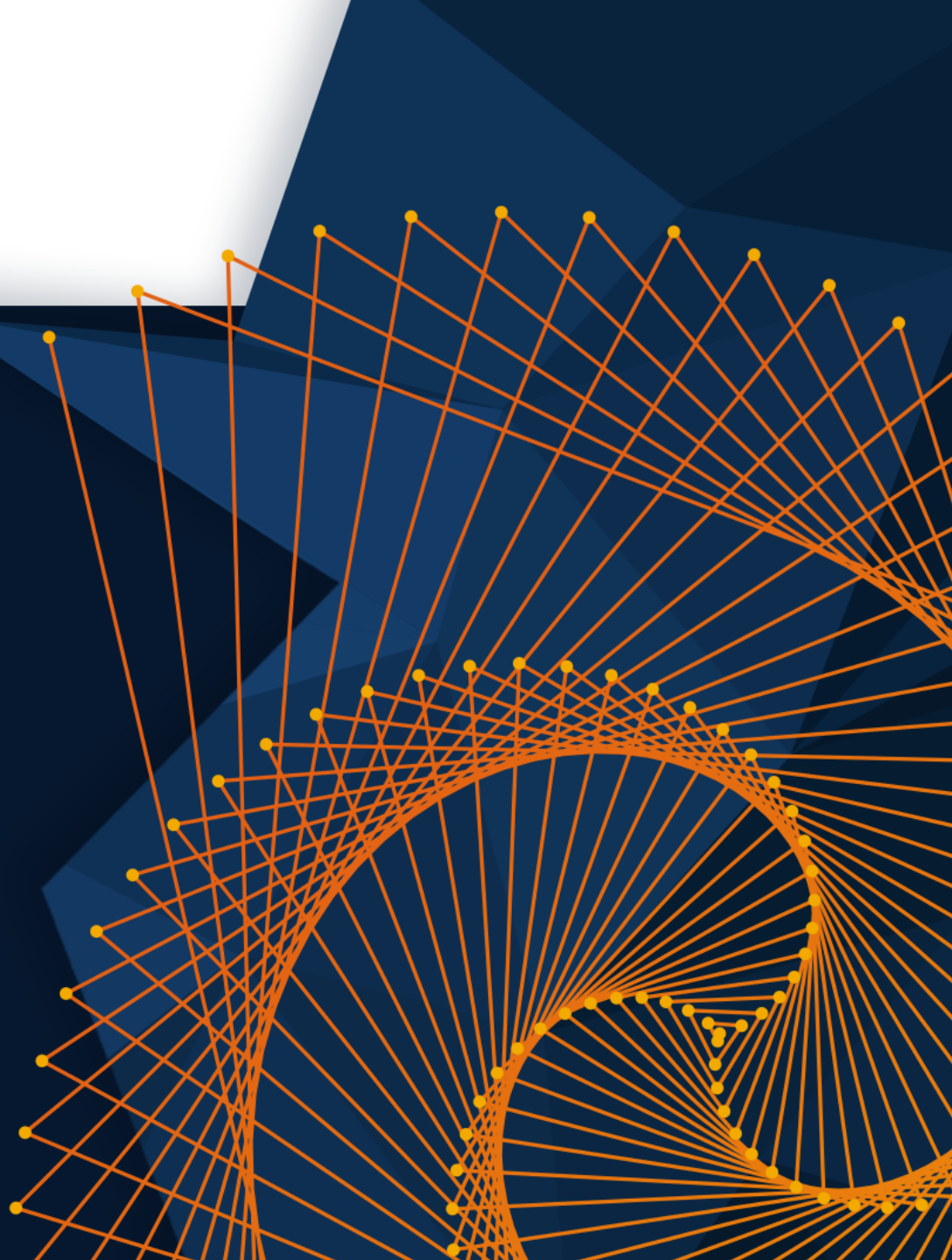
supernal

November 13–14, 2024

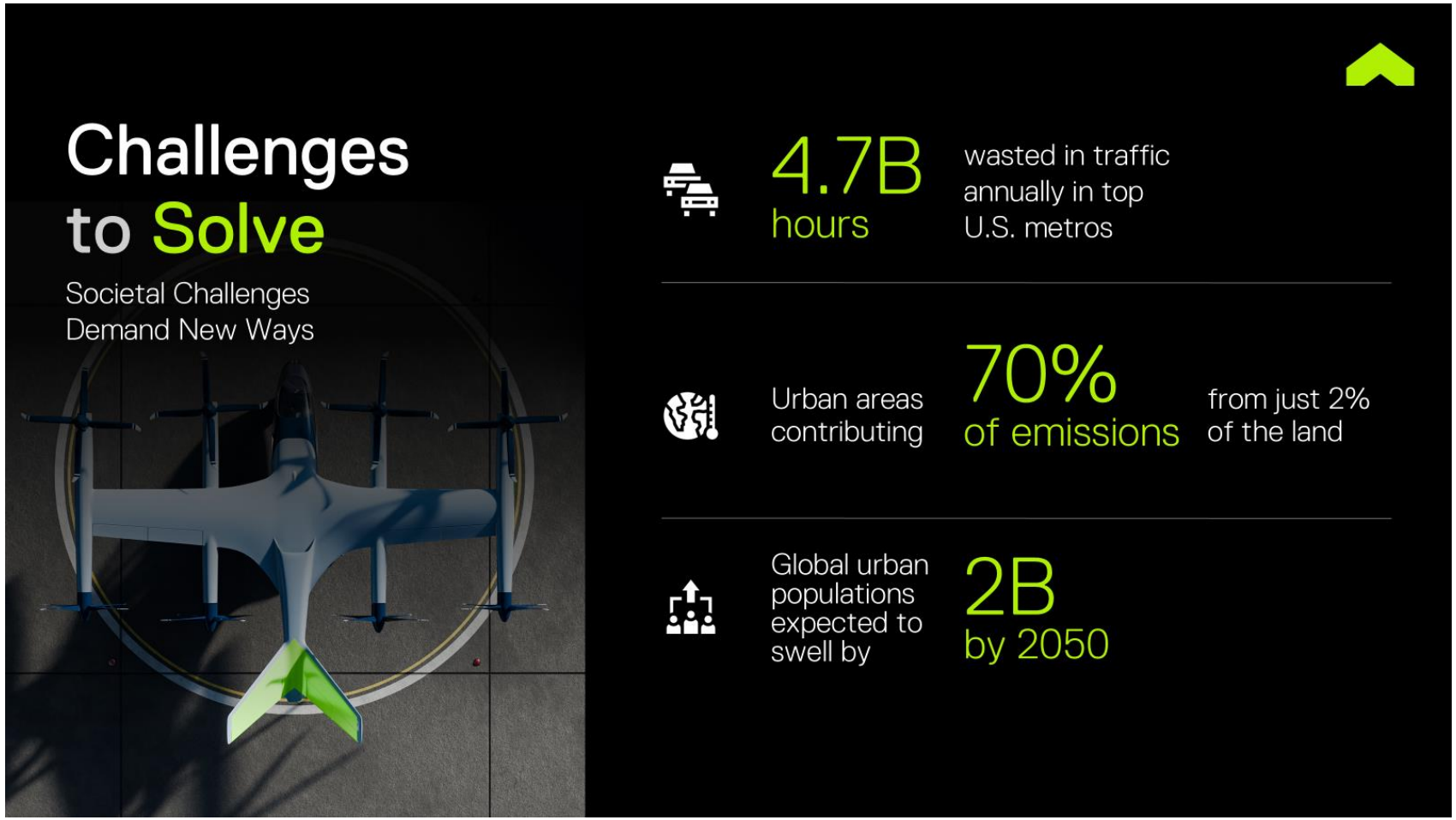
Functional Simulator: A model-based simulation system of systems approach to emulating eVTOL aircraft.

Fahad Khan 퍼하드 칸
Head of Integrated Simulation Systems, Supernal
Cedric Ma, Lead Simulation Engineer, Supernal
Stephen Chung, Lead Simulation Engineer, Supernal

MATLAB EXPO



One of humanity’s biggest challenges in dense urban areas is billions of hours of our lives wasted in traffic every year

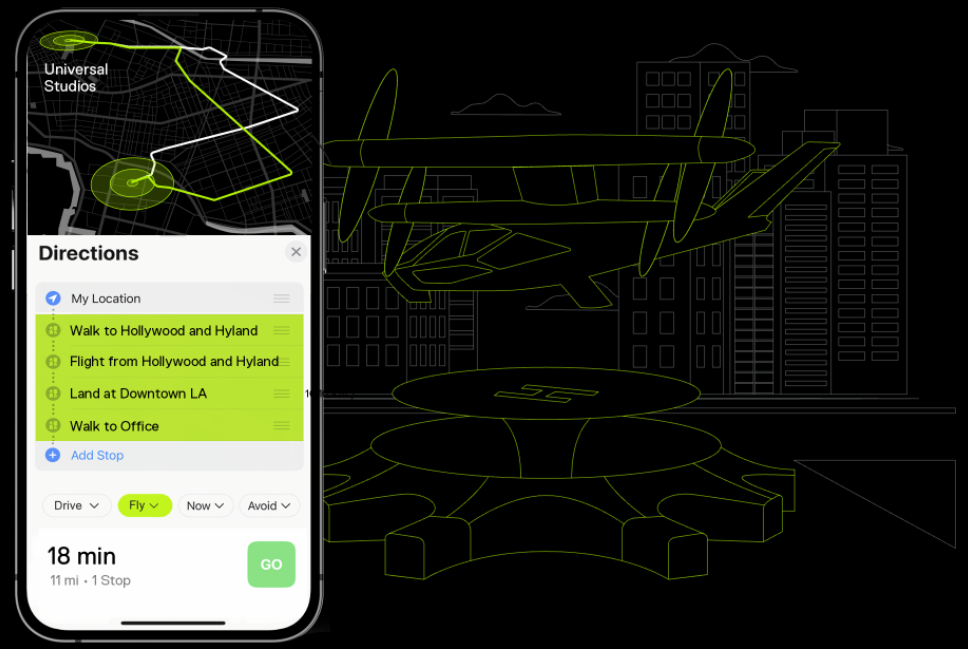


Sources:
World Bank Group
McKinsey & Co
Frontier Group

Advanced Air Mobility presents a promising opportunity to advance humanity and change how we live, move, and connect

AAM: A New Transportation Option

Similar to trains, subways and walking, eVTOL flights will be an **additional transportation option** travelers can choose from when charting their multi-modal journeys



A next-generation aerospace company inside a large conglomerate with a strong history of commitment to advance humanity

Supernal is the embodiment of Hyundai Motor Group's commitment to the future of transportation

