

Release Notes

R2021b



MATLAB® & SIMULINK®



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Release Notes

© COPYRIGHT 2021 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

September 2021	Online only	New for Release 2021b
----------------	-------------	-----------------------

1	Release Notes	
	Installation and Licensing Changes	1-2
	Installation Changes	1-2
	Licensing Changes	1-2
	New Features	1-3

Release Notes

Installation and Licensing Changes

Installation Changes

There are no installation changes for R2021b prerelease.

Licensing Changes

There are no licensing changes for R2021b prerelease.

New Features

MATLAB

Environment

- Editor Selection: Select and edit a rectangular area of code
- Editor Display: Zoom in and out in the Editor
- Editor Sections: Create sections with an improved appearance
- Editor Code: Show code suggestions and completions automatically
- Editor Code: Automatically complete block endings, match delimiters, and wrap comments while editing code
- Editor Code: Change case of text and code
- Editor Debugging: Diagnose problems in scripts and functions using inline debugging controls and a breadcrumb-style function call stack
- Editor Bookmarks: Maintain bookmarks after closing a file
- Editor Refactoring: Automatically convert selected code to a function
- Live Editor Controls: Set default values for sliders, drop-down lists, check boxes, and edit fields
- Live Editor Figures: Interact with real MATLAB figures and resize them with improved layouts
- Live Editor Animations: Export animations to movies or animated GIFs
- Importing Preferences from Previous Releases: MATLAB R2021b checks for preferences from up to four previous releases
- Display language: MATLAB uses Windows display language settings for selecting desktop language
- Comparison Tool: Compare and merge text files with improved usability, appearance, and syntax highlighting

Language and Programming

- `cast` Function: Consistent output for all syntaxes with same data type conversion
- Run Code in the Background: Use parallel language to run code asynchronously
- Prototype Parallel Code: Share parallel code and seamlessly run in parallel
- Compact Display for Classes: Customize display of information about classes when space is limited
- Class Aliasing: Create aliases for renamed classes to maintain backward compatibility
- Modular Indexing: Customize class indexing operations individually using new superclasses
- Scalar Classes: Inherit from `matlab.mixin.Scalar` superclass to ensure instances behave as scalars

Data Analysis

- **Compute by Group** Live Editor Task: Interactively summarize, transform, or filter groups of data
- **Normalize Data** Live Editor Task: Interactively center and scale data
- **Clean Missing Data** Live Editor Task: Define missing values
- `trenddecomp` Function: Find trends in data

- `min` and `max` Functions: Specify comparison method for determining minimum and maximum values
- `uniquetol` Function: Options to control element selection and preserve range of data
- Data Preprocessing Functions: Specify table variable as sample points vector
- `dateshift` Function: Shift to next occurrence of weekday or weekend day
- `isbetween` Function: Support for open, closed, and half open intervals
- `isregular` Function: Support for `datetime` and `duration` data types
- `istabular` Function: Determine if input is table or timetable
- `retime` and `synchronize` Functions: Median and mode methods supported
- `timeofday` Function: Return date as second output argument
- `timeseries2timetable` Function: Convert `timeseries` objects to timetables

Data Import and Export

- `sftp` Function: Connect to SFTP servers
- Datastores: Specify `FileSet` objects as data locations for some datastores
- HDF5 Functions: Use new and enhanced functionality in low-level interface
- NetCDF Function Interface: Read and write `NC_STRING` data
- Scientific File Format Libraries: HDF5 and NetCDF libraries are upgraded
- Audio, Video, and Image I/O Functions: Run functions in a thread-based environment
- Image File Format Libraries: LibTIFF library upgraded to version 4.2.0
- New **Serial Explorer** and **TCP/IP Explorer** apps

Mathematics

- `ode78` and `ode89` Functions: High-order Runge-Kutta solvers for ordinary differential equations
- `pagesvd` Function: Perform singular value decomposition on pages of N-D arrays
- `svd` Function: Option to control output format of singular values
- `mpower` Function: Improved algorithm for defective matrices

Graphics

- Plotting Table Data: Create scatter plots, bubble charts, and swarm charts by passing tables directly to plotting functions
- Axes Ticks and Colors: Control the appearance of axis tick marks and tick label colors
- Create Plot Live Task: Add additional visualizations to generated plots
- Create Plot Live Task: Configuration dropdown for overloaded charts
- `exportgraphics` Function: Capture and append graphics to existing PDFs
- `stackedplot` Function: Support for semi-log y-axes
- Text Objects: Use `editInteractions` in the `Interactions` property to click or tap on text to edit
- MATLAB Figure Code Generation: Code generated for Figure Toolstrip interactions
- `dataTipTextRow` Function: use other data properties like `UserData` as value to customize data tip content

- MATLAB Online™ Accessibility: Use a screen reader to interact with figures

App Building

- `uiaalert`, `uiconfirm`, and `uiprogressdlg` Functions: Mark up text and display equations in dialog boxes
- `addStyle` Function: Add styles to nodes and levels in a tree UI component
- `uitable` Function: Set and query table selections programmatically and control table selection options
- `uitextarea` Function: Program apps to respond while a user is typing in a text area component
- App Designer: Debug code in Code View
- App Designer: Efficiently manage your app code with tools and shortcuts from Live Editor
- App Designer: Interactively modify canvas zoom level and fit canvas to view
- App Designer: Convert between similar UI components
- App Designer: Add help text for your app
- App Designer: Remove auto-reflow behavior from an app with auto-reflow
- App Testing Framework: Perform press gestures on axes and UI axes with different selection types
- App Testing Framework: Perform drag gestures on axes and figures with different selection types
- App Testing Framework: Use any units of measurement in gestures at the center of components

Performance

- `table` Data Type Indexing: Improved performance when assigning elements by subscripting with curly braces
- Titles and Labels in Plots: Improved performance when creating and querying titles or labels in a loop
- Plot Interactions: Improved performance for rendering data tips and rotating scatter plots of large data sets
- App Designer: Improved performance when opening Start Page and loading apps
- App Designer: Improved performance when saving apps
- Comparison Tool: Improved performance loading and saving MLAPP files
- `uigridlayout` Function: Improved performance when adding components spanning multiple columns with 'fit' width
- `uigridlayout` Function: Improved resizing performance when wrapping text in resizable columns
- `uitable` Function: Improved performance loading table data

Software Development Tools

- Projects: Collaborate using Projects in MATLAB Online
- Unit Testing Framework: Use the `TestCase` class template to create tests more quickly and accurately
- Unit Testing Framework: Run live-function-based tests interactively in MATLAB Online
- App Testing Framework: Perform press gestures on axes and UI axes with different selection types
- App Testing Framework: Perform drag gestures on axes and figures with different selection types

- App Testing Framework: Use any units of measurement in gestures at the center of components

External Language Interfaces

- C++ interface `clibgen.generateLibraryDefinition` function: Overwrite existing library definition files
- C++ interface: `char []` parameters behave like `char *` parameters
- C++ interface: Publisher options for defining arguments
- C++ interface: Support for static data members
- C++ interface: Support for `void**`
- C++ interface: Options to define argument shape
- Java interface: Specify JRE path for MATLAB
- Java: Calling into MATLAB from a Java program called by MATLAB
- Python interface: Run Python commands and scripts from MATLAB
- Python: Version 3.9 support
- WSDL Web Services Documents: Apache CXF version 3.4.2 support
- Perl 5.32.1: MATLAB support on Windows

Simulink

Simulink Editor

- Simulink Online supports Projects
- Signal Tracing Command Line API
- Improved search experience in quick insertion of block
- Dock Diagnostic Viewer to a model
- View how to pan and zoom in the Simulink Editor by pressing and holding the space bar
- Miniature map provides context when viewing part of a block diagram
- Move Simulink Editor tab to a new window
- Quick action search menu provides default results
- Reorder categories in Simulink Favorite Commands gallery
- Visually identify line or port as primary selection
- Enhancements to right-click context menu
- Display of port label information and automatic editing of port labels on Subsystem blocks
- Display related blocks in open diagrams or new tabs
- Propagate tag change from Goto block to related From and Goto Tag Visibility blocks

Simulation Analysis and Performance

- Solver Switching with Fast Restart
- Run multiple simulations for scenarios specified in the Signal Editor block from the Simulink Editor
- Copy, cut, and paste plotted signal selection from one run to another in the Simulation Data Inspector

- View multidimensional signal data on an array plot in the Simulation Data Inspector
- Export video data to an MP4 file using the Simulation Data Inspector
- Configure comparison constraints using the Simulation Data Inspector UI
- Easily swap x and y data on XY plots in the Simulation Data Inspector
- Simulation Data Inspector UI enhancements
- Interactive comparison report enhancements
- Simulation Data Inspector snapshot supports capturing a map visualization in a MATLAB figure
- SIMD hardware acceleration support for Just-in-Time (JIT) simulation
- Reuse compiled block diagram for all normal-mode instances of referenced model
- Extract data from multiple elements of a `Simulink.SimulationData.Dataset` object and put the data into timetables using the `extractTimetable` function
- Use runtime callbacks to set inport data and outport data at the root level of a Simulink model

