MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

On-Device AI Implementation for Mobility Controllers with Model-Based Design

Soo-Hyeok Kang, Hyundai KEFICO

MathWorks®

Contents

- Hyundai Kefico Introduction
- On-Device AI Process using MATLAB & Simulink
- Data Preparation using MATLAB
- AI Model Design using Deep Network Designer
- AI Model Training using Deep Learning Toolbox
- MBD Design using Simulink
- Code Generation using Embedded Coder
- Deploy to Mobility Controller
- Verification of Real Time Inference Performance on Controller
- Conclusions and Future Work

What kind of company is

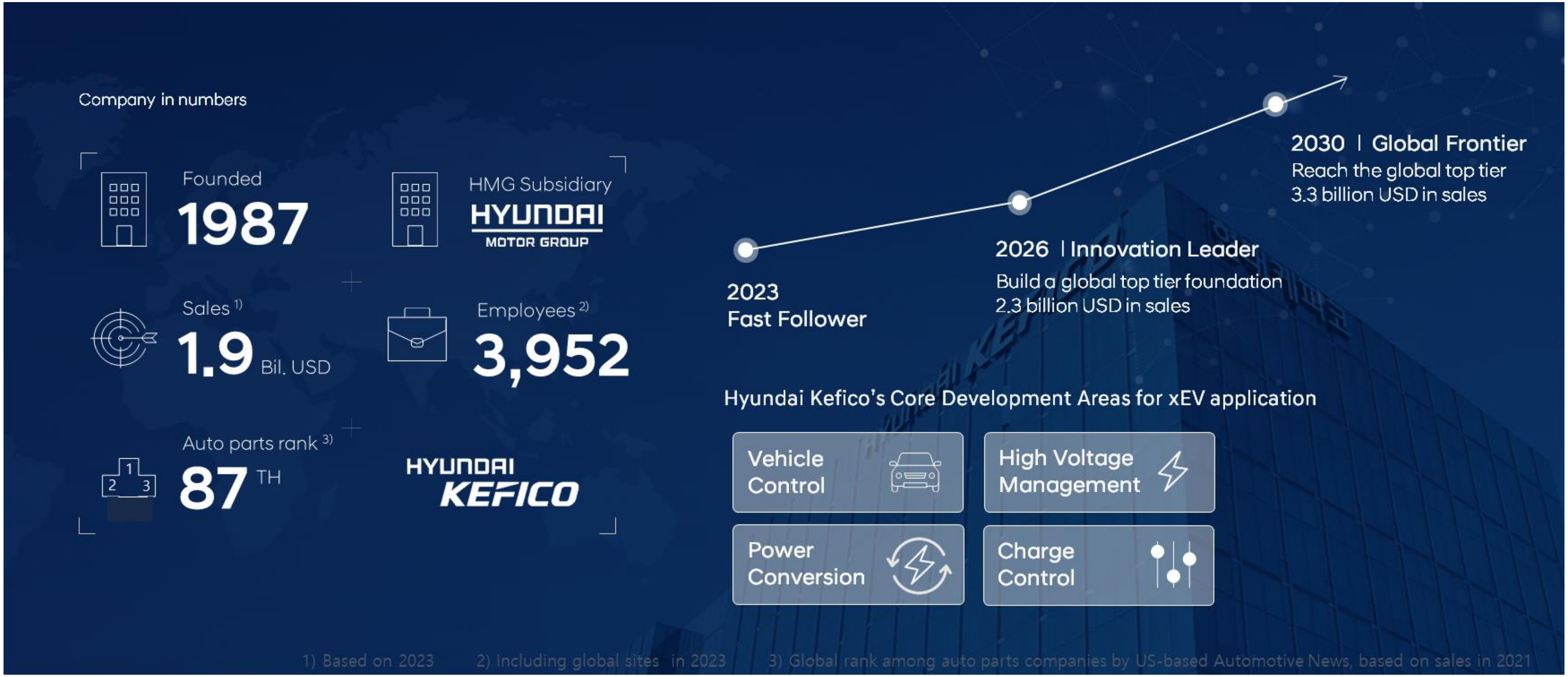
HYUNDAI KEFICO?

MathWorks

HYUNDAI KEFICO

HYUNDAI MOTOR GROUP

Hyundai KEFICO Introduction



Company in numbers

Founded
1987

HMG Subsidiary
HYUNDAI MOTOR GROUP

Sales [1]
1.9 Bil. USD

Employees [2]
3,952

Auto parts rank [3]
87 TH

HYUNDAI KEFICO

2023
Fast Follower

2026 | Innovation Leader
Build a global top tier foundation
2.3 billion USD in sales

2030 | Global Frontier
Reach the global top tier
3.3 billion USD in sales

Hyundai Kefico's Core Development Areas for xEV application

- Vehicle Control
- High Voltage Management
- Power Conversion
- Charge Control

[1] Based on 2023
[2] Including global sites in 2023
[3] Global rank among auto parts companies by US-based Automotive News, based on sales in 2021

Hyundai KEFICO Introduction

Organization
4 Centers / 19 Teams / 1 Lab / 1 TFT

R&D Employees
640 (55% of all)

No. of Equipment
496

RESEARCH DESIGN VALIDATION CALIBRATION

Electrification Development Center	Control Solution Center	Design Center	Reliability Center
▪ New business technology - EV Charger, Power Conversion System, Micro-Mobility ▪ Advanced Technology Lab 	▪ Control Unit Design (SW/HW), Integration and Verification ▪ System Architecture / Platform Development 	▪ Product Design - Sensor, Actuator, Module ▪ CAE 	▪ Reliability Test / Evaluation ▪ Prototype Development ▪ Control System Validation

(As of Jan. 1st, 2024)

What is HYUNDAI KEFICO developing lately?

R&D Objectives

Global Top Mobility Control Solution Specialist

Powertrain Control System

- Optimization and Stabilization -

- Securing Cost Competitiveness
  - Global new regulations for powertrain system
- Stabilization of EV 2-Wheelers Control system

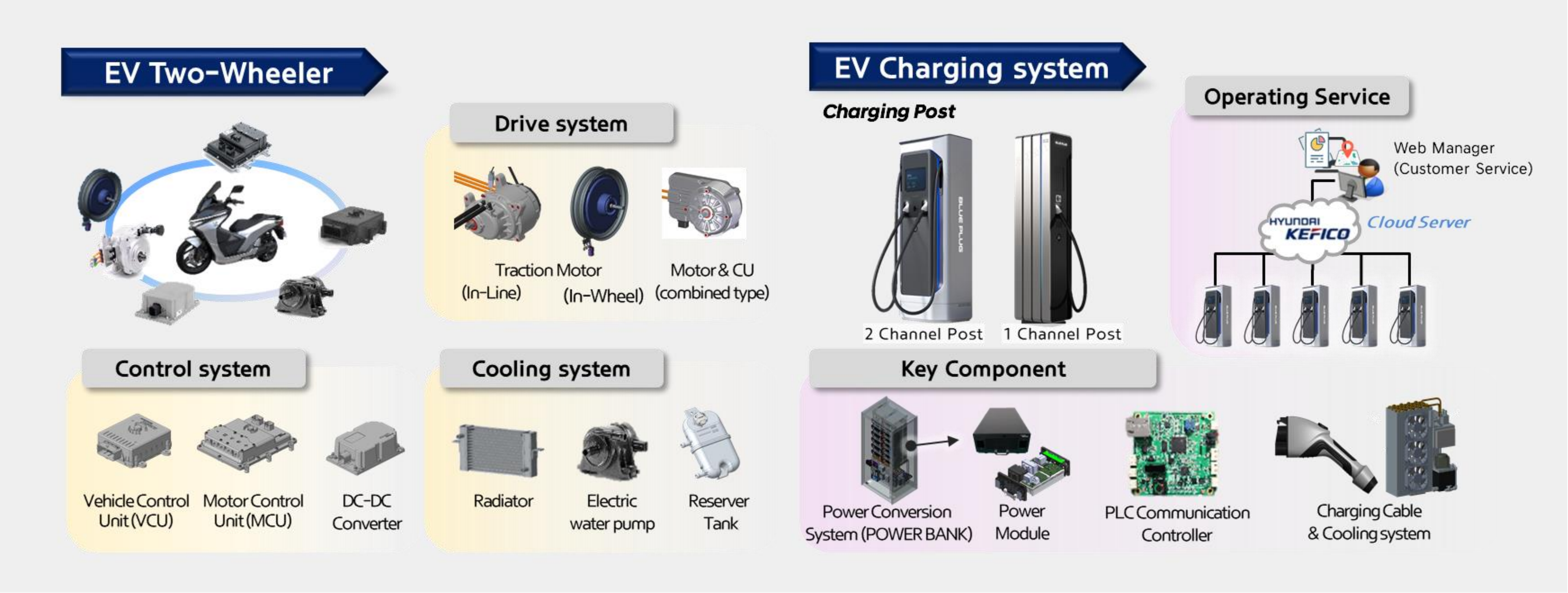
Eco-friendly Control System

- Expansion with Variety -

- Expansion of EV Charging System Portfolio
  - Power conversion & Charging control unit
  - Fast charging system
- High-Performance Vehicle Platform Control (VPC-P)

HYUNDAI KEFICO has a new business?

