NIKOLA®

4 Advantages of making Esoteric HIL Testing Accessible

Avinash Divecha, Nikola Corporation

(He/Him)

Pratik Mahamuni, Nikola Corporation

(He/Him)

MathWorks
AUTOMOTIVE CONFERENCE 2024

4 ADVANTAGES OF MAKING ESOTERIC HIL TESTING ACCESSIBLE

NIKOLA®

NIKOLA

TRE® BEV

COMMERCIAL DELIVERIES
Q2 2022

- 645 HP / 480 kW - Continuous Power
- 90 Minutes – Charge Time[1]
- Up to 330-mile zero-emission range[2]

[1] *80% charge at 350 kW*
[2] *Range estimate was calculated using data obtained from Nikola proving grounds testing, real-world vehicle operation, and computational-based engineering and validation tools. Actual range will vary based on several factors including use case, vehicle characteristics, driver behavior, and environmental conditions.*
Specifications subject to change

TRE® FCEV

COMMERCIAL DELIVERIES
Q4 2023

- 536 HP / 400 kW - Continuous Power
- ~20 Minutes - Refuel Time[1]
- Up to 500-mile zero-emission range[2]

[1] *Based on vehicle capability. Actual refuel time will vary based on characteristics of the hydrogen fueling location, including fueling hardware and software protocol, fuel quantity, and fueling conditions., driver*

[2] *Range estimate was calculated using data obtained from Nikola proving grounds testing, real-world vehicle operation, and computational-based engineering and validation tools. Actual range will vary based on several factors including use case, vehicle characteristics, driver behavior, and environmental conditions.*

HYLA FUELING SOLUTIONS

MODULAR FUELING

- Modular Fueler Program - fastest way to establish dispensing capacity
- Each unit can serve multiple vehicles/day, depending on fueling cadence and usage
- Can be located at greenfield sites or behind-the-fence (BTF) at customer locations

OPEN-ACCESS FUELING

- Medium-term method to address demand
- Development of fixed fueling infrastructure enables additional growth and expansion

BEHIND-THE-FENCE FUELING

- Fixed BTF fueling stations tailored to the customer's use case, optimizing vehicle movements and uptime
- Modular fuelers can also be deployed in BTF capacity

PRESENTERS

AVINASH DIVECHA

- Staff Engineer, Product Architecture
  - Responsible for development of Vehicle Functions
  - Introduction of new features
  - Previously led Performance Analysis and Validation
- Worked at Cummins in Indiana and China
  - Development of Vehicle Models for MIL & HIL
  - Focus on ICE, hybrid and electric vehicles
  - Closed-loop systems model dev. & Architecture simulation
- Education:
  - Bachelors Mechanical Engineering (Univ of Mumbai)
  - Masters Mechanical Engineering (Ohio State University)

PRATIK MAHAMUNI

- Lead Engineer Vehicle Performance Integration at Nikola
  - Responsible for BEV and FCEV HIL testing
  - MIL / HIL model development
  - Engineering Tools Development
- Previously worked at passenger automotive companies
  - Focus on virtual engineering
  - Fuel economy / electric range expert
  - Decided to work on Class 8 trucks as a new challenge
- Education:
  - Bachelors Mechanical Engineering
  - Masters Mechanical Engineering (Michigan Tech)
  - MBA (Oakland University)

