Virtual XCU Calibration with Neural Networks
NARX/Sequential Neural Networks for Dynamical Systems
## Agenda

1. Classical ECU Functions
2. Deep Dynamical Systems
3. Deploying AI As Virtual Testbench
Classical ECU Functions

What is an ECU Function?

- Mapping Input Signals to Output signals

- Pedal value
- Engine speed
- Temperature

Fuel Mass
Classical ECU Functions
Advantages/Disadvantages

**Advantages**

- Physically motivated
- High understanding of what's going on (intermediate signals have typically physical units)
- Enabling “transfer learning” for single HW change

**Disadvantage**

- Require development (modelling + coding)
- Require methodology development for calibration = training
- Require tooling for the training (backpropagation)
- Require very special measurements from engine test bench
Deep Dynamical Systems
Network overview

› No LSTM (Long Short-Term Model)
› NARX (Nonlinear autoregressive neural network)
Deep Dynamical Systems
Temperature example 40min of driving (validation)
Deep Dynamical Systems
Training at AWS

AWS Cloud

API Gateway

Trigger
Sagemaker

Measurement
file

Hyperparameter
Training
Sagemaker

Trained Model

Codebuild

Amazon ECR
NARX Training

Engineer

MATLAB

Diagram showing the integration of various AWS services for training deep dynamical systems.
Deep Dynamical Systems
Applications

Deploy to ECU
Virtual Testbench

Calibrate Controller for Dynamical System
Reinforcement Learning
Deploying Artificial Intelligence As Virtual Testbench
Matlab/Simulink Workflow

Measurement → Train Neural Network → Gensim

Simulink Embedded Coder → Compile with ASAP Interface and external mode → Model.exe

A2L → XCP over TCP/IP → Engineer

Model.exe → XCP over TCP/IP
Deploying Artificial Intelligence As Virtual Testbench

Measure Neural Network with INCA

Test Bench

A2L

measure

Test Bench

A2L

Neuronal Network

Virtual A2L

XCP over TCP/IP