Verification and Validation Report: Mechatronics

Team 28, Controls Freaks
Abhishek Magdum
Dharak Verma
Jason Surendran
Laura Yang
Derek Paylor

April 7, 2023

Revision History

Date	Version	Notes
March 8, 2023	1.0	Initial Revision
April 7, 2023	2.0	Final Revision: Added results for additional unit and integration tests

Symbols, Abbreviations and Acronyms

See SRS Documentation here.

Contents

List of Figures

This document provides the results of unit tests and integration tests the system underwent according to the VnV Plan documentation and to ensure requirements layed out in the SRS are met and traceable to tests performed.

1 Functional Requirements Evaluation

Based on the results in Sections 3 & 4, our control system meets basic functionality requirements. This VnV Report has prompted the team to generate a more comprehensive selection of tests on the system level, to cover requirements that are currently only covered under unit testing.

2 Nonfunctional Requirements Evaluation

All non-functional requirements have been met sufficiently.

2.1 Look and Feel

TID	Outcome	Result
NFT1	Manual scan of all layers of the control system shows consistent colour schemes and variable naming according to <origin>_<valueType>_<description>	Pass

This naming scheme made development, debugging, and maintaining the model much easier. Since finding the exact reference to a signal can be cumbersome in Simulink, our naming scheme greatly decreased the time and effort required to understand a model or set of blocks. On top of that, when the control system was converted to C code, the naming scheme was a great aid in integrating the relevant input and output variables into the embedded C layer, as we knew exactly where the signal came from.

2.2 Performance

TID	Outcome	Result
NFT2	As demonstrated during the rev0 demo, code deployment process was successful to STM32 hardware	Pass

The Simulink model was successfully converted into C code using the Simulink code generation toolkit. The autogenerated code was then integrated into our embedded C wrapped and successfully flashed onto MFE's STM32F767ZI Front Controller ECU. The Front Controller was able to step through the control system at its required periodicity, interfacing with IO via CAN (demonstrated with live CAN decoding), and GPIO/ADC (demonstrated by moving a potentiometer and seeing real-time variable change).

2.3 Maintainability and Support

TID	Outcome	Result
NFT3	Through the use of Simulink library blocks, each module was encapsulated and independently modifiable for implementation	Pass

This proved to be a mainstay in the ability of our team to gradually add in features without breaking the system.

TID	Outcome	Result
NFT4	As shown in the Plant models in the system test environments (simulation and bench testing), hardware interfaces are isolated from control system logic	Pass

During bench testing (running the control system with real CAN hardware and motor kit), hardware hiding was critical as CAN scaling / unit conversions were a continual point of issue, but this could be reflected in our model separately from the control logic.

TID	Outcome	Result
NFT5	As shown in Section 3 of this document, the unit test environments are sufficient to provide base-level functional validation	Pass

TID	Outcome	Result
NFT6	As shown in Section 4 of this document, the system test environment is sufficient to provide integration validation of key features through simulating vehicle responses.	Pass

2.4 Security and Support

NFT7 is a fairly self-evident check and does not require formalized verification.

3 Unit Testing

Unit testing was carried out by creating single-module environments in which module inputs are supplied via manually-created timeseries signals (created via Simulink's Root-level importer and Signal Editor), and outputs are logged during simulation for inspection with a scope-like interface (Simulink Data Inspector). These environments are shown for each module below.

3.1 Governor Module

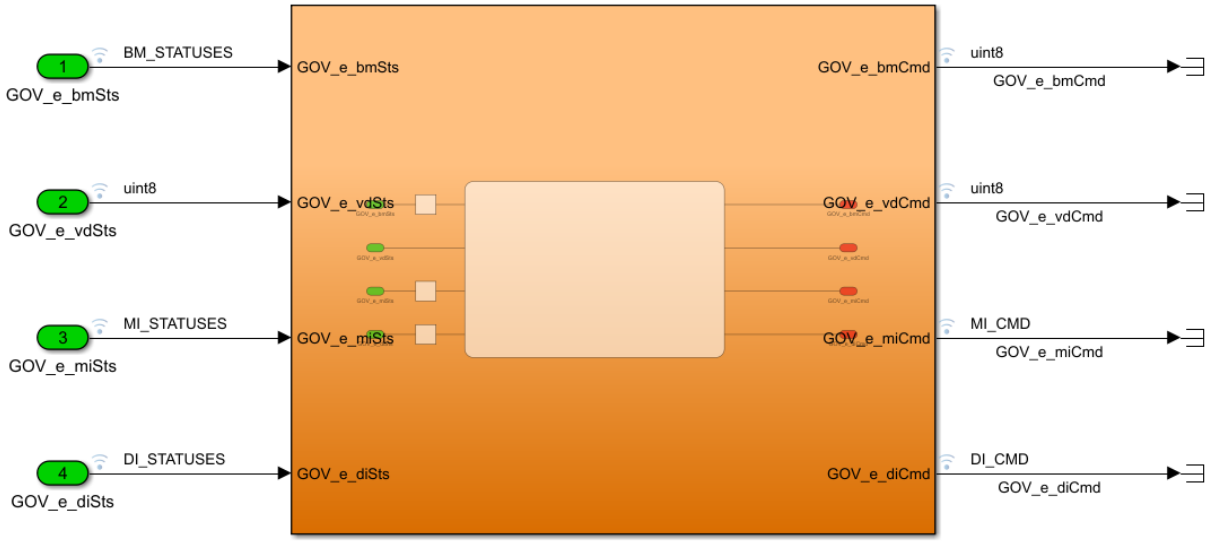

Figure 1: Governor - Unit Test Simulation Environment

TID	Expected Result	Actual Result	Result
UT_G1	Governor issues MI command 'motor startup' following BM status = 'running'; issues 'Ready to drive' command to DI following MI status = 'running'	Governor issues MI command 'motor startup' following BM status = 'running'; issues 'Ready to drive' command to DI following MI status = 'running'	Pass

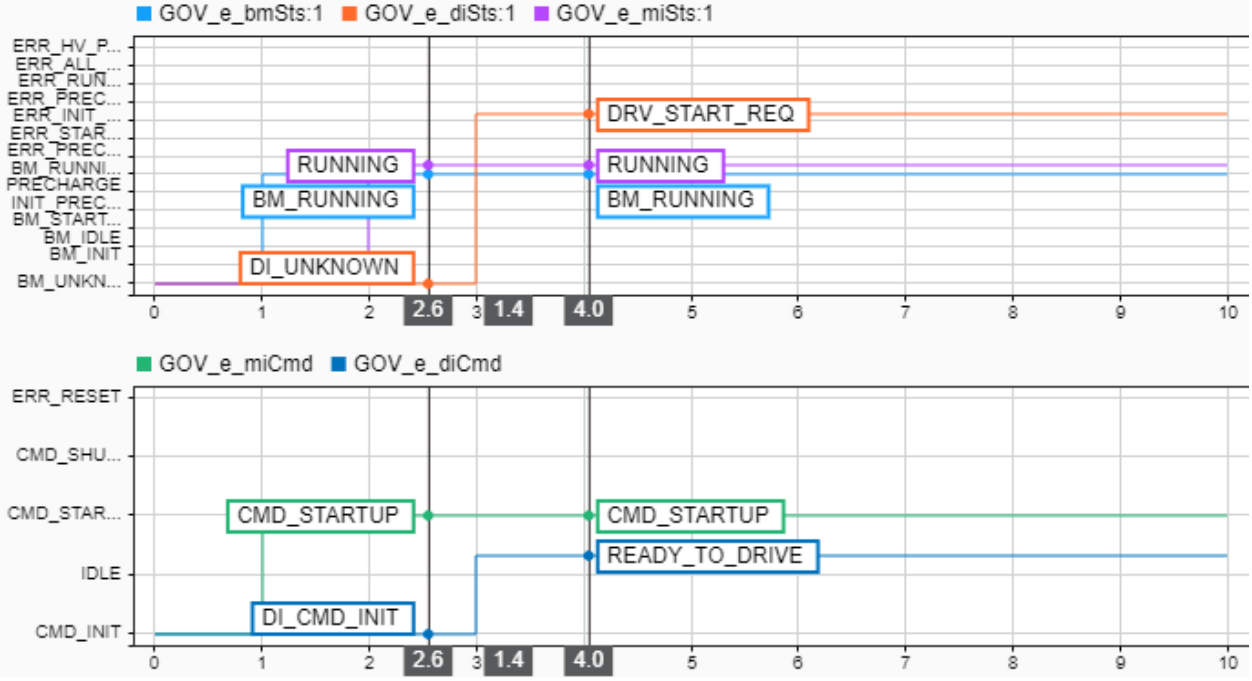

Figure 2: UT_G1 Simulation Results

TID	Expected Result	Actual Result	Result
UT_G2	Governor issues DI command 'system error' following MI status = 'error'	Governor issues DI command 'system error' following MI status = 'error'	Pass

