What is Simulink?

- Simulink is a graphical extension to MATLAB for modeling and simulation of systems.
- It enables rapid construction of virtual prototypes to explore design concepts at any level of detail with minimal effort.
- For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams.
- It includes a comprehensive library of pre-defined blocks to be used to construct graphical models of systems using drag-and-drop mouse operations.

What is Simulink?

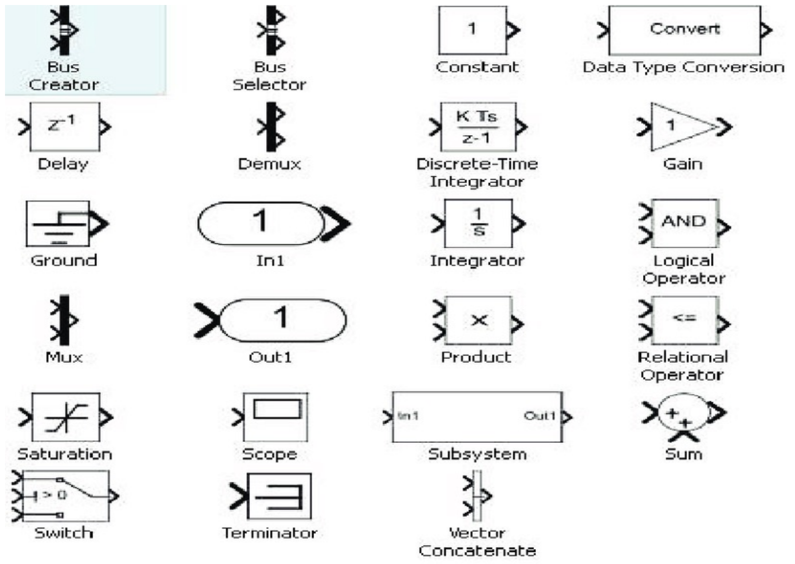
- A **Visual Programming Powerhouse:** Build systems intuitively by connecting pre-built blocks resembling real-world components. Forget complex coding!
- **Extending Beyond Math**: Integrates seamlessly with MATLAB, providing access to mathematical and computational tools for even greater depth.
- A **Multifaceted Gem**: Caters to diverse disciplines, from engineering and science to control systems and signal processing.
- *Its visual programming approach lets you build systems by dragging and dropping blocks representing functionalities, making it accessible even if you're not a coding pro.*
- *Plus, its tight integration with MATLAB offers mathematical and computational muscle, expanding its capabilities further.*

Basic elements of Simulink

- There are two major categories of elements in Simulink:
  - Blocks
  - Lines
- Blocks are used to generate, modify, combine, output, and display signals.
- Lines, on the other hand, are used to transfer signals from one block to another.

Blocks

- Blocks have zero to several input terminals and zero to several output terminals.
- Unused input terminals are indicated by a small open triangle.
- Unused output terminals are indicated by a small triangular point.
- The block shown below has an unused input terminal on the left and an unused output terminal on the right.



Blocks

- There are several general classes of blocks within the Simulink library:
  - **Continuous** function blocks such as Derivative and Integrator
  - **Dashboard**-Blocks that can control parameter values and display signal values during simulation
  - **Customizable Blocks** - Blocks with customizable appearance that control parameter values and display signal values during simulation
  - **Discontinuous** function blocks such as Saturation
  - **Discrete** time blocks such as Unit Delay
  - **Logic or bit operation** blocks such as Logical Operator and Relational Operator

