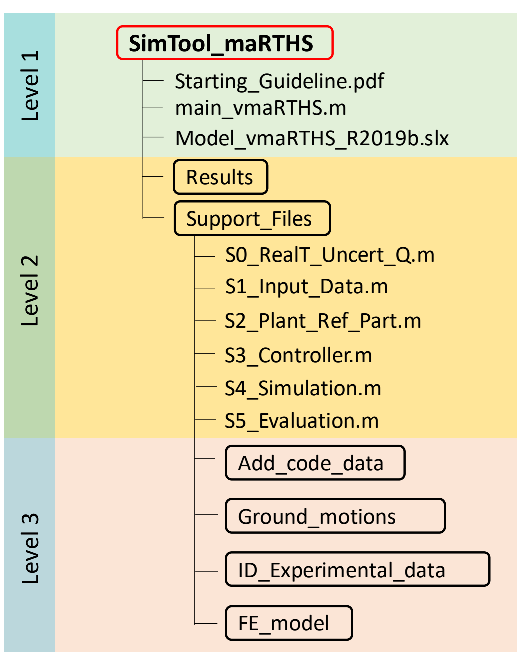

Fig. 1: Directory-level organization

Table 2: File and directory description

Level	File	Type	Need Modification?	Description
1	**SimTool_maRTHS_BenchM**	Directory		
	main_vmaRTHS.m	Script	Partially	It is the main script used to run the vmaRTHS demo tool by executing sequentially the support files (level 2). Users might want to modify essential data such as: sampling frequency, input ground motion, number of simulations, and real-time.
	Model_vmaRTHS_R2019b.slx	Model	Partially	It is the block diagram created in Simulink for the vmaRTHS model. Users can replace the <MIMO CONTROLLER> block according to their control approach design.
2	**Results**	Directory	NO	This is the place where all the time-history responses are saved. Also, the performance indices and their averages are stored. A ".mat" file is created for every execution or simulation.
	Support files	Directory		
	S0_RealT_Uncert_Q	Script	NO	It prompts the user for the following preferences: • Real-time. • Number of additional trials considering uncertainties in the plant. • Plotting time-history responses
	S1_Input_Data.m	Script	NO	It defines sensor noise parameters, saturation limits, conversion factors, etc., and load the Simulink model.
	S2_Plant_Ref_Part.m	Script	NO	It calls the scripts and functions that define the main components or blocks in the Simulink model: Control plant with uncertainties, reference structure, numerical substructure, kinematic transformation parameters, and a model for the physical frame that is used to estimate the feedback force from the experimental frame.
	S3_Controller.m	Script	YES	It defines the control sample design included in this benchmark and the Kalman filter based on the identified model. It also loads the necessary parameters to be used by the <MIMO CONTROLLER> block in the Simulink model. Users should change this file accordingly.
	S4_Simulation.m	Script	NO	It sets up the simulation parameters, run the simulation, and save the outputs in array format.
	S5_Evaluation.m	Script	NO	It computes the performance indices of the set of simulations, calculates their mean values and prints them on the screen. It also shows if the control is realizable by checking the saturation limits for the control inputs.
3	**Add_code_data**	Directory		
	build_reference.m	Script	NO	It builds the reference finite element model described in the companion paper
	build_physical_ideal.m	Script	NO	It defines an ideal finite element model of the physical substructure (frame) that is used to obtain later a numerical substructure.
	build_numerical.m	Script	NO	It computes the numerical substructure using the reference model and the ideal physical model.
	physicalFrame_feedbackForce.m	Script	NO	It only computes an estimation of the feedback force from the physical (experimental) frame only. This force will be fed back to the numerical substructure (see Simulink model) to close the