We want to build **human-centered cities**

That leverage **innovative technology** advancements

To deliver **affordable and universal** mobility services

HYUNDAI MOTOR GROUP

GENESIS

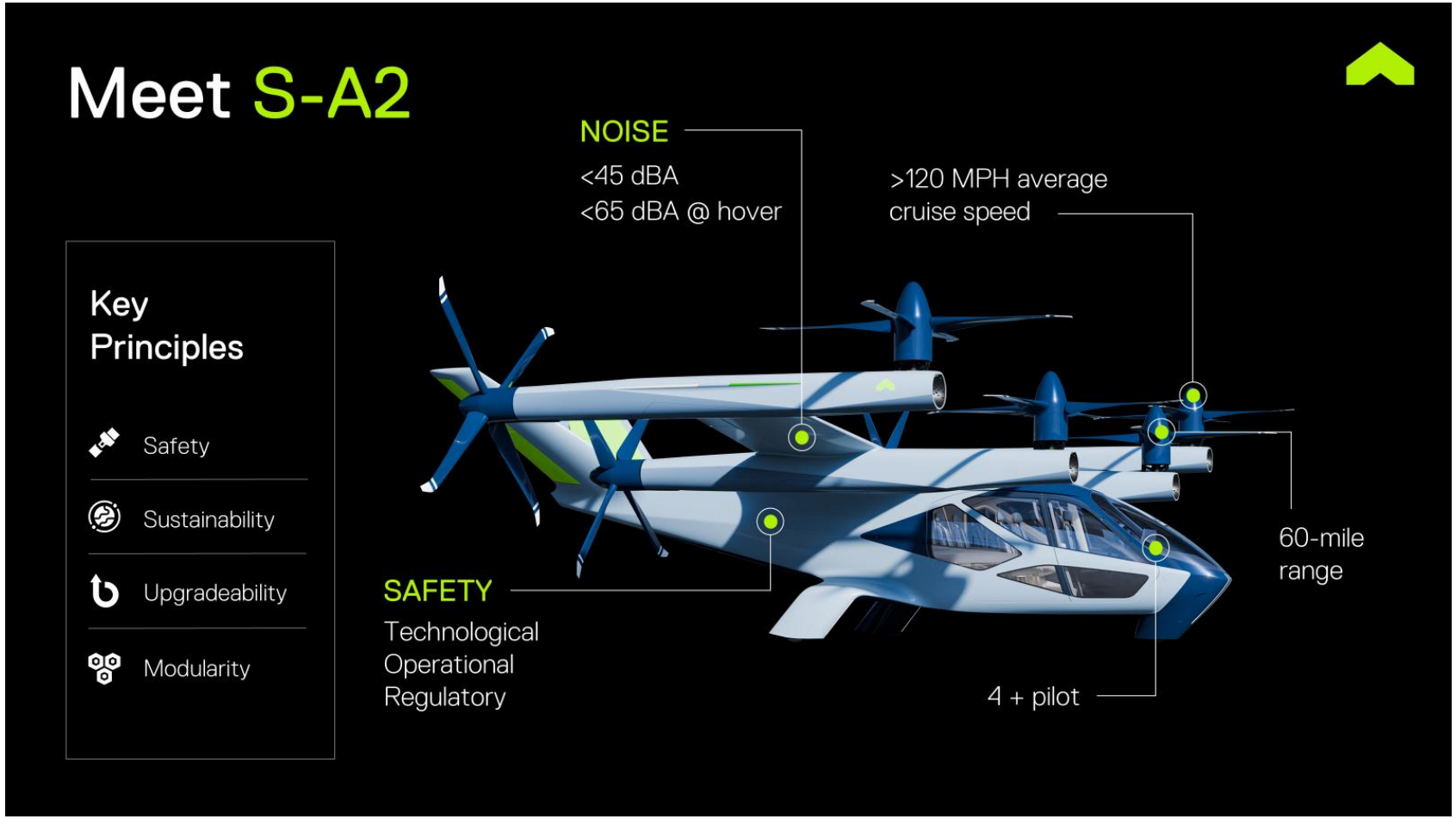
HYUNDAI

KIA

Boston Dynamics

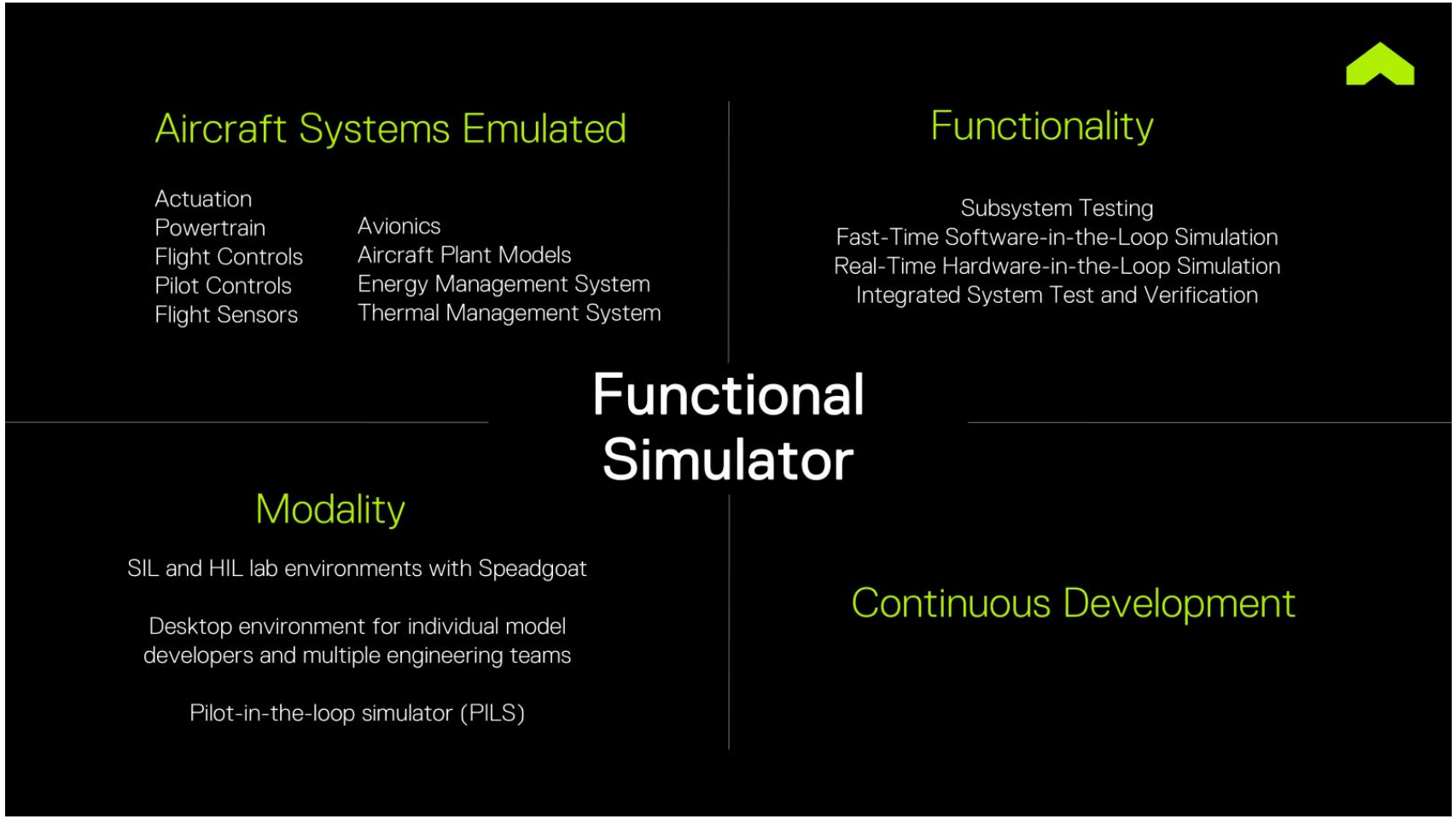
THE AI INSTITUTE

We are building an electric vertical take-off and landing (eVTOL) aircraft and a culture of simulation-driven engineering



We need simulation systems more than ever before as complexity of building an electric aircraft far outpaces that of a non-electric aircraft.

We built Functional Sim to be the integrated simulation system of systems capable of emulating all systems of our aircraft in a variety of technical settings



We built Functional Sim with Simulink and MATLAB product family tools coupled with internal and external tools

Flight Dynamics using **Simulink** and external/internal tools
- Aerospace Blockset
- Runs in “real-time” mode

Simulink, Stateflow & MATLAB functions for most native models
- MATLAB-function is particularly suited to fill gaps for rapid prototyping logical functions

Actuation and Powertrain system hardware model using vendor compiled Simulink model
- Closed model: runnable but not browsable

Interface Modeling using **Speedgoat** to support HWIL use cases

Thermal & Fluid sim using external **multi-physics systems simulation tool**, tied in with Simulink
- Full model needs model reduction to run 1x time

