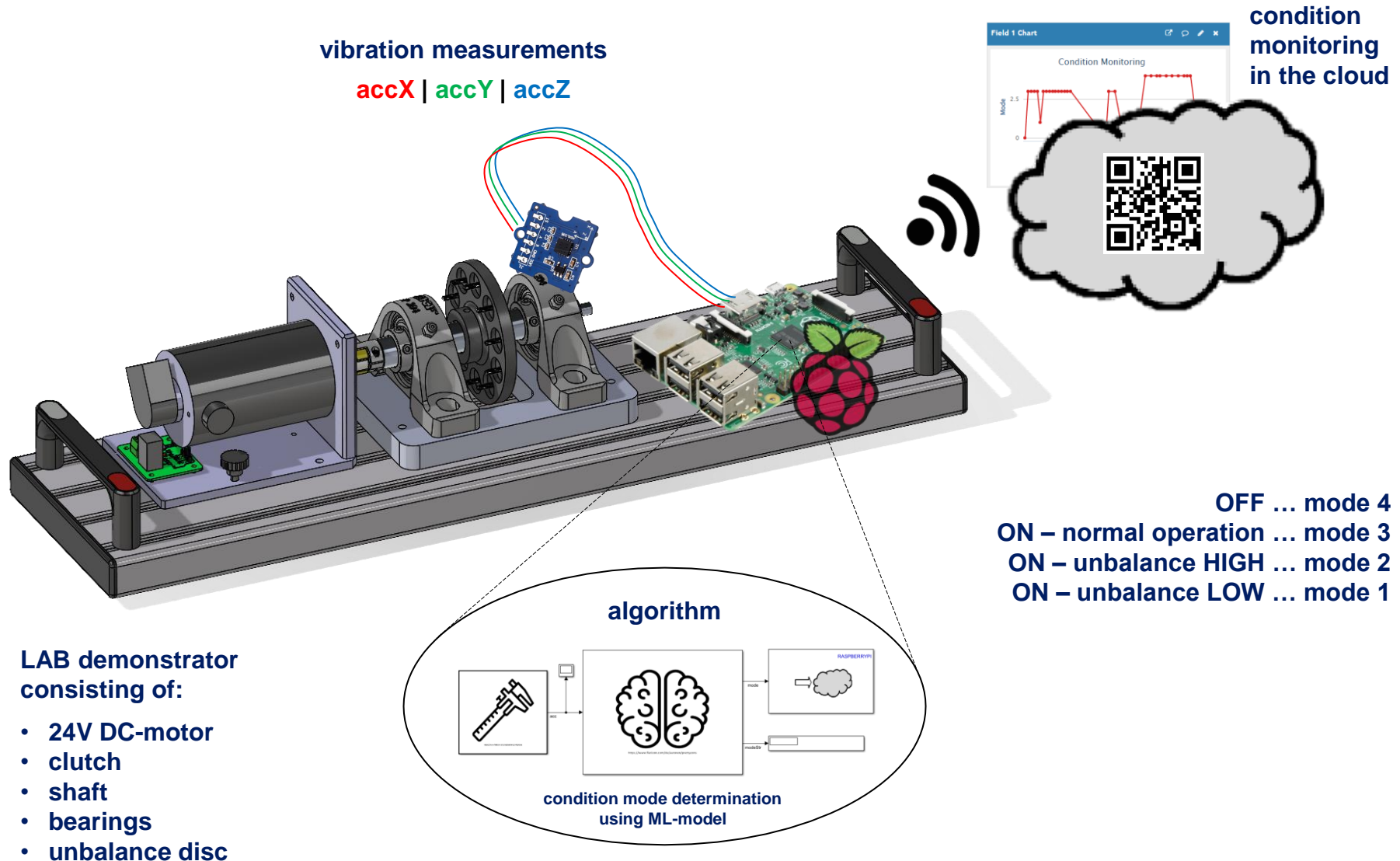
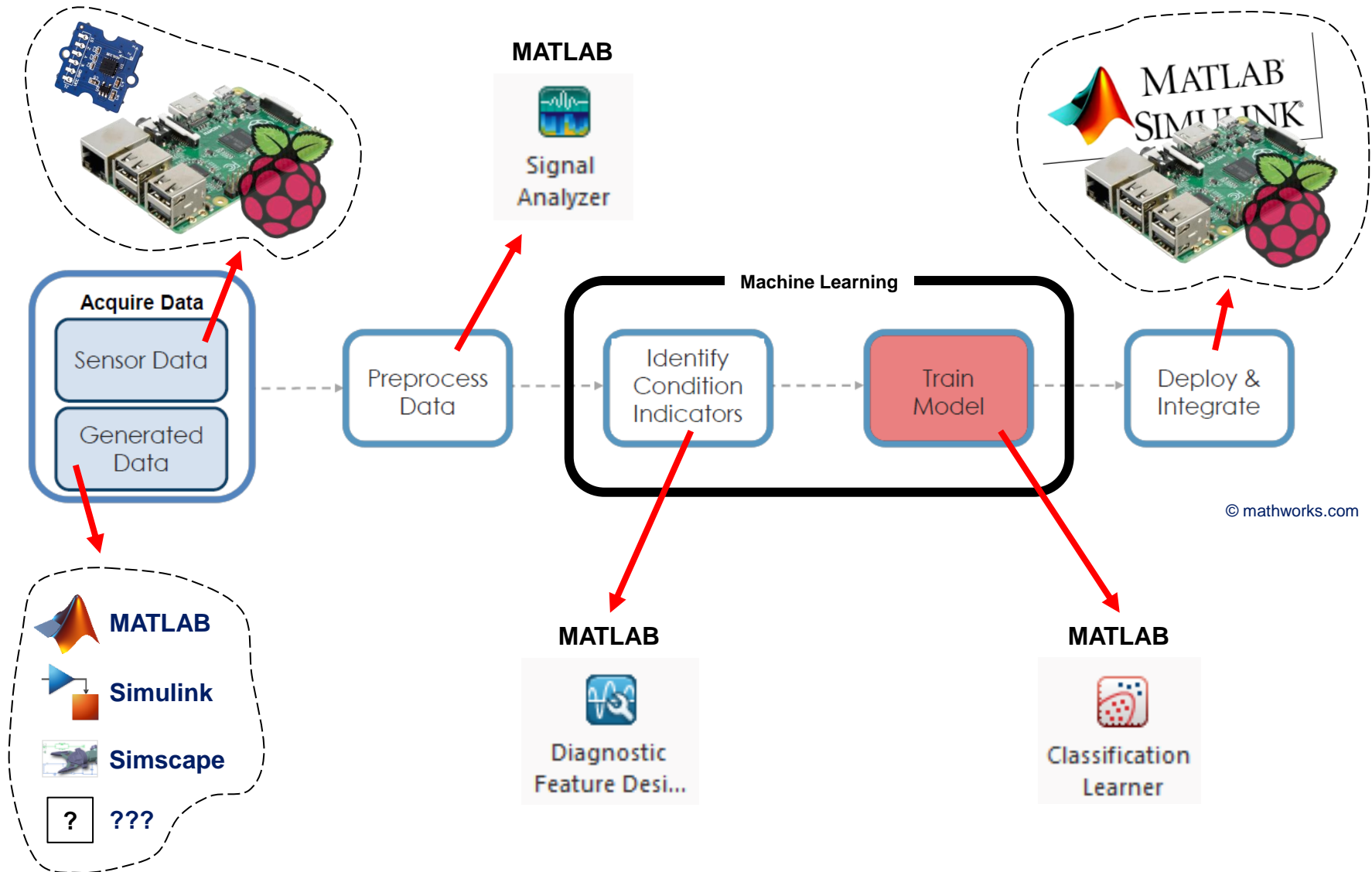


# Labor Condition Monitoring – Setup



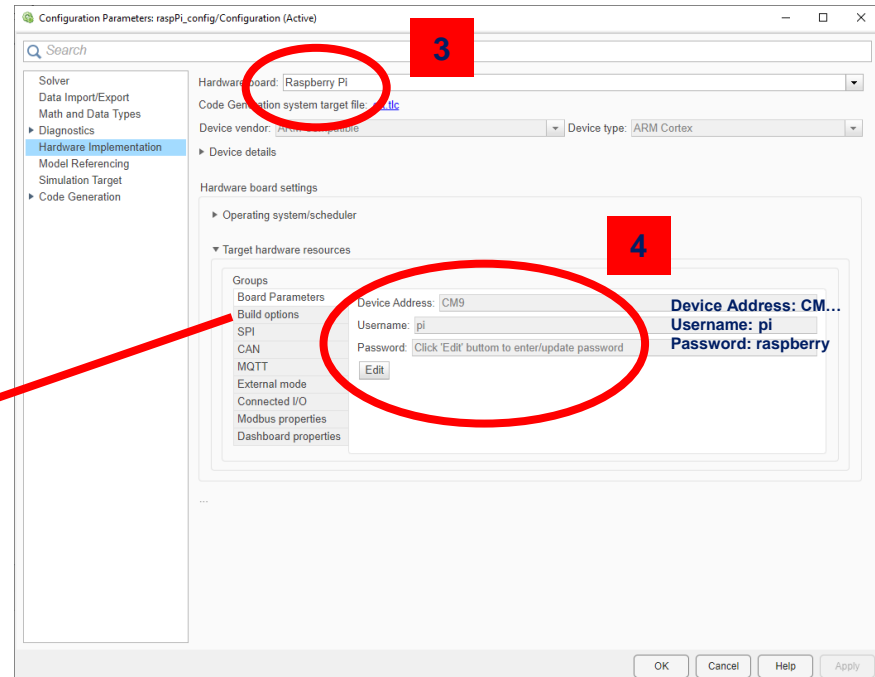
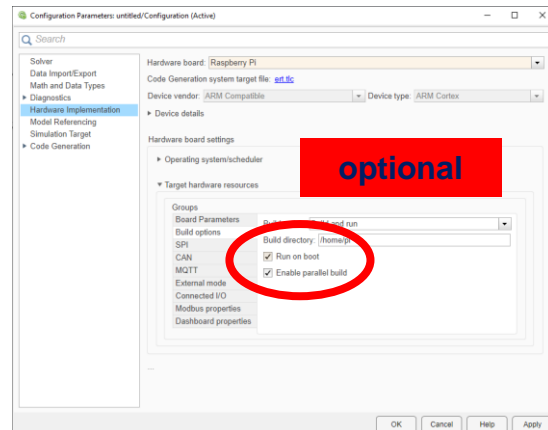
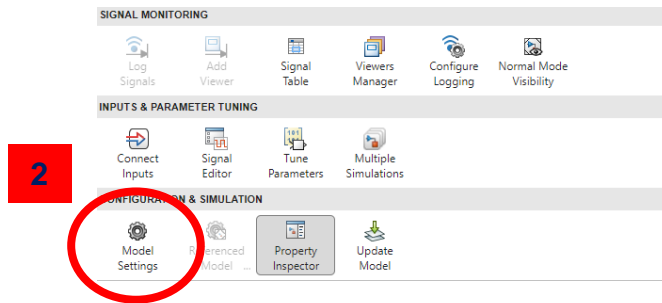
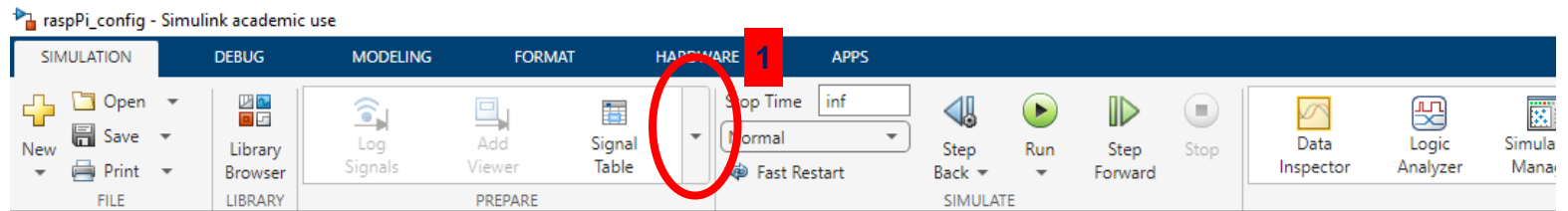
# Condition Monitoring – Goal / Workflow