New Business



EV Two-Wheeler

Drive system

Traction Motor
(In-Line)

(In-Wheel)

Motor & CU
(combined type)

Control system

Vehicle Control
Unit (VCU)

Motor Control
Unit (MCU)

DC-DC
Converter

Cooling system

Radiator

Electric
water pump

Reserver
Tank

EV Charging system

Charging Post

2 Channel Post

1 Channel Post

Operating Service

Web Manager
(Customer Service)

HYUNDAI KEFICO

Cloud Server

Key Component

Power Conversion
System (POWER BANK)

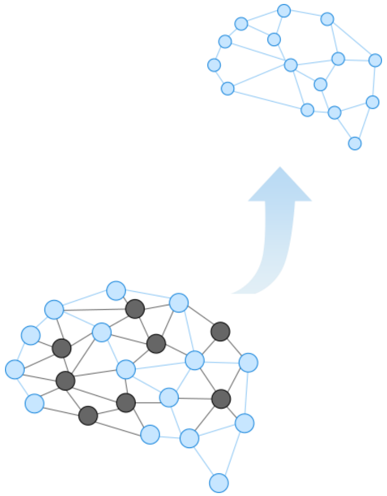
Power
Module

PLC Communication
Controller

Charging Cable
& Cooling system

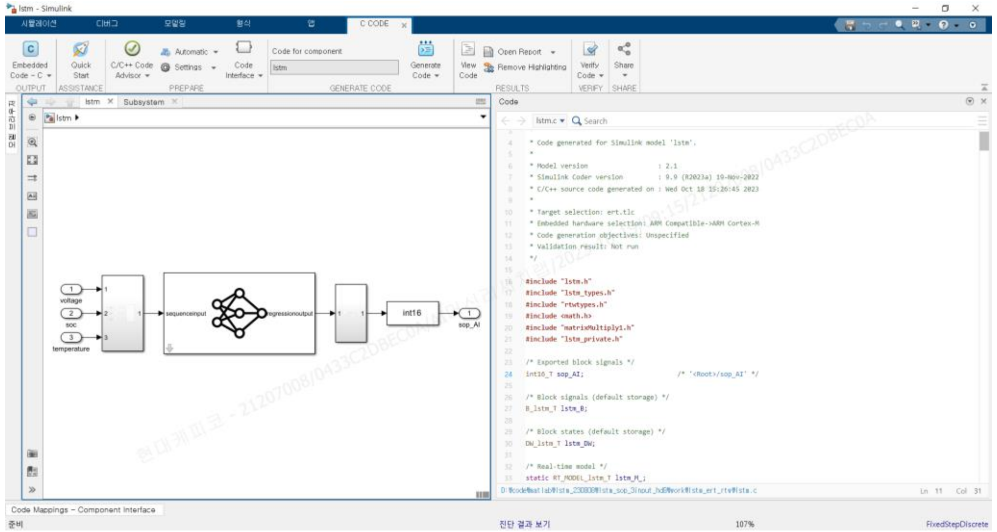
On-Device AI Feasibility of Mobility Controllers

- Running a deep learning model
- Inference time
- Memory usage

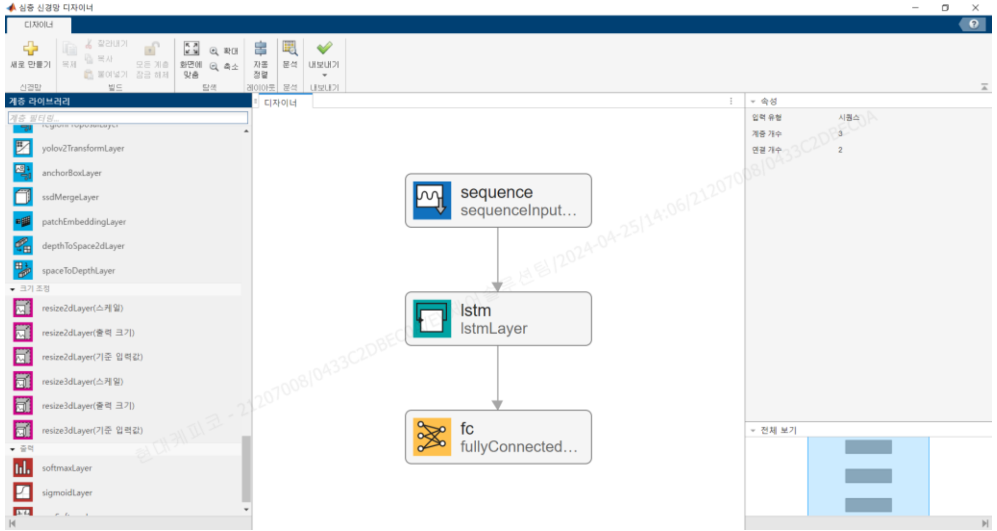


MATLAB & Simulink is easy…

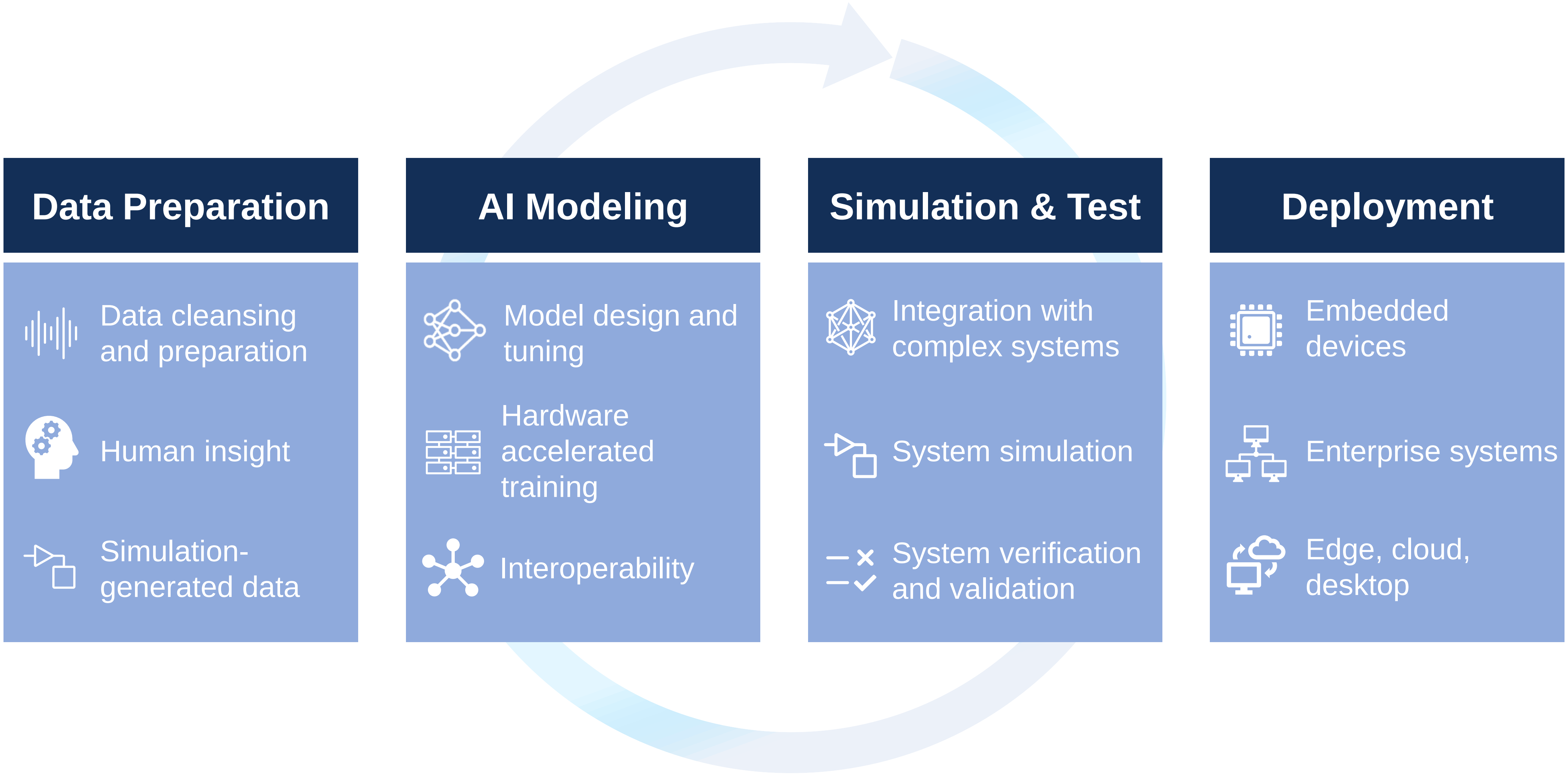
- to integrate AI models with control logic designed with existing MBD
- to implement On-Device AI using code generation of MATLAB
- to development deep learning models



```
 * Code generated for Simulink model 'lstm'.
 *
 * Model version                  : 2.1
 * Simulink Coder version         : 9.9 (R2023a) 19-Nov-2022
 * C/C++ source code generated on : Wed Oct 18 15:26:45 2023
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: ARM Compatible->ARM Cortex-M
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "lstm.h"
#include "lstm_types.h"
#include "rtwtypes.h"
#include <math.h>
#include "matrixMultiply1.h"
#include "lstm_private.h"

/* Exported block signals */
int16_T sop_AI;                       /* '<Root>/sop_AI' */

/* Block signals (default storage) */
B_lstm_T lstm_B;

/* Block states (default storage) */
DW_lstm_T lstm_DW;

/* Real-time model */
static RT_MODEL_lstm_T lstm_M_;
```



On-Device AI Process using MATLAB & Simulink



Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Simulation & Test

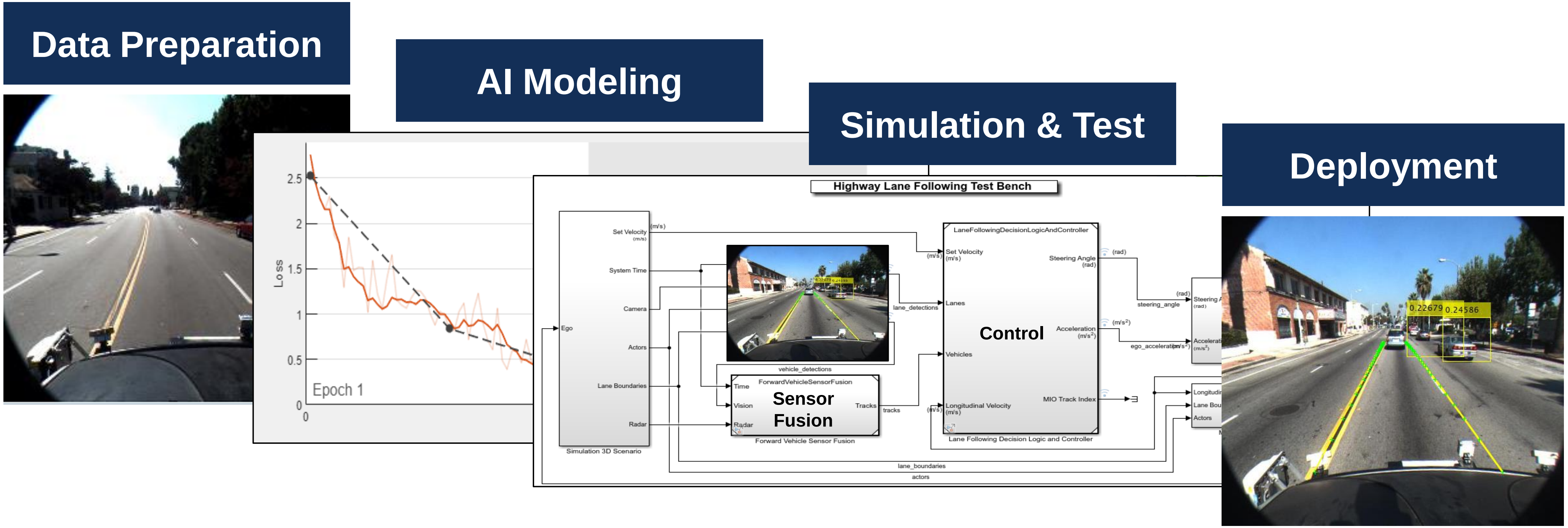
- Integration with complex systems
- System simulation
- System verification and validation