Component-Based Modeling

- Convert input and output ports to bus ports at root-level of model
- Support for code generation workflows in Variant Reducer
- Use new field in `CompiledVariantInfo` parameter to identify execution status of a block
- Prevent accidental edits to subsystem reference using the read-only property
- Open test harness automatically while running it on a subsystem
- Schedule Editor: Use a global scoped event to combine multiple instances of an event
- Call periodic function-call subsystem intermittently using Function-Call Generator block
- Guard header files in generated code for S-Function blocks in variant models
- Allow message root-level In Bus Element block to provide receive queue properties
- 'startup' Variant Activation Time: Change active choice before simulation or in the generated code during model initialization
- Variant Parameters: Enhanced simulation speed, improved code layout, and option to select active choice while running executable

Project and File Management

- Model Comparison: Compare Simulink models in MATLAB Online
- Projects: Collaborate using Projects in MATLAB Online
- Text Comparison: Compare and merge text files with improved usability, appearance, and syntax highlighting
- OPC format for MDL files
- Project Preferences: Detect project-wide references when renaming Simulink buses and bus elements

Data Management

- Specify signal properties based on application-specific value types such as wind velocity
- Signal Editor user interface now accepts simple MATLAB expressions in tabular area
- Signal Editor adds `timeseries` data by default

- `Simulink.io.FileType` has new methods
- New `Simulink.io.ModelWorkspace` class
- Configure complexity for `Simulink.Parameter` objects
- Define complex model arguments without specifying a default value
- Import `struct` and `enum` types without `typedef` from custom C code

Block Enhancements

- New Matrix Operations library
- New button blocks in the Customizable Blocks library
- New switch blocks in the Customizable Blocks library
- New Rotary Switch block in the Customizable Blocks library
- New Lamp block in the Customizable Blocks library
- Upgrade Signal Builder blocks to Signal Editor blocks with Upgrade Advisor
- Upgrade Advisor: Check output dimensions of MATLAB Function blocks
- Lookup table viewer updates
- Parameter Tunability Enhancements for Continuous Blocks
- Signal Builder Import File dialog box updated
- Visualize and edit n-dimensional table data in mask dialog control using lookup table editor
- Control initial size of column in a custom table
- Specify Row and AlignPrompts for containers in System objects blocks
- Specify descriptions with In Bus Element and Out Bus Element blocks
- String support for MATLAB System Block
- Push Button block enhancements
- FMU Import updates

Connection to Hardware

- Deploy customizable Gauge and Knob Dashboard blocks on Android device
- Generate code from Simulink model to run directly on Android devices using PIL simulation
- Read and write commands to TalkBack ID on ThingSpeak TalkBack for Android devices
- Send and receive data using Bluetooth low energy protocol on Arduino boards
- Create interactive display of Dashboard Push Button block on Arduino boards
- Deploy Input Capture and SD Card File Read blocks on Arduino Nano 33 BLE Sense boards
- Measure rotation of motor interfaced with rotary encoder on Arduino boards
- Read and write from ThingSpeak data channels simultaneously on Arduino boards
- Encode and decode data packets based on any messaging format for Arduino-based peripheral communication
- SocketCAN support added for CAN Transmit and CAN Receive blocks on Raspberry Pi boards
- Deploy customizable Gauge and Knob Dashboard blocks on Raspberry Pi boards
- Read and write commands to TalkBack ID on ThingSpeak TalkBack on Raspberry Pi boards

MATLAB Function Blocks

- **MATLAB Function Block Editor** in Simulink window
- Output column vectors as two-dimensional or one-dimensional data in MATLAB Function blocks
- Expanded code generation for tables and timetables
- Code generation for more MATLAB functions
- Code generation for more toolbox functions
- MATLAB Function blocks support implicit expansion

Modeling Guidelines

- Use modeling guidelines to verify compliance of your models

S-Functions

- S-Function Builder: Support for array of buses, nested array of buses, and single-word `fixdt` data types

5G Toolbox

- **5G Waveform Generator** app updates
- Enhanced PUCCH support in programmatic waveform generation
- Enhanced support for PUCCH decoding
- Release 16 updates to PRACH generation
- Support for TR 38.901 propagation path loss
- TDL and CDL channel model updates
- Compute RI
- 5G NR Waveform Acquisition and Analysis
- System-level simulation updates

AUTOSAR Blockset

- AUTOSAR software component modeling
- AUTOSAR adaptive software component modeling
- Export software component mapping for AUTOSAR ECU
- AUTOSAR mapping workflow enhancements

Aerospace Blockset

- New Spacecraft Dynamics block
- Support for 3D simulation rendered using Unreal Engine from Epic Games
- Support Package containing preconfigured Unreal Engine project and maps
- 3DOF equations of motion block changes
- Horizontal Wind Model 07 and Horizontal Wind Model 14 blocks updated
- Orbit Propagator block update

Aerospace Toolbox

- Construct boundary line for visualization
- New plot functions for MIL-F-8785C subplots
- New function to draw altitude envelope contour plot
- New functions to define fixed-wing aircraft
- `atmoshwm` function updated
- Improved performance for the `satelliteScenario` object

Antenna Toolbox

- PCB Antenna Designer App: Design, analyze, and fabricate PCB antennas
- STL Files: Detect bad features in STL file
- Fast Multipole Method (FMM): Support for electrically large antennas and installed antennas
- Method of Moments (MoM) Physical Optics (PO): Faster performance for analysis of electrically large structures
- Infinite Arrays: Support for metal-dielectric structures and metal antennas without ground plane
- Realized Gain: Calculate and visualize realized gain of antennas and arrays
- `EHfields` enhancements: Support for polarization and calculating fields for spherical coordinate system
- Backing Structures Update: Support for metal-dielectric exciters
- RF Propagation: Perform ray tracing analysis using SBR method with diffraction of rays at surface edges
- RF Propagation: Import and view propagation paths within indoor scenes

Audio Toolbox

- Audio Playback Controls in Signal Labeler: Listen to audio signals while annotating them interactively
- Code generation support for audio plugins that use deep learning networks or code replacement libraries
- Pretrained i-Vector System: Apply and customize pretrained speaker verification
- Enhanced `ivectorSystem` training, evaluation, and inference
- Sound Classifier, YAMNet, and YAMNet Preprocess Blocks: Classify sound signals with deep learning using Simulink
- Multiband Parametric EQ Library Block: Design a multiband parametric equalizer in Simulink
- Gammatone Filter Bank Block: Model the human auditory system in Simulink
- GPU acceleration for preprocessing inputs of pretrained deep learning models, speech detection, and feature extraction
- Audio Datastores: Specify `FileSet` objects as data locations
- Functions being removed

Automated Driving Toolbox

Ground Truth Labeling

- Labeler Enhancements: Edit cuboid ROI labels more easily in top, side, and front 2-D view projections, segment ground from lidar data using SMRF algorithm
- Velodyne Lidar Sources: Load data from Velodyne VLS-128 lidar device into Ground Truth Labeler app

Scenario Simulation

- Parking Lots: Add parking lots to driving scenarios programmatically
- ASAM OpenDRIVE Import Enhancements: Import a road network using ASAM OpenDRIVE file versions V1.5 and V1.6
- ASAM OpenDRIVE Export Enhancements: Export a road network to ASAM OpenDRIVE file versions V1.5 and V1.6
- ASAM OpenSCENARIO Export Enhancements: Export the routes of actors using instances of Trajectory element
- INS Block: Generate synthetic readings from an inertial navigation and GPS sensor in driving scenarios in Simulink
- Scenario Reader Block: Obtain position, velocity, orientation, and acceleration information from Ego Vehicle State port
- Road Heading Angles: Create more precise roads using fewer road centers
- Unreal Engine Environment Upgrade: Run 3D simulations using Unreal Engine, Version 4.25
- Position Adjustments of Unreal Engine Cameras: Update relative translation and rotation of camera sensors during simulation
- Unreal Engine Environment Performance Improvements: Run 3D simulations faster than real-time
- Unreal Engine Visualization Example: Visualize logged data for post-simulation analysis

Detection and Tracking

- Perturbations: Perturb object properties using truncated normal distribution
- Code Generation: Generate more memory-efficient C/C++ code from trackers and tracking filters

Applications

- Automated Driving Reference Applications: Automate testing of components of highway lane following and highway lane change systems, and establish message-based communication between model components

Bioinformatics Toolbox

- `rnaseqde`: Perform differential expression analysis on RNA-seq count data

Communications Toolbox

- Multiband signal combiner System object and block
- Sample rate offset System object

- Low-density parity check (LDPC) encoding and decoding functions
- Ultra wideband (UWB) wireless communications examples
- **Wireless Waveform Generator** app export-to-Simulink capability
- Ray tracing analysis support for diffraction of rays at surface edges with SBR method
- Ray tracing channel System object support for control of start time and frame continuity
- Computation of channel impulse response without applying channel filtering
- Support for importing and viewing RF propagation paths within indoor scenes
- Model and visualize RF propagation using MATLAB Online
- Updated display for constellation diagram System object and block
- **Bit Error Rate Analysis** app acceleration enhancements
- Phase noise System object and block support for multichannel input
- Functions for converting between integer and binary data include column-wise and 3D input-output support
- Improved speed performance for certain Communications Toolbox features

