Electrical Vehicle Fault Tolerant Motor Control

E. Santiago[1], B.Vignasse[1], A. Ibtissem Derbal[1], V-L. Buituan[2], Y. Baakili[3]

1: NXP Semiconductors, 134 Av. Du Général Eisenhower, 31100 Toulouse
2: SERMA Ingénierie, 35 Av. Jean François Champollion, 31100 Toulouse
3: IT-Link, 59 allées Jean Jaurès, 31000 Toulouse

Abstract: The paper describes how NXP Semiconductors uses modelling and simulation tools to develop advanced algorithms for fault tolerant safety concept of EV Traction Inverter. After a brief introduction on the traction inverter safety concept, the paper focus on observers as a digital twin solution for real time sensor estimation used to improve sensors fault tolerances and maintain e.motor torque propulsion. The end of this paper demonstrates efficiency of this solution for detection, isolation and reconfiguration of the control loop after fault injection a in real embedded system environment to achieve fault tolerance of the traction inverter application for the next generation autonomous driving vehicles.

***Keywords*:** *Electrical Vehicles traction inverter, Fault tolerant concept, Sensor-less, Observer*.

1. EV traction inverter safety concept

1.1. *Safety context*

EV traction inverter is a subsystem of EV Powertrain system that controls the motor propulsion of an electric vehicle and provide torque requested by the Vehicle Control Unit on the axles or on wheels.

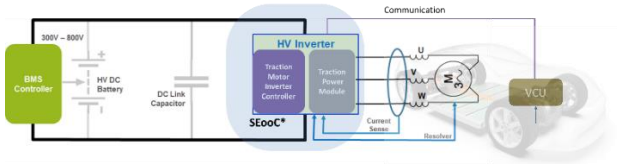

Figure 1 : *EV traction inverter Safety context.*

Today these systems are fail safe: if a critical failure occurs during driving situation the motor control (thus the vehicle) will stop. For future autonomous driving vehicle, the need of availability is increasing for all systems related to the propulsion to avoid stopping the vehicle in a non-safe location. Traction Inverter systems will then have to provide fault tolerant solutions to maintain the propulsion even in case of critical failure. Those fault tolerant solutions become then part of the safety concept.

1.2.*Safety goals*

For fail safe EV traction inverter systems two main safety goals are identified:

SAFETY GOALS		ASIL	Reaction	FTTI
SG1	Avoid unintended acceleration	D	0 torque applied to the motor	200ms
SG2	Avoid unintended deceleration	D	0 torque applied to the motor	200ms

Table 1: *EV traction inverter Safety goals.*

For fault tolerant safety concept, a third safety goal is added to avoid loss of propulsion and the safety reaction in case of failure is not anymore “Apply zero torque” but “Provide degraded mode”:

SAFETY GOALS		ASIL	Reaction	FTTI
SG1	Avoid unintended acceleration	D	Degraded mode	200ms
SG2	Avoid unintended deceleration	D	Degraded mode	200ms
SG3	Avoid shutdown of the motor control	B	Degraded mode	200ms

Table 2 : *Inverter Fault Tolerant Safety goals.*

Here the degraded mode is to notify that a first fault has been triggered and reconfigure the system or control loop to continue providing torque to the motor.

Depending on the failure criticality if the degraded mode cannot be achieved or a second fault occurs, a safe state is still required following traditional safety concept and safe state requirements.

1.3. *Safe state definition*

In the Safe state, the electric motor shall achieve zero torque output. Other than the normal operation by regulating zero torque by PWM technique, there are other means that can achieve near zero torque without PWM. Additionally, the back electromotive force (EMF) level shall be considered to ensure the system reaches a Safe state, preventing damage to the HV battery from excessively high back EMF voltage, which is the voltage induced in the motor windings when the motor is in motion. Based on these targets, there are four safe states which can be allocated to three safe states categories.

- *Active short-circuit of the three motor phases*: The active short circuit of the three-phase motor uses at high-speed operation. It permits to avoid back EMF generation to High Voltage DC bus to not damage the HV battery and avoid creating high braking torque which may create vehicle stability hazard not controllable by the driver.

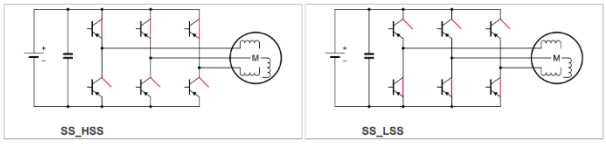

Figure 2 : *Safe State: Active short-circuit of the 3 motor phases.*

• *Open-circuit of the three motor phases*: This mode allows and controlled by the EV traction inverter only if the back EMF generated between the motor terminals is less than the HV DC bus voltage. This corresponds to low motor speed operation.

