

WLAN Toolbox™ Release Notes



MATLAB®

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

WLAN Toolbox™ Release Notes

© COPYRIGHT 2015–2021 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.

R2021a

IEEE 802.11be Single-User Support	1-2
IEEE 802.11ax HE TB Feedback NDP Support	1-2
IEEE 802.11az Positioning and Localization	1-2
WLAN Waveform Generator App Updates	1-2
Generate and Parse IEEE 802.11 MAC Trigger Frames	1-2
Write MAC Frames to PCAP or PCAPNG File	1-3
Application Traffic Pattern Generation	1-3
Model and Test IEEE 802.11ax RF Receiver with 5G interference	1-3
Model IEEE 802.11ax Residential Scenario	1-3
Measure IQ Gain and Phase Imbalance of IEEE 802.11ac Waveform	1-3
Single-Precision Support	1-3
Generate IEEE 802.11ax Waveforms with Punctured 20 MHz Subchannels	1-4
Functionality Being Removed or Changed	1-4
wlanRecoveryConfig has been removed	1-4

R2020b

IEEE 802.11az Draft 2.0 Support	2-2
Non-HT Duplicate Transmission and Reception	2-2
Spatial Reuse and BSS Coloring Support	2-2
Model and Test IEEE 802.11ax RF Transmitter	2-2
Write MAC Frames to PCAP File	2-2

MAC Frame Generator and Parser Updates	2-3
Application Traffic Pattern Generation	2-3
App-Based WLAN Waveform Generation	2-3
IEEE 802.11ax Post-FEC Padding	2-3

R2020a

IEEE 802.11ax Draft 4.1 Support	3-2
IEEE 802.11ax Trigger-Based Format	3-2
Data Recovery Functions Using New Multicore LDPC Decoding Algorithms	3-2
Transmit and Receive Signals with Unlimited Antennas	3-2
Blindly Detect, Decode, and Analyze Packets in Recovered Waveform ...	3-2
Multinode 802.11ax Downlink Throughput Comparison of OFDM and OFDMA	3-3
Single-Precision Support in WLAN Channel Models	3-3
Wireless Waveform Generator App Update	3-3
Functionality Being Removed or Changed	3-3
wlanRecoveryConfig will be removed	3-3

R2019b

IEEE 802.11ax Signal Recovery	4-2
IEEE 802.11ax Packet Extension	4-2
Detect Format of IEEE 802.11 Waveforms	4-2
Calculate PSDU and APEP Length from Transmission Time	4-2
System-Level Simulation of IEEE 802.11 Networks	4-2
IEEE 802.11ax RU Visualization	4-2
Wireless Waveform Generator App Updates	4-2

R2019a

IEEE 802.11ax Signal Recovery Functions: Decode L-SIG and HE-SIG-A fields	5-2
IEEE 802.11ay Support: Simulate 802.11ay propagation conditions	5-2
Wireless Waveform Generator App Updates	5-2
MAC Frame Parser Functions: Deaggregate A-MPDUs and decode MPDUs	5-2
IEEE 802.11ax Draft 3.1 Support	5-2
System-Level Simulation Examples: Simulate a WLAN network	5-2

R2018b

IEEE 802.11ax Support: Generate IEEE 802.11ax (Draft 2.0) high-efficiency (HE) waveforms and simulate end-to-end links	6-2
Wireless Waveform Generator App: Create, impair, visualize, and export modulated waveforms	6-2
MAC Frame Support: Generate data, management, and control frames	6-2
IEEE TGay Channel Model Example: Simulate an end-to-end, DMG format WLAN link over a TGay millimeter wave channel	6-2
IEEE 802.11ad RF Beamforming Example: Simulate an end-to-end, DMG format WLAN link with RF beamforming	6-2
IEEE 802.11p V2V Link Example: Simulate an end-to-end link over a vehicular channel model	6-3
Functionality Being Removed or Changed	6-3

R2018a

IEEE 802.11ax Examples: Generate IEEE 802.11ax high efficiency (HE) waveforms and simulate downlink end-to-end links	7-2
IEEE TGax Indoor Channel Model: Simulate indoor 802.11ax propagation conditions for both single and multiuser scenario	7-2

IEEE 802.11ad Single Carrier PHY Example: Simulate an end-to-end, DMG format WLAN link over an AWGN with synchronization and equalization	7-2
IEEE 802.11ad Extended MCS Support: Generate IEEE 802.11ad waveforms with the extended modulation and coding schemes (MCS) for the single carrier PHY	7-2

R2017b

IEEE 802.11ad Receiver: Demodulate and decode directional multi-gigabit waveforms	8-2
IEEE 802.11ah Example: Simulate an end-to-end sub 1 GHz link with the TGah channel model	8-2
Simulink Example: Model a WLAN link in Simulink	8-2
Bit-Level Processing Functions: Explore standard-compliant bit-level algorithms	8-2

R2017a

IEEE 802.11ad Support: Generate IEEE 802.11ad compliant waveforms	9-2
IEEE TGah Indoor Channel Model: Simulate 802.11ah propagation conditions	9-2
OFDM Timing Synchronization: Detect and estimate symbol timing offsets in a received OFDM signal	9-2
MATLAB Compiler Support: Compile WLAN models into standalone applications	9-2
Scrambler Bit Mapping Change: Mapping of scrambler initialization seed is swapped	9-2