Computer Vision Toolbox

Ground Truth Labeling

- Labeler Enhancements: Labeling interactions and other enhancements

Feature Detection and Extraction

- SIFT Feature Detector: Scale-invariant feature transform detection and feature extraction

Recognition, Object Detection, and Semantic Segmentation

- Experiment Manager App Support: Track progress of deep learning detector training
- Deep Learning ROI Pooling: Nonquantized ROI pooling
- Train Video Classifier: Extract video clips for labeling and training workflow
- Deep Learning Object Detector Block: Simulate and generate code for deep learning models in Simulink

Camera Calibration

- Camera Calibration: Circle grid calibration pattern detection
- Camera Calibration: Custom pattern detection
- Rigid 3-D Support: Pass Rigid 3-D object to calibration functions
- OpenCV Camera Parameters: Relay camera intrinsics and stereo parameters to and from OpenCV

Structure from Motion and Visual SLAM

- Bundle Adjustment Solver: Specify optimization solver
- Rigid 3-D Support: Pass 3-D rigid transformation object to camera parameters functions
- Image View Set Support: Find views and view connections
- Bag of Features: Support binary features

- Bag of Features Search Index: Support for Visual SLAM workflow

Point Cloud Processing

- Point Cloud Simultaneous Localization and Mapping (SLAM): Detect loop closures
- Point Cloud View Set Support: Find views and view connections
- Multiquery Radius Search: Optimized radius search for point cloud segmentation
- Point Cloud Viewers: Modify background color programmatically

Code Generation, GPU, and Third-Party Support

- Generate C and C++ Code Using MATLAB Coder: Support for functions
- Generate C and C++ Code Using MATLAB Coder: Compiler links to OpenCV libraries
- Computer Vision Toolbox Interface for OpenCV in Simulink: Specify image data type in Simulink model

Control System Toolbox

- Frequency-Domain Analysis: Support for models with complex coefficients
- `frd`: Support for negative frequencies
- `stepinfo` and `lsiminfo`: Support for nonzero initial value
- `mechss` and `spars`: Linearize structural and thermal models to sparse models

DDS Blockset

- Streamlined IDL import for RTI
- Easy conversion of Simulink data dictionary data types to DDS data types for Data Samples
- Enhanced modeling pattern for Publishers and Subscribers
- String support for C++ code generation
- Class name and namespace control for C++ code generation

DO Qualification Kit (for DO-178)

- Qualification Artifacts for R2021b
- Qualification of HDL Coder Model Advisor checks for verification of compliance with DO-254 safety standards
- Polyspace test suites
- SIL and PIL modes for Simulink Test test suites
- Qualification of additional Model Advisor checks for verification of compliance with high-integrity systems modeling guidelines
- Consolidated Polyspace user documentation

DSP System Toolbox

- New default filter design for FIR rate conversion objects

- New automatic design for FIR rate conversion objects
- Improved speed performance for certain DSP System Toolbox features
- Specify overlap length in moving statistics objects
- Arbitrary frame size support for `dsp.FarrowRateConverter`
- SIMD Code Generation: New code replacement library (CRL) customized for DSP System Toolbox features
- Complex support for SOS filter coefficients
- Tunable `Filename` property for `dsp.AudioFileWriter` and `dsp.AudioFileReader` objects in generated code
- POSIX threads (Pthreads) and OpenMP threading support for multicore custom targets using dataflow domain
- Enhancements for Multicore Tab analysis results
- `dspunfold` does not support Xcode 12.0 or later
- Spectrum Analyzer CCDF measurement reference line changed
- `dsp.ArrayPlot` supports multiple frame sizes
- Functionality being removed
- Objects being removed
- Blocks being removed

Data Acquisition Toolbox

- Analog Input Recorder: Additional measurement types

Database Toolbox

- MongoDB C++ interface

Datafeed Toolbox

- Money.Net Web Socket Interface

Deep Learning HDL Toolbox

- Trimmed deep learning processor configuration
- Generic deep learning processor generation
- Custom reference design functionality for custom boards for deep learning processor IP core integration
- Custom bitstream generation and deployment
- Deep learning processor streaming handshake modes
- Updates to `estimatePerformance`
- Updates to `estimateResource`
- Enhancements for quantization of directed acyclic graph (DAG) networks
- Network prototyping and validation without hardware

- Updated supported layers

Deep Learning Toolbox

- Experiment Manager: Use Bayesian optimization in custom training experiments
- Experiment Manager: Run deep learning experiments in your web browser using MATLAB Online
- Experiment Manager: Improved accessibility with keyboard shortcuts
- Experiment Manager: Stop experiments faster by discarding the results of running trials
- Deep Network Designer: Export trained network to Simulink
- Deep Network Designer: Analyze for `dlnetwork`
- 1-D Convolutional Networks: Create and train networks with 1-D convolution and pooling layers for sequence and time-series data
- 1-D Convolutional Networks: Specify minimum sequence length
- Recurrent Networks: Pass recurrent layer states between layers
- Network Training: Create layer graphs without specifying layer names
- Network Training: Return network with lowest validation loss
- Network Analyzer: Use example inputs when analyzing networks for custom training workflows
- MEX Acceleration: Use MEX acceleration with multi-input and multi-output networks
- Residual Networks: Create 2-D and 3-D residual networks
- Neural Network Apps: New toolstrip design for improved usability
- Function Layer: Create layer that applies a function
- Parallel Inference: Predict, classify, and extract features in parallel with `DAGNetwork` and `SeriesNetwork` objects
- Custom Layers: Define stateful custom layers
- Custom Training Loops: Apply neural ODE operations
- Custom Training Loops: Calculate L_1 and L_2 loss
- Custom Training Loops: Use MEX acceleration with `dlnetwork` prediction
- Custom Training Loops: Compute gradients of loss functions involving complex numbers
- Custom Training Loops: Specify network outputs
- Custom Training Loops: Use flatten layer in `dlnetwork` objects
- Automatic Differentiation: Use complex numbers with `dlarray`
- Automatic Differentiation: Use more functions with `dlarray` input
- Network Training: Stop training when loss is NaN
- Network Training: Create layer graphs from series networks
- Network Training: Include softmax layers in regression networks
- Network Training: Train classification networks without a softmax layer

Econometrics Toolbox

- Threshold-Switching Dynamic Regression Models: Analyze and model nonlinear multivariate time series

- State-Space Models: Apply Kalman filter in real time to estimate state distribution moments
- Markov-Switching Dynamic Regression Models: Assign names to time series and summarize estimation results
- Time Series Preprocessing: Aggregate timetable data to different periodicities ranging from daily to annually
- Diebold-Li Model Example: Analyze yield-curve model augmented with macroeconomic variables

Embedded Coder

Code Generation from MATLAB Code

- Communication I/O information display during SIL or PIL execution
- Visualization of task scheduling
- Reduction of violations for MISRA C++:2008 and AUTOSAR C++14 rules in generated code

Model Architecture and Design

- Built-in storage class for multi-instance data
- Symbolic dimension inputs for Bitwise Operator, Saturation, and Data Type Propagation blocks

Code Interface Configuration and Integration

- Storage class with pointer data access in Embedded Coder Dictionary
- Unstructured Embedded Coder Dictionary storage class application to model reference root I/O
- Embedded Coder Dictionary storage class application to signals and parameters with symbolic dimensions
- Changes to model hierarchy requirements
- Calibration file customization
- TLC code storage classes in default mapping
- Configure additional properties from the Code Mappings editor
- View In Bus Element and Out Bus Element blocks in a hierarchy in the Code Mappings editor
- Configuring C/C++ function prototypes for subsystems not recommended
- Reusable storage class in Code Mappings editor

Code Generation

- Accessibility of step entry-point functions generated for models designed for multitasking and concurrency streamlined
- Code view for MATLAB Function block
- Enhanced code to reduce MISRA C:2012 Rule 10.3 and Directive 4.1 violations
- Changes to generated C++ header files
- `const` member functions for C++ class interface
- Minimized variable visibility for C++ code
- Image data by using OpenCV class `cv::Mat`
- Shared types and parameters storage in same header file

Deployment

- New TLC variable `OverrideSampleERTMain` for disabling generation of example main program

Performance

- Generation of SIMD code by using new configuration parameter
- Image Processing Toolbox functions enhanced with multithreading and algorithm improvements
- Reduced data copies for models that have Bus Creator blocks
- SIMD optimization for more integer data types
- Root output initialization code performance improvements
- Readability improvement for root output initialization code
- Optimize code by unrolling parallel for-loops
- Improved common subexpression elimination
- Optimized SIMD code that performs fused multiply add operations
- Redundant data copies elimination by reusing S-function block buffers
- Optimized code for models containing referenced models
- Nonstatic data class member initialization of instance-specific parameters
- Code replacement for trigonometric functions that use lookup table approximation