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

On-Device AI Process using MATLAB & Simulink

- AI for mobility detection



Data Preparation

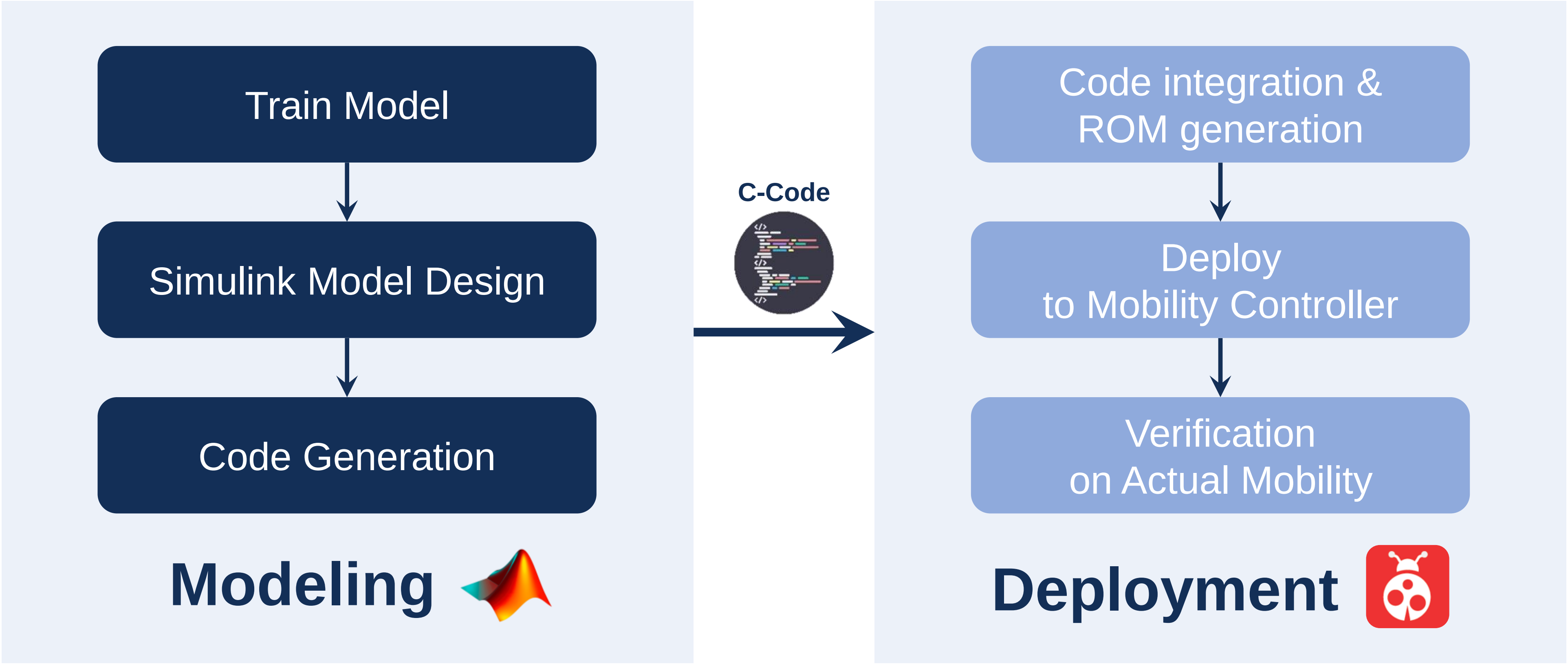
AI Modeling

Simulation & Test

Deployment

On-Device AI Process using MATLAB & Simulink

- Is the MATLAB version different?
- Integration on Simulink or Integration on C code



Train Model

Simulink Model Design

Code Generation

Modeling

C-Code

Code integration & ROM generation

Deploy to Mobility Controller

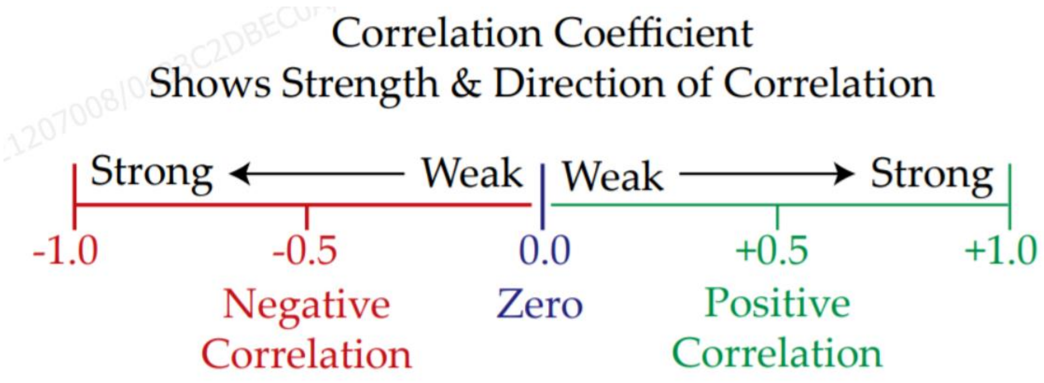
Verification on Actual Mobility

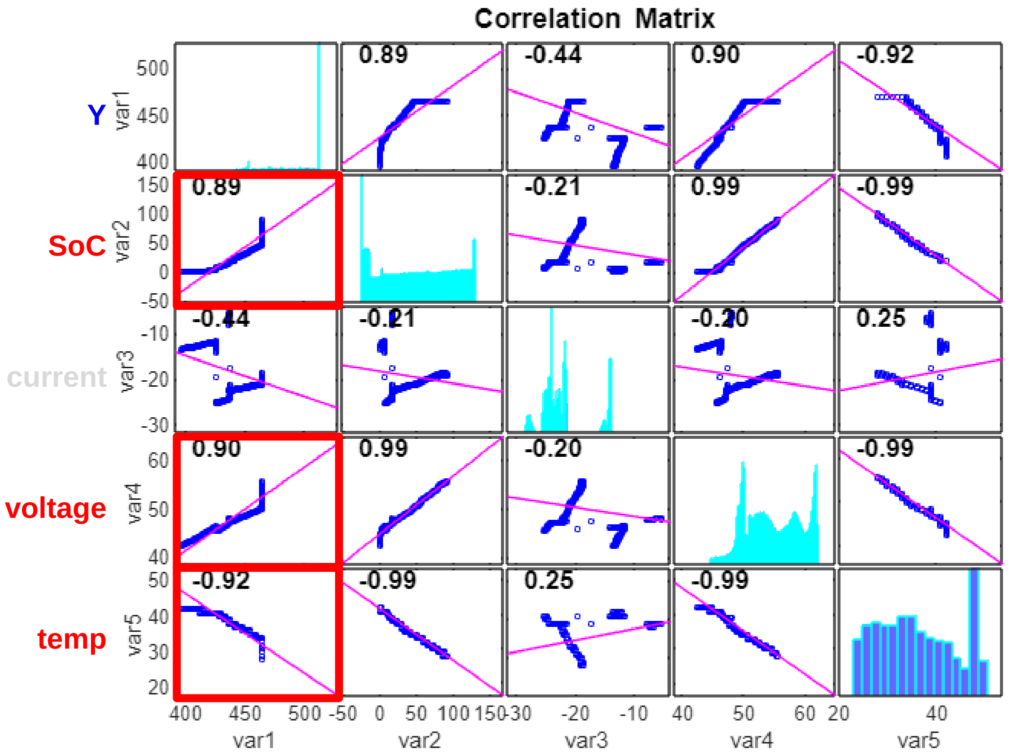
Deployment

Data Preparation

- Model 1. LSTM model for battery-related variables prediction
- Model 2. Binary Classification model
- Pearson’s correlation coefficients

```
[r, p_value] = corrplot(data_corr)
```



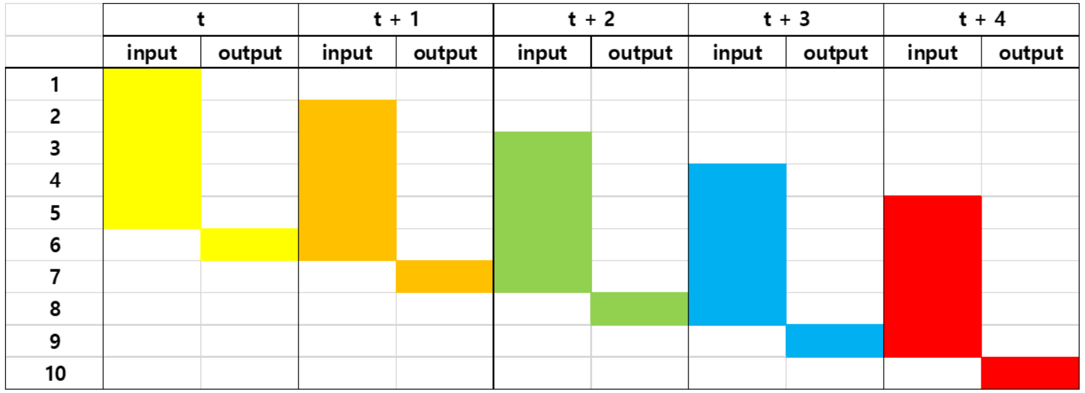


Data Preparation

- Sequence data set

ex) Prediction of 1 data in the future using 5 data in the past(window size)

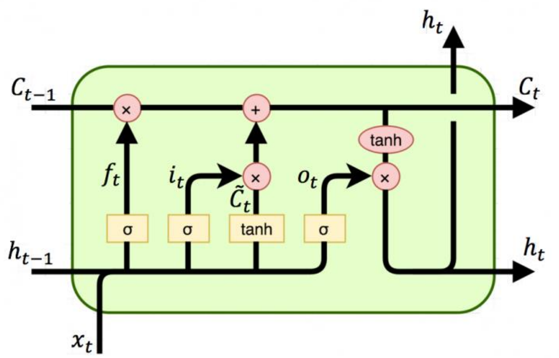
```
win_size = 5;
last = size(voltage, 1) - win_size;

x_data = cell([last 1]);
y_data = zeros(last, 1);

for i=1:last
    x_data{i, 1} = [...
        voltage(i:i+win_size-1)';...
        soc(i:i+win_size-1)';...
        temp(i:i+win_size-1)'];
    y_data(i,1) = Y(i+win_size);
end
```

