

MATLAB EXPO

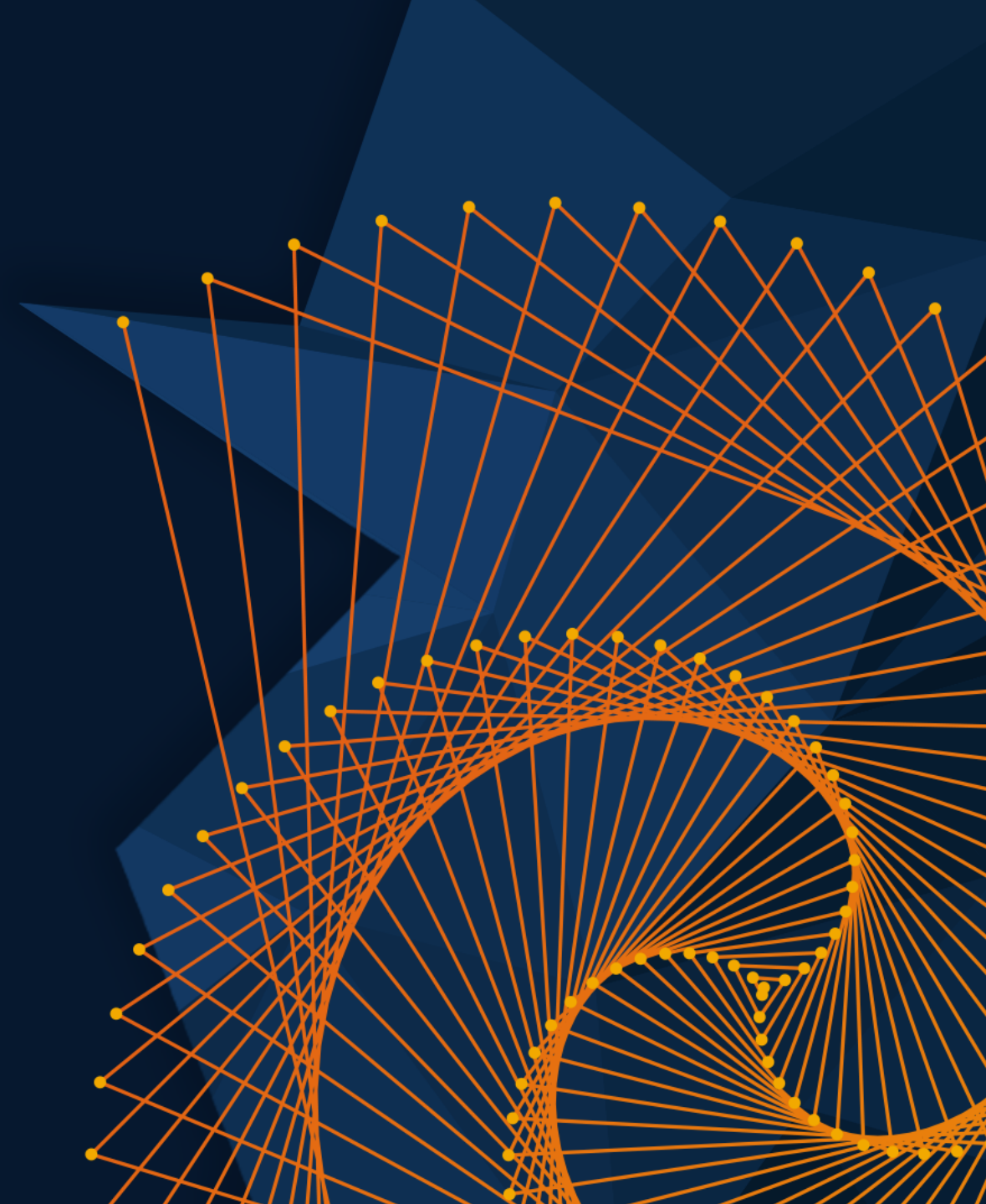
November 13–14, 2024 | Online

Modeling Dynamic Systems with MATLAB and Simulink

Kishen Mahadevan, MathWorks



Brian Douglas, MathWorks



Models are crucial in any engineering project

Do you want to:

Design complex systems?

Simulate and test your system early and often?


Analyze and validate your design?

Optimize system performance?


You need a model

And tools and methods for
creating those models


There are many different capabilities that can help you model dynamic systems

MathWorks **Modeling Dynamic Systems with MATLAB and Simulink** Explore more capabilities for modeling dynamic systems → 

Model Structures Use MATLAB® and Simulink® to support linear and nonlinear model structures, including integration of third-party models.

Linear Models	Nonlinear Models	Integrate Third-Party Models
<p>TRANSFER FUNCTION</p> $G(s) = \frac{\omega^n}{s^2 + 2\zeta\omega s + \omega^2}$ <p>LINEAR PARAMETER VARYING</p> <p>FREQUENCY RESPONSE DATA</p> 	<p>STATE SPACE</p> $\dot{x} = Ax + Bu$ $y = Cx + Du$ <p>TIME SERIES (ARX, ARMA)</p> <p>ZPK</p> <p>ODEs</p> $\frac{dy}{dt} = Q(t)y + R(t)y^2$ <p>NONLINEAR MODELS</p> <p>Nonlinear portion could be represented with AI</p> <p>AI-BASED MODELS</p> <p>HAMMERSTEIN-WIENER</p> <p>GAUSSIAN PROCESS</p> <p>SUPPORT VECTOR MACHINE</p> <p>REGRESSION TREE</p> <p>NEURAL NETWORK</p> <p>NEURAL STATE SPACE</p> <p>NLARX</p>	<p>FUNCTIONAL MOCKUP UNIT</p> <p>PYTHON IMPORTERS</p> <p>py.</p> <p>S-FUNCTIONS</p> <p>FMU IMPORT</p> <p>SYSTEM</p>

Model Parameters Determine model parameters through first principles, grey box, and data-driven methods.

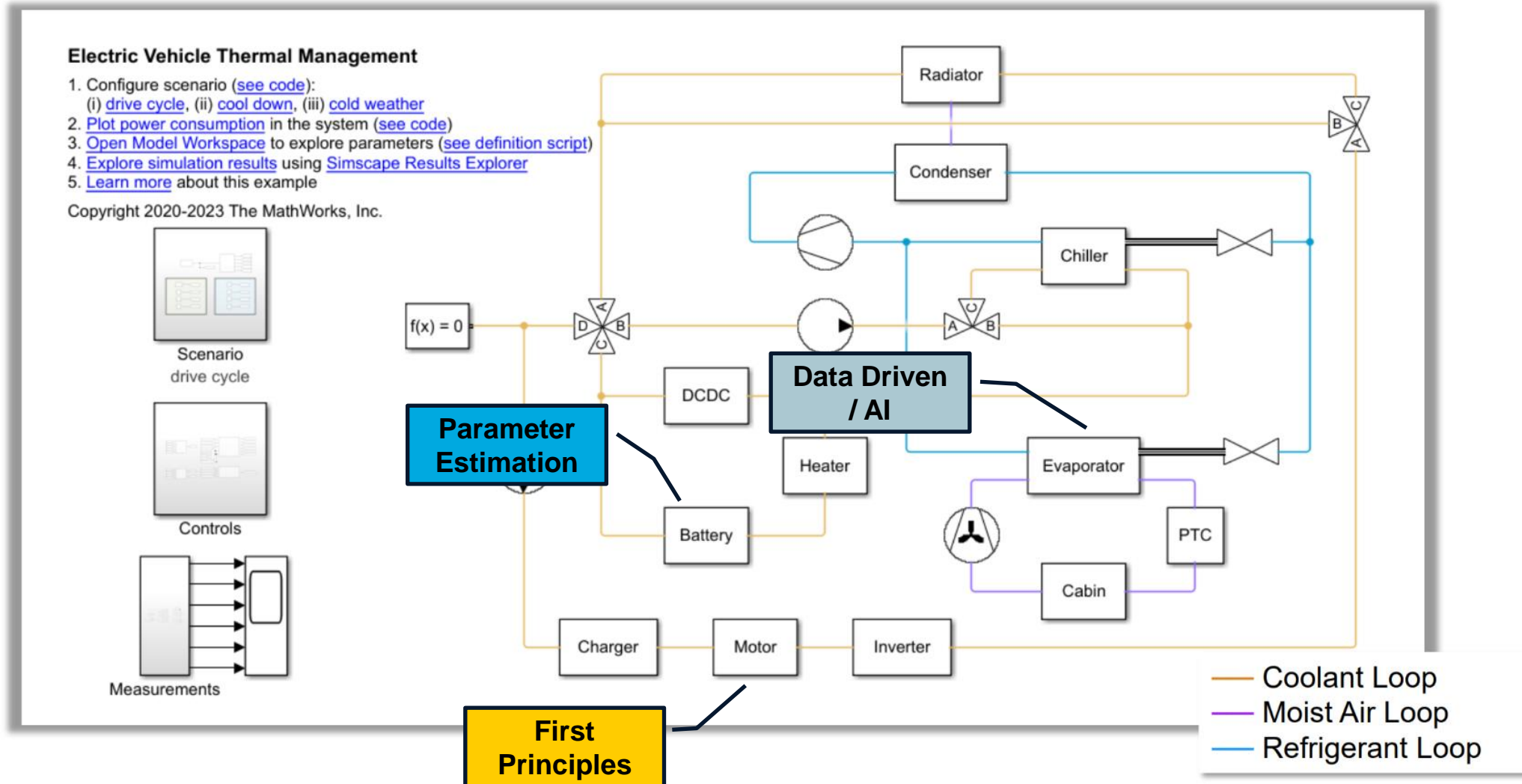
MODELING BASED ON PHYSICAL LAWS	GREY BOX	MODELING BASED ON SYSTEM DATA
WHITE BOX		BLACK BOX
<p>FIRST PRINCIPLES</p> $f(x, t) = m \frac{d^2x}{dt^2}$ <p>PHYSICAL MODELING WITH SIMSCAPE</p> 	<p>GREY BOX ODEs</p> $\frac{dx(t)}{dt} = \begin{bmatrix} a & 1 \\ 0 & b \end{bmatrix} x(t) + \begin{bmatrix} c \\ d \end{bmatrix} u(t)$ <p>PARAMETER ESTIMATION IN SIMULINK MODELS</p> <p>PARAMETER ESTIMATOR</p>	<p>SYSTEM IDENTIFICATION (TRADITIONAL AND AI-BASED)</p> <p>SYSTEM IDENTIFICATION</p> <p>ONLINE ESTIMATION</p> <p>MODEL ANALYSIS</p> <p>DATA PREPARATION</p> <p>OFFLINE ESTIMATION</p>

Model Manipulation Modify models through transformation, linearization, and order reduction methods.

Model Transformation	Linearization	Reduced Order Modeling
<p>MODEL TYPE</p> $\frac{\omega^n}{s^2 + 2\zeta\omega s + \omega^2} \leftrightarrow \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$ <p>CONTINUOUS-DISCRETE</p> $f(t) \leftrightarrow F[k]$ <p>STATE-COORDINATE</p> $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \leftrightarrow \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ <p>MODAL DECOMPOSITION</p> $G(z) = H_0 + H_1(z) + H_2(z)$	<p>NUMERICAL PERTURBATION</p> <p>BLOCK-BY-BLOCK</p> <p>FREQUENCY RESPONSE ESTIMATION</p> <p>MODEL LINEARIZER</p>	<p>MODEL-BASED</p> <p>BALANCED TRUNCATION</p> <p>MODEL REDUCER</p> <p>POLE-ZERO SIMPLIFICATION</p> <p>DATA-DRIVEN</p> <p>Use model data to learn a lower order model</p> <p>REDUCED ORDER MODELER</p>

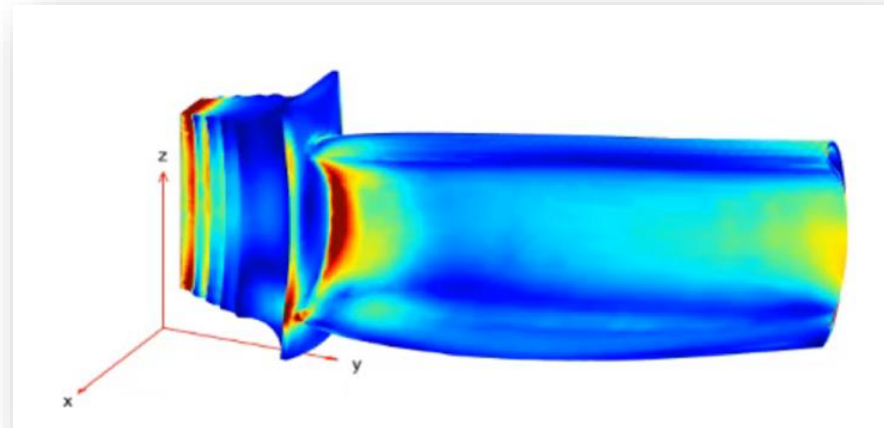
© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders. 2/2024

Components could be modeled with different methods



Models can **change form** depending on the use case.