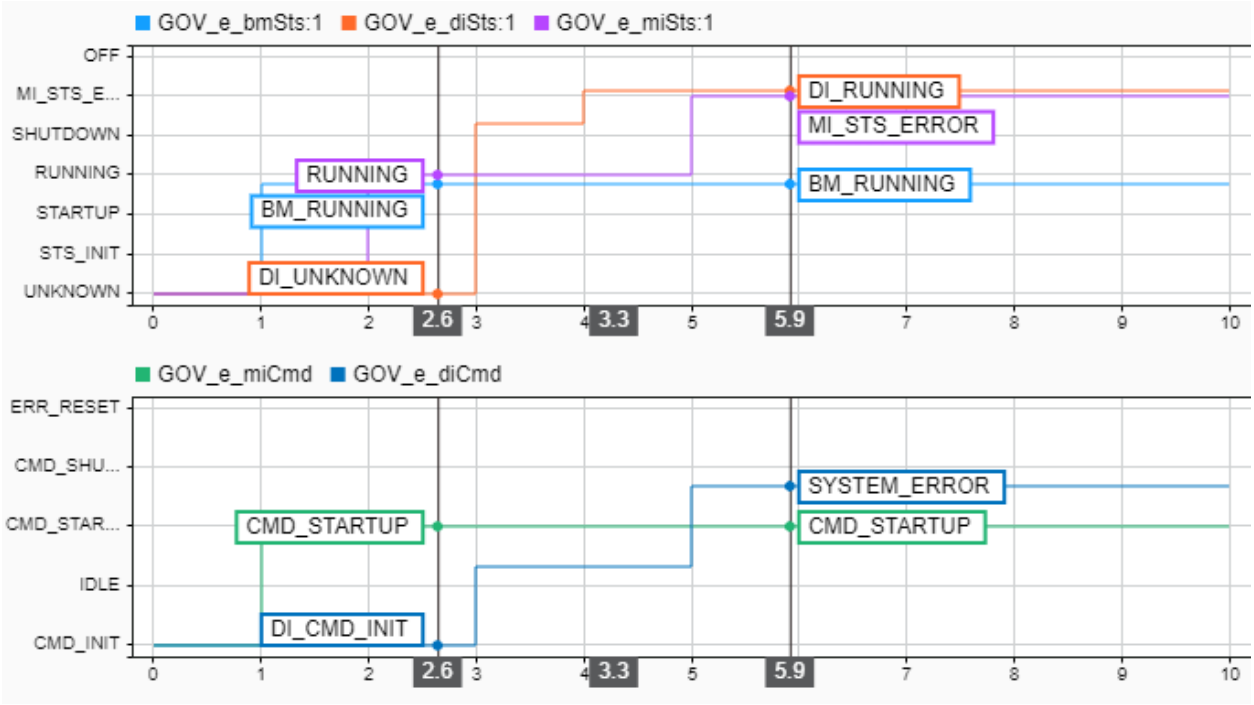

Figure 3: UT_G2 Simulation Results

3.2 Driver Interface Module

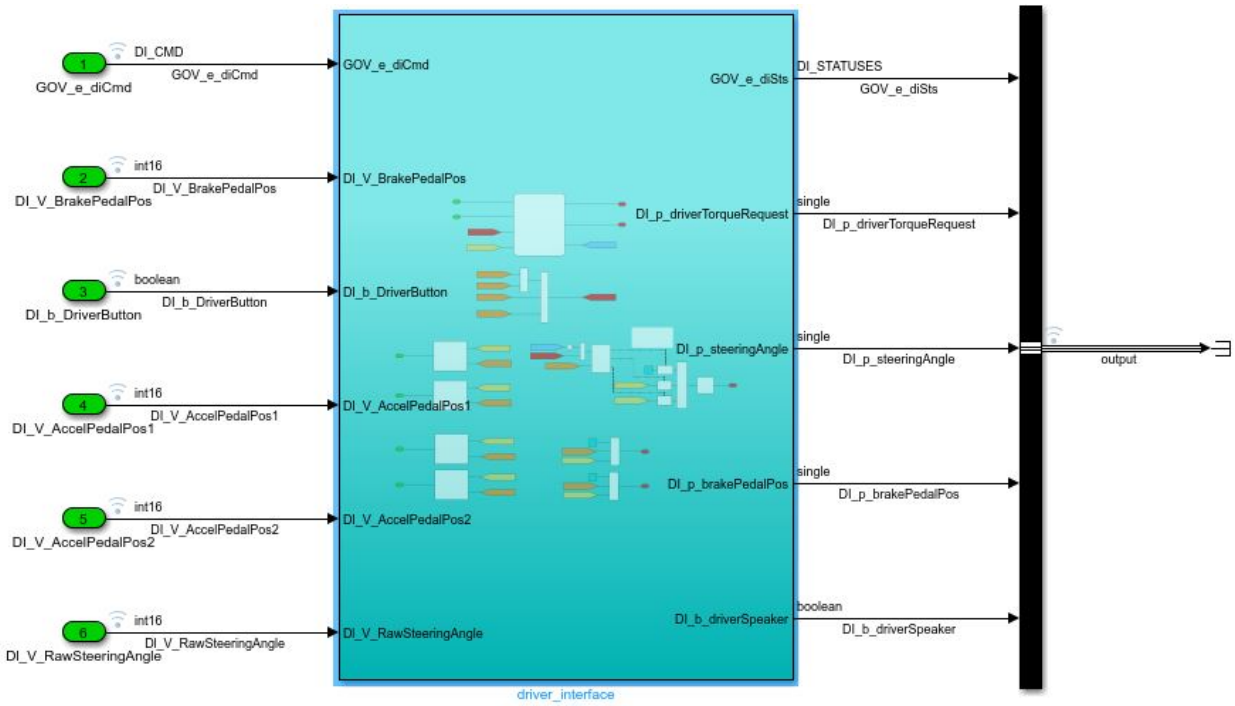

Figure 4: Driver Interface - Unit Test Simulation Environment

TID	Expected Result	Actual Result	Result
UT_DT1	DI reports "driver start request" status after button is on	DI reports "driver start request" status after button is on	Pass

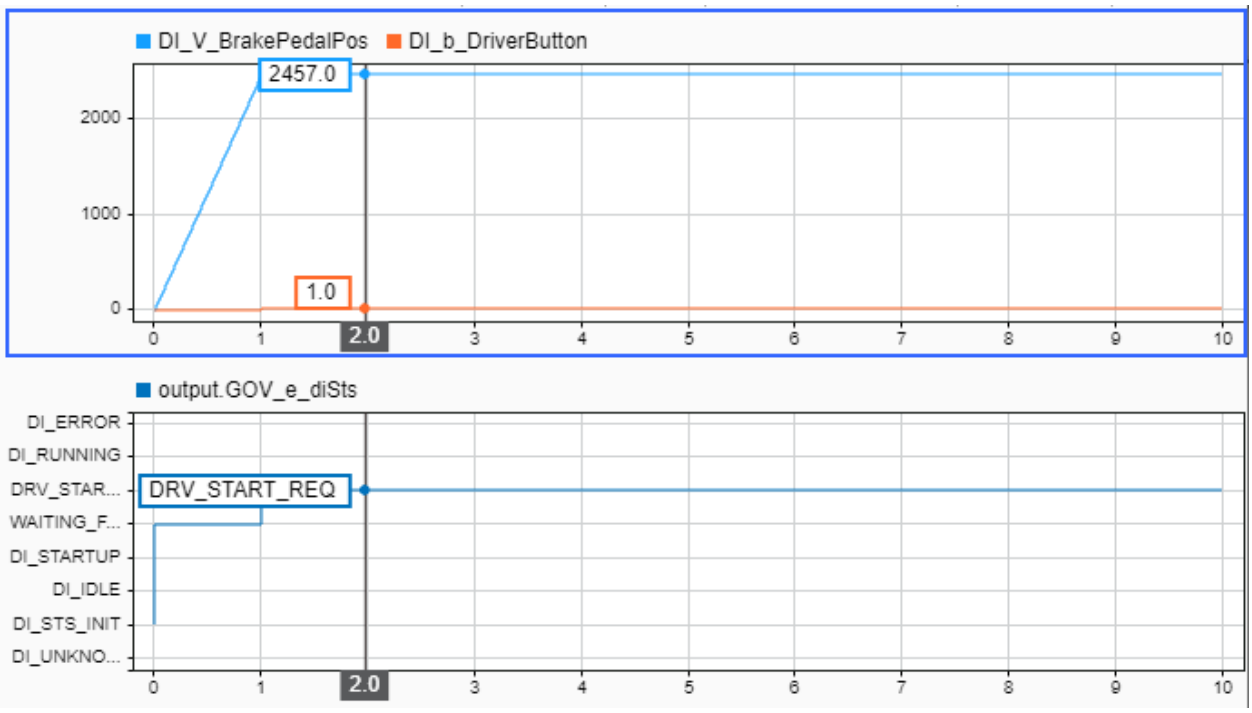

Figure 5: UT_DI1 Simulation Results

TID	Expected Result	Actual Result	Result
UT_DT2	Phase 1 - APPS1 faulted, driver torque request = 50%; Phase 1 - APPS2 faulted, driver torque request = 50%; Phase 3 - BPPS faulted, DI reports 'DI error' status; Phase 4 - Steering position faulted, DI reports 'DI error' status	Phase 1 - APPS1 faulted, driver torque request = 50%; Phase 1 - APPS2 faulted, driver torque request = 50%; Phase 3 - BPPS faulted, DI reports 'DI error' status; Phase 4 - Steering position faulted, DI reports 'DI error' status	Pass

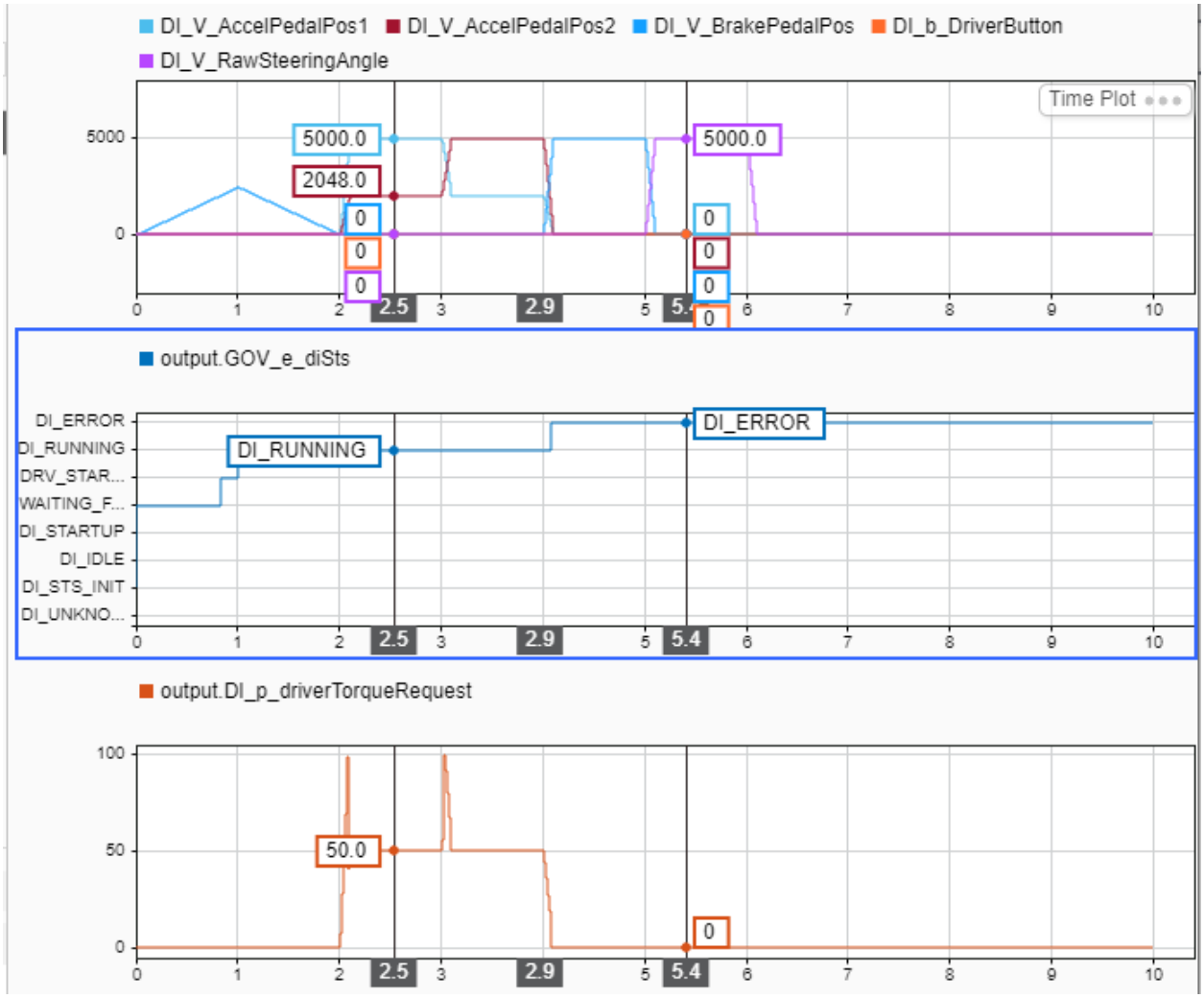

Figure 6: UT_DI2 Simulation Results

TID	Expected Result	Actual Result	Result
UT_DT3	Driver torque request drops to 50% to 0% in response to the system error	Driver torque request drops to 50% to 0% in response to the system error	Pass

