



PRACTICAL romAI

CONTENTS

1. DISCLAIMER	2
2. INTRODUCTION	3
3. GENERAL APPROACH	4
4. EXAMPLE	5

1 Disclaimer

Every effort has been made to keep this document free from technical as well as other mistakes. However, publishers and authors will not be responsible for loss, damage in any form and consequences arising directly or indirectly from the use of this document. © 2021 Altair Engineering, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, or translated to another language without the written permission of Altair Engineering, Inc. To obtain this permission, write to the attention of Altair Engineering legal department at: 1820 Big Beaver, Troy, Michigan, USA, or call +1-248-614-2400

2 Introduction

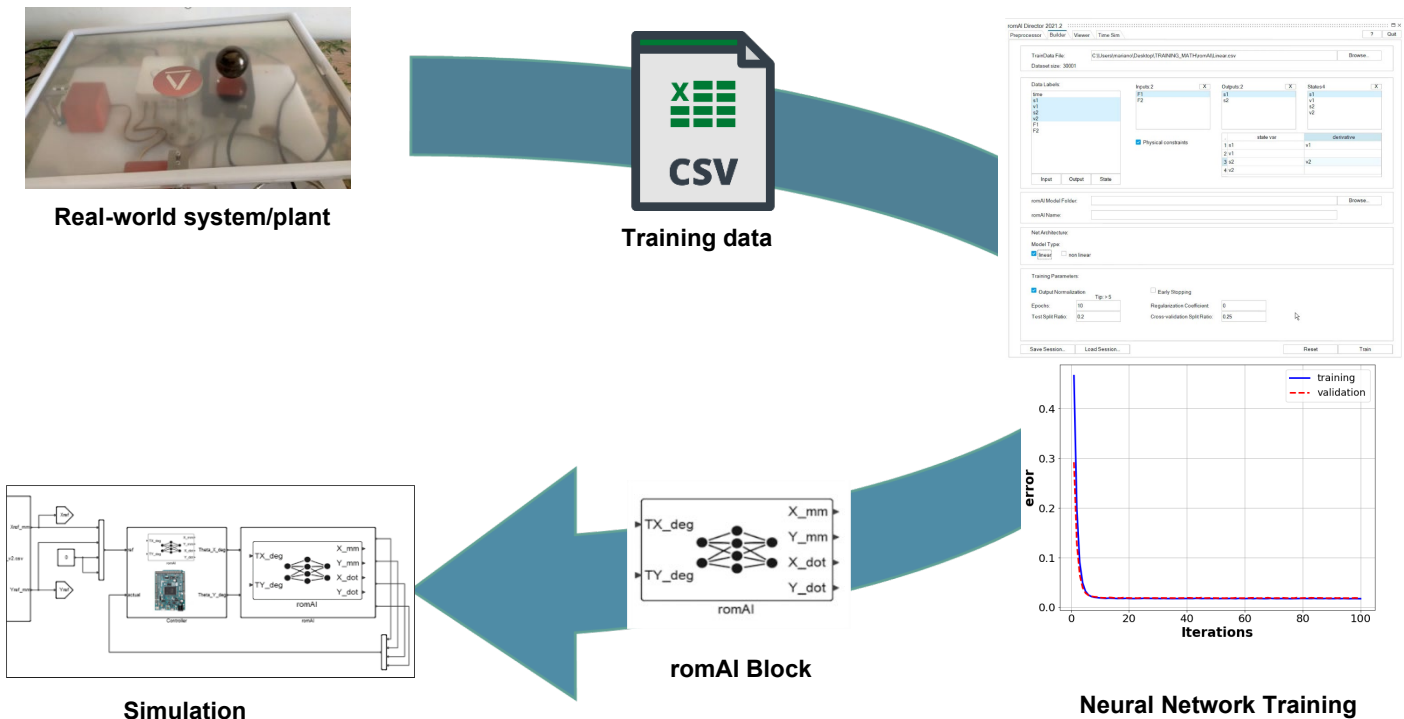
romAI is an application that is capable of approximating static and dynamic systems by making use of either simulation, test or real-world data. Its primary advantage is that it can be used to speed up system output estimations or predictions, in response to some sort of new control scheme or input parameter change. And this can then be used to refine the design of the system through simulation, or during actual operation to predict and obtain outputs.