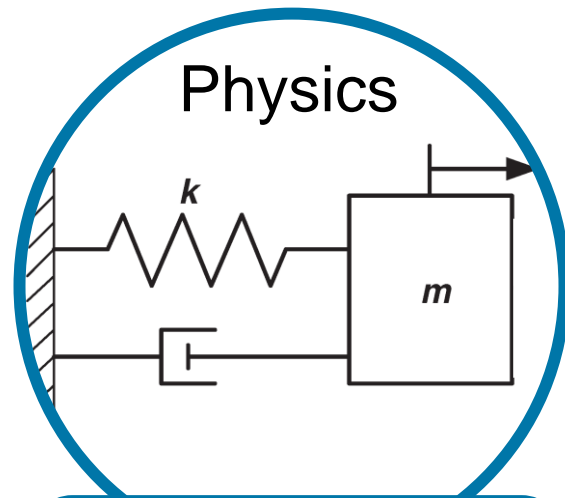
High fidelity model



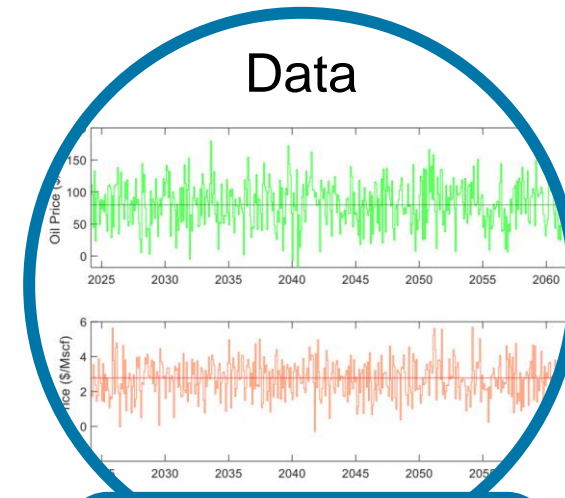
Reduced order modeling

Real-time simulation with HIL testing

Where do you get the information to create a model?



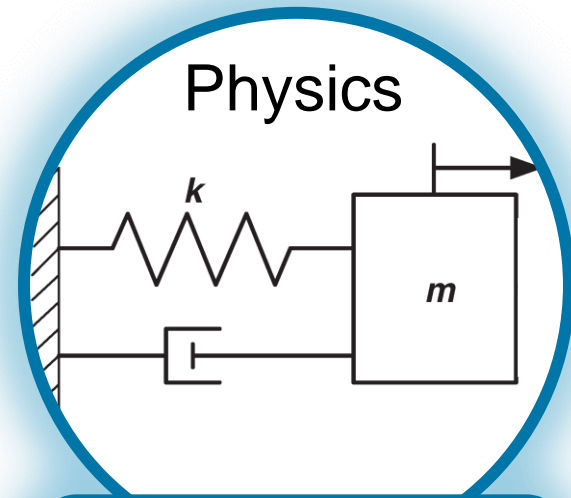
First principles modeling



Data-driven modeling



Where do you get the information to create a model?

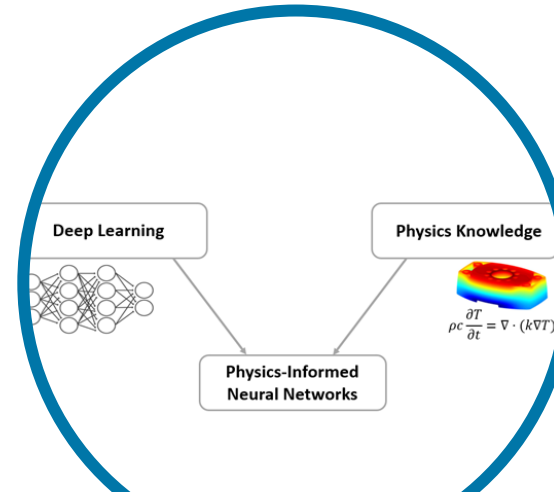


Physics

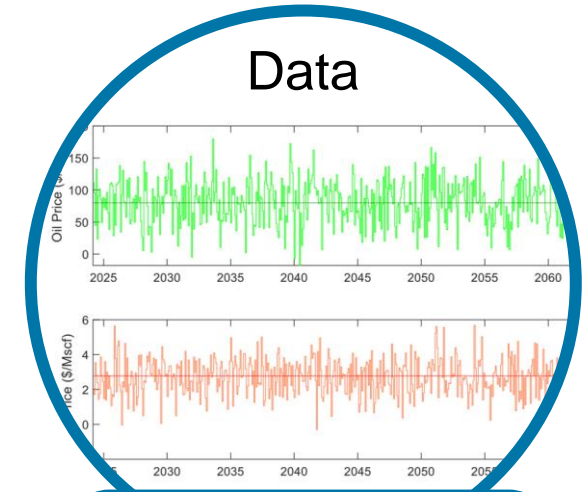
First principles modeling

$$m\ddot{x} + b\dot{x} + kx = 0$$

Parameter estimation



Physics-Informed ML



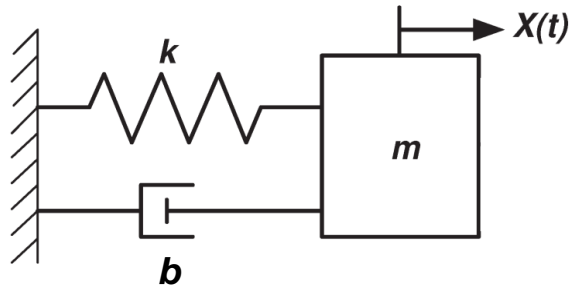
Data

Data-driven modeling

WHITE BOX

BLACK BOX

With first principles modeling you build models that are based on physical laws

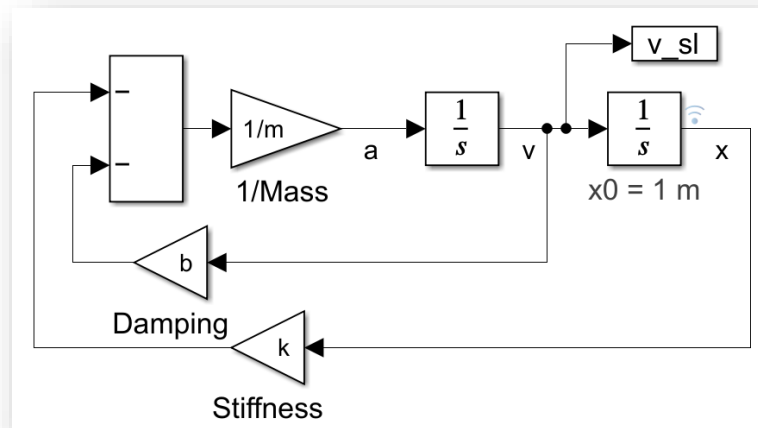


$$m\ddot{x} + b\dot{x} + kx = 0$$

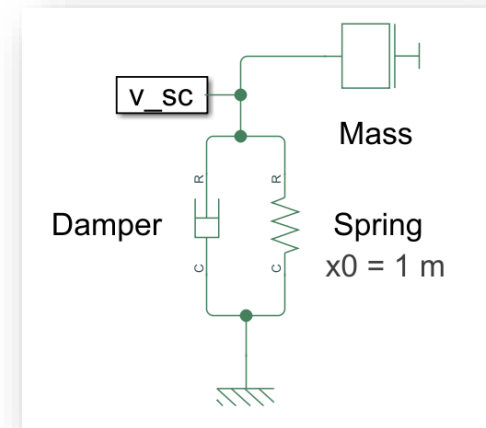
Text-based code

```
msd.m  x  +
1 % Mass-Spring-Damper System Simulation
2
3 % Parameters
4 m = 1.0; % Mass (kg)
5 b = 0.5; % Damping coefficient (Ns/m)
6 k = 10.0; % Spring constant (N/m)
7
8 % Initial conditions
9 x0 = 1.0; % Initial displacement (m)
10 v0 = 0.0; % Initial velocity (m/s)
11 initial_conditions = [x0; v0];
12
13 % Time span for the simulation
14 t_span = [0 10]; % From 0 to 10 seconds
15
16 % Define the system of ODEs
17 % dx/dt = v
18 % dv/dt = -(b/m)*v - (k/m)*x
19 mass_spring_damper = @(t, y) [y(2); -(b/m)*y(2) - (k/m)*y(1)];
20
21 % Solve the ODE
22 [t, y] = ode45(mass_spring_damper, t_span, initial_conditions);
23
```

Executable block diagrams

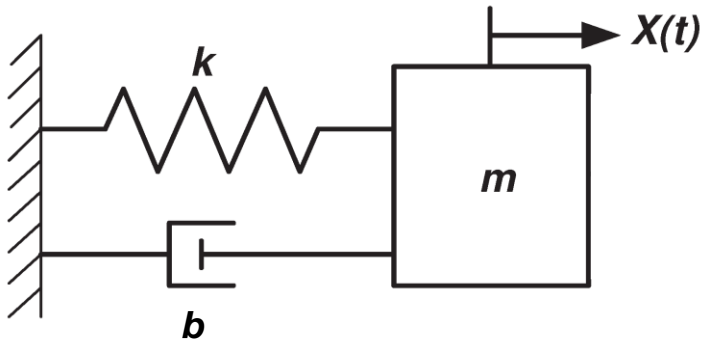


Physical modeling



There are benefits to first principles modeling

Insight and interpretability



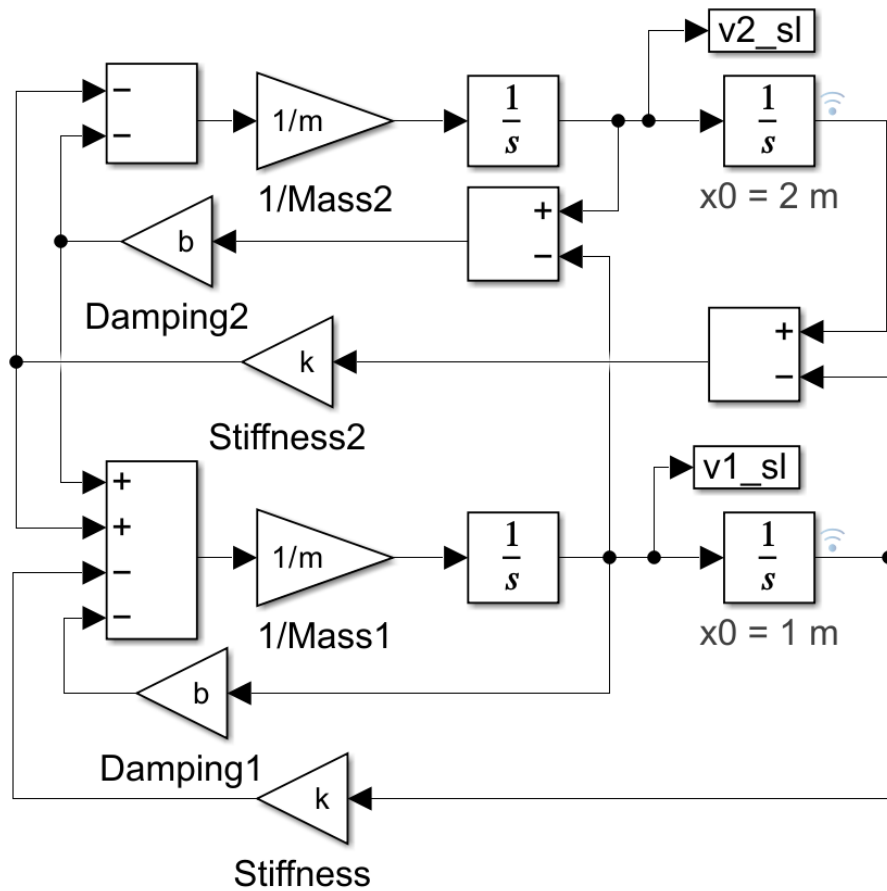
First principles model

$$m\ddot{x} + b\dot{x} + kx = 0$$

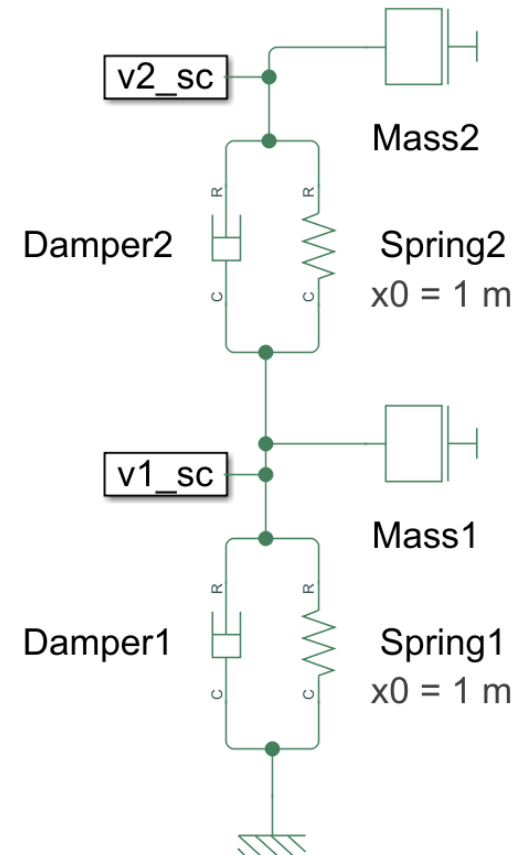
This is understandable

Physical modeling of first principles can be even more interpretable than equations

