MATLAB EXPO

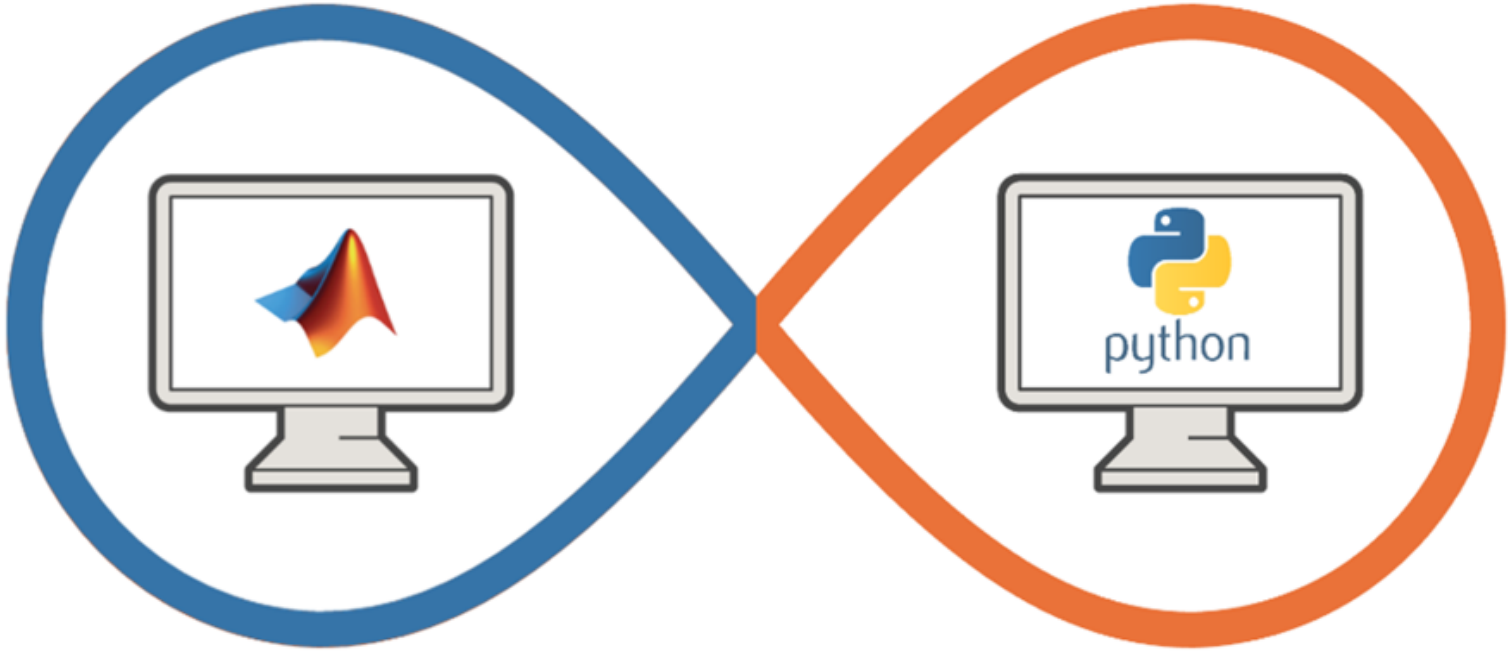
Using Simulink with Python

Weiwu Li, MathWorks

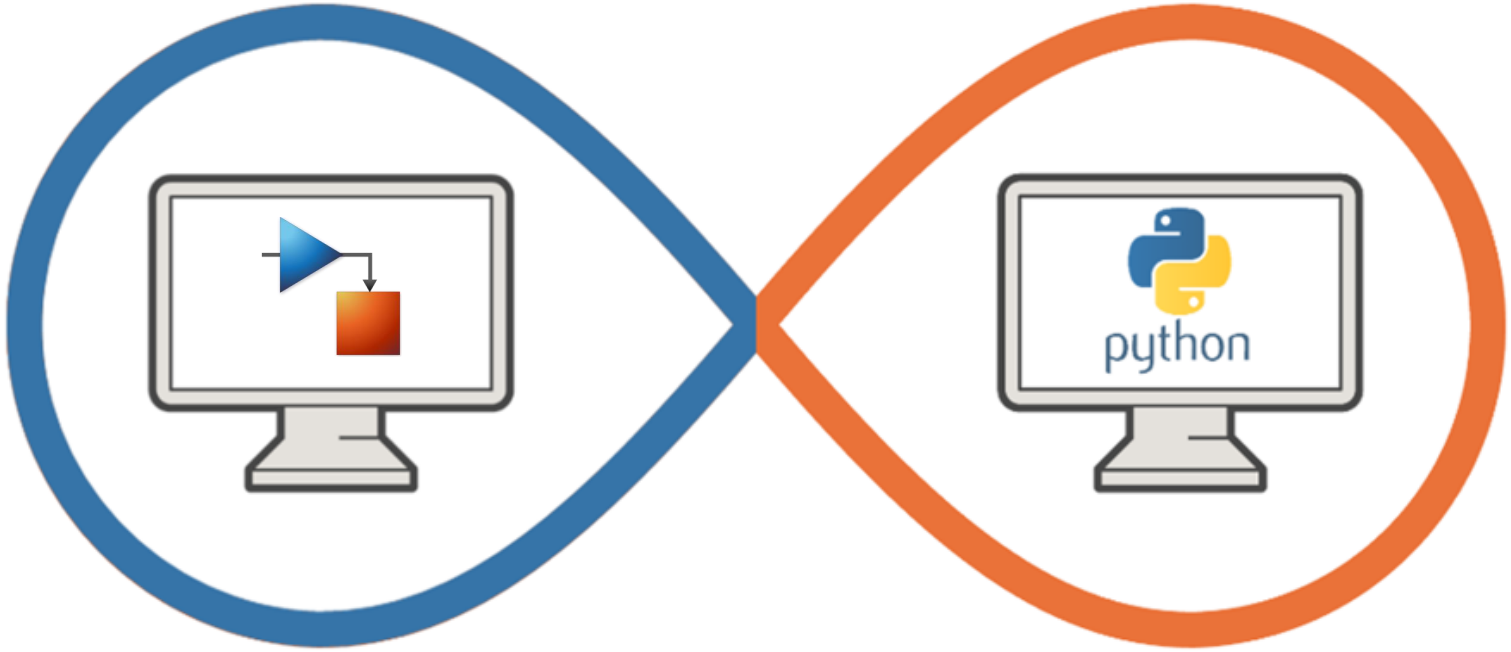
Yann Debray, MathWorks

MathWorks®

You probably have heard a lot about using MATLAB and Python



But wondered what about using Python with Simulink?