Because romAI is a mathematical approach it can be widely applied to a variety of problems, applications and physics. The training data can be obtained from real-world sources or simulations that approximate reality well.

The generated models are efficient, and real-time compliant. They run very fast, and they can be used in place of real-world systems to test the response and performance of control strategies, or the response to varying inputs.

The romAI tool is based on Neural Networks, so in essence it generates, internally, a mapping between inputs and outputs. The quality of this mapping will depend on factors such as the quality and amount of data available to “train” the Neural Network, the careful selection of the parameters of the network itself (called hyperparameters), and the time available to train the network.

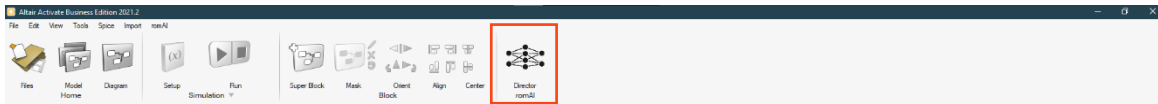
The romAI workflow is fairly simple. First, data consisting of inputs, outputs and system states are presented to the application in the form of a comma-separated values file (.csv). A neural network is then trained based on this data and selected hyperparameters. This trained network can then be used in Activate™ as a romAI block, which has the same inputs and outputs that the original plant/system would have. This can then be used for further simulation and testing.



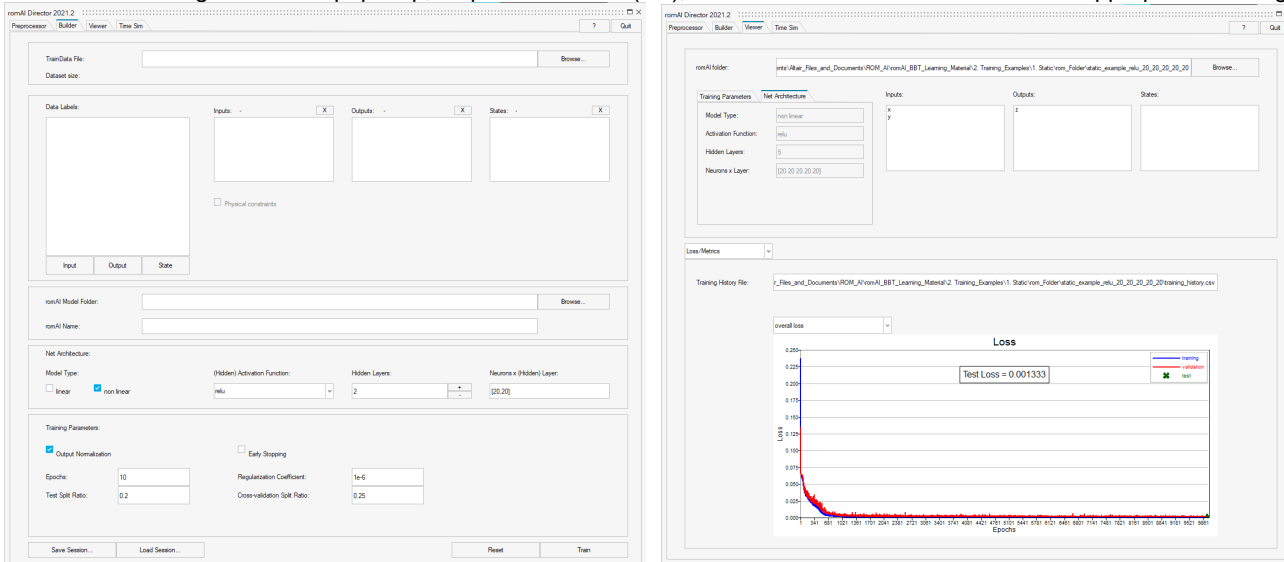
Workflow Overview

3 General Approach

With the romAI and romAIdirector libraries installed for Activate™ and Compose™. We begin by starting Activate™ and then launching romAI Director from the ribbon.