Basic elements of Simulink

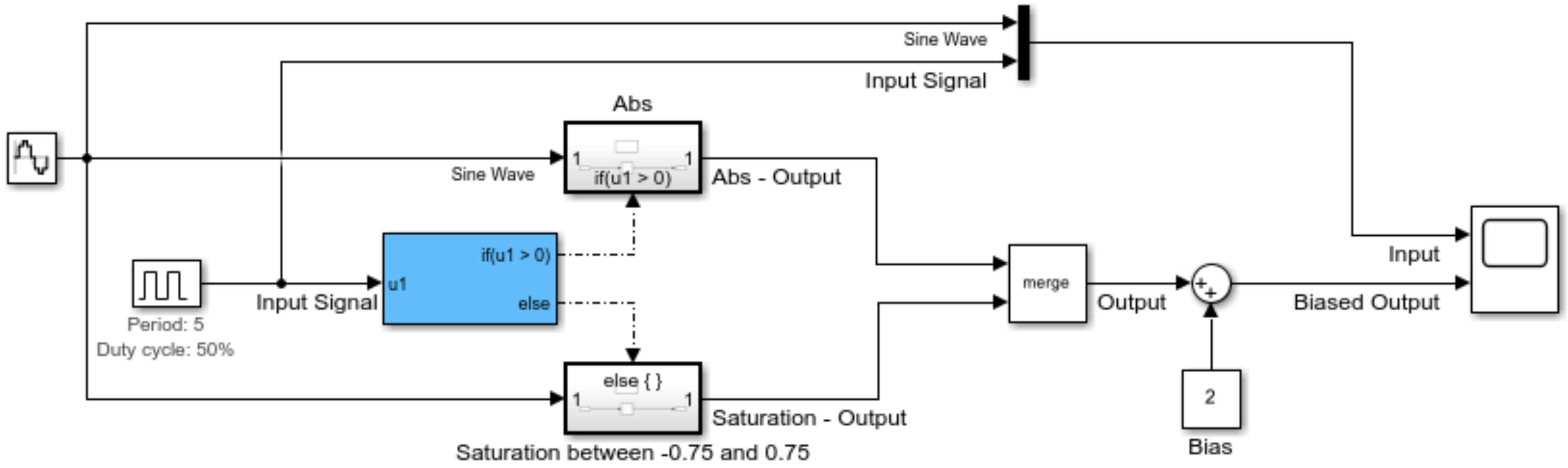
- **Lookup table** blocks such as Cosine and Sine
- **Math Operations -** Mathematical function blocks such as Gain, Product, and Sum
- **Matrix Operations -** Blocks for modelling matrix operations
- **Messages & Events** - Blocks for modeling message-based communication
- **Model Verification** - Blocks for self-verifying models, such as Check Input Resolution
- **Model-Wide Utilities** - Model-wide operation blocks such as Model Info and Block Support Table
- **Ports and Subsystems -** Blocks related to subsystems, such as Inport, Outport, Subsystem, and Model
- **Signal Attributes** - Blocks for modifying signal attributes such as Data Type Conversion
- **Signal Routing** - Route signal blocks such as Bus Creator and Switch

Basic elements of Simulink

- **Sinks** - Log and visualize signal data and terminate signal lines
- **Sources** - Provide inputs for simulation using blocks that define and generate signals or load signal data
- **String -** String manipulation blocks
- **User-Defined Functions** - Custom function blocks such as MATLAB Function, MATLAB System, Simulink Function, and Initialize Function
- **Additional Math and Discrete** - Mathematical and discrete function blocks such as Decrement Stored Integer

Lines

- Lines transmit signals in the direction indicated by the arrow.
- Lines must always transmit signals from the output terminal of one block to the input terminal of another block.
- On exception to this is a line can tap off of another line, splitting the signal to each of two destination blocks.



Key Features of Simulink

- **Extensive Library**: Leverage a wealth of pre-built blocks representing motors,sensors, actuators, and countless other components.
- **Customization Freedom**: Craft unique blocks to model specific components or modify existing ones for bespoke requirements.
- **Simulation Mastery**: Run sophisticated simulations, analyze results, and refine your designs in a virtual environment.
- **Code Generation Power**: Seamlessly translate Simulink models into real-world code for deployment on hardware platforms.

Benefits of Simulink

- **Accelerated Development**: Rapid prototyping and efficient simulations shave off precious time from your development cycle.
- **Enhanced Design Confidence**: Test and analyze designs rigorously, leading to more robust and reliable systems.
- **Improved Safety**: Experiment in a virtual environment, minimizing risks associated with real-world testing.
- **Streamlined Collaboration**: Share models and components effortlessly, fostering teamwork and knowledge sharing.

UAV Toolbox

- A specialized toolbox within MATLAB, offering dedicated tools for UAV design, simulation, testing, and deployment.
- Build UAV systems graphically using pre-built blocks, simplifying development compared to text-based coding.
- Leverage a rich library of models for motors, sensors, actuator, autopilots, and more, saving development time.

Key features of UAV Toolbox

- **Design & Model UAV Systems**: Create detailed models of your UAVs, including control systems, autopilots, and sensor fusion algorithms.
- **Simulate Realistic Scenarios**: Test your UAV designs in diverse simulated environments, evaluating performance and identifying potential issues.
- **Rapid Prototyping & Testing**: Generate code from Simulink models for real-time testing on hardware-in-the-loop setups or actual UAVs.
- **Advanced Control Design**: Develop and test sophisticated control algorithms using dedicated libraries and analysis tools within the toolbox.