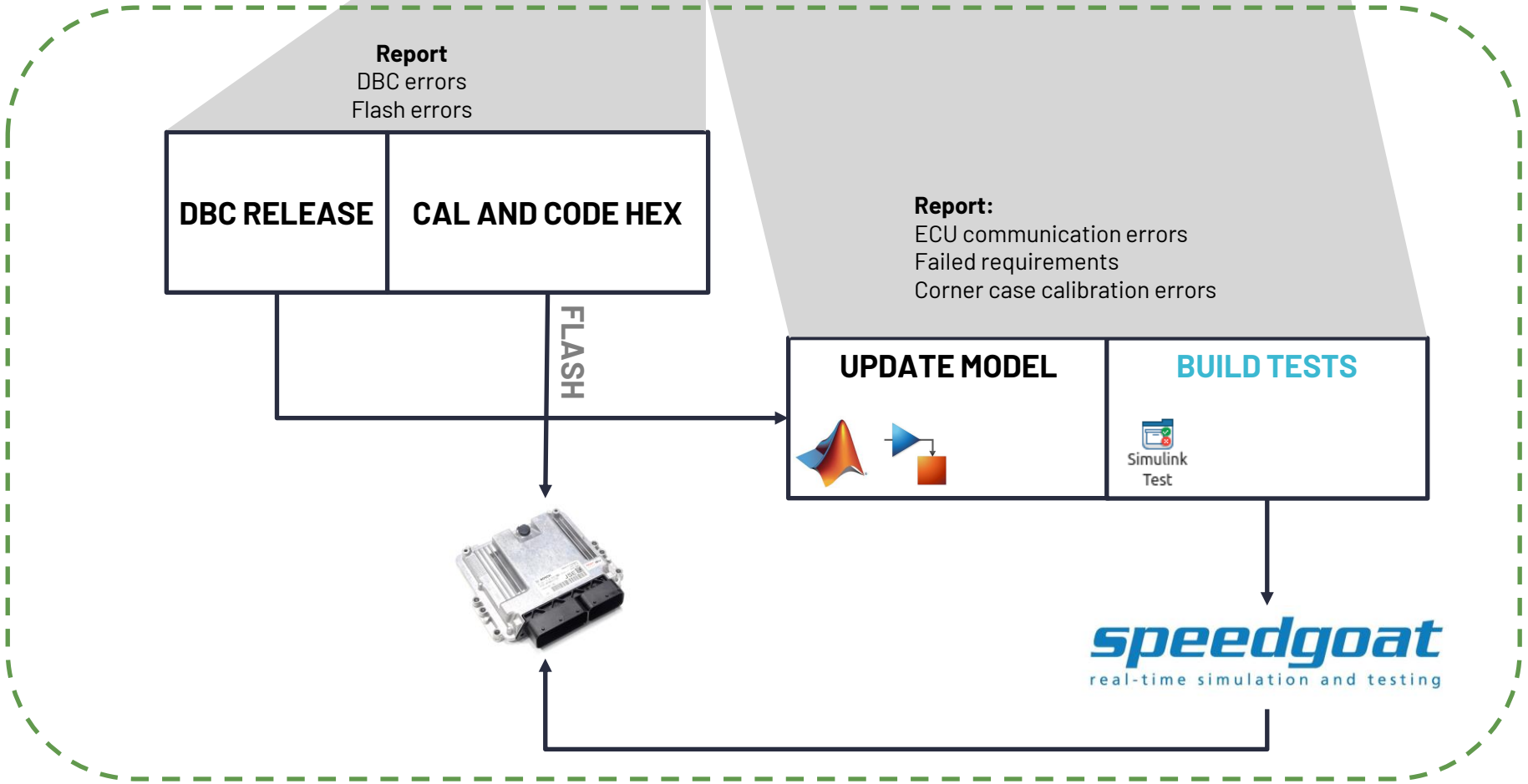
AGENDA

- Software release and testing
- HIL setup & model update
- Why is HIL testing esoteric?
- Fostering accessibility
- Our solution
- Benefits of making HIL testing accessible
- Future work

NIKOLA

SOFTWARE RELEASE AND TESTING

SOFTWARE RELEASE CADENCE



Today's focus

- HIL tests used for:
  - Software verification
  - Requirements testing
  - Simple calibration testing

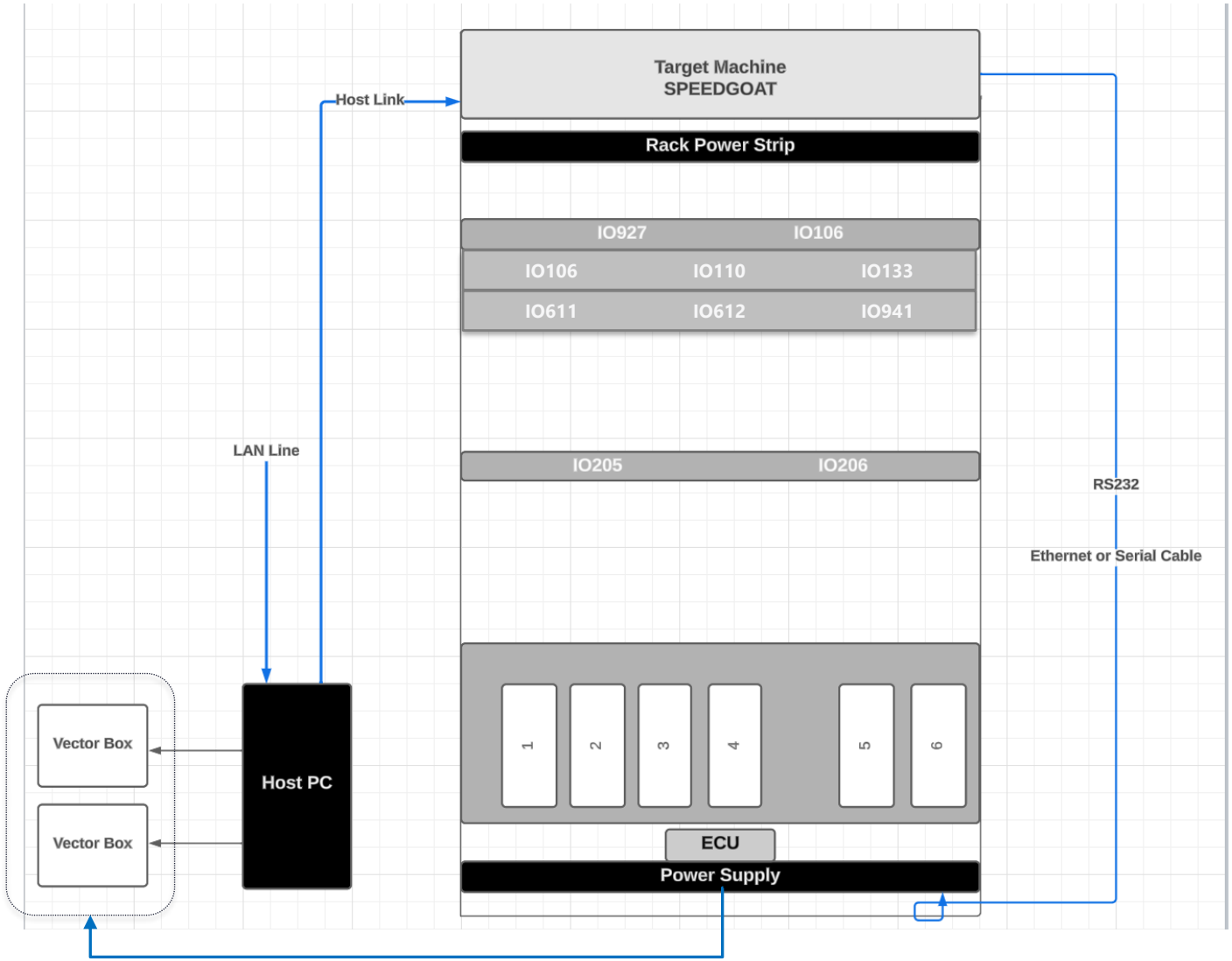
- MIL setup with component models:
  - Battery
  - Inverter
  - Vehicle Dynamics
  - Driver etc.