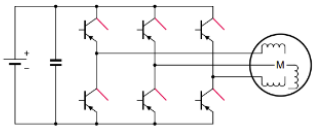

Figure 3 : *Safe State: 3 Phase Open circuit.*

• *Zero torque control*: This mode allows when a non-critical failure in the current regulation loop still permits a nominal control of the inverter. It is applied in case of communication error between VCU and EV traction inverter which led to a non-reliable torque request to the EV traction inverter.

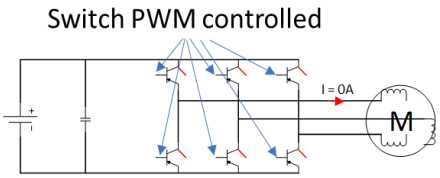

Figure 4 : *Safe State: Zero torque control.*

1.4. *Safety architecture*

The EV traction inverter functional architecture can be divided into five elements:

1-communicate: The inverter shall receive the commands from VCU and provide feedback about the system status to VCU.

2-process: The inverter shall analyze the command from VCU and translate the torque request into a current request.

3-actuate: The inverter shall regulate the current flowing into the electric motor by switching high voltage to respect to the current request.

4-sense: The Inverter shall measure the state of the Motor (Phase current, Position and Temperature).

5-power: The inverter shall provide relevant supply voltages to distinct functions and isolate HV domain from user.

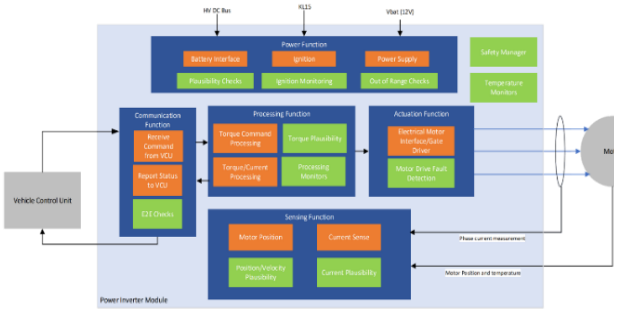

Figure 5 : *EV inverter Functional Safety concept.*

Several safety functions and associated safety mechanisms are defined to be able to detect any faults affecting each of the functions listed above:

• *Safe Communication*: End to End checks on the CAN/Ethernet black channels.

• *Safe Processing*: Doer/checker architecture implemented in a multi core environment with the checkers processed on a lock step core to monitor the Doer functions. MCU HW is also monitored through External watch dog.

• *Safe actuation*: High voltage transistors/gate drivers monitoring, PWM checkers, short circuit protections on the hardware.

• *Safe Sensing,* power: range checks, plausibility checks on current sensing, resolver position sensing.

2. Evaluation of Inverter availability

To increase the fault tolerance of the Inverter it is needed to figure out what are the main technical functions and the higher contributor leading to the loss of availability in case of failure.

The technical architecture can be decomposed in five main functions for the availability analysis.

- Phase current sensing.
- Motor position sensing.
- Control loop processing.
- Power stage inverter.
- Power supply.

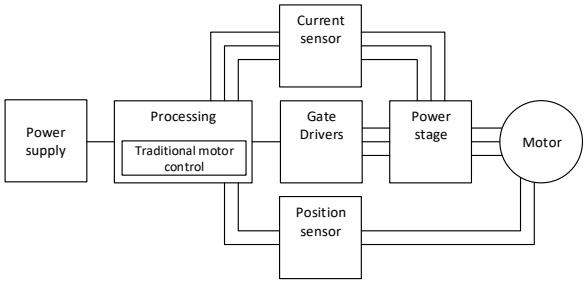

Figure 6 : *EV inverter technical blocks.*

The evaluation and analysis of the failure rate of each technical block of the inverter architecture is done by calculating the failure rate of each component with IEC 62380.

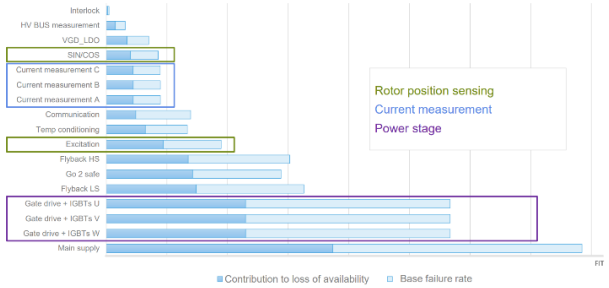

Figure 7 : *Failure rate of inverter.*

Additionally, it is evaluated the ratio of the failure rate which contributes to this loss of availability, directly (the failure leads to loss of function) or indirectly (the failure leads to loss of function due to detection and reaction by safety mechanism and safe state actuation).