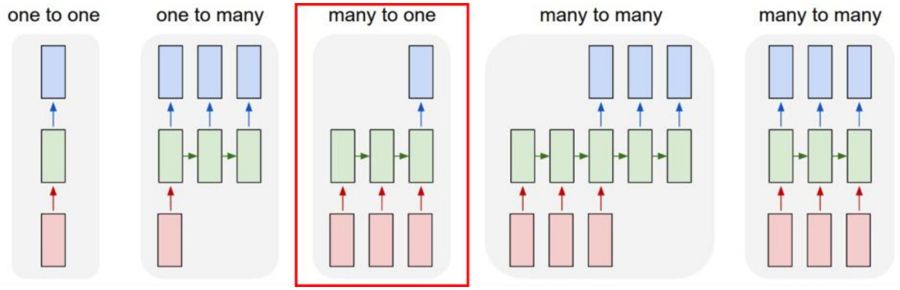


	t		t + 1		t + 2		t + 3		t + 4	
	input	output	input	output	input	output	input	output	input	output
1	🟨									
2	🟨		🟧							
3	🟨		🟧		🟩					
4	🟨		🟧		🟩		🟦			
5	🟨		🟧		🟩		🟦		🟥	
6		🟨	🟧		🟩		🟦		🟥	
7				🟧	🟩		🟦		🟥	
8						🟩	🟦		🟥	
9								🟦	🟥	
10										🟥

AI Model Design

- Using deepNetworkDesigner
- LSTM Long Sort-Term Memory





```
input_size = size(x_train{1}, 1);
output_size = size(y_train(1), 1);
hidden_size = 8;

layers = [
       sequenceInputLayer(input_size,
       "Normalization","zscore", "MinLength",5)
       lstmLayer(hidden_size,"OutputMode","last")
       fullyConnectedLayer(output_size)
       regressionLayer]
```

```
layers =
  다음 계층을 포함한 4×1 Layer 배열:
    1 '' 시퀀스 입력 시퀀스 입력 (차원 3개)
    2 '' LSTM LSTM (은닉 유닛 8개)
    3 '' 완전 연결 1 완전 연결 계층
    4 '' 회귀 출력 mean-squared-error
```

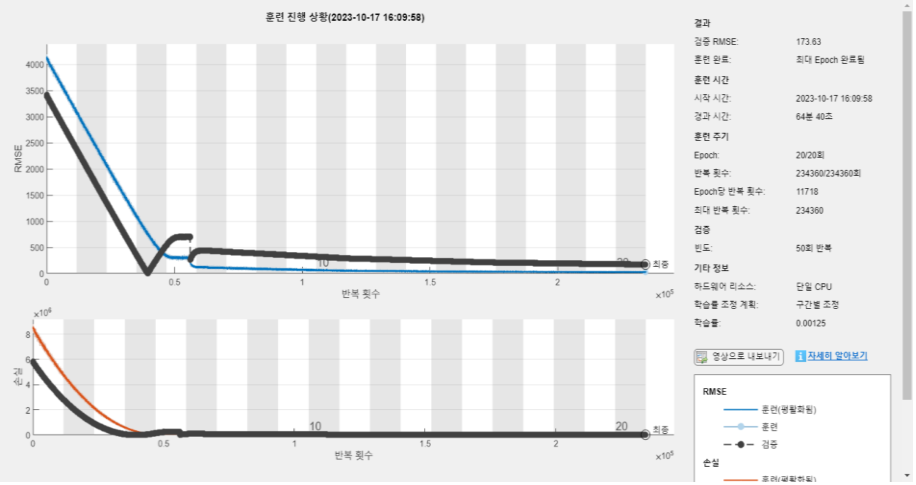
AI Model Training

```
maxEpochs = 20;

options = trainingOptions('adam', ...
    'LearnRateSchedule', "piecewise", ...
    'LearnRateDropFactor', 0.5, ...
    'LearnRateDropPeriod', 5, ...
    'MaxEpochs',maxEpochs, ...
    'ValidationData',{x_validation y_validation}, ...
    'OutputNetwork', 'last-iteration', ...
    'InitialLearnRate',0.01, ...
    'Plots', 'training-progress', ...
    'Verbose',0)
```

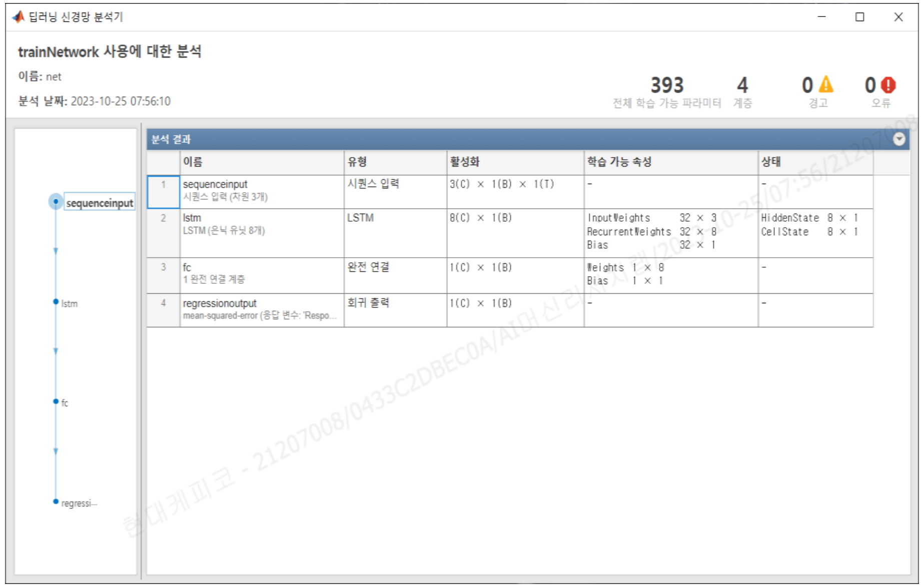
```
net = trainNetwork(x_train,y_train,layers,options);
```

```
save("lstm.mat",'net')
exportONNXNetwork(net, "lstm.onnx")
```

```
analyzeNetwork(net)
```



결과
검증 RMSE: 173.63
훈련 완료: 최대 Epoch 완료됨
훈련 시간
시작 시간: 2023-10-17 16:09:58
경과 시간: 64분 40초
훈련 주기
Epoch: 20/20회
반복 횟수: 234360/234360회
Epoch당 반복 횟수: 11718
최대 반복 횟수: 234360
검증
빈도: 50회 반복
기타 정보
하드웨어 리소스: 단일 CPU
학습률 조정 계획: 구간별 조정
학습률: 0.00125



trainNetwork 사용에 대한 분석

이름: net

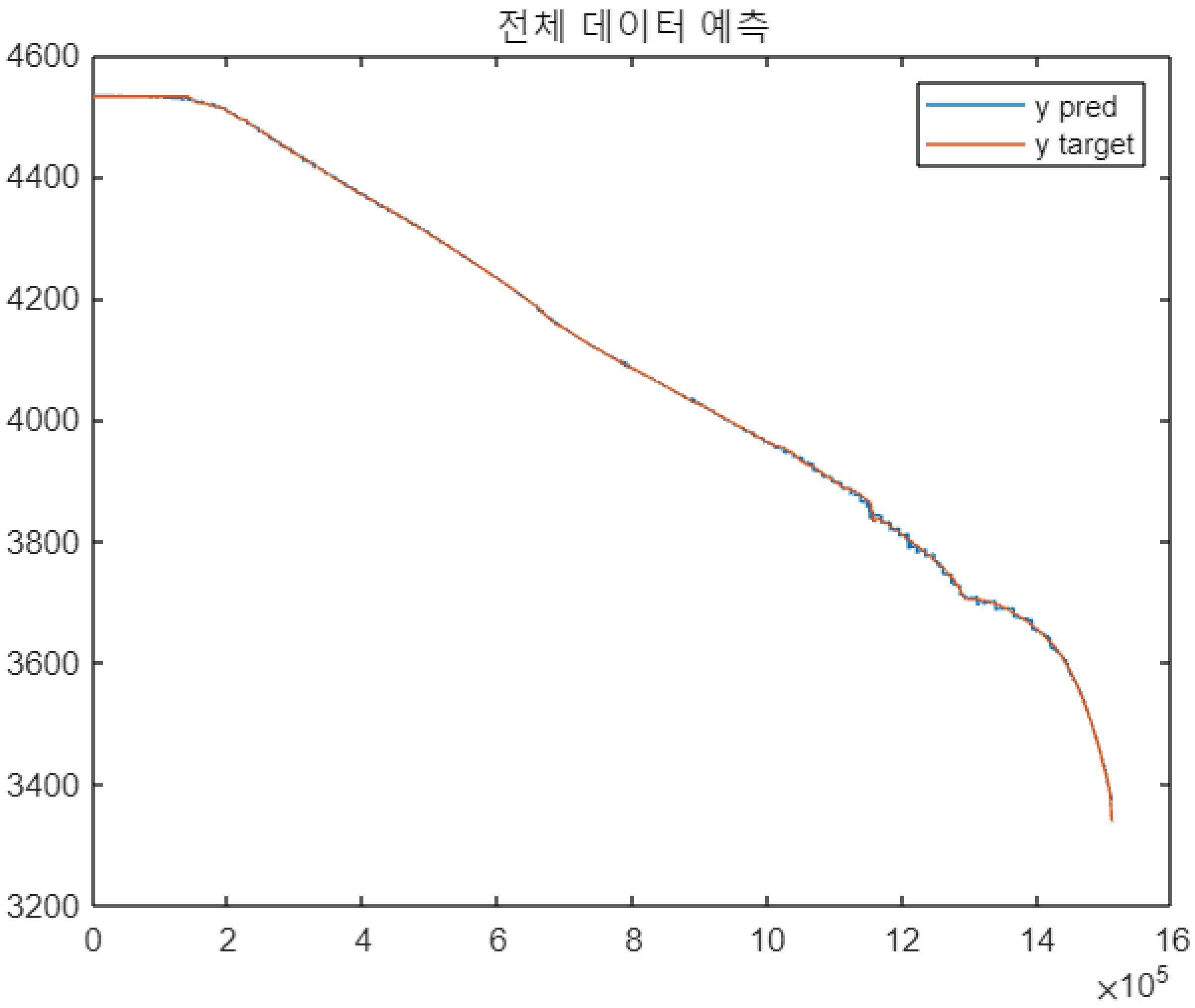
분석 날짜: 2023-10-25 07:56:10

393 전체 학습 가능 파라미터 · 4 계층 · 0 경고 · 0 오류

	이름	유형	활성화	학습 가능 속성	상태
1	sequenceinput 시퀀스 입력 (차원 3개)	시퀀스 입력	3(C) × 1(B) × 1(T)	-	-
2	lstm LSTM (은닉 유닛 8개)	LSTM	8(C) × 1(B)	InputWeights 32 × 3 RecurrentWeights 32 × 8 Bias 32 × 1	HiddenState 8 × 1 CellState 8 × 1
3	fc 1 완전 연결 계층	완전 연결	1(C) × 1(B)	Weights 1 × 8 Bias 1 × 1	-
4	regressionoutput mean-squared-error (응답 변수: 'Respo...	회귀 출력	1(C) × 1(B)	-	-