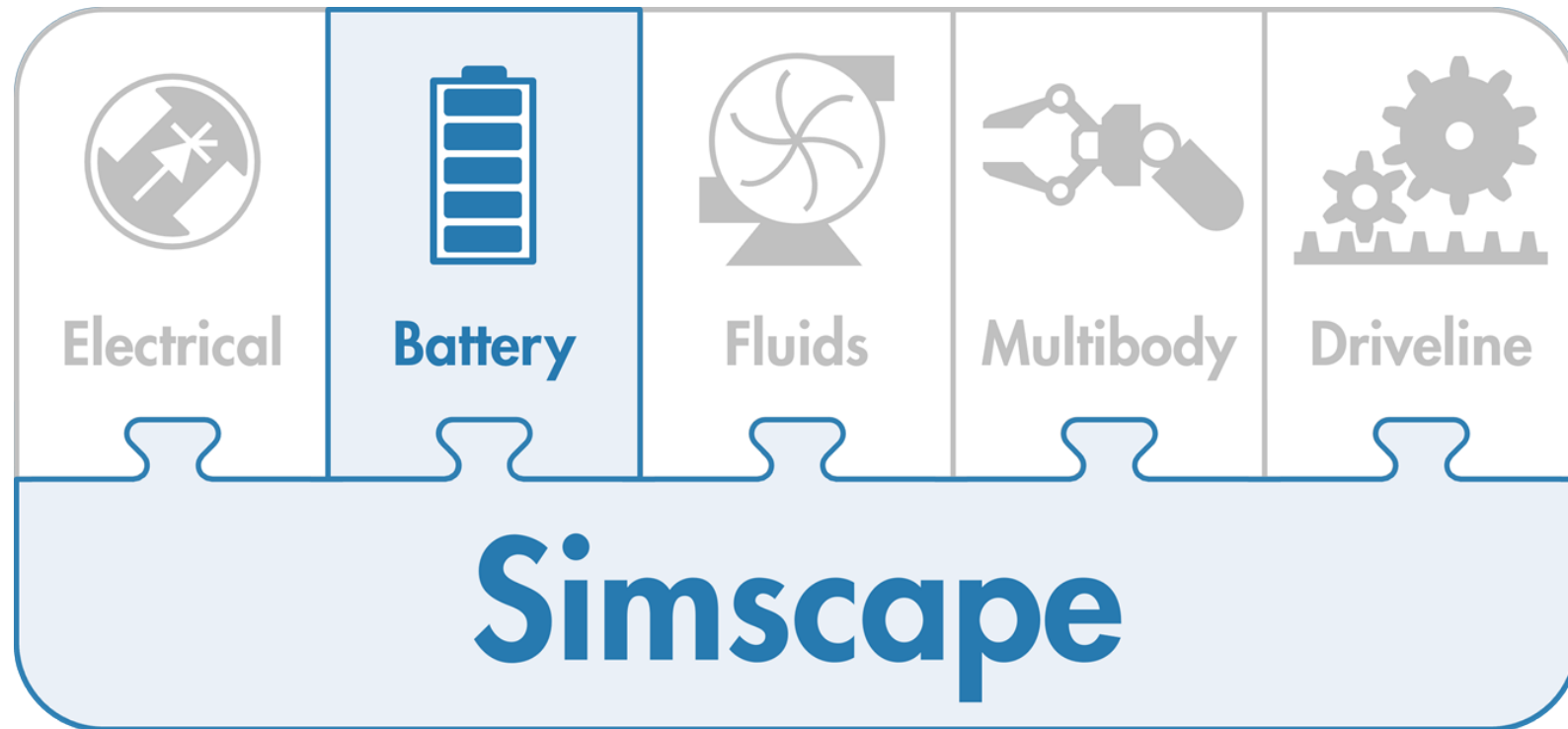
Simulink Model



Simscape Model



Spring-mass-dampers are good examples to learn but what about more complex dynamics?



Build a detailed model of a battery pack easily with physical modeling

Build Detailed Model of Battery Pack from Cylindrical Cells

This example shows how to create and build Simscape™ system models for various battery designs and configurations based on cylindrical battery cells in Simscape™ Battery™. The `buildBattery` function allows you to automatically generate Simscape models for these Simscape Battery objects:

- `ParallelAssembly`
- `Module`
- `ModuleAssembly`
- `Pack`

This function creates a library in your working folder that contains a system model block of a battery pack. Use this system model block to create a battery pack. The open-circuit voltage, are defined after the model creation and are therefore not covered by the Battery Pack Builder `MaskParameters` argument of the `buildBattery` function.

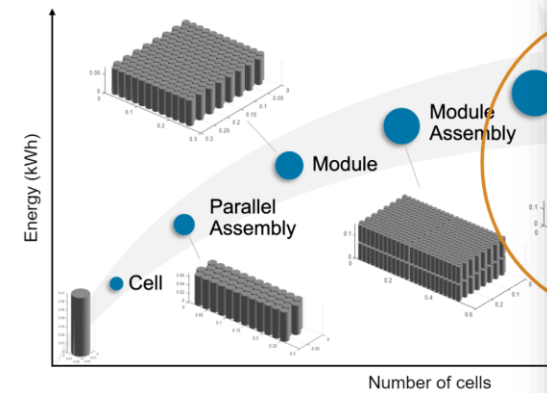
During the first half of this example, you first define the key properties of a cylindrical battery cell and block model. You also call a "sub-module", a "super-cell", a "P-set", or just a "cell". You later employ this parallel assembly to define a fundamental repeating unit. Throughout the workflow, you visualize the geometry and the relative positioning of the battery objects in the workspace.

In the second half of the example, you modify the modeling methodology and the model resolution of the `Module`, the stacking of any battery object along the sequence either along the X or Y axis. These axis mirror the Coordinate System

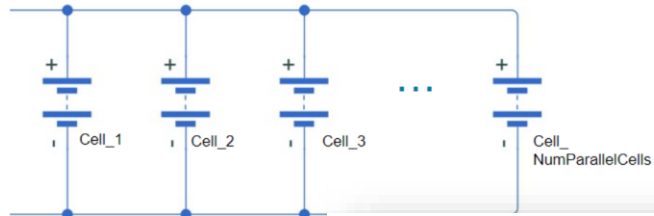
Create and Visualize Battery Objects in MATLAB

To create a battery pack, you must first design and create the foundational elements of the battery pack.

This figure shows the overall process to create a battery pack object in a bottom-up approach:



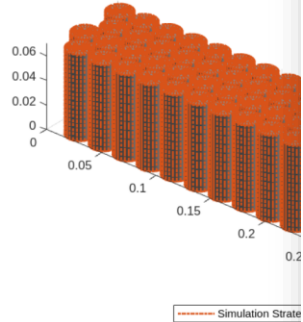
If you set the `ModelResolution` property of the parallel assembly to "Detailed", the `ParallelAssembly` object instantiates a number of cell model blocks equal to the value of the `NumParallelCells` property and connects them electrically in parallel in Simscape.



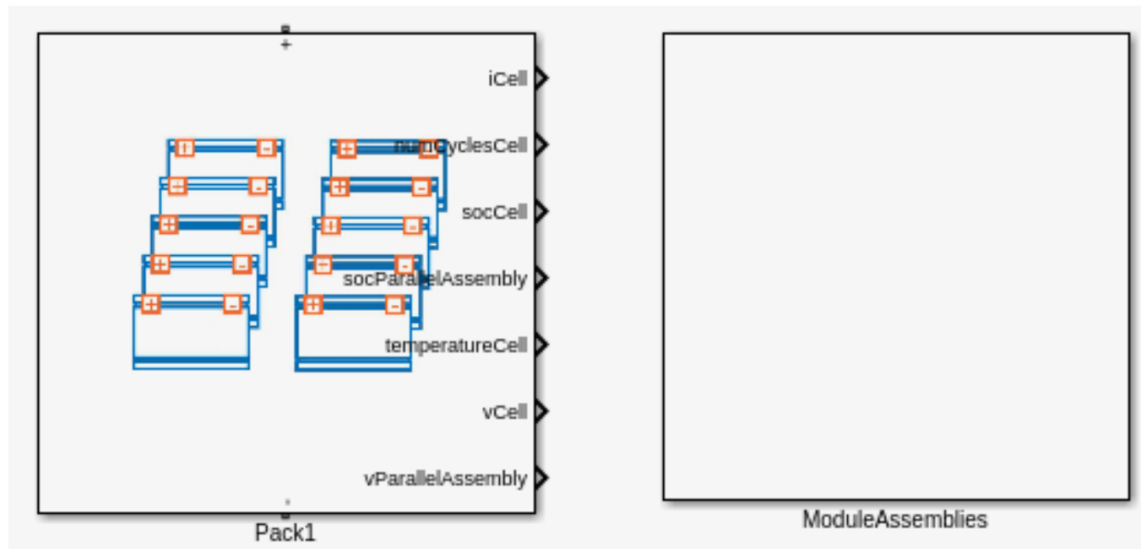
Change the model resolution of the previous `ParallelAssembly` object. Set the `SimulationStrategyVisible` specified as "on".

```

detailedParallelAssembly = lumpedParallelAssembly
detailedParallelAssembly.ModelResolution = "Detailed"
f = figure("Color","white");
detailedParallelAssemblyChart = batteryChart(f,detailedParallelAssembly);
    
```



The `cylindricalPackExample` library contains the Simscape models of your `ModuleAssembly` and `Pack` objects.



First principles modeling is used by our customers for a variety of applications



[Link to User Story](#)

Virtual Design and Testing of an Autonomous Rescue Drone Speeds Up Product Development

Simscape™ was used for detailed models of the subsystems, including the electric powertrain with battery, intermediate circuit, inverter, and engine.



Image credit: SEGULA Technologies

[Link to User Story](#)

This Clean Power Source Is Helping Fuel the Future of Transportation



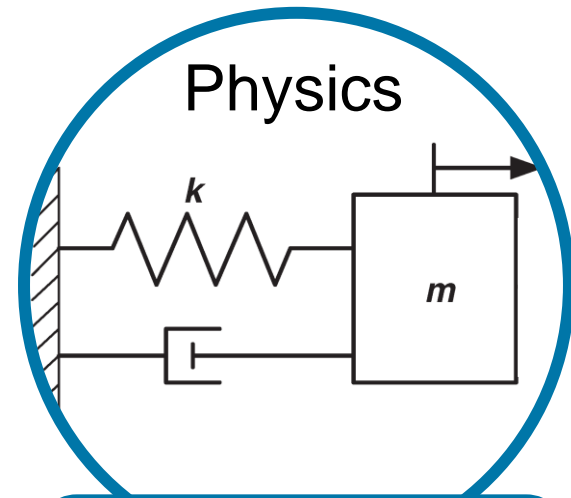
Starting with a Simscape model shaves four to six weeks off of the initial development time.

-Dirk Rensink

Technical lead for fuel cell simulation, SEGULA Technologies



Where do you get the information to create a model?