Key takeaways

- Simulink as an open simulation platform supports versatile ways to interoperate with Python:
  - Bring Python code into Simulink as a library for co-execution
  - Integrate TensorFlow and PyTorch models for both simulation and code generation
  - Simulate a Simulink model directly from Python
  - Export a Simulink model as a Python package for deployment

Why use Python and Simulink together?

- Need to **integrate** code from a colleague
- **Facilitate** development by enhancing **an AI workflow**
- Need **functionality** available in MATLAB and Simulink or in Python
- Leverage the work from the **community**

The best way to use Simulink with Python is case specific

- Let's illustrate that through 4 typical scenarios

- In a team setting:
  - Weiwu is a Simulink user and Yann is a Python user
  - Yann and Weiwu need to work with each other to deliver a project

Weiwu

Simulink user

Yann

Python user

Scenario #1

Yann

I'm an algorithm developer using Python; I develop image processing and computer vision algorithms (and many more).

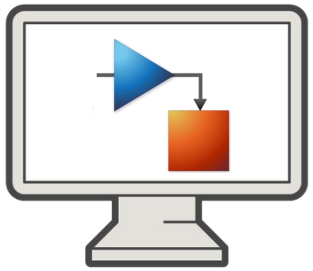
Weiwu

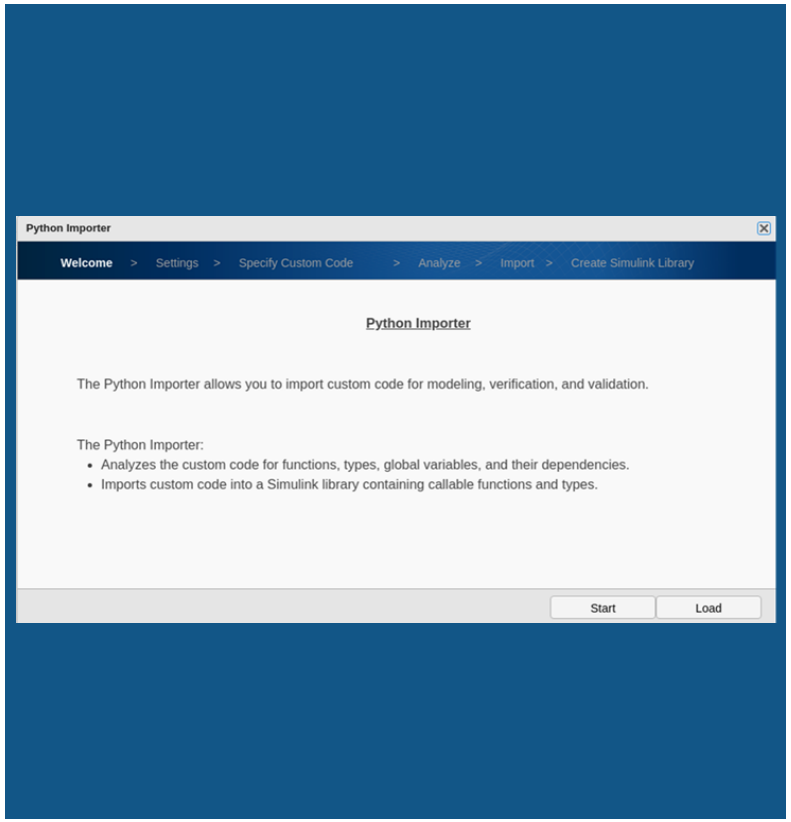
I'm a system engineer who integrates multiple components together. I want to **simulate the whole virtual system including Yann's Python algorithm in Simulink.**