Verification

- Communication I/O information display during SIL or PIL simulation

Financial Instruments Toolbox

- Interest-Rate Instruments: SABR-Libor market model
- Equity Instruments: Price cliquet options
- Monte Carlo Simulation: Price interest-rate instruments
- Equity Instrument: Partial lookback options
- Instrument Pricing: `fininstrument` supports vector of instruments
- Interest-Rate Instrument: Enhanced swap and bond curve construction
- Interest-Rate Instrument: LIBOR transition to Alternative Reference Rate (ARR) securities
- Reinforcement Learning Example: Price and hedge instruments using Reinforcement Learning Toolbox

Financial Toolbox

- Portfolio Management Example: Portfolio optimization using social criteria constraints
- Portfolio Management Example: Portfolio diversification
- Portfolio Management: Robustness improvements for MINLP solvers

Fixed-Point Designer

- Generate native half-precision C code for embedded hardware targets

- Rapid Accelerator mode support for half-precision floating-point data types in Simulink
- Generate an optimized lookup table approximation as a MATLAB function
- Improved numerical accuracy and generated code efficiency for fixed-point multiplication
- Improved numerical accuracy and generated code efficiency for fixed-point division
- Implicit Expansion: For `fi plus`, `minus`, and `times`, automatically expand dimensions of length 1
- Fixed-Point Tool: Pause and resume data type optimization search
- Data Type Optimization: Restrict instrumentation to a subsystem
- Data Type Optimization: Enforce known data types for variables in a system
- Fixed-Point Tool: Import `fxpOptimizationOptions` object
- Analytically determine fixed-point data types when solving linear systems of equations
- `fixed.cordicDivide` and `fixed.cordicReciprocal`: Fixed-point divide and reciprocal using CORDIC
- Data Type Optimization: Maintain original values of model parameters that are altered by `fxpopt`
- Fixed-Point Tool provides guided workflow selection
- New functions supported for half-precision inputs
- `fi` support for `dec2base`, `dec2bin`, and `dec2hex`
- Data Type Optimization: Warn about unsupported constructs
- Data Type Optimization: Specify multiple types of tolerances

Fuzzy Logic Toolbox

- `plotfis` Function: Visualize FIS tree structure
- FIS Trees: Specify names for `fi` objects
- New Example: Design fuzzy logic controller for artificial pancreas

GPU Coder

- GPU Memory Manager: Improve allocation efficiency and run-time performance through GPU memory pools
- Atomic Functions: Generate code that uses CUDA atomic intrinsics
- Improvements to reduction operations by using `gpuCoder.reduce`
- Function Inlining: Fine-tune readability and speed of generated code
- GPU Profiling: Generate code execution profiling report by using NVIDIA Nsight Systems
- Deep Learning Workflow: Update network parameters after code generation
- Deep Learning Arrays: Generate code for more functions that use `darray`
- Custom Layers: Use `darray` in deep learning networks that have custom layers
- Code generation from MATLAB for `dlnetwork` that contains sequences
- Mixed-Precision Deep Learning: Perform inference in INT8 precision for additional networks
- Simulink Deep Learning: Generate code for custom layers
- Deep Learning Layers: Generate code for additional layers
- Code generation for page-wise matrix multiplication

- Code generation for additional Computer Vision Toolbox functions
- Code generation for more Image Processing Toolbox functions
- Code generation for additional Signal Processing Toolbox function

Global Optimization Toolbox

- Global Optimization Toolbox Solvers in Problem-Based Workflow: Solve nonsmooth or global optimization problems using optimization variables
- Integer Constraints in `gamultiobj`: Solve multiobjective problems with integer, bound, linear, and nonlinear inequality constraints
- Integer Genetic Algorithm: Solve problems with linear equality constraints or nondefault operators
- Multiobjective Plots: View 3-D `gaplotpareto` plots; Pareto plot tips give index for finding control variables

HDL Coder

Model and Architecture Design

- Layout choices for model generation
- RAM style attributes for Intel/Altera and Microchip
- HDL code check for trigonometric blocks
- Timestamp macro in custom file header comments
- Enhanced multiple enumeration in Verilog
- HDL Industry Coding Standard check for the presence of assignments to the same variable in multiple cascaded conditional regions

Block Enhancements

- Newton-Raphson algorithm for Math Reciprocal block
- Magnitude square function in Math Function block
- Half-precision data types for MATLAB Function block
- Double-Precision data types for Logarithmic block
- For-Generate loops for Reshape and Concat blocks
- Fixed-point output types for Divide block and Reciprocal block
- Enhanced HDL math library
- Optimizations support for Counter blocks
- 4-D and 5-D lookup table support
- Improved denormal optimizations for half-precision data types
- Improved multiplier partitioning DSP QoR
- Reset minimization in Native Floating-Point (NFP) for ASIC
- Set-Reset (SR) flip-flops
- HDL Code Generation for Discrete State-Space block
- Trigger and event modes for subsystems, MATLAB Function blocks, and Stateflow blocks

Code Generation and Verification

- Code View: View your generated HDL code directly in Simulink model window
- Stateflow multicycle path enhancements
- Register-to-register path info option not recommended in HDL Coder
- Execute chart at initialization option for Stateflow charts
- HDL code generation performance improvement for matrix multiplication

Speed and Area Optimizations

- Enhanced sharing and streaming optimizations for matrix-types
- User control for tunable parameter processing and improve code generation time
- Improved zero-protection in Simulink-to-HDL
- Minimize Intermediate initialization of variables in generated HDL code
- Improved optimizations for conditional subsystems
- Delay-balancing behavior standardization in BalanceDelays=off network
- Lookup Table blocks mapping to RAM and adaptive pipelining

IP Core Generation and Hardware Deployment

- Microsemi Libero System On A Chip (SoC) support for IP core generation workflow
- MATLAB Prototyping API Enhancements: Support complex data in AXI4 Stream Interface and input register readback in AXI4 Interface
- Upgrade to Intel Quartus Pro 20.2
- Inserted JTAG AXI Master at fixed frequency to avoid timing issue
- Unsupported tool version in HDL workflow advisor
- Multicycle path constraint packaging for IP core
- HDL Coder Workflow Advisor: Option to expose DUT clock enable port and clock enable output port
- Devicetree generation for IP cores
- Updates to `addAXI4StreamInterface` function for fpga hardware connection object
- Reset AXI4-Stream TLAST counter
- HDL Coder Workflow Advisor: Improved code generation times
- Data type for Speedgoat PCIe Interface: Map bus data types to Speedgoat PCIe Interface
- HDL Coder Support Package for Xilinx RFSoc Devices:Generate, build, and deploy reference designs on Xilinx RFSoc devices

Simscape Hardware-In-The-Loop Workflow

- Support Multiple solver times in Simscape models
- Enable FPGA parameters in the protected model
- RAM Mapping for partition solver

HDL Verifier

- DPI generation supports variable-sized vectors
- DPI generation supports complex data types for test points
- Extended support for FTDI USB-JTAG cable
- MATLAB AXI master support for MII and SGMII interfaces for Xilinx boards
- FPGA data capture support for Ethernet connection for Xilinx boards
- Additional board support
- Updates to supported software

IEC Certification Kit (for ISO 26262 and IEC 61508)

- Qualification and Certification Artifacts for R2021b
- Qualification of additional Model Advisor checks for applications using HDL Coder
- Polyspace test suites
- SIL and PIL modes for Simulink Test tests suites
- Additional guidance for using System Composer in the Model-Based Design Reference Workflows
- Continuous integration by using Jenkins in ISO 26262:2012 case study
- ISO modeling guidelines for compliance with ISO 26262:2018
- Qualification of additional Model Advisor checks for verification of compliance with high-integrity systems modeling guidelines
- Consolidated Polyspace user documentation

Image Acquisition Toolbox

- Support added for GenICam GenTL compliant frame grabbers

Image Processing Toolbox

- Blocked Images: Create and display labeled blocked images
- DICOM: Find and set attributes in DICOM metadata
- Image Quality Metrics: Calculate SSIM metric of deep learning arrays and specify dimensions of computation
- Deep Learning: Added examples using deep neural networks
- `medfilt3` Function: Improved performance for small neighborhood sizes
- C Code Generation: Generate code from five functions using MATLAB Coder
- C Code Generation: Generate portable C code that has improved performance for seven functions
- GPU Acceleration for `ssim` Function
- Thread-Based Environment: Run functions in a thread-backed pool

Instrument Control Toolbox

- New **Serial Explorer** and **TCP/IP Explorer** apps

- TCP/IP Client Interface: Simulink blocks have expanded functionality

LTE Toolbox

- **LTE Waveform Generator** App Updates

Lidar Toolbox

- Lidar Viewer: Visualize and analyze lidar data using app
- Unorganized to Organized Conversion: Convert unorganized point clouds to organized point clouds
- Object Detection Interface: Train and use PointPillars network to detect objects in lidar point clouds
- PointNet++ Network: Create PointNet++ network for segmentation
- Digital Elevation Model: Create digital elevation model of point cloud data
- Lidar Labeler Enhancements: SMRF ground segmentation algorithm
- Lidar Camera Calibrator Enhancements: Manually select checkerboard points for calibration
- Code Generation Support: Generate C/C++ code using MATLAB Coder

