MATLAB EXPO

November 13–14, 2024 | Online

From Idea to MCU Deployment: Applying Tiny Machine Learning on FOC for PMSMs

Danilo Pau, Technical Director, IEEE and ST Fellow

System Research and Applications, STMicroelectronics

MathWorks

Motivations for interest

Developing Edge AI in the context of Motor Electrification poses challenges due to the well-known Field Oriented Control technique.

Introducing AI mandates to optimize accuracy, execution speed and energy efficiency, which requires a joint understanding of both AI and motor control systems.

The combination of MathWorks and STMicroelectronics AI methodologies and tools simplifies this process, easing efficient deployment of AI models to MCUs.

Let’s review together how this can be achieved.

life.augmented

Key Words

IDEA

Interoperability

pmsm

Machine Learning

Methodology

Productivity

Development

Automation

Journey

FOC

Non Linearities

Embedded Systems

Methodology

D$

DATA ACQUISITION

D1 Modelize the PMSM FOC control loop.

D2 Define Case Studies based on fast changing speed.

D3 Build datasets with PIDs highlighting their limits.

D4 Run the experiments to record enough data.

M$

MODELING TINY NEURAL NETWORK

M1 Set data container to access the data correctly during training steps.

M2 Devise a feed forward NN for compensating PID errors.

M3 Measure the deployability of the network by ST Edge AI Developer Cloud.

M4 Run the training progress and the seek for the best model configuration during validation.

O$

OPTIMIZATIONS

O1 Seek for the most accurate model on the test set.

O2 Perform Hyper Parameters tuning for the best compromise as model exploring configurations.

O3 Prune the model for low-cost deployment on the device.

O4 Optimize model performance monitoring in the control loop (add the NN to the PID).

MCU$

DEPLOY ON ST MCUs

ST Edge AI Developer Cloud

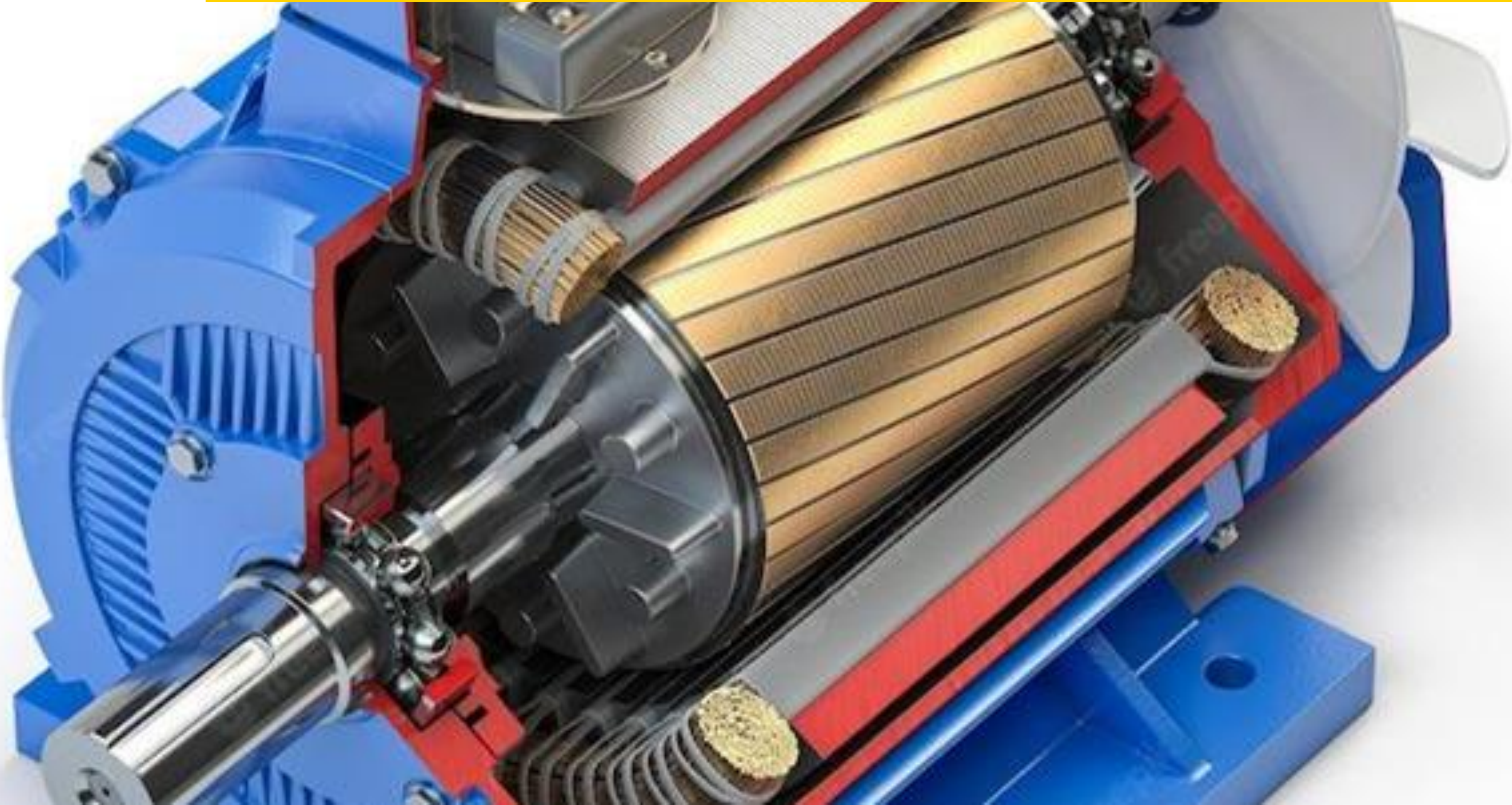
MCU1 Import and PTQ model (ONNX) by the ST Edge AI Developer Cloud.

MCU2 Optimize the deployment (required RAM size w.r.t. inference time).

MCU3 Benchmark on ST MCUs and measure the inference time. Export detailed logs.

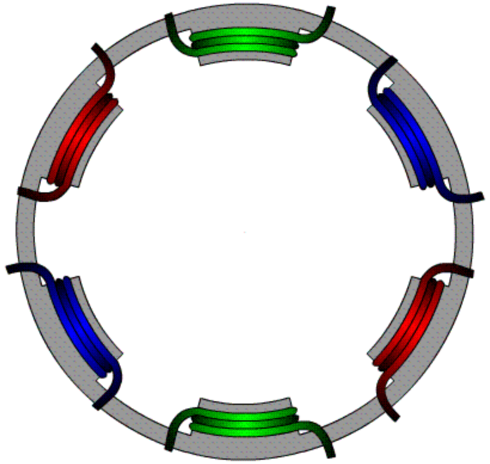
MCU4 Loop between PHASE 2, 3 and 4 until a satisfactory solution can be signed off.

Introduction



Permanent Magnet Synchronous Motor

- The magnetic field of the permanent magnets placed on the rotor interacts with the one created by the synchronous sinusoidal alternating current in the stator windings.
- This interaction produces a torque, which causes the rotor to rotate.
- The EMF (Electromagnetic Field) force shall be controlled to produce the required torque over the time.