R2016b

IEEE 802.11ah Support: Generate IEEE 802.11ah compliant waveforms	10-2
--------------------------------------------------------------------------------	-------------

Multuser-MIMO Receiver: Decode and analyze 802.11ac multiuser waveforms	10-2
LDPC Channel Coding: Analyze the performance of 802.11ac/n links using low-density parity-check (LDPC) channel coding techniques	10-2
Beacon Frame Generation Example: Create Beacon frames for receiver testing and over-the-air transmission	10-2
Productize functions	10-2

R2016a

Support for 802.11p Standard: Simulate 802.11p systems to analyze Intelligent Transportation Systems (ITS) applications	11-2
Support for 802.11j Standard: Simulate 802.11j systems to analyze Japanese WiFi market applications	11-2
Functionality Being Removed or Changed	11-2

R2015b+

IEEE 802.11ac and 802.11b/a/g/n standard-compliant physical layer models	12-2
Very high throughput (VHT), high throughput (HT-mixed), and legacy (non-HT) waveform generation	12-2
Channel coding, modulation (OFDM, DSSS, CCK), spatial stream mapping, and MIMO receivers	12-2
Channel models, including TGac and TGn	12-2
Measurements including channel power, spectrum mask, EVM, PER, and occupied bandwidth	12-2
Waveform transmission and reception with radio devices and instruments	12-2
C code generation support	12-2

R2021a

Version: 3.2

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11be Single-User Support

The “802.11be 4096-QAM 320 MHz Waveform Generation and Analysis” example shows how to generate an IEEE® 802.11be™ single-user waveform with a channel bandwidth of 320 MHz and 4096-point quadrature amplitude modulation (4096-QAM). The example shows how to measure transmitter modulation accuracy, spectrum emission mask, and spectral flatness.

IEEE 802.11ax HE TB Feedback NDP Support

Configure and generate high-efficiency trigger-based (HE TB) feedback null data packets (NDPs) by using the `wlanWaveformGenerator` function with the updated `wlanHETBConfig` object and its associated `getNDPFeedbackConfiguration` object function.

Demodulate received HE TB feedback NDPs by using the `wlanHEDemodulate` and `wlanHEOFDMInfo` functions with the `wlanHETBConfig` object. Recover NDP feedback status by using the `wlanHETBNDFeedbackStatus` function.

The “802.11ax Feedback Status Misdetection Simulation for Uplink Trigger-Based Feedback NDP” example shows how to measure the probability of incorrectly detecting feedback status information in an uplink HE TB feedback NDP transmission from multiple stations (STAs).

IEEE 802.11az Positioning and Localization

The “Three-Dimensional Indoor Positioning with 802.11az Fingerprinting and Deep Learning” example shows how to train a convolutional neural network (CNN) for localization and positioning by using Deep Learning Toolbox™ features and IEEE 802.11az™ data generated using WLAN Toolbox features.

The updated “802.11az Positioning Using Super-Resolution Time of Arrival Estimation” example shows how to estimate the position of a STA in a multipath environment by using a multiple signal classification (MUSIC) algorithm defined in the IEEE 802.11az standard.

WLAN Waveform Generator App Updates

The **WLAN Waveform Generator** app now supports the configuration, generation, and visualization of IEEE 802.11™ non-high-throughput (non-HT) duplicate packets.

Generate and Parse IEEE 802.11 MAC Trigger Frames

Generate WLAN medium access control (MAC) trigger frames by using the updated `wlanMACFrame` function with the `wlanMACFrameConfig`, `wlanMACTriggerConfig`, and `wlanMACTriggerUserConfig` objects.

Decode WLAN MAC trigger frames by using the updated `wlanMPDUDecode` function with the `wlanMACFrameConfig`, `wlanMACTriggerConfig`, and `wlanMACTriggerUserConfig` objects.

You can generate and decode basic, multi-user block ack request (MU-BAR), and multi-user request to send (MU-RTS) MAC trigger frames, as defined in IEEE P802.11ax™/D4.1.

Write MAC Frames to PCAP or PCAPNG File

Write generated and recovered WLAN MAC frames to a file with the .pcap or .pcapng extension by using the wlanPCAPWriter object and its associated write object function.

Application Traffic Pattern Generation

Generate a video conference application traffic pattern by using the networkTrafficVideoConference object and its associated generate object function. Accurately model real-world data traffic by using this object in system-level simulations.

Model and Test IEEE 802.11ax RF Receiver with 5G interference

The “Modeling and Testing an 802.11ax RF Receiver with 5G Interference” example shows how to characterize the impact of RF impairments, such as phase noise and power amplifier nonlinearities, on IEEE 802.11ax waveform reception with an adjacent 5G or 802.11ax interferer. The example shows how to generate baseband waveforms by using WLAN Toolbox and 5G Toolbox™ features and how to model the RF receiver by using RF Blockset™ features.

Model IEEE 802.11ax Residential Scenario

The “802.11ax Multinode System-Level Simulation of Residential Scenario Using MATLAB” example shows how to configure and simulate a multinode IEEE 802.11ax residential scenario and how to measure network statistics such as throughput, latency, and packet loss.