Key features of UAV Toolbox

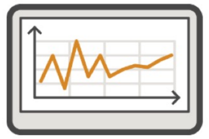
- First, you have to develop a model of the actual plant (hardware) in a simulation environment such as Simulink, which captures most of the important features of the hardware system.
- After the plant model is created, develop the controller model and verify if the controller can control the plant (which is the model of the motor in this case) as per the requirement.
- This step is called Model-in-Loop (MIL) and you are testing the controller logic on the simulated model of the plant.
- If your controller works as desired, you should record the input and output of the controller which will be used in the later stage of verification.

Building box for UAV Simulation

Plan Mission

Generate flight path for the mission

Simulated Waypoints

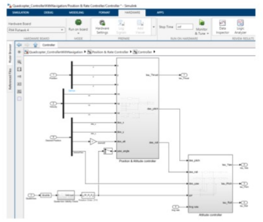


Ground Control Station

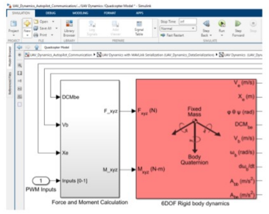
Design & Simulate

Design flight controller and simulate plant behavior in virtual scenarios

Flight Controller



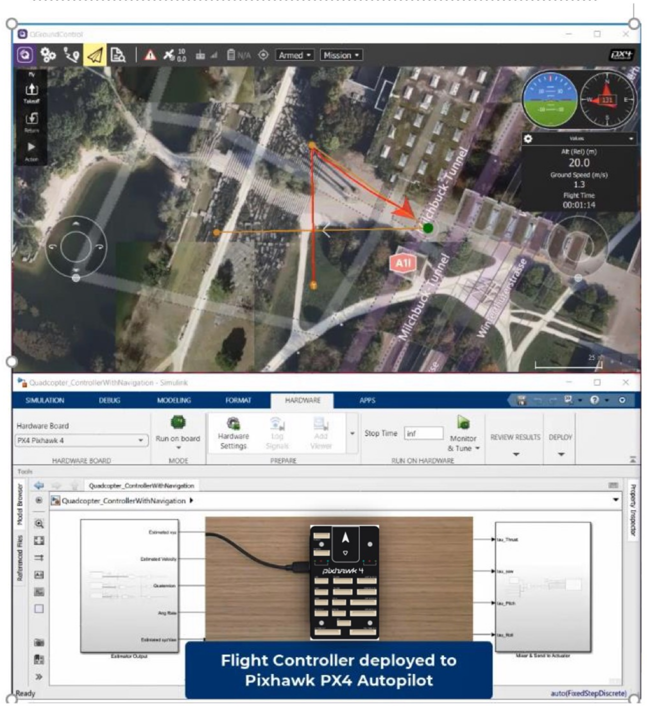
Plant Model



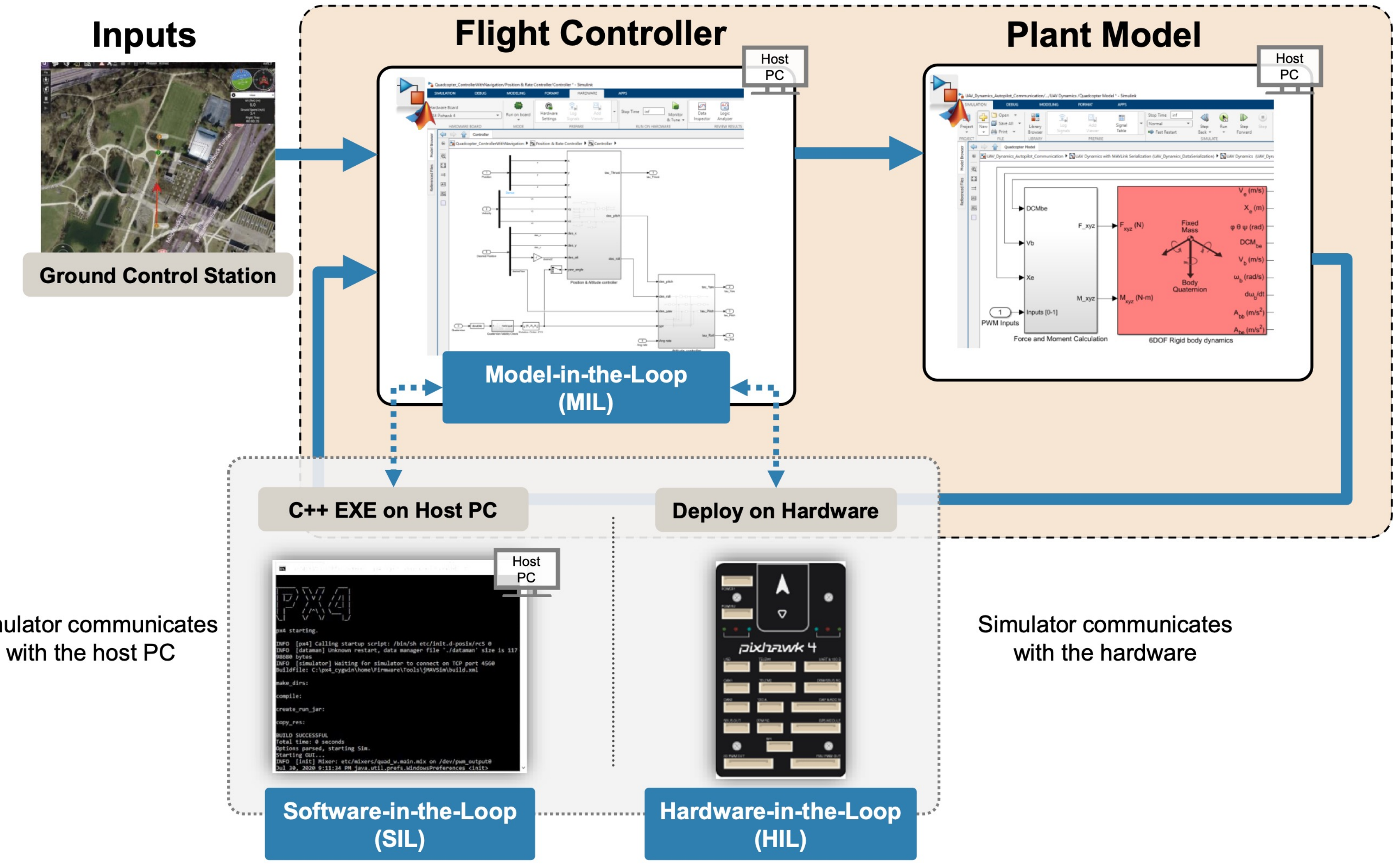
Scenario Simulation

Validate & Deploy

Deploy flight controller and autonomy algorithms to the platform



MIL, SIL, HIL Workflows for UAV Simulation

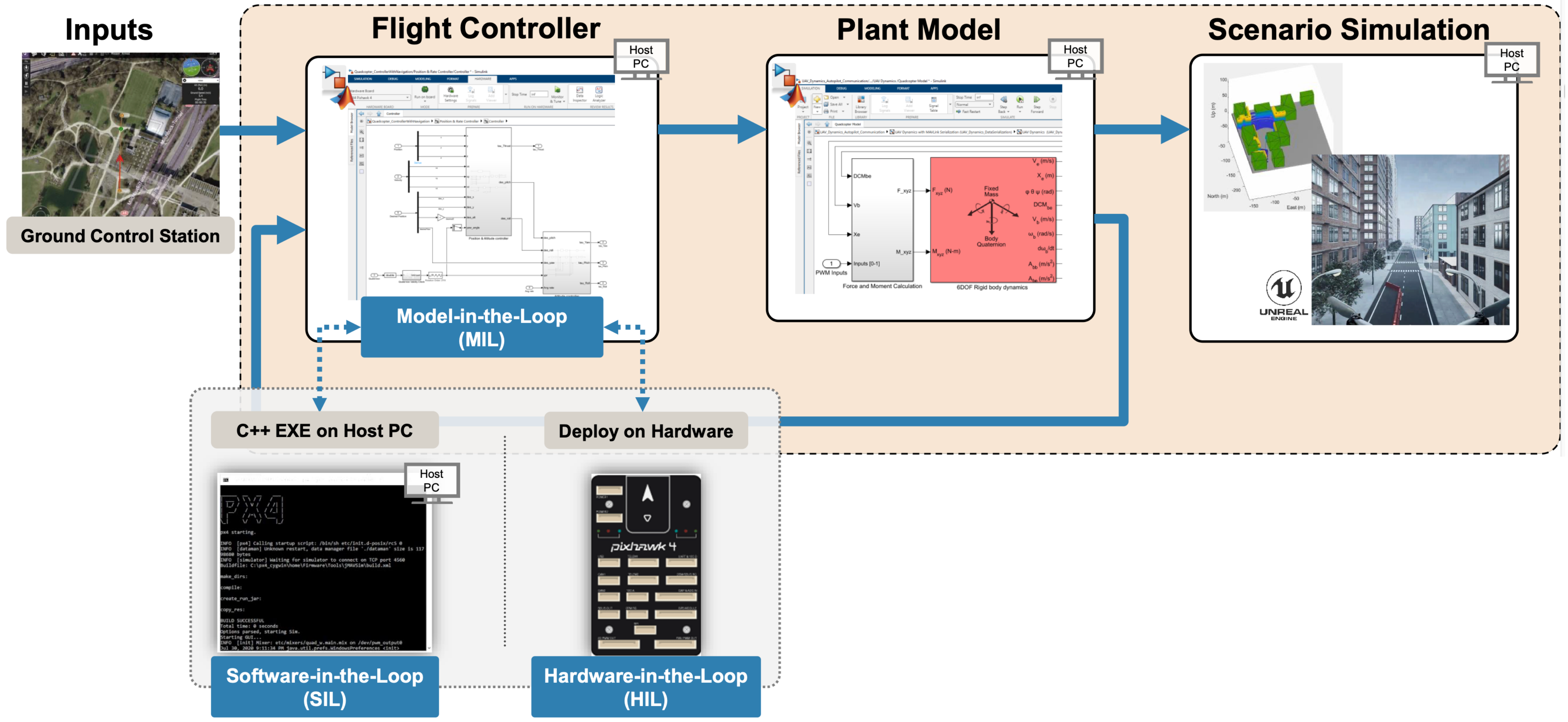


```
PX4

px4 starting.