21207008/0433C2DBEC0A/AI 2023-10-25/07:56/21207008

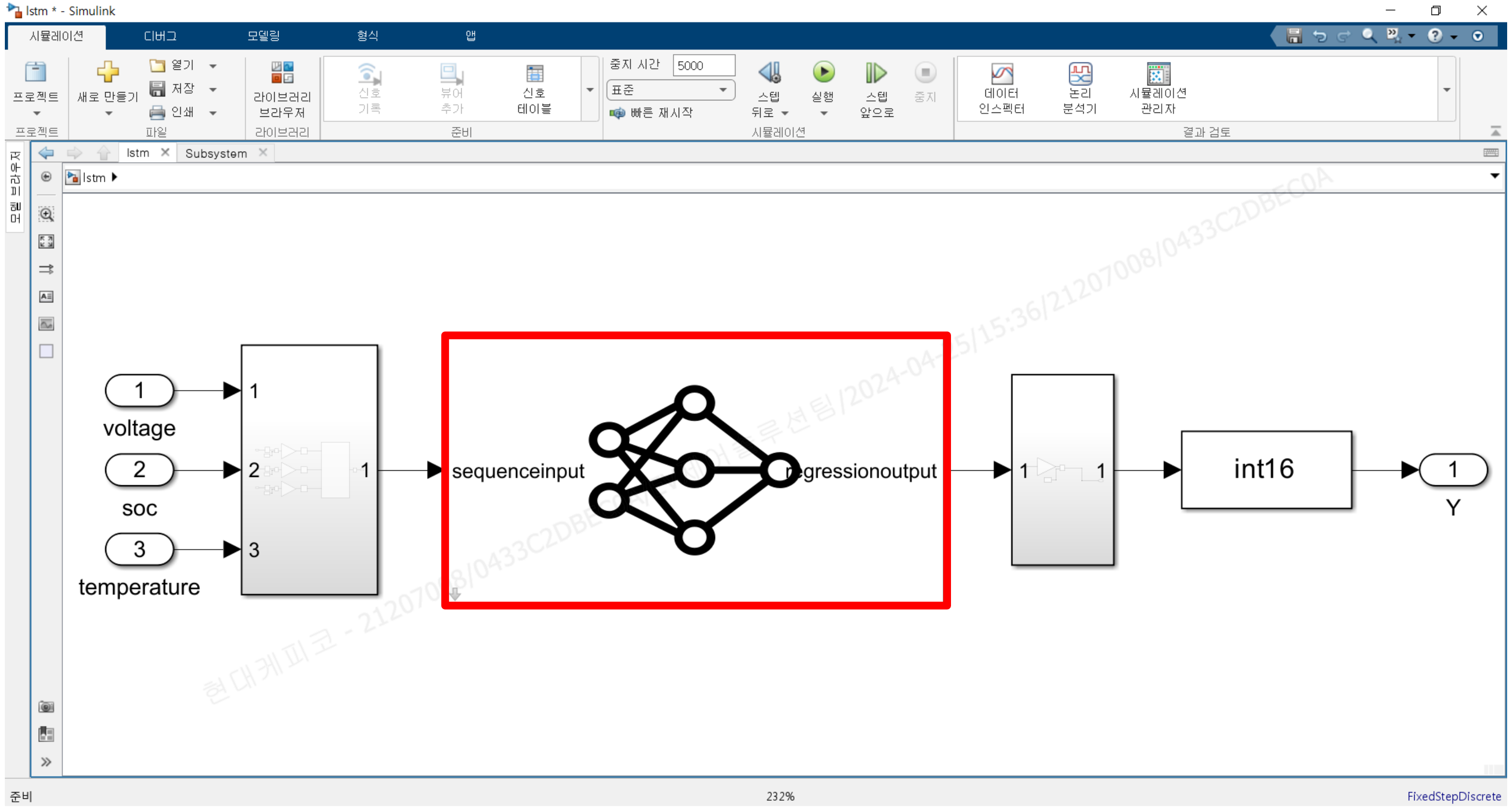
AI Model Training



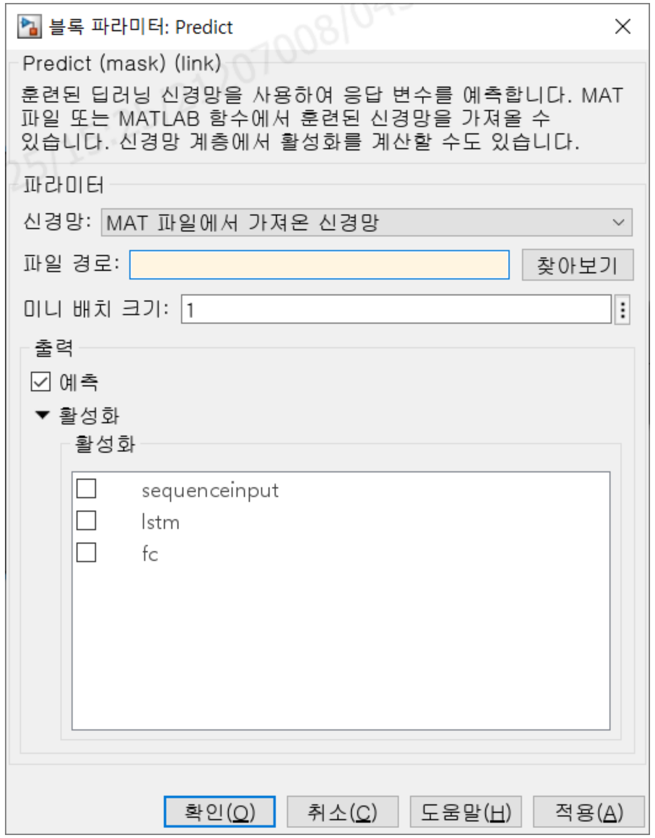
```
predict(net, x_train,'MiniBatchSize',1)
```

MBD Design

- Design of Simulink model for MBD-based code generation of learned AI model



현대케피코 - 21207008/0433C2DBEC0A
2024-04-25/15:36/21207008/0433C2DBEC0A


블록 파라미터: Predict

Predict (mask) (link)

훈련된 딥러닝 신경망을 사용하여 응답 변수를 예측합니다. MAT 파일 또는 MATLAB 함수에서 훈련된 신경망을 가져올 수 있습니다. 신경망 계층에서 활성화를 계산할 수도 있습니다.

파라미터

신경망: MAT 파일에서 가져온 신경망

파일 경로:

미니 배치 크기: 1

출력

☑ 예측

활성화

- □ sequenceinput
- □ lstm
- □ fc

Data Preparation

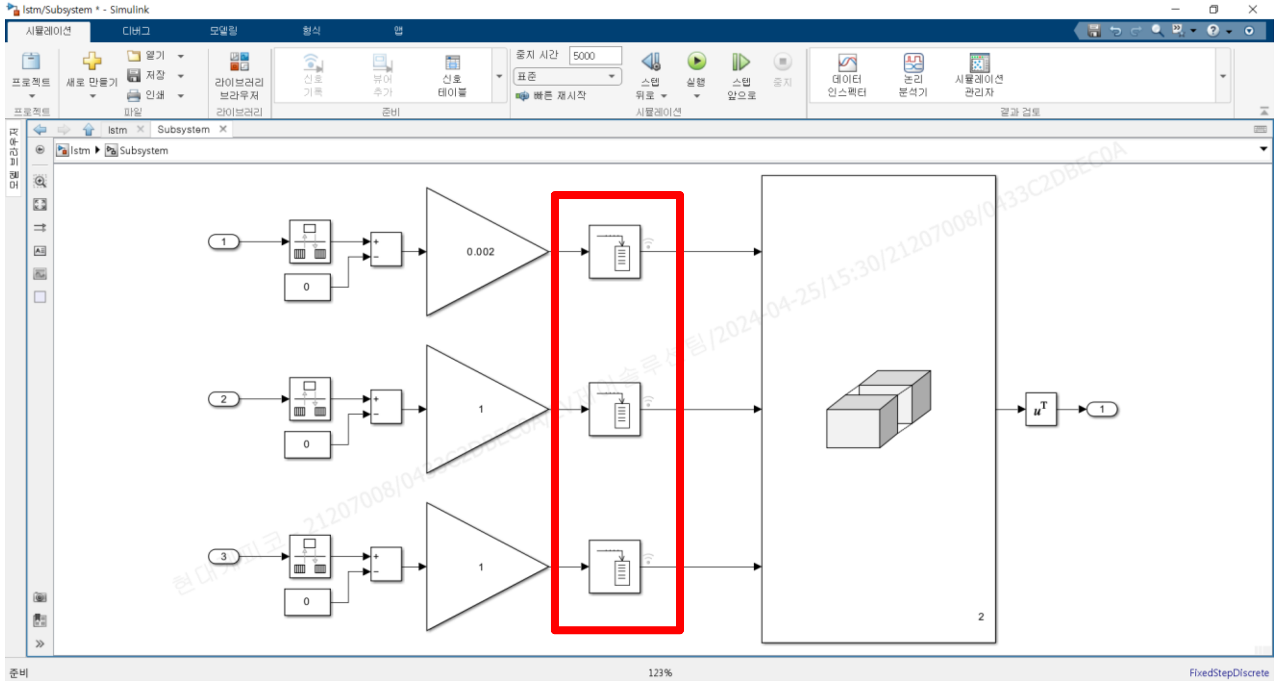
AI Modeling

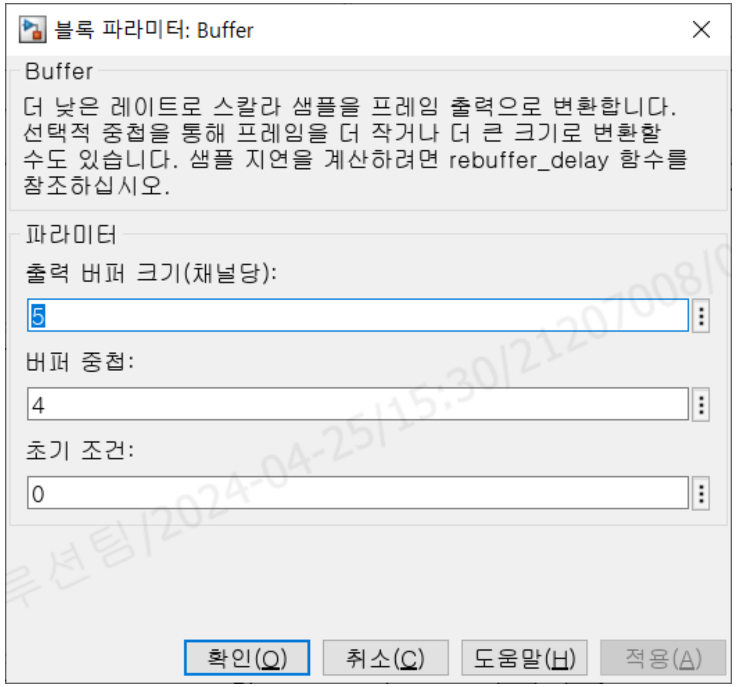
Simulation & Test

Deployment

MBD Design

- Set buffer block for sequence data
- Set buffer size equal to window size
- Buffer overlap = window size - 1