Measure IQ Gain and Phase Imbalance of IEEE 802.11ac Waveform

The “802.11ac Transmitter Measurements” example now shows how to measure the in-phase and quadrature (IQ) gain and phase imbalance of an IEEE 802.11ac™ waveform.

Single-Precision Support

These functions now support single-precision values for their numeric input arguments.

Function	Description	Single-Precision Support
wlanCoarseCFOEstimate	Perform coarse carrier frequency offset (CFO) estimation	All numeric input arguments
wlanFineCFOEstimate	Perform fine CFO estimation	All numeric input arguments
wlanPacketDetect	Detect orthogonal frequency-division multiplexing (OFDM) packet by using legacy short training field (L-STF)	rxSig input only
wlanSymbolTimingEstimate	Perform fine symbol timing estimation by using legacy long training field (L-LTF)	rxSig input only

Generate IEEE 802.11ax Waveforms with Punctured 20 MHz Subchannels

Configure 80 MHz and 160 MHz high-efficiency multi-user (HE MU) transmissions with punctured 20 MHz subchannels by setting the `PrimarySubchannel` and `AllocationIndex` properties of the updated `wlanHEMUConfig` object. You can configure the primary 20 MHz subchannel and puncture other 20 MHz subchannels for efficient coexistence with legacy IEEE 802.11 networks.

Functionality Being Removed or Changed

`wlanRecoveryConfig` has been removed

Errors

The `wlanRecoveryConfig` object has been removed.

Starting in R2020a, all functions that supported the `wlanRecoveryConfig` object as an input argument support name-value arguments corresponding to the properties of the object. To parameterize an affected function, use one or more of these name-value arguments instead of the `wlanRecoveryConfig` object as an input argument. This table lists the affected functions, their syntaxes that are no longer supported, and the recommended replacement syntax for each case.

Function	Removed Syntax	Recommended Replacement Syntax
<code>wlanFormatDetect</code>	<code>wlanFormatDetect(rxSig, chEst, noiseVarEst, cbw, cfgRec)</code> , where <code>cfgRec</code> is a <code>wlanRecoveryConfig</code> object	<code>wlanFormatDetect(rxSig, chEst, noiseVarEst, cbw, Name, Value)</code> , where <code>Name, Value</code> specifies one or more name-value arguments
<code>wlanHTDataRecover</code>	<code>wlanHTDataRecover(rxSig, chEst, noiseVarEst, cfg, cfgRec)</code> , where <code>cfgRec</code> is a <code>wlanRecoveryConfig</code> object	<code>wlanHTDataRecover(rxSig, chEst, noiseVarEst, cfg, Name, Value)</code> , where <code>Name, Value</code> specifies one or more name-value arguments
<code>wlanHTSIGRecover</code>	<code>wlanHTSIGRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec)</code> , where <code>cfgRec</code> is a <code>wlanRecoveryConfig</code> object	<code>wlanHTSIGRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value)</code> , where <code>Name, Value</code> specifies one or more name-value arguments
<code>wlanLSIGRecover</code>	<code>wlanLSIGRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec)</code> , where <code>cfgRec</code> is a <code>wlanRecoveryConfig</code> object	<code>wlanLSIGRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value)</code> , where <code>Name, Value</code> specifies one or more name-value arguments
<code>wlanNonHTDataRecover</code>	<code>wlanNonHTDataRecover(rxSig, chEst, noiseVarEst, cfg, cfgRec)</code> , where <code>cfgRec</code> is a <code>wlanRecoveryConfig</code> object	<code>wlanNonHTDataRecover(rxSig, chEst, noiseVarEst, cfg, Name, Value)</code> , where <code>Name, Value</code> specifies one or more name-value arguments

Function	Removed Syntax	Recommended Replacement Syntax
wlanVHTDataRecover	wlanVHTDataRecover(__, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTDataRecover(__, Name, Value), where Name, Value specifies one or more name-value arguments
wlanVHTSIGARRecover	wlanVHTSIGARRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTSIGARRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value), where Name, Value specifies one or more name-value arguments
wlanVHTSIGBRecover	wlanVHTSIGBRecover(__, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTSIGBRecover(__, Name, Value), where Name, Value specifies one or more name-value arguments

R2020b

Version: 3.1

New Features

Bug Fixes

IEEE 802.11az Draft 2.0 Support

The 802.11az Waveform Generation example shows how to parameterize and generate HE ranging null data packet (NDP) waveforms with optional secure long training fields (LTFs), as defined in IEEE P802.11az/D2.0.

The 802.11az Indoor Positioning Using Time of Arrival example shows how to estimate the position of an indoor device by using the time of arrival and trilateration technique.

Non-HT Duplicate Transmission and Reception

Configure and generate IEEE 802.11 non-high-throughput (non-HT) duplicate packets by using the `wlanWaveformGenerator` function with the updated `wlanNonHTConfig` object. You can also configure and generate the Data field of a non-HT duplicate transmission by using the `wlanNonHTData` function.

Recover data from non-HT duplicate transmissions by using these functions: `wlanFieldIndices`, `wlanInterpretScramblerState`, `wlanNonHTDataBitRecover`, `wlanNonHTDataRecover`, `wlanNonHTOFDMDemodulate`, and `wlanNonHTOFDMInfo`.