As noticed in the results, the main contributors to the loss of availability are:

- *Power stage*
- *Current measurement*
- *Rotor position sensing*

The processor seems not represented in the figure, but it is split in the different blocks.

Therefore, these four blocks are the potential and most relevant targets for innovative solutions to increase the fault tolerance.

One can note the main supply is main contributor, but it is no possibility for fail-operational other than a full redundancy, including the battery input.

3. Fault tolerant concept solution

3.1. *Trend and Solutions*:

To reduce the risk of loss of inverter availability, degraded mode and/or redundant functions shall provide in the system to maintain the torque control loop active when a fault occurs. A straightforward way could be to do HW redundancy, like a full double inverter but this is not realistic for automotive market, due to cost, space and weight impact, as well as sustainability.

Future trends for fault tolerances are more on digital twin solutions, with SW redundancy, smart algorithms, Artificial Intelligences, etc... In automotive embedded systems, thanks to recent powerful MCU integrating several cores, Virtual Sensor and Model Predictive Control are becoming more popular.

3.2. *Fault tolerant challenges*:

Fault tolerance to electronic HW failure pose significant challenges in real-time applications like motor control for electric vehicles. Thus, identifying, and isolating faults in real time environment is necessary and is particularly challenging due to the limited sensing capabilities of embedded systems and the presence of noise, which can make distinguishing between actual faults and transient disturbances difficult. Once a fault is detected, reconfiguring the control strategy promptly is essential as well to avoid violating the loss of function safety goal. However, this is complicated by the need to maintain system stability while adhering to tighter voltage and current constraints that arise under faulty conditions.

Addressing these challenges requires robust fault detection methods, identification and isolation of faulty data, adaptive control strategies, and efficient algorithms that can operate within the constraints of embedded systems.

3.3. *Digital solutions for degraded mode*:

For the inverter fault tolerant concept, the following digital solution could be defined to maintain the torque on the motor while failure is detected on main contributor functions:

- *Power stage failures*: reconfiguration of the control on two legs or reconfiguration to a fourth spare leg.
- *Control loop failures*: SW redundant control loop or MPC running in parallel in another core.
- *Phase current sensing failures*: real-time phase current estimator (SW virtual sensor) using Observers algorithm.
- *Motor position sensing failures*: real-time motor speed and angle estimator (SW virtual sensor) using Observers algorithm.

Following those identified solutions, the focus of our fault tolerant study and development for motor control is addressed by the observer solution for SW redundancy of the sensing functions.

3.4. *Observers for State Estimation of Dynamical PMSM Systems*:

The Luenberger observer is a widely used approach for estimating the rotor position and current in PMSM systems, particularly in sensor-less control applications. This observer used the mathematical model of the motor to estimate unmeasured states based on available inputs and outputs, such as applied voltages and measured three-phase currents. By designing the observer with a specific gain matrix, it minimizes the estimation error, ensuring robust and accurate tracking of rotor dynamics.

The state-space representation of the PMSM system includes the α-β axis currents and rotor position as states, while the observer reconstructs these states by incorporating the motor's voltage equations and measured currents. The error dynamics are stabilized by appropriately tuning the observer gain, ensuring convergence of the estimated states to the actual values. This approach is particularly effective for real-time applications, providing precise state estimation under varying operating conditions.

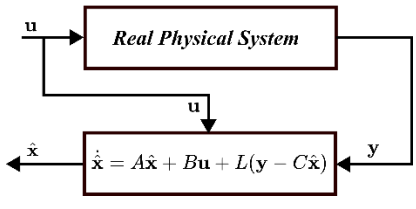

Figure 8 : *Observer principle* – [6].

The Luenberger observer's computational efficiency and compatibility with linear systems make it an

excellent choice for implementing robust sensor-less and fault tolerance control in PMSM applications.

3.5. *Fault tolerant concept for sensing functions*:

Current sensors provide essential feedback for controlling motor torque and speed, while motor position sensors like resolver, deliver accurate rotor angle and velocity information. Faults in these sensors-such as signal drift, noise, hardware failures, or offset errors-can disrupt feedback loops, degrade performance, and potentially lead to system failure.

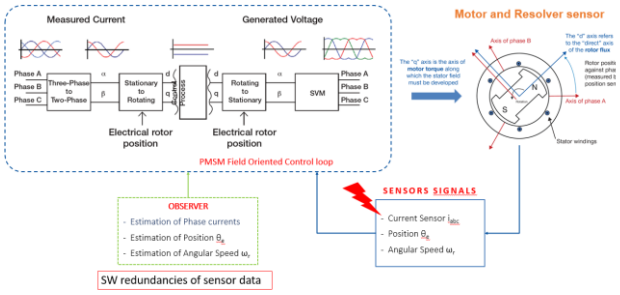

Figure 9 : *Traction inverter control loop.*

The fault tolerance concept using observer design revolves around enhancing the reliability and safety of systems in critical applications. The main challenge lies in achieving robust fault detection under constraints like noise interference, limited processing resources, and the need for rapid response.