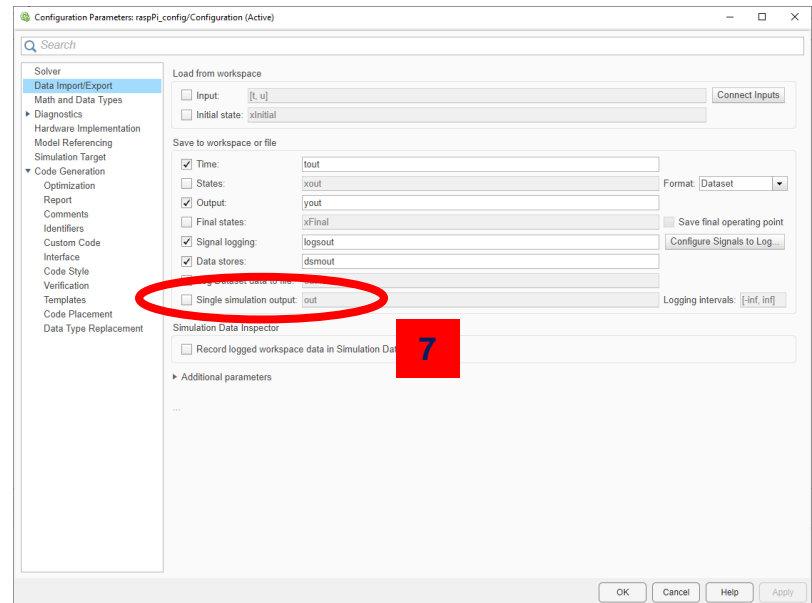
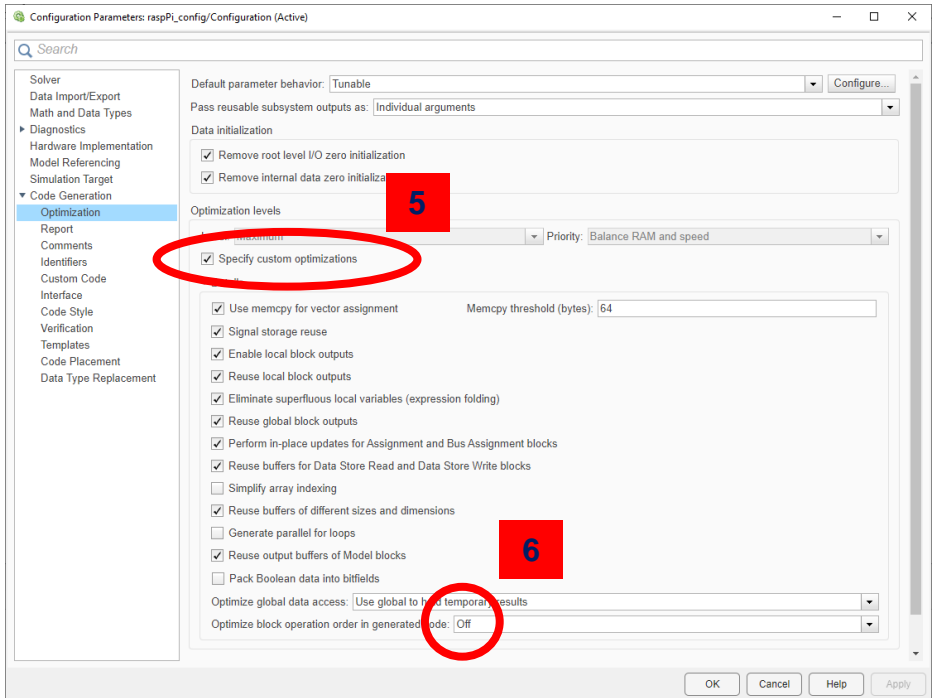
# Condition Monitoring – Add-Ons (MATLAB R2024a)

Add-Ons	Requires
Communications Toolbox <small>version 24.1</small>	DSP System Toolbox <small>version 24.1</small> Signal Processing Toolbox <small>version 24.1</small>
Embedded Coder <small>version 24.1</small>	MATLAB Coder <small>version 24.1</small>
MATLAB Support Package for Raspberry Pi Hardware <small>version 24.1.2</small>	
Predictive Maintenance Toolbox <small>version 24.1</small>	Signal Processing Toolbox <small>version 24.1</small> Statistics and Machine Learning Toolbox <small>version 24.1</small> System Identification Toolbox <small>version 24.1</small>
Simulink Coder <small>version 24.1</small>	Simulink <small>version 24.1</small> MATLAB Coder <small>version 24.1</small>
Simulink Support Package for Raspberry Pi Hardware <small>version 24.1.1</small>	Simulink <small>version 24.1</small>

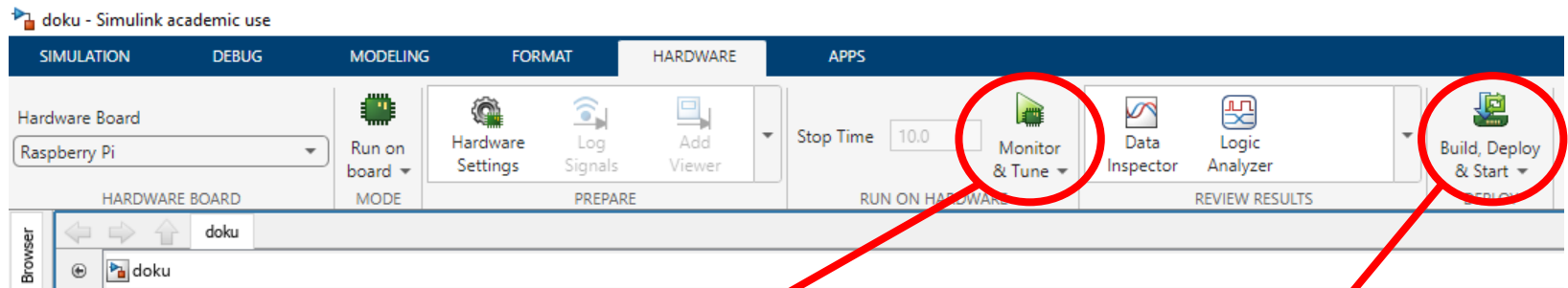
# Raspberry Pi Configuration in Simulink (i)



# Raspberry Pi Configuration in Simulink (ii)



# Raspberry Pi Configuration in Simulink (iii)



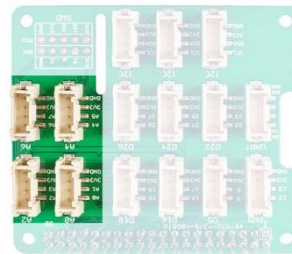
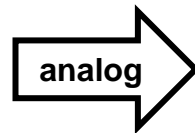
**run on RaspPi  
& monitor in Simulink**

**run on RaspPi only**

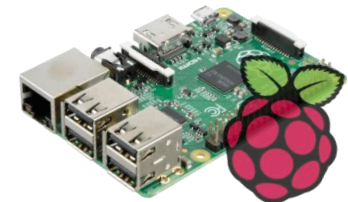
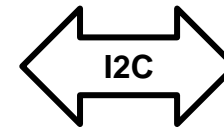
# Measuring System / Hardware