Spatial Reuse and BSS Coloring Support

The Spatial Reuse with BSS Coloring in 802.11ax Residential Scenario example shows how to perform spatial reuse with basic service set (BSS) coloring, as specified in IEEE P802.11ax/D4.1, in a system-level simulation. The example shows how to measure the throughput of each BSS and of the network by simulating overlapping BSS packet-detection-based spatial reuse operations.

Model and Test IEEE 802.11ax RF Transmitter

The Modeling and Testing an 802.11ax RF Transmitter example characterizes the impact of RF impairments, such as in-phase and quadrature imbalance, phase noise, and power amplifier nonlinearities, on IEEE 802.11ax waveform transmission. The example demonstrates how to measure the constellation, error vector magnitude (EVM), and spectrum attributes such as the mask, occupied bandwidth, channel power, and complementary cumulative distribution function (CCDF) by using WLAN Toolbox and RF Blockset features.

Write MAC Frames to PCAP File

Write generated and recovered WLAN medium access control (MAC) frames to a file with the `.pcap` or `.pcapng` extension by using the `pcapWriter` or `pcapngWriter` object, respectively, and the associated `write` object function.

With the `pcapWriter` object, you can configure the file name and add a timestamp to each packet.

With the `pcapngWriter` object, you can configure the file name, add a timestamp and metadata for each packet, and write custom data.

MAC Frame Generator and Parser Updates

Generate WLAN MAC frames with four address fields by using the `wlanMACFrame` function with the updated `wlanMACFrameConfig` object. Parse MAC frames with four address fields by using MAC frame parser functions `wlanAMPDUDeaggregate` and `wlanMPDUDecode`.

The MAC frame parser functions now support these additional IEEE 802.11 physical layer (PHY) configurations.

- Parse MAC frames in HE multi-user (HE MU) or HE trigger-based (HE TB) transmissions by using the parser functions with the `wlanHEMUConfig` or `wlanHETBConfig` object, respectively.
- Parse MAC frames recovered from HE single-user (HE SU), extended-range SU (HE ER SU), or HE MU transmissions by using the parser functions with the `wlanHERecoveryConfig` object.

Suppress all warnings when attempting to parse unsupported MAC frames by specifying the 'SuppressWarnings' name-value pair argument of the parser functions. For a list of supported MAC frame formats, see the 802.11 MAC Frame Generation example.

Application Traffic Pattern Generation

Generate a file transfer protocol (FTP), a voice over internet protocol (VoIP), or an On-Off application traffic pattern by using the `networkTrafficFTP`, `networkTrafficVoIP`, or `networkTrafficOnOff` object, respectively, and the associated `generate` object function. You can use these objects in system-level simulations to accurately model real-world data traffic.

The Generate and Visualize FTP Application Traffic Pattern example shows how to generate an FTP application traffic pattern based on the IEEE 802.11ax Evaluation Methodology and the 3GPP TR 36.814 specification.

App-Based WLAN Waveform Generation

The **WLAN Waveform Generator** app now supports configuration, generation, and visualization of:

- HE TB waveforms, as defined in IEEE P802.11ax/D4.1
- All supported IEEE 802.11 waveforms with an arbitrary number of transmit antennas

The App-Based WLAN Waveform Generation example outlines waveform configuration, generation, and visualization workflows.

IEEE 802.11ax Post-FEC Padding

Generate HE waveforms with post-forward-error-correction (post-FEC) padding bits by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig`, `wlanHEMUConfig`, `wlanHEMUUser`, or `wlanHETBConfig` configuration object. To determine the number of required post-FEC padding bits for each format, use the `getNumPostFECPaddingBits` object function for the corresponding configuration object. You can configure the source of the post-FEC padding bits to be the current global random number stream, the `mt19937ar` algorithm, or custom post-FEC padding bits.

R2020a

Version: 3.0

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11ax Draft 4.1 Support

Generate high-efficiency single-user (HE SU) null data packets (NDPs) with preamble puncturing, as defined in IEEE P802.11ax/D4.1, by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig` object.

IEEE 802.11ax Trigger-Based Format

Configure and generate high-efficiency trigger-based (HE TB) waveforms by using the `wlanWaveformGenerator` function with the `wlanHETBConfig` object.

Demodulate and decode HE TB waveforms by using the `wlanHEDemodulate` and `wlanHEDataBitRecover` functions.

The updated 802.11ax Packet Error Rate Simulation for Uplink Trigger-Based Format example demonstrates minimum-mean-square-error-based ordered successive interference cancellation (MMSE-SIC) equalization.

Data Recovery Functions Using New Multicore LDPC Decoding Algorithms

Reduce simulation times by using low-density parity-check (LDPC) decoding with multicore processing in these data recovery functions: `wlanDMGDataBitRecover`, `wlanDMGHeaderBitRecover`, `wlanHEDataBitRecover`, `wlanHTDataRecover`, and `wlanVHTDataRecover`.

These functions support the layered belief propagation, normalized min-sum, and offset min-sum LDPC decoding algorithms.

Transmit and Receive Signals with Unlimited Antennas

Transmit and receive WLAN signals with an arbitrary number of antennas for any of these formats.