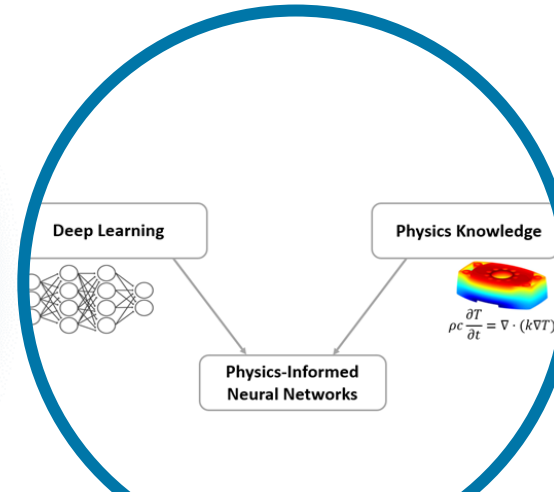


Physics

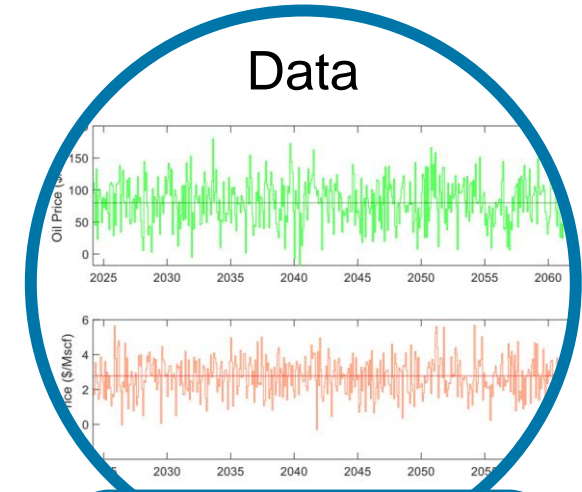
First principles modeling

$$m\ddot{x} + b\dot{x} + kx = 0$$

Parameter estimation



Physics-Informed ML



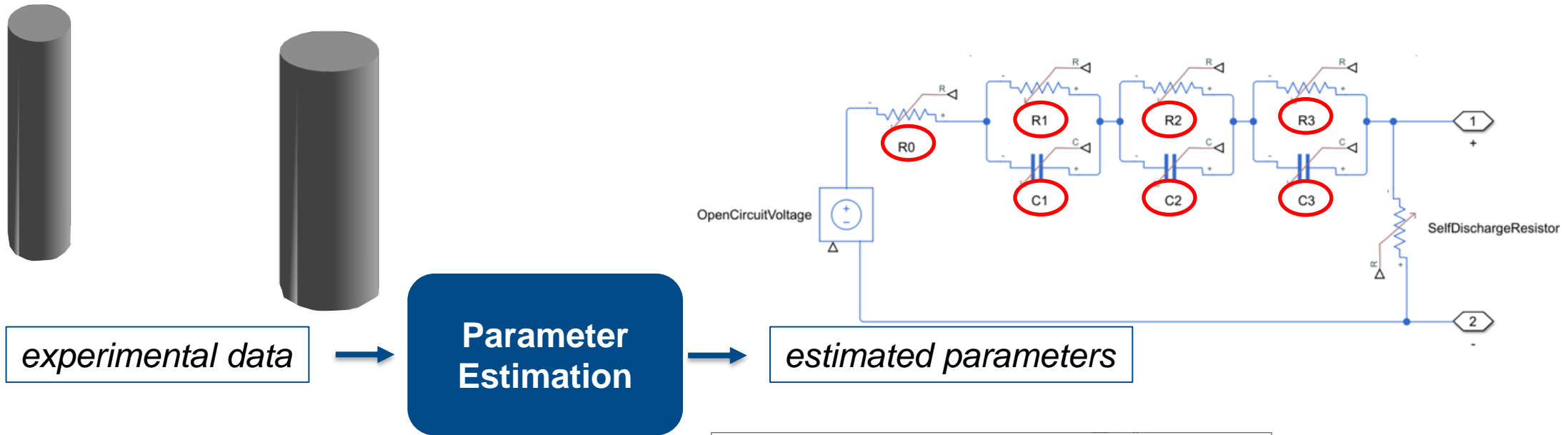
Data

Data-driven modeling

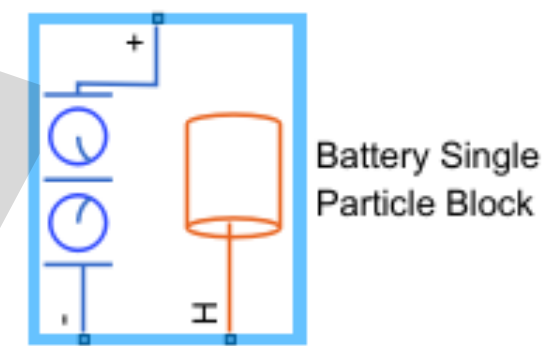
WHITE BOX

BLACK BOX

With parameter estimation you identify physically meaningful parameters using system equations and measured data



NAME	VALUE
Geometry	
Electrodes Properties	
> Volume fraction of anode active material	0.58
> Volume fraction of cathode active mater...	0.374
> Anode open-circuit potential	[484; 335; 253; 209; 2; 181; 165; 1...
> Cathode open-circuit potential	[3.81; 3.608; 3.512; 3.478; 3.459; 3.453; 3...
> Normalized stoichiometry breakpoints	linspace(0.025, 0.975, 39)
> Anode maximum ion concentration	30555 mol/m ³
> Anode maximum stoichiometry	0.831



Parameter estimation with Simulink Design Optimization

Specify Experiment

The screenshot displays the Simulink Design Optimization environment. The central workspace shows a battery model with a probe connected to its terminal voltage. A red arrow labeled "signal" points to the probe output. The "Edit Signal Matching: SignalMatching" dialog box is open, showing the "Outputs" section with a table of output signals. The "Configure Measured Data" section is also visible, with the "Time" and "Data" dropdown menus highlighted by a red box. The "measured test data" section shows the data updated at 2:23:17 pm.

Output Signal	Text Entered
sdoSPMBattery/Probe:1 (terminal voltage (V))	[VoltageMeasTime(:), VoltageMeasData(:)]

measured test data
Data updated (2:23:17 pm)

Parameter estimation with Simulink Design Optimization

Specify Experiment

Select Parameters

Parameters Tuned for All Experiments

Select Parameters

Parameter	Initial Value	Estimate
CDC	2.63545798543486e-19	Yes
CPR	5.89191365922431e-08	Yes
ADC	3e-15	No
APR	5e-06	No

Configure: CDC

- Estimate
- Initial Value: 2.635457... [dropdown] [edit icon]
- Minimum: 2.54e-19 [dropdown] [edit icon]
- Maximum: 1.3664e-18 [dropdown] [edit icon]
- Scale: 5.421010... [dropdown] [edit icon]

Parameters and Initial States Tuned per Experiment

Experiment: VoltageMatching

Select experiment initial states for estimation
There are no initial states defined for this experiment.

Select experiment parameters for estimation
There are no parameters defined for this experiment.

Help Update Model Close

Parameter Estimator* - sdoSPMBattery

PARAMETER ESTIMATION VALIDATION EXPERIMENT PLOT

Open Save New Select Select Sensitivity Add Plot Plot Model Cost Function: Sum Squared Error Estimate
Session Session Experiment Experiments Parameters Analysis More Options ESTIMATE

Parameters: CDC, CPR, ADC, APR

Experiments: VoltageMatching

Results

Preview

Experiment plot: VoltageMatching

VoltageMatching terminal voltage (V)

Amplitude

Time (seconds) $\times 10^4$

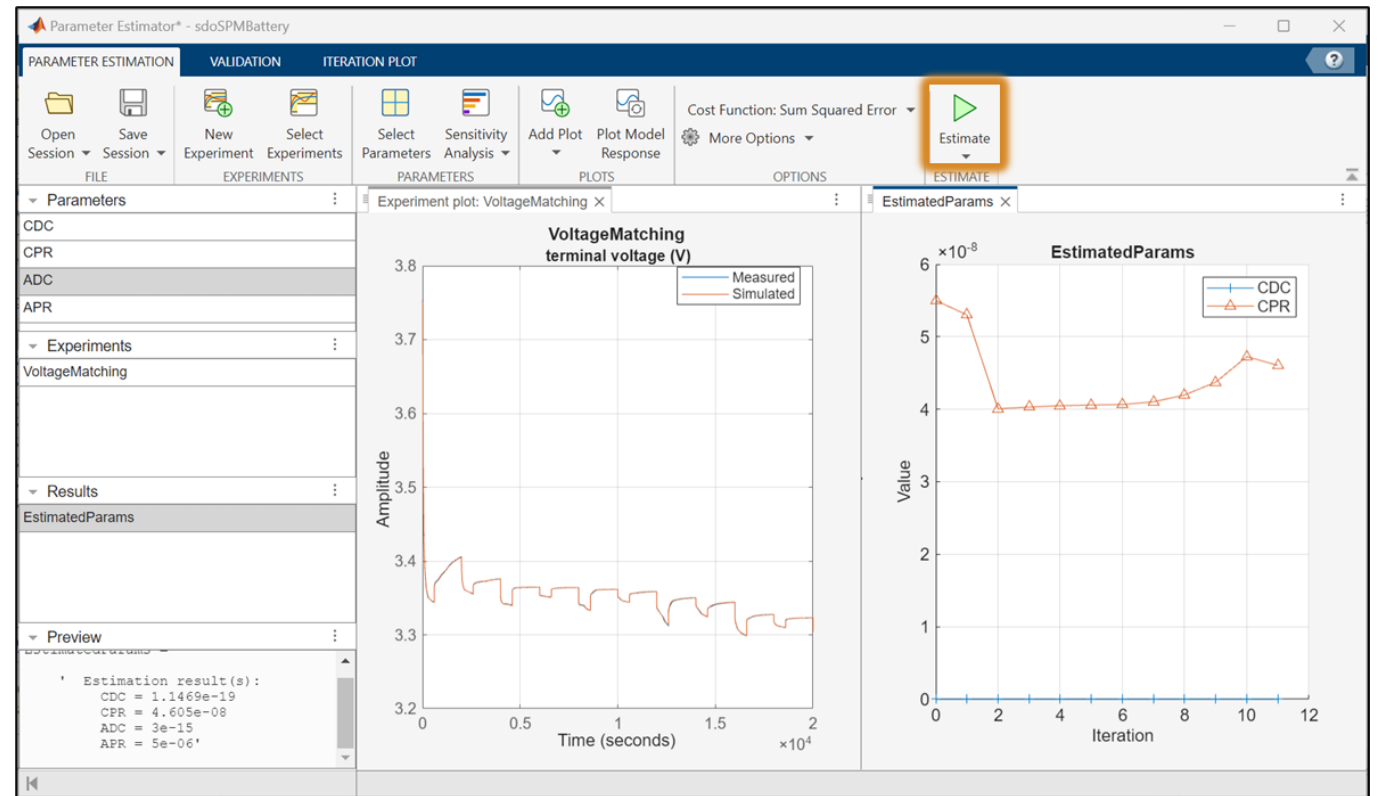
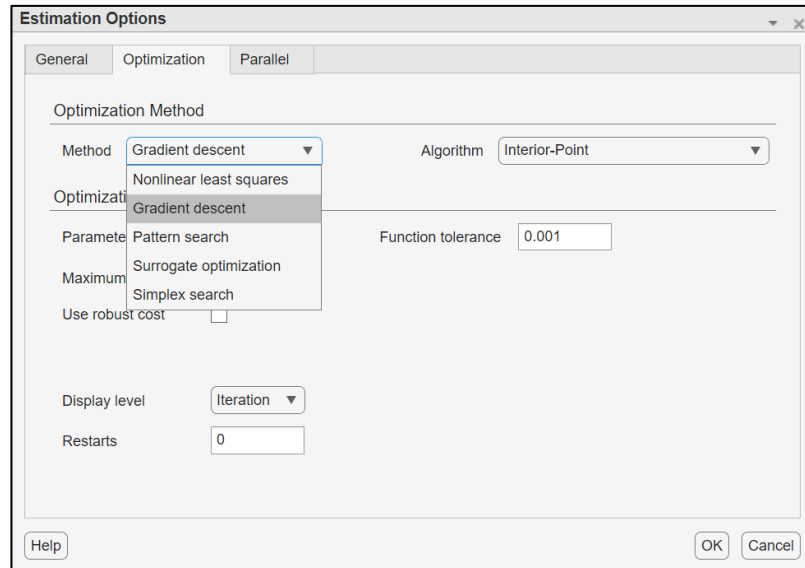
Measured Simulated