**ADXL335**

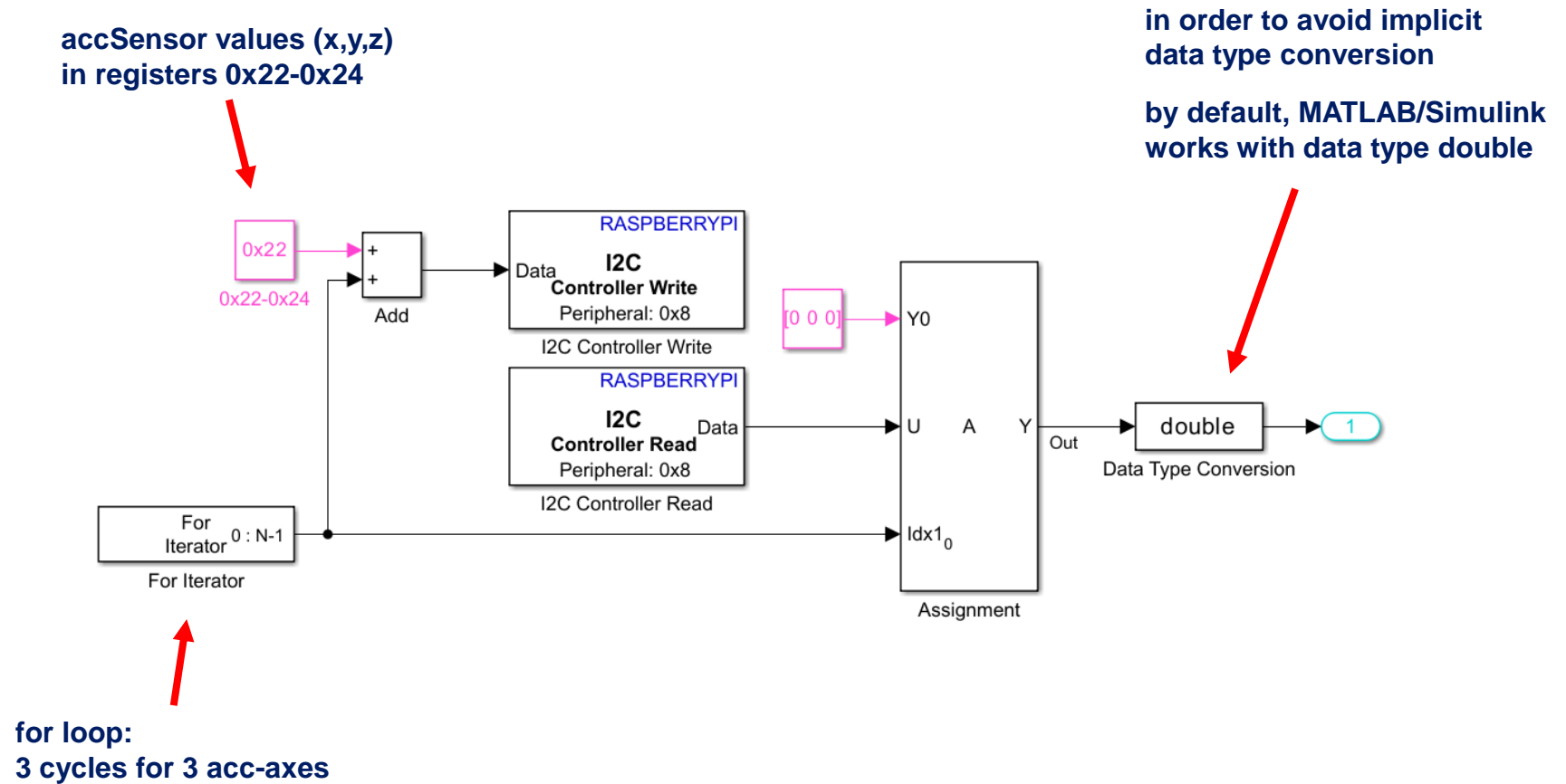


**Grove Base HAT**



**Raspberry Pi 4**  
(flash card preconfigured)

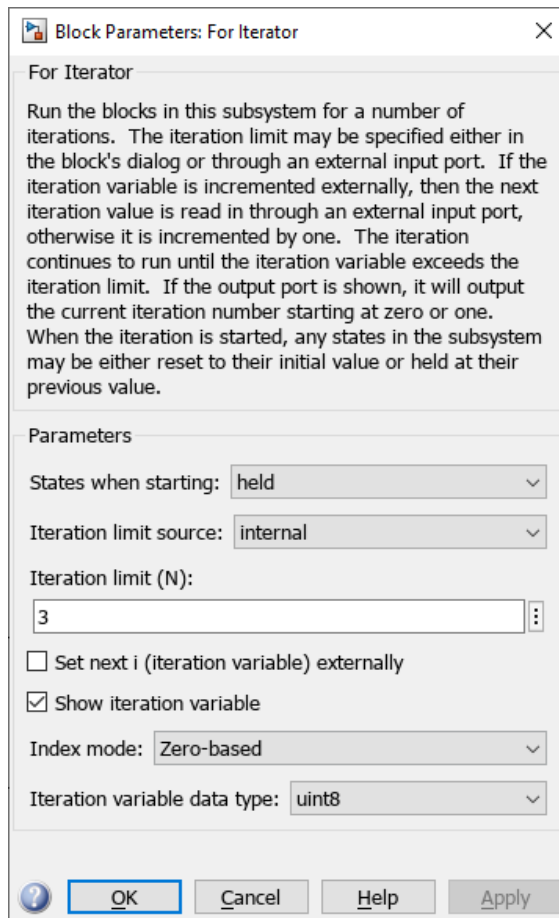
# Measuring System / Software (i)





# Measuring System / Software (ii)

## For Iterator



**Block Parameters: For Iterator**

For Iterator

Run the blocks in this subsystem for a number of iterations. The iteration limit may be specified either in the block's dialog or through an external input port. If the iteration variable is incremented externally, then the next iteration value is read in through an external input port, otherwise it is incremented by one. The iteration continues to run until the iteration variable exceeds the iteration limit. If the output port is shown, it will output the current iteration number starting at zero or one. When the iteration is started, any states in the subsystem may be either reset to their initial value or held at their previous value.

Parameters

States when starting: held

Iteration limit source: internal

Iteration limit (N): 3

☐ Set next i (iteration variable) externally

☒ Show iteration variable

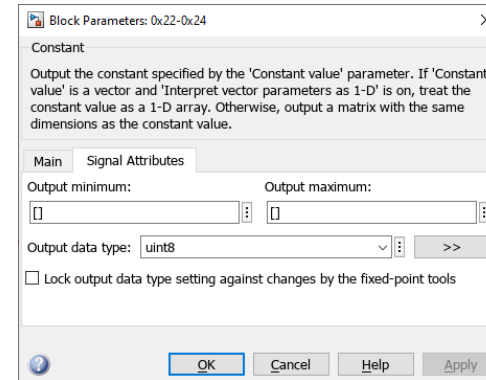
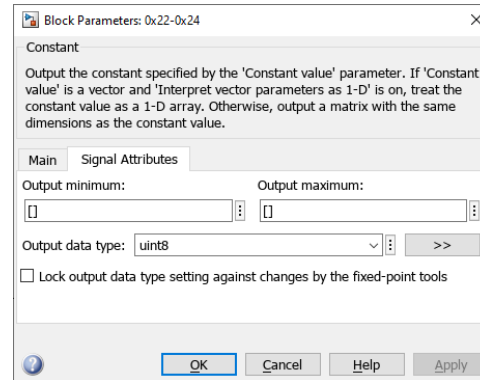
Index mode: Zero-based

Iteration variable data type: uint8

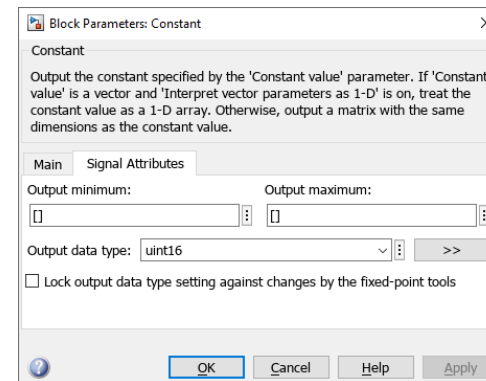
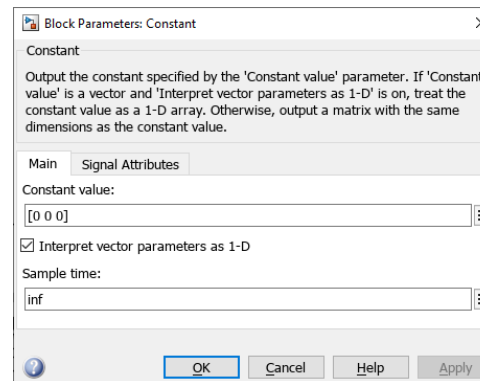
? OK Cancel Help Apply

# Measuring System / Software (iii)

## Constant (0x22-0x24)



## Constant [0 0 0]



# Measuring System / Software (iv)

## Add

**Block Parameters: Add**

Sum

Add or subtract inputs. Specify one of the following:

- a) character vector containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
- b) scalar, >= 1, specifies the number of input ports to be summed.

When there is only one input port, add or subtract elements over all dimensions or one specified dimension.

Main | **Signal Attributes**

Icon shape: rectangular

List of signs:

++

OK Cancel Help Apply

**Block Parameters: Add**

Sum

Add or subtract inputs. Specify one of the following:

- a) character vector containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
- b) scalar, >= 1, specifies the number of input ports to be summed.

When there is only one input port, add or subtract elements over all dimensions or one specified dimension.

Main | **Signal Attributes**

☐ Require all inputs to have the same data type

Accumulator data type: Inherit: Inherit via internal rule

Output minimum: Output maximum:

Output data type: Inherit: Inherit via internal rule

☐ Lock data type settings against changes by the fixed-point tools

Integer rounding mode: Floor

☐ Saturate on integer overflow

OK Cancel Help Apply

# Measuring System / Software (v)

## I2C Controller Write/Read

**Block Parameters: I2C Controller Write**

I2C Controller Write

Write data to an I2C peripheral device or an I2C peripheral device register.

The block accepts an [Nx1] or [1xN] array of data type int8, uint8, int16, uint16, int32, uint32, single or double.

[View pin map](#)

Board: Pi 4 Model B

Peripheral address: 8

Peripheral byte order: LittleEndian

☐ Enable register access

☐ Send NACK at the end of data transfer

☐ Remove stop bit at the end of data transfer

☐ Output error status

[OK](#) [Cancel](#) [Help](#) [Apply](#)

**Block Parameters: I2C Controller Read**

I2C Controller Read

Read data from an I2C peripheral device or an I2C peripheral device register.

The block outputs the values received as an [Nx1] array.