- HE SU
- HE extended range SU (HE ER SU)
- HE multi-user (HE MU)
- HE TB
- Very high throughput (VHT)
- High throughput (HT)
- Non-HT

Filter a signal through a multipath fading channel with an arbitrary number of links by using these updated channel models: `wlanTGacChannel`, `wlanTGahChannel`, `wlanTGaxChannel`, and `wlanTGnChannel`.

Blindly Detect, Decode, and Analyze Packets in Recovered Waveform

The Recover and Analyze Packets in 802.11ax Waveform example demonstrates how to detect, decode, and analyze 802.11a™ and 802.11ax packets in a recovered waveform. The receiver can

decode OFDM non-HT, HE MU, HE SU, and HE ER SU packets without knowledge of any transmission parameters aside from the channel bandwidth. You can configure the example to capture over-the-air waveforms by using software-defined radio (SDR) devices.

Multinode 802.11ax Downlink Throughput Comparison of OFDM and OFDMA

The 802.11ax Downlink Throughput Comparison of OFDM and OFDMA Through System-level Simulation example shows how to configure and simulate multinode downlink OFDM and OFDMA 802.11ax transmissions by using SimEvents®, Stateflow®, and WLAN Toolbox software.

Single-Precision Support in WLAN Channel Models

Reduce simulation times and memory consumption by specifying applicable properties of these channel models with single precision: wlanTGacChannel, wlanTGahChannel, wlanTGaxChannel, and wlanTGnChannel.

Wireless Waveform Generator App Update

Generate HE waveforms with packet extension by using the **Wireless Waveform Generator** app.

Functionality Being Removed or Changed

wlanRecoveryConfig will be removed

Warns

The wlanRecoveryConfig object will be removed in a future release.

Starting in R2020a, all functions that support a wlanRecoveryConfig object as an input argument now support name-value pair arguments corresponding to the properties of the object. To parameterize an affected function, use one or more of these name-value pairs instead of a wlanRecoveryConfig object. This table lists the affected functions, their syntaxes that are no longer recommended, and the recommended replacement syntax for each case.

Function	Not Recommended	Recommended
wlanFormatDetect	wlanFormatDetect(rxSig, chEst, noiseVarEst, cbw, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanFormatDetect(rxSig, chEst, noiseVarEst, cbw, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanHTDataRecover	wlanHTDataRecover(rxSig, chEst, noiseVarEst, cfg, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanHTDataRecover(rxSig, chEst, noiseVarEst, cfg, Name, Value), where Name, Value specifies one or more name-value pair arguments

Function	Not Recommended	Recommended
wlanHTSIGRecover	wlanHTSIGRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanHTSIGRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanLSIGRecover	wlanLSIGRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanLSIGRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanNonHTDataRecover	wlanNonHTDataRecover(rxSig, chEst, noiseVarEst, cfg, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanNonHTDataRecover(rxSig, chEst, noiseVarEst, cfg, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanVHTDataRecover	wlanVHTDataRecover(__, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTDataRecover(__, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanVHTSIGARRecover	wlanVHTSIGARRecover(rxSig, chEst, noiseVarEst, cbw, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTSIGARRecover(rxSig, chEst, noiseVarEst, cbw, Name, Value), where Name, Value specifies one or more name-value pair arguments
wlanVHTSIGBRecover	wlanVHTSIGBRecover(__, cfgRec), where cfgRec is a wlanRecoveryConfig object	wlanVHTSIGBRecover(__, Name, Value), where Name, Value specifies one or more name-value pair arguments

R2019b

Version: 2.2

New Features

IEEE 802.11ax Signal Recovery

Decode the HE-SIG-B common and user fields in a high-efficiency multiuser (HE MU) transmission by using the `wlanHESIGBCommonBitRecover` and `wlanHESIGBUserBitRecover` functions.

IEEE 802.11ax Packet Extension

Generate HE-format packets with packet extension, as specified in Draft 3.1 of the IEEE 802.11ax standard, by using the `wlanWaveformGenerator` function with the updated `wlanHESUConfig` and `wlanHEMUConfig` objects.

Detect Format of IEEE 802.11 Waveforms

The `wlanFormatDetect` function now detects these 802.11ax packet formats in a received waveform: HE single-user (HE SU), HE extended-range single-user (HE ER SU), HE multiuser (HE MU), and HE trigger-based (HE TB).

Calculate PSDU and APEP Length from Transmission Time

You can now generate single-user IEEE 802.11 waveforms for a given transmission duration or number of symbols.

- Calculate PSDU length by using the `wlanPSDULength` function.
- Calculate APEP length by using the `wlanAPEPLength` function.

System-Level Simulation of IEEE 802.11 Networks

These examples show how to simulate multinode 802.11 networks by using `SimEvents`, `Stateflow`, and `WLAN Toolbox`.

- 802.11 MAC and Application Throughput Measurement - Measure MAC and application layer throughput in an 802.11a/n/ac/ax network.
- 802.11 MAC QoS Traffic Scheduling - Simulate quality of service (QoS) traffic scheduling in an 802.11a/n/ac/ax network.
- 802.11ax System-Level Simulation with Physical Layer Abstraction and Physical Layer Abstraction for System-Level Simulation - Model a large 802.11ax network by using physical layer abstraction.

IEEE 802.11ax RU Visualization

Display the resource unit (RU) allocation for HE-format transmissions by using the `showAllocation` object function with the `wlanHESUConfig` and `wlanHEMUConfig` objects.