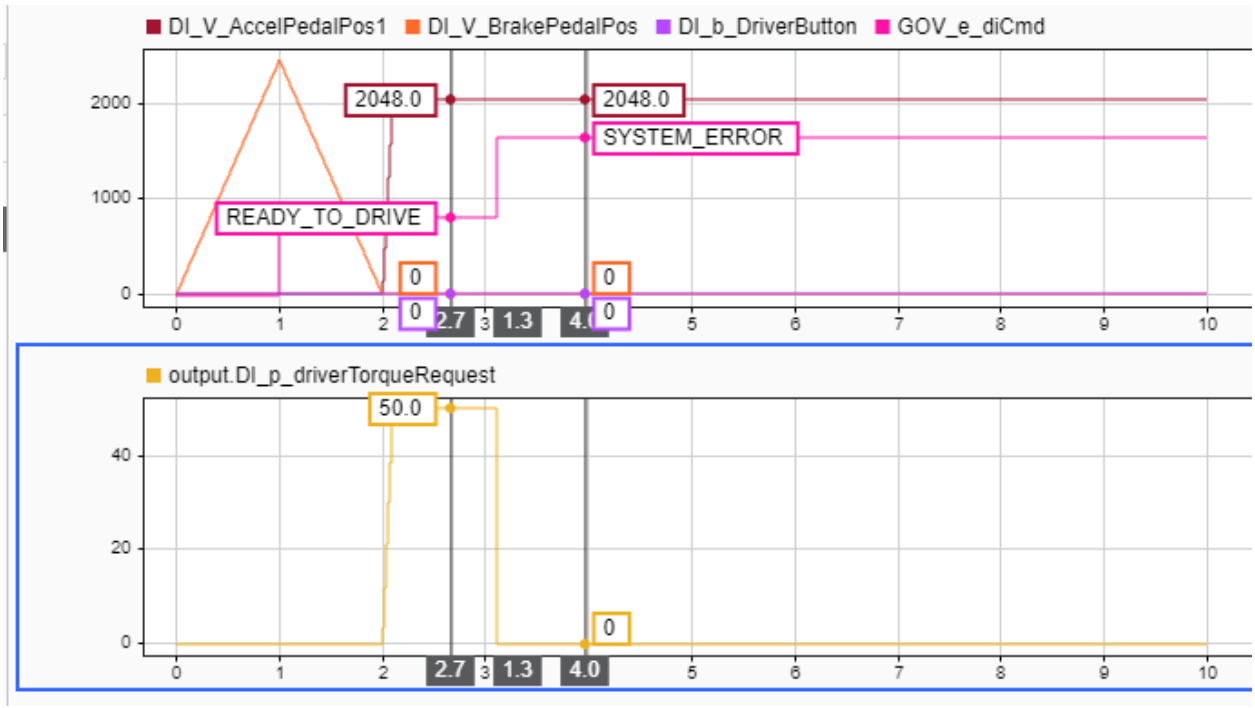

Figure 7: UT_DI3 Simulation Results

3.3 Battery Monitor Module

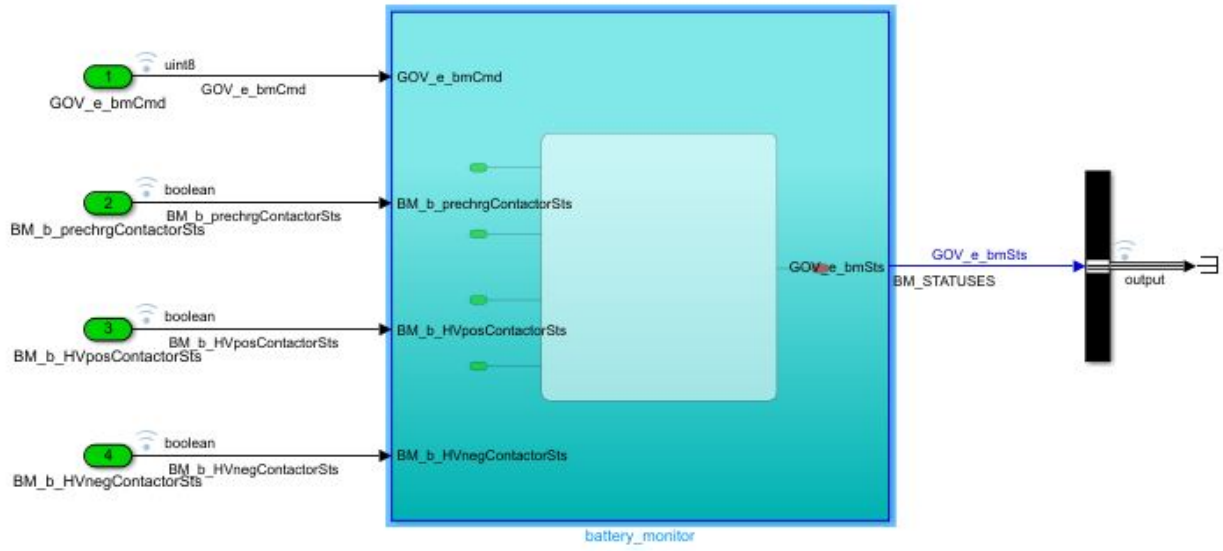

Figure 8: Battery Monitor - Unit Test Simulation Environment

TID	Expected Result	Actual Result	Result
UT_BM1	Battery Monitor reports Error state on startup	Battery Monitor reports Error state on startup	Pass

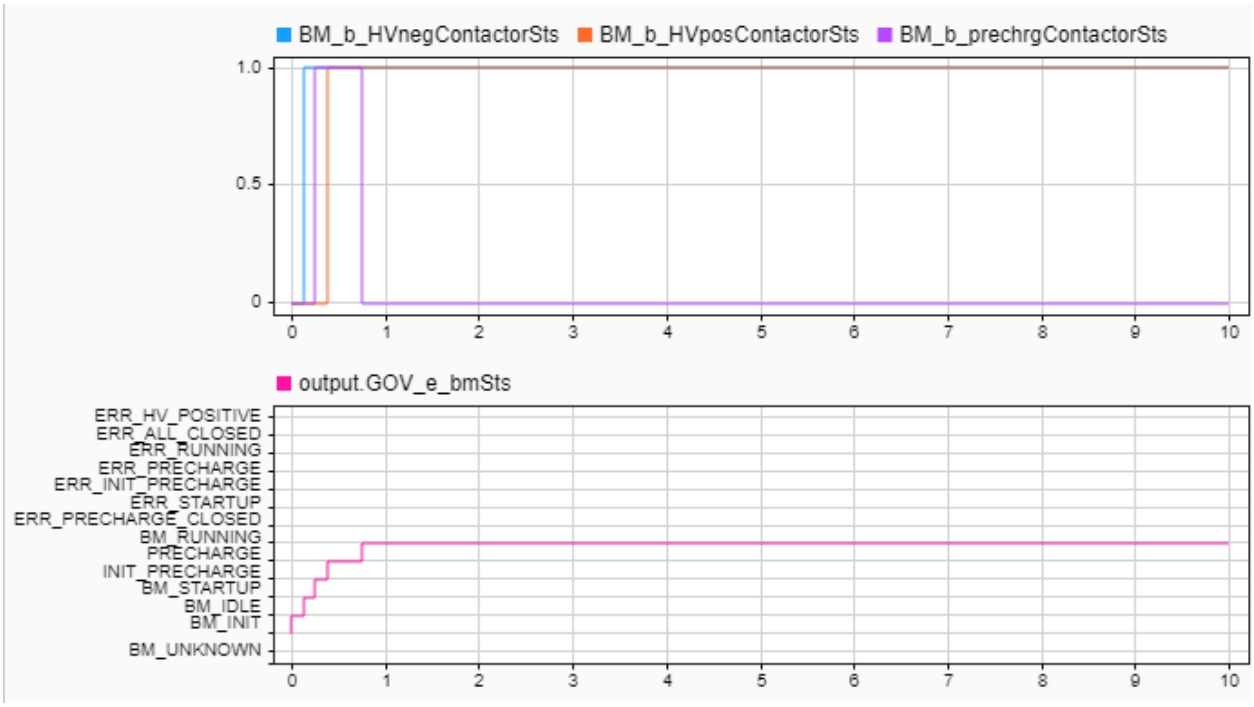

Figure 9: BM1 Simulation Results

TID	Expected Result	Actual Result	Result
UT_BM2	Battery Monitor reports Error state on startup	Battery Monitor reports Error state on startup	Pass