[View pin map](#)

Board: Pi 4 Model B

Peripheral address: 8

Peripheral byte order: LittleEndian

☐ Enable register access

Data type: uint16

Data size (N): 1

☐ Send NACK at the end of data transfer

☐ Remove stop bit at the end of data transfer

☐ Output error status

Sample time: -1

[OK](#) [Cancel](#) [Help](#) [Apply](#)

# Measuring System / Software (vi)

## Assignment

**Block Parameters: Assignment** [X]

**Assignment**  
Assign values to specified elements of a multidimensional output signal. The index to each element is identified from an input port or this dialog. You can choose the indexing method for each dimension by using the "Index Option" parameter.

**Parameters**

Number of output dimensions:

Index mode:

	Index Option	Index	Output Size
1	Index vector (port) ▾	from port <I...>	Inherit from ...

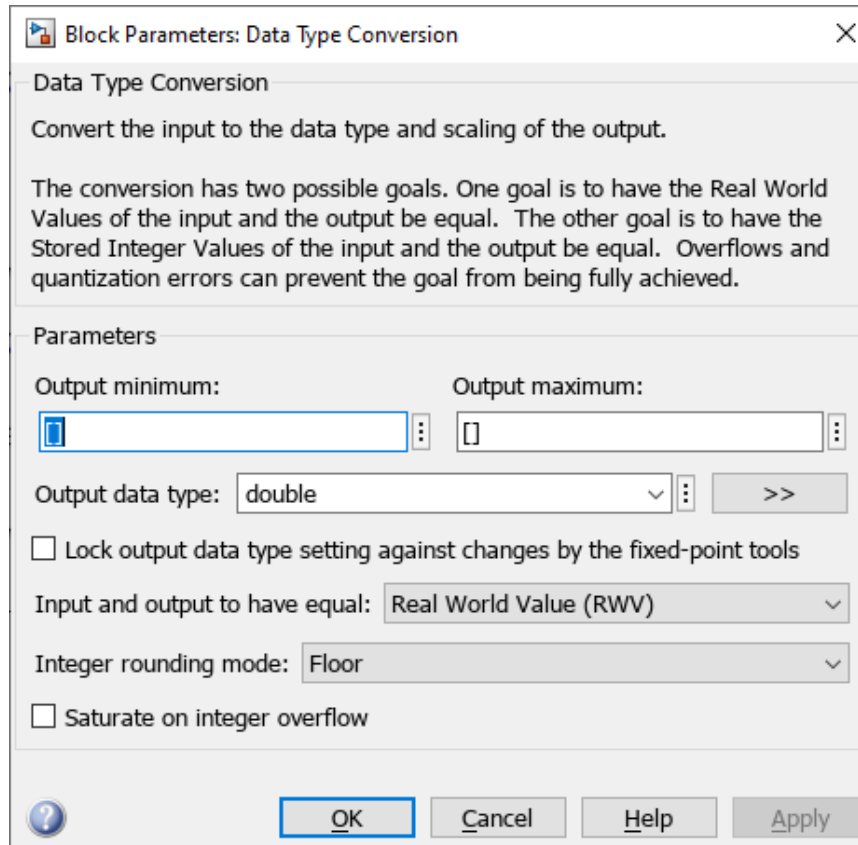
Initialize output (Y):

☐ Check for out-of-range index in accelerated simulation

[?] [OK] [Cancel] [Help] [Apply]

# Measuring System / Software (vii)

## Data Type Conversion



**Block Parameters: Data Type Conversion**

**Data Type Conversion**

Convert the input to the data type and scaling of the output.

The conversion has two possible goals. One goal is to have the Real World Values of the input and the output be equal. The other goal is to have the Stored Integer Values of the input and the output be equal. Overflows and quantization errors can prevent the goal from being fully achieved.

**Parameters**

Output minimum:  Output maximum:

Output data type:  >>

☐ Lock output data type setting against changes by the fixed-point tools

Input and output to have equal:

Integer rounding mode:

☐ Saturate on integer overflow

? OK Cancel Help Apply



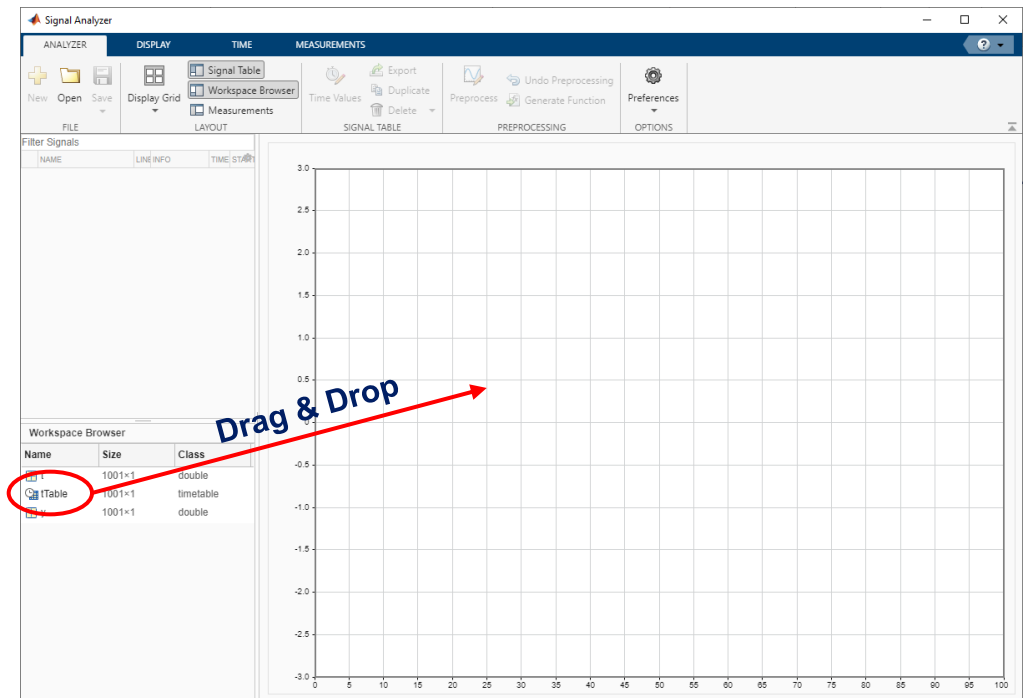
# Signal Analyzer (i)

## input format

```
t = (0:0.01:10)';
y = sin(t);
```

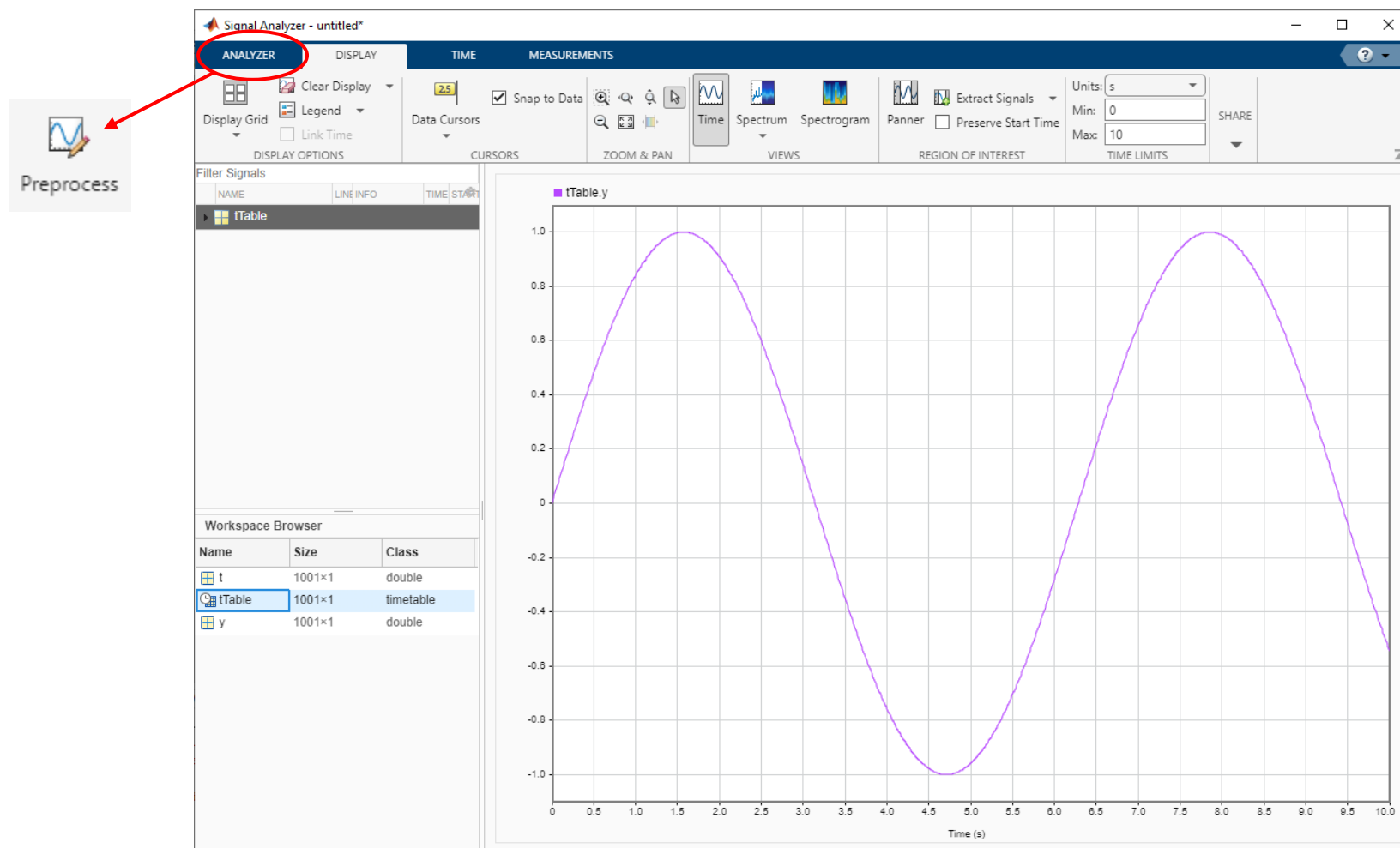
```
% prepare for SignalAnalyzer
x = seconds(t);
tTable = timetable(x,y);
```

tTable		
1001x1 timetable		
	x	1 y
1	0 sec	0
2	0.01 sec	0.0100
3	0.02 sec	0.0200
4	0.03 sec	0.0300
5	0.04 sec	0.0400
6	0.05 sec	0.0500
7	0.06 sec	0.0600
8	0.07 sec	0.0699
9	0.08 sec	0.0799
10	0.09 sec	0.0899
11	0.1 sec	0.0998
12	0.11 sec	0.1098
13	0.12 sec	0.1197
14	0.13 sec	0.1296
15	0.14 sec	0.1395
16	0.15 sec	0.1494
17	0.16 sec	0.1593
18	0.17 sec	0.1692
19	0.18 sec	0.1790
20	0.19 sec	0.1889
21	0.2 sec	0.1987