MATLAB Coder

MATLAB Programming for Code Generation

- Implicit Expansion: Generate code for element-wise operations and functions with automatic expansion of operand dimensions
- Generate code for MATLAB code that uses class aliases
- Access name of currently running MATLAB function during debugging by using `coder.mfunctionname`

Supported Functions

- Expanded code generation for tables and timetables
- Code generation for more MATLAB functions
- Code generation for more toolbox functions

Generated Code Improvements

- Generate C++11 enumerations that specify underlying type

Code Generation Workflow

- Specify custom hardware targets during code generation

Performance

- SIMD code generation for Intel hardware
- C Code Generation: Generate portable C code that has improved performance for five functions
- Generate optimized code by unrolling parallel for loops

- Eliminated dead code lines containing variable indices
- Improved execution speed through common subexpression elimination
- Generation of vectorized MEX code in JIT compilation mode
- Optimized dynamic array access
- Specify threads to parallelize `for` and `parfor`-loops

Deep Learning with MATLAB Coder

- Deep Learning Workflow: Update network parameters after code generation
- Deep Learning Arrays: Generate code for more functions that use `dlarray`
- Custom Layers: Use `dlarray` in deep learning networks that have custom layers
- Code generation from MATLAB for `dlnetwork` that contains sequences
- Generate generic C/C++ code for more deep learning layers
- Deploy generic C/C++ code on ARM Cortex-M processors
- Generate C++ code that performs inference computations in 8-bit integers for more layers
- Generate C++ code that uses third-party libraries for more deep learning layers

MATLAB Compiler SDK

- Enhanced protection: Additional options for specifying support packages
- Support Packages: Additional options for specifying support packages
- Excel Add-In for MATLAB Production Server: Create an Excel add-in for MATLAB Production Server using the `excelClientForProductionServer` function
- C++ Shared Library Integration: Map MATLAB data types to common C++ types
- Python Version Support for Python Package Integration

MATLAB Parallel Server

- Parallel Language in MATLAB: Share parallel code with any MATLAB user
- GPU Functionality: Use new and enhanced `gpuArray` functions
- GPU Functionality: Use new and enhanced `gpuArray` functions in Statistics and Machine Learning Toolbox
- GPU Functionality: Use new and enhanced `gpuArray` functions for working with signals and audio
- Memory Usage: Use `whos` to check memory used by `gpuArray` and `distributed` variables
- Distributed Arrays: Use new and enhanced distributed array functionality
- Thread-Based Environment: Use new and enhanced functionality on threads for working with audio, video, and images
- Reference Architectures: Deploy and Run MATLAB Parallel Server from Azure Marketplace
- Reference Architectures: Deploy and Run Network License Manager from Azure Marketplace

MATLAB Production Server

- RESTful API for Server Metrics: Retrieve server metrics
- Japanese Language Localization Support: Parameters, variables, and error messages are available in Japanese
- Python Client: Support for Python 3.9

MATLAB Report Generator

- Find and report on Axes objects
- `mlreportgen.dom.PageBorder`: Specify page borders programmatically
- Custom attributes for page borders in PDF Templates
- Include SVG images in PPT API presentations
- PPT API clone Methods: Copy presentation objects

Mapping Toolbox

- Vector Data: Read, display, and write vector data using geospatial tables
- Basemaps: Interactively change basemap of geographic plots
- Raster Import: Read SRTM Height data
- WMS Database: Search updated database

Mixed-Signal Blockset

- Operational Amplifier: Model double pole or multiple pole operational amplifier
- Delta Sigma Modulator: Model delta sigma modulator ADC

Model Predictive Control Toolbox

- Implement Multistage Nonlinear MPC Controllers with FORCESPRO Solver
- Custom QP Solvers: active-set QP solver enabled as custom solver
- ISO 26262: Improved compliance for linear and ADAS blocks

Model-Based Calibration Toolbox

- Transient Calibration: Simplified workflow and robust algorithms for large-scale optimizations
- Transient Calibration: Simplified workflow for production calibration using calibration parameter files
- Calibration History: Restore previous calibrations, including lookup tables, normalizers, and scalars

Motor Control Blockset

- Induction Motor Parameter Estimation: Determine parameters of AC induction motor from experiments with motor hardware

- PMSM Parameter Estimation: Estimate PMSM parameters using quadrature encoder and custom hardware
- Algorithm Export Workflow For Custom Hardware: Reference example
- Sliding Mode Observer: Improve position tracking and automatically estimate observer parameters
- Reference Example Using Field Oriented Control Autotuner Block: Automatically tune current and speed loops running on hardware
- Field Oriented Control Autotuner Block: Reduce execution time on hardware by performing frequency response estimation experiment using sinestream signals
- Initial Rotor Position Estimation: Reference example
- PWM Reference Generator Block: Use modulation strategies that reduce switching losses
- Speed Measurement Block: Support for `uint16`, `uint32`, and `uint64` data types

Navigation Toolbox

- Motion Planning Using Control-Based RRT
- Visualize Rigid Body Pose
- Simultaneous Localization and Mapping (SLAM) Using Extended Kalman Filter (EKF)
- GPS Sensor Block
- Hybrid A* Path Planner Update
- Perturb IMU sensor readings
- NMEA Parser Enhancements

Optimization Toolbox

- `fmincon` 'interior-point' Algorithm: Obtain feasible solutions using new feasibility routine

Parallel Computing Toolbox

- Parallel Language in MATLAB: Share parallel code with any MATLAB user
- GPU Functionality: Use new and enhanced `gpuArray` functions
- GPU Functionality: Use new and enhanced `gpuArray` functions in Statistics and Machine Learning Toolbox
- GPU Functionality: Use new and enhanced `gpuArray` functions for working with signals and audio
- Memory Usage: Use `whos` to check memory used by `gpuArray` and `distributed` variables
- Distributed Arrays: Use new and enhanced distributed array functionality
- Thread-Based Environment: Use new and enhanced functionality on threads for working with audio, video, and images

Partial Differential Equation Toolbox

- Electromagnetic Analysis: Solve 3-D electrostatic and magnetostatic problems
- Local Mesh Refinement: Set target mesh sizes around specified vertices, edges, and faces

- Structural and Thermal Analysis: Extract sparse linear models for use with Control System Toolbox
- PDEVisualization Properties: Display or hide axes and control background color

Phased Array System Toolbox

- Parameterizable 5G antenna
- Steering vector and array response for spherical wavefronts
- Sinc, Gaussian, and cardioid antenna element patterns
- Compute aperture from beamwidth
- Enhancements to plotting and visualization functionality

Polyspace Bug Finder

Documentation

- Documentation: View combined documentation for all Polyspace Bug Finder products
- Documentation: View web documentation by default
- Contextual Help: View contextual help in web browser

Analysis Setup

- IAR Embedded Workbench Compiler: Set up Polyspace analysis for code compiled by using RISC-V target
- Updated GCC Compiler Version Support: Set up Polyspace analysis for code compiled with GCC versions 9.x and 10.x
- C17 Support: Run Polyspace analysis on code that has C17 features
- Configuration from Build System: Copy console output to log file
- Changes in analysis options and binaries
- Functionality being removed: Polyspace stubs for Standard Template Library
- Functionality being removed: Compilation assistant

Analysis Results

- AUTOSAR C++14 Support: Check for 345 AUTOSAR C++14 rules including 18 new rules
- Bug Finder Checkers: Check for inefficient C++ algorithms or function usage and other issues
- CERT C++ Support: Check for violations associated with exception handling
- Changes to external coding standards checking
- Updated Bug Finder defect checkers

Reviewing Results

- Results in Macros: See results in macro expansions when macro parameters cause an issue
- Additional Info in Result Details: See expected and actual values for numerical defects
- Functionality removed: Polyspace Metrics

Polyspace Code Prover

Documentation

- Documentation: View combined documentation for all Polyspace Code Prover products
- Documentation: View web documentation by default
- Contextual Help: View contextual help in web browser

Verification Setup

- Faster Analysis: Reduction in analysis time on code that uses C++ `std::string` library
- IAR Embedded Workbench Compiler: Set up Polyspace analysis for code compiled by using RISC-V target
- Updated GCC Compiler Version Support: Set up Polyspace analysis for code compiled with GCC versions 9.x and 10.x
- C17 Support: Run Polyspace analysis on code that has C17 features
- Configuration from Build System: Copy console output to log file
- Functionality being removed: Coding standards checking and code metrics computation with Code Prover
- Functionality being removed: Polyspace stubs for Standard Template Library
- Functionality being removed: Compilation assistant

Verification Results

- String Library Function Checks: New checks on arguments to C++ `std::string` methods

Reviewing Results

- Functionality removed: Automatic Orange Tester
- Functionality removed: Polyspace Metrics