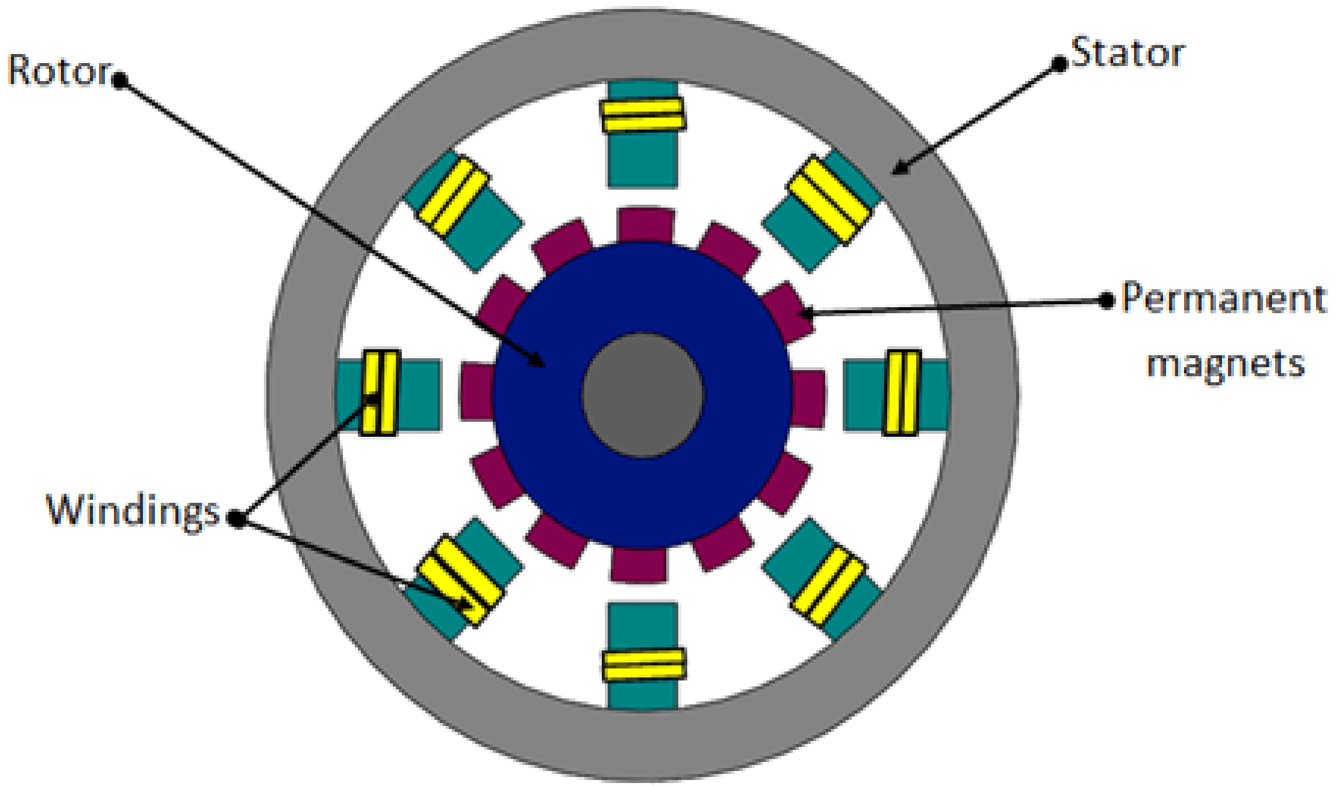
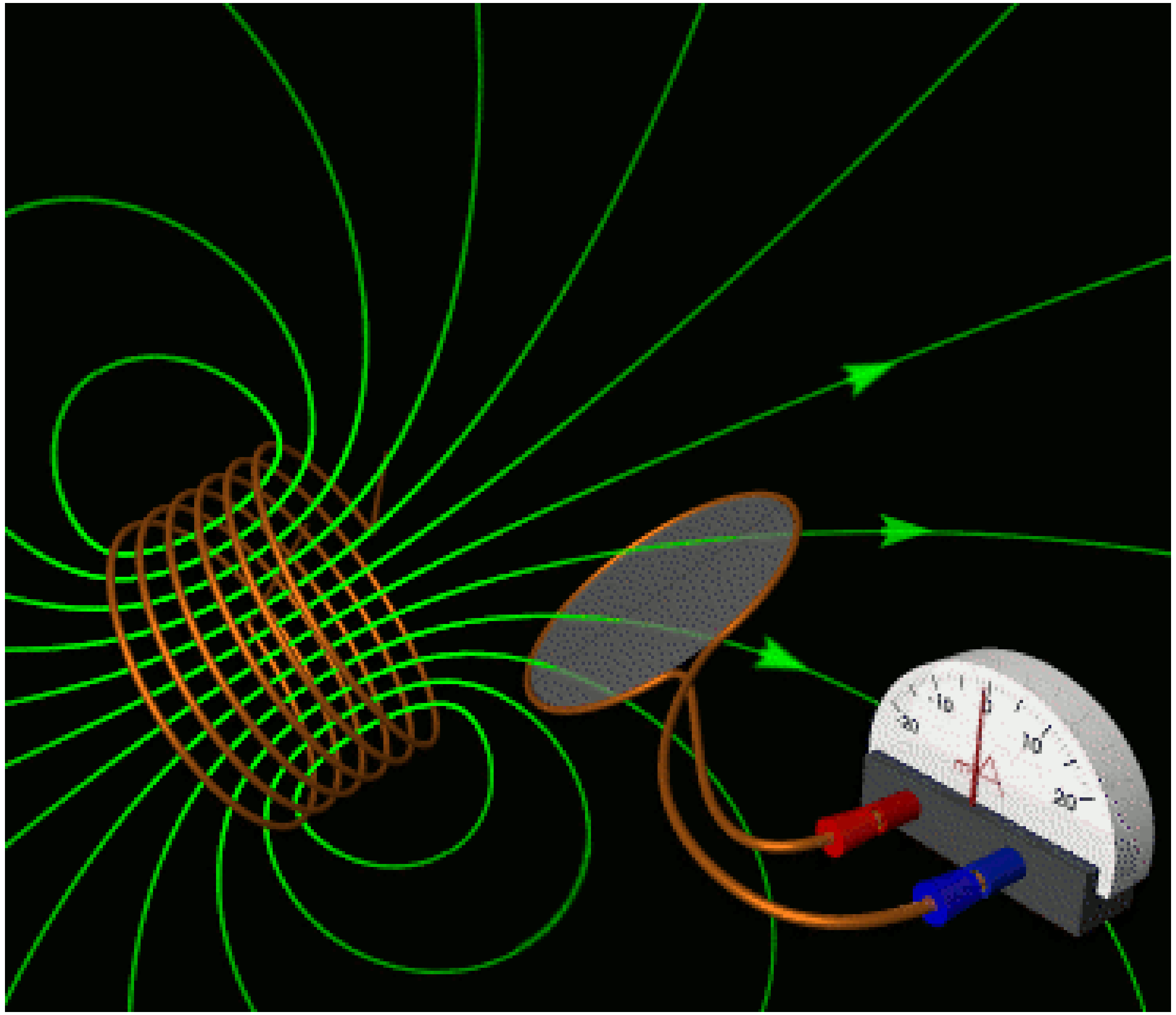
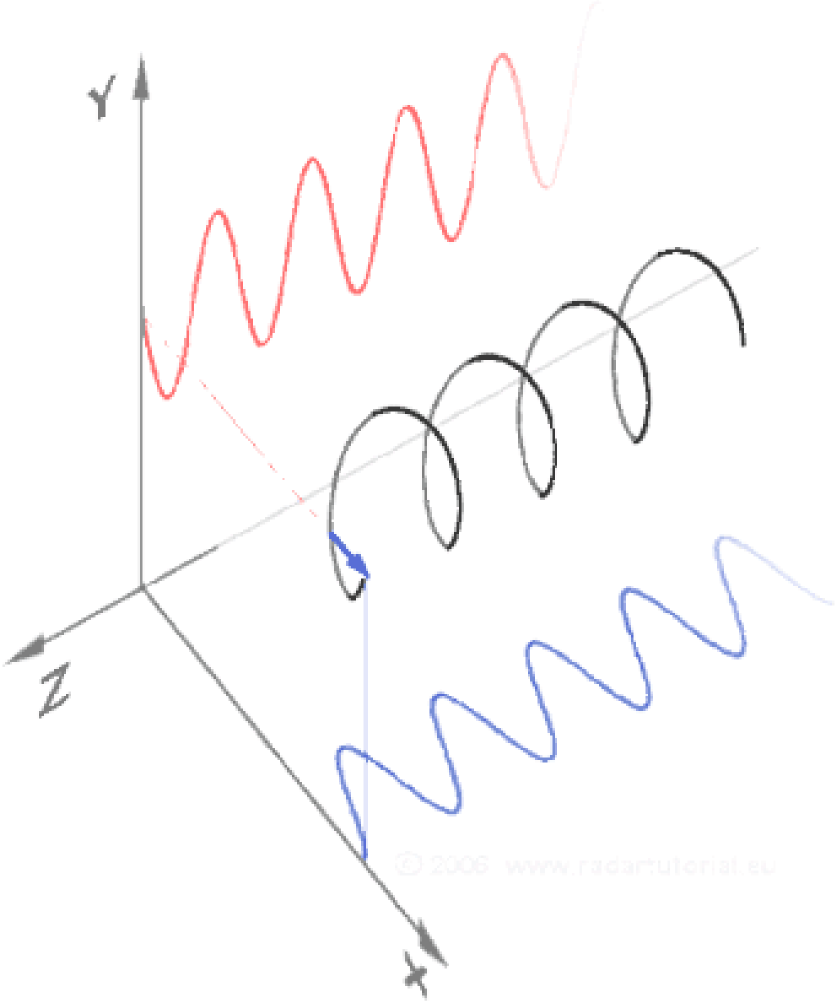


Image Source: http://m.vectormagnets.com/n1854547/Permanent-magnet-synchronous-motor.htm