# Signal Analyzer (ii)







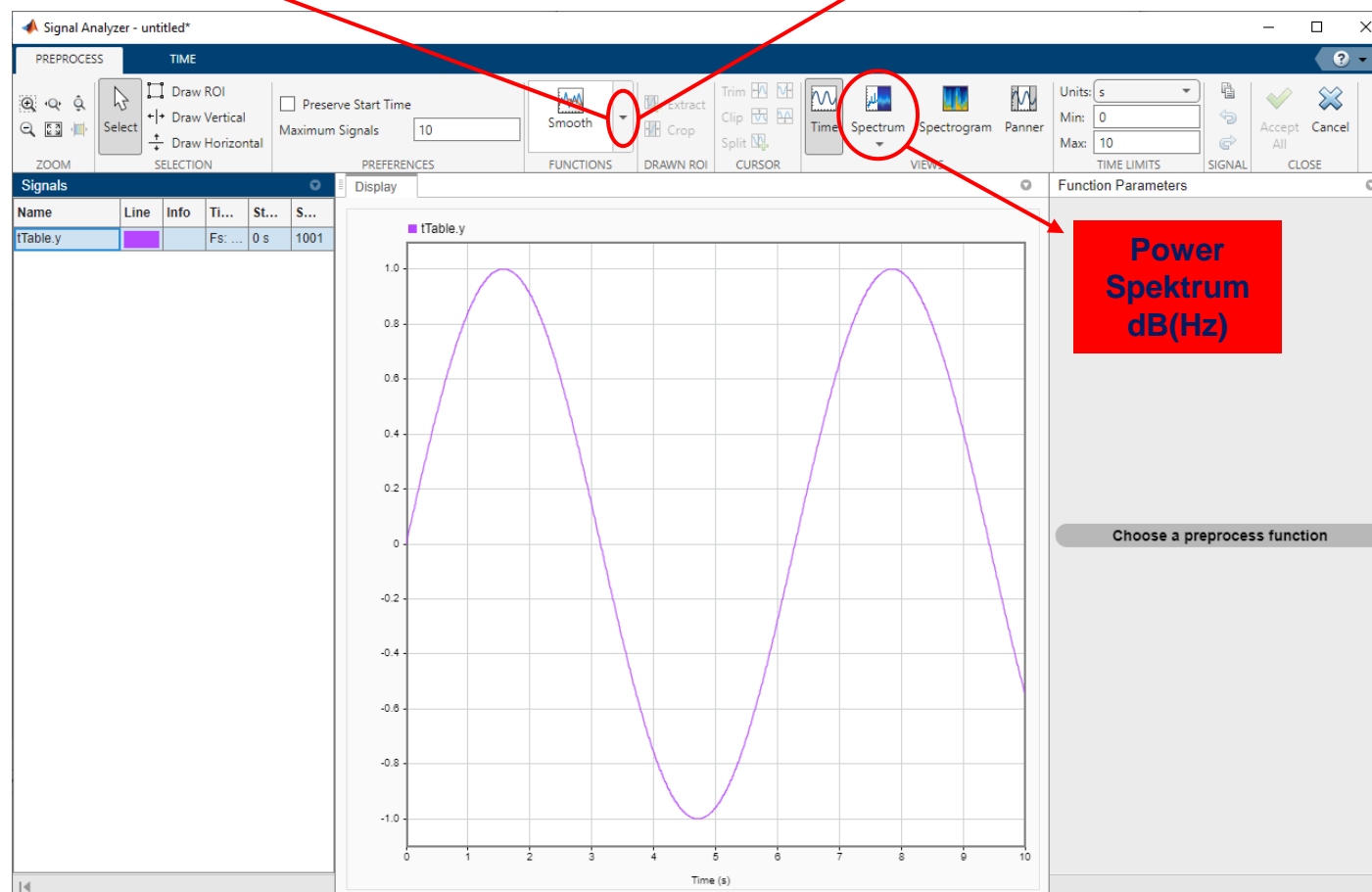
# Signal Analyzer (iii)

**Resample**  
Change sample rate when uniform  
or interpolate when nonuniform

**required if data type not  
consistent (e.g. single vs. double)**

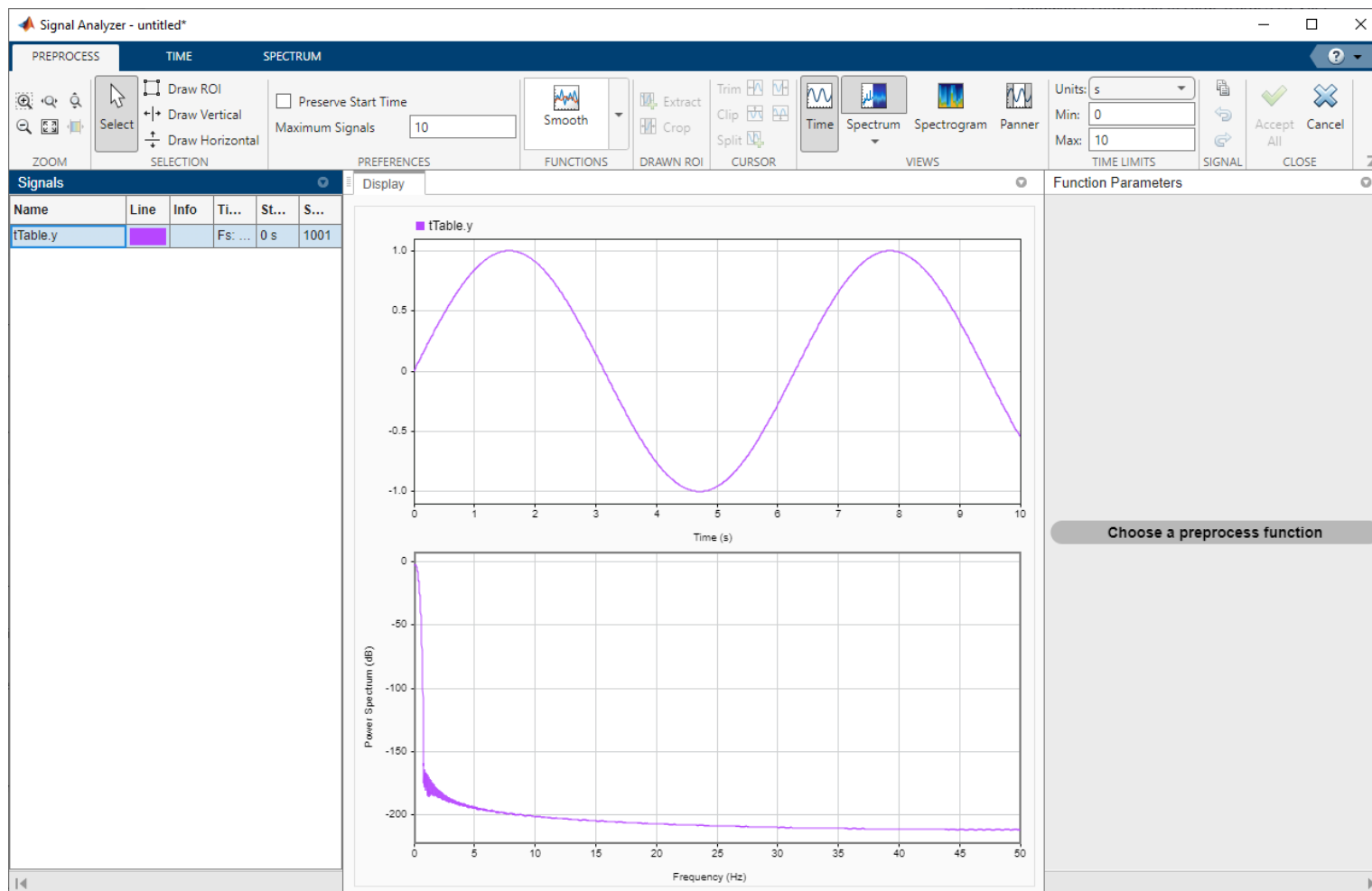
**Detrend**  
Remove constant, linear, or  
piecewise linear trends

**required for e.g.  
offset cleanup**





# Signal Analyzer (iv)





# Signal Analyzer (v)

## Example 1: Resample Function



### Resample

Change sample rate when uniform or interpolate when nonuniform

MATLAB function: `resample()`

generating realistic signal (incl. **data type(s) unequal double**)

```
t = single(0:0.01:10)';
y = sin(t);
```

```
% prepare for SignalAnalyzer
x = seconds(t);
tTable = timetable(x,y);
```