- MIL tests used for:
  - Drive cycle scenarios
  - Detailed calibration testing
  - Corner case / extreme conditions testing

NIKOLA

HIL SETUP

- Real time Speedgoat target machine
- I/O cards that interface with Speedgoat and ECUs
- ECU with a power supply
  - Power supply can be controlled through Speedgoat
- Host PC which runs MATLAB and Simulink
  - Users of the HIL-UI will book time on this PC
- Vector boxes for redundant data recording



NIKOLA

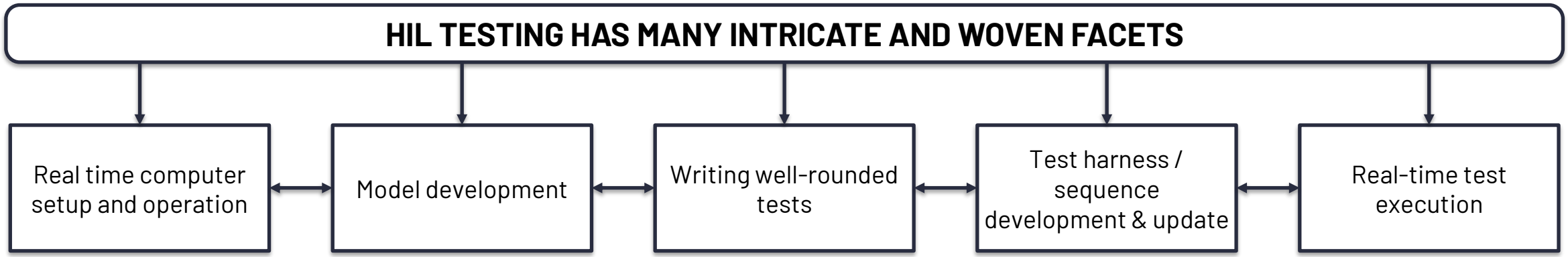
MODEL UPDATE PIPELINE

These functions can be called with ECU specific arguments

```
% Get ecu and tre configuration if not already provided
[~, dbcArray] = CreateCANMsgSignalDataBase(ecu, ...
    'Configuration', configuration, 'TreCommsPath', treCommsPath);

dbcArray = rearrangeCANMap(ecu, configuration, dbcArray, defaultCANMap);
ioCards = rearrangeIOCards(ecu, configuration, defaultIOMap);

dbcArrayFullPath = getDBCArrayPaths(treCommsPath, configuration,...
    dbcArray);

% Generate Simulink CAN blocks
disp(['Generating CAN Rx and Tx Models.' newline]);
GenerateNetworkFromDBC.CreateCANModel(dbcArrayFullPath, ecu, ioCards, ...
    'modelName', mainModel);

% Create CAN_Tx_Routing Subsystems
disp(['Generating CAN_Tx_Routing Subsystems.' newline]);
createCANTxBus('Configuration', configuration, 'TreCommsPath', treCommsPath ...
    'ECU', ecu, 'MainModelName', mainModel);

% Refresh Rx Subsystems
disp(['Updating CAN_Rx Subsystems in the main model' newline]);
refreshCANRxSubsystems(regexprep(dbcArray, '.dbc', '_Rx'), mainModel);

% Generate Simulink LIN Blocks
if (isequal(ecu, 'FCM') && contains(configuration, 'FCEV')) || ...
        isequal(ecu, 'CGW')
    disp('Skipping LIN creation for this ECU');
else
    disp(['Creating LIN Subsystem.' newline]);
    CreateLinBlocksFromLdf(ecu, 'Configuration', configuration, ...
        'TreCommsPath', treCommsPath, 'TimeStep', '0.001', ...
        'MainModelName', mainModel);
end
```

```
HIL
├── System Model Development
│   ├── Board IO Automation
│   │   └── createBOARD_IO_Routing.m
│   ├── Bus Object Automation
│   │   ├── CreateBusObjects.m
│   │   ├── CreateNonVirtualBus.m
│   │   └── reset2Virtual.m
│   ├── CANAnalyzer_Py
│   │   ├── CAN_Start.py
│   │   └── CAN_Stop.py
│   ├── CAN Block Automation
│   │   ├── ApplyCANOverrides.m
│   │   ├── createBusECUCtrlTags.m
│   │   ├── createCANTxBus.m
│   │   ├── refreshCANRxSubsystems.m
│   ├── createHILModel.m
│   ├── Data Dictionary Automation
│   │   └── GenerateDictionaryOverrideM.m
│   ├── +GenerateNetworkFromDBC
│   │   ├── CANMsg_ModelWrapper.m
│   │   ├── CreateCANModel.m
│   │   ├── DbcMsg2CANBlk.m
│   │   ├── ECUCANBusMsgs.m
│   │   ├── GenerateNetworkFromDBC.m
│   │   ├── GetIoModAndChn.m
│   │   ├── GetMsgs.m
│   │   ├── CreateLinBlocksFromLdf.m
│   │   └── sortLINSignals.m
│   └── PositionTools
│       ├── PositionTools_Example.m
│       ├── PositionTools.m
│       ├── RatioCalc.m
│       └── RectPos.m
├── System Model Integration
│   ├── ConnectModels.m
│   └── InitializeValues.m
├── System Model Update
│   ├── Continue_Remaining_CAN.m
│   ├── GetRoutingHistory.m
│   └── ModifyCanBlocksFromDbc.m
└── Tools
    ├── CharacterizeCANBus.m
    ├── CharacterizeCANBus_Sep.m
    ├── CompareCANBlocks.m
    ├── FormatAsMessage.m
    ├── getDataDictionaryDesignData.m
    ├── GetFlatValueStatus.m
    ├── Get LDF and DBC files from GitLab.vbs
    ├── GetNVbusFromBlock.m
    ├── prettyTime.m
```