Image Source: https://www.lumsyn.com/products/pmsm-vs-bldc

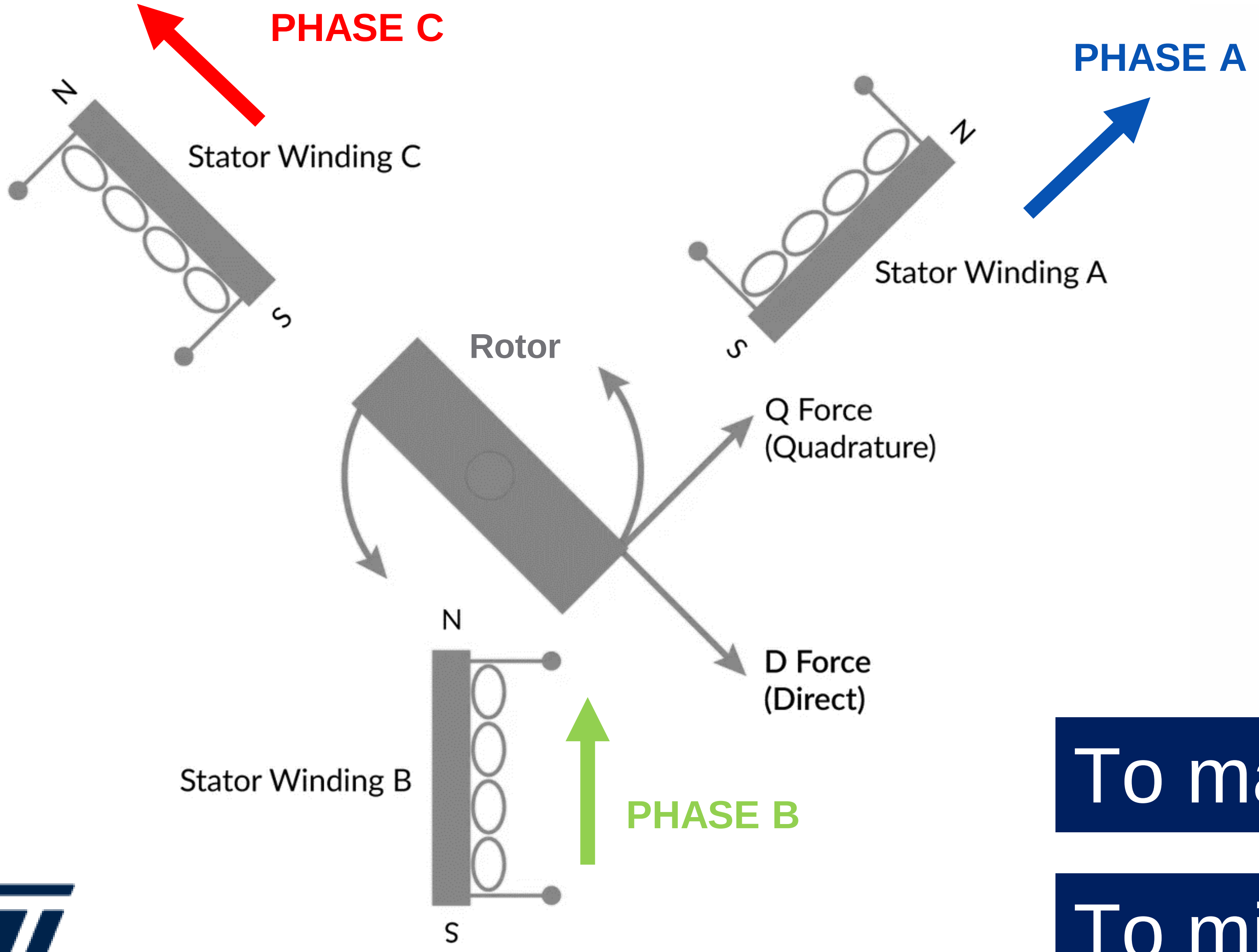
Time Varying Magnetic Field

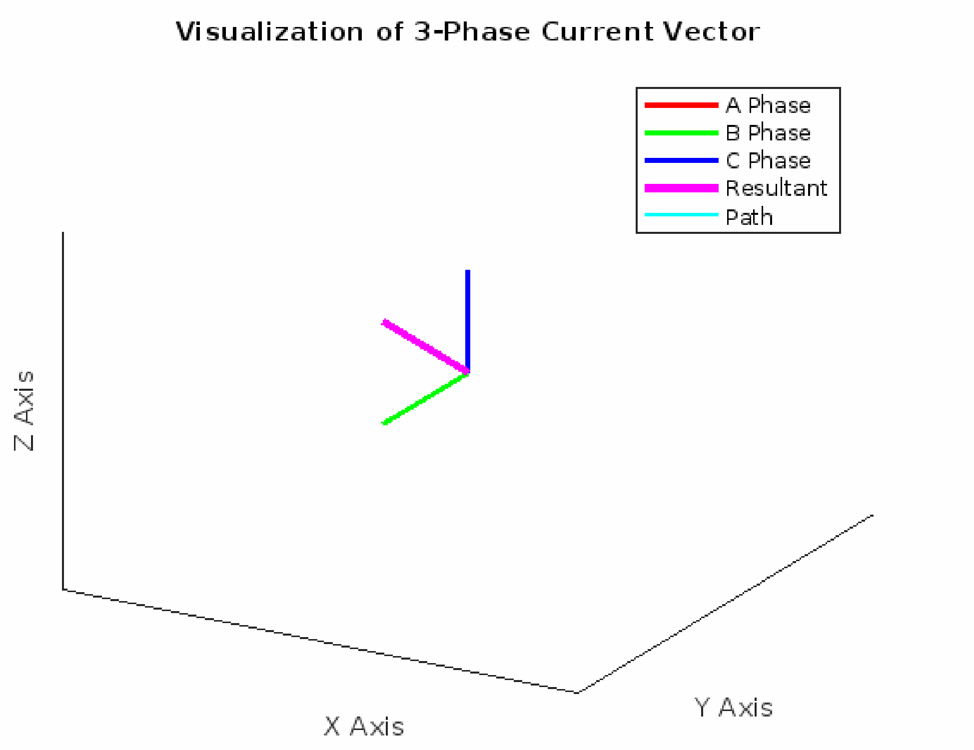




life.augmented

Mission is to generate rotations





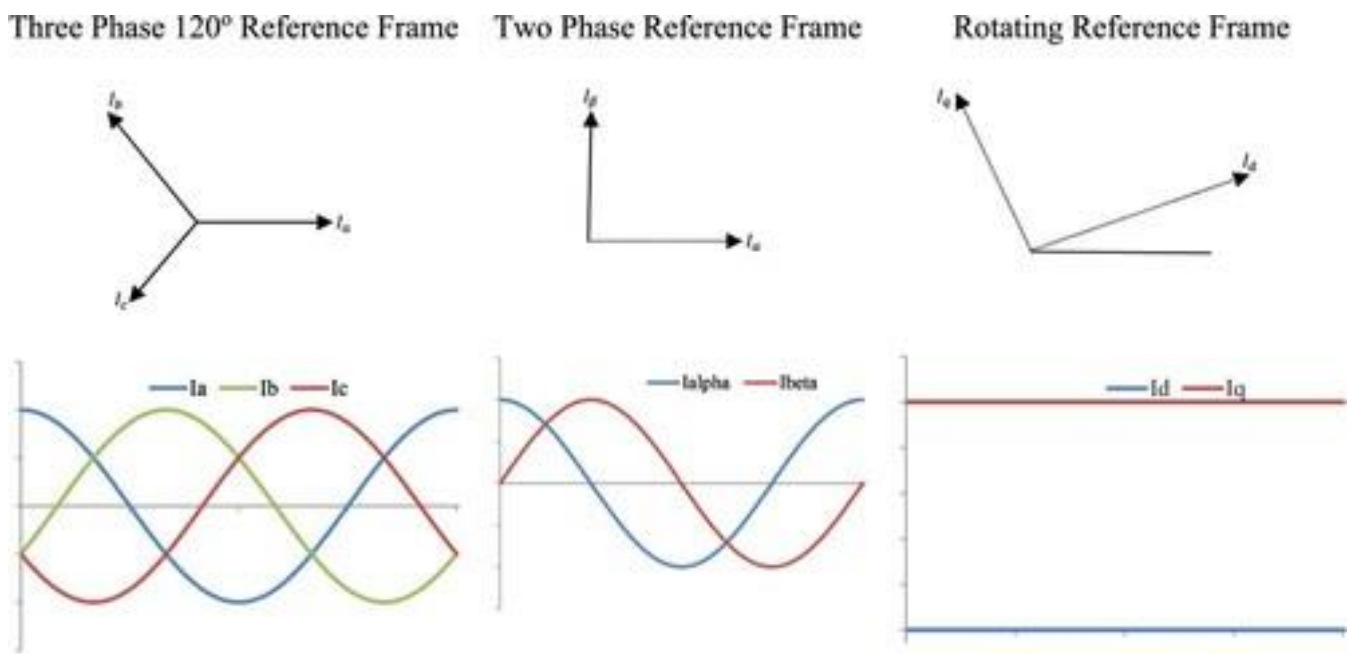
To maximize Q Force

To minimize D Force

life.augmented

Field-Oriented Control

- In **FOC,** simplified, PID control based is required.
- To achieve that, Voltage and Current signals shall be no longer sinusoidal but direct so that the control loop occurs de-referenced from the 3D vector's rotation.
- This happens through the Clark and Park (and their inverse) transforms.
- The **time varying three-phase system** in rotor's ABC reference frame is transformed to **time invariant D Q** components.



life.augmented

FOC for PMSM

Higher top speed

High energy efficiency

>> 97%to 99.5% <<

Essential for BEVs

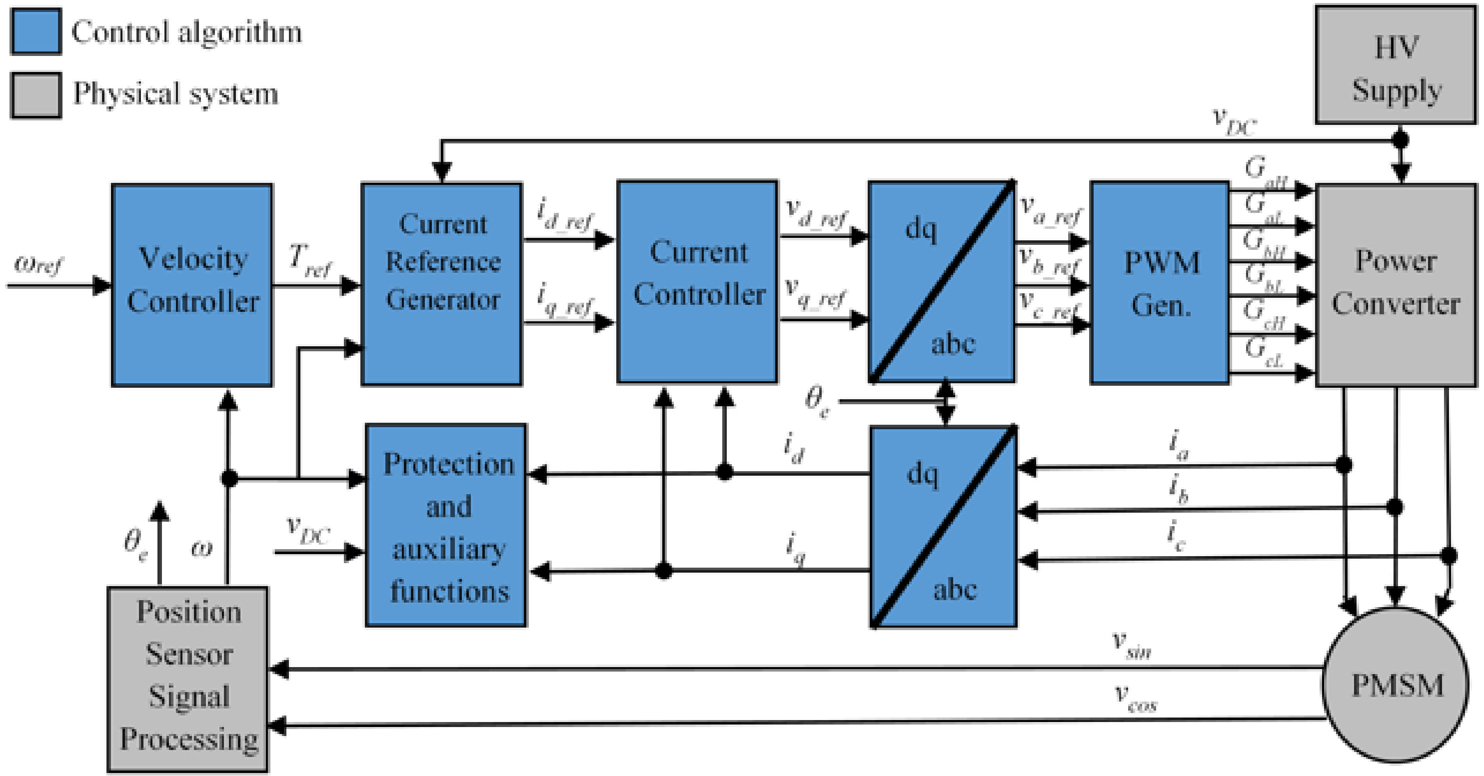
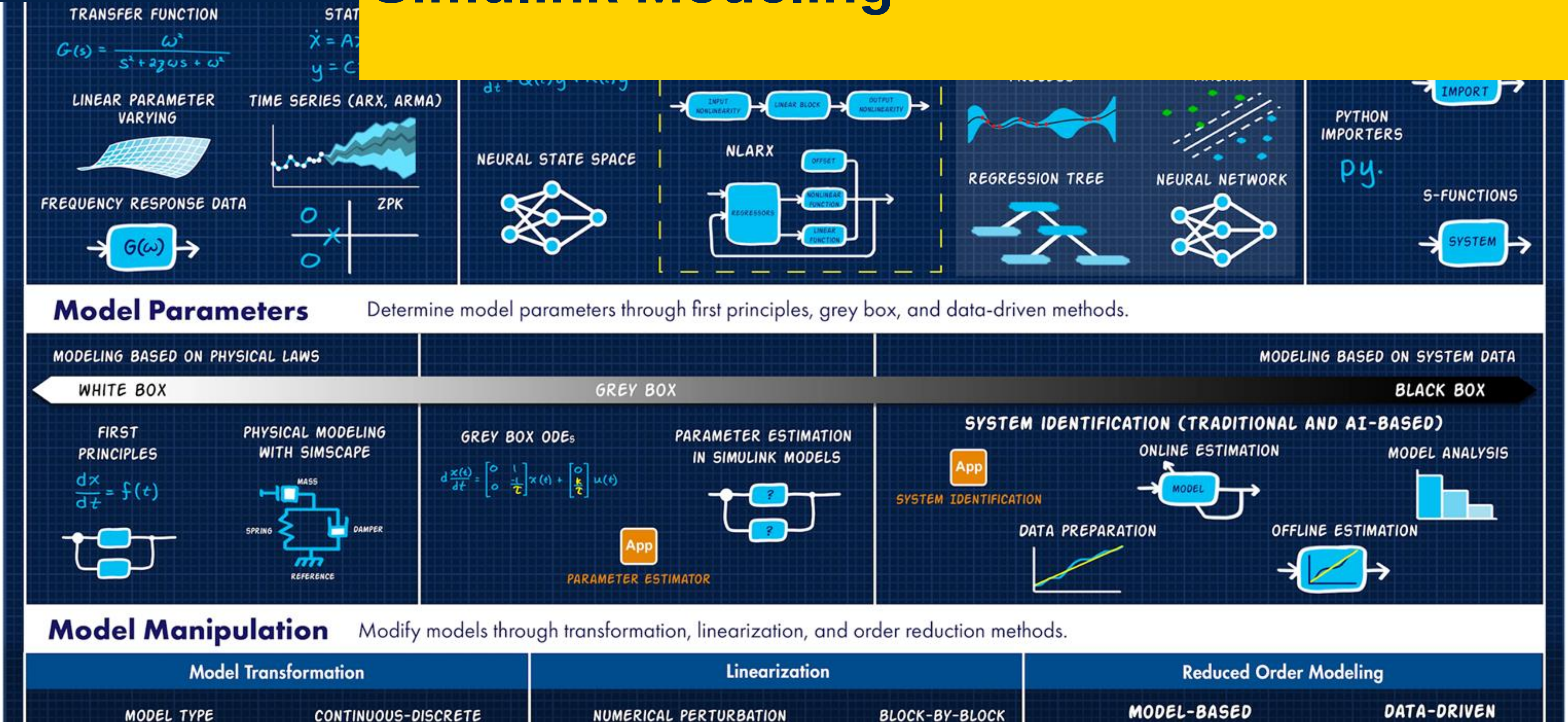