Calling Python from Simulink

- Use Python Importer

Use **Python Importer** to bring Python functions into Simulink



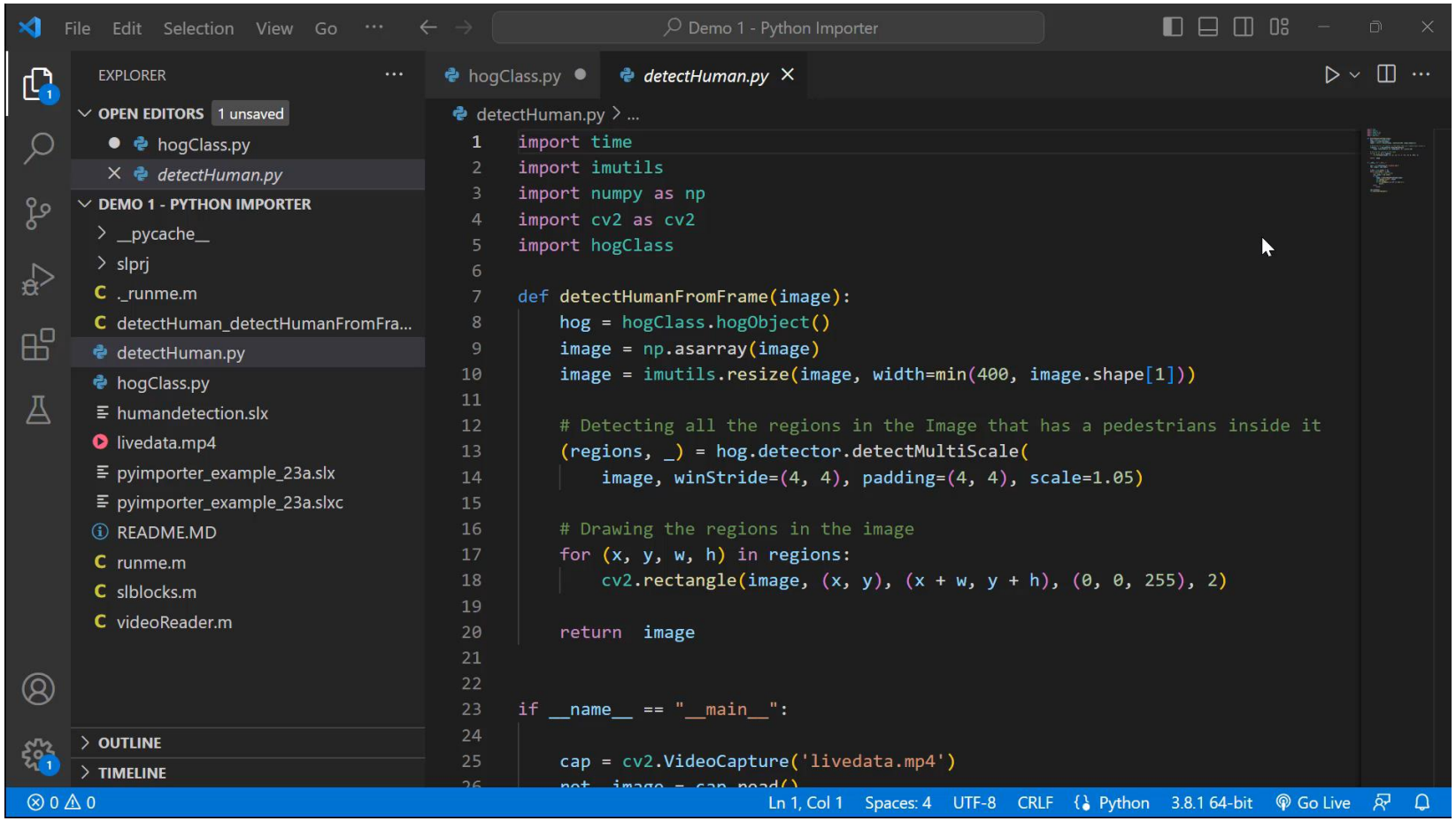


- Graphical wizard for step-by-step guidance, no/minimal manual code
- Integrating a package of Python functions with each Python function corresponding to a library block
- Convenient for re-use or building a custom blockset

R2023a

Calling Python from Simulink
- Use Python Importer

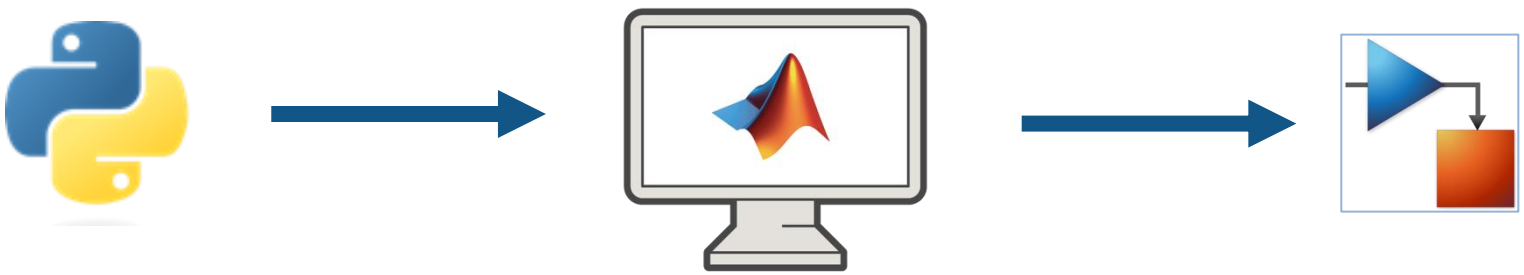
- Demo: Integrate human detection algorithm in Python

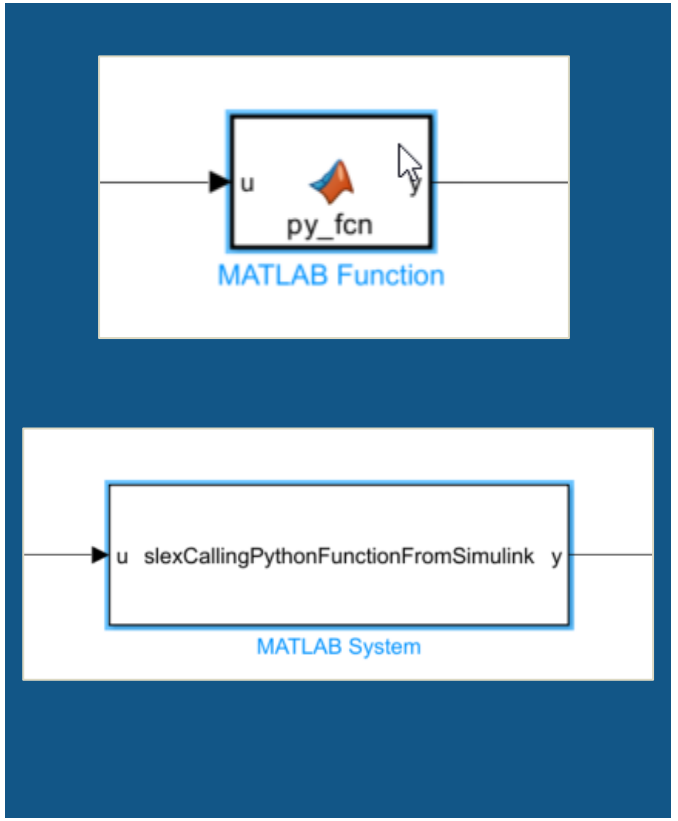


Calling Python from Simulink
- Using MATLAB Interface for Python

Use Python Interface API

Embed MATLAB code inside a MATLAB Function block or a MATLAB System block





If you are using a version earlier than R2023a, or you would like to write code manually:

- Write MATLAB functions or MATLAB System objects
- Access Python libraries directly by adding the py. prefix to the Python name

Scenario #2

Yann

I'm a data scientist using TensorFlow & PyTorch to develop deep learning models (e.g., Battery State of Charge estimation)

Weiwu

I need to bring Yann's pretrained deep learning model into Simulink for system validation. But **co-simulation is not enough, we also need code generation for hardware implementation**.