Wireless Waveform Generator App Updates

You can now export a WLAN waveform generated using the **Wireless Waveform Generator** app as a MATLAB® script with a `.m` extension. You can recreate the waveform by running the exported script in MATLAB.

R2019a

Version: 2.1

New Features

IEEE 802.11ax Signal Recovery Functions: Decode L-SIG and HE-SIG-A fields

You can now decode the L-SIG and HE-SIG-A fields for IEEE 802.11ax format. The 802.11ax Signal Recovery with Preamble Decoding example shows how to recover single-user and multi-user 802.11ax signals by using the `wlanHERecoveryConfig` object and the `wlanLSIGBitRecover` and `wlanHESIGABitRecover` functions.

IEEE 802.11ay Support: Simulate 802.11ay propagation conditions

Use the `wlanTGayChannel` System object™ to simulate 802.11ay™ propagation conditions as demonstrated in the 802.11ad Packet Error Rate Single Carrier PHY Simulation with TGay Channel example. The ray-tracing-based channel model, described by Task Group ay, supports antenna array specification, SU-SISO and SU-MIMO, unpolarized and polarized signals, and beamforming. For more information, see WLAN Channel Models and Propagation Channel Models.

Wireless Waveform Generator App Updates

With the **Wireless Waveform Generator** app, you can now:

- Create WLAN waveforms that are compliant with 802.11ax modulation schemes.
- Generate a waveform that you can transmit with a connected lab test instrument. The Wireless Waveform Generator app can generate waveforms for instruments supported by the `rfsiggen` function. Use of the transmit feature in the Wireless Waveform Generator app requires Instrument Control Toolbox™.

MAC Frame Parser Functions: Deaggregate A-MPDUs and decode MPDUs

You can now parse WLAN medium access control (MAC) frames. Decode MAC protocol data units (MPDUs) with the `wlanMPDUDecode` function and deaggregate aggregated MPDUs (A-MPDUs) with the `wlanAMPDUDeaggregate` function.

This feature is demonstrated in the added 802.11 MAC Frame Decoding example and the updated 802.11 OFDM Beacon Receiver with Live Data example.

IEEE 802.11ax Draft 3.1 Support

With the added IEEE 802.11ax support, you can:

- Generate HE-format WLAN MAC frames compliant with IEEE 802.11ax Draft 3.1 using the `wlanMACFrame` function.
- Simulate an HE-MU packet with more than 16 HE-SIG-B symbols.
- Generate HE packets containing a PSDU of up to 6,500,631 bytes.

System-Level Simulation Examples: Simulate a WLAN network

These examples show you how to simulate 802.11 networks by simulating individual links between different nodes:

-
- 802.11ax PHY-Focused System-Level Simulation
 - Multi-Node 802.11a Network Modeling with PHY and MAC

R2018b

Version: 2.0

New Features

Bug Fixes

IEEE 802.11ax Support: Generate IEEE 802.11ax (Draft 2.0) high-efficiency (HE) waveforms and simulate end-to-end links

WLAN Toolbox now supports 802.11ax Draft 2.0 waveform generation and link-level simulation of single user (HE-SU, HE-EXT-SU) and multi-user (HE-MU) formats.

The 802.11ax Packet Error Rate Simulation for Uplink Trigger-Based Format example shows how to measure the packet error rate of an uplink trigger-based (HE-TB) format link.

Wireless Waveform Generator App: Create, impair, visualize, and export modulated waveforms

Using the **Wireless Waveform Generator** app you can:

- Create OFDM, QAM, or PSK modulated waveforms or a sine wave test waveform.
- Create WLAN waveforms that are compliant with various 802.11 modulation schemes (a/b/g/j/p/n/ac/ad/ah).
- Add RF impairments to your waveforms.
- Plot waveforms in constellation diagram, spectrum analyzer, and time scopes.
- Export waveforms from the app to your MATLAB workspace or a file.

MAC Frame Support: Generate data, management, and control frames

This release adds support for inbuilt MAC Frame generation. This feature includes the following functions:

- `wlanMACFrame`: WLAN MAC frame generation (MPDU or A-MPDU)
- `wlanMSDULengths`: Returns a vector of MSDU lengths required to generate a MAC frame of given length, MAC configuration, and PHY configuration.
- `wlanMACFrameConfig`: MAC frame configuration object with properties relevant to the IEEE 802.11 MAC frame.
- `wlanMACManagementConfig`: MAC management configuration object with properties relevant to the IEEE 802.11 MAC management frame body.

IEEE TGay Channel Model Example: Simulate an end-to-end, DMG format WLAN link over a TGay millimeter wave channel

The 802.11ad Packet Error Rate Single Carrier PHY Simulation with TGay Channel example shows how to determine the packet error rate for an 802.11ad™ SC link with a TGay fading channel at a selection of SNR points.

IEEE 802.11ad RF Beamforming Example: Simulate an end-to-end, DMG format WLAN link with RF beamforming

The 802.11ad Single Carrier Link with RF Beamforming in Simulink example shows how to model an IEEE 802.11ad single carrier link with RF beamforming. This example also displays the packet error rate after processing each packet, the equalized symbol constellation, and EVM measurements.