In the tool's dialogue box that pop's up, as pictured below (left), in the Builder tab one can select the appropriate training data



file, as well as the hyperparameters for the romAI neural network model to be trained. After training, in the Viewer tab, the resulting network can be evaluated for its performance and accuracy v/s a .csv file of new inputs and expected outputs.

And finally in the Time Sim tab it is possible to compare the romAI's output to that of a data set for the same system or plant.

4 Example

Some prepared examples have been provided along with this document. Training data is present in a folder titled “Training Data”, example trained romAIs as well as the parameters used to generate them, in the form of Builder session files (.oml), in a folder titled “Training_Examples”, and finally a folder named “romAI_Verification” which contains Activate™ sketches which will compare the romAI and another conventional system model.

An example for the generation of a romAI block for use within activate is as follows, considering the Static example:

1. Launch Activate™ and then launch the romAI Director tool.
2. Under the Builder tab, in the TrainData File field, add the full path to the “1. Static.csv” file located in the “Training Data” folder.

Or simply select the file in the selection dialogue after clicking “Browse...”

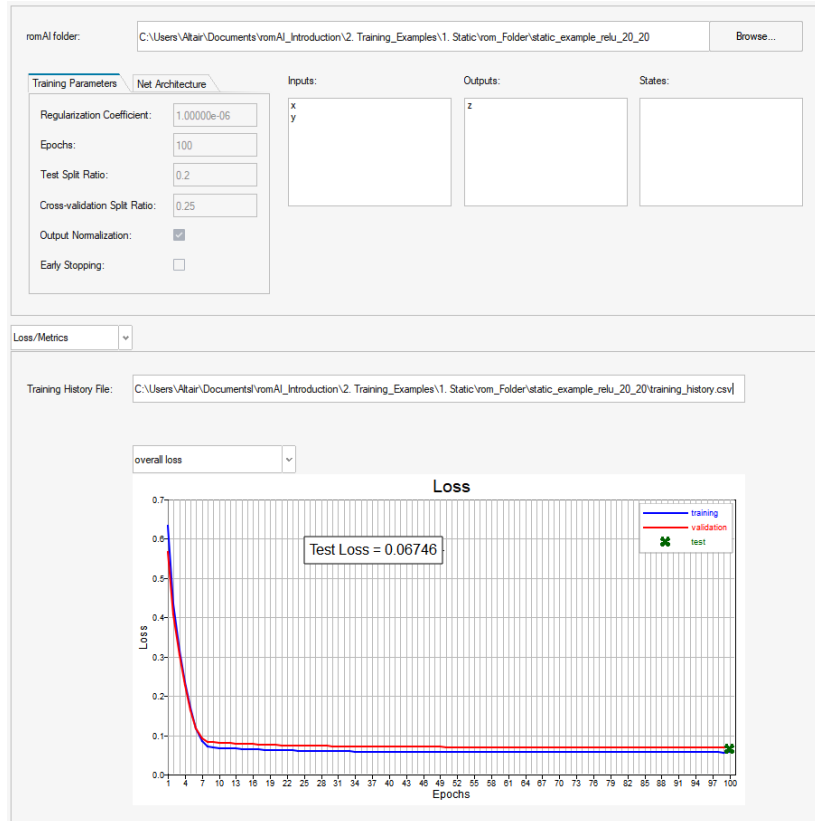
The data labels will automatically be selected from the csv file’s header.

