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

Engineer

MATLAB

Measurement file

API Gateway

Trigger

Sagemaker

Hyperparameter

Training

Sagemaker

Trained Model

AWS Cloud

Amazon ECR

NARX Training

Codebuild

Engineer
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 → Model.exe → XCP over TCP/IP → Engineer

Simulink Embedded Coder → Compile with ASAP Interface and external mode → A2L → XCP over TCP/IP → Model.exe → Engineer
Deploying Artificial Intelligence As Virtual Testbench
Measure Neural Network with INCA

Test Bench

measure

A2L

Neuronal Network

XCP over TCP/IP

Virtual A2L