Image Source: https://it.mathworks.com/help/sps/ref/pmsmfieldorientedcontrol.html

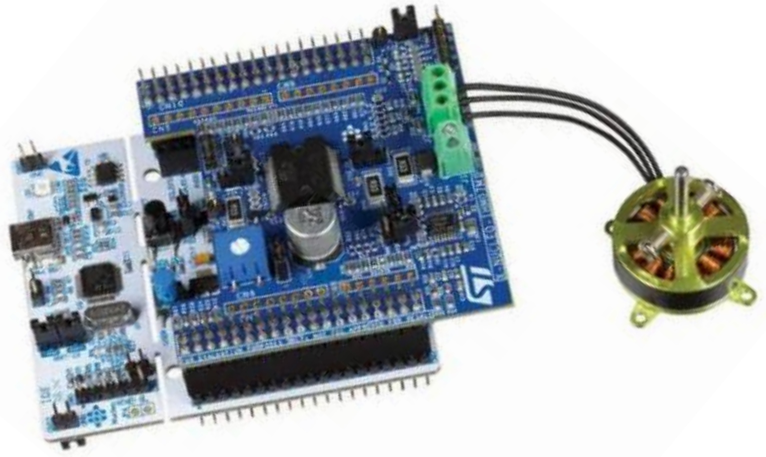
life.augmented

Simulink Modeling



Exemplary Motor

- The **BR2804-1700KV** motor operates at a nominal voltage of 11.1V, within the **X-NUCLEO-IHM07M1**'s 8-48V range.
- Additionally, the motor's maximum current of 5A aligns closely with the board's 2.8A output peak current per phase, making it a **safe and effective for educational purposes.**
- The motor's 7 pole pairs are well-suited for FOC, which is efficiently handled by the **X-NUCLEO-IHM07M1** board, ensuring high torque and smooth operation, crucial for precision control applications.
- This motor was used to parametrize the Simulink model

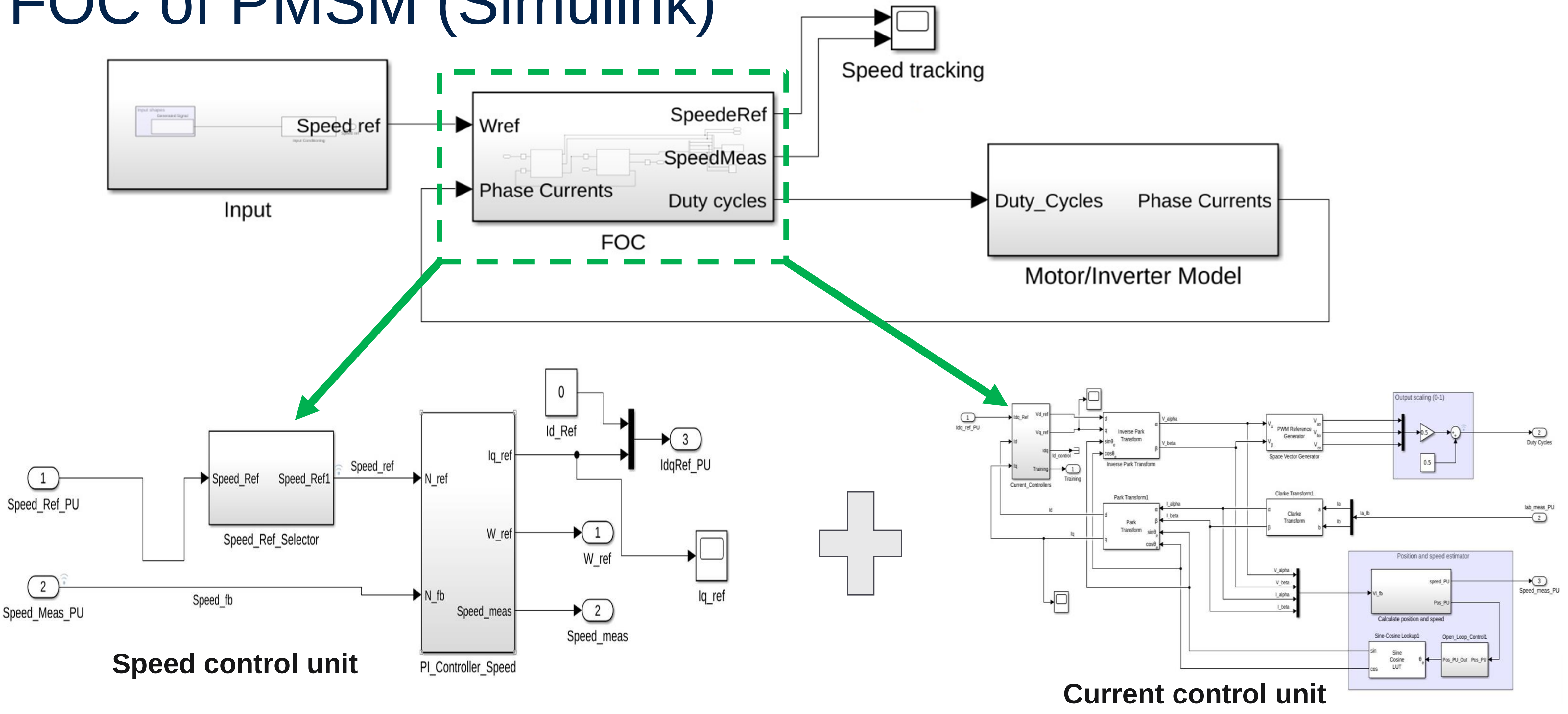


https://www.st.com/en/evaluation-tools/p-nucleo-ihm001.html

Model: Bull-Running model BR2804-1700 kV
Nominal voltage: 11.1 V DC (battery up to 3S)
Maximum DC current: 5 A
Poles: 7 pole pairs
Max speed: 19,000 RPM

Field oriented control dataset of a 3-phase permanent magnet synchronous motor
Nustes J.C. , Pau D.P., Gruosso G.
Data in Brief, Volume 47, 109002, April 2023

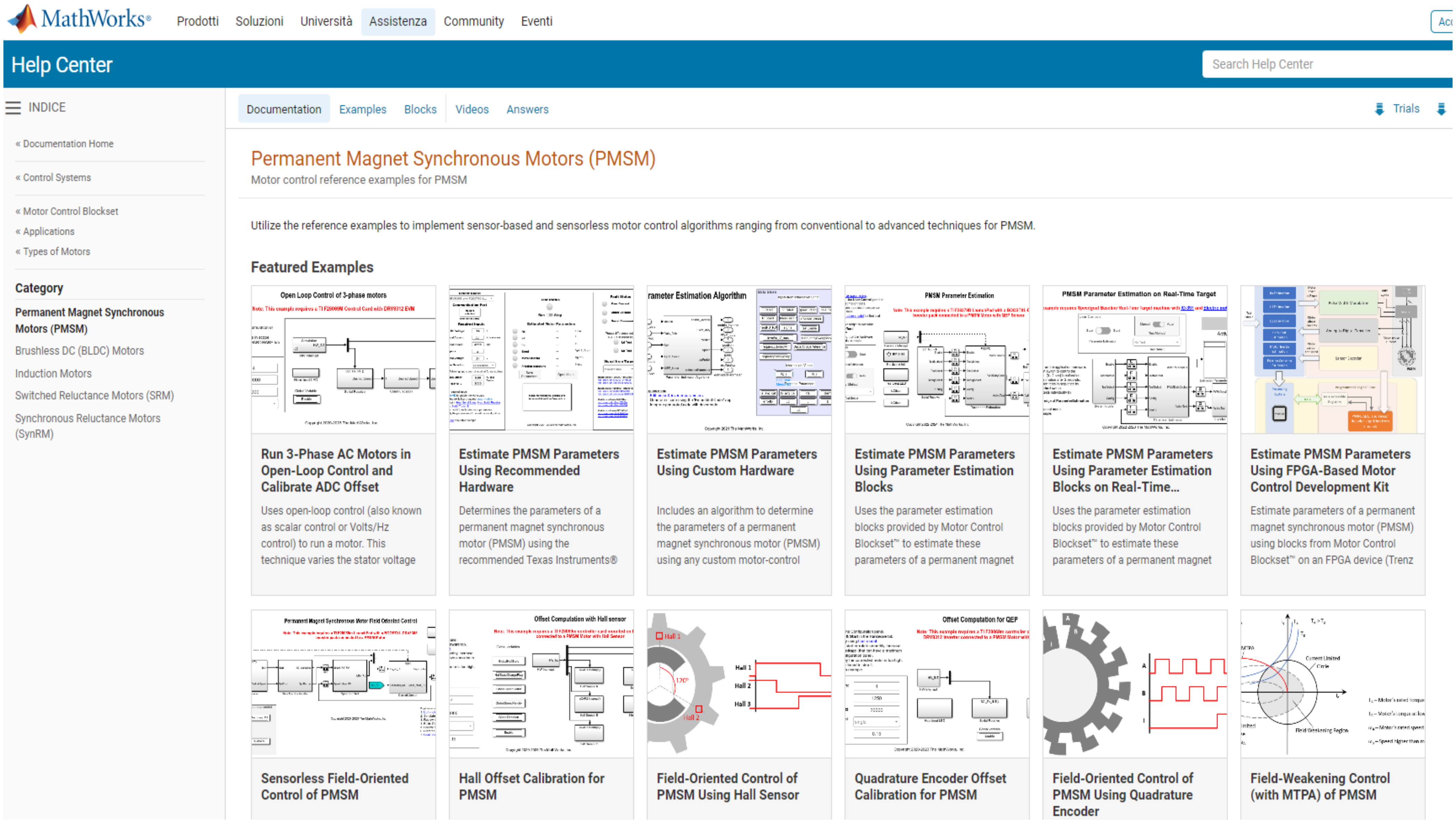
FOC of PMSM (Simulink)