블록 파라미터: Buffer

Buffer

더 낮은 레이트로 스칼라 샘플을 프레임 출력으로 변환합니다. 선택적 중첩을 통해 프레임을 더 작거나 더 큰 크기로 변환할 수 있습니다. 샘플 지연을 계산하려면 rebuffer_delay 함수를 참조하십시오.

파라미터

출력 버퍼 크기(채널당): 5

버퍼 중첩: 4

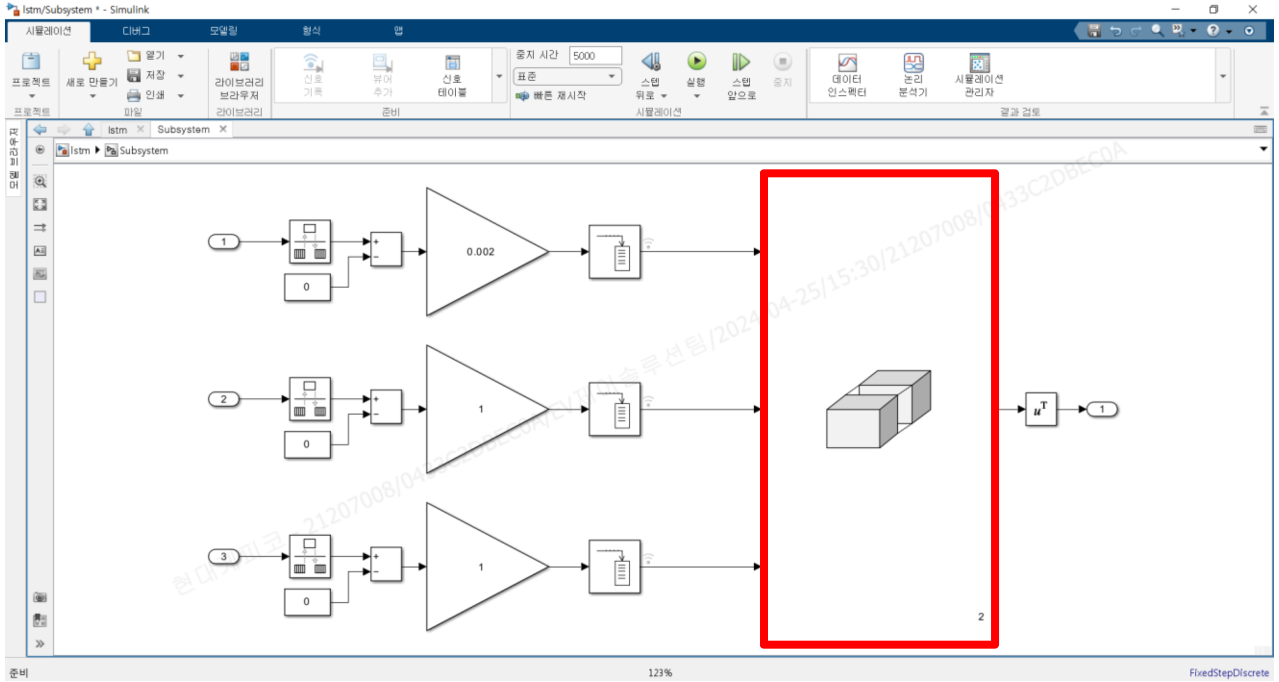
초기 조건: 0

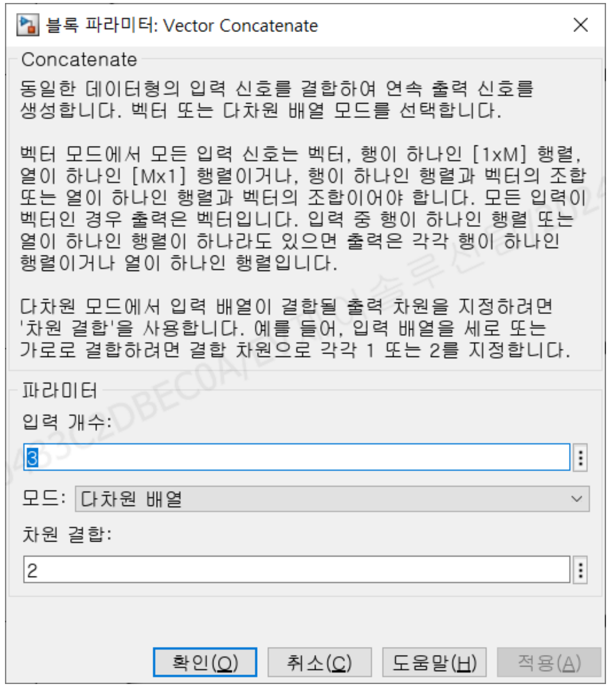
확인(O) 취소(C) 도움말(H) 적용(A)

루션팀/2024-04-25/15:30/21207008/0433C2DBEC0A

MBD Design

- Set concatenate block for sequence data
- Input size = 3
- Dimension combining = 2




블록 파라미터: Vector Concatenate

Concatenate

동일한 데이터형의 입력 신호를 결합하여 연속 출력 신호를 생성합니다. 벡터 또는 다차원 배열 모드를 선택합니다.

벡터 모드에서 모든 입력 신호는 벡터, 행이 하나인 [1xM] 행렬, 열이 하나인 [Mx1] 행렬이거나, 행이 하나인 행렬과 벡터의 조합 또는 열이 하나인 행렬과 벡터의 조합이어야 합니다. 모든 입력이 벡터인 경우 출력은 벡터입니다. 입력 중 행이 하나인 행렬 또는 열이 하나인 행렬이 하나라도 있으면 출력은 각각 행이 하나인 행렬이거나 열이 하나인 행렬입니다.

다차원 모드에서 입력 배열이 결합될 출력 차원을 지정하려면 '차원 결합'을 사용합니다. 예를 들어, 입력 배열을 세로 또는 가로로 결합하려면 결합 차원으로 각각 1 또는 2를 지정합니다.