Parameter estimation with Simulink Design Optimization

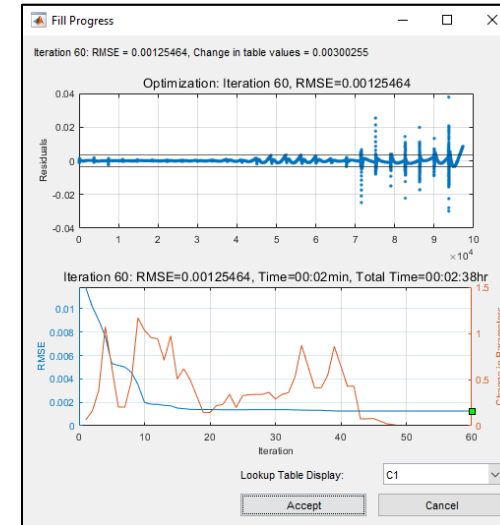
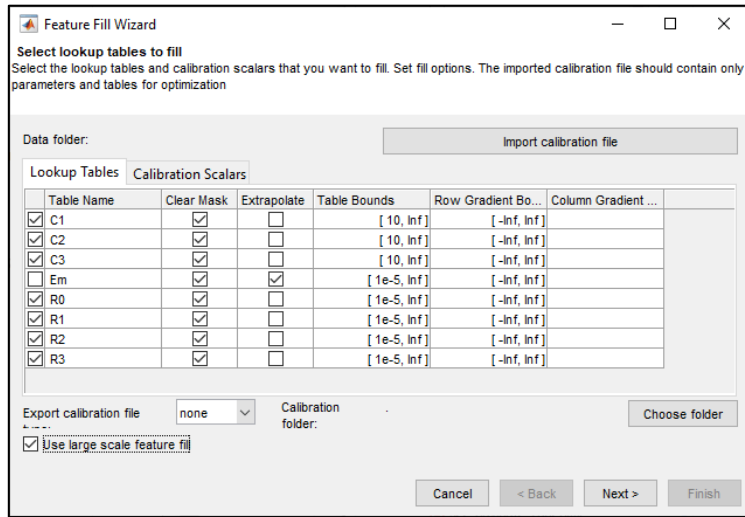
Specify Experiment

Select Parameters

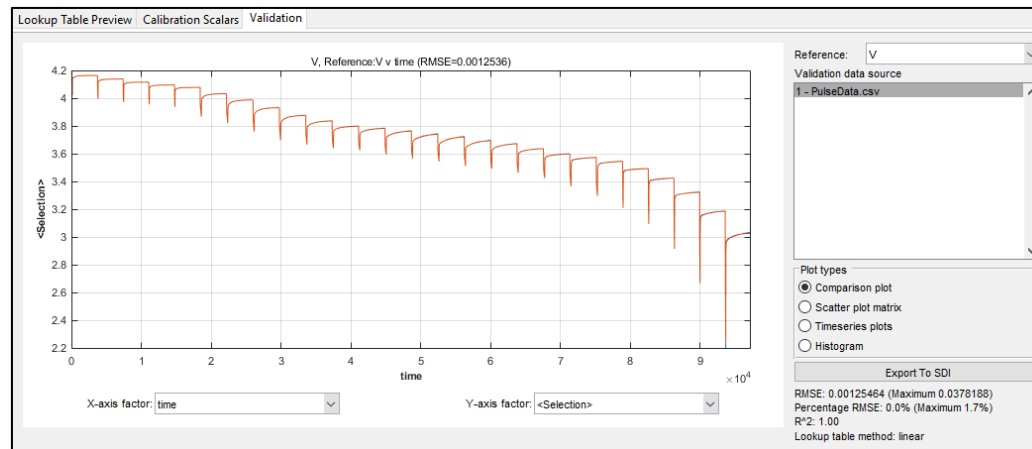
Estimate Parameters



Parameter estimation with Model-Based Calibration Toolbox



Estimate battery model lookup tables from experimental data



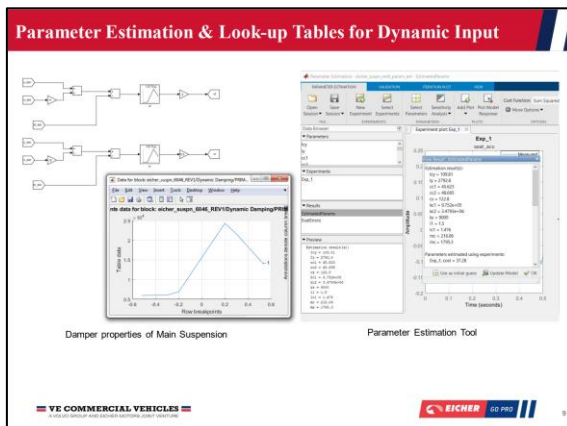
Parameter estimation is used by our customers for a variety of applications



[Link to User Story](#)

Krones Develops Package-Handling Robot Digital Twin

Simulink Design Optimization™ was used to enhance model accuracy by fitting parameters to experimental data from the tripod robot tests.



[Link to User Story](#)

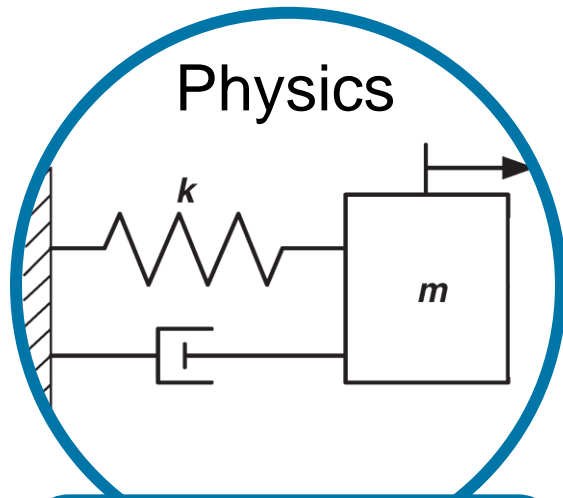
VE Commercial Vehicles Establishes Ride Comfort Characteristics of Tractor-Semitrailers

“ In a tractor-trailer, we get many components from suppliers and sometimes the parameter what we need may or may not occur during our testing. Simulink Design Optimization™ is a very handy tool when it comes to identify some of these unknown parameters.

-Sarnab Debnath
VE Commercial vehicles Ltd



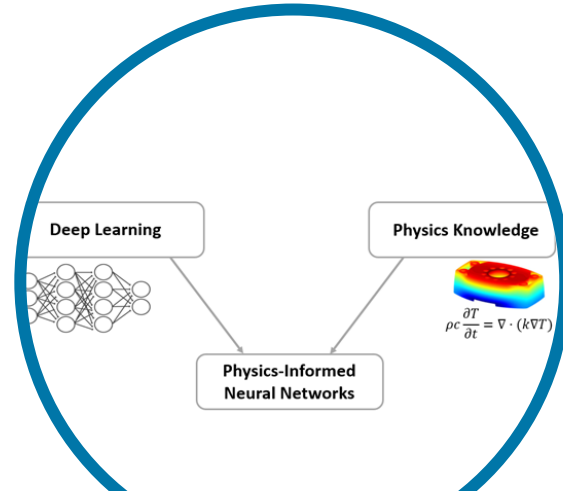
Where do you get the information to create a model?



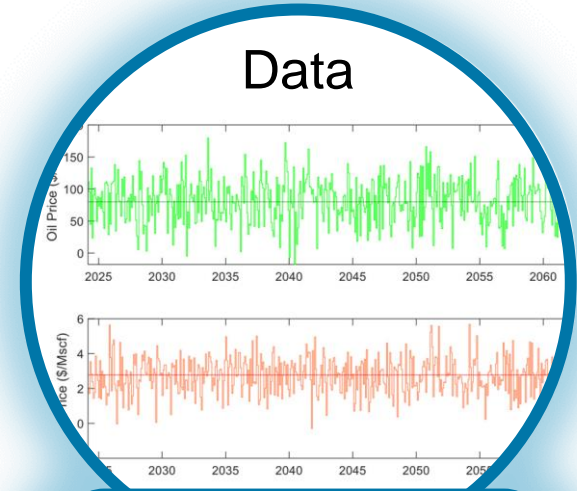
First principles modeling

$$m\ddot{x} + b\dot{x} + kx = 0$$

Parameter estimation



Physics-Informed ML



Data-driven modeling



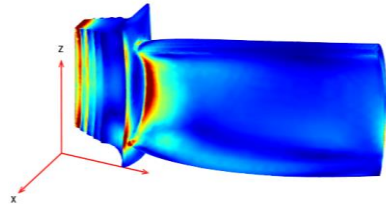
WHITE BOX

BLACK BOX

With data-driven modeling you build models that are based on measured or generated data

Gather Experimental Data

Input



*data from hardware
(measured data)*

*data from full-order models
(generated data)*

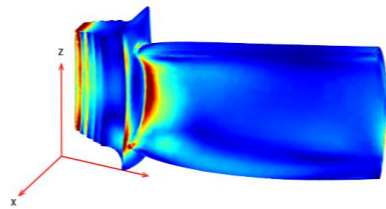
Output

With data-driven modeling you build models that are based on measured or generated data

Gather Experimental Data

Estimate Model from Data

Input



data from hardware
(measured data)

data from full-order models
(generated data)

Output

Linear Models

TRANSFER FUNCTION

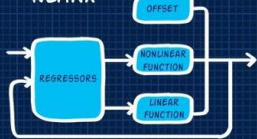
$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

STATE SPACE

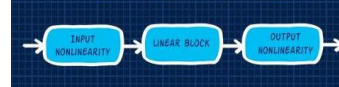
$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

Nonlinear Models

NLARX



HAMMERSTEIN-WIENER



NEURAL STATE SPACE



LSTM



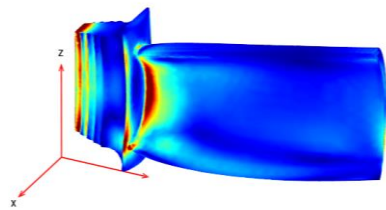
With data-driven modeling you build models that are based on measured or generated data

Gather Experimental Data

Estimate Model from Data

Validate Model with Independent Data

Input



data from hardware
(measured data)

data from full-order models
(generated data)

Output

Linear Models

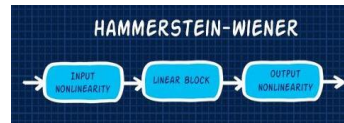
TRANSFER FUNCTION

$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

STATE SPACE

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

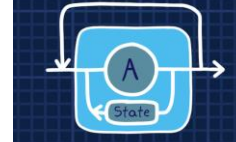
Nonlinear Models



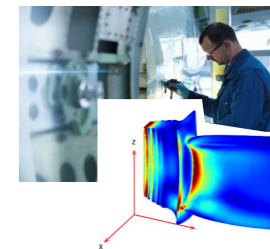
NEURAL STATE SPACE



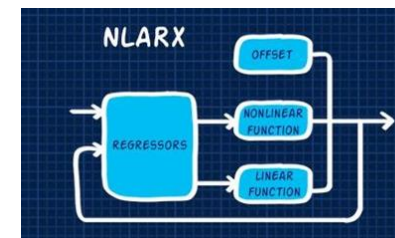
LSTM



Input



original model



estimated model

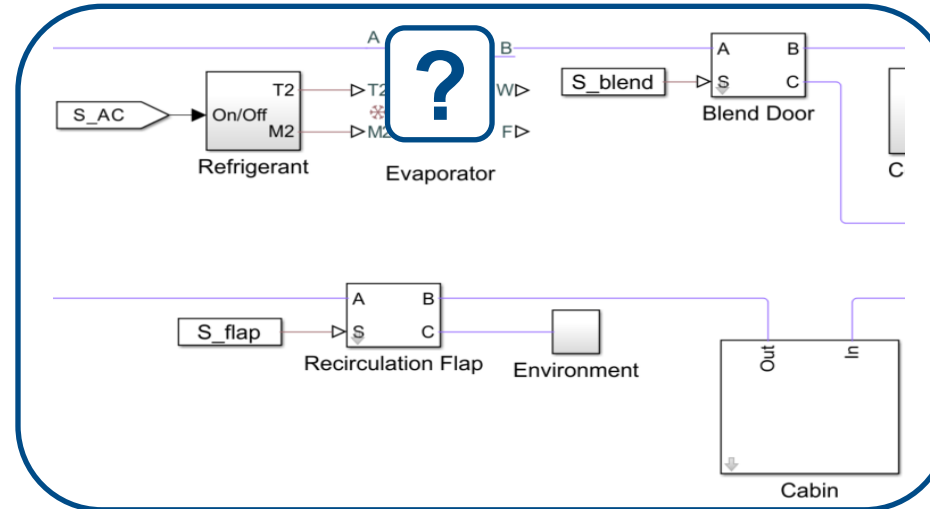
Real Output

compare

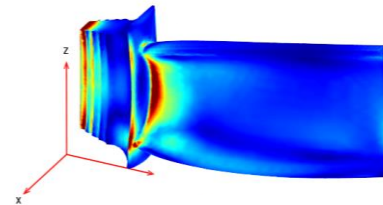
Estimated Output

There are benefits to data-driven modeling

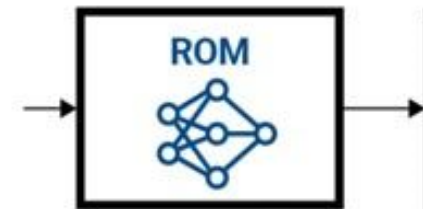
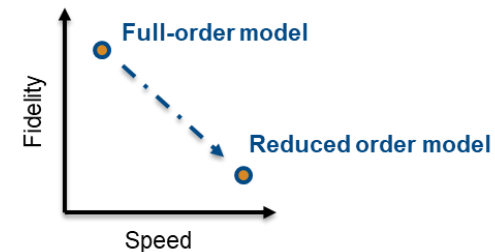
Identify models when first-principles models are difficult to derive