				loop in the vmaRTHS simulation. Force readings from sensors (load cells) includes the inertial effect of the coupler (see Section 3.4 of the companion paper), which is important from a control perspective, but it is not part of the equation of motion of the original partitioned system.
	fcn_build_uncert_plant.m	Function	**NO**	It takes the identified plant and adds random variation to its poles and zeros to generate a transfer function that represent an uncertain control plant to be controlled.
	fcn_coupler_geometry.m	Function	**NO**	It defines the geometry of the coupler.
	fcn_newmark_beta_const.m	Function	**NO**	It defines the parameters for the numerical integration of the reference structure and numerical substructure.
	ID_Control_plant_100Hz.mat	Data	**NO**	It contains the model of the identified control plant in the form of transfer function matrix components and a state-space realization. This model is used as the nominal model for the sample controller design.
	Plot_Results	Script	**NO**	This script plots all the time-history tracking response of the vmaRTHS cases. It can be used in the MATLAB Command Window any time after the set of simulations has finished.
	Ground_motions	Directory	**NO**	It contains three historic ground acceleration records without scaling: El Centro 1940, Kobe 2008, and Morgan Hill 1984.
	ID_experimental_data	Directory	**NO**	It contains a set of experimental input-output data of the control plant based on band-limited white noise input signals and axial displacement of hydraulic actuators utilized in the maRTHS experiment as the outputs. This information will help the users to define their own control plant model if their control approach requires a different control plant model description than the model provided, or if the users prefer to define their own model description.
	FE_model	Directory	**NO**	This directory contains the definition of the finite element models for the reference structure and numerical substructure.

2. Running the virtual maRTHS

The following steps describe the procedure to execute the vmaRTHS demo tool:

1. Start MATLAB and change to the Benchmark folder “SimTool_maRTHS_BenchM”
2. Open the script `main_vmaRTHS.m`. By default, the sampling frequency and ground motion record are:

   - fs = 1024 Hz
   - El Centro, scaling factor: 0.40

   The users can modify these values according to their needs in the first section of the script.

3. Run `main_vmaRTHS.m`. This script will execute the five scripts of the level two directory, which in turn they will call/use the additional scripts and data of the level three directories, see Table 2.

The simulation will finish with a table of the indices shown on the screen and a final message concluding whether the controller is realizable. The responses of each simulation and the average indices are stored in the folder `Results`.

2.1 Changing the sample control implementation

The sample controller provider is defined by the script `S3_Controller.m` and the block <MIMO CONTROLLER> in the Simulink model `Model_vmaRHTS_R2020b.slx`. Both, the script and the block, must be modified or replaced by the users to implement their own control approach.

3. Simulink desktop real-time

Running the vmaRTHS in real-time requires a real-time kernel that must be installed in the machine. Basically, this kernel allows the real-time task (this vmaRTHR simulation) to have the highest priority to use the CPU for execution at the selected sample rate without interference.

To know if you have installed a Simulink Desktop Real-Time kernel in your machine, use the following command in the Command Window of MATLAB:

```
>> sldrtkernel -version
```

If no kernel is installed in your machine, the following message will appear:

```
There is no Simulink Desktop Real-Time kernel installed.
```

Or you can use the following command with its output:

```
>>rtwho

Simulink Desktop Real-Time installation is not complete. Please type
'sldrtkernel -setup' to complete the installation. Type 'help
sldrtkernel' for more information.
```

Users should refer to the online MATLAB Help documentation for detailed instructions on the install/uninstall procedure. The following link provides access to it:

https://www.mathworks.com/help/sldrt/ug/real-time-windows-target-kernel.html

Does your PC shut down when you run a model with Simulink Desktop Real-Time?

If you see the following messages when trying to run in real-time:

```
The Hyper-V hypervisor has been detected. The Simulink Desktop Real-
Time kernel cannot run in the presence of the hypervisor. Please
disable the Hyper-V operating system component before attempting to use
Simulink Desktop Real-Time.
```

Or, on MATLAB versions previous to R2022a:

```
Warning: The "Real-Time Synchronization" block has timed out while
trying to synchronize to real-time kernel.
```

You can try the following solution:

https://www.mathworks.com/help/sldrt/ug/troubleshoot-slow-or-halted-simulation-on-windows.html

Some Windows versions have updates that are incompatible with the real-time kernel. The following Bug Report link provides patches for some MATLAB releases. The user should contact MathWorks Technical Support if running a different version:

https://www.mathworks.com/support/bugreports/1719571