@ https://github.com/heixiaopengyou/TINY-ML-for-FOC-of-PMSM-20092024

life.augmented

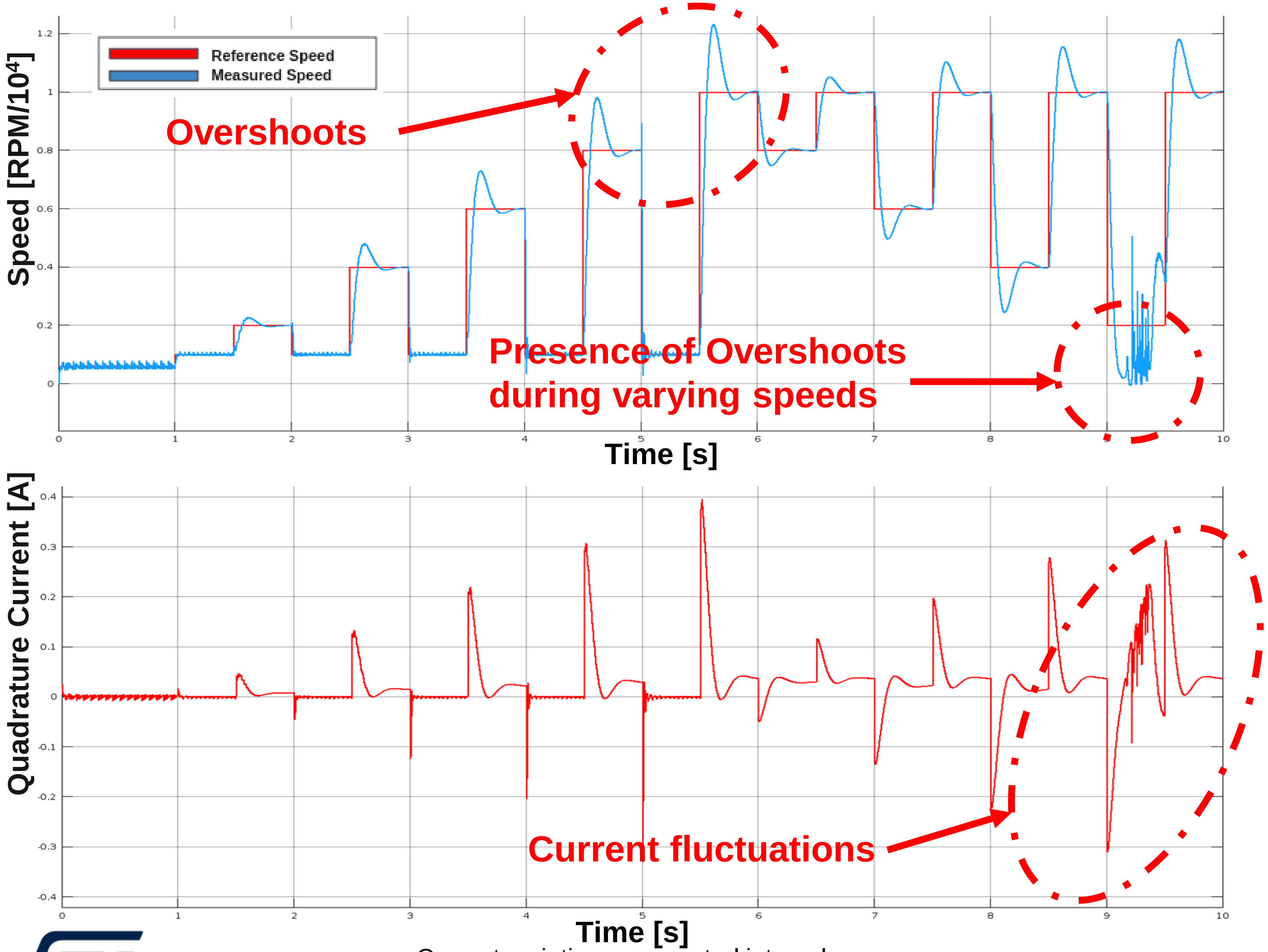
A Wide Set of Resources



life.augmented

Problem definition and Requirements

Case Study 1

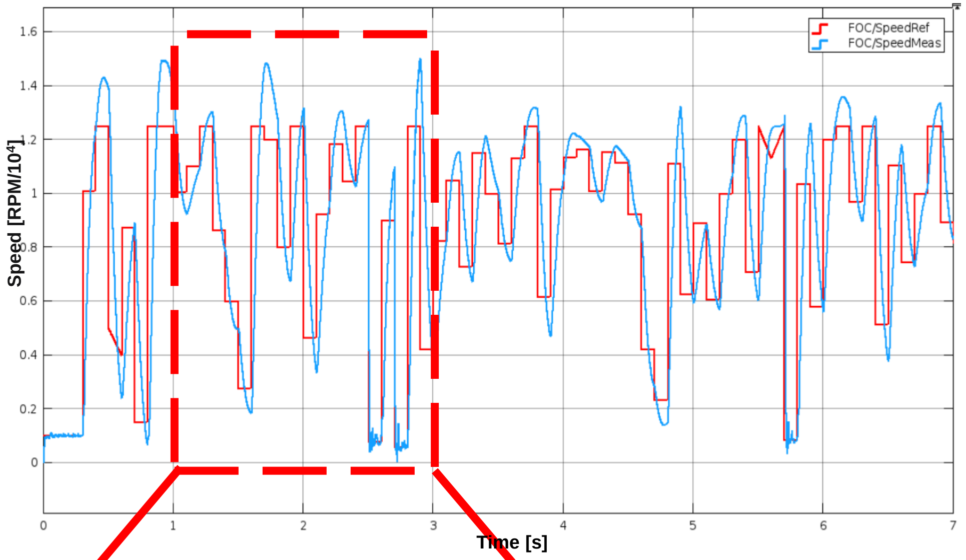


Current variation over control interval

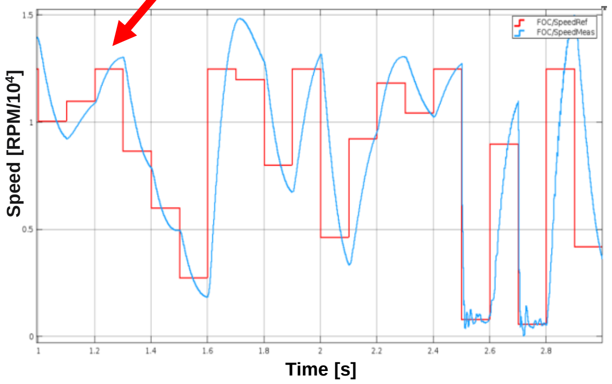
- **Case 1** introduced a speed signal with 2 transitions per second.
- The **PI(D) controller struggled** to quickly adapt to rapid changes in the reference speed leading to **poor dynamic** performances and **sluggish responses**.
- Moreover, it produces **significant (0.81) deviation** and **longer settling times**, impacting the precision and stability of motor control.

life.augmented

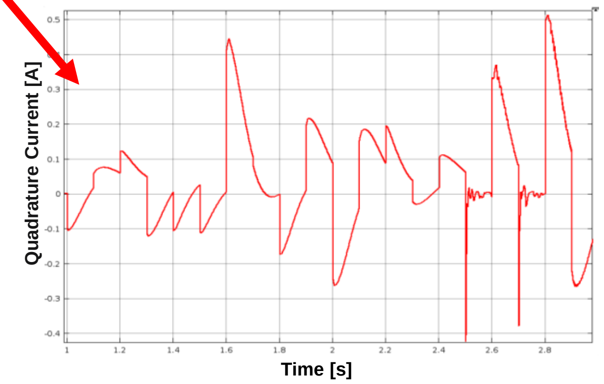
Case Study 2



PI(D) speed control over a 7 seconds interval, with about 10 Hz speed variations



Poor control over each interval

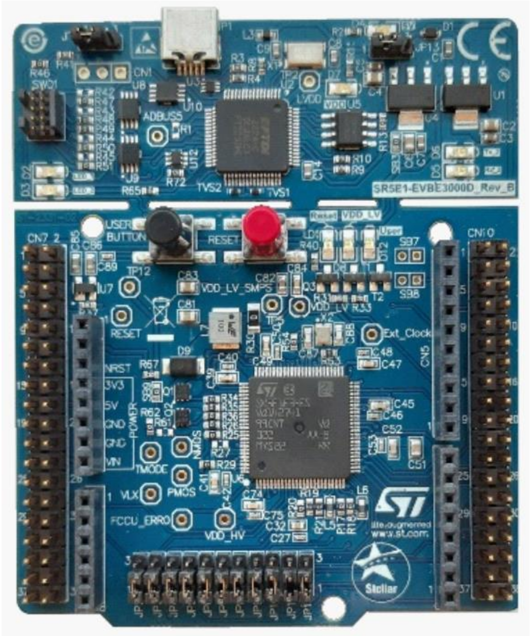


Reference current display over 2 seconds interval

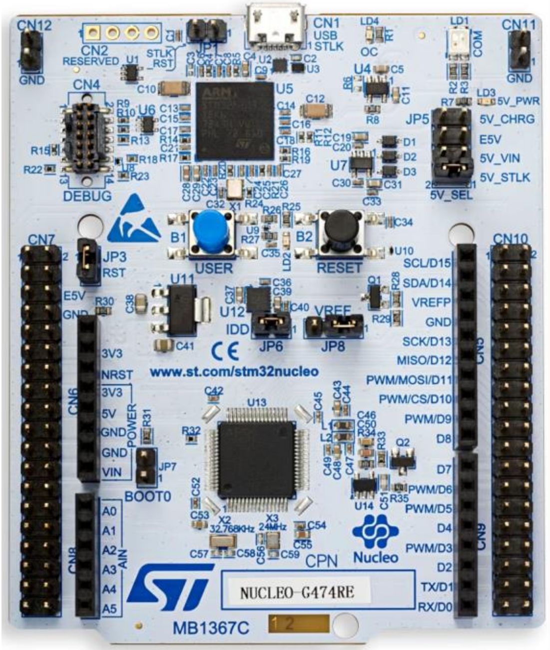
- **Case 2** introduced even more transitions (**10**) in one second.
- At even faster transitions, though the PI(D) controller follows the overall speed trend, it **significantly fails** to stabilize around the desired speed (for each interval).
- This since the calculated reference (quadrature **q**) current generated by the speed PI(D) controller contains deviations (errors) for most of the time steps

life.augmented

Embedded MCU Targets



Board: **SR5R1-EVBE3000D**
Processor Speed: 300 MHz
Internal RAM: 256 KiB
Internal Flash: 1920 KiB

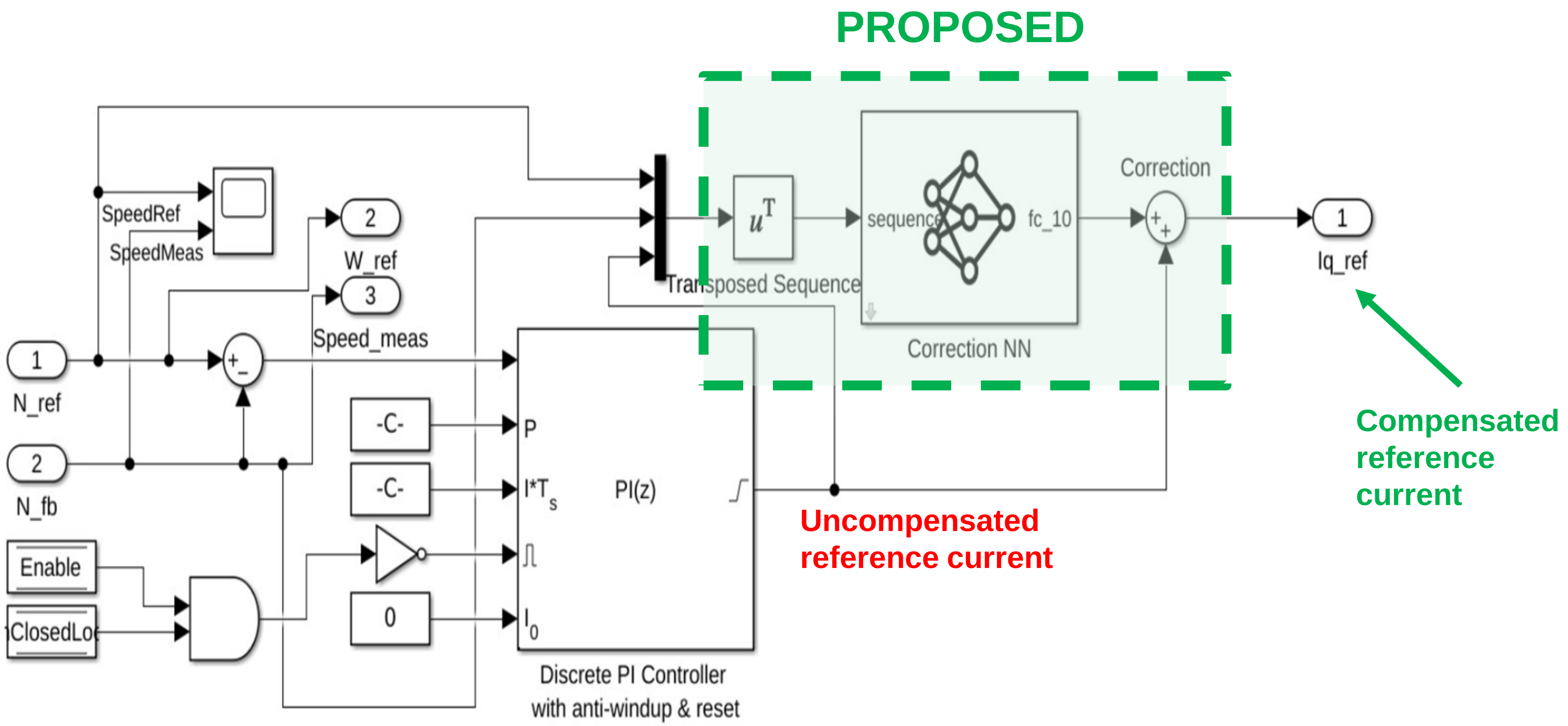


Board: **NUCLEO-G474RE**
Processor Speed: 170 MHz
Internal RAM: 128 KiB
Internal Flash: 512 KiB

- Correcting PI(D) signals requires **extra computations**.
- These approaches shall be deployable on tiny MCUs.
- Two ST MCU boards, automotive (Stellar) and IoT (STM32), have been considered.