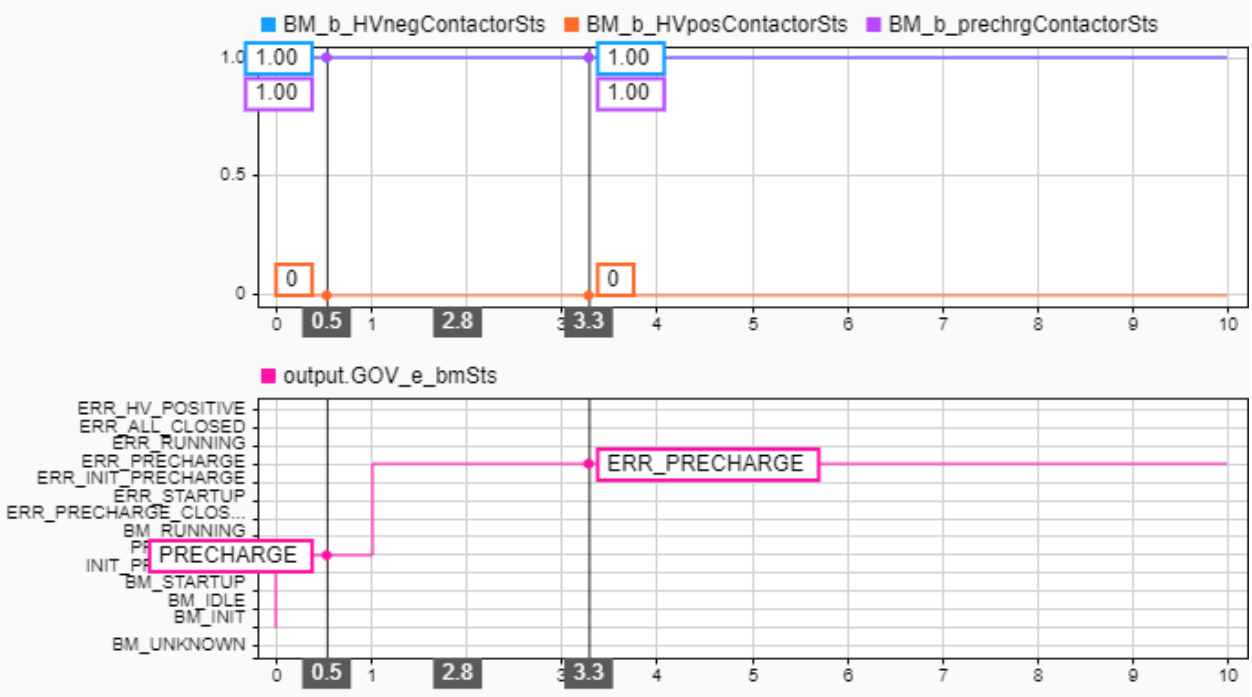

Figure 10: BM1 Simulation Results

3.4 Motor Interface Module

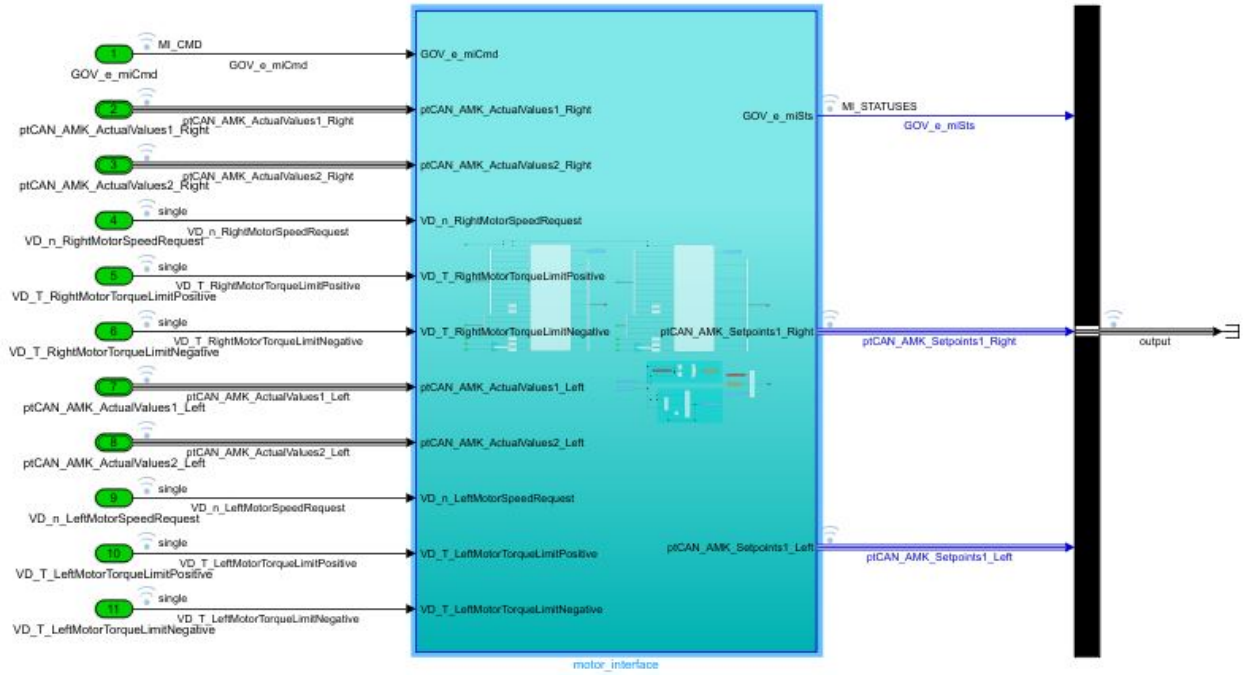

Figure 11: Motor Interface - Unit Test Simulation Environment

TID	Expected Result	Actual Result	Result
UT_MI1	Left motor status = 'running', followed by Right motor status = 'running'; overall MI status = 'running' only once both motors are running	Left motor status = 'running', followed by Right motor status = 'running'; overall MI status = 'running' only once both motors are running	Pass

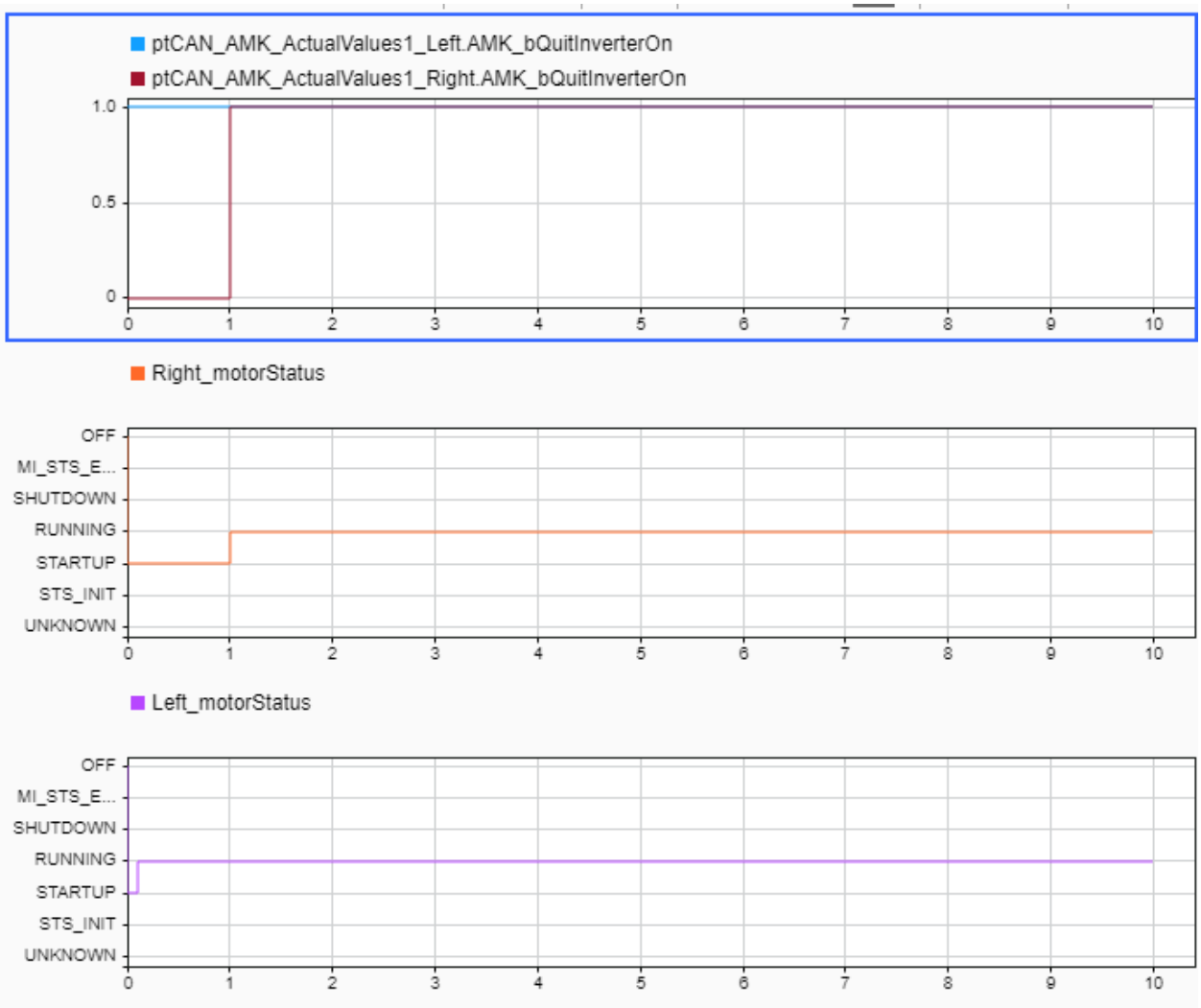

Figure 12: UT_DI1 Simulation Results

TID	Expected Result	Actual Result	Result
UT_MI2	both motors proceed through 'shutdown' to 'off', along with MI status.	both motors proceed through 'shutdown' to 'off', along with MI status.	Pass