IEEE 802.11p V2V Link Example: Simulate an end-to-end link over a vehicular channel model

The 802.11p Packet Error Rate Simulation for a Vehicular Channel example shows how to measure the packet error rate of an IEEE 802.11p™ link with a V2V fading channel and additive white Gaussian noise.

Functionality Being Removed or Changed

The `wlanGeneratorConfig` object has been removed. Use `Name, Value` pair syntax of `wlanWaveformGenerator` for parameterizing the `wlanWaveformGenerator` function. See `wlanWaveformGenerator` for the recommended parameter `Name, Value` pair syntax.

R2018a

Version: 1.5

New Features

IEEE 802.11ax Examples: Generate IEEE 802.11ax high efficiency (HE) waveforms and simulate downlink end-to-end links

This release adds three examples which show how WLAN System Toolbox functions can be used to simulate IEEE 802.11ax Draft 1.1:

- 802.11ax Parameterization for Waveform Generation and Simulation shows how to parameterize and generate different types of IEEE 802.11ax high efficiency (HE) formats.
- 802.11ax Packet Error Rate Simulation for Single User Format shows how to measure the packet error rate of an IEEE 802.11ax high efficiency (HE) single user format link.
- 802.11ax OFDMA and Multi-User MIMO Throughput Simulation shows the transmit and receive processing for an IEEE 802.11ax multi-user downlink transmission over a TGax indoor fading channel. Three transmission modes are simulated: OFDMA, MU-MIMO, and a combination of OFDMA and MU-MIMO.

IEEE TGax Indoor Channel Model: Simulate indoor 802.11ax propagation conditions for both single and multiuser scenario

In this release, WLAN System Toolbox™ provides the channel model described by the TGax task group. Use the `wlanTGaxChannel` System object to model 802.11ax indoor propagation conditions. For more information, see WLAN Channel Models and Propagation Channel.

IEEE 802.11ad Single Carrier PHY Example: Simulate an end-to-end, DMG format WLAN link over an AWGN with synchronization and equalization

This example enables you to study synchronization and equalization techniques in an end-to-end IEEE 802.11ad Single Carrier PHY link.

IEEE 802.11ad Extended MCS Support: Generate IEEE 802.11ad waveforms with the extended modulation and coding schemes (MCS) for the single carrier PHY

The toolbox now supports the extended modulation and coding schemes (MCS) for the IEEE 802.11ad single carrier PHY as specified in IEEE 802.11-2016.

R2017b

Version: 1.4

New Features

IEEE 802.11ad Receiver: Demodulate and decode directional multi-gigabit waveforms

This release adds support for IEEE 802.11ad demodulation and decoding steps. The 802.11ad specification defines the WLAN directional multi-gigabit (DMG) format.

This feature is contained in these functions:

- `wlanDMGDataBitRecover`: Recovers data bits from DMG Data field.
- `wlanDMGHeaderBitRecover`: Recovers header bits from DMG Header field.

Three new IEEE 802.11ad examples explain how to perform an end-to-end simulation in the different PHY types.

IEEE 802.11ah Example: Simulate an end-to-end sub 1 GHz link with the TGah channel model

802.11ah Packet Error Rate Simulation for 2x2 TGah Channel shows how to measure the packet error rate of an IEEE 802.11ah™ S1G short preamble link with a fading TGah indoor channel model and additive white Gaussian noise.

Simulink Example: Model a WLAN link in Simulink

WLAN Link Modeling in Simulink: Use the end-to-end 802.11n™ HT link example to explore the capabilities of WLAN System Toolbox in Simulink®.

Bit-Level Processing Functions: Explore standard-compliant bit-level algorithms

WLAN Toolbox includes standard compliant bit-level functions:

- `wlanScramble`: Scramble and descramble binary input.
- `wlanBCCInterleave` and `wlanBCCDeinterleave`: Interleave and deinterleave binary convolutionally encoded input.
- `wlanBCCEncode` and `wlanBCCDecode`: Convolutionally encode and decode input.
- `wlanStreamParse` and `wlanStreamDeparse`: Stream-parse and deparse binary input.
- `wlanSegmentParseBits` and `wlanSegmentDeparseBits`: Segment-parse and deparse binary input.
- `wlanSegmentParseSymbols` and `wlanSegmentDeparseSymbols`: Segment-parse and deparse symbols.
- `wlanConstellationMap` and `wlanConstellationDemap`: Modulate and demodulate input.

You can explore these functions and customize their algorithms and functionality.

R2017a

Version: 1.3

New Features

Bug Fixes

Compatibility Considerations

IEEE 802.11ad Support: Generate IEEE 802.11ad compliant waveforms

This release adds support for IEEE 802.11ad waveform generation. The 802.11ad specification defines the WLAN directional multi-gigabit (DMG) format. For details, see `wlanDMGConfig`.

IEEE TGah Indoor Channel Model: Simulate 802.11ah propagation conditions

In this release, WLAN System Toolbox provides the channel model described by the TGah task group. Use the `wlanTGahChannel` System object to model 802.11ah propagation conditions. For more information, see [WLAN Channel Models and Propagation Channel](#).

OFDM Timing Synchronization: Detect and estimate symbol timing offsets in a received OFDM signal

A WLAN OFDM timing synchronization function, `wlanSymbolTimingEstimate`, returns the symbol timing estimate of a received OFDM waveform.