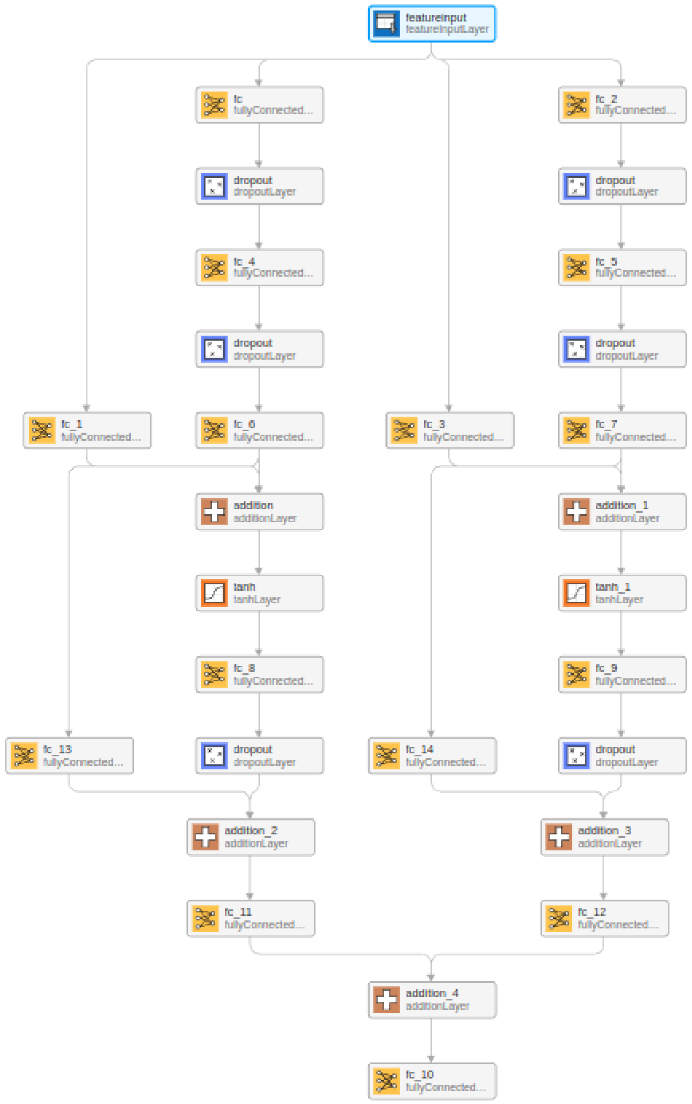
Approach devised

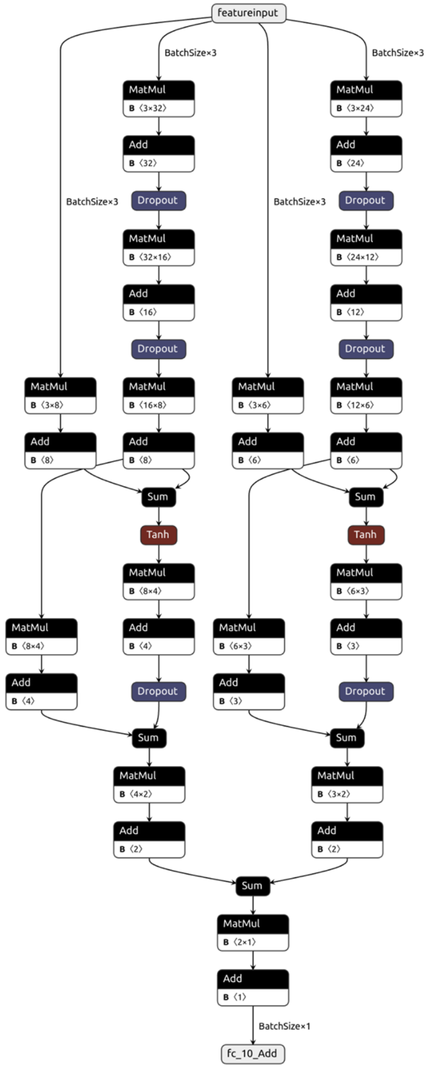
AI Augmented FOC



Speed control set-up, with TinyNN to predict the PI(D)'s deviations

Network Topology





Model visualized by MATLAB R2024a

Model visualized by ST Edge AI Developer Cloud

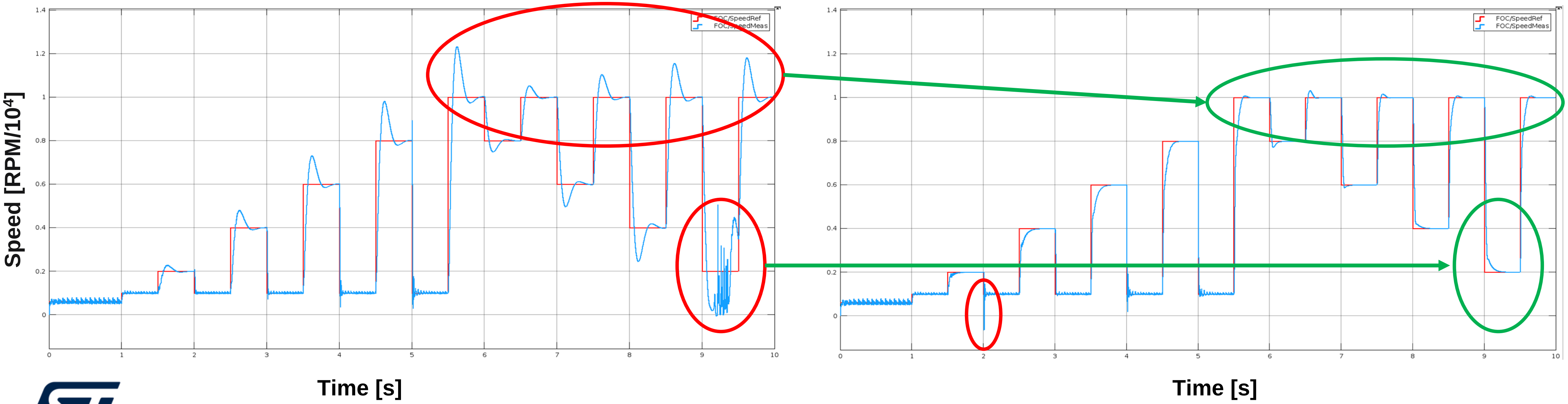
- The proposed model (w/o spatial dilation) was a **1.4 K (weights)** model size, moderately deep with residual connections.
- Ratio training samples and weights = **171.4**

life.augmented

Experimental Results

Case Study 1

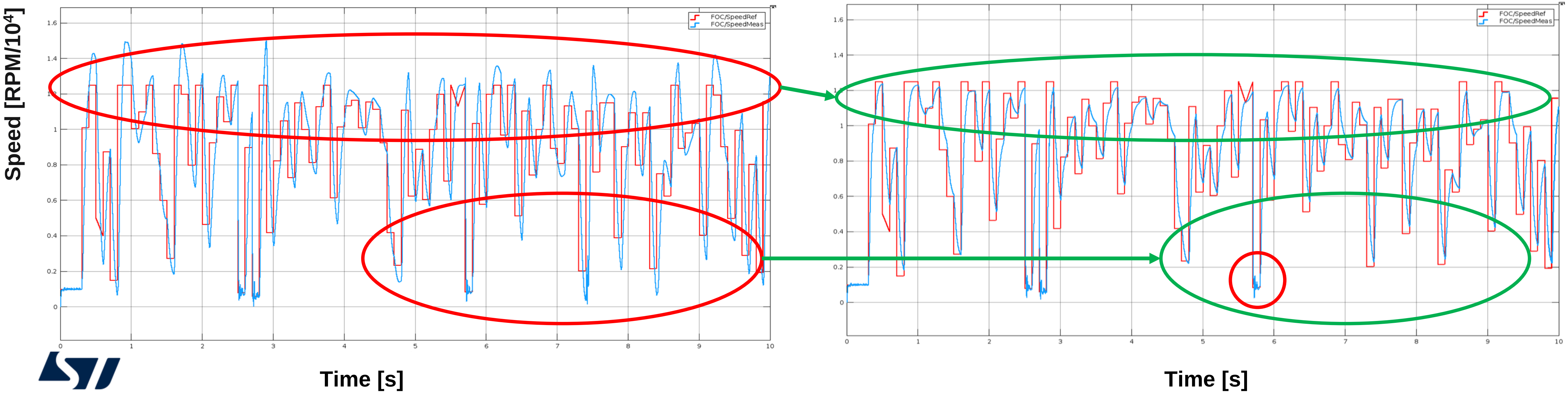
	Max Deviation	Average Deviation	Max overshoot
PI(D)	0.81	0.05	0.24
PI(D) + TinyNN	0.89	0.02	0.03
Percentage Change (%)	+10	- 60	- 87.5