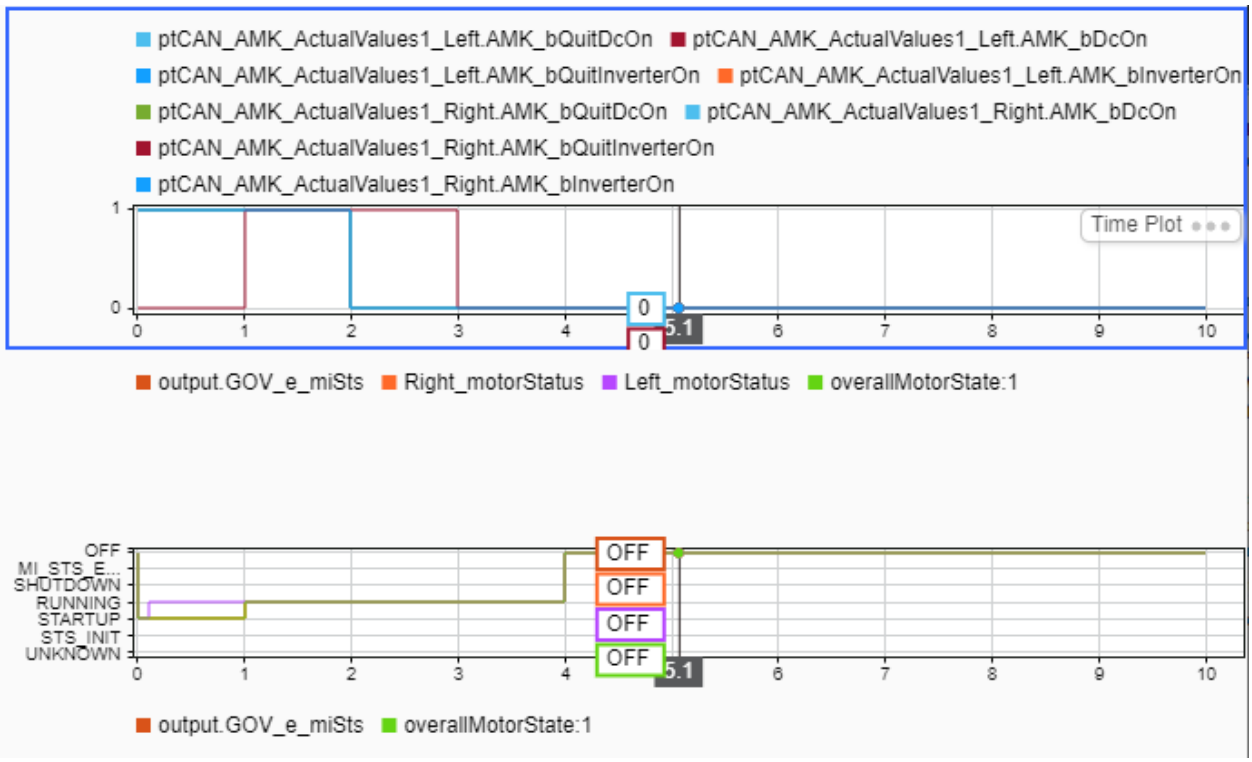

Figure 13: UT_MI2 Simulation Results

TID	Expected Result	Actual Result	Result
UT_MI3	Left motor remains in 'running' state; Right motor transitions to 'error' state; MI status transitions from 'running' to 'error'.	Left motor remains in 'running' state; Right motor transitions to 'error' state; MI status transitions from 'running' to 'error'.	Pass

Note: TMR6 and TMR7 from the SRS are validated under test case VD1.

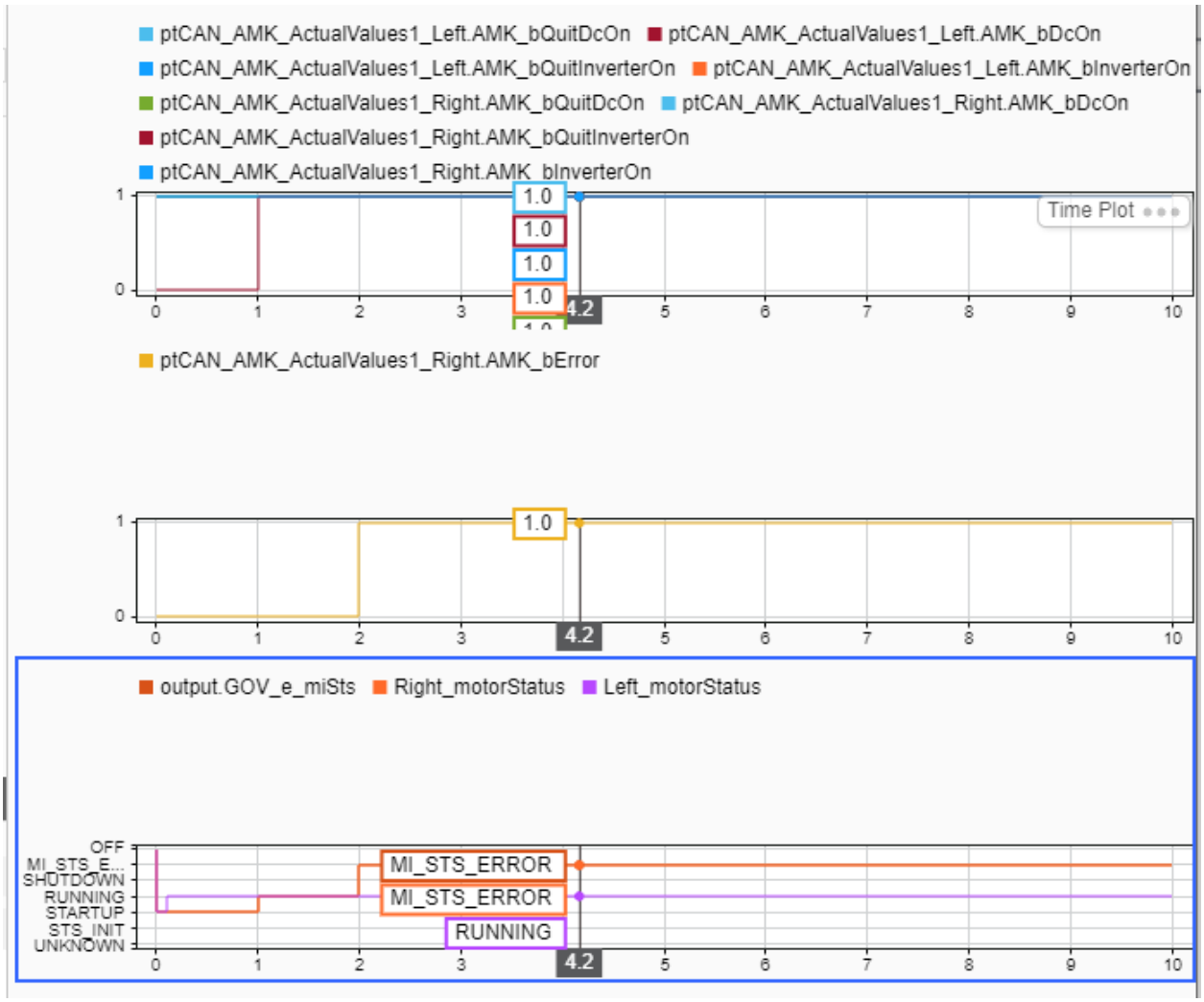

Figure 14: UT_MI3 Simulation Results

3.5 Vehicle Dynamics Module

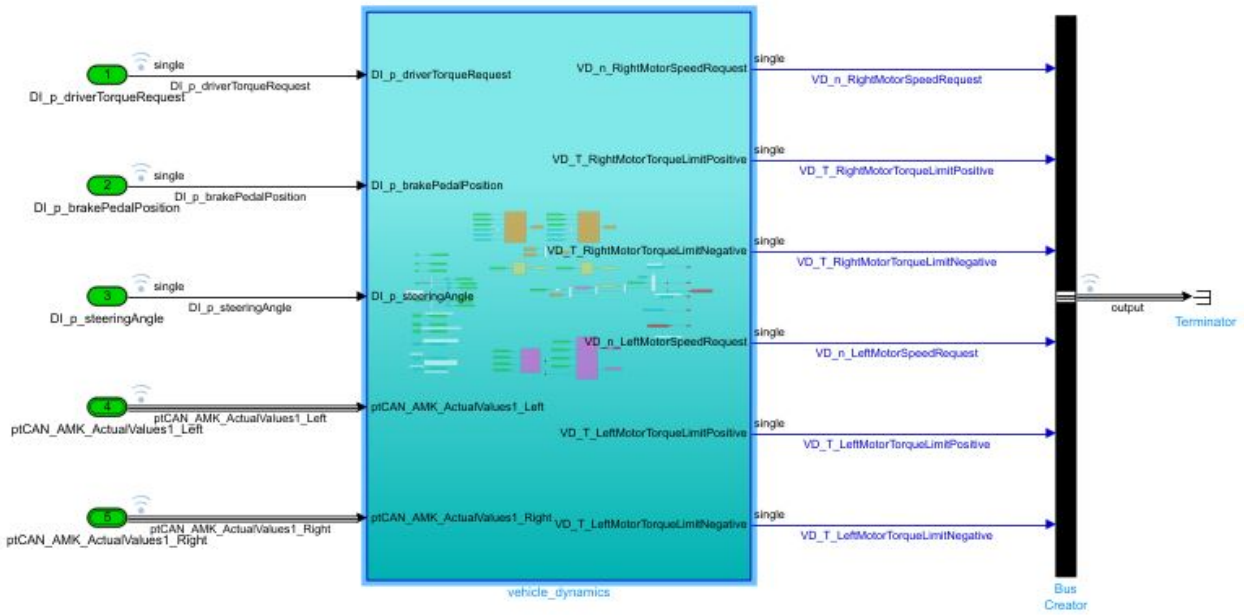

Figure 15: Vehicle Dynamics - Unit Test Simulation Environment

TID	Expected Result	Actual Result	Result
UT_VD1	Torque limit positive (left & right) begins at maximum, then is cut to 0 as the brakes are applied	Torque limit positive (left & right) begins at maximum, then is cut to 0 as the brakes are applied	Pass