Powertrain Blockset

- Fuel Cell Electric Vehicle Reference Application: Design an energy system for a hydrogen-based vehicle
- Electric Vehicle Reference Application: Utilize new systems and updated parameters
- Transfer Case Block: Implement transfer cases in drivetrain models
- Simple Engine: Implement simple engines using basic specifications
- Motorcycle Blocks: Implement in-plane longitudinal motorcycles, including chain dynamics
- Simulink Model Data: Use functions that read and write block parameter data for Excel
- Transmission Controllers: Use improved transmission controllers in reference applications

Predictive Maintenance Toolbox

- Diagnostic Feature Designer: Generate spectral features for characteristic fault frequency bands in rotating machinery
- Diagnostic Feature Designer: Rank unlabeled features

- Diagnostic Feature Designer: Use a streamlined workflow for plotting, data processing, and feature extraction
- Remaining Useful Life (RUL) Prediction: Generate C/C++ code using MATLAB Coder for RUL prediction that is based on a survival model
- Rotating Machinery Metrics: Generate C/C++ code using MATLAB Coder for gear condition metrics and fault band metrics

RF Blockset

- N-Port S-Parameters block: Model S-Parameter objects with up to 65 ports

RF Toolbox

- Richards'-Kuroda Workflow: Apply Richards' and Kuroda's transformations to circuit objects
- Phase Shift Element: Design phase shift element using circuit objects or RF Budget Analyzer app
- RF Antenna Element: Design receiver antenna using circuit objects or RF Budget Analyzer app
- Microstrip Transmission Line: Design standard, inverted, embedded, and suspended microstrip transmission lines
- LC Ladder Element: Design LC ladder in RF budget chain or RF Budget Analyzer app
- RF Plot: Support for LC ladder element
- Group Delay, Noise Figure, and S-parameters: Support for electrical length transmission line

ROS Toolbox

- C++ Code Generation Support for ROS: Generate ROS nodes for deploying to target hardware using MATLAB Coder
- C++ Code Generation Support for ROS 2: Generate ROS 2 nodes for deploying to target hardware using MATLAB Coder
- CUDA Optimized Code Generation Support for ROS: Deploy CUDA-optimized ROS nodes from Simulink to target hardware using GPU Coder
- Enhancements for `rosdevice` and `ros2device`: Run ROS and ROS 2 nodes on local device and nodes generated from MATLAB
- ROS Log files: Open, parse and write to rosbag files
- ROS 2 Log Files: Read ROS 2 bag files in Simulink using Read Data Block
- ROS 2 Services: Create ROS 2 service servers and clients
- ROS 2 Call Service Block: Call service in ROS 2 network
- ROS 2 Read Image Block: Read ROS 2 image messages in Simulink
- ROS 2 Read Point Cloud Block: Read ROS 2 point cloud messages in Simulink
- ROS Support for Custom Actions
- ROS 2 Support for Custom Services

Radar Toolbox

- Radar Designer App: Plot vertical coverage diagrams

- Synthetic Aperture Radar: Convert between ground range resolution and slant range resolution
- Radar Data Generator block: Generate radar data in Simulink
- New custom scan mode for `radarDataGenerator`
- Merge detections into clustered detections using `mergeDetections`
- Generate more memory-efficient C/C++ code from tracking filters

Reinforcement Learning Toolbox

- Rewards: Generate reward functions from controller specifications
- Episode Manager: Improved layout management for single and multiple agent training
- Neural Network Representations: Improved internal handling of `dlnetwork` objects
- Trust Region Policy Optimization Agent: Prevent significant performance drops by restricting updated policy to trust region near current policy
- PPO Agents: Improve agent performance by normalizing advantage function

Risk Management Toolbox

- Lifetime Credit Risk Analysis: Exposure at Default (EAD) models
- Deep Learning Example: Credit risk analysis using Deep Learning Toolbox
- Insurance Example: Mean Square Error of Prediction (MSEP) for Estimated Ultimate Claims
- Live Editor Task: Predictor screening
- Lifetime Credit Risk Analysis: Cox lifetime probability of default (PD) model

Robotics System Toolbox

- State Space and State Validation for Robot Manipulator Models
- Ignore Self Collisions for Manipulator RRT Path Planning
- Trajectory Generation
- Simulation Description Format (SDF) Support
- Collada Mesh Support
- Load Robot Function Update

Robust Control Toolbox

- `loopsyn` Command: Balance performance and robustness when designing controllers by loop shaping
- `loopsyn` Command: Synthesize controllers for two-degree-of-freedom architecture
- `ncfsyn` Command: Adjust tolerance to help eliminate fast controller dynamics
- `ncfmr` Command: Reduce model order fully programmatically

Satellite Communications Toolbox

- Satellite Scenario enhancements

- Satellite Link Budget Analyzer app enhancements
- Support for DVB-RCS2
- Support for satellite navigation systems

Sensor Fusion and Tracking Toolbox

- Visualize tracking scenario in virtual globe using `trackingGlobeViewer`
- Handle out-of-sequence measurement (OOSM) using retrodiction
- Import tracking scenario using the Tracking Scenario Recorder Simulink block
- Track objects using Grid-Based Multi Object Tracker Simulink block
- Model and Simulate GPS sensor using GPS Simulink block
- Visualize rigid body position and orientation using `poseplot`
- INS Simulink block provides more parameters to specify its characteristics
- Perturb `imuSensor` properties
- Perturb object properties using truncated normal distribution
- Partition detections using DBSCAN algorithm
- Merge detections into clustered detections using `mergeDetections`
- Generate more memory-efficient C/C++ code from trackers and tracking filters

SerDes Toolbox

- S-Parameter Support of Channel Models: Import S-Parameter using **SerDes Designer** app

Signal Processing Toolbox

- Design Filter Live Editor Task: Design digital filter interactively
- Signal Labeler App: Inspect distribution of label counts on heatmap
- Signal Labeler App: Show outliers in Dashboard
- Signal Labeler App: Listen to audio signals while annotating them interactively
- Signal Analyzer App: Denoise signals interactively using wavelet methods
- Feature Extraction: Extract time-domain and frequency-domain features of signals
- Feature Extraction: Compute zero-crossing rates of signals
- Deep Learning: Short-time Fourier transform layer
- Signal Datastores: Specify `FileSet` objects as data locations
- C/C++ Code Generation Support: Code generation for filtering, spectral analysis, and vibration analysis
- GPU support for digital filtering, feature extraction, signal processing, transforms, and waveform generation
- MATLAB Online support for Signal Analyzer and Signal Labeler

SimBiology

- Specify Different Parameter Values for Different Groups During Fitting
- Generate Reports for Model and Program Results
- Model Simulation Tool
- Percentile Plots
- Plotting Improvements
- Hide or Show Grid Lines from Diagram
- Integration-based Profile Likelihood Support for Parameter Confidence Intervals in **SimBiology Model Analyzer**
- Perform Global Sensitivity Analysis Using Elementary Effects (Morris Method)
- Support for Nonuniform Sampling in `sbiosobol` and `sbiompgsa`
- Sampling Options for `SimBiology.Scenarios`
- `fitproblem` Object for Parameter Estimation

Simscape

Simscape Language

- Improved Code Reuse: Override base class members in derived classes

Foundation Library

- Reservoir (2P) and Controlled Reservoir (2P) Blocks: Specify fluid boundary conditions using extended set of options
- `hydraulicToIsothermalLiquid` Conversion Tool Enhancement: Easily upgrade models containing customized hydraulic blocks
- `simscape.Value` and `simscape.Unit` Objects: Use MATLAB interface to manipulate physical values with units
- Interface Specification for Simscape Connections: Lock down connection types for Simscape Bus and Connection Port blocks
- System Composer Support for Simscape Models: Create physical interfaces, ports, and connections on architecture components

Simulation

- Scalable Compilation: Reduce compilation time for models containing repeated reusable components
- Simscape Variable Scaling Analyzer Tool: Identify issues with model scaling to improve performance
- Stiffness Impact Analysis Tool Enhancement: Perform stiffness analysis at multiple time points
- `simscape.getLocalSolverFixCostInfo` function: Expedite model conversion to fixed-cost
- Data Logging Support for Rapid Accelerator Mode: Use rapid accelerator mode to simulate models with data logging enabled
- Statistics Viewer Enhancement: View and trace secondary variables for 1-D physical systems
- Improved Handling of Implicit Asserts: Avoid runtime errors without impacting performance

Simscape Driveline

- Propeller Block: Add a versatile marine propeller to your model
- Backlash Modeling for Gears: Include backlash with the Simple Gear block
- Longitudinal Vehicle Block: Efficiently perform a system-level analysis using an abstracted vehicle
- Thermal Dampers: Include thermal effects in linear faultable damper blocks
- Modal Pulley Blocks: Simulate pulley networks with improved ease and numerical robustness
- Logic-Controlled Clutch Block Updates: Conduct system-level analysis using an abstracted clutch with optional smoothing