There are two options to bring your deep learning model into Simulink if code gen is a must have

- Import your deep learning model in MATLAB directly

  + Multi-platform code generation: library-free C/C++ code, optimized code for Intel and ARM processors, and CUDA code for NVIDIA® GPU

  Δ import process can be painful, need for custom code, and validation testing

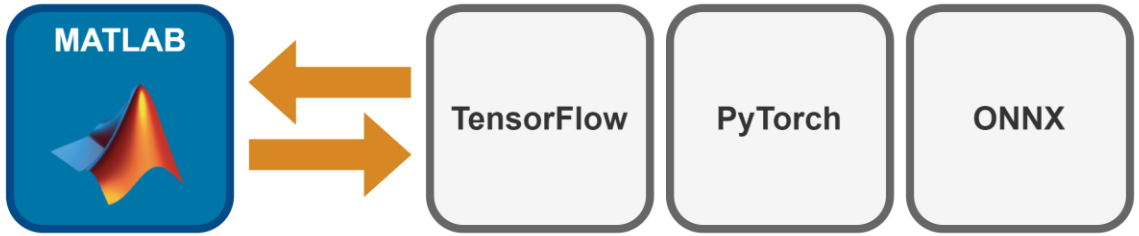
- Simulate and generate code for TensorFlow™ Lite model

  + requiring only a simple Python code to compile the model

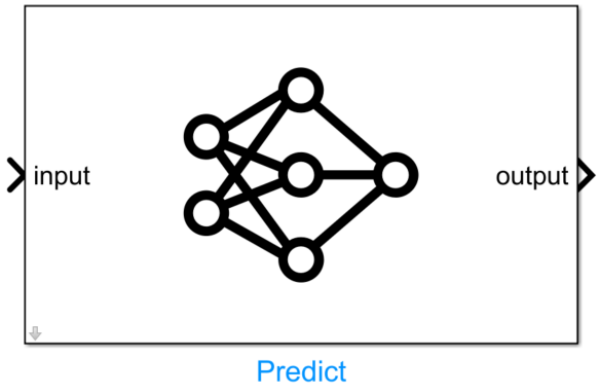
  Δ requires the TensorFlow Lite interpreter and libraries built on the target hardware, which is currently limited to Windows and Linux targets

Integrate deep learning models from Python

-using MATLAB model converters for TensorFlow, PyTorch, and ONNX



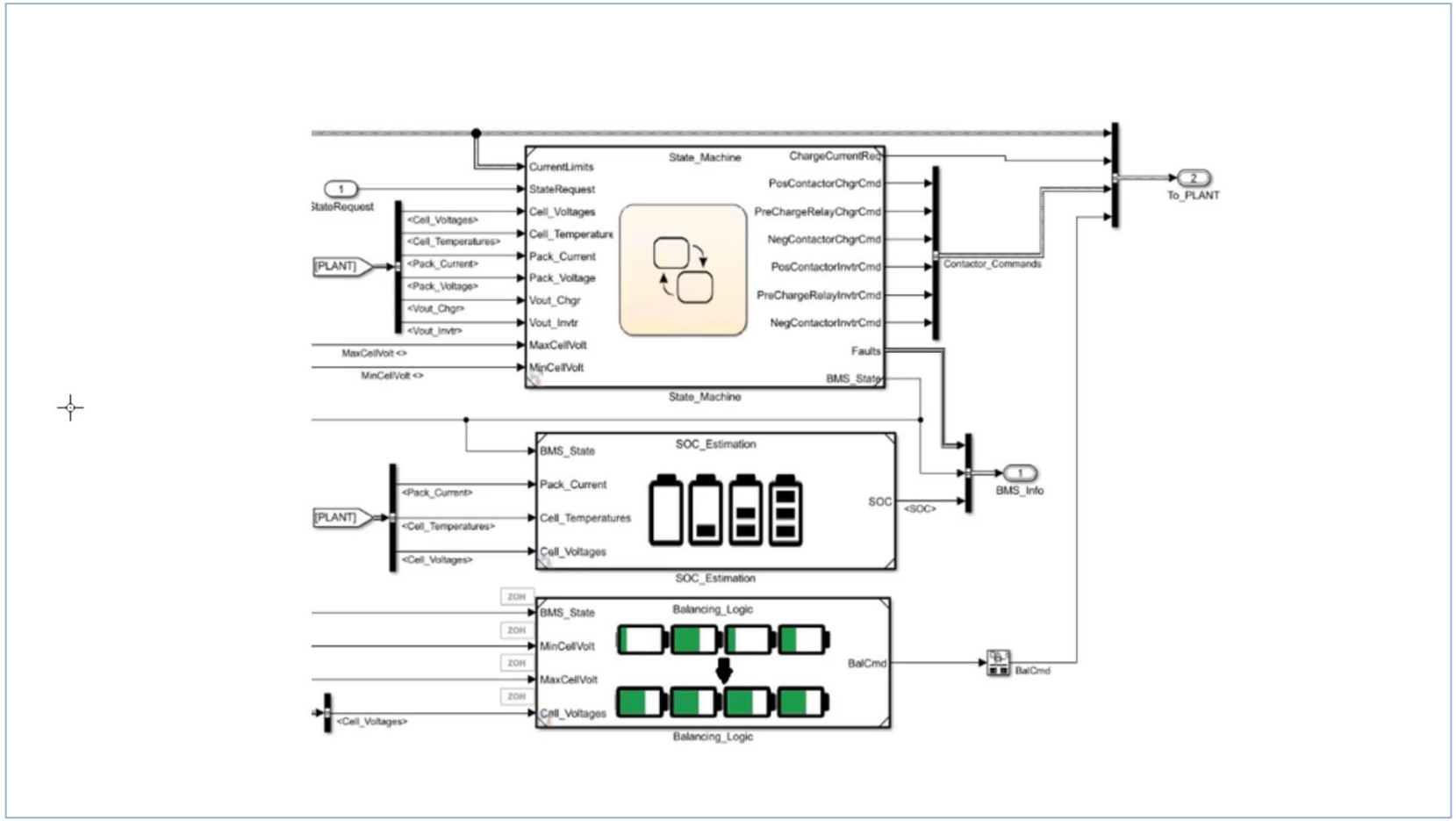
- Once the model is converted in MATLAB, use the deep neural networks blocks to bring it into Simulink, for both simulation and code generation