파라미터

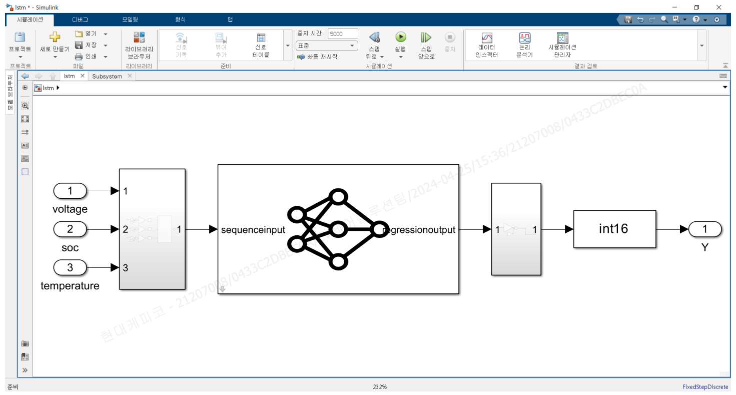
입력 개수: 3

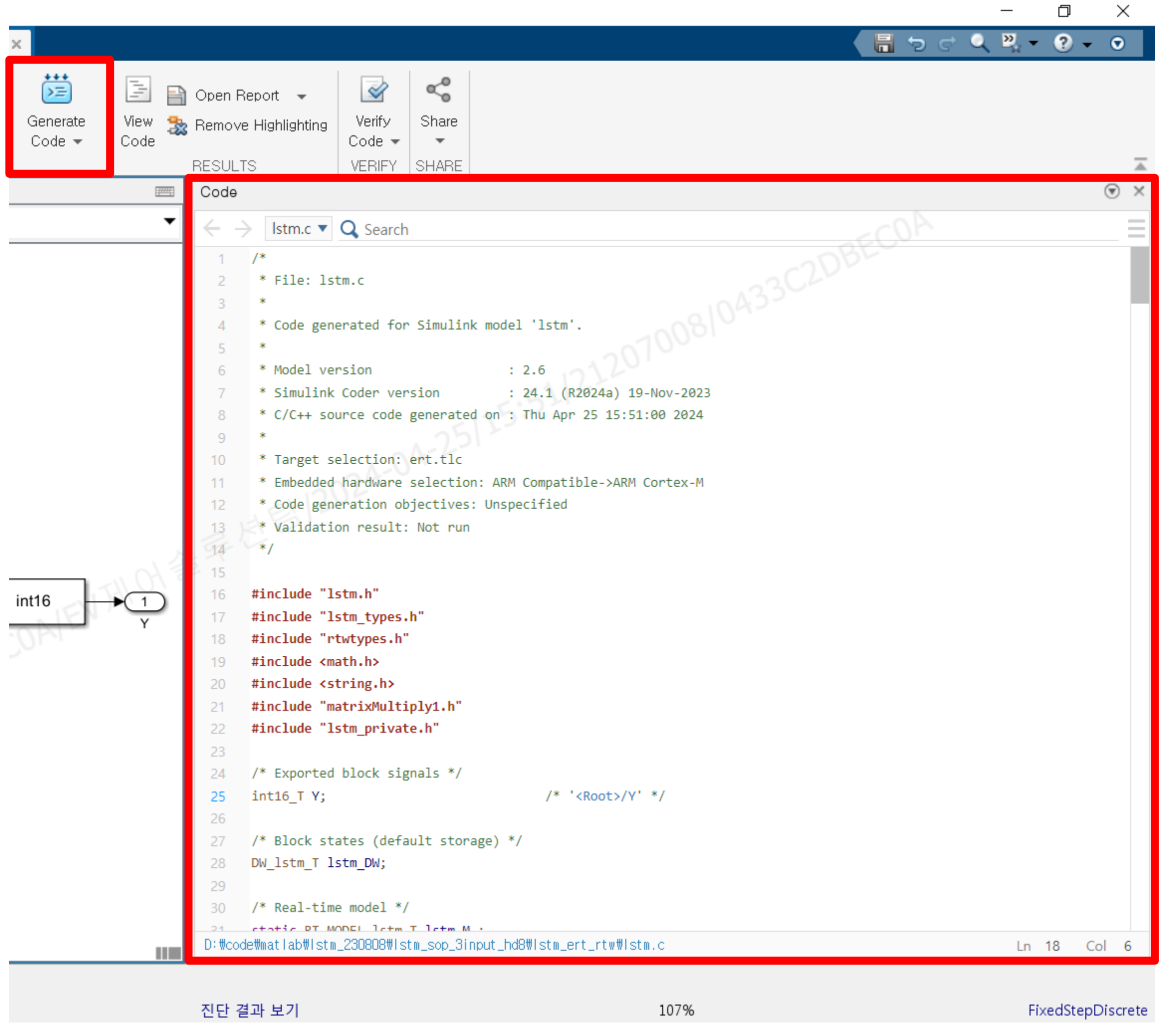
모드: 다차원 배열

차원 결합: 2

Code Generation

- Using Embedded Coder
- Code Mappings
  - Inports: ImportedExtern
  - Outports: ExportedGlobal
- Set target device etc…
- Click [Generate Code]

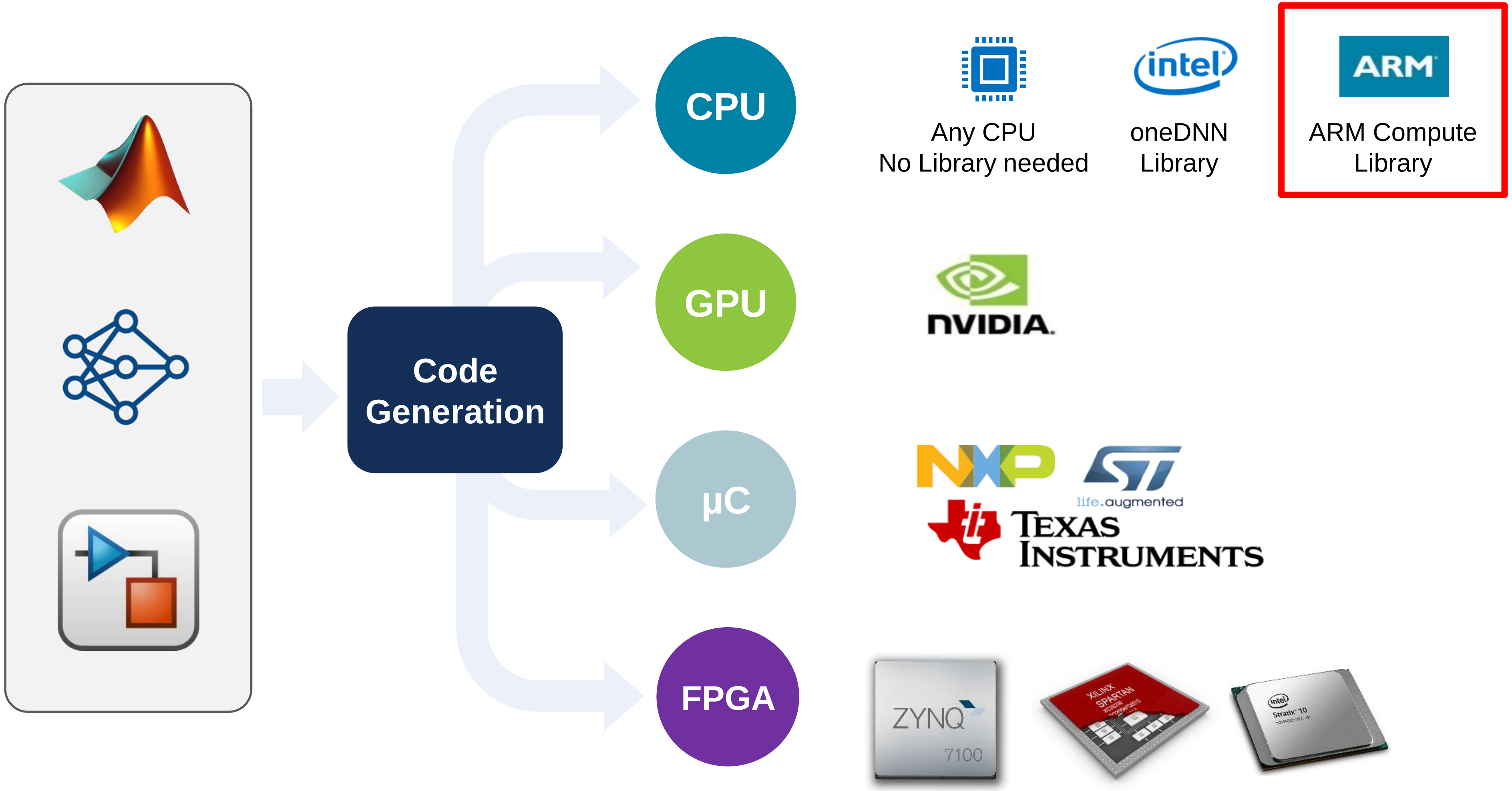




```
/*
 * File: lstm.c
 *
 * Code generated for Simulink model 'lstm'.
 *
 * Model version                  : 2.6
 * Simulink Coder version         : 24.1 (R2024a) 19-Nov-2023
 * C/C++ source code generated on : Thu Apr 25 15:51:00 2024
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: ARM Compatible->ARM Cortex-M
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "lstm.h"
#include "lstm_types.h"
#include "rtwtypes.h"
#include <math.h>
#include <string.h>
#include "matrixMultiply1.h"
#include "lstm_private.h"

/* Exported block signals */
int16_T Y;                             /* '<Root>/Y' */

/* Block states (default storage) */
DW_lstm_T lstm_DW;

/* Real-time model */
static RT_MODEL_lstm_T lstm_M_;
```

D:₩code₩matlab₩lstm_230808₩lstm_sop_3input_hd8₩lstm_ert_rtw₩lstm.c   Ln 18   Col 6

Deploy to Mobility Controller



Code Generation

CPU

Any CPU
No Library needed

oneDNN
Library

ARM Compute
Library

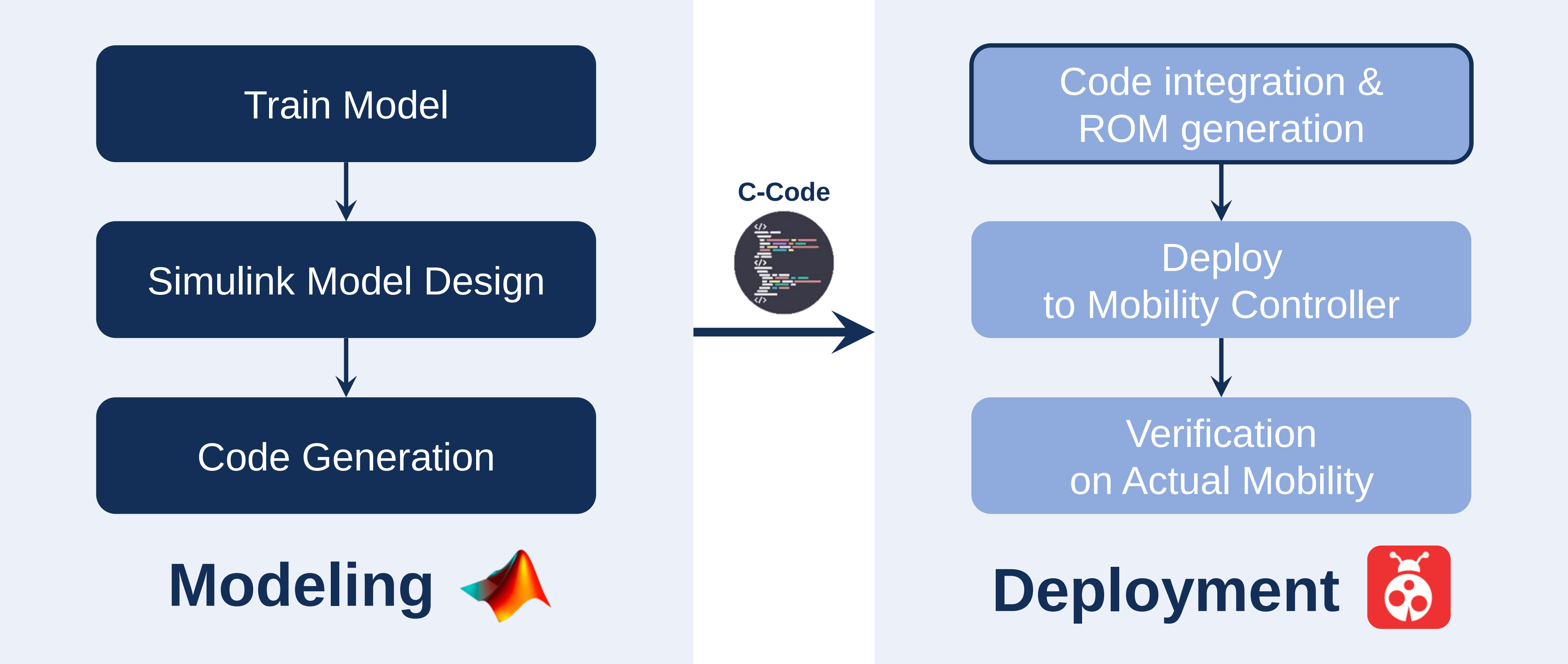
GPU

µC

FPGA

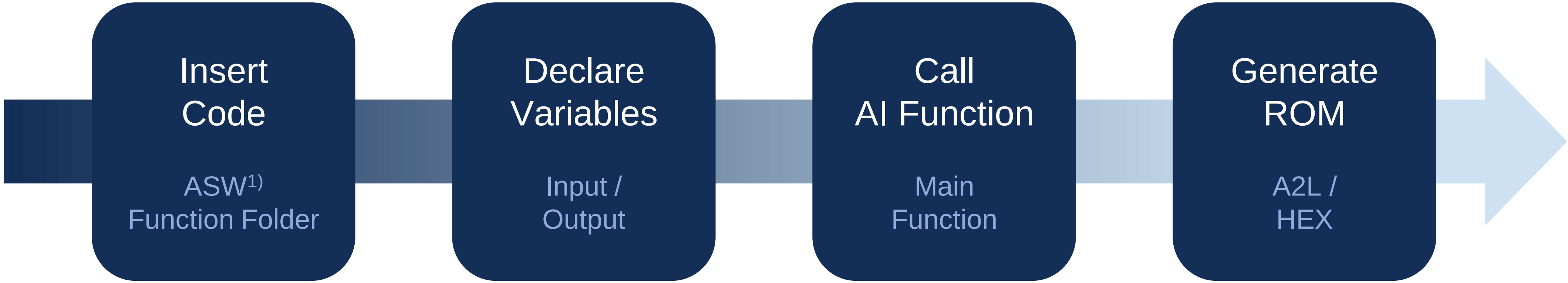
Deploy to Mobility Controller

- Is the MATLAB version different?
- Integration on C code


Train Model
Simulink Model Design
Code Generation
Modeling
C-Code
Code integration & ROM generation
Deploy to Mobility Controller
Verification on Actual Mobility
Deployment