Reduce complexity by capturing only the dynamics of interest

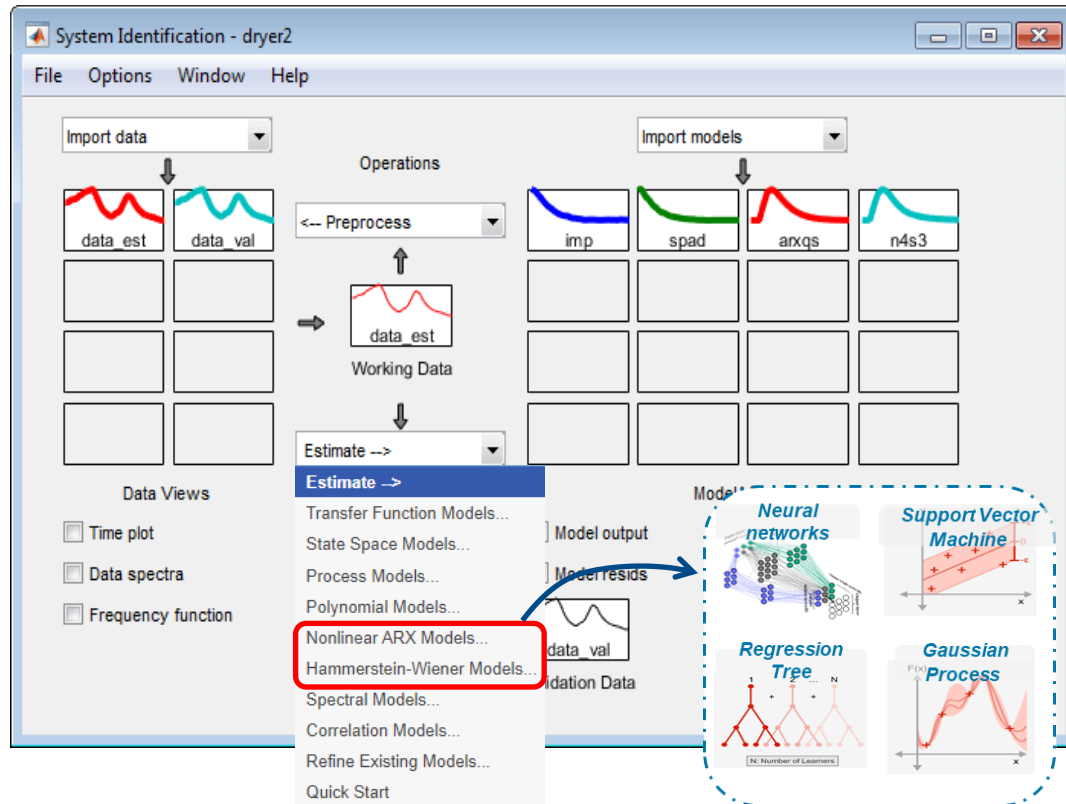


Reduced Order Modeling (ROM)

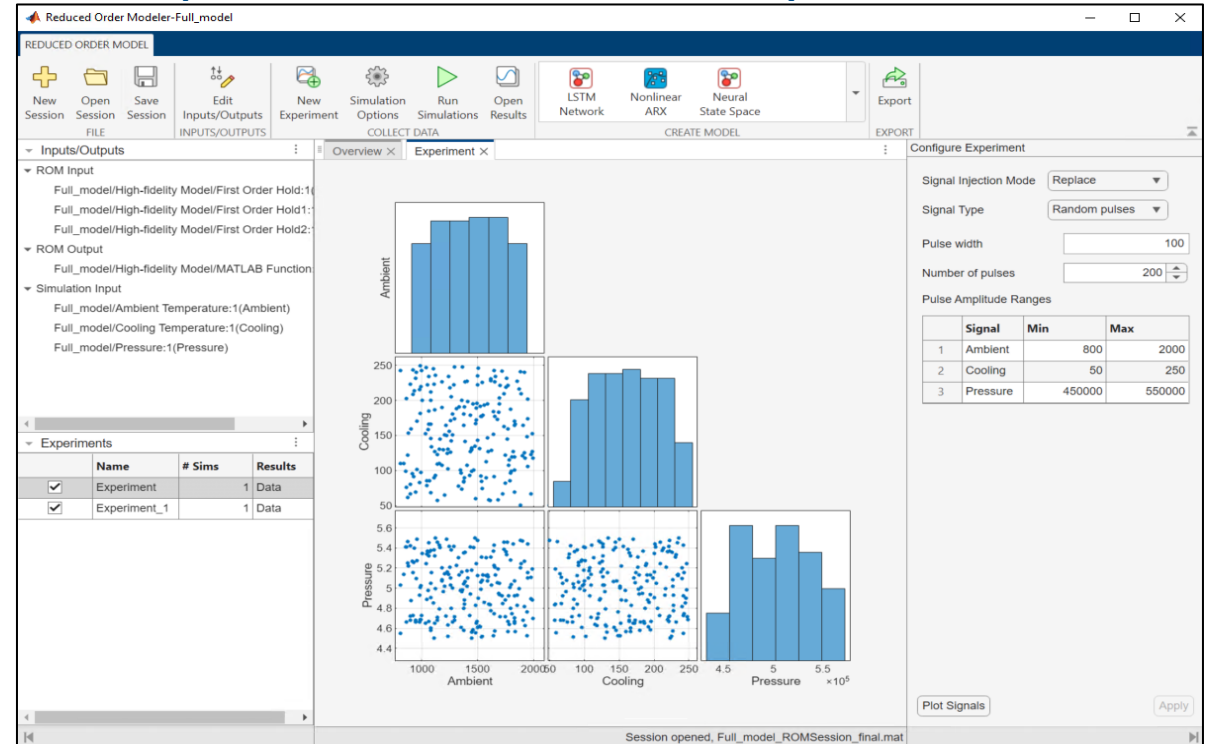


Data-Driven Modeling in MATLAB and Simulink

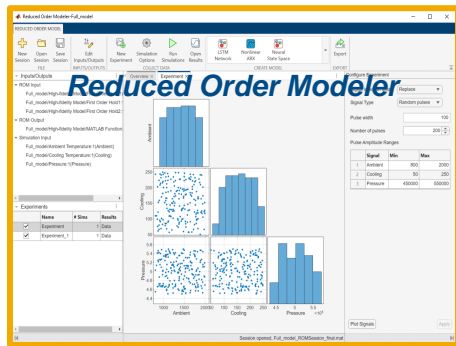
System Identification App (data from hardware)



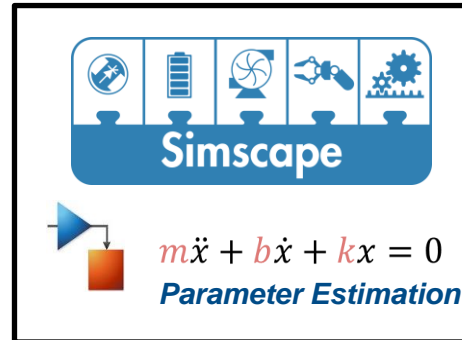
Reduced Order Modeler App (data from full-order model)



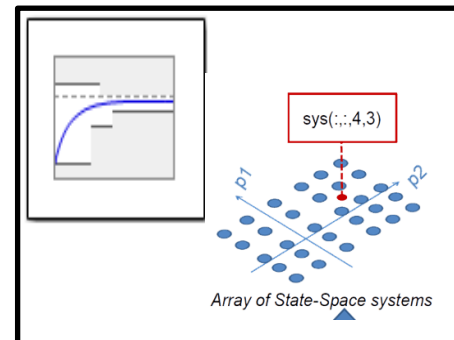
AI-based Reduced Order Modeling is one of the many ROM Techniques that MathWorks offers