For instance, in sensor-less PMSM control, an observer can estimate the rotor position as a fallback when a position sensor fails. The structure diagram of the sensor fault detection system and reconfiguration based on the rotor position estimation system is presented as in the diagram in Figure 10.

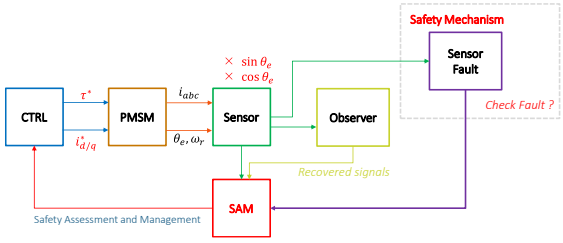

Figure 10 : *Fault Detection and Reconfiguration Concept for Position Sensor Fault.*

This concept ensures that the system maintains performance and safety, even in the presence of faults, making observer-based fault tolerance an integral part of modern control strategies.

Current observers are employed to estimate system states, this model-based design estimates system behavior and compares it with actual sensor outputs to detect inconsistencies. Redundant sensor configurations provide additional reliability by cross-verifying readings. These discrepancies, or residuals, serve as indicators of faults in the system, such as sensor malfunctions, parameter variations, or external disturbances.

The common concept of the three-phase current sensor failure diagnosis is to check the deviation value between the measurement system and the estimation system if the sum of the phase current is different from zero.

But real-time phase balance monitoring can detect a failure in the measurement system but cannot identify the faulty-phase current. Furthermore, sensor fault diagnosis algorithm is not able to detect phase errors in the closed-loop control systems (because the feedback signals from the sensor distort the controller's adjustment value, affecting the remaining phases in the next iterative-loop).

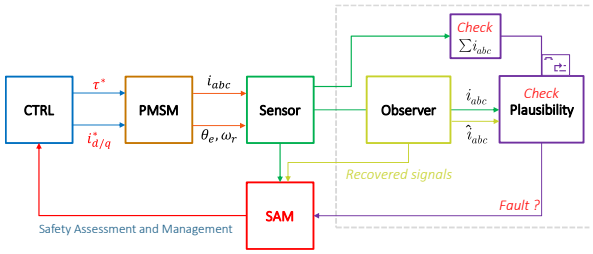

Figure 11 : *Observer-based Fault Detection Designs for Phase-Current Sensor Fault.*

By integrating fault detection and isolation (FDI) advance mechanisms, observer-based designs can identify and localize faults, allowing for corrective actions like switching to backup control modes or reconfiguring the system parameters.

The FDI process involves detecting anomalies in current sensor outputs, the faulty component, and diagnosing the nature of the fault to reconfigure for fault-tolerant operation. Furthermore, incorporating the FDI and fault-tolerant strategies into the motor control system ensures continued operation even in the presence of sensor faults, thus enhancing the system's overall reliability and efficiency.

These designed architectures can operate within the constraints of embedded systems, which are optimized the computational complexities and require real-time processing capabilities. The results of this design strategy will be presented in the simulation results and test bench sections demonstrating the effectiveness of timely fault detection and isolation as well as system reconfiguration.

4. Model-Based Design Approach

4.1. *Model Based Design steps*:

To develop the configurable safety software, we used the MBD approach. This approach is an efficient process to work on a complex system model while maintaining a single model used from the beginning of the concept (numerical simulation) to the proof-of-concept phase (Hardware test bench).

This process is represented by five main steps:

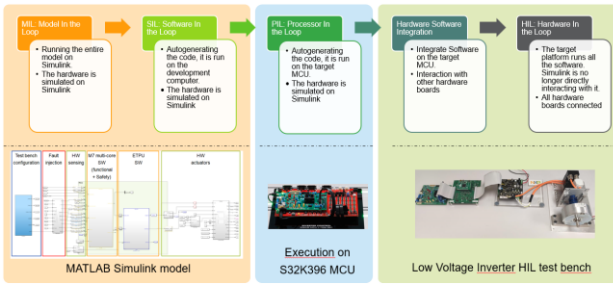

Figure 12 : *Integration of Generated code.*

- *Model In the Loop*: The MIL testing allows to verify the System's behavior from a safety and functional aspect thanks to time-based simulations configured with multiples scenario, including fault injection scenarios. It permits to verify the fault detection mechanisms and the respective system reaction.
- *Software In the Loop*: The SIL testing allows to verify the behavior of the code generated by the coder tool using the same testing scenarios, thus assuring it not affecting the behavior and performances of the system.
- *Processor In the Loop:* The PIL testing validates a proper code execution of a software block into the target core of the microcontroller. It permits to get initial code's metrics such as the application's CPU load.
- *Hardware Software Integration*: This step will be explained in the section 4.2 below.
- *Hardware In the Loop*: The HIL is the final validation test of the software code in the real environment, including sensors, actuators, and load equipment's.