Simscape Electrical

Simscape Electrical

- GTO and Thyristor (Piecewise Linear) Blocks Enhancement: Measure the current at a specific time after reaching the current threshold
- `subcircuit2ssc` Function Enhancement: Support smoothing functions
- Thermal Network for Semiconductor Devices Enhancement: Simulate the heat generated by switching events and conduction losses with a Cauer model with N elements
- Cauer Thermal Model Block: Simulate heat transfer through multiple layers of a semiconductor module
- Foster Thermal Model Block Enhancement: Initialize block in steady state or specify initial conditions
- Moving Average and Fourier Analysis Blocks Enhancement: Specify buffer size
- PWM Generator Block Enhancement: Specify phase delay
- Diode Block Enhancement: Use reverse recovery energy to model charge dynamics
- Ideal Switching Semiconductor Devices Enhancement: Tabulate voltage as a function of temperature and current
- Half-Bridge (Ideal, Switching) Block: Model a half-bridge with ideal switches and thermal port
- Solar Cell Block Enhancement: Model a specific solar cell from multiple built-in parameterizations
- BLDC Block Enhancement: Model a specific brushless DC motor from multiple built-in parameterizations
- Heatsink Block: Dissipate heat from power semiconductors to ambient
- Transmission Line (Three-Phase) Block Enhancement: Model faults
- `ee_convertedMofetValidation` Function Enhancement: Compare MOSFET conversions results with SIMetrix
- `ee_importDeviceParameters` Function: Import device parameters from an XML file to a Simscape block
- Stepper Motor Driver Block Enhancement: Simulate microstepping mode to increase the resolution of the position of the rotor
- SM DC4C Block: Model a discrete-time or continuous-time synchronous machine DC4C excitation system
- Delta-Connected Variable Load Block: Model a three-phase variable load wired in delta configuration

- Solar Cell Block Enhancement: Specify parallel-connected strings in series-connected cells
- Machine Blocks Enhancement: Simulate the effects of generated heat and device temperature by exposing the thermal port
- Composite or Expanded Ports Selection Option Enhancement: Switch between composite and expanded ports in block mask
- Induction Machine Type Selection Option Enhancement: Switch between induction machine types in block mask
- Featured Examples

Specialized Power Systems

- Three-Phase Tap-Changing Transformer (Two-Windings) and Three-Phase Tap-Changing Transformer (Three-Windings) Blocks: Model two-winding and three-winding three-phase tap changing transformers
- Three-Phase Autotransformer with Tertiary Winding: Model a three-phase autotransformer
- Impedance Measurement Tool App: Compute impedance versus frequency of Impedance Measurement blocks in model
- Featured Examples

Simscape Fluids

- New Heat Exchanger Blocks: Model system-level heat exchangers
- New and Modified Fan Blocks: Model fans in moist air
- Run-Time Parameters for Thermal Liquid Blocks: Modify parameter values without regenerating C code
- Updated Blocks in the Pumps & Motors Library: Improve estimation of waste heat added to fluids
- Directional and Pressure Control Valve Block Updates: Model fault types and triggers
- Valve and Centrifugal Pump (IL) Block Updates: Plot block settings
- Updated Isothermal Liquid Valve Blocks: Parameterize blocks with tabulated data or predefined parameterizations
- Centrifugal Pump (IL) Block: Calculate brake power from pump efficiency

Simscape Multibody

- General Flexible Plate Block: Model flexible plates
- Property Inspector: Set the parameters of blocks that do not use the visualization pane
- Magic Formula Tire Force and Torque Block: Model Magic Formula tires
- `simscape.multibody.tirread` Function: Read tire properties files
- Grid Surface Block: Model grid surfaces for contact
- Featured Examples

Simulink Check

- View compliance status of metrics in the Model Testing Dashboard

- Organize models using unit testing hierarchy in the Model Testing Dashboard
- Measure pass and fail criteria metrics in the Model Testing Dashboard
- Added functions for programmatically analyzing requirements-based testing metrics
- Trace additional test results in the Model Testing Dashboard
- View summary of artifacts for each unit in the model testing metrics report
- Artifact tracing enhancements for the Model Testing Dashboard
- Generate report from the Model Testing Dashboard
- Find clones anywhere within the model
- Programmatically detect clones in multiple models
- Refactor model with multiple Interpolation n-D blocks by using the Model Transformer
- Improved edit-time check diagnostic interface for block constraint violations
- Simplified block constraint check authoring
- Additional Model Slicer support for Simulink constructs
- Guideline Sub-ids for additional MAB/JMAAB checks
- High Integrity Systems Modeling Checks: Improve quality and compliance to guidelines
- Observe impact of Simulink parameters using Model Slicer
- Additional checks to verify compliance with CERT C secure coding standards
- Enhancements to edit-time checking to identify more incompatibilities

Simulink Code Inspector

- Code inspection support for more code generation optimizations
- Code inspection for models by using Code Inspector app
- Navigation between generated code and inspection results in Code view
- Verification results of top-level model in verification results panel
- Incorporation of custom data in code inspection report with Command-line APIs
- Option to skip inspection of non-inlined functions
- Additional model compatibility checks
- Enhancements to edit-time checking to identify more incompatibilities

Simulink Coder

Model Architecture and Design

- New ID for check "Check for relative execution order change for Data Store Read and Data Store Write blocks"
- Protection for models that use noninlined S-functions

Code Interface Configuration and Integration

- Changes to model hierarchy requirements
- Calibration file customization
- Configure additional properties from the Code Mappings editor

- View In Bus Element and Out Bus Element blocks in a hierarchy in the Code Mappings editor

Code Generation

- Code generation report enhanced with new functionalities
- Accessibility of step entry-point functions generated for models designed for multitasking and concurrency streamlined
- Target hardware data management
- File size reduction by using `memset` function for zero initialization
- Data reference of model parameter in `model.rtw`
- Model parameter value in TLC
- Configure C++ language standard by using language standard parameter

Deployment

- CMake ships with MATLAB
- XCP external mode simulation through concurrent execution
- Simplified creation of XCP target connectivity objects
- Parameter upload for external mode simulations
- Hold Updates button for Run on Custom Hardware app

Performance

- Generation of SIMD code for Intel hardware

Simulink Compiler

- Improvements for FMU Export
- Export nested FMU
- Save and restore `FinalState` in `Dataset` format in deployed applications

Simulink Control Design

- Discrete Extremum Seeking Control: Design hardware-deployable extremum seeking controller with a specified sample time
- Closed-Loop PID Autotuner Block: Reduce execution time on hardware by performing frequency response estimation experiment using `sinestream` signals
- Operating Points: Obtain state, input, and output indices from operating point search reports

Simulink Coverage

- Collect coverage when running multiple simulations
- Collect and aggregate coverage for `Startup` variants

Simulink Design Optimization

- `surrogateopt`: New Solver Option for Design Optimization Workflow at the Command Line

Simulink Design Verifier

- Detect violations of high-integrity systems modeling guidelines for MATLAB Function blocks and Stateflow charts
- Support for dynamic indexing on Data Store blocks
- Enhancements to Observer support
- Analyze server runnable global Simulink Function blocks inside export-function models
- Generate nominal test cases for models without objectives
- Improved test creation for models and components when using test extensions
- Support for nested functions
- Analysis support for complex and composite signals

Simulink Desktop Real-Time

- Select Run in Kernel mode or Connected IO mode for real-time simulation

Simulink PLC Coder

- Variable-Size Data Code Generation
- Simulink PLC Coder Model Advisor Checks: Check your Simulink model for model configuration and block settings
- Externally Defined Blocks: Replace externally defined blocks during code generation
- Pure function generation

Simulink Real-Time

- Application Compiler for standalone executable instrument panels
- App Designer components for operations interfacing to real-time applications
- Bind mode for signals to instruments in Simulink Editor
- Additional target object utility functions
- Additional target object ECU and XCP page functions
- Compu method conversion for XCP blocks
- LIN Pack and LIN Unpack blocks for LIN bus data
- EtherCAT Get Device State and Set Device State blocks for control of EtherCAT terminal devices state
- Execution profiler for task schedule display
- Disk usage display
- Custom messages in system log
- ASAM XIL protocol for ECU-Test
- API for C# program control of real-time applications
- API for Python program control of real-time applications
- Parameter structure for `getparam` and `setparam` functions

- USB to serial for RS232 blocks

Simulink Report Generator

- `slreportgen.report.CFunction`: Report on C Function blocks in Report API reports
- C Function Component: Report on C Function blocks in Report Explorer reports
- Report on Stateflow ports with the Report API
- Report on Stateflow ports with Report Explorer
- `slreportgen.utils.HierarchyNumber`: Generate number that represents subsystem position in model hierarchy
- Improved reporting of configurations by data dictionary reporter
- `slreportgen.report.SystemIO` reports signal port number and block name for models and subsystems

Simulink Requirements

- Visualize traceability between Model-Based Design items with a traceability diagram
- Open the Requirements Editor from the MATLAB and Simulink apps galleries
- Customize the Requirements Editor quick access toolbar
- Merge changes to requirement set and link set files from multiple users
- Enhanced support for importing and exporting ReqIF files
- Register a custom authentication callback for integration with IBM DOORS Next or other web-based servers
- Import a subset of requirements from an IBM Rational DOORS module