4. Navigate to a folder, where you’d like to generate your rom

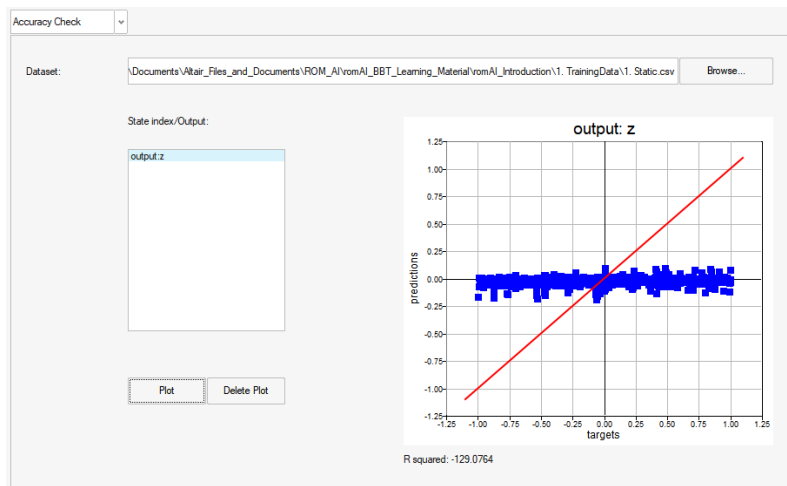
AI files

in, using the “Browse...” button. It is suggested to use the appropriate “rom_Folder” under the example being solved for in the “Training_Examples” folders. Pick an appropriate name for the romAI.

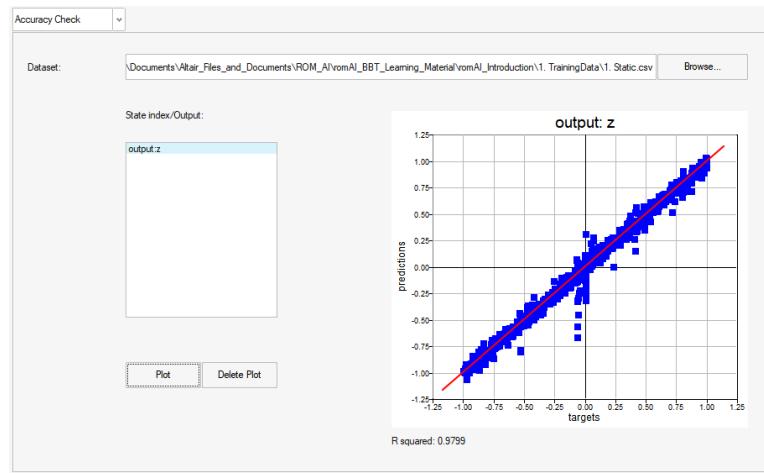
6. Click on the “Train” button. The application will now run based on the set conditions.
7. When it is done, the Viewer tab is selected and the Network parameters and the Loss/Metrics are visible.



8. The accuracy of the trained romAI model can also be checked based on the R-Squared value by picking the “Accuracy Check” option in the drop down. This requires a data file that is similar or the same as the training data. It can also be a newer file with more recent data. It appears as follows...



In this case the agreement is very poor. But for a network with different parameters the results can be far better...

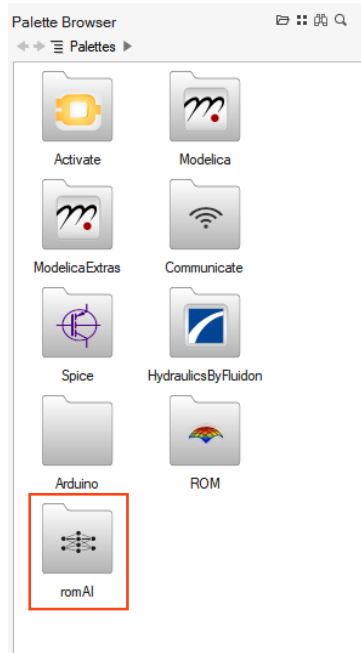


With a trained romAI it is simply a matter of including the romAI block in an Activate™ sketch and using it to replace a mathematical model or the like.

The verification sketches in the “romAI_Verification” folder all use the romAI block and the corresponding models to compare results.

To use a romAI block in Activate™:

1. In the Palette Browser, in the base Palettes location look for the romAI folder.



2. Within it is the romAI block. To use it, click and drag it into the diagram area. To select the rom folder to be used, either double click on the block or right-click and select “Parameters”. In the subsequent dialog box, pick out the desired romAI folder to be used. And proceed to connect the block as required.