4.2. *Code generation and integration*:

The MBD tool used for the FTMC model is Simulink as it allows an automatic code generation translating its models into C code for embedded applications.

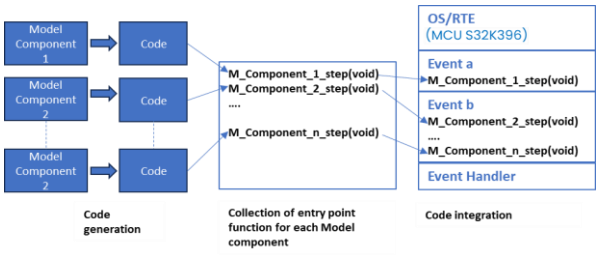

Figure 13 : *Integration of Generated code.*

Each software component is represented by an independent model referenced inside the parent model.

Code is generated from each Software component in the form of a step-function with single entry point. The integration of the generated code is fulfilled by collecting their entry point function and by placing them into application code, while adhering to the requirements (event, core & priority) as is explained in Figure 14.

5. Multicore Safety Concept Architecture:

ASIL decomposition is applied at the Technical Safety concept in such a way that the SW is split in two main parts:

- *Doer SW:* Corresponds to the functional software (Field oriented motor control algorithm and is classified as QM(D).
- *Checker SW:* Corresponds to the monitoring software and is classified as ASILD(D).

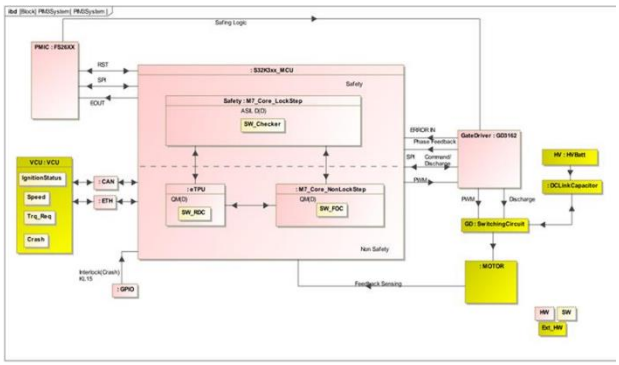

Figure 14 : *Core architecture and allocation.*

The QM(D) part of the Software is executed on several standard core (non-trusted environment) and the Safety functions/mechanisms ASIL D(D) are executed on a M7 lock step core (trusted environment). The SW architecture is designed in such a way to ensure the Safety mechanisms shall detect the errors/faults that can occurs in the functional software and take the system to a safe state within the Fault Tolerant Time Interval.

5.1. *Classic Safety Architecture (Checker + Safety manager)*:

The safety software elements are implemented in a configurable way to ensure a compatibility with a large type of motor control applications:

- *Safety mechanisms*: The usage of safety mechanisms and their configuration can be adapted based on the application need (limits, debouncing).
- *Reaction matrix*: The reaction matrix permits to build a configurable reaction based on the fault detection of dedicated combination of safety mechanisms confirmed detection.

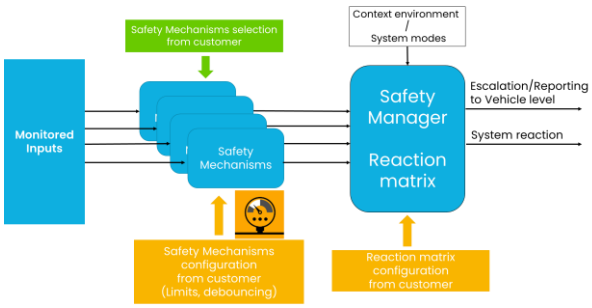

Figure 15 : *Configurable Safety software*.

5.2. *Fault tolerant safety manager implementation*:

Fault tolerant algorithms have been incorporated as a separate function into the safety manager. The latter will call this function depending on the reaction matrix.

The safety manager, mechanisms, and the fault tolerant safety manager, are all integrated in the safety core which is configured to run on lock-step to ensure ASIL (D) compliance.

5.3. *Observer implementation*:

These observer design processes explained in sections 3.4-3.5 provides redundant input data to the control loop in case of sensor failure.

For this reason, these estimators are usually implemented in a separate Core, and synchronized with the functional Core.