Simulink Test

- Multiple release testing support for test assessments
- Customize additional default harness creation properties
- Updates to Test Sequence block and test sequence scenarios
- Remove test results for comparison signals and verify statements
- Create test harnesses for System Composer components

SoC Blockset

- Algorithm Analyzer functions support static analysis
- Updates to ADC Interface block
- Updates to PWM Interface block
- Digital I/O Interface block simulates hardware GPIO reads and writes of signals
- Code Execution Profiling Report
- **SoC Model Creator** and **SoC Builder** Tools: Create SoC model based on RFSoc reference design
- Support for C2000 F2838xD Processor

Stateflow

- Create entry and exit connections across hierarchy boundaries
- Detect rising and falling edges in data expressions
- MATLAB Function block editor in Stateflow window
- Index and assign values to arrays of structures in C action language
- String support for charts that use MATLAB as the action language

Statistics and Machine Learning Toolbox

Deployment

- Neural Network Prediction Blocks: Simulate models and generate code in Simulink

Apps

- **Cluster Data** Live Editor Task: Interactively cluster data using *k*-means clustering and generate code
- Machine Learning Apps: Compare plots across models by changing the plot layout
- Machine Learning Apps: Optimize hyperparameters of neural network models
- Classification Learner: Train Gaussian kernel classifiers for nonlinear classification of data with many observations
- Machine Learning Apps: Export models for deployment to MATLAB Production Server (requires MATLAB Compiler SDK)

Machine Learning

- `iforest` Function: Detect anomalies in data using isolation forest algorithm
- Feature Engineering: Automatically create new features before training a regression model
- Automated Model Selection: Automatically select a model with tuned hyperparameters using ASHA optimization
- Machine Learning Using Neural Networks: Optimize hyperparameters using `fitcnet` and `fitrnet`
- Generalized Additive Model (GAM): Optimize hyperparameters using `fitcgam` and `fitrgam`
- GAM for Regression: Compute the prediction interval of the response
- GPU Support: `fitctree`, `fitrtree`, `fitcensemble`, and `fitcecoc` now accept `gpuArray` inputs (requires Parallel Computing Toolbox)
- One-Hot Encoding: Encode and decode categorical data into vectors
- Incremental Learning: Naive Bayes incremental learner supports multinomial or multivariate multinomial predictor variables, and custom prediction and loss options
- Bayesian Optimization: Create an `optimizableVariable` object for a nonnegative, integer-valued, log-transformed variable

Statistics

- `ecdf` Function: Empirical cumulative distribution function enhanced to include left-censoring, double-censoring, and interval-censoring

- `mle` and `mlecov` Functions: Maximum likelihood estimation enhanced to include left-censoring, double-censoring, interval-censoring, and truncation
- Loguniform Distributions: Evaluate distributions and generate random samples using the `LoguniformDistribution` object
- `lasso` Regularization: Accelerate fitting
- GPU Support: `fitdist`, `mle`, `betafit`, `gevfit`, `gpfid`, and `nbinfit` now accept `gpuArray` inputs (requires Parallel Computing Toolbox)
- `gather` Function: Enhanced functionality

Visualization

- `shapley` Enhancements: Display query point prediction and average prediction values in a Shapley value plot

Symbolic Math Toolbox

- Next Step Suggestions: Use suggestion menus for symbolic workflows in Live Editor
- Compact Display: Command Window displays symbolic values in table, cell, and structure arrays explicitly
- Symbolic Matrix Variables: `gradient`, `inv`, `latex`, and `subs` functions accept symbolic matrix variables as input arguments
- `svd` Function: Return singular values in a column vector or diagonal matrix

System Composer

- Physical Interfaces with Simscape: Create physical interfaces, ports, and connections on components
- Simulink Subsystem Component: Add Simulink and Simscape behaviors to components
- Create Software Architectures from Existing Components
- Functions Editor: Visualize component functions in software architectures
- Value Types as Interfaces: Describe atomic piece of data
- Owned Interfaces: Define interfaces local to ports
- Architecture Views: Add port filters
- Architecture Hierarchy: Display the hierarchy of unique component types
- Class Diagrams: Display software architecture model as a class diagram
- Import and Export Software Architectures
- Test Harnesses for System Composer Components

System Identification Toolbox

- Nonlinear ARX Models: Create models that use regression functions based on machine learning algorithms
- Hammerstein-Wiener Models: Fix model linear component and nonlinearities to their known values
- Live Editor Tasks: Interactively estimate spectral models and generate MATLAB code

- Renaming of Nonlinear Model Mapping Objects

ThingSpeak

May 2021

- MQTT3 Interface

March 2021

- Upgrade to MATLAB R2021a on ThingSpeak

UAV Toolbox

- Simulate UAV Scenarios using Simulink
- GPS Sensor Block
- Video Streaming to UAV Hardware
- Fixed-Wing UAV Point Mass Block
- Obstacle Avoidance Block
- Trajectory Generation
- Flight Log Analyzer App Enhancements
- Simulation 3D Environment Upgrade: Run 3D simulations using Unreal Engine, Version 4.25
- Simulation 3D Environment Performance Improvements: Run 3D simulations faster than real-time
- Position Adjustments of Unreal Engine Cameras: Update relative translation and rotation of camera sensors during simulation
- Unreal Engine Vehicle Enhancements: Import custom meshes

Vehicle Dynamics Blockset

- Motorcycle Reference Application: Evaluate longitudinal motorcycle braking performance
- Motorcycle Blocks: Implement in-plane longitudinal motorcycles, including chain dynamics
- Tire Blocks: Implement Magic Formula models compliant with CPI and STI standards
- Powertrain Blocks: Implement transfer cases in drivetrain models and simple engines using basic specifications
- Trailer Blocks: Implement 3DOF and 6DOF trailers with one axle
- Simulation 3D Blocks: Visualize motorcycles and trailer dollies in the Unreal Engine 3D environment
- Simulation 3D Environment Upgrade: Run 3D simulations using Unreal Engine, Version 4.25
- Simulation 3D Environment Performance Improvements: Run 3D simulations faster than real-time
- Simulink Model Data: Use functions that read and write block parameter data for Excel
- Reference Applications: Generate and start maneuvers from steady-state operating points
- Japanese Language Localization Support: Block masks, parameters, variables, and error messages are available in Japanese

Vehicle Network Toolbox

- A2L Version 1.7.1: xcpA2L function creates A2L version 1.7.1 object
- MATLAB XCP: Support over CAN FD
- MDF-File Enhancements: Finalizing and sorting
- Linux Platform Support: BLF-file access
- Pack and Unpack J1939 parameter group signals
- Vehicle CAN Bus Monitor removed
- New Examples

Vision HDL Toolbox

- Multicamera Prototyping: Target multicamera designs to FPGA using Avnet Multicamera FMC module (requires Vision HDL Toolbox Support Package for Xilinx Zynq-Based Hardware)
- Multipixel-Multicomponent Streaming: Implement color space conversion and demosaic interpolation algorithms for high-frame-rate color video
- Image Normalization Example: Scale pixel values to specified output range
- Image Rotation Example: Rotate frames by small angles
- Reflection Padding: Pad image frames by reflecting around the edge pixel

WLAN Toolbox

- IEEE 802.11ba Waveform Generation
- IEEE 802.11be Support
- System-Level Simulation of IEEE 802.11 Networks
- Waveform Generation with Oversampling
- **WLAN Waveform Generator** App Updates
- MATLAB Online Support

Wavelet Toolbox

- Tunable Q-Factor Wavelet Transform: Specify your own Q-factor
- 2-D Lifting: Analyze SSCB data using lifting
- Laurent Polynomials and Laurent Matrices: Operate on Laurent functions and study `LiftingScheme` properties
- Wavelet Signal Denoiser App: Performance improvements and MATLAB Online support
- Signal Multiresolution Analyzer App: Analyze single-precision data
- Denoising: Denoise signals using wavelet methods with Signal Analyzer
- C/C++ Code Generation: Automatically generate code for wavelet functions
- Machine Learning and Deep Learning Examples: Classify signals using wavelet-derived features
- Wavelet Packets Example: Remove harmonic interference components from a signal

Wireless HDL Toolbox

- 5G SIB1 Reference Application: Implement 5G NR SIB1 recovery on SoC or ASIC
- WLAN SISO Receiver Example: Detect frame format and decode signal and data field according to WLAN standard
- DVB-S2 Physical Layer (PL) Header Recovery Example: Implement DVB-S2 HDL receiver synchronization and PL header recovery system on FPGA or ASIC
- WLAN LDPC Decoder Block: Implement decoding of LDPC codes according to WLAN standard
- CCSDS RS Decoder Block: Decode and recover messages from RS codeword according to CCSDS standard
- DVBS2 Symbol Demodulator Block: Demodulate complex constellation symbols to LLR values
- APP Decoder Block: Decode coded LLR values using MAP decoding algorithm
- 5G NR Parity-Aided Polar Codes: Encode and decode short length uplink PUCCH messages