INFO  [px4] Calling startup script: /bin/sh etc/init.d-posix/rcS 0
INFO  [dataman] Unknown restart, data manager file './dataman' size is 117
08680 bytes
INFO  [simulator] Waiting for simulator to connect on TCP port 4560
Buildfile: C:\px4_cygwin\home\Firmware\Tools\jMAVSim\build.xml

make_dirs:

compile:

create_run_jar:

copy_res:

BUILD SUCCESSFUL
Total time: 0 seconds
Options parsed, starting Sim.
Starting GUI...
INFO  [init] Mixer: etc/mixers/quad_w.main.mix on /dev/pwm_output0
Jul 30, 2020 9:11:34 PM java.util.prefs.WindowsPreferences <init>
```

Simulator communicates with the host PC

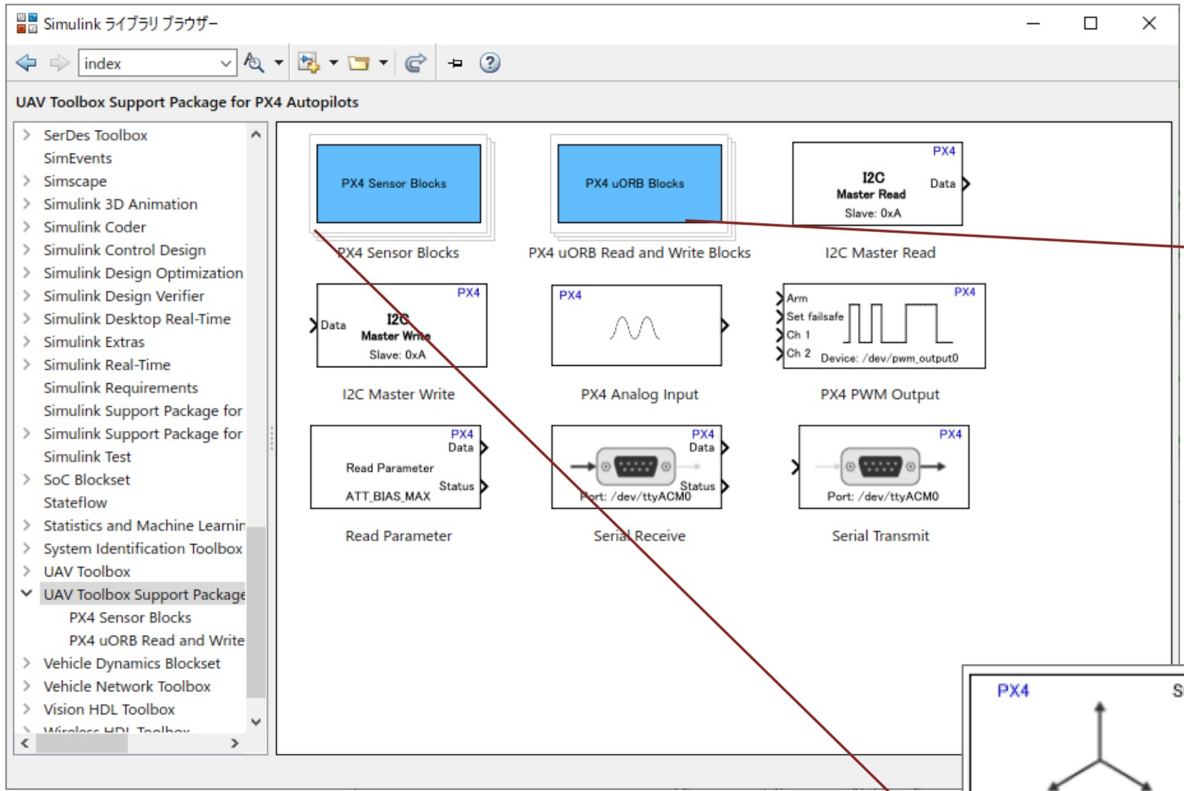
Simulator communicates with the hardware