5.4. *Synchronization and isolation for three-core implementations* **:**

An initial obstacle of the application integration is the synchronization of the three cores at each 100us cycle. To overcome this issue, a SW barrier function is developed as “a gate” that stays closed until the three cores reach it.

The second challenge is the cores isolation, the only shared resource of the three cores is the memory. Before each 100us cycle, the cores will fetch the new data in the shared memory and copy the necessary it into local variables, this operation is also monitored via a barrier, which will ensure the core isolation and prevent mutual memory access.

6. Verification results (MIL)

6.1. *Position Observer*:

In this MIL test scenario, the electric motor is simulated for 1 second and slowly accelerates from 0 rad/s to maximum speed (~500 rad/s).

The estimations (as shown in Figure 16) exhibit a high degree of accuracy and reliability, validating the effectiveness of the position observer design. The estimated rotor angle and speed closely match the reference signals obtained from the resolver, with minimal steady-state error.

The performance at low and zero speeds is particularly noteworthy, as it highlights the robustness of the observer in scenarios where traditional back-EMF-based methods typically struggle. The good estimation results underscore the observer’s potential for enhancing reliability and safety in sensor-less PMSM control systems.

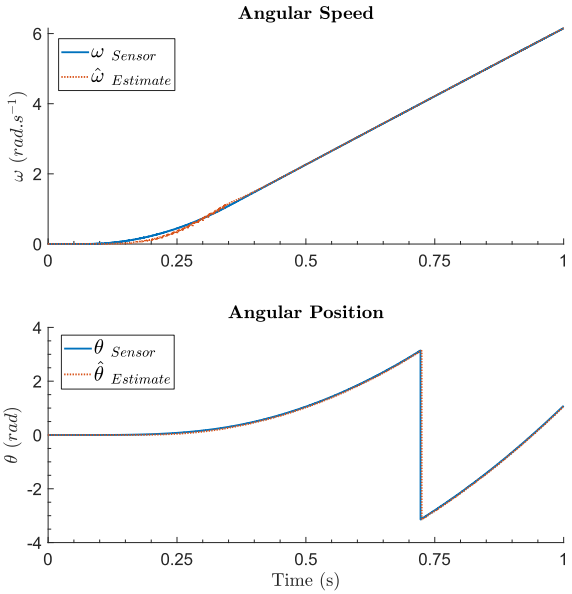

Figure 16 : *Observer Performance – Position and Speed.*

This alignment confirms the observer’s capability to maintain precise tracking under normal operating conditions.

6.2. *Three-phase Current Observer***:**

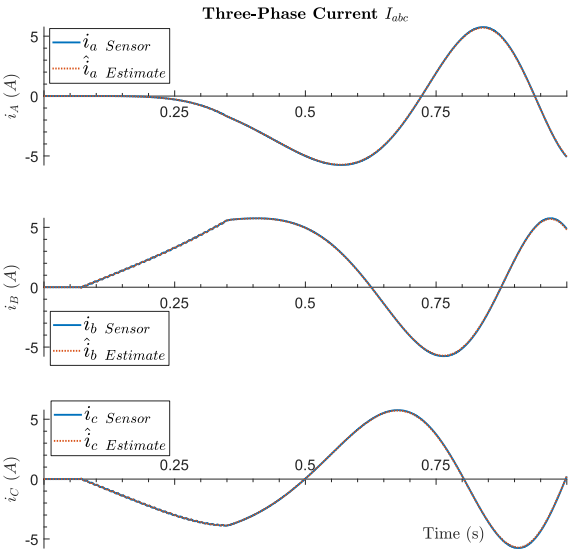

Figure 17 : *Observer Performance – Three-phase Current***.**

As seen in Figure 17, the three-phase current observer demonstrates exceptional tracking accuracy, effectively estimating the motor currents with minimal deviation from the measured signals.

The estimated current waveforms closely follow the actual phase currents under varying load and speed conditions, indicating robust performance. This reliability is critical for ensuring precise field-oriented control (FOC), as accurate current estimation directly impacts the torque and flux control loops.

6.3. *Fault tolerance control (Resolver faults)***:**

In this MIL test scenario, the electric motor is simulated for 0.9 second and accelerates from 0 rad/s to maximum speed (~500 round/s). A fault is injected to the resolver sensor at 0.6th s, immediately after the fault is detected and isolated by the system the PMSM is slowed down and then stopped. The electric motor is then restarted sensorless.