Function Parameters

Resample Parameters

▼ RESAMPLE SETTINGS

Resampling Method

Sample Rate

Frequency Units

Hz

Sample Rate

100

▼ INTERPOLATION

Interpolation Method

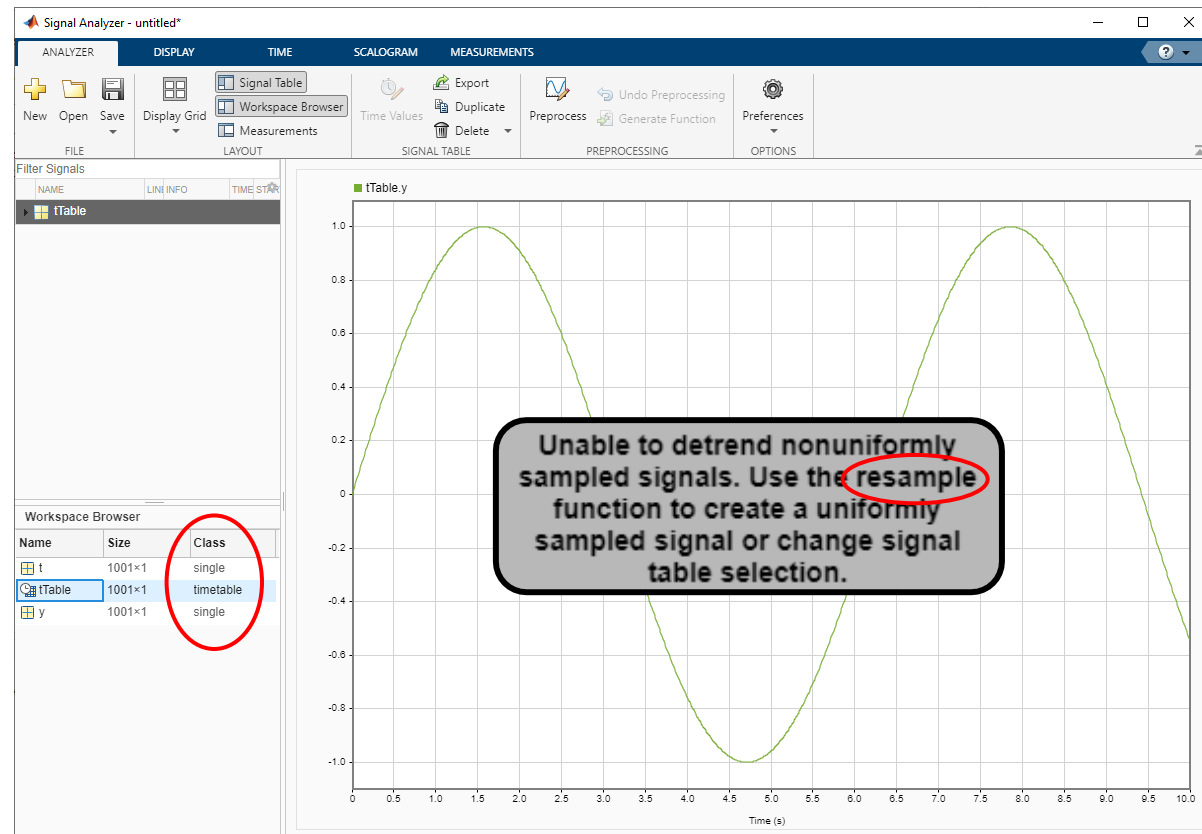
Linear

Frequency Units

Hz

Critical Frequency

Auto





# Signal Analyzer (vi)

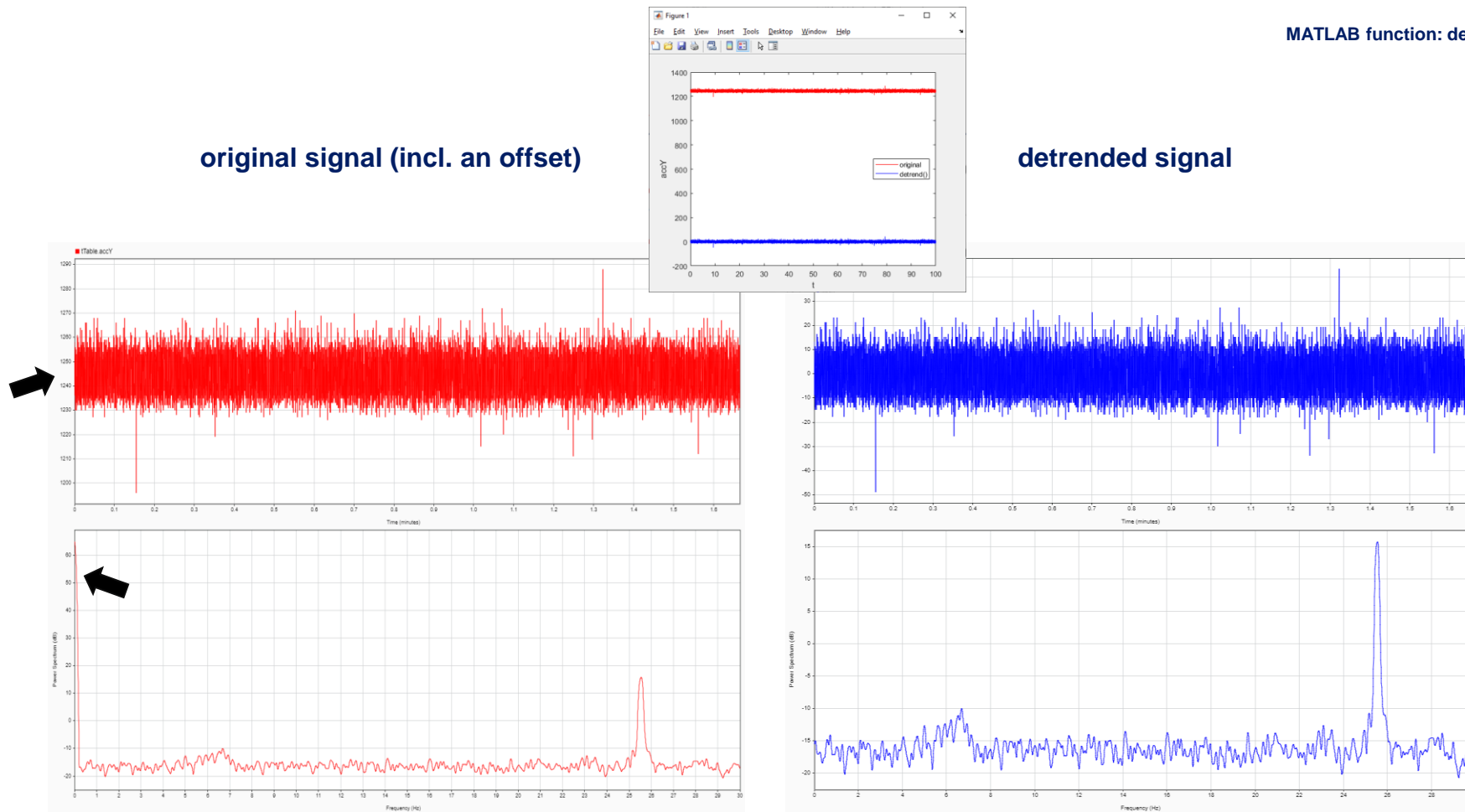
## Example 2: Detrend Function

**Detrend**  
Remove constant, linear, or  
piecewise linear trends

**MATLAB function: detrend()**

original signal (incl. an offset)

detrended signal



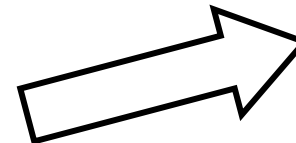


# Diagnostic Feature Designer (i)

input format (table of timetables)

each row represents one measurement

dataTrainEnsemble				
20x4 table				
	1 accX	2 accY	3 accZ	4 mode
1	1001x1 time...	1001x1 time...	1001x1 time...	4
2	1001x1 time...	1001x1 time...	1001x1 time...	3
3	1001x1 time...	1001x1 time...	1001x1 time...	1
4	1001x1 time...	1001x1 time...	1001x1 time...	2
5	1001x1 time...	1001x1 time...	1001x1 time...	2
6	1001x1 time...	1001x1 time...	1001x1 time...	1
7	1001x1 time...	1001x1 time...	1001x1 time...	3
8	1001x1 time...	1001x1 time...	1001x1 time...	4
9	1001x1 time...	1001x1 time...	1001x1 time...	4
10	1001x1 time...	1001x1 time...	1001x1 time...	3
11	1001x1 time...	1001x1 time...	1001x1 time...	1
12	1001x1 time...	1001x1 time...	1001x1 time...	2
13	1001x1 time...	1001x1 time...	1001x1 time...	2
14	1001x1 time...	1001x1 time...	1001x1 time...	1
15	1001x1 time...	1001x1 time...	1001x1 time...	3
16	1001x1 time...	1001x1 time...	1001x1 time...	4
17	1001x1 time...	1001x1 time...	1001x1 time...	4
18	1001x1 time...	1001x1 time...	1001x1 time...	3
19	1001x1 time...	1001x1 time...	1001x1 time...	1
20	1001x1 time...	1001x1 time...	1001x1 time...	2



Diagnostic  
Feature Desi...

dataTrainEnsemble			
dataTrainEnsemble.accY(7,1)			
	Time	1 accY	2
1	0 sec	-6.7157	
2	0.01 sec	3.2842	
3	0.02 sec	0.2842	
4	0.03 sec	0.2842	
5	0.04 sec	3.2841	
6	0.05 sec	-3.7159	
7	0.06 sec	4.2840	
8	0.07 sec	0.2840	
9	0.08 sec	0.2839	
10	0.09 sec	0.2839	
11	0.1 sec	2.2839	
12	0.11 sec	1.2838	



# Diagnostic Feature Designer (ii)

New Session

1

Select dataset from workspace

Source: dataTrainEnsemble

Select All Unselect All

Similar dataset

☒ dataTrainEnsemble

Select source variables

Reset Selection Unselect All

2

3

3

3

Configure source variable properties

Variable name: mode

Variable type: Condition Variable

Unit:

Summary

Ensemble name: Ensemble1

Variable Name	Variable Type	Independent Variable
accX	Signal	Time
accY	Signal	Time
accZ	Signal	Time
mode	Condition Variable	