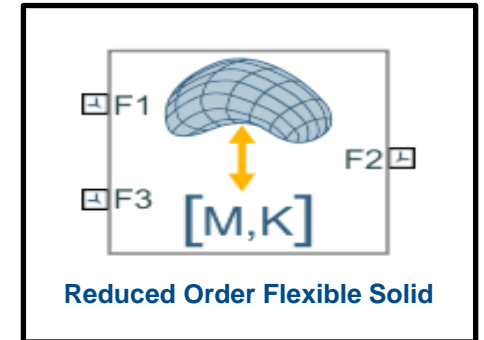
AI-Based Data-Driven



Physics-Based



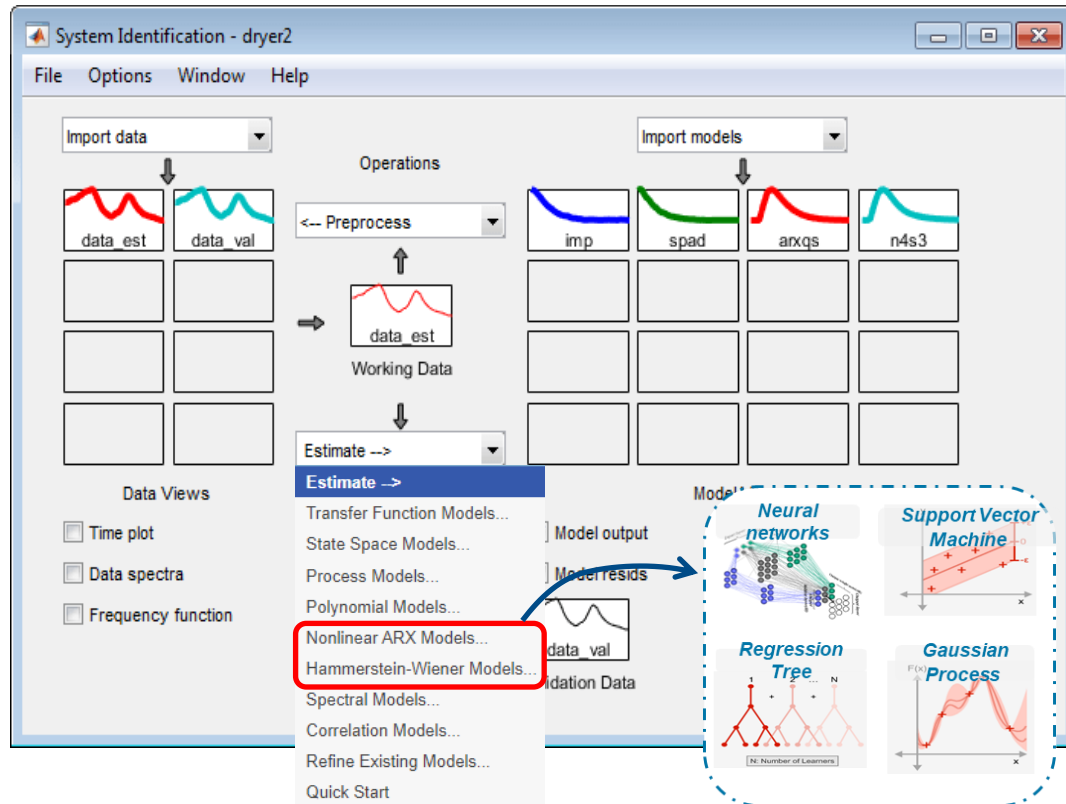
Linearization



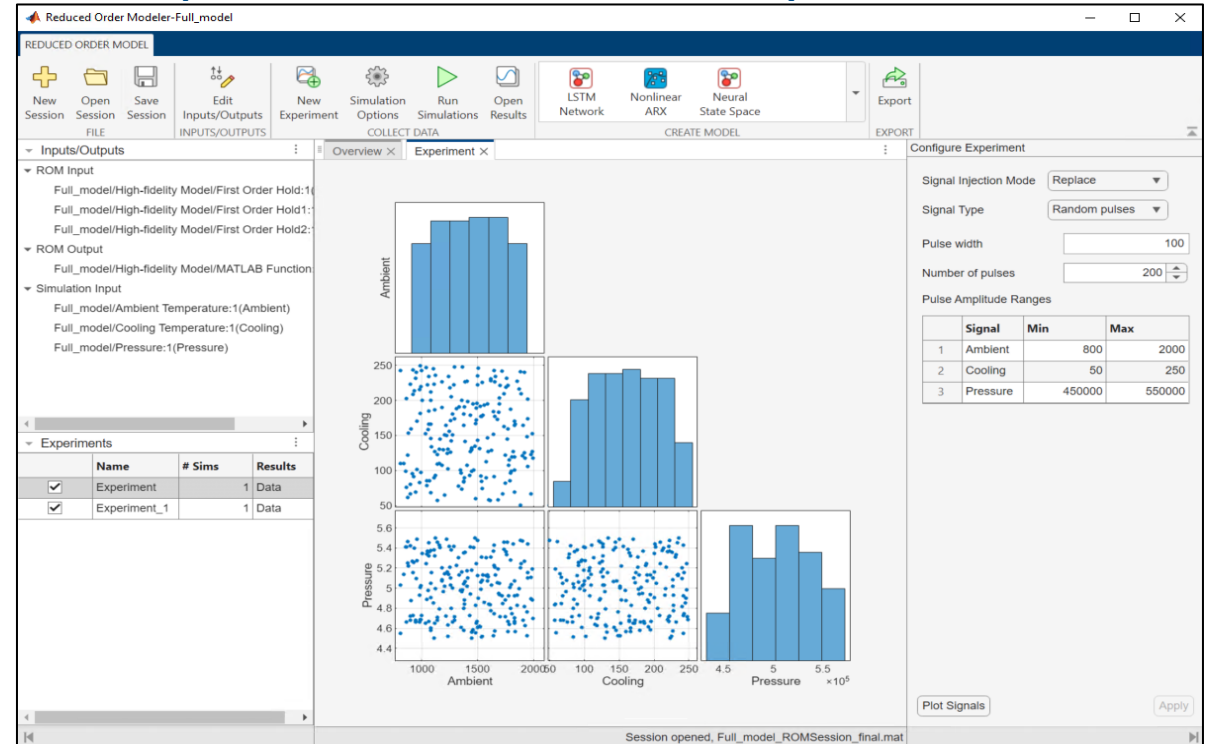
Model-Based

Data-Driven Modeling in MATLAB and Simulink

System Identification App (data from hardware)



Reduced Order Modeler App (data from full-order model)



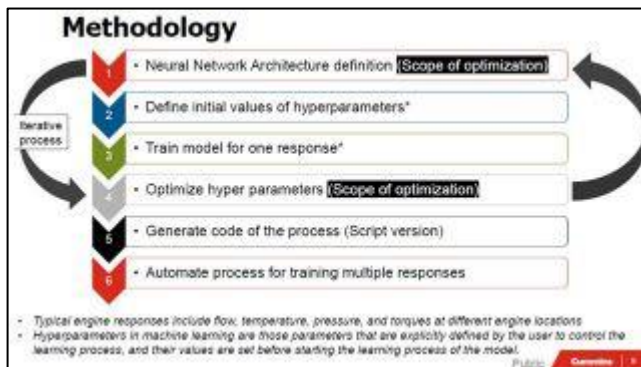
Data-Driven modeling is used by our customers for a variety of applications



[Link to User Story](#)

Ather Energy Develops Electric Two-Wheeled Scooter and Charging Stations Using Model-Based Design

System Identification Toolbox™ was used to create a model of battery cells, capturing their electrical and thermal characteristics using input-output data.

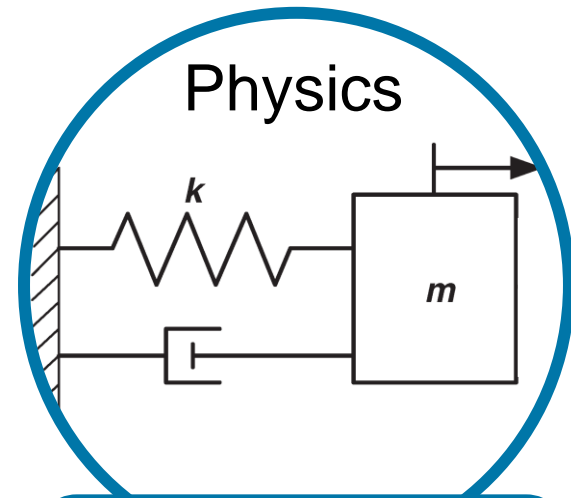


[Link to User Story](#)

Cummins Uses AI-Based Reduced Order Modeling to Predict Engine Performance and Emissions

Cummins used LSTM-based neural networks to reduce engine cycle simulation run times to one-eighth of real time.

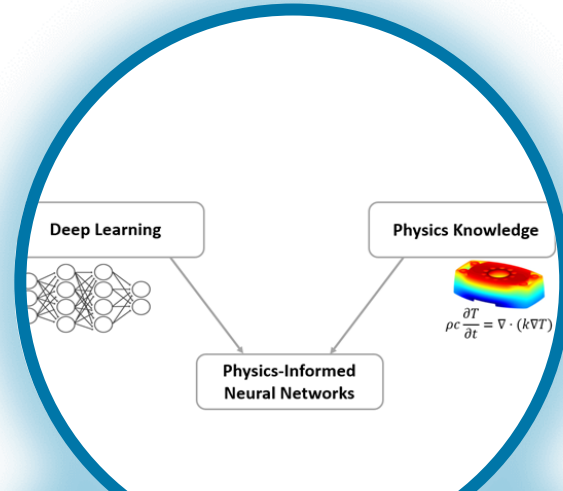
Where do you get the information to create a model?



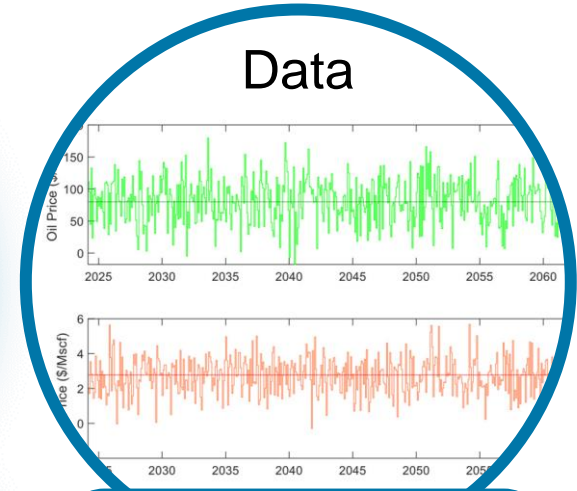
First principles modeling

$$m\ddot{x} + b\dot{x} + kx = 0$$

Parameter estimation



Physics-Informed ML



Data-driven modeling

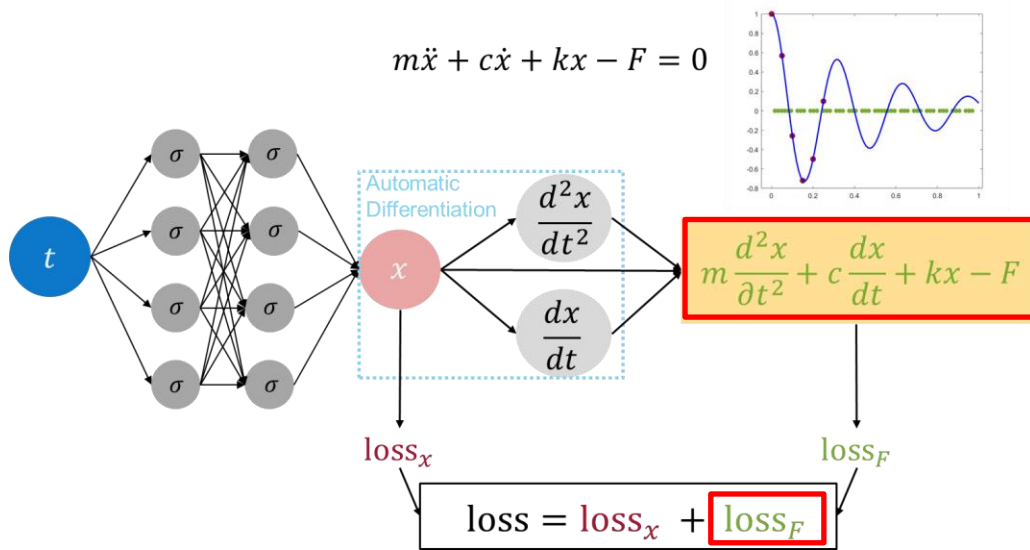


WHITE BOX

BLACK BOX

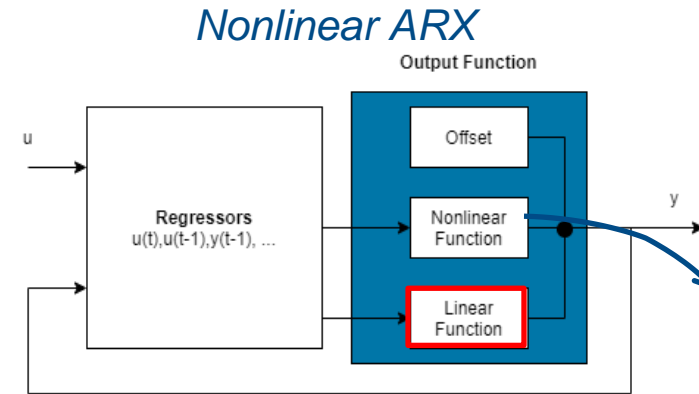
Physics-Informed Machine Learning techniques let you incorporate your understanding of physics to enhance machine learning methods

Loss Functions



Physics-Informed Neural Networks (PINNs)

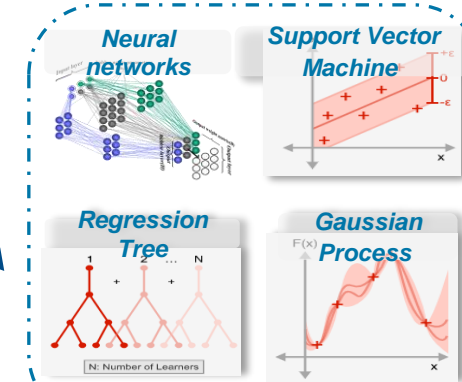
Model Architectures



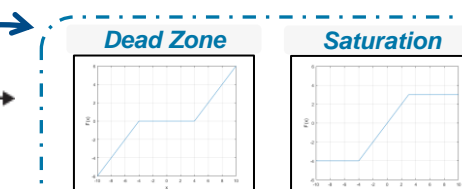
Hammerstein-Wiener



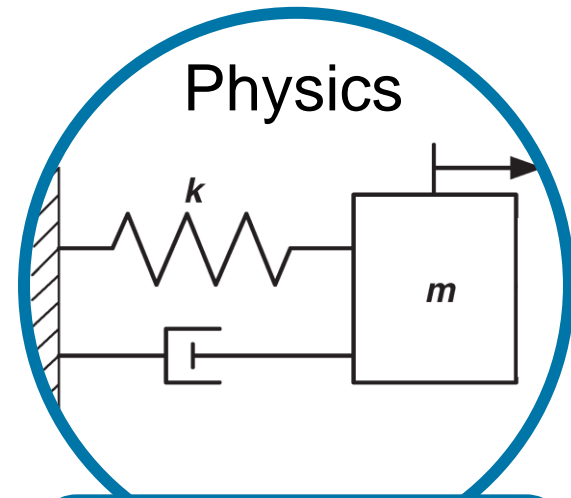
AI-based nonlinear functions



Physics inspired nonlinear functions



Where do you get the information to create a model?