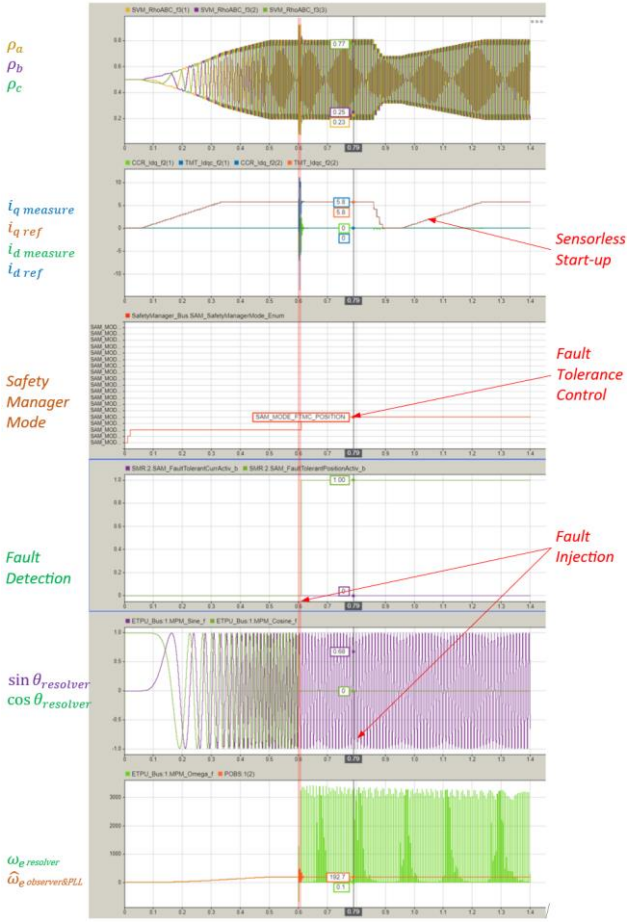

Figure 18 : *Clockwise rotation and restarting after fault injection on* $\boldsymbol{cos\,\theta_{res}}$ *– sensor-less restarting.*

The results show in the following figures demonstrates strong fault tolerance and effective reconfiguration capabilities after the injection of a fault in the resolver's cosθ signal. The observer-based fault detection mechanism successfully identifies the discrepancy caused by the fault (at 0.6th second), triggering the safety reconfiguration process (after 10 ms). Post-reconfiguration, the system adapts by relying on the estimated position from the observer, ensuring continuous and accurate control of the PMSM. The tracking performance of the observer remains stable, maintaining precise position and speed estimates despite the fault. This robust fault recovery highlights the effectiveness of the observer design in maintaining control system reliability, ensuring that the PMSM operates seamlessly even under fault conditions. These results reinforce the observer's role as a critical component for safety in sensor-less control applications.

Several similar tests were conducted with fault scenarios occurring on the current sensing phases. The system was able to locate the faulty phase and isolate it, then reconfigure to fault tolerance control mode (however, due to the limitation of the paper format, the results are exposed in the associated presentation).

7. Validation results (HIL)

For the HIL tests, a 24V PMSM machine is controlled by the MCU, which sends PWM commands to the power board.

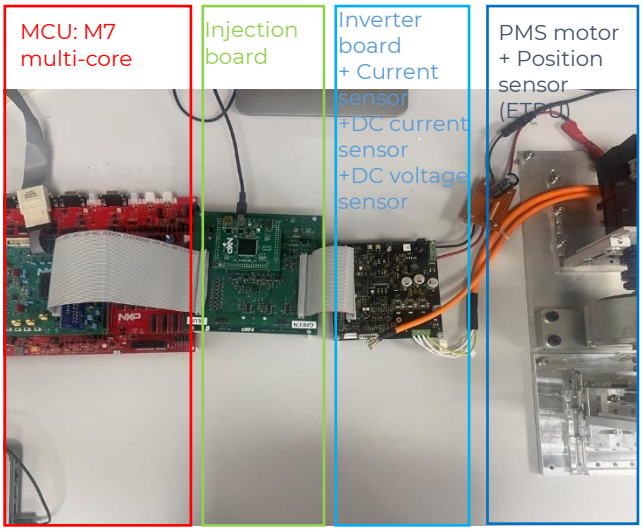

Figure 19 : *Test Bench – PMSM drives***.**

There is the injection board which distorts the sensors signals to create faults. This setup allows to assess the behavior of our functional safety concept during normal operation and in the event of faults.

The position observer system exhibits a rapid transient response, with the observer accurately capturing dynamic changes in speed and position without significant lag or overshoot. The system's resilience to noise and disturbances further reinforces its suitability for pratique applications, ensuring stable and consistent operation. The smooth tracking even in the presence of noise and system disturbances,

validates the design's robustness and its suitability for pratique applications.