How we use Functional Sim is driven by use-cases, which are tied to our development lifecycle for system maturity

1 Software-in-the-loop Simulation (non-real-time)

Fully simulate aircraft subsystems and plant models in MATLAB Simulink

2 Hardware-in-the-loop Simulation (real-time)

Increase fidelity with Speedgoat stimulating various subsystems software hosted on target hardware

3 Pilot-in-the-loop Simulation (real-time)

Exercise aircraft subsystems with realistic test inputs in a dynamic environment to flesh out potential flight test anomalies

Functional Sim system design considerations

Modular

- Swappable components

Realistic

- High-fidelity models
- Real-time hardware-in-the-loop simulation

Configurable

- Dynamic scenarios
- System and subsystem level configuration

Performant

- Fast-time simulation

Functional Sim's modeling capabilities

Energy Management System

- System Software
- Battery Pack Model
- High/Low-Voltage Systems

Thermal Management System

- System Software
- Thermal Dynamics
- Cooling Loop

Flight Control System

- Control Software
- Flight Dynamics and Landing Gear

Flight Sensor Systems

- Navigation Sensors

Powertrain and Actuation Systems

- Motor and Actuator Models
- Prop/Lift Rotor and Swashplate Models

Functional Sim's modeling capabilities (Cont.)

Pilot Control
- Inceptor
- Cockpit Switches

Scenario Configuration
- Flight Profile
- System and Subsystems Configuration

Fault Modeling
- Subsystems Fault Injection

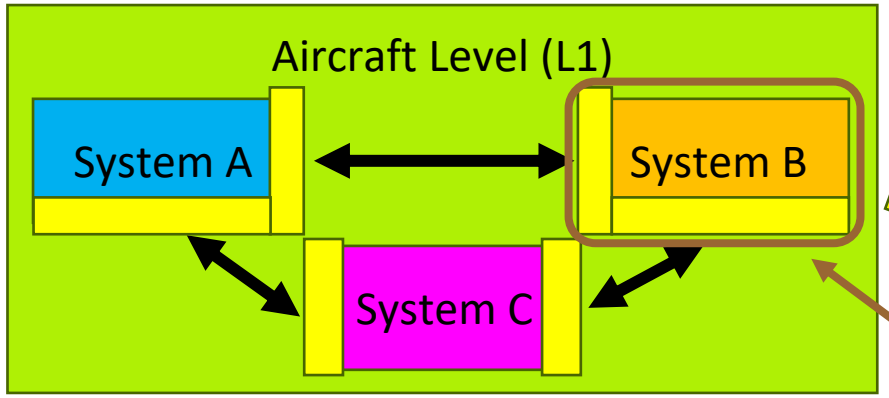
Analytics
- Data Recording and Plots
- Monte-Carlo Simulation