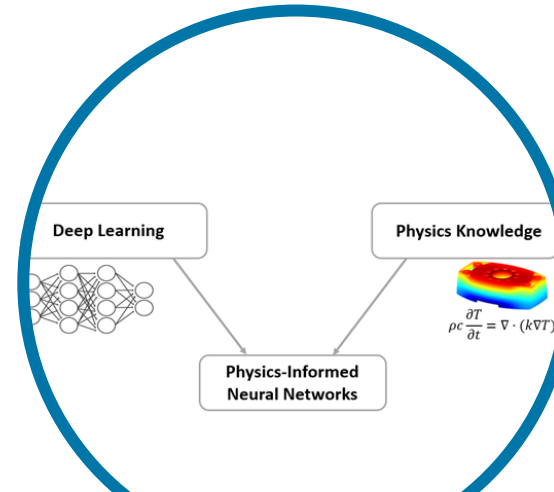


Physics

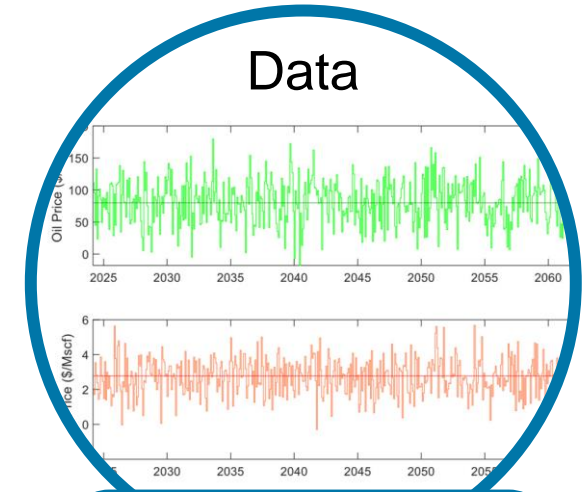
First principles modeling

$$m\ddot{x} + b\dot{x} + kx = 0$$

Parameter estimation



Physics-Informed ML



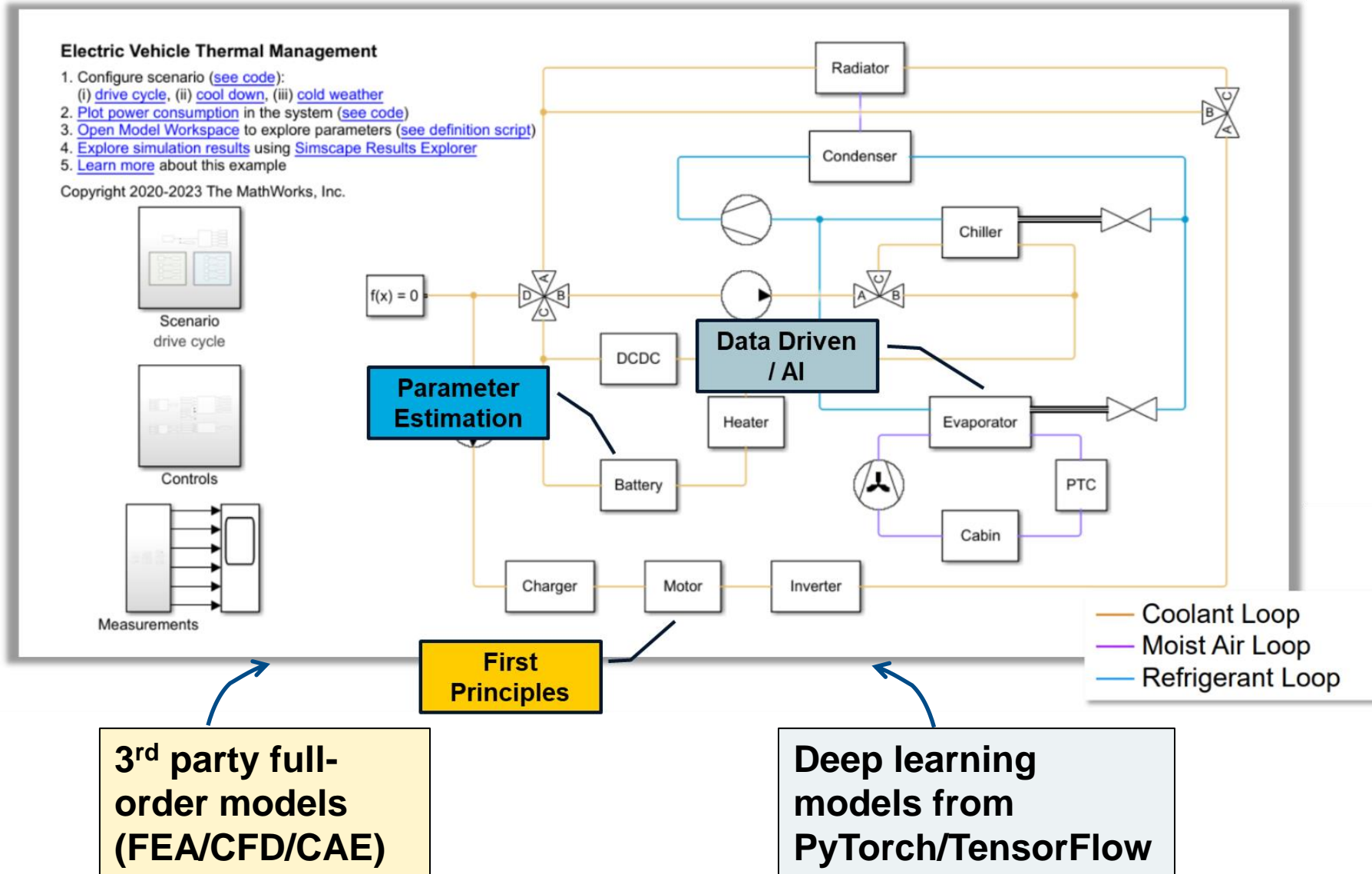
Data

Data-driven modeling

WHITE BOX

BLACK BOX

Models come together for a purpose



Desktop Simulation

Hardware-in-the-Loop Testing

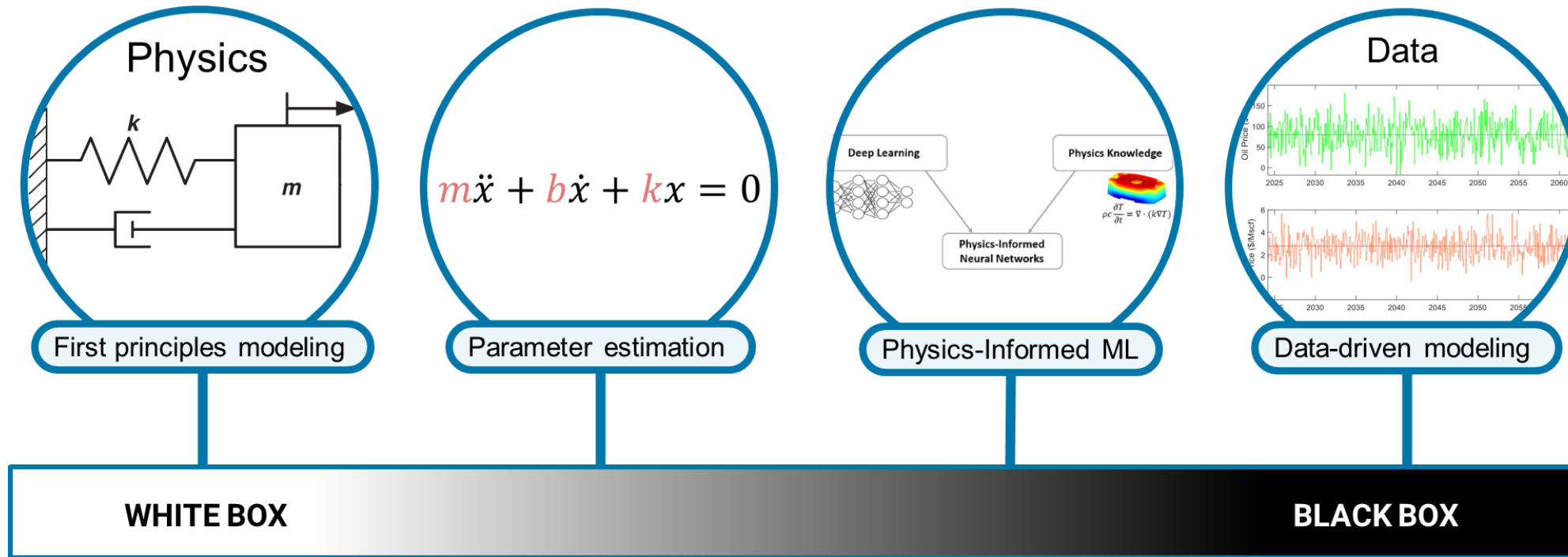
System-Level Analysis

Control Design

Design Optimization

and more

There are advantages across the modeling spectrum



Advantages

- Have clear (explainable) physical meaning
- Do not require data engineering

Advantages

- Identify models when first-principles models are difficult to derive
 - Reduce complexity

Check out this poster for a larger list of modeling capabilities

Modeling Dynamic Systems with MATLAB and Simulink

Use MATLAB® and Simulink® to support linear and nonlinear model structures, including integration of third-party models.

Explore more capabilities for modeling dynamic systems

Model Structures

Use MATLAB® and Simulink® to support linear and nonlinear model structures, including integration of third-party models.

Linear Models

TRANSFER FUNCTION: $G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$

STATE SPACE: $\dot{x} = Ax + Bu$, $y = Cx + Du$

LINEAR PARAMETER VARYING

TIME SERIES (ARX, ARMA)

FREQUENCY RESPONSE DATA

ZPK

Nonlinear Models

ODEs: $\frac{dy}{dt} = Q(t)y + R(t)y^2$

NEURAL STATE SPACE

Nonlinear portion could be represented with AI

AI-BASED MODELS

HAMMERSTEIN-WIENER

GAUSSIAN PROCESS

SUPPORT VECTOR MACHINE

REGRESSION TREE

NEURAL NETWORK

Integrate Third-Party Models

FUNCTIONAL MOCKUP UNIT

PYTHON IMPORTERS

S-FUNCTIONS

Model Parameters

Determine model parameters through first principles, grey box, and data-driven methods.

MODELING BASED ON PHYSICAL LAWS

WHITE BOX

FIRST PRINCIPLES: $f(x, t) = m \frac{d^2x}{dt^2}$

PHYSICAL MODELING WITH SIMSCAPE

GREY BOX

GREY BOX ODEs: $\frac{dx(t)}{dt} = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ c \end{bmatrix} u(t)$

PARAMETER ESTIMATION IN SIMULINK MODELS

PARAMETER ESTIMATOR

MODELING BASED ON SYSTEM DATA

BLACK BOX

SYSTEM IDENTIFICATION (TRADITIONAL AND AI-BASED)

SYSTEM IDENTIFICATION

DATA PREPARATION

ONLINE ESTIMATION

OFFLINE ESTIMATION

MODEL ANALYSIS

Model Manipulation

Modify models through transformation, linearization, and order reduction methods.

Model Transformation

MODEL TYPE: $\omega^2 \leftrightarrow \dot{x} = Ax + Bu$, $y = Cx + Du$

CONTINUOUS-DISCRETE: $f(t) \leftrightarrow f[k]$

STATE-COORDINATE: $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \leftrightarrow \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

MODAL DECOMPOSITION: $G(z) = H_0 + H_1(z) + H_2(z)$

Linearization

NUMERICAL PERTURBATION

FREQUENCY RESPONSE ESTIMATION

MODEL LINEARIZER

Block-by-Block

Reduced Order Modeling

MODEL-BASED: BALANCED TRUNCATION, POLE-ZERO SIMPLIFICATION

DATA-DRIVEN: Use model data to learn a lower order model, REDUCED ORDER MODULER

© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders. 2/2024

Explore capabilities:

Model Structures

Model Parameters

Model Manipulation

Model Structures

Linear Models

LINEAR PARAMETER VARYING

TRANSFER FUNCTION: $G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$

STATE SPACE: $\dot{x} = Ax + Bu$, $y = Cx + Du$

TIME SERIES (ARX, ARMA)

FREQUENCY RESPONSE DATA

ZPK

[Transfer Function - What Is a Transfer Function? - Discovery](#)

[State Space](#)

[What Is State Space? - Documentation](#)

[Uncertain State-Space Model - Function](#)

[Sparse First-Order State-Space Model - Function](#)

[Sparse Second-Order State-Space Model - Function](#)

[Linear Parameter Varying - LPV and LTV Models - Documentation](#)

[Time Series \(ARX, ARMA\) - Time Series Analysis - Documentation](#)

[ZPK - Zero-Pole-Gain Model - Function](#)

[Frequency Response Data - Frequency Response Data Models - Documentation](#)

Nonlinear Models

ODEs: $\frac{dy}{dt} = Q(t)y + R(t)y^2$

HAMMERSTEIN-WIENER

NEURAL STATE SPACE

NLARX

[ODEs](#)

[Getting Started with Simulink for Controls \(11:30\) - Video](#)

[Thermal Model of a House in Simulink - Example](#)

[ODEs with Symbolic Math - Documentation](#)

[Solving ODEs in MATLAB - Video Series](#)

[Neural State Space](#)