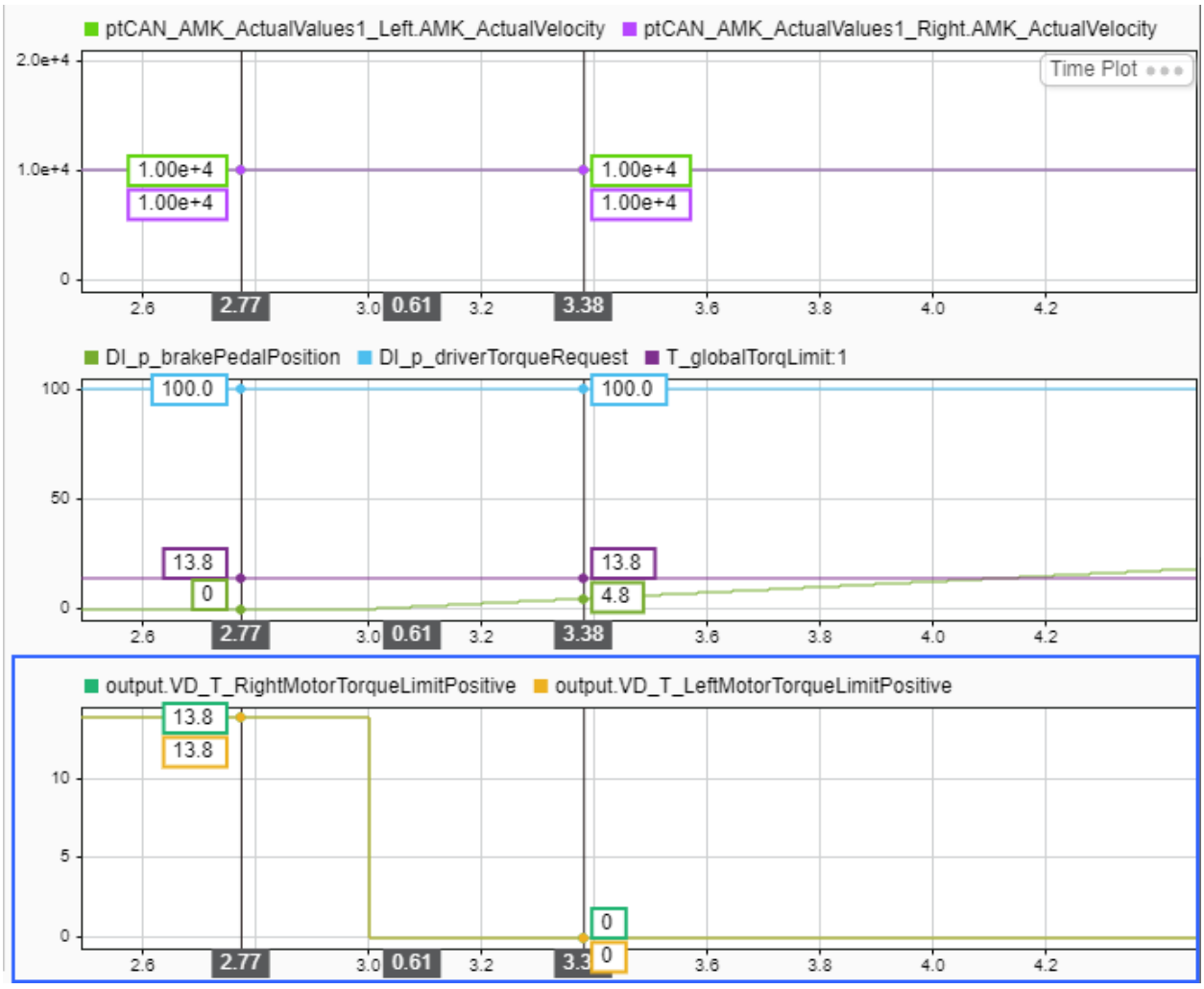

Figure 16: VD1 Simulation Results

TID	Expected Result	Actual Result	Result
UT_VD2	Torque limit positive (left & right) begins at maximum (13.8) in constant-torque operation, then ramps down as the motor enters constant-power operation	Torque limit positive (left & right) begins at maximum (13.8) in constant-torque operation, then ramps down as the motor enters constant-power operation	Pass

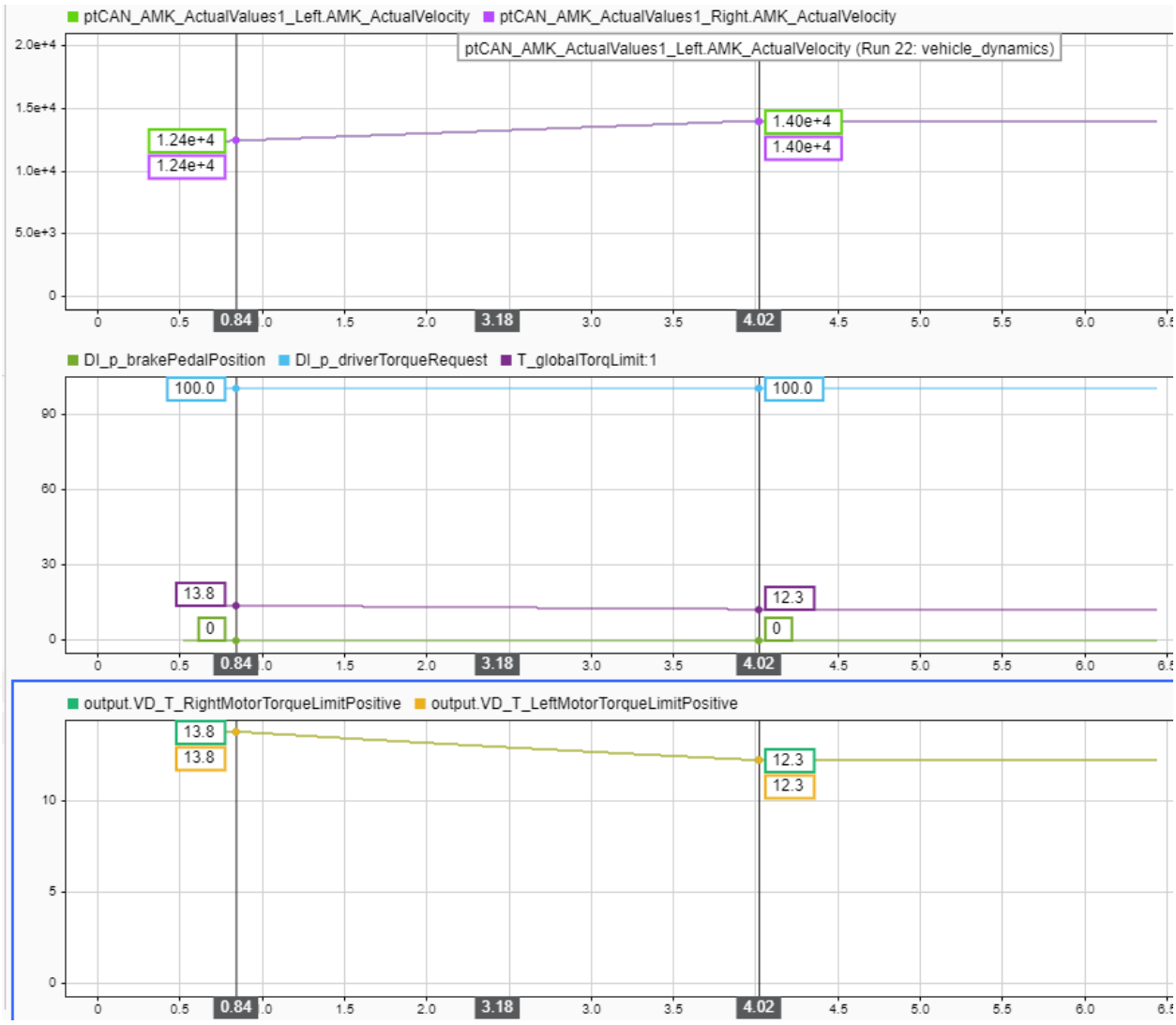

Figure 17: VD2 Simulation Results

4 Integration Testing

Shown below is the simulation environment for our control system. The Controller model (on the left), containing all the subsystems (tested in Section 3), all running simultaneously and supplying signals to a Plant model (on the right). The Plant model was designed by our team to simulate feedback from the vehicle's sensors and other controllers. AMK motor feedback, vehicle speed and acceleration are simulated using a physics model. Driver Inputs and Battery Contactor states are modelled simply by manually-created timeseries signals.