Real-Time Hardware-in-the-Loop Execution
- Speedgoat

Visualization
- Common Image Generator Interface (CIGI)

Systems testing driven by L2 system requirements

- System level test is the testing of individual system at L2 level, i.e. powertrain
  - Testing of multiple such systems in combination is testing at L1 “aircraft level”
- System level tests are against L2 requirements written for that system
  - L2 requirements generally don’t span multiple systems
- Functional Simulator is especially useful for L2 level testing
  - Many L2 tests implicitly require interactions with adjacent systems
    - **Most traditional L2 tests avoid requiring closed loop simulation**
  - Speedgoat interfaces provide wrap-around coverage of “system under test”

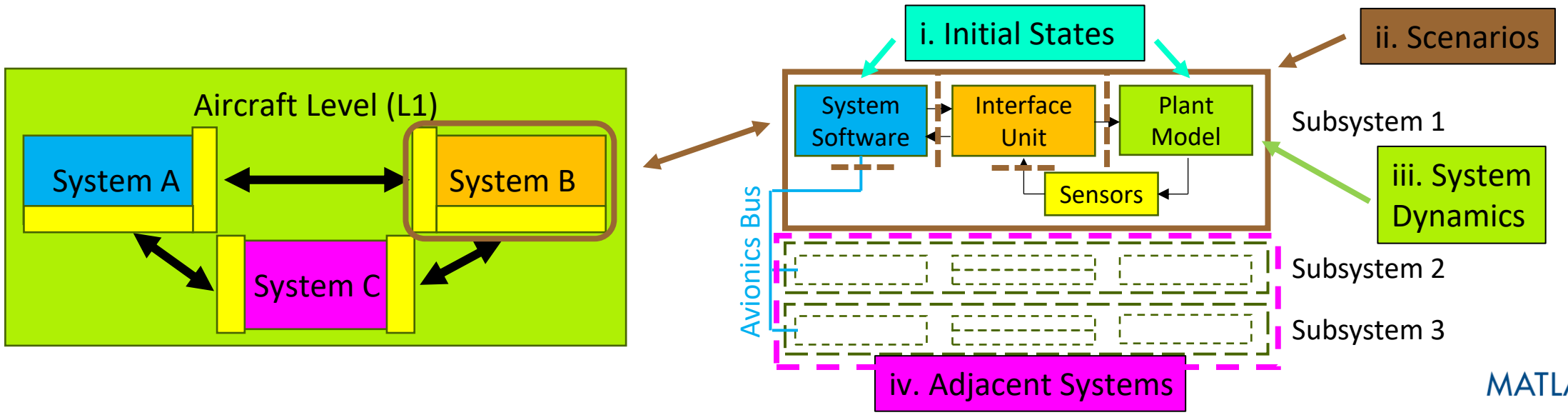


Speedgoat Interface Sim:
1. Provides simulated data traffic to targeted system
2. Verifies output data traffic from targeted system

L2 tests usually only target one system at a time. Adjacent system interactions are either simulated or prescribed (i.e. recorded)

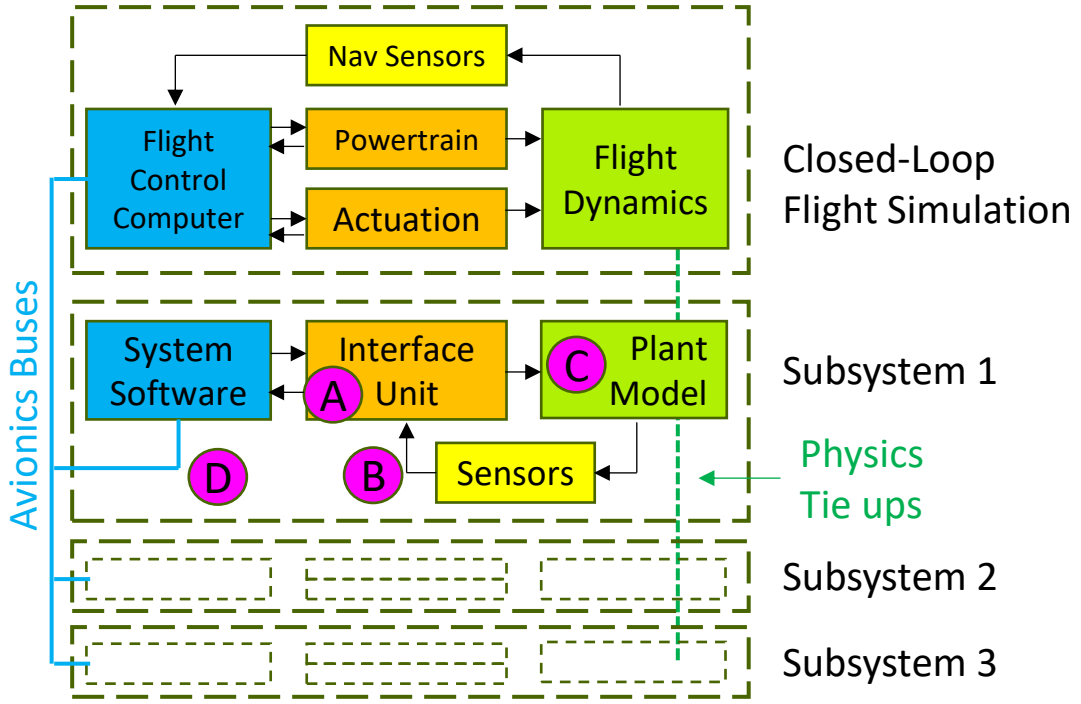
Functional Sim provides wrap-around coverage of “system under test”