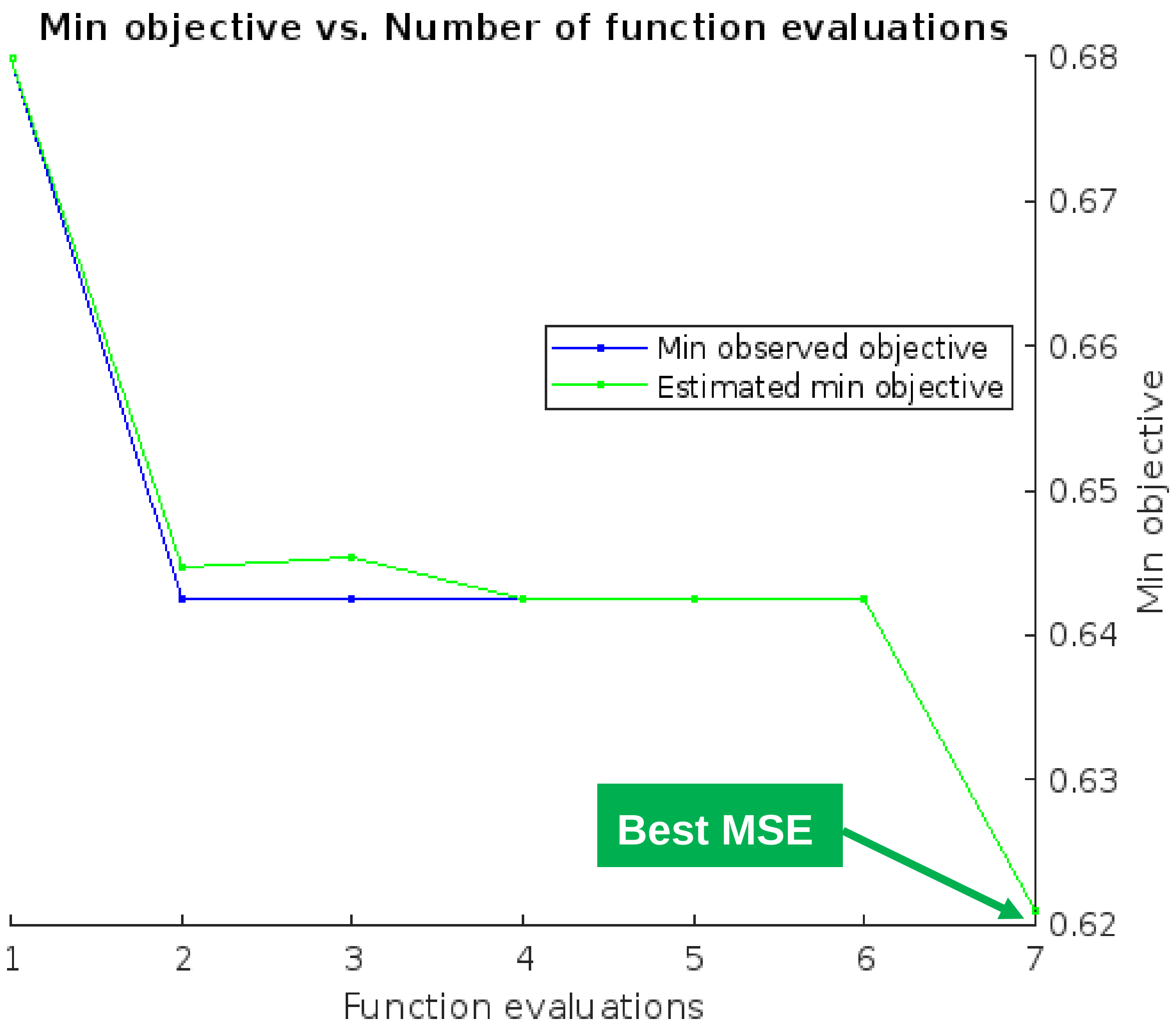
life.augmented

Case Study 2

	Max Deviation	Average Deviation	Max overshoot
PI(D)	1.21	0.18	0.25
PI(D) + TinyNN	1.19	0.15	0.08
Percentage Change (%)	**- 1.65**	**- 16.7**	**- 68**



Hyper Parameters Optimization



- The objective was to further reduce the number of trainable parameters.
- Obtained the best model iterating through several model configurations.
- Each model was trained over the same number of iterations as the initial model, and the model with least MSE was found.

Reference: https://www.mathworks.com/help/stats/bayesopt.html

HPO results

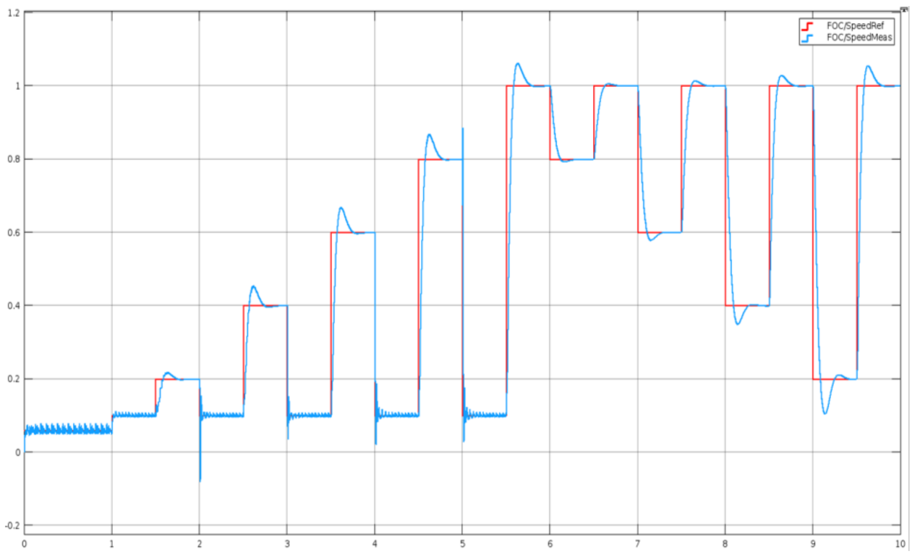
	Trainables	Max Deviation	Average Deviation	Max overshoot
PI(D) + TinyNN	1.4 K	0.89	0.02	0.03
PI(D) + HPO TinyNN	0.67 K	0.896	0.0316	0.06
Percentage Change (%)	**- 52.1**	**+0.67**	**+58**	**+100**

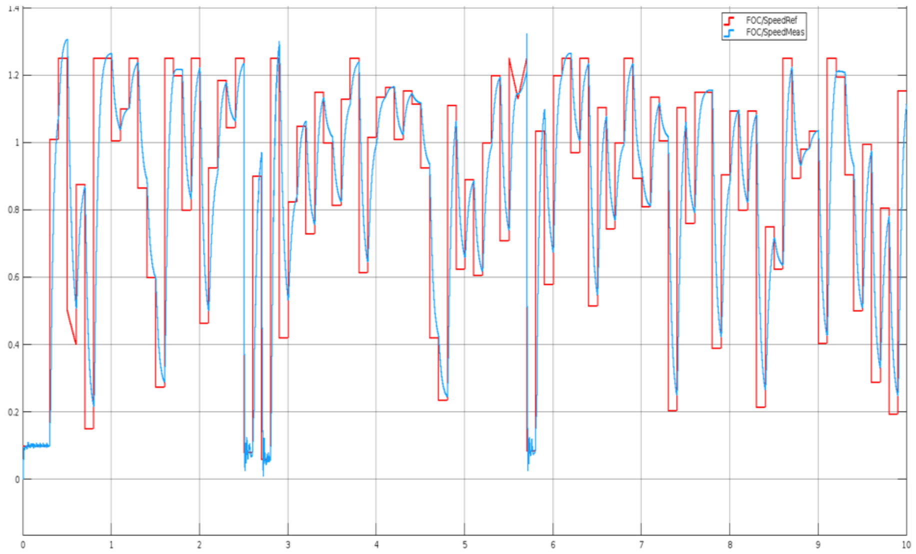
	Trainables	Max Deviation	Average deviation	Max overshoot
PI(D) + TinyNN	1.4 K	1.19	0.15	0.08
PI(D) + HPO TinyNN	0.34 K	1.23	0.15	1.24
Percentage Change (%)	**-75.7**	**+3.36**	**No change**	**+1450**

Case 1

Case 2

Speed [RPM/10^4]





Time [s]

life.augmented

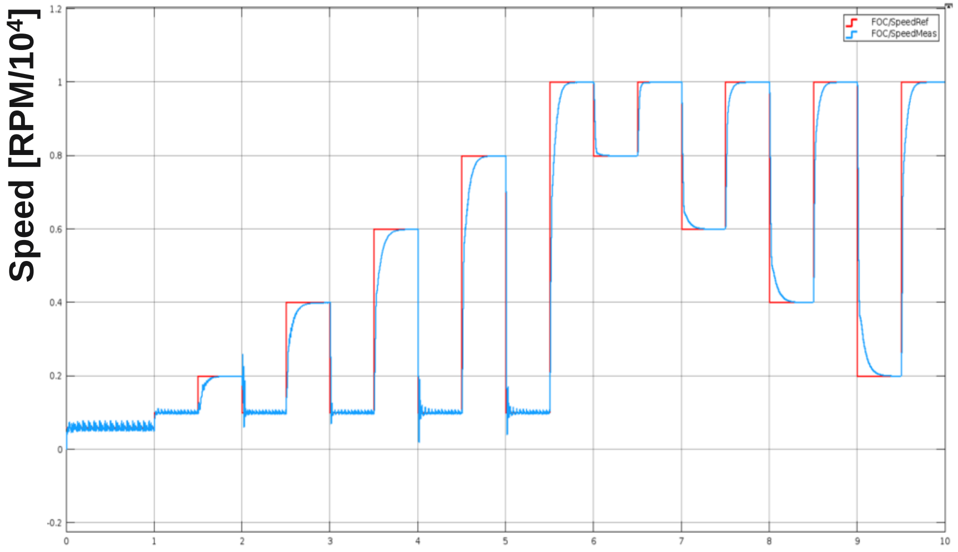
Pruning results

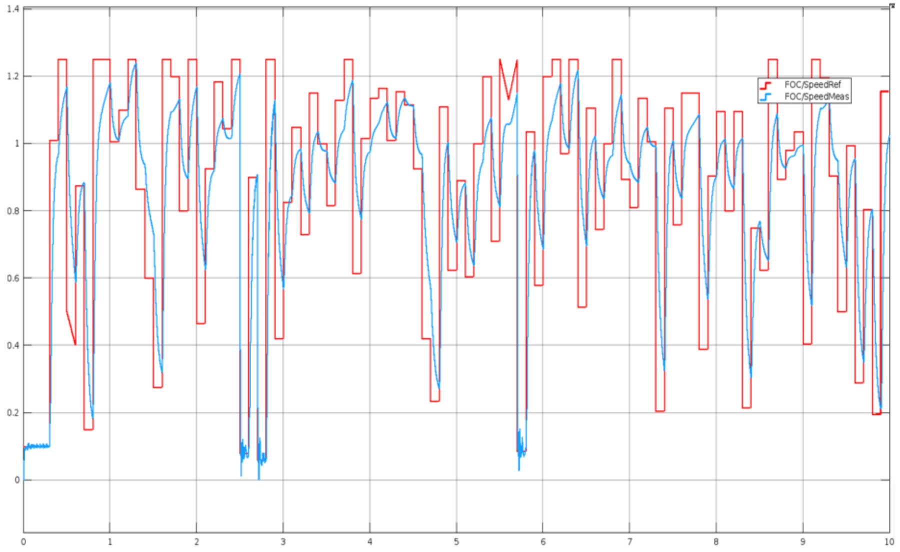
	Trainables	Max Deviation	Average Deviation	Max overshoot
PI(D) + TinyNN	1.4 K	0.89	0.02	0.03
PI(D) + Pruned TinyNN	0.873 K	0.9	0.02	~ 0.0
Percentage Change (%)	**-37.64**	**+ 1.12**	**No change**	**-100**

Case 1

	Trainables	Max Deviation	Average deviation	Max overshoot
PI(D) + TinyNN	1.4 K	1.19	0.15	0.08
PI(D) + Pruned TinyNN	0.6 K	1.20	0.16	0.03
Percentage Change (%)	**-57.14**	**+0.84**	**+6.67**	**-62.5**