Deploy to Mobility Controller

- Integration of generated code with existing code
- Mobility ROM generation A2L & Hex → Memory usage satisfaction



Insert Code
ASW[1] Function Folder

Declare Variables
Input / Output

Call AI Function
Main Function

Generate ROM
A2L / HEX

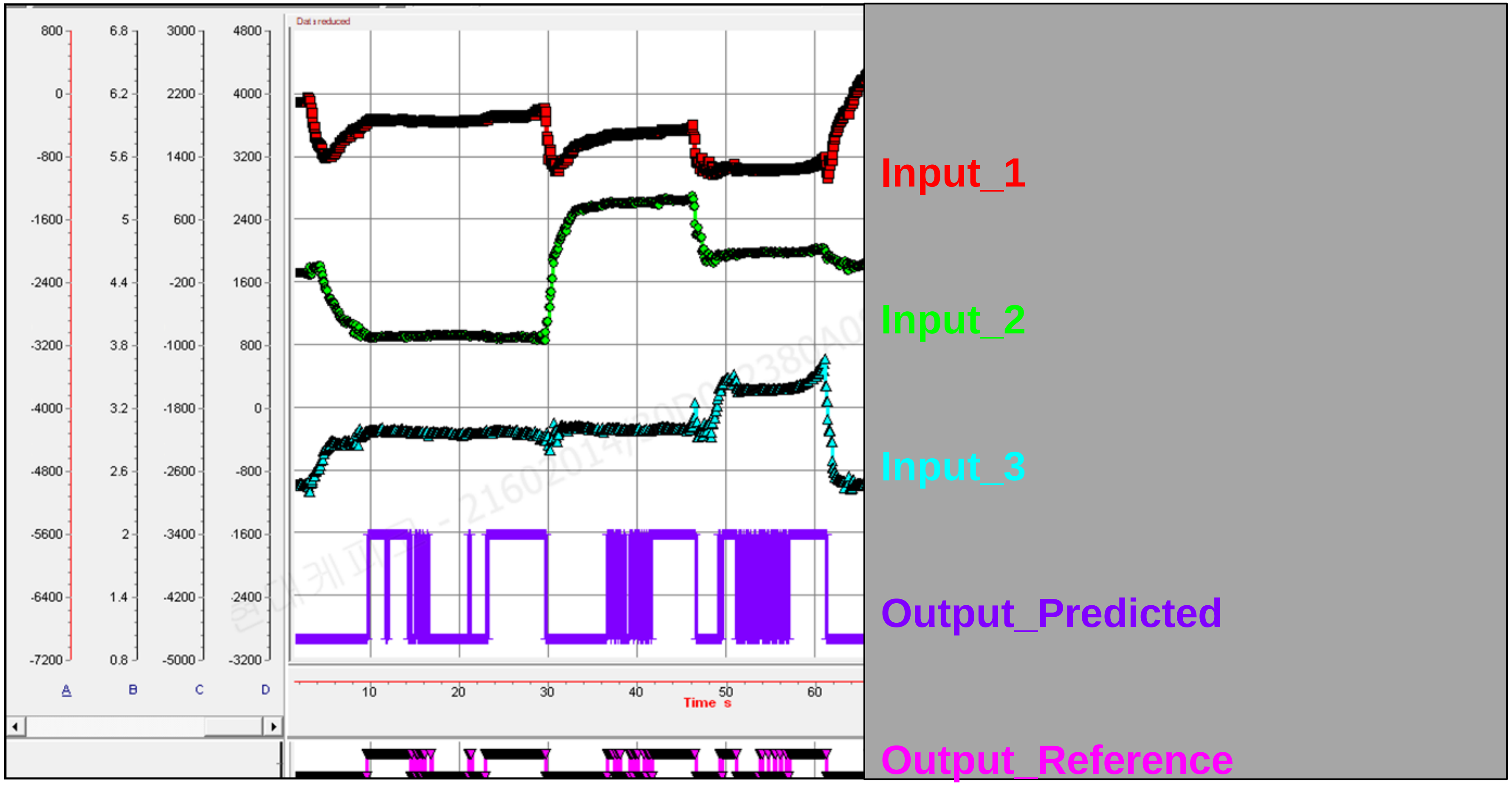
[1] ASW: Application S/W

Verification of Real Time Inference Performance on Controller

		Classification	LSTM
Inference Time [ms]	**ARM Cortex-M4F**	**0.1 ~ 0.2**	**11 ~ 14**
	ARM Cortex-A53	**0.005 ~ 0.068**	**0.121 ~ 0.212**
Train Parameter [#]		62	393
Model Structure		▪ Input Layer: Feature = 3 ▪ FC Layer: Output Size =10 ▪ Relu Layer ▪ FC Layer: Output Size = 2 ▪ Softmax Layer ▪ Output Layer: Classification	▪ Input Layer: Feature = 3, Window Size = 5 ▪ LSTM Layer: Hidden Size = 8 ▪ FC Layer: Output Size =1 ▪ Output Layer: Regression
Verification Result		**Completion of AI model calculation within mobility control logic calculation cycle**	

Verification of Real Time Inference Performance on Controller

- Measurement by ETAS INCA Tool

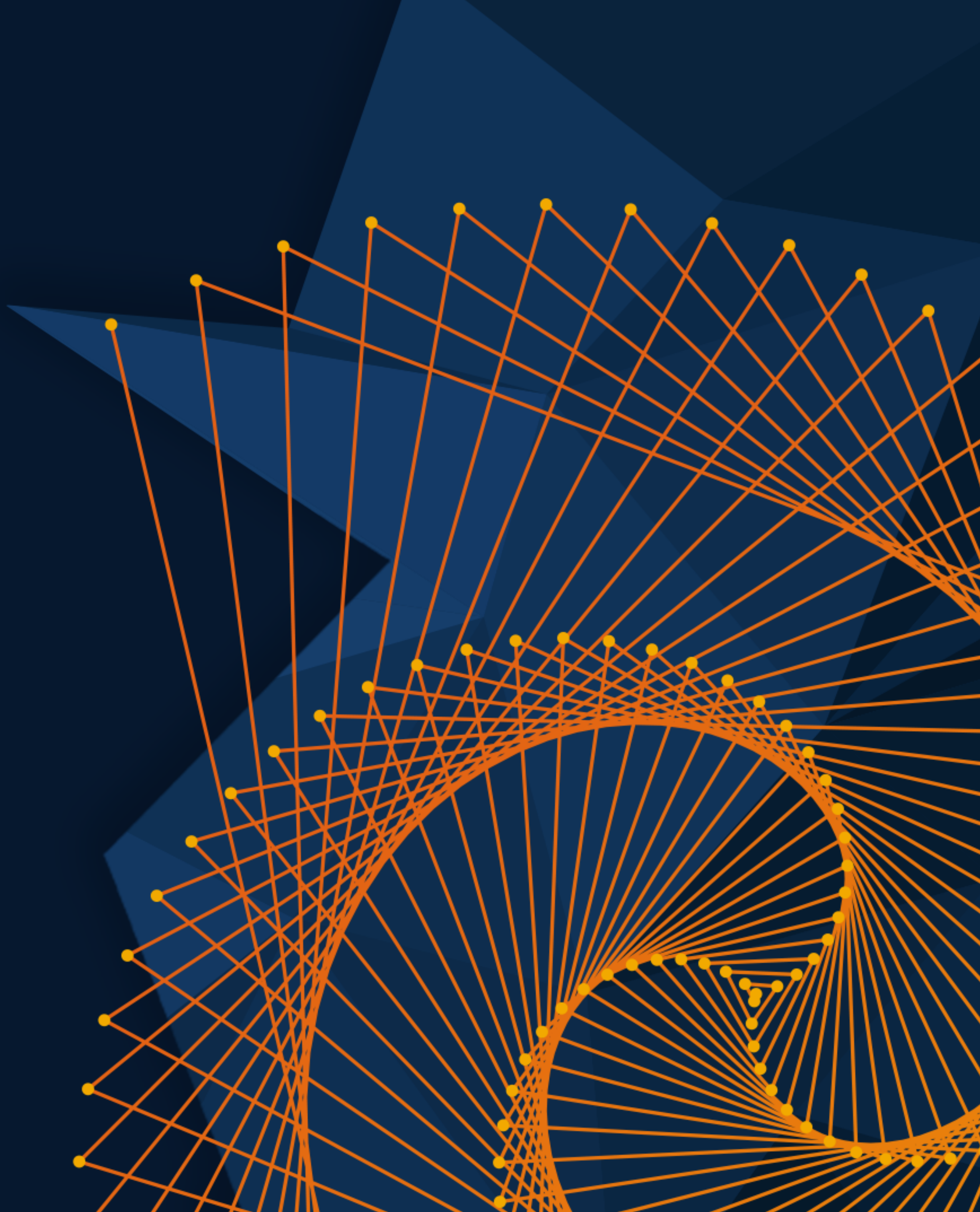


Conclusions and Future Work

- Conclusions
  - The AI model was developed using MATLAB & Simulink and deployed to our mobility controller.
  - The feasibility of AI model execution was confirmed on our mobility controller.
    - Mild increase in memory usage before and after deploying AI model
    - Completion of AI model calculation within mobility control logic calculation cycle

- Future Work
  - AI model compression using MATLAB & Simulink
    - Quantization
    - Knowledge Distillation
    - Pruning

MATLAB EXPO



MathWorks®

© 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.