1. Hardware interface allow testing of external and intermediate interfaces
2. Modeling of multiple system dynamics and populating interfaces by providing:
   i. Internal initial states for targeted system (start of test)
   ii. Scenario driver for adjacent systems and/or inputs for stimulation
   iii. Physical dynamical response integral to targeted system
   iv. Adjacent systems response in open or closed loop feedback to targeted system

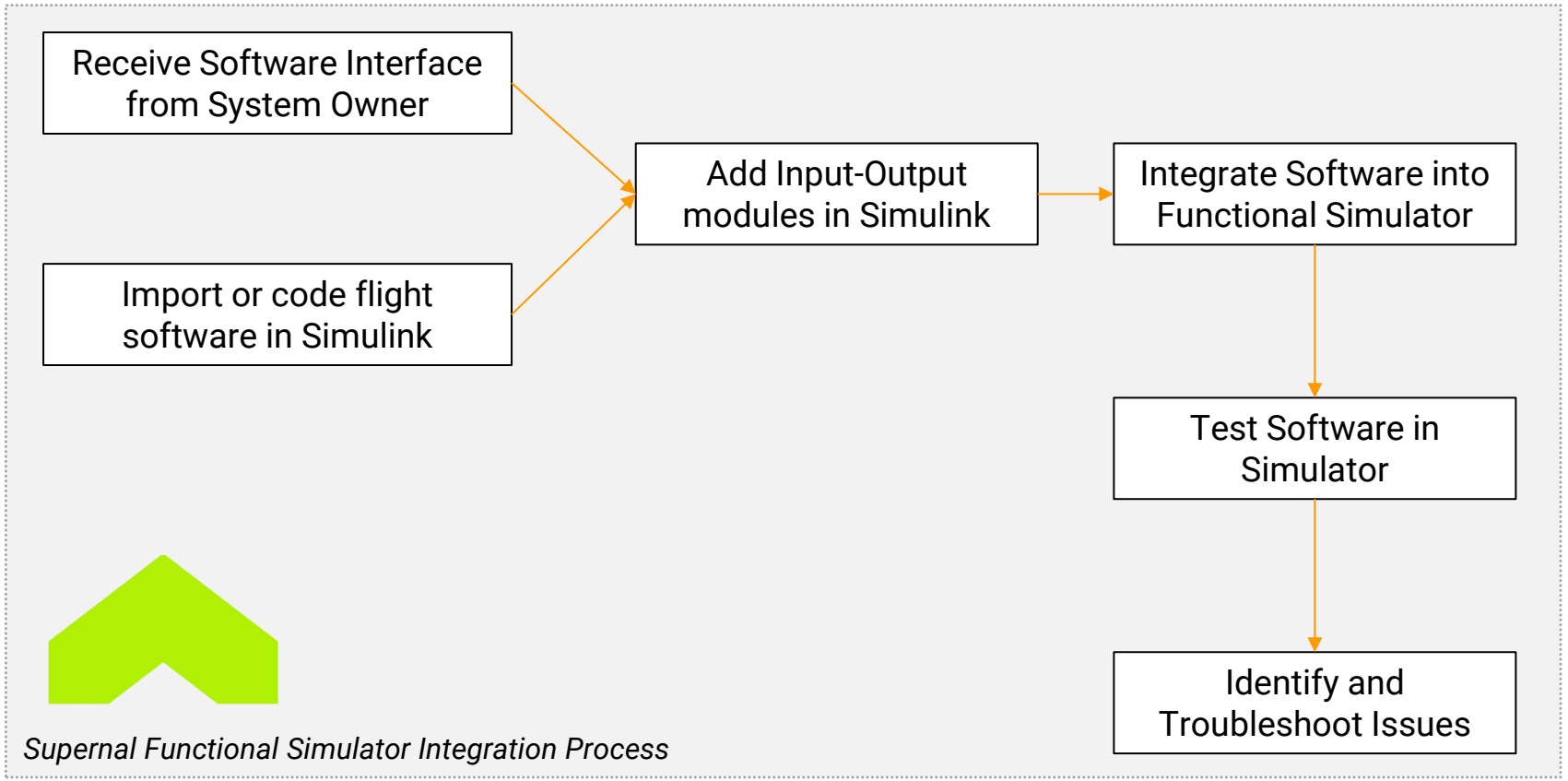


Functional Sim for system testing

- System test comprises a collection of test cases
  - Each test case tests a distinct part of the system
  - Each test case may require a number Functional Sim components
    - Most tests require only a single component
    - Most tests can be done open loop
    - **Closed loop simulation is a potential game changer for system tests**
- Functional Sim components serve to:
  - A. Simulate IO from system software to interface unit
  - B. Simulate sensor output feeding into interface units
  - C. Monitor system command to hardware
  - D. Simulate interfaces between adjacent systems