Integrate deep learning models from Python

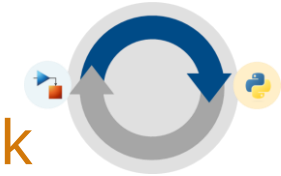
-using MATLAB model converters for TensorFlow, PyTorch, and ONNX

- Demo: Integrate a TensorFlow model for battery SoC estimation

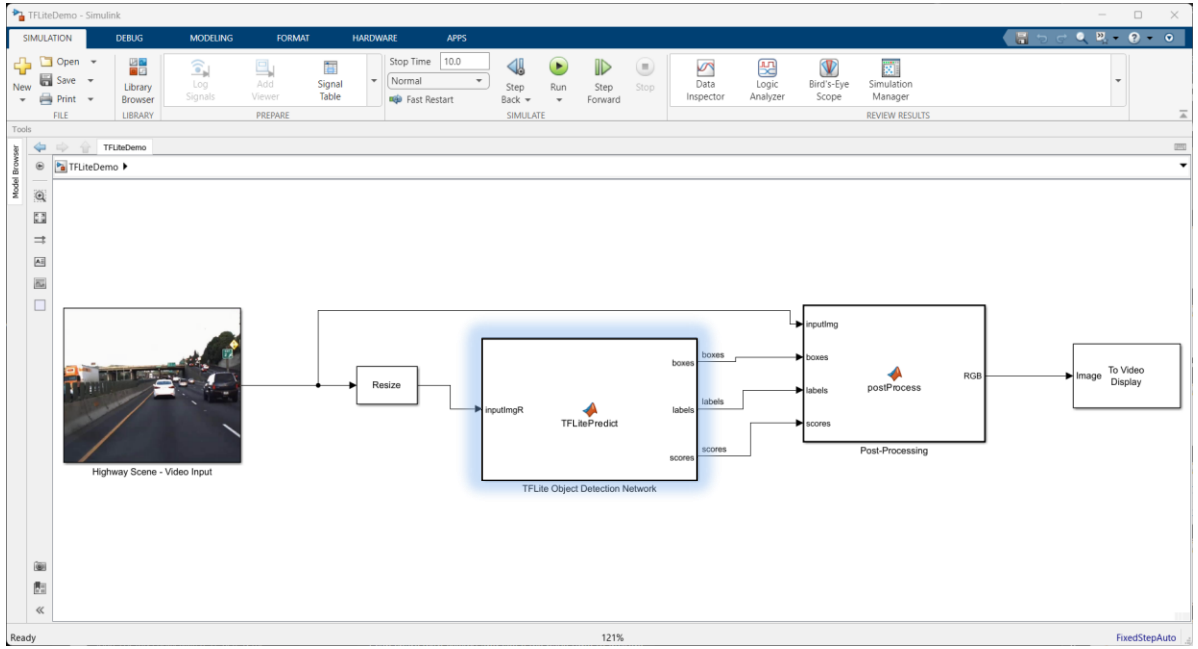


Integrate deep learning models from TensorFlow Lite (TFLite)

-using the MATLAB function to load a pre-trained TFLite model in Simulink



Example: TFLite Object Detector integrated with Simulink




```
169     memcpy(&b[0], &x[0], 100U * sizeof(real32_T));
```

```
171
172   // Function for MATLAB Function: '<Root>/TFLite Object Detection Network'
173   static void TFLiteDemo_TFLiteModel_predict(coder_TFLiteModel_TFLiteDemo_T
174     *b_this, const uint8_T varargin_1[307200], real32_T varargout_1[400], real32_T
175     varargout_2[100], real32_T varargout_3[100], real32_T *varargout_4)
176   {
177     real_T count;
178     std::mem_fn(&invokeinterpreter::setVerbose)(b_this->Network, b_this->Verbose);
179     std::mem_fn(&invokeinterpreter::setProfiling)(b_this->Network,
180       b_this->EnableProfiling);
181     std::mem_fn(&invokeinterpreter::setNumThreads)(b_this->Network,
182       b_this->NumThreads);
183     std::mem_fn(&invokeinterpreter::setInputMean)(b_this->Network, b_this->Mean);
184     std::mem_fn(&invokeinterpreter::setInputStdDeviation)(b_this->Network,
185       b_this->StandardDeviation);
```