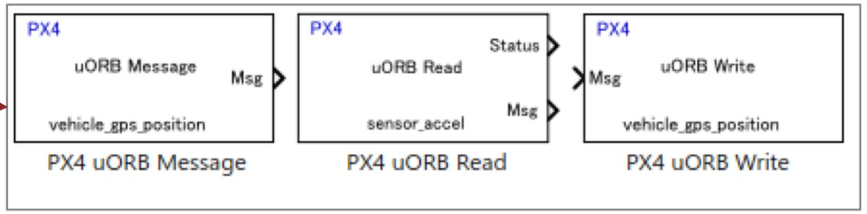
MIL, SIL, HIL Workflows for UAV Simulation



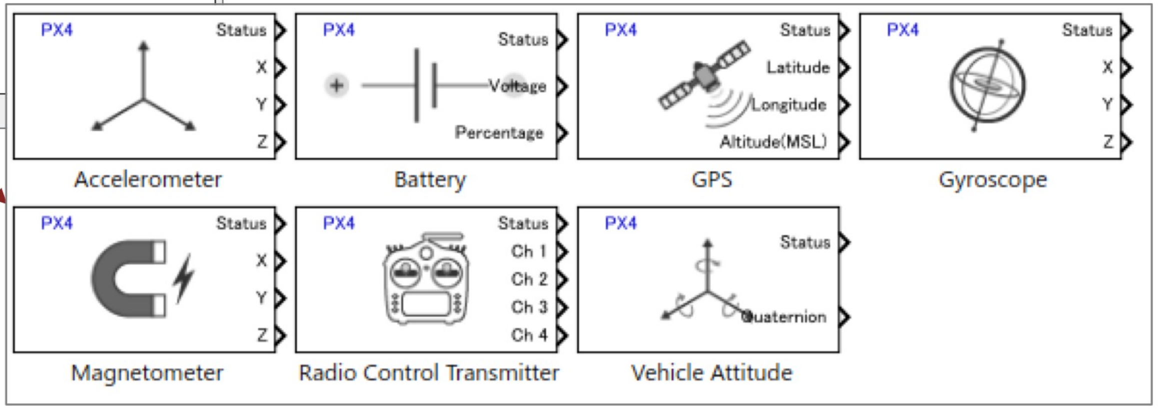
```
px4 starting.

INFO  [px4] Calling startup script: /bin/sh etc/init.d-posix/rcS 0
INFO  [dataman] Unknown restart, data manager file './dataman' size is 117
98680 bytes
INFO  [simulator] Waiting for simulator to connect on TCP port 4560
Buildfile: C:\px4_cygwin\home\Firmware\Tools\jMAVSim\build.xml

make_dirs:

compile:

create_run_jar:

copy_res:

BUILD SUCCESSFUL
Total time: 0 seconds
Options parsed, starting Sim.
Starting GUI...
INFO  [init] Mixer: etc/mixers/quad_w.main.mix on /dev/pwm_output0
Jul 30, 2020 9:11:34 PM java.util.prefs.WindowsPreferences <init>
```

UAV Toolbox support package for PX4





uORB: Asynchronous pub/sub messaging API (middleware)


PX4
Status
X
Y
Z
Accelerometer
Voltage
Percentage
Battery
Latitude
Longitude
Altitude(MSL)
GPS
Gyroscope
Magnetometer
Ch 1
Ch 2
Ch 3
Ch 4
Radio Control Transmitter
Quaternion
Vehicle Attitude