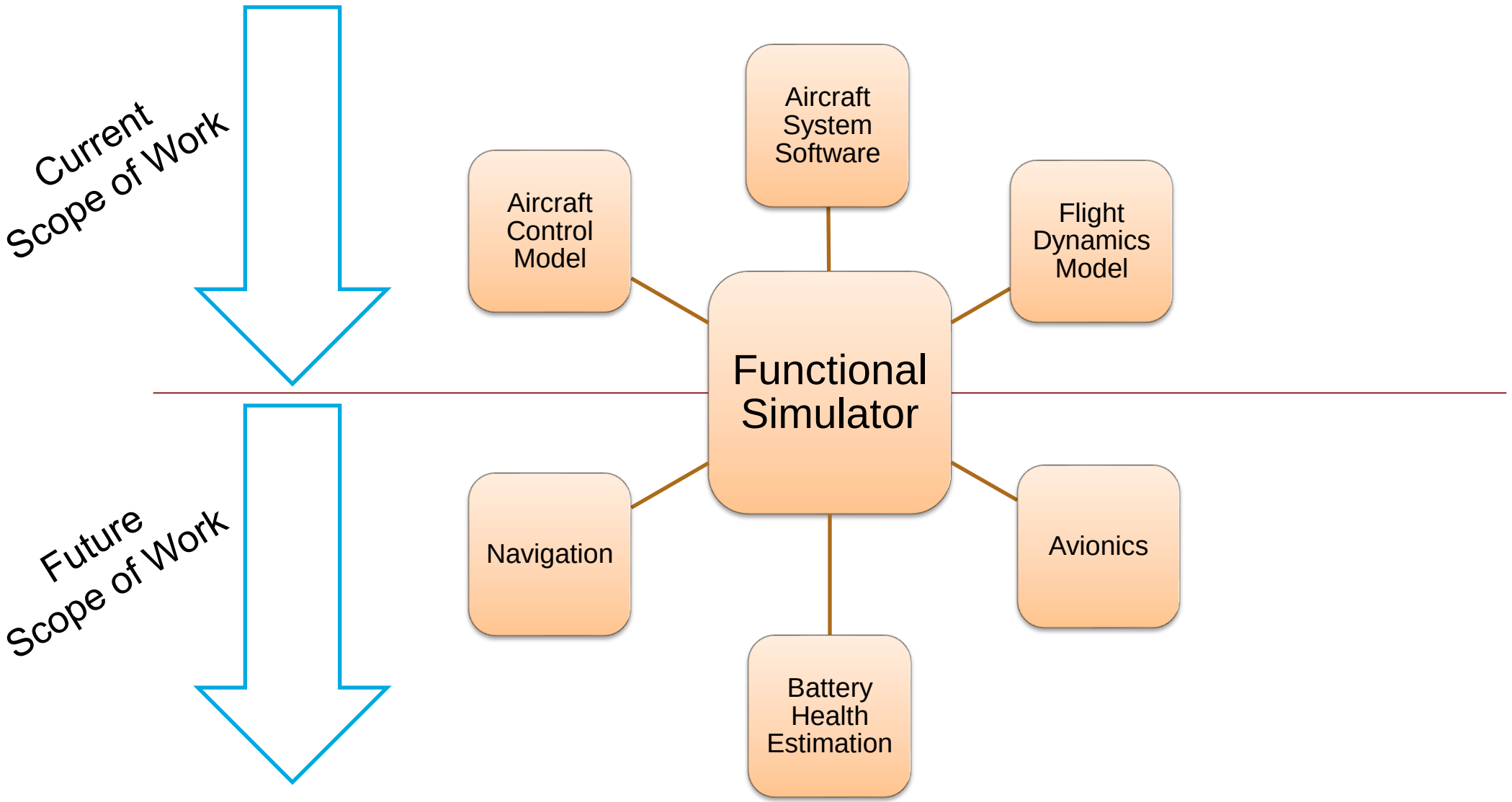


Engineering process through which new modules are added to the Functional Sim



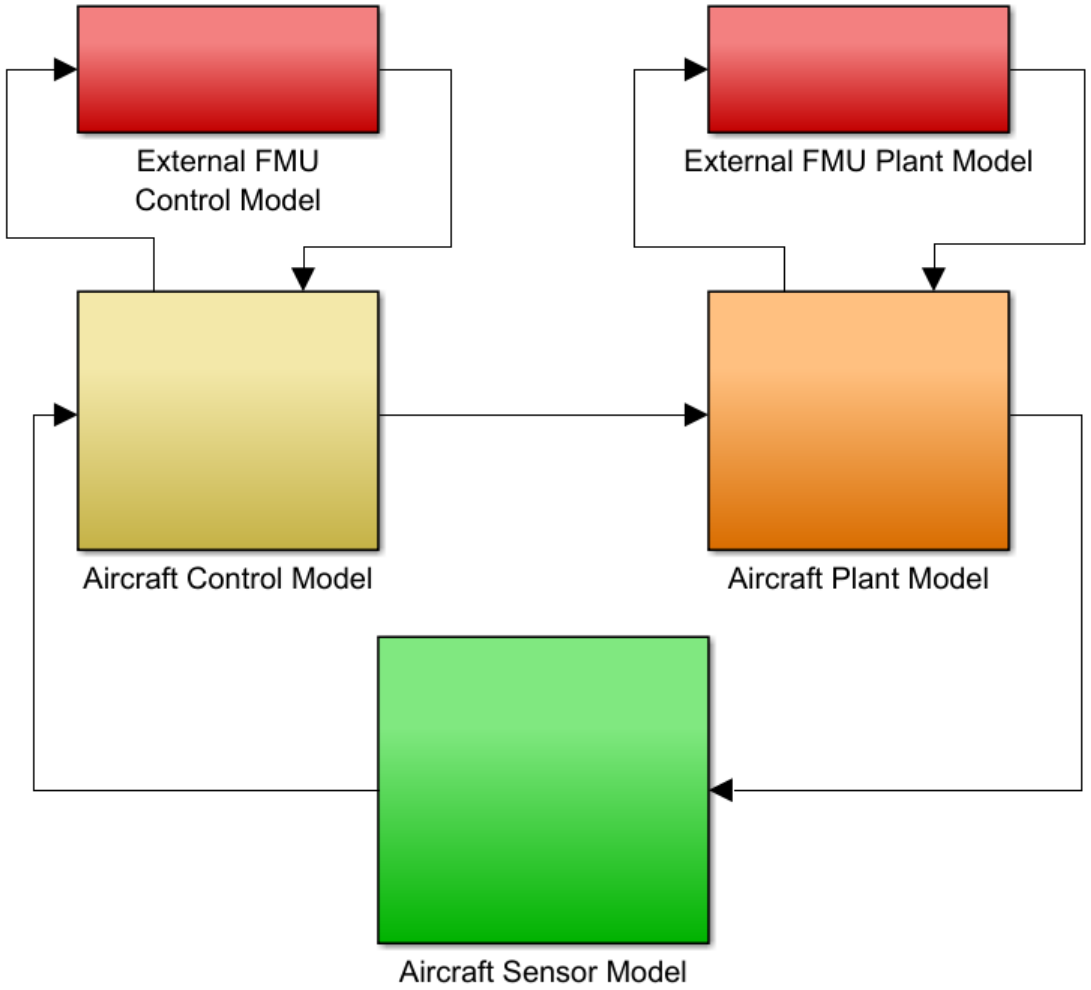
Supernal Functional Simulator Integration Process