MATLAB Compiler Support: Compile WLAN models into standalone applications

WLAN System Toolbox now supports MATLAB Compiler™ enabling you to produce standalone executables. For details, see [Code Generation and Deployment](#).

Scrambler Bit Mapping Change: Mapping of scrambler initialization seed is swapped

The scrambling operation now swaps the MSB-to-LSB mapping for the initialization seed for all supported WLAN packet formats. This change aligns the scrambler operation with the mapping indicated for 802.11ad. These functions perform the scrambling operation `wlanWaveformGenerator`, `wlanNonHTData`, `wlanHTData`, and `wlanVHTData`.

R2016b

Version: 1.2

New Features

IEEE 802.11ah Support: Generate IEEE 802.11ah compliant waveforms

This release adds support for IEEE 802.11ah waveform generation. For details, see `wlanSIGConfig` and 802.11ah Waveform Generation.

Multuser-MIMO Receiver: Decode and analyze 802.11ac multuser waveforms

This release adds support for IEEE 802.11ac MU-MIMO receiver. For details, see 802.11ac Multi-User MIMO Precoding.

LDPC Channel Coding: Analyze the performance of 802.11ac/n links using low-density parity-check (LDPC) channel coding techniques

This release adds LDPC support for IEEE 802.11ac and 802.11n. For details, see 802.11n Packet Error Rate Simulation for 2x2 TGn Channel.

Beacon Frame Generation Example: Create Beacon frames for receiver testing and over-the-air transmission

This release adds an example that creates and transmits a WLAN OFDM Beacon. You can transmit Beacon frames using an SDR platform, and then receive and view the Beacon using a standard Wi-Fi® device. For details, see 802.11 OFDM Beacon Receiver with USRP Hardware.

Productize functions

WLAN packet detection and format detection functions are productized in this release.

- Packet detection — Use the function `wlanPacketDetect` to return the offset from the start of the input waveform to the start of the detected preamble.
- Format detection — Use the function `wlanFormatDetect` to detect and return the packet format for the specified received signal.

R2016a

Version: 1.1

New Features

Bug Fixes

Support for 802.11p Standard: Simulate 802.11p systems to analyze Intelligent Transportation Systems (ITS) applications

The toolbox now supports IEEE 802.11p communications systems. Transmit and receive OFDM functions now include options for 10 MHz or 5 MHz channel bandwidth. For examples on how to use the 802.11p standard, see:

- 802.11p and 802.11a Packet Error Rate Simulations — Measures packet error rates in 802.11p and 802.11a links using an end-to-end simulation with a fading channel and AWGN.
- 802.11p Spectral Emission Mask Testing — Performs spectrum emission mask tests for an 802.11p transmitted waveform.

Support for 802.11j Standard: Simulate 802.11j systems to analyze Japanese WiFi market applications

Support is enabled for IEEE 802.11j™ communications systems. Transmit and receive OFDM functions now provide option for 10 MHz channel bandwidth.

Functionality Being Removed or Changed

Use of the `wlanGeneratorConfig` object is discouraged for parameterizing the `wlanWaveformGenerator` function. See `wlanWaveformGenerator` for the recommended parameter `Name, Value` pair syntax.

Functionality	Result	Use Instead	Compatibility Considerations
<code>wlanWaveformGenerator(bits, cfgFormat, cfgWaveGen)</code> where <code>cfgWaveGen</code> is a <code>wlanGeneratorConfig</code> object.	Still runs	<code>wlanWaveformGenerator(bits, cfgFormat, Name, Value, ..., NameN, ValueN)</code>	Replace all instances of old function syntax with new function syntax.

R2015b+

Version: 1.0

New Features

IEEE 802.11ac and 802.11b/a/g/n standard-compliant physical layer models

WLAN System Toolbox provides standard-compliant functions for the design, simulation, and verification of IEEE 802.11 b/a/g/n/ac communications systems.

Very high throughput (VHT), high throughput (HT-mixed), and legacy (non-HT) waveform generation

WLAN System Toolbox provides 802.11 standard-compliant waveform generation for VHT, HT, and non-HT formats.

For more information, see [Waveform Generation and Signal Transmission](#).

Channel coding, modulation (OFDM, DSSS, CCK), spatial stream mapping, and MIMO receivers

WLAN System Toolbox provides 802.11 standard-compliant functions for individual preamble field creation and decoding, OFDM demodulation and channel estimation, carrier frequency offset estimation, and data recovery. Single input single output (SISO) and multiple input multiple output (MIMO) antenna configurations are supported.

For more information, see [Packet Recovery and Signal Reception](#).

Channel models, including TGac and TGn

WLAN System Toolbox provides channel models described by the TGac and TGn task groups. For more information, see [WLAN Channel Models and Propagation Channel](#).

Measurements including channel power, spectrum mask, EVM, PER, and occupied bandwidth

WLAN System Toolbox examples show how to measure signal characteristics and system performance.

Waveform transmission and reception with radio devices and instruments

When coupled with a radio hardware support package, WLAN System Toolbox functions enable transmission and reception of packet contents with radio devices and instruments.

C code generation support

All WLAN System Toolbox functions support C code generation. Also, you can access the underlying code to customize the algorithms and functions.