WHY IS HIL TESTING ESOTERIC?

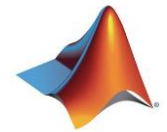


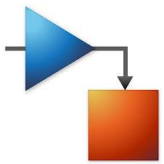
- Controls / Calibration engineers focus on development of features
  - Absolved from understanding HIL system intricacies
- New requirements / features need new test authoring
  - Require HIL test engineers
  - Feedback delayed till tests are developed
  - Resource constraints could halt / delay test development
  - Extended test regression affects S/W release timing
- Desire ability to test new features with system level dependencies
  - Experience analogous to real world vehicle testing

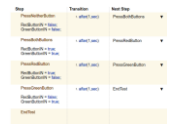
Test development requires:
- MATLAB / Simulink
- Test Sequence
- Test Assessment

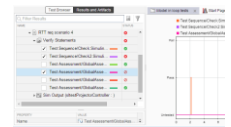
Test execution requires:
- Test Manager
- Speedgoat
- slrtExplorer

Knowledge of current test development processes internal to the company









NIKOLA

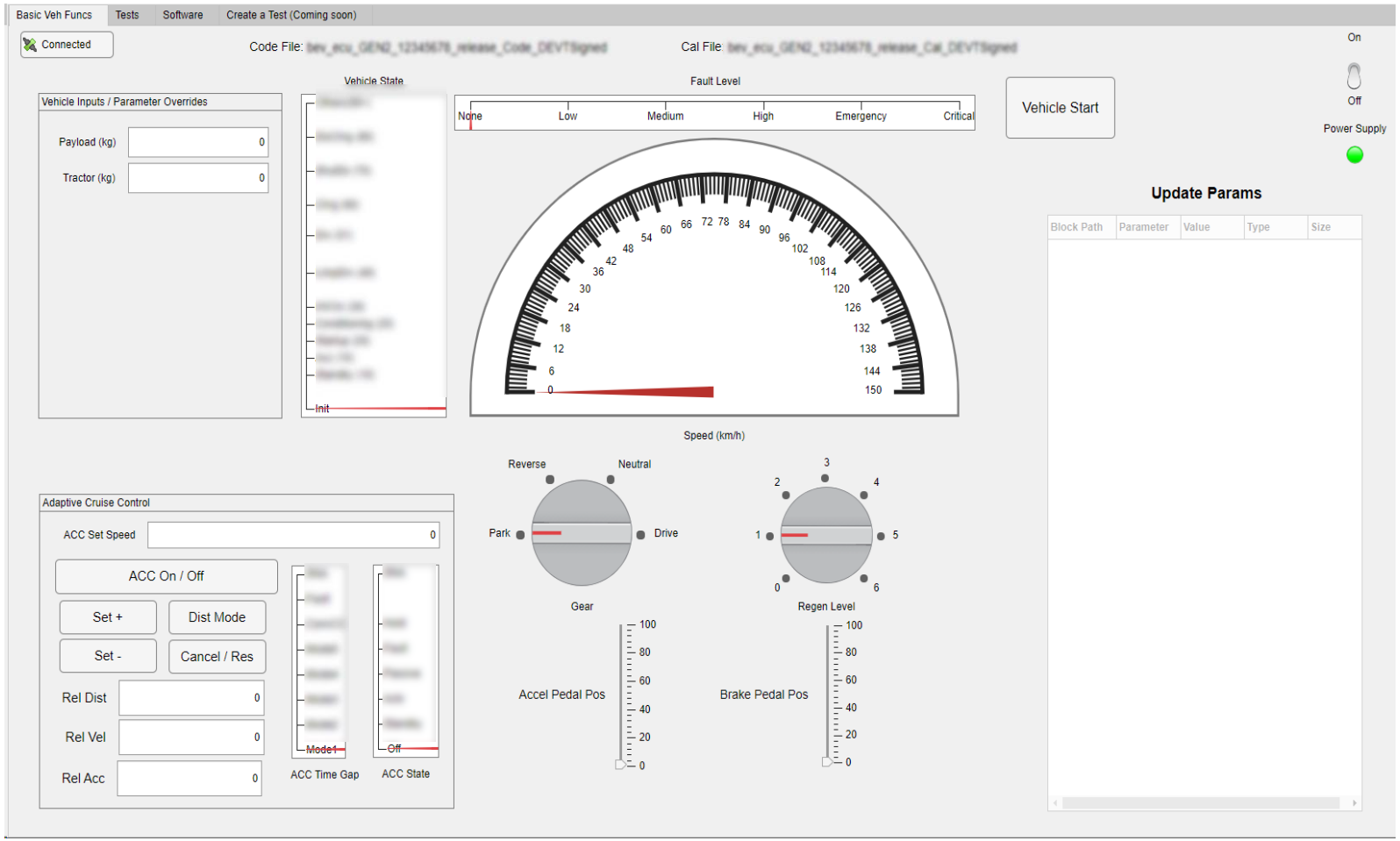
FOSTERING ACCESSIBILITY

Sharing the power of HIL systems – bringing the ECU to you

- Easy access to the HIL bench(es) via an intuitive interface
- Bringing the truck to the user's laptop for easy testing / debugging
- Ability to control frequently used parameters
- Visualize truck behavior / regularly used signals