Case 2



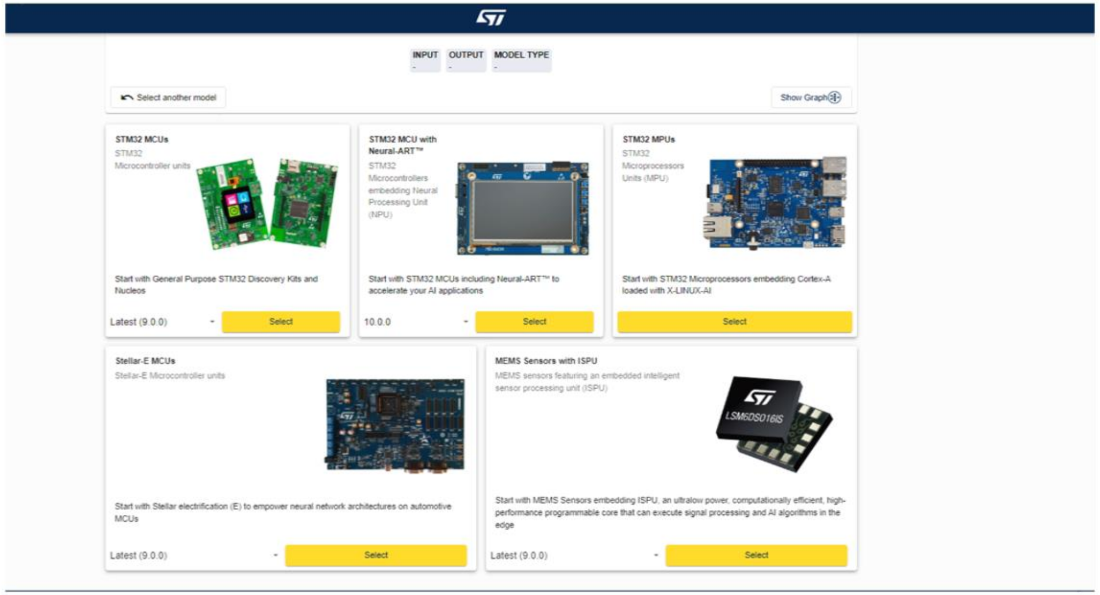


Time [s]

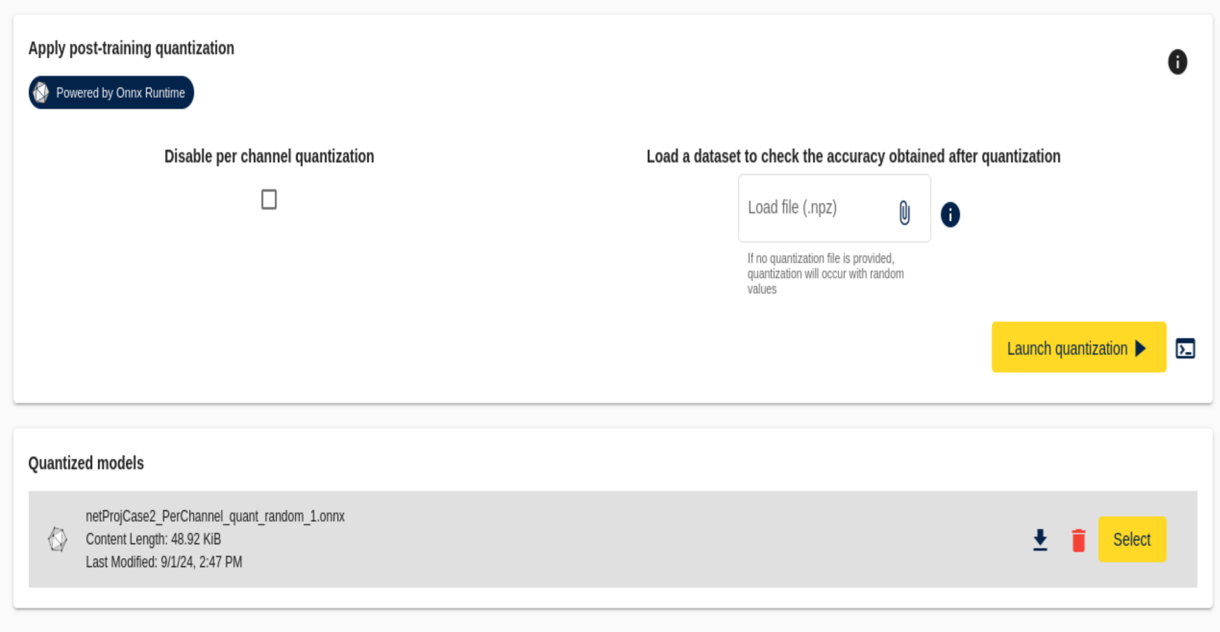
Deployability on MCU

ST Edge AI

Developer Cloud



8bits Post Training Quantization (PTQ)



https://stm32ai-cs.st.com/home

life.augmented

Case 1

	Number of Parameters	MACC	FLASH (KiB)	RAM (KiB)	Execution Time (us)
Fp32 Tiny NN	1400	1620	Weights: 5.68 Library STM32: 15 Library Stellar-E: 16	Activations: 0.199 **Library: 6**	NUCLEO-G474RE 207.4 SR5E1-EVBE3000D 92.8
HPO Model	**670**	764	Weights: 2.61 **Library STM32: 15** **Library Stellar-E: 16**	Activations: 0.152 **Library: 6**	**NUCLEO-G474RE 144.8** **SR5E1-EVBE3000D 69.4**
Pruned Model	873	1034	Weights: 3.28 Library STM32: 20 Library Stellar-E: 20	**Activations: 0.148** Library: 10	NUCLEO-G474RE 232.2 SR5E1-EVBE3000D 102.4
8 bits Quantized Pruned NN on ST Edge AI Dev Cloud	873	910	**Weights: 1.37** Library STM32: 32 Library Stellar-E: 30	Activations: 0.383 Library: 13	NUCLEO-G474RE 371.7 SR5E1-EVBE3000D 167.9

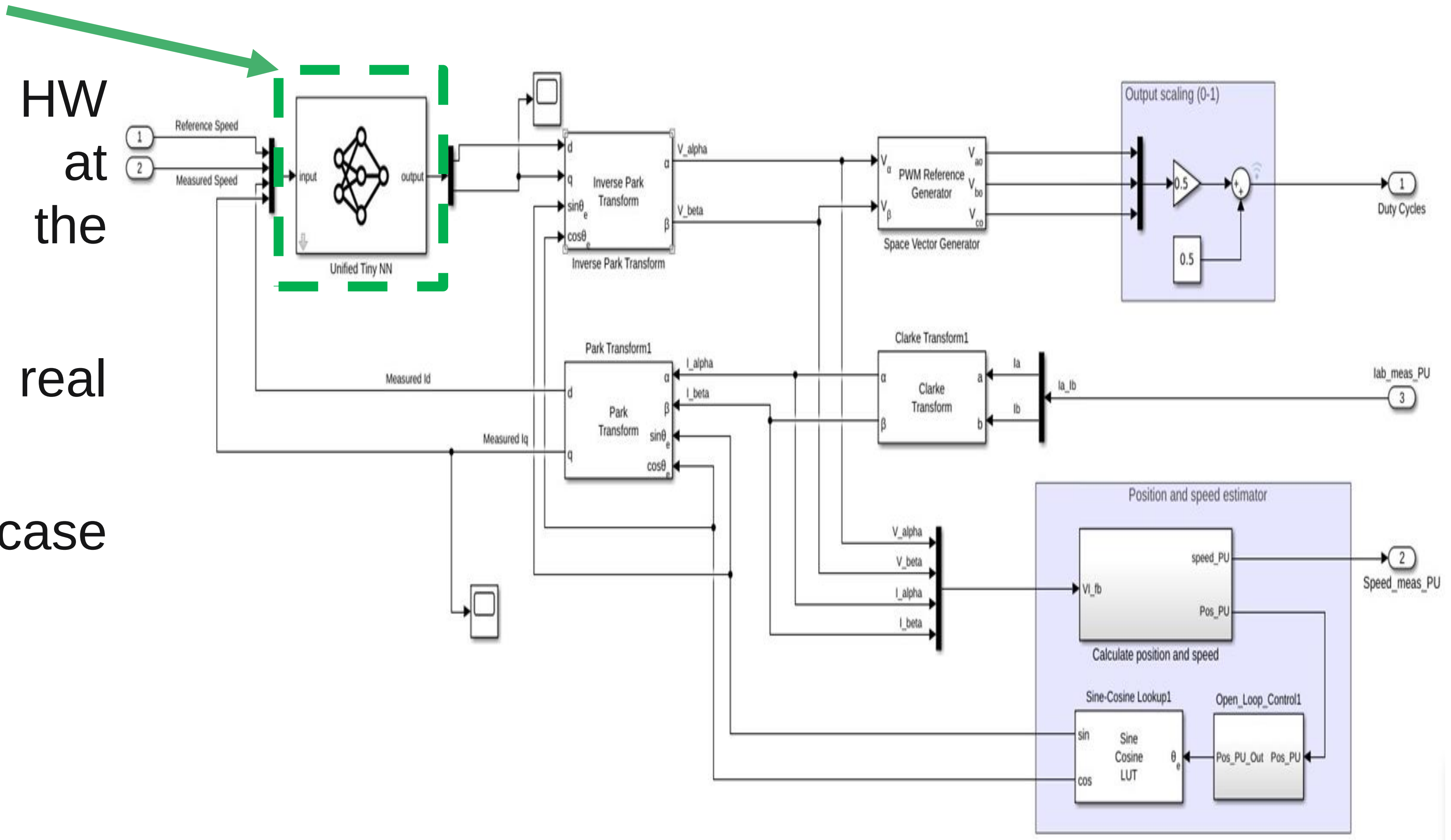
Case 2

	Number of Parameters	**MACC**	**FLASH** (KiB)	**RAM** (KiB)	**Execution Time** (us)
Fp32 Tiny NN	1400	1620	Weights: 5.68 Library STM32: 15 Library Stellar-E: 16	Activations: 0.199 Library: 6	NUCLEO-G474RE 207.4 SR5E1-EVBE3000D 92.8
HPO Model	**340**	**470**	Weights: 1.33 **Library STM32: 15** **Library Stellar-E: 16**	**Activations: 0.105** **Library: 6**	**NUCLEO-G474RE 127.6** **SR5E1-EVBE3000D 61.2**
Pruned Model	600	760	Weights: 2.26 Library STM32: 20 Library Stellar-E: 20	Activations: 0.145 Library: 10	NUCLEO-G474RE 215.2 SR5E1-EVBE3000D 96.20
8 bits Quantized Pruned NN on ST Edge AI Dev Cloud	600	640	**Weights: 1.07** Library STM32: 32 Library Stellar-E: 32	Activations: 0.360 Library: 13	NUCLEO-G474RE 361.4 SR5E1-EVBE3000D 112.1

Future Work

Future Works

- Extend AI approach.
- Study TinyNN HW acceleration. E.g. at 100KHz to close the control loop in 10µs.
- Tests on the field with real PMSM motors.
- Introduce new case studies.
- Explore quantization.



life.augmented

MATLAB EXPO

Thank you

danilo.pau@st.com

MathWorks


MathWorks
@MathWorks
Share the EXPO experience
#MATLABEXPO
MATLAB
EXPO