4

Import Cancel



# Diagnostic Feature Designer (iii)

1

2

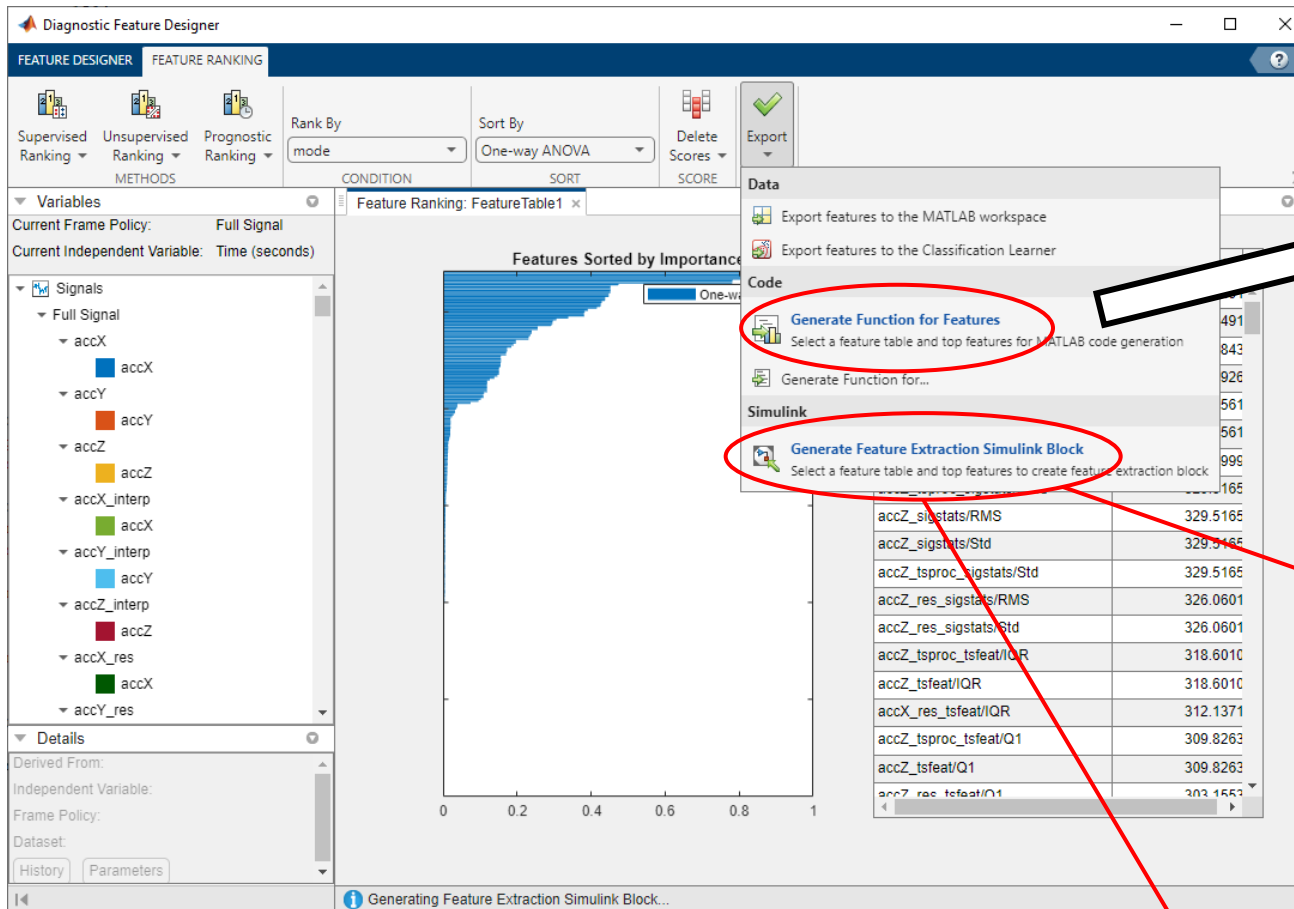
3

The screenshot displays the 'Diagnostic Feature Designer' application. The main window has a dark blue header with the title 'Diagnostic Feature Designer'. Below the header is a toolbar with icons for 'New Session', 'Open Session', 'Save Session', 'Auto Features' (circled in red), 'Plot Options', 'Signal Trace', 'Frame Policy', 'Options', 'Filtering & Averaging', 'Residue Generation', 'Spectral Estimation', 'Time-Domain Features', 'Frequency-Domain Features', 'Rank Features', and 'Export'. The 'Auto Features' button is also labeled with a red '2'. The left sidebar shows a tree view under 'Variables' with 'Full Signal' selected (circled in red). Below this, there are sections for 'Condition Variables' and 'Full Signal' with 'mode' selected. The bottom section shows 'Details' with 'Derived From: Imported', 'Independent Variable: Time (seconds)', 'Frame Policy: Full Signal', and 'Dataset: Ensemble1 (20 Me...'. An 'Auto Features' dialog box is open on the right, showing configuration settings. The 'Configuration' section includes 'Selected variables: accX/accX, accY/accY, accZ/accZ', 'Independent variable: Time', 'Condition variable: mode', 'Frame policy: Full Signal', and 'Feature table: FeatureTable1'. The 'Features to Generate' section has 'Standard' checked and 'Rotating Machinery' unchecked, with a note '261 time-domain, 9 frequency-domain'. The 'Settings' section has 'General' selected and 'Rotating Machinery' unselected. The 'Plot feature histograms' checkbox is unchecked. The 'Number of top features to show' is set to 5. The 'Compute' button is circled in red and labeled with a red '3'.

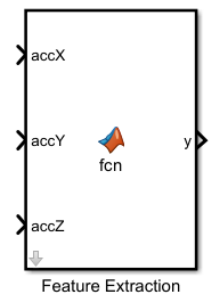
# Diagnostic Feature Designer (iv)



Diagnostic  
Feature Desi...



Classification  
Learner



```
diagnosticFeatures.m  x  +
1  function featureTable = diagnosticFeatures(inputData)
504
```





# Classification Learner (i)

```

diagnosticFeatures.m  x  +
18      % This function computes features:
19      % accY_sigstats/RMS
20      % accY_sigstats/Std
21      % accY_tsfeat/Q1
22      % accY_tsfeat/IQR
23      % accX_tsfeat/Q1
24      % accY_tsproc_tsfeat/Q1
25      % accY_tsproc_tsfeat/IQR
26      % accY_ps_spec/PeakAmp1
  
```



**Export Features To Classification Learner**

Feature table: FeatureTable1

Features sorted by: Kruskal-Wallis

Select features

☐ Select top features: 12

<input checked="" type="checkbox"/>	accY_ps_spec/PeakAmp1
<input checked="" type="checkbox"/>	accY_tsfeat/Q1
<input type="checkbox"/>	accY_res_tsfeat/IQR
<input checked="" type="checkbox"/>	accY_tsproc_tsfeat/Q1
<input type="checkbox"/>	accY_res_tsfeat/Q3
<input type="checkbox"/>	accY_res_tsfeat/Q1
<input checked="" type="checkbox"/>	accY_tsfeat/IQR
<input checked="" type="checkbox"/>	accY_tsproc_tsfeat/IQR
<input checked="" type="checkbox"/>	accY_sigstats/RMS
<input type="checkbox"/>	accX_res_tsfeat/Q1
<input checked="" type="checkbox"/>	accY_sigstats/Std
<input checked="" type="checkbox"/>	accX_tsfeat/Q1
<input type="checkbox"/>	accY_tsproc_sigstats/RMS
<input type="checkbox"/>	accY_tsproc_sigstats/Std

Select condition variables

<input checked="" type="checkbox"/>	mode
-------------------------------------	------



# Classification Learner (ii)

New Session from File

**Data set**

Data Set Variable  
FeatureTable1 64x9 table

Response  
mode double 1 .. 4

**Predictors**

	Name	Type	Range
<input type="checkbox"/>	mode	double	1 .. 4
<input checked="" type="checkbox"/>	accY_sigstats/RMS	double	1.26655 .. 28.6153
<input checked="" type="checkbox"/>	accY_sigstats/Std	double	1.26718 .. 28.6296
<input checked="" type="checkbox"/>	accY_tsfeat/Q1	double	-26.7448 .. -0.360832
<input checked="" type="checkbox"/>	accY_tsfeat/IQR	double	1.04459 .. 51.922
<input checked="" type="checkbox"/>	accX_tsfeat/Q1	double	-8.79928 .. -0.663183

Add All Remove All

[How to prepare data](#)

**Validation**

Validation Scheme  
Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds 5

[Read about validation](#)

**Test**

☐ Set aside a test data set

Percent set aside 10

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

[Read about test data](#)

Response variable is numeric. Distinct values will be interpreted as class labels.

Start Session Cancel



# Classification Learner

## Classification Learner (iii)

Classification Learner for Predictive Maintenance - untitled\*

**CLASSIFICATION LEARNER**

FILE: New Session, Open, Save, Feature Selection, PCA, Costs, Optimizer

MODELS: All Quick-To-Train, **All** (1), Summary, Duplicate, Delete, Results Table, Use Parallel, **Train All** (2), Scatter, Confusion Matrix, Layout

TEST: Test Data, Test All, Export Plot to Figure, Generate Function, Export Model

**Models**

Model Number	Model Name	Status	Last change	Features
1	Tree	Draft	Fine Tree	8/8 features
2	Multiple	Draft	All	8/8 features

**Model 2: All**  
Status: Draft

**Model Hyperparameters**

This model does not have hyperparameter options.