Functional Sim 2.0: going beyond flight dynamics and aircraft systems software

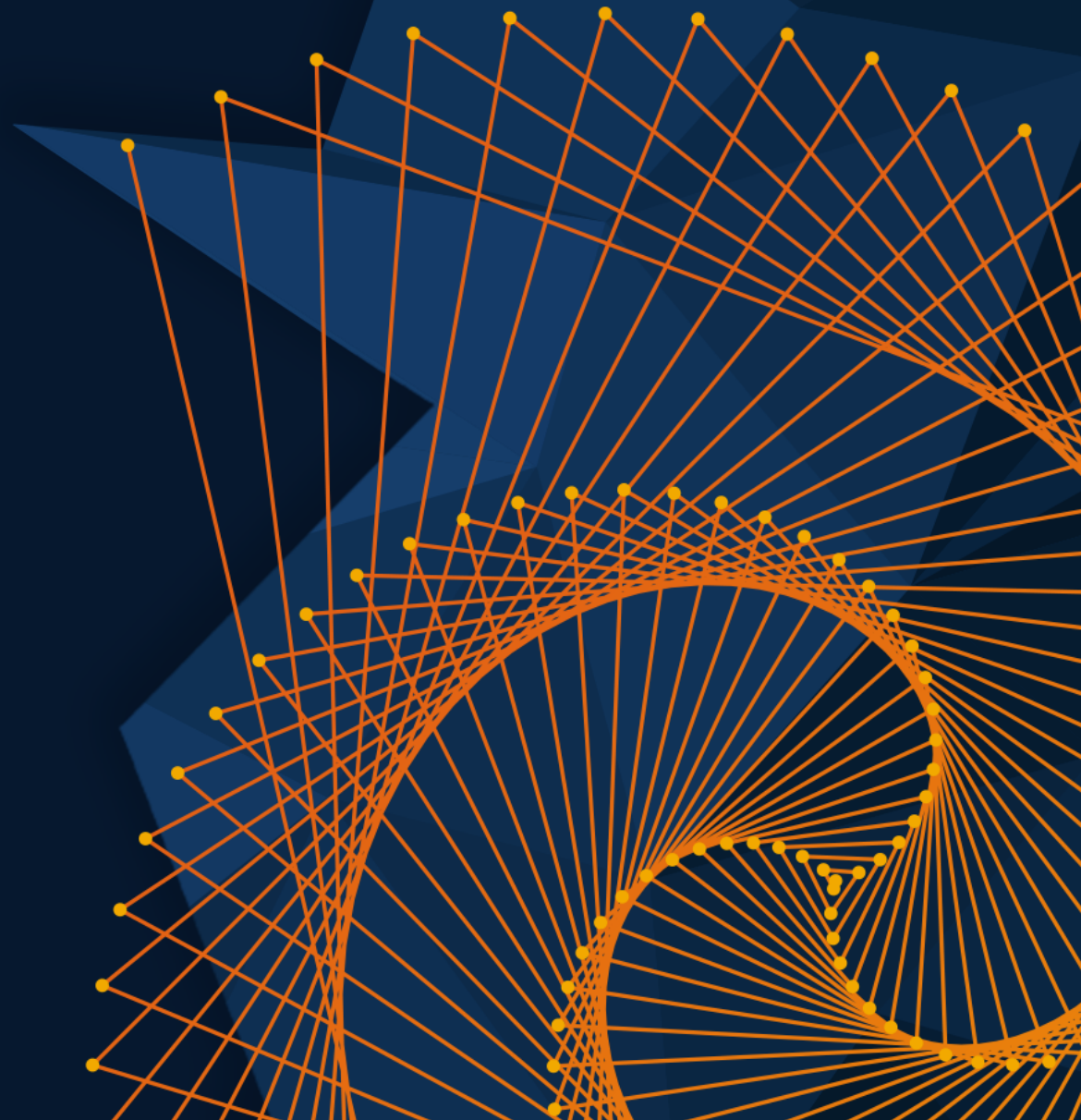


Increasing cross-org collaboration and ecosystem participation by adding Functional Mock-up Unit (FMU) co-simulation modality

Functional Mock-Up Interface



MATLAB EXPO



MathWorks®