Import TFLite models using a MATLAB Function block

Scenario #3

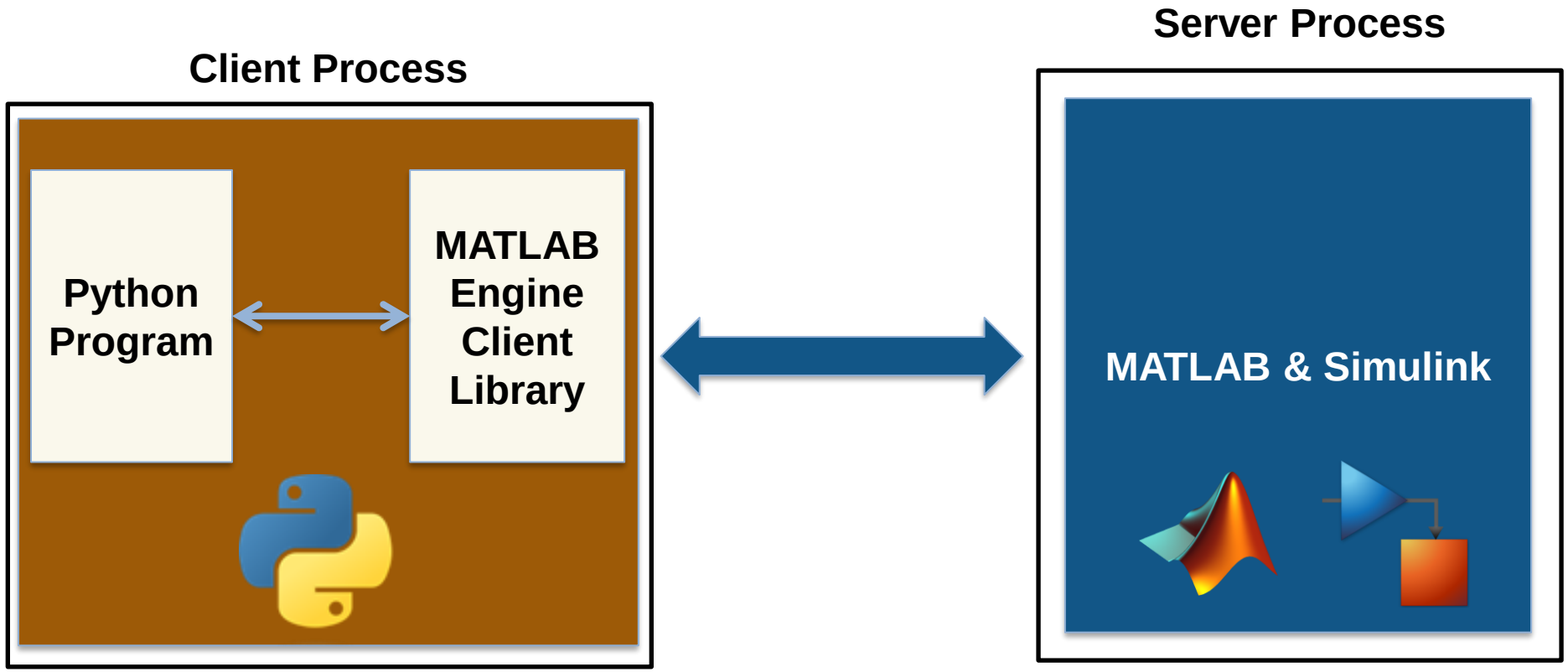
Weiwu

I use Simulink to model a dynamic system, for example, a vehicle suspension system.

Yann

I want to use a Python based automation framework to run Simulink simulations. I need to **invoke Weiwu's Simulink model from Python** for automated testing.

Simulate a Simulink Model from Python
- Using MATLAB Engine API



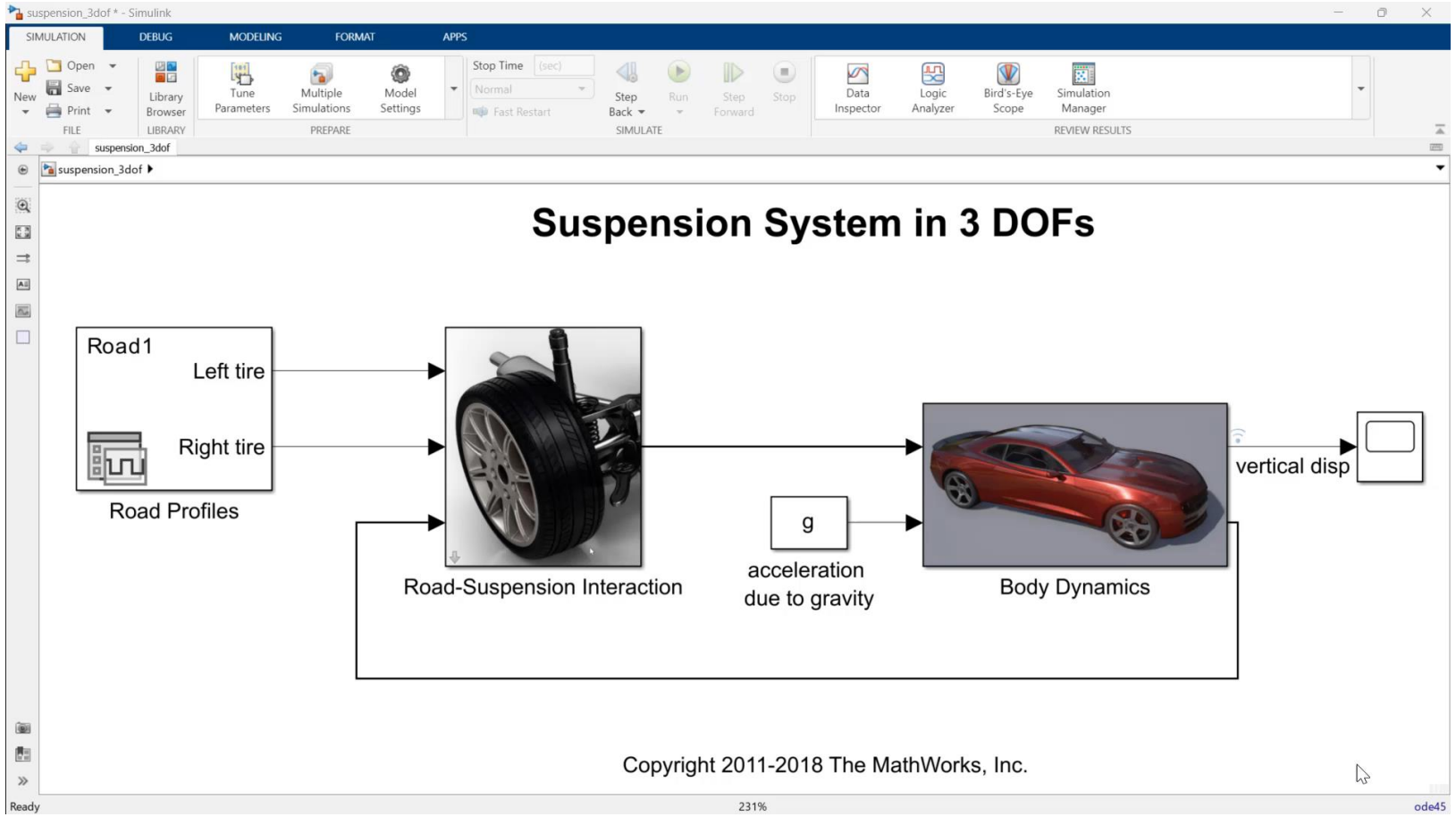

```
mle = matlab.engine.start_matlab(); # start the MATLAB engine
res[0] = mle.sim_the_model();  # # run Simulink simulation within a MATLAB function
```

Simulate a Simulink Model from Python
- Using MATLAB Engine API

- Create/terminate MATLAB
- Put variable into MATLAB workspace
- Get variable from MATLAB workspace
- Provide flexible Simulink simulation capabilities including changing non-tunable parameters and running simulations in normal mode

Simulate a Simulink Model from Python
- Using MATLAB Engine API

- Demo: Simulate a road suspension model in Python



Scenario #4

Weiwu

I use Simulink to model a dynamic system, for example, a vehicle suspension system.