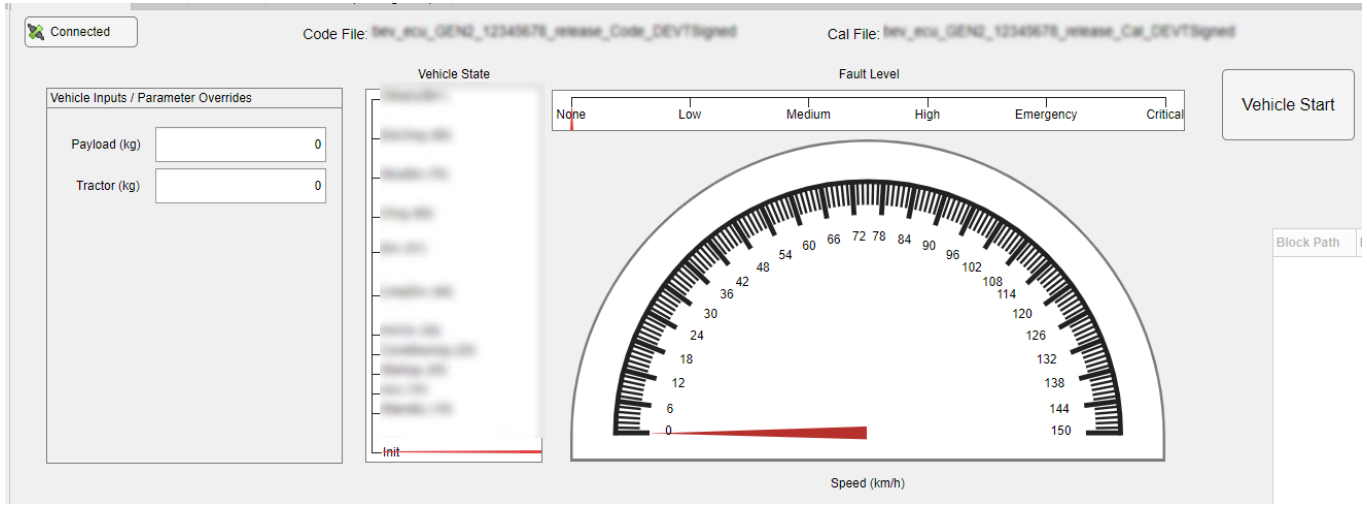
OUR SOLUTION

- User interface created using AppDesigner
- Intuitive interface to execute different scenarios without much training
- Replicate the experience of being in a truck
- Tabs within AppDesigner allow different views & functionality within the same GUI

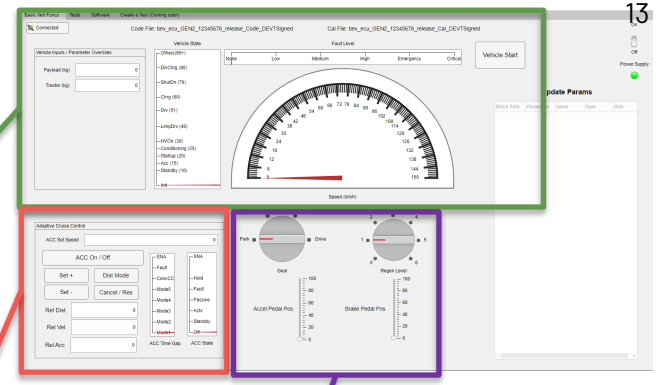


NIKOLA

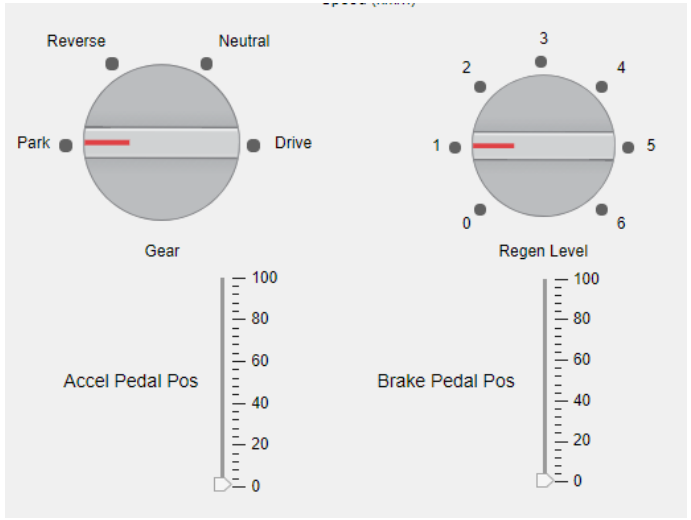
HIL-UI (BASIC VEHICLE FUNCTIONS)