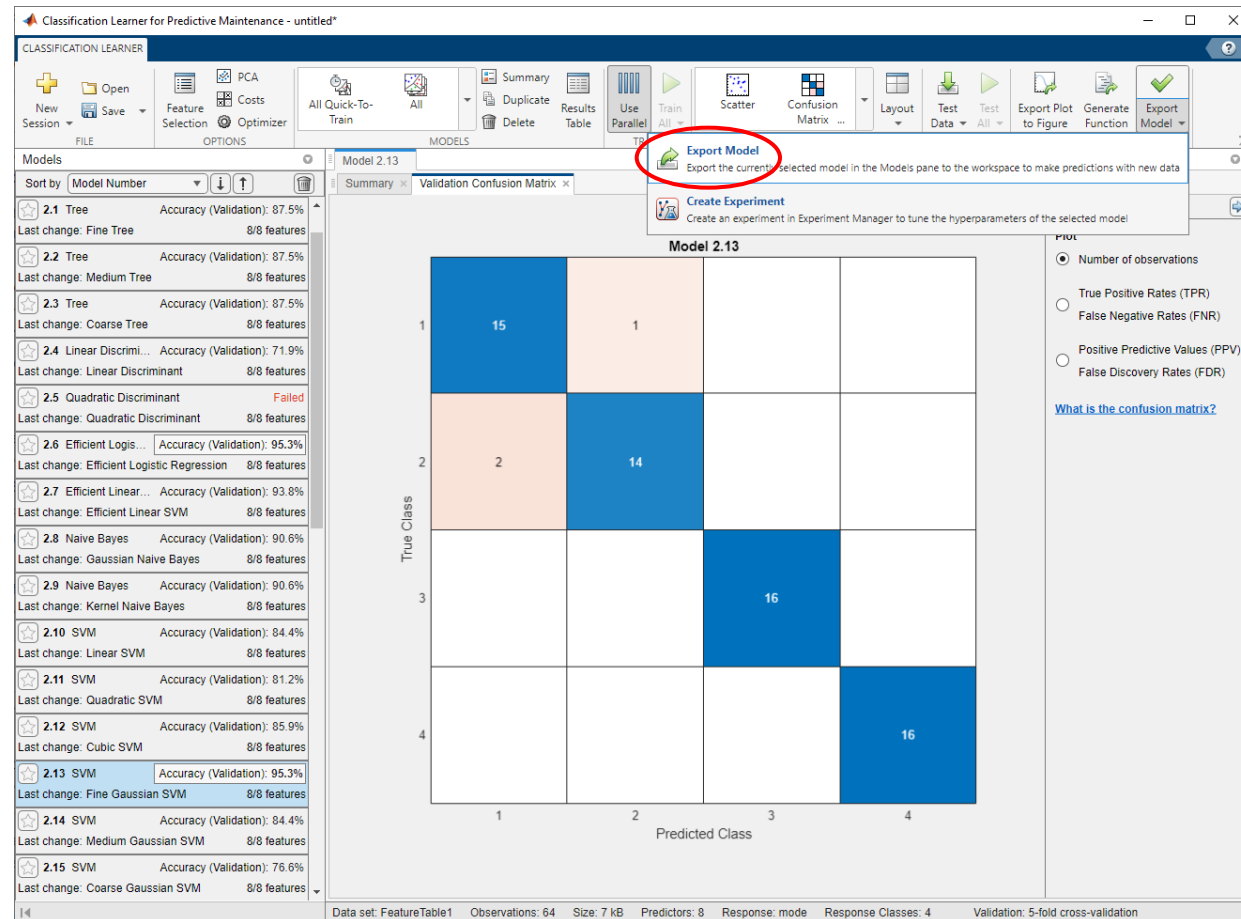
- Feature Selection: 8/8 individual features selected
- PCA: Disabled
- Misclassification Costs: Default
- Optimizer: Not applicable

Data set: FeatureTable1   Observations: 64   Size: 7 kB   Predictors: 8   Response: mode   Response Classes: 4   Validation: 5-fold cross-validation



# Classification Learner (iv)

chose a model and export it to MATLAB workspace






# Classification Learner (v)

commands to use trained models in MATLAB/Simulink:

a) `saveLearnerForCoder(trainedModel.Classification***, 'trainedModel');`

e.g. `saveLearnerForCoder(trainedModel.ClassificationSVM, 'trainedModel');`



trainedModel	
1x1 struct with 5 fields	
Field	Value
predictFcn	@(x)exportableModel.predictFcn(p
RequiredVariables	1x8 cell
ClassificationSVM	1x1 ClassificationECOC
About	'This struct is a trained model expo
HowToPredict	1x552 char

b) `classifier = loadLearnerForCoder("trainedModel.mat");`

c) `mode = predict(classifier, featureTable);`

# ThingSpeak (i)

---

Sign in to [ThingSpeak](#) using your MathWorks Account → New Channel

## New Channel

Name

Condition Monitoring

Description

Field 1

mode



Save Channel

# ThingSpeak (ii)

## Condition Monitoring

Channel ID: **2485082**  
 Author: **mwa0000032484217**  
 Access: Private

Private View   Public View   Channel Settings   Sharing   API Keys   Data Import

## Write API Key

Key

1SY126GB2QX4EQYD

Generate New Write API Key

copy Write API Key



Block Parameters: ThingSpeak Write

ThingSpeak Write

Send data to ThingSpeak, MathWorks IoT Analytics Platform. The hardware must be connected to the Internet. Input ports only accept numeric scalar signals.

Enter the Write API Key for your ThingSpeak channel. Each input variable is written to a field in your ThingSpeak channel. Set Update interval to the number of seconds to wait between two successive data send requests. Visit your ThingSpeak account page to find the minimum update interval for your channel.

Check "Print diagnostic messages" to send the ThingSpeak server response to standard output on the hardware.

Main   **Optional**

Write API key:

Number of variables to send:

Update interval:

☒ Print diagnostic messages

OK

Cancel

Help

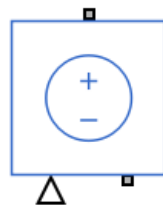
Apply

# Simscape – Add-Ons (MATLAB R2024a)

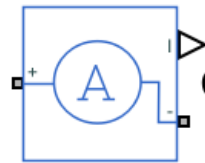
Add-Ons	Requires
Simscape <small>version 24.1</small>	Simulink <small>version 24.1</small>
Simscape Driveline <small>version 24.1</small>	Simulink <small>version 24.1</small> Simscape <small>version 24.1</small>
Simscape Electrical <small>version 24.1</small>	Simulink <small>version 24.1</small> Simscape <small>version 24.1</small>



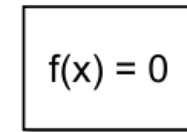
# Simscape – Electrical Components



Controlled Voltage Source



Current Sensor



Solver Configuration



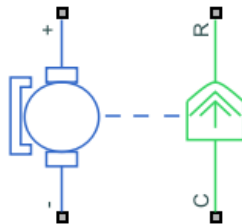
Resistor



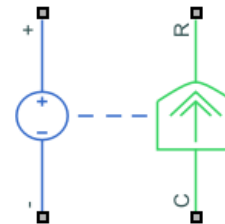
Inductor



Electrical Reference

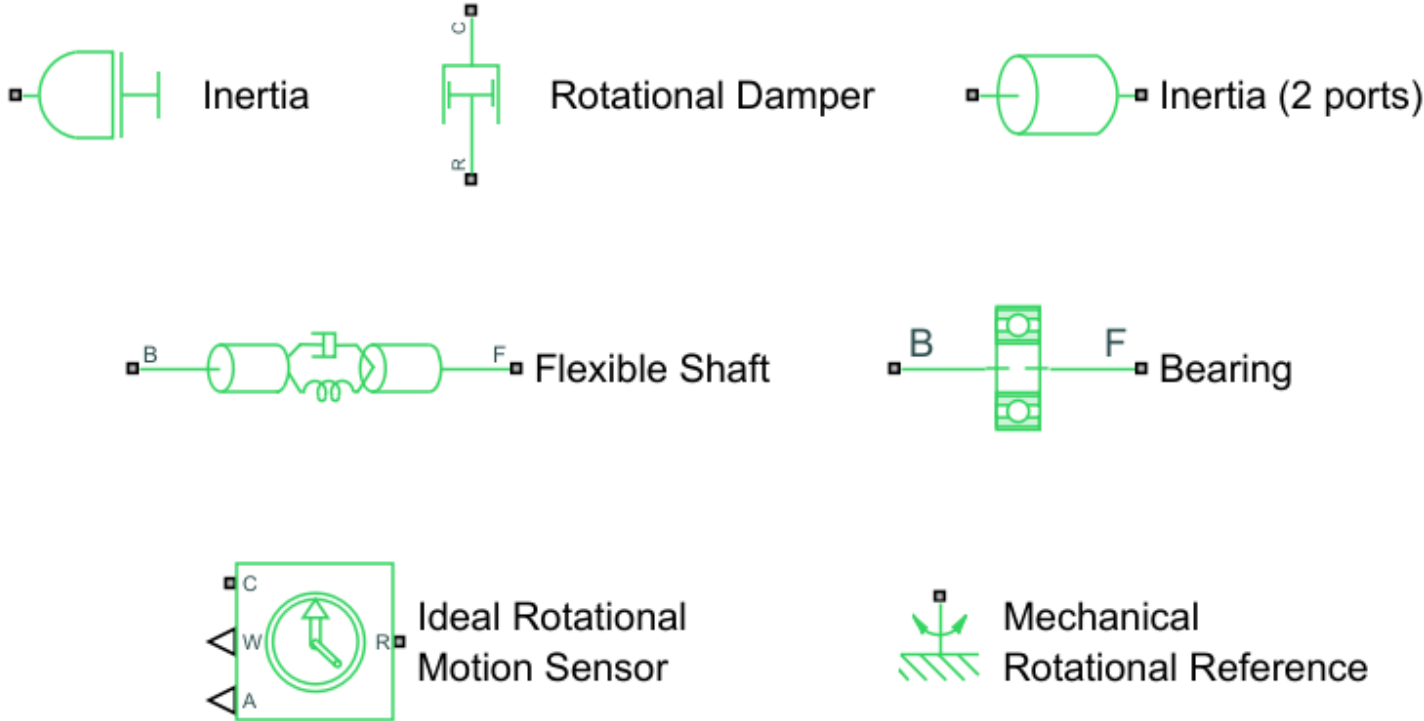


DC Motor

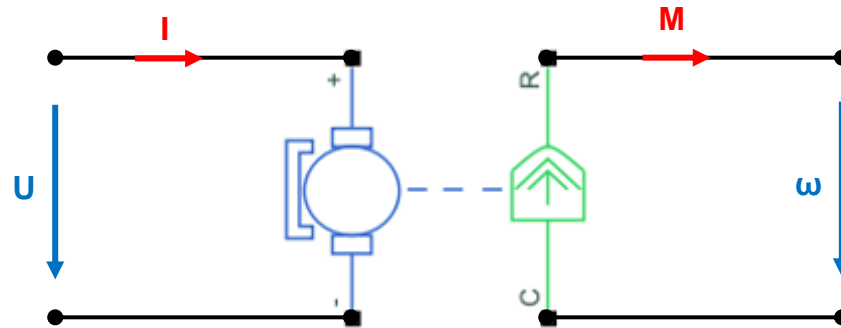


Rotational Electromechanical Converter

# Simscape – Mechanical Components



# Simscape – Interfaces



Simulink-PS  
Converter



PS-Simulink  
Converter