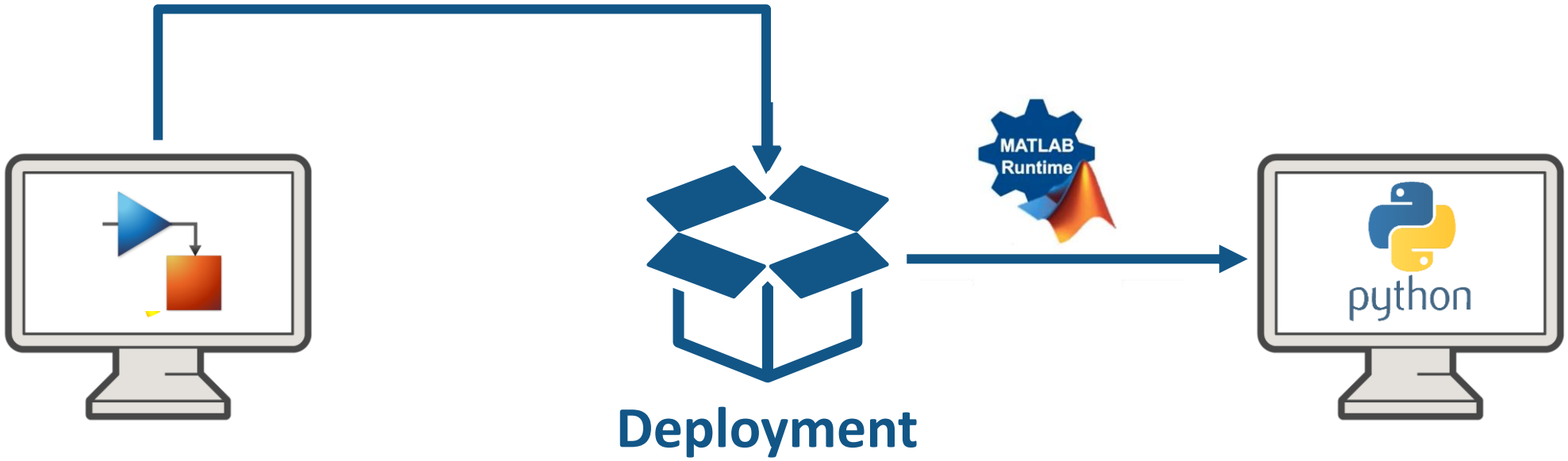
Yann

We need to deploy Weiwu's Simulink model in a Python-based production environment. I want to **get a Python package which encapsulates a Simulink simulation** which can be used for deployment.

Call a compiled Simulink model from Python

- Using MATLAB Runtime

Generate a Python package from a MATLAB function that encapsulates a Simulink simulation

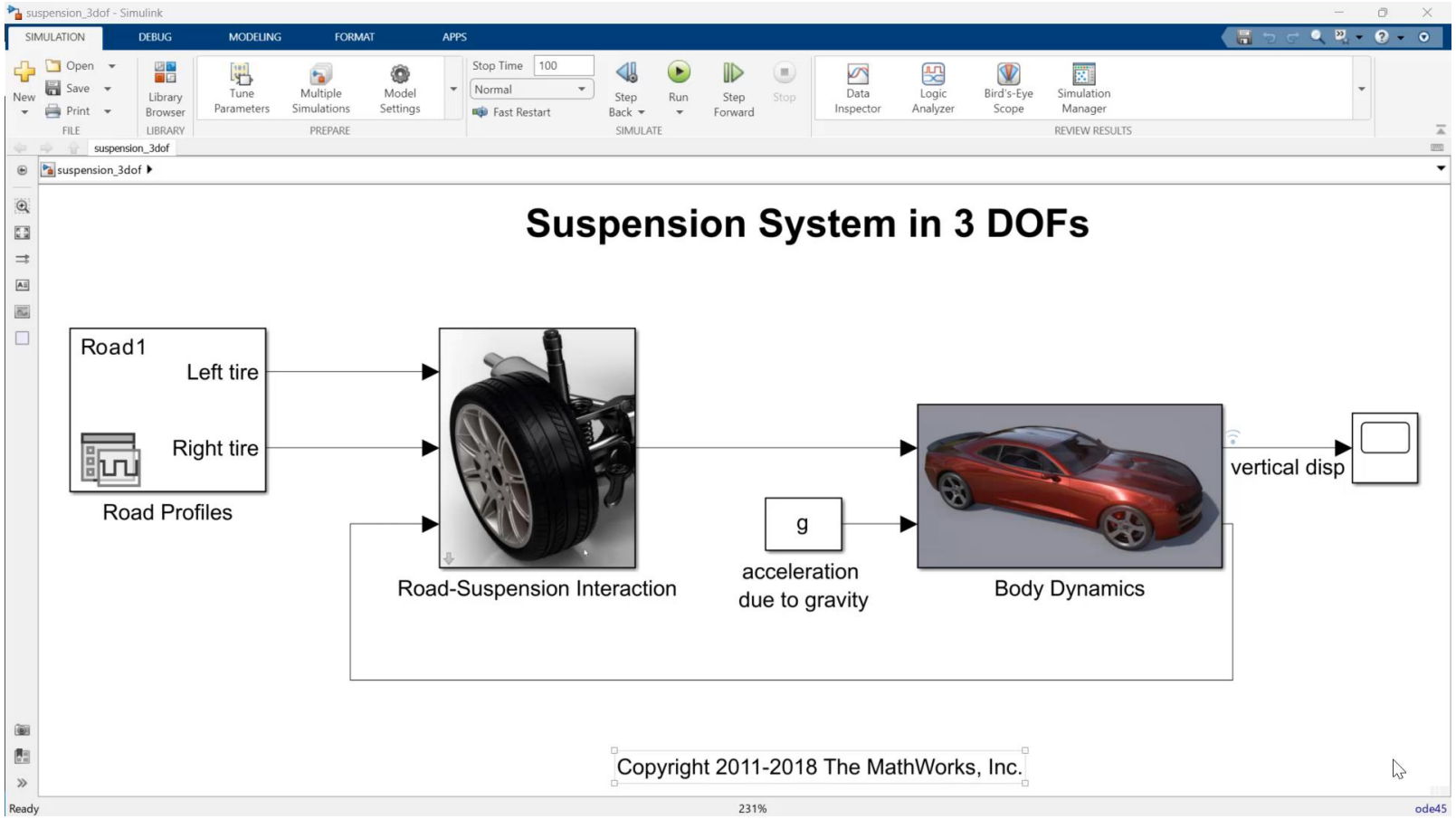



```
import sim_the_model
mlr = sim_the_model.initialize()
res[0] = mlr.sim_the_model()
```