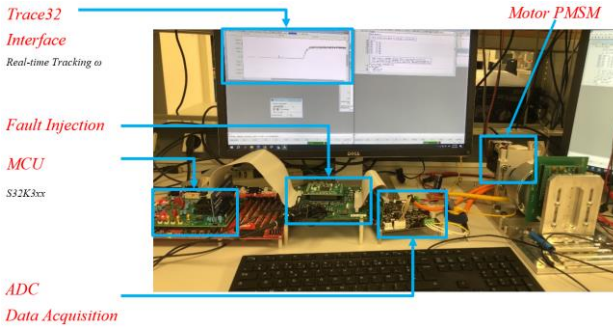

Figure 20 : *Hardware in the Loop – PMSM motor control Kit S32K3xx with Trace32*.

The designed observer's ability to manage transient conditions, such as sudden load changes or speed variations, further highlights its reliability, maintaining stable and consistent estimations throughout. These results affirm the observer's potential to enhance the efficiency and safety of PMSM control systems.

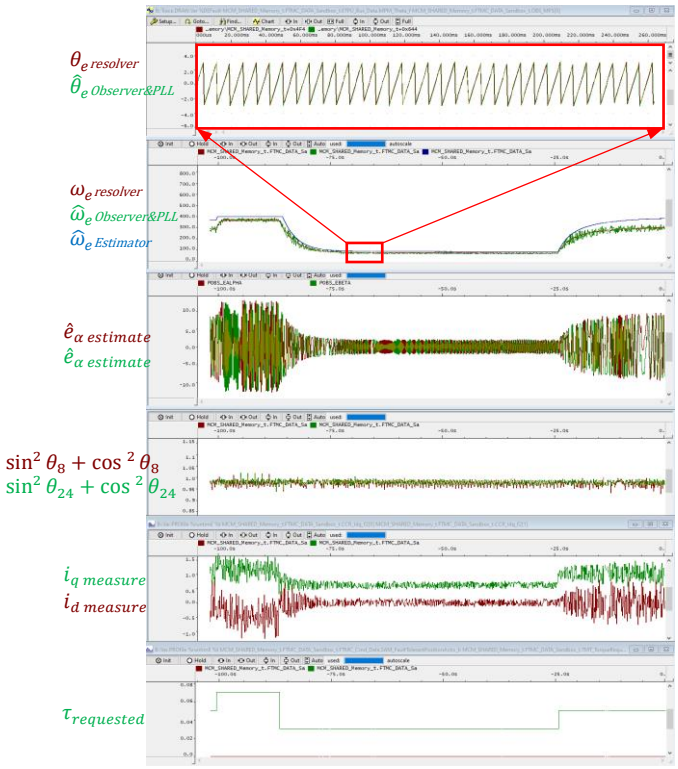

Figure 21 : *Performance of Observer – Angular Position and Speed estimations – Varying torque*.

8. Conclusion

This paper has provided guidelines and solution to increase availability in a safety context for EV traction inverter using SW redundancies like digital twin instead of HW solutions. It has demonstrated use of MBD development for complexes algorithm design and simulation for fault tolerant safety concept verification around motor position and phase current observers. Challenges for multicore implementation have been and rapid prototyping have been detailed. Finally, validation on bench have demonstrated as well good performance of detection, isolation and system reconfiguration in case of fault thanks to those sensor digital twin embedded in a real automotive system multicore environment.

Acknowledgement

The authors acknowledge the contribution of all NXP Semiconductors CTO functional safety colleagues for their participation/contribution to this work. Special thanks to Jerôme Dietsch for his review and for his contribution through the Model Base Design project development for inverter.

References

[1] ISO Standard: "ISO 26262 Functional Safety", 2018.
[2] NXP: " *S32K3 Arm Cortex-M7-Based Automotive MCUs Brochure", 2024.*
[3] J Dietsch: " *Usage of complex systems simulation for the development of functional safety algorithms*", SIA, France, 2025.
[4] Authors: "AN14326: 3-phase Motor Control Kit with S32K396 Application Note", 2024.
[5] A.Dubois: "safety-concept-overview-hv-traction-inverter", White Paper, 2021.
[6] https://aleksandarhaber.com/correct-explanation-of-observer-state-estimators-of-state-space-models/

Glossary

MBD: Model Based design.
EV: Electrical Vehicle.
VCU: Vehicle Control Unit.
HV: High Voltage.
DC: Direct Current.
HW: Hardware.
SW: Software.
MCU: Microcontroller Unit.
MIL: Model In the Loop.
SIL: Software In the Loop.
PIL: Processor In the loop.
HSI: Hardware Software Integration.
HIL: Hardware In the Loop.
PWM: Pulse Width Modulation.
EMF: Electromagnetic Field.
ASIL: Automotive Safety Integrity Level.
QM: Quality Managed.
FDI: Fault detection and identification.
PMSM: Permanent Magnet Synchronous Motor.