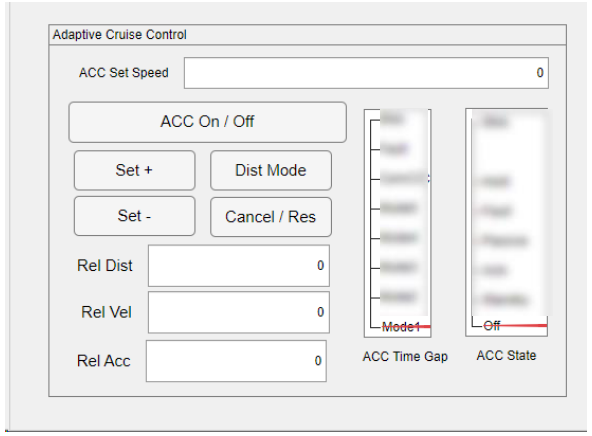
Vehicle speed, vehicle state and fault levels for the ECU are displayed for basic diagnostics using Gauges and Text Boxes



Fine control over gear position, regen levels and accelerator / brake pedal percents using Knobs

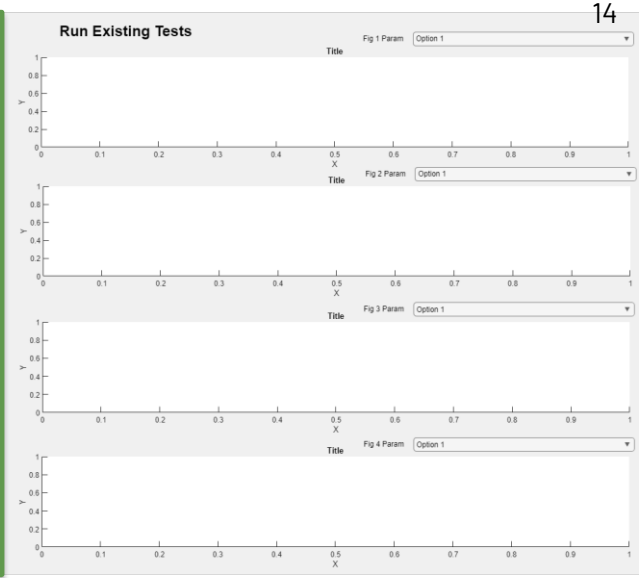


Adaptive cruise control settings and monitoring can be used for controlling ACC behavior and test the ECU in conjunction with ACC using Buttons and Text Boxes for Inputs