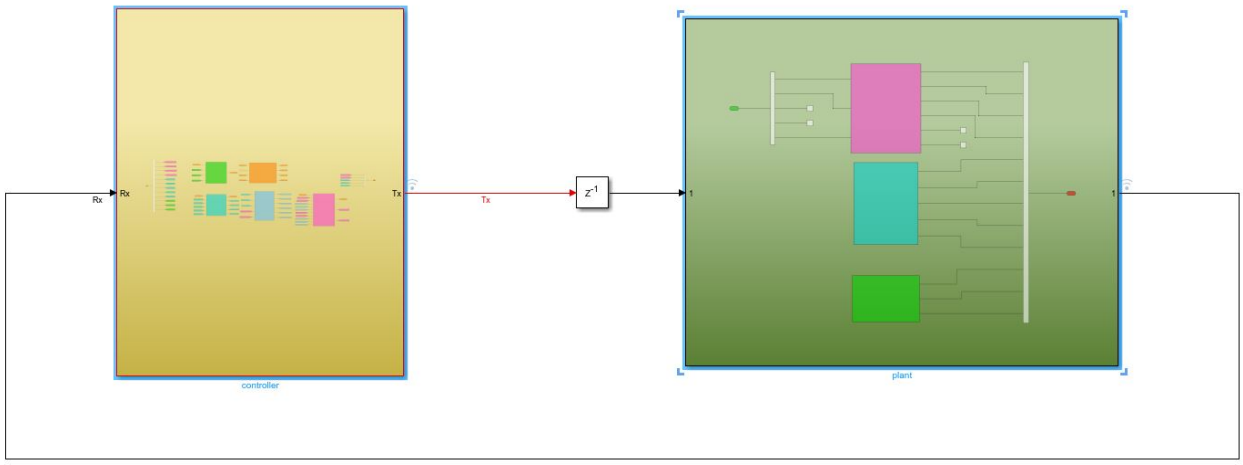

Figure 18: Vehicle Control System - System Test Simulation Environment

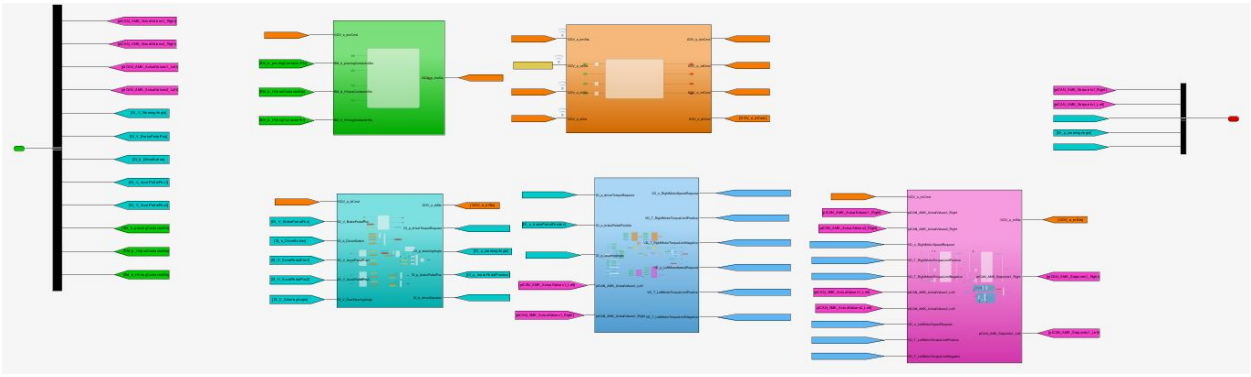

Figure 19: Vehicle Control System - Controller model

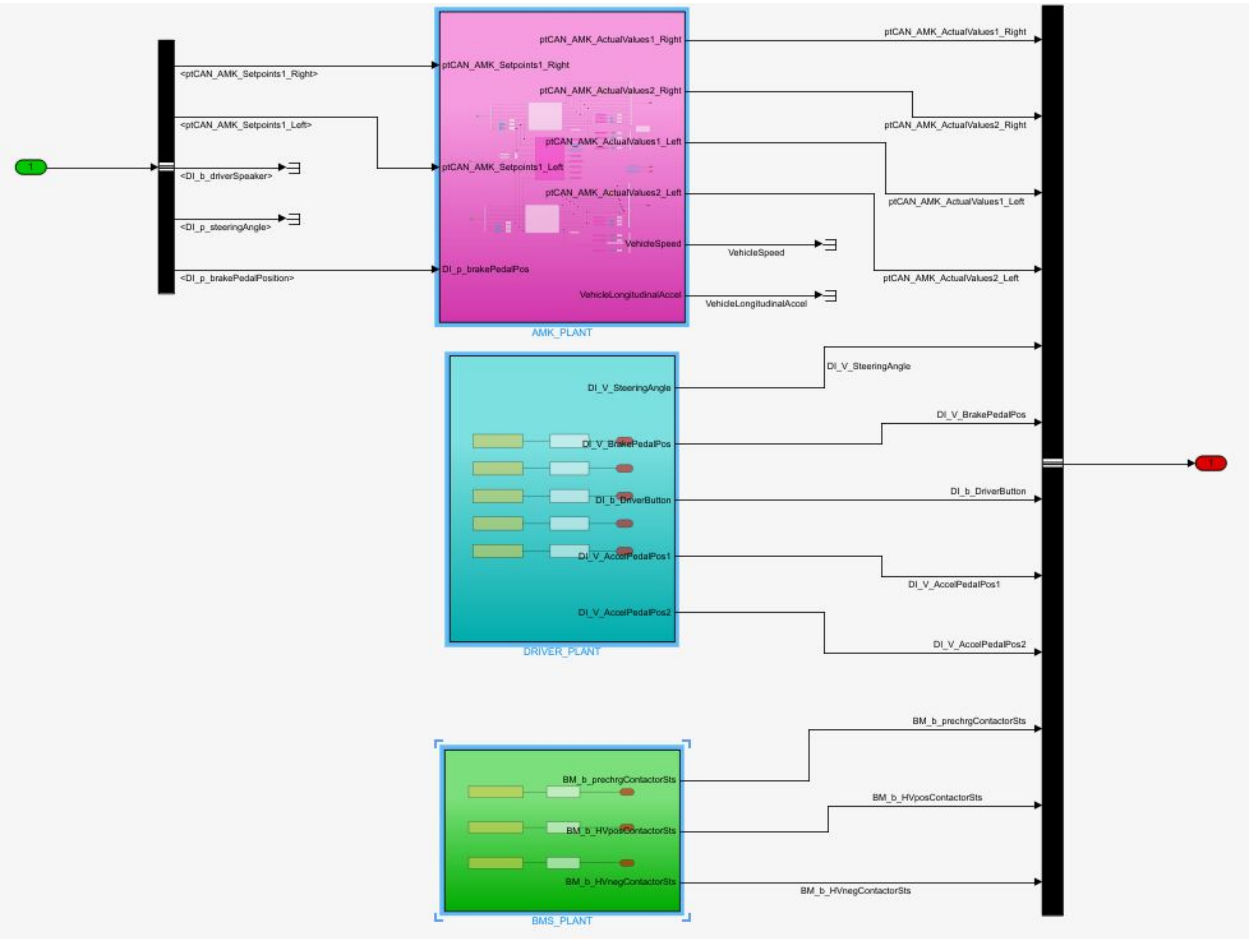

Figure 20: Vehicle Control System - Plant model

TID	Expected Result	Actual Result	Result
ST_VCS1	Governor commands MI to start motors after BM reports 'running', then sends 'ready to drive' after DI reports 'driver start request'. When APPS1 ramps, motor torque is applied and the vehicle accelerates, then coasts down when APPS1 is 0%.	Governor commands MI to start motors after BM reports 'running', then sends 'ready to drive' after DI reports 'driver start request'. When APPS1 ramps, motor torque is applied and the vehicle accelerates, then coasts down when APPS1 is 0%.	Pass

TID	Expected Result	Actual Result	Result
ST_VCS2	MI status reports 'error', Governor issues DI command 'system error', Driver torque request is zeroed, vehicle coasts down	MI status reports 'error', Governor issues DI command 'system error', Driver torque request is zeroed, vehicle coasts down	Pass

TID	Expected Result	Actual Result	Result
ST_VCS3	Motor torque limits (left & right) are zeroed while braking, vehicle comes to an abrupt stop.	Motor torque limits (left & right) are zeroed while braking, vehicle comes to an abrupt stop.	Pass