Call a compiled Simulink model from Python

- Using MATLAB Runtime

- Demo: Simulate the compiled suspension system model as a Python package

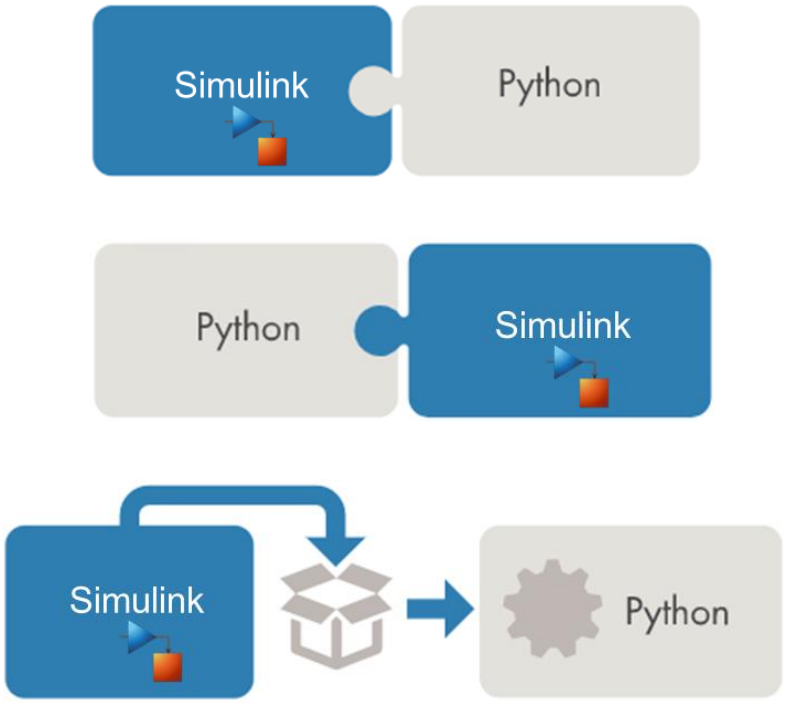


Other ways to call a compiled Simulink model in Python

- Package the Simulink model as a Functional Mockup Unit (FMU)
  - Call the FMU from Python using third-party libraries such as FMPy
- Package the Simulink model as a simulation service API (using MATLAB Production Server)
  - RESTful API for scalable applications
- Generate C/C++ code or shared library from the Simulink model
  - Call the generated code using CTYPES or related wrappers

Key takeaways

- Simulink as an open simulation platform supports versatile ways to interoperate with Python:
  - Bring Python code into Simulink as a library for co-execution
  - Integrate TensorFlow and PyTorch models for both simulation and code generation
  - Simulate a Simulink model directly from Python
  - Export a Simulink model as a Python package for deployment



Customer Reference: Mercedes-Benz Simulates Hardware Sensors with Deep Neural Networks

Challenge

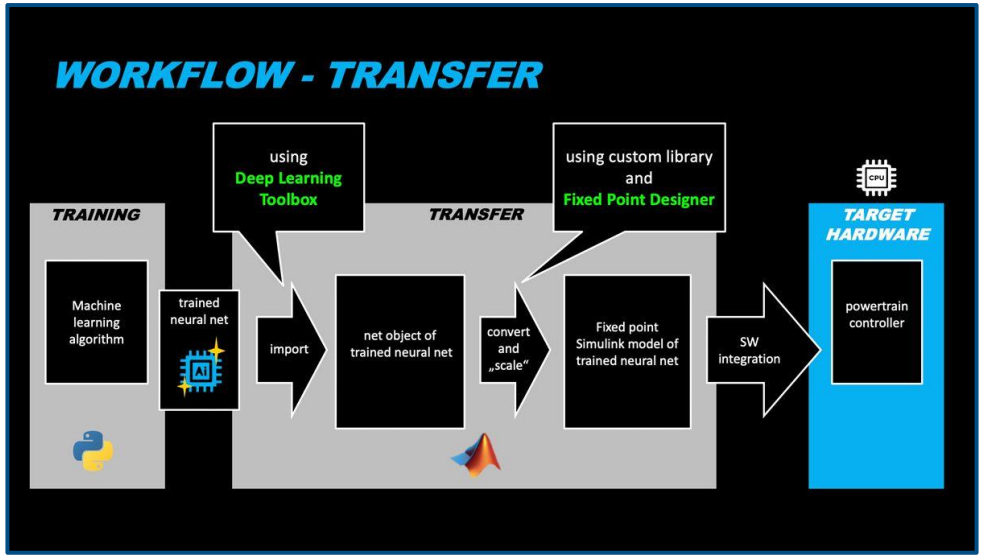
Simulate automotive hardware sensors with deep neural networks

Solution

Use MATLAB, Simulink, Deep Learning Toolbox, and Fixed-Point Designer to convert Qkeras deep learning models into code that can be deployed to an automotive ECU

Results

- CPU, memory, and performance requirements met
- Flexible process established
- Development speed increased 600%



Automated workflow for deploying virtual sensors to powertrain ECU.

"This was the first time we were simulating sensors with neural networks on one of our powertrain ECUs. Without MATLAB and Simulink, we would have to use a tedious manual coding process that was very slow and error-prone."

- Katja Deuschl, AI developer at Mercedes-Benz

Link to user story

To learn more

- Import Python Code to Simulink Using Python Importer Wizard
- Integrate Python Code with Simulink
- Deep learning with Simulink
- Importing Models from TensorFlow, PyTorch, and ONNX
- MATLAB Engine API
- Call Simulink from Python
- Python Package Integration

MATLAB EXPO

Thank you