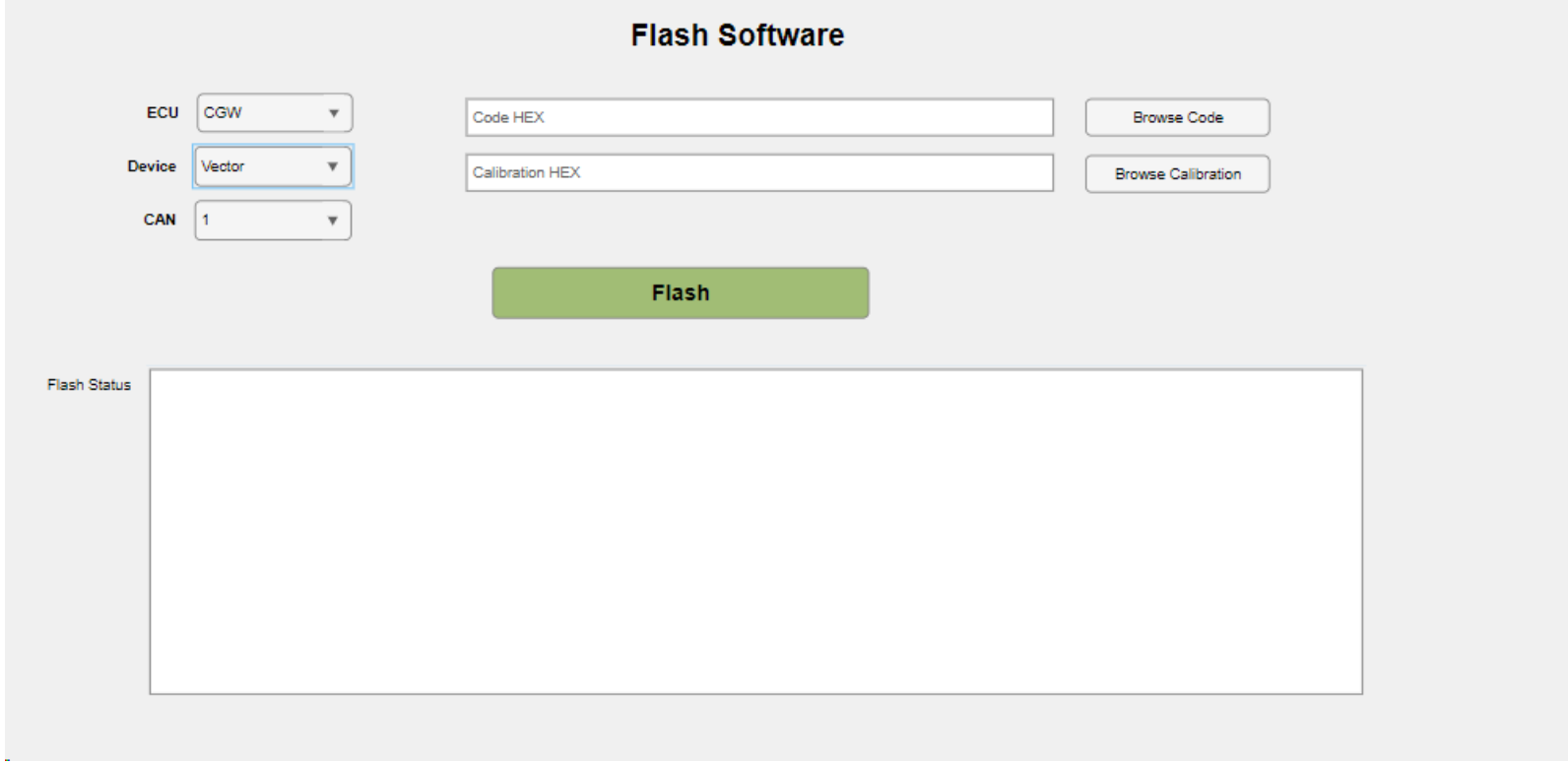
NIKOLA

HIL-UI (PRE-DEFINED TESTS)



- Pre-defined tests that the HIL team has created over the years
  - Regression support tests
  - Torque logic tests for motor / inverter control engineers
  - Functional safety tests for safety and regulatory engineers
  - Fault injection tests for ECU control / embedded systems engineers
  - And more …
- Plots to visualize test progression via CAN communication

HIL-UI (SOFTWARE FLASHING)

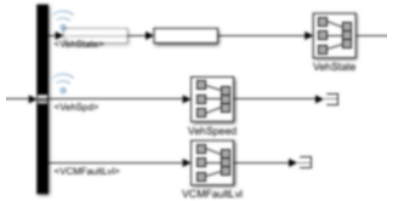


- Ability to flash software to support pre-release software / calibration testing
  - Useful for debugging purposes where a RelDeb software is flashed to diagnose root causes
  - Verification of new features before release software is sent to trucks

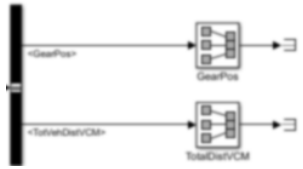
- Users can flash software on the ECU and run a suite of tests without having to contact a HIL engineer

HIL-UI (CODE THAT RUNS BEHIND)

- Instrument Objects help display outputs in real-time
- Outputs from the model are mapped to gauges & text boxes
- slrealtime API is used for
  - Connecting to the target PC
  - Creating an Instrument object for mapping to the UI



OUTPUTS:



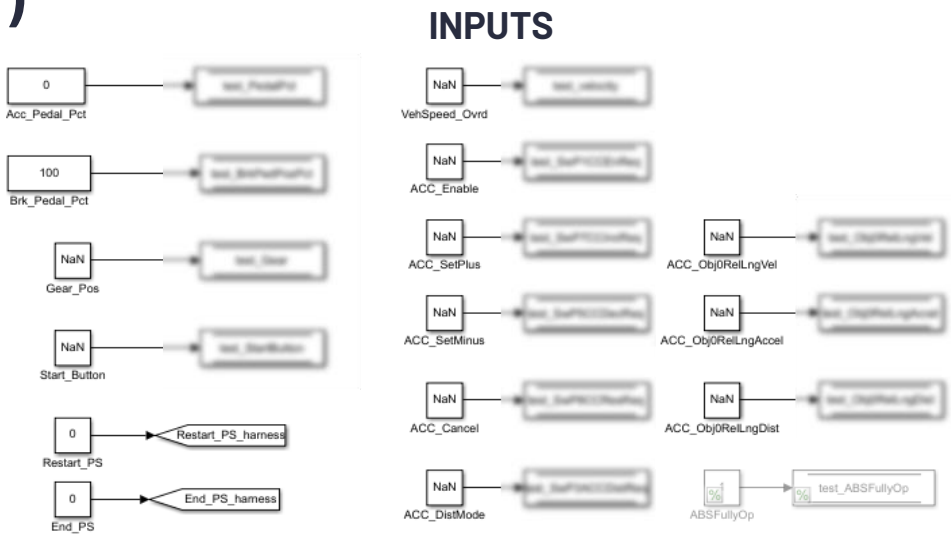
```
% Code that executes after component creation
function startupFcn(app)

    TargetPC= slrealtime('TargetPC1');
      app.tg = TargetPC;
    targetSelector = app.TargetSelector;

    TargetPC= slrealtime('TargetPC1');
    if TargetPC.isConnected
        app.EditField.Value = 'Connected';
        app.EditField.BackgroundColor = [0 1 0];
    else
        TargetPC.disconnect;
        app.EditField.Value = 'Disconnected';
        app.EditField.BackgroundColor = [1 0 0];
    end

    inst = slrealtime.Instrument();
    inst.connectScalar(app.VehicleSpeedGauge,  , 1);
    inst.connectScalar(app.VehSpeedEditField,  , 1);
    inst.connectScalar(app.VehStateGauge,  1);
    inst.connectScalar(app.VehStateEditField,  , 1);
    inst.connectScalar(app.GearPosGauge,  1);
    inst.connectScalar(app.FaultLvlGauge,  , 1);
    inst.connectScalar(app.Odometer,  1);
    inst.connectScalar(app.ACCStateGauge,  1);
    inst.connectScalar(app.ACCSetSpeedEditField,  , 1);
    inst.connectScalar(app.ACCTimeGapModeGauge,  , 1);

    slrtcomp1 = slrealtime.ui.tool.ParameterTuner(app.UIFigure, 'TargetSource', targetSelector);
    slrtcomp1.Component = app.AccelPedalPctSlider;
    slrtcomp1.BlockPath = 'VCM_GUI_Harness/Acc_Pedal_Pct';
    slrtcomp1.ParameterName = 'Value';
      slrtcomp.ConvertToComponent = @app.convToDouble;

    slrtcomp2 = slrealtime.ui.tool.ParameterTuner(app.UIFigure, 'TargetSource', targetSelector);
    slrtcomp2.Component = app.BrakePedalPctSlider;
    slrtcomp2.BlockPath = 'VCM_GUI_Harness/Brk_Pedal_Pct';
    slrtcomp2.ParameterName = 'Value';
```