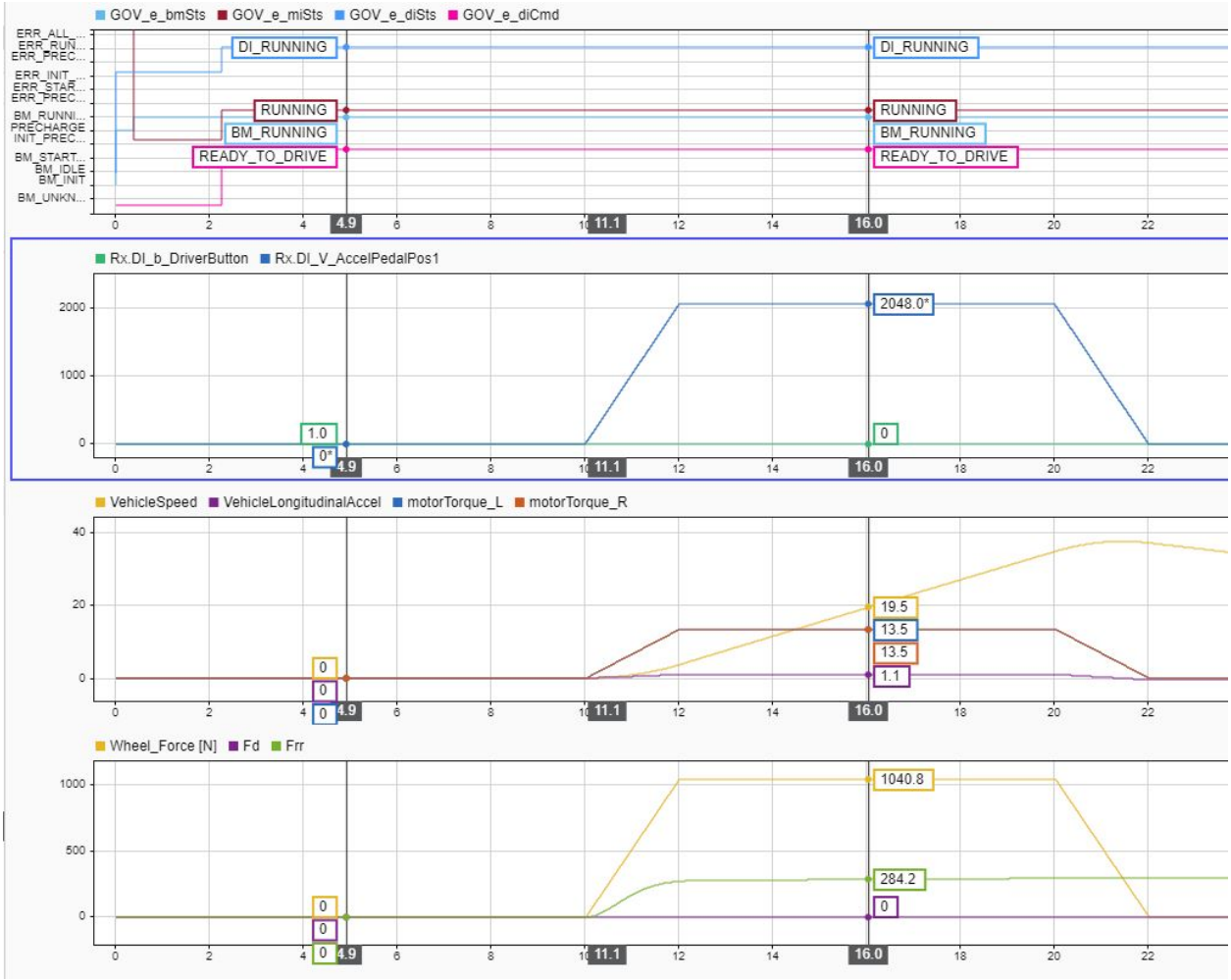

Figure 21: ST_VCS1 Simulation Results

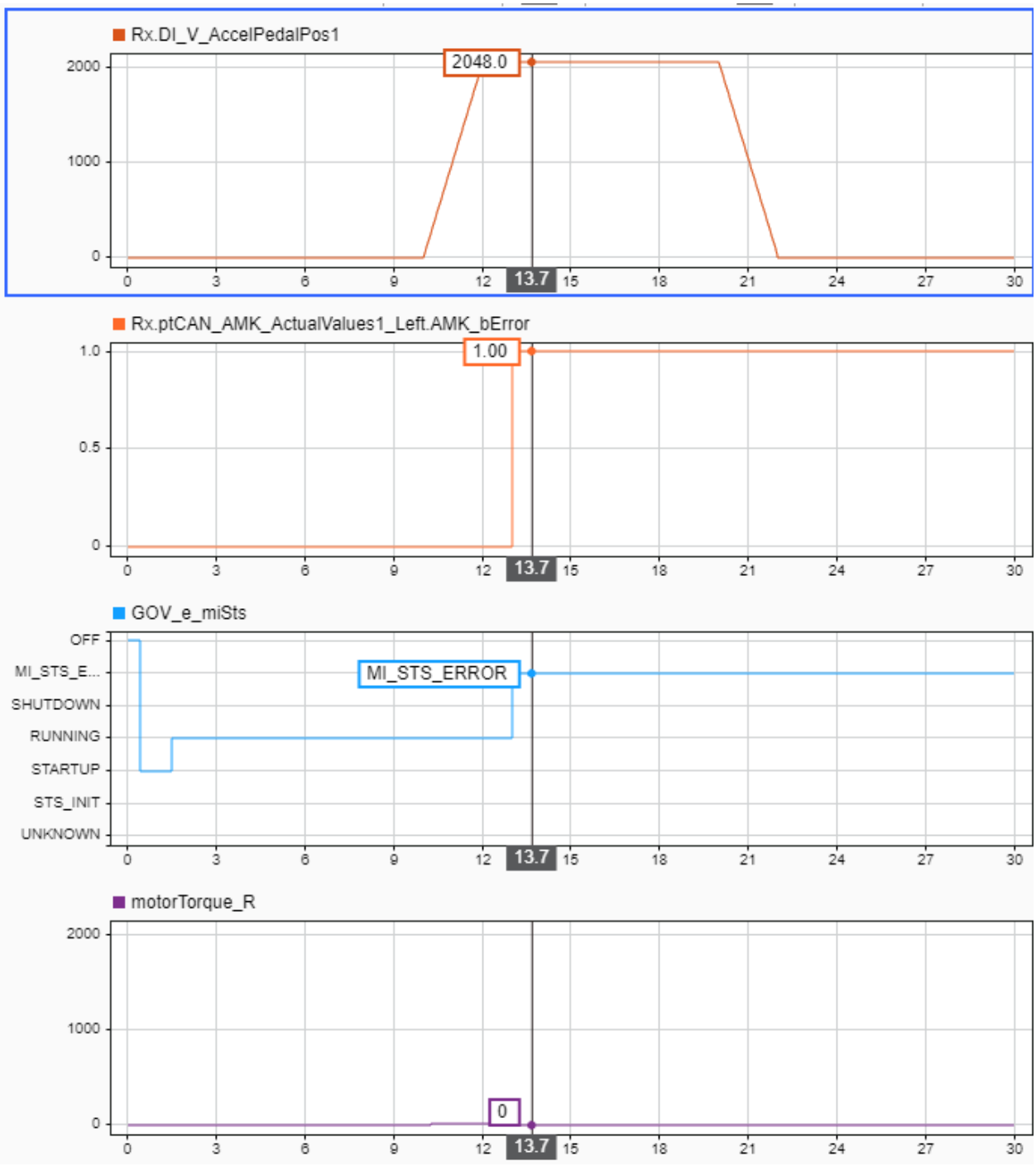

Figure 22: ST_VCS2 Simulation Results

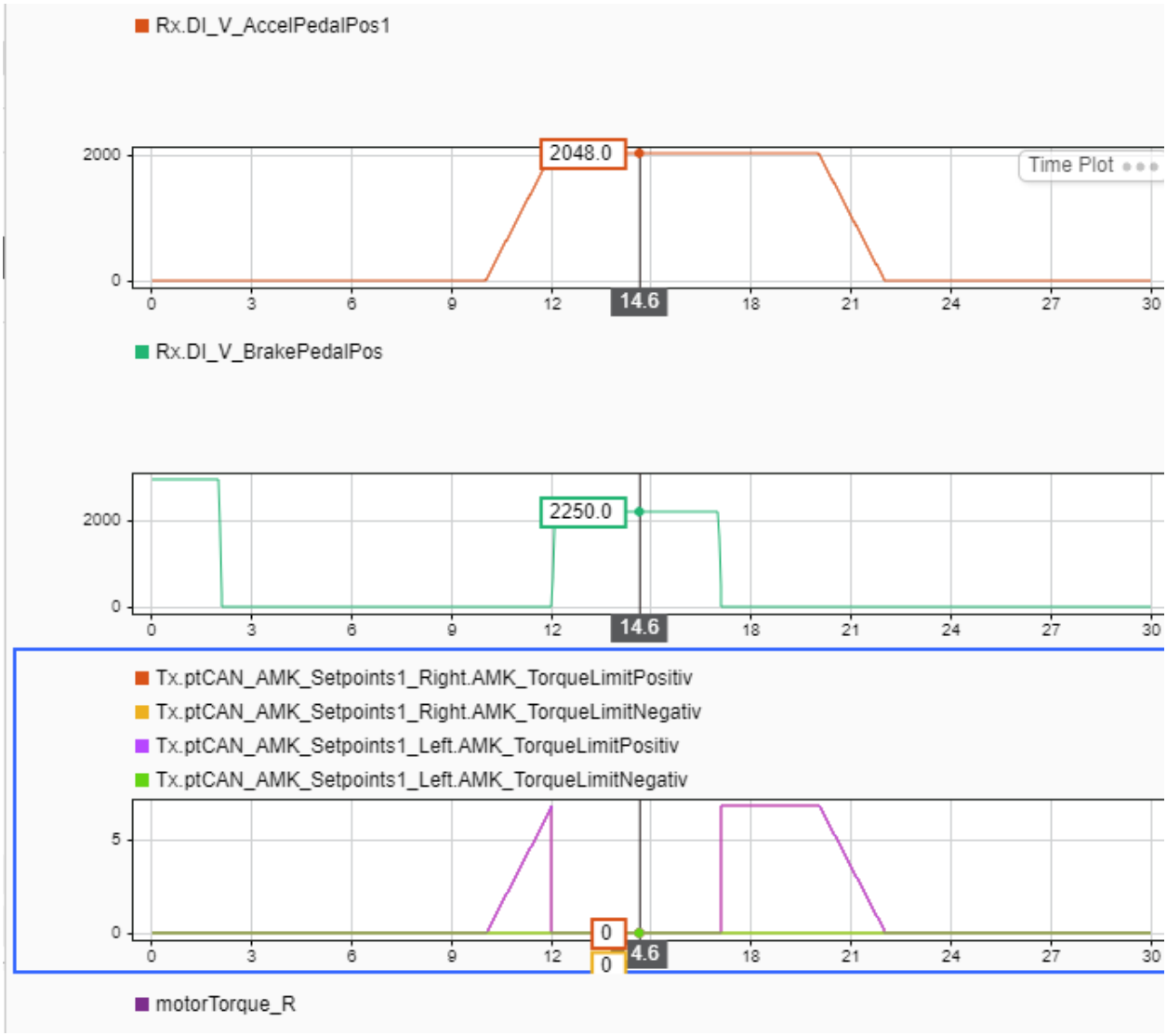

Figure 23: ST_VCS3 Simulation Results

5 Changes Due to Testing

The process of validation (as well as instructor questioning during the PoC and rev0 demos) has highlighted the need to add control logic to handle various edge cases, such as errors in driver input sensors. This was implemented in the Driver Interface module, in the form of sub-modules which monitor each potentiometer reading for error states. When the Driver Interface detects any such errors, this error state is reported to the rest of the control system, and the motor torque request is set to 0 for the vehicle to coast down.

6 Automated Testing

It was not in the project scope or timeline to develop automated test tools for model-based development. Testing was carried out by building manual simulation environments as described in the VnV Plan.

7 Trace to Requirements

Requirement	Test Case
G1	UT_G1, UT_G2, ST_VCS1, ST_VCS2, ST_VCS3
G2	UT_G1, UT_G2, ST_VCS1, ST_VCS2, ST_VCS3
G3	UT_G1, ST_VCS1
G4	UT_G1, ST_VCS1
G5	UT_G1, ST_VCS1
G6	UT_G1, ST_VCS1
G7	UT_G2, ST_VCS2

Requirement	Test Case
DI1	UT_DI1, UT_DI2, UT_DI3, ST_VCS1, ST_VCS2, ST_VCS3
DI2	UT_DI1, UT_DI2, UT_DI3, ST_VCS1, ST_VCS2, ST_VCS3
DI3	UT_DI1, UT_DI2, UT_DI3, ST_VCS1, ST_VCS2, ST_VCS3
DI4	UT_DI1, ST_VCS1
DI5	UT_DI1, UT_DI3, ST_VCS1
DI6	UT_DI2
DI7	UT_DI2
DI8	UT_DI1, UT_DI2, UT_DI3, ST_VCS1, ST_VCS2, ST_VCS3
DI9	UT_DI2
DI10	UT_DI3

Requirement	Test Case
BM1	UT_BM1, UT_BM2, ST_VCS1, ST_VCS2, ST_VCS3
BM2	UT_BM1, UT_BM2, ST_VCS1, ST_VCS2
BM3	UT_BM1, UT_BM2, ST_VCS1, ST_VCS2

Requirement	Test Case
MI1	UT_MI1, UT_MI2, UT_MI3, ST_VCS1, ST_VCS2, ST_VCS3
MI2	UT_MI1, UT_MI2, UT_MI3, ST_VCS1, ST_VCS2, ST_VCS3
MI3	UT_MI1, UT_MI2, UT_MI3, ST_VCS1, ST_VCS2, ST_VCS3
MI4	UT_MI1, ST_VCS1
MI5	UT_MI2, UT_MI3
MI6	-
MI7	UT_MI1, ST_VCS1
MI8	UT_MI3

Requirement	Test Case
VD1	UT_VD1, UT_VD2, ST_VCS1, ST_VCS2, ST_VCS3
VD2	UT_VD1, UT_VD2, ST_VCS1, ST_VCS2, ST_VCS3
VD3	- *
VD4	- *
VD5	- **
VD6	UT_VD2
VD7	UT_VD1, ST_VCS3
VD8	UT_VD1, ST_VCS3

*met by default since the control system's target motor speed and torque are saturated below 0 (never go below 0)
**out of scope for project; no traction control, stability program, or torque vectoring

Requirement	Test Case
NFR1	NFT1
NFR2	NFT2
NFR3	NFT3
NFR4	NFT4
NFR5	NFT5
NFR6	NFT6
NFR7	NFT7

8 Code Coverage Metrics

The concept of code coverage is not applicable in model-based design.

Appendix — Reflection

The testing procedure (simulation environments) outlined in the VnV Plan, for system-level and unit tests, was adhered to fairly closely. This can be seen in the unit test and system test environments above, where module/system inputs are simulated using plant models, or supplied from manual signal building tools, and outputs are inspected using signal visualization tools.

Despite this, in terms of specific test cases, we found our VnV Plan to be inadequate for testing minimum-level functionality of the control system. This is due primarily to its lack of consideration of unit tests, as this was early-stage of the project well before Design Documentation. Moreover, as our team honed our project's scope based on continual feedback and changing requirements from McMaster Formula Electric, the control system had major functionality cuts - most notably, the removal of the cooling control subsystem and battery management system (Formula will be using an off-the-shelf BMS) - as well as architecture design changes following early design discussions, such as a re-design of the "torque path" (any and all control logic between driver input and motor command output; in our control system, this was reflected in the Driver Interface -> Vehicle Dynamnics -> Motor Interface module flow).

In the future, we would know to better manage interaction with our "user" - Mac Formula Electric. Discussion of specific control system topics was fragmented across time and across team members - in hindsight, it would've been ideal to call a team-lead "all-hands" in mid-Fall to review the control system's intended role and function on the vehicle, and how this relates to each team-lead's system.