HIL-UI (CODE THAT RUNS BEHIND)

- Inputs from the model (knobs, switches & text boxes) are mapped to constant blocks

- Changes in values are transmitted to the real-time computer
  - setparam is used to update parameter values in real-time
  - Should not be confused with Simulink's API command – set_param
  - Parameter Tuners to connect sliders to inputs

- AppDesigner auto creates a lot of boiler-plate code for the User Interface
  - Helps manage functionality




```
% Callback function
function GearTempButtonGroupSelectionChanged(app, event)
    selectedButton = app.GearTempButtonGroup.SelectedObject;
    TargetPC= slrealtime('TargetPC1');
    if strcmp(selectedButton.Text,'Park')
        TargetPC.setparam('VCM_GUI_Harness/Gear_Pos','Value',1);
    elseif strcmp(selectedButton.Text,'Reverse')
        TargetPC.setparam('VCM_GUI_Harness/Gear_Pos','Value',2);
    elseif strcmp(selectedButton.Text,'Neutral')
        TargetPC.setparam('VCM_GUI_Harness/Gear_Pos','Value',3);
    elseif strcmp(selectedButton.Text,'Drive')
        TargetPC.setparam('VCM_GUI_Harness/Gear_Pos','Value',4);
    end
end
```

```
slrtcomp1 = slrealtime.ui.tool.ParameterTuner(app.UIFigure, 'TargetSource', targetSelector);
slrtcomp1.Component = app.AccelPedalPctSlider;
slrtcomp1.BlockPath = 'VCM_GUI_Harness/Acc_Pedal_Pct';
slrtcomp1.ParameterName = 'Value';
  slrtcomp.ConvertToComponent = @app.convToDouble;

slrtcomp2 = slrealtime.ui.tool.ParameterTuner(app.UIFigure, 'TargetSource', targetSelector);
slrtcomp2.Component = app.BrakePedalPctSlider;
slrtcomp2.BlockPath = 'VCM_GUI_Harness/Brk_Pedal_Pct';
slrtcomp2.ParameterName = 'Value';
```

BENEFITS OF MAKING HIL TESTING ACCESSIBLE

Increased **HIL Accessibility = Increased Efficiency ➔ STAY AGILE**

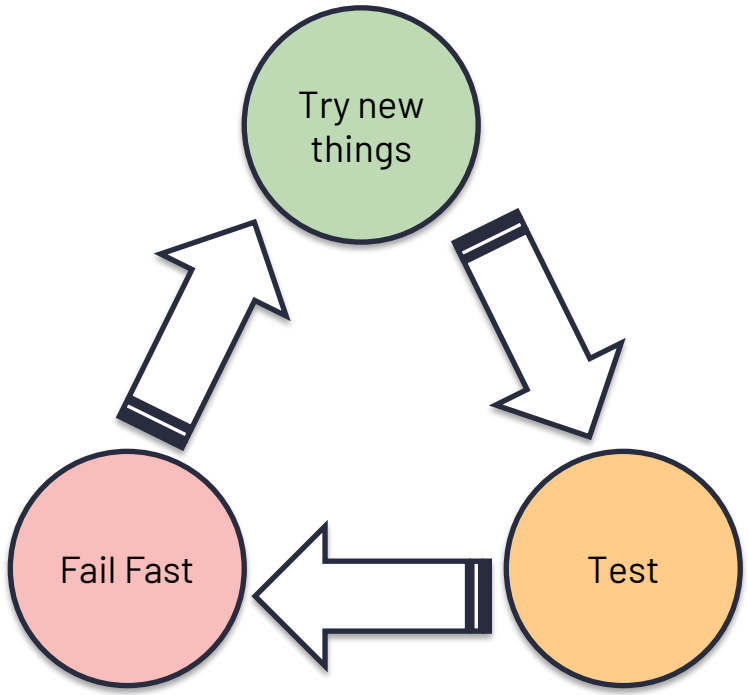
- Reduce development time for new test cases (20-40 minutes) & eliminate necessity of HIL Engineer to accompany HIL Testing

  Time needed between runs with tests with minor modifications
  - *Updating harness (5-10 minutes)*
  - *Building & compiling harness (10-20 minutes)*
  - *Loading harness using slrtExplorer (5-10 minutes)*

- Reduce time between software release, model development and testing

- Easy access to debugging opportunities to reduce track /dyno testing

- Allow engineers to run a slew of pre-defined test cases



FUTURE WORK

- Deploy tool as a Web App, allowing engineers to access the tool remotely without having to log-in to the HIL Bench
- Increasing tool capability to allow users to write pre-defined tests
- Development of User Interfaces to check other vehicle functionality like Lights, Fan, etc


NIKOLA
LAMBI
NIKOLA

Q&A