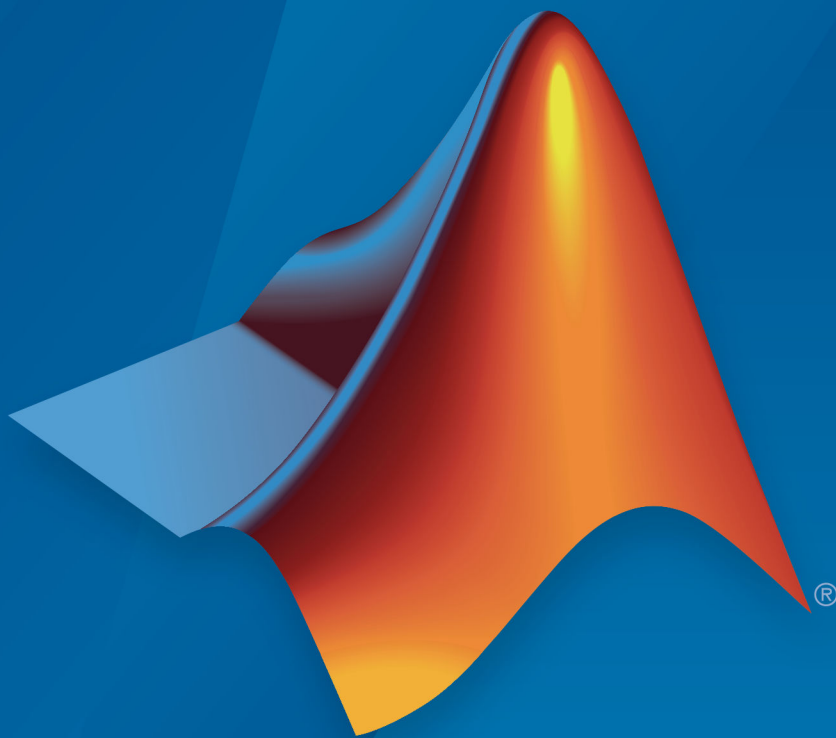


Sensor Fusion and Tracking Toolbox™ Release Notes



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.
3 Apple Hill Drive
Natick, MA 01760-2098

Sensor Fusion and Tracking Toolbox™ Release Notes

© COPYRIGHT 2018 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.

R2018b

Single-Hypothesis and Multi-Hypothesis Multi-Object Trackers	1-2
Estimation Filters for Tracking	1-2
Inertial Sensor Fusion to Estimate Pose	1-2
Active and Passive Sensor Models	1-3
Trajectory and Scenario Generation	1-4
Visualization and Analytics	1-4
Orientation, Rotations, and Representation Conversions	1-4
Sensor Fusion and Tracking Examples	1-6

R2018b

Version: 1.0

New Features

Single-Hypothesis and Multi-Hypothesis Multi-Object Trackers

Sensor Fusion and Tracking Toolbox provides multi-object trackers that fuse information from various sensors. Use `trackerGNN` to maintain a single hypothesis about the objects it tracks. Use `trackerTOMHT` to maintain multiple hypotheses about the objects it tracks.

Estimation Filters for Tracking

Sensor Fusion and Tracking Toolbox provides estimation filters that are optimized for specific scenarios, such as linear or nonlinear motion models, linear or nonlinear measurement models, or incomplete observability.

Estimation filters include:

Estimate Filters	Description
<code>trackingABF</code>	Alpha-beta filter
<code>trackingKF</code>	Linear Kalman filter
<code>trackingEKF</code>	Extended Kalman filter
<code>trackingUKF</code>	Unscented Kalman filter
<code>trackingCKF</code>	Cubature Kalman filter
<code>trackingPF</code>	Particle filter
<code>trackingMSCEKF</code>	Extended Kalman filter in modified spherical coordinates
<code>trackingGSF</code>	Gaussian-sum filter
<code>trackingIMM</code>	Interacting multiple model filter

Inertial Sensor Fusion to Estimate Pose

Sensor Fusion and Tracking Toolbox provides algorithms to estimate orientation and position from IMU and GPS data. The algorithms are optimized for different sensor configurations, output requirements, and motion constraints.

Inertial sensor fusion algorithms include:

Inertial Sensor Fusion Algorithm	Description
<code>ecompass</code>	Estimate orientation using magnetometer and accelerometer readings.
<code>imufilter</code>	Estimate orientation using accelerometer and gyroscope readings
<code>ahrsfilter</code>	Estimate orientation using accelerometer, gyroscope, and magnetometer readings
<code>insfilter</code>	Estimate position and orientation (pose) using IMU and GPS readings.

Active and Passive Sensor Models

Sensor Fusion and Tracking Toolbox provides active and passive sensor models. You can mimic environmental, channel, and sensor configurations by modifying parameters of the sensor models. For active sensors, you can model the corresponding emitters and channels as separate models.

Sensor models include:

Sensor Model	Description
<code>imuSensor</code>	IMU measurements of accelerometer, gyroscope, and magnetometer
<code>gpsSensor</code>	GPS position, velocity, groundspeed, and course measurements
<code>insSensor</code>	INS/GPS position, velocity, and orientation emulator
<code>monostaticRadarSensor</code>	Radar detection generator
<code>sonarSensor</code>	Active or passive sonar detection generator
<code>irSensor</code>	Infrared (IR) detection generator
<code>radarSensor</code>	Radio frequency detection generator

Trajectory and Scenario Generation

Generate ground-truth trajectories to drive sensor models using the `kinematicTrajectory` and `waypointTrajectory` System objects. Simulate tracking of multiple platforms in a 3-D arena using `trackingScenario`.

Visualization and Analytics

Use `theaterPlot` with `trackingScenario` to plot the ground-truth pose, detections, and estimated pose tracks for multi-object scenarios. Get error metrics for tracks using `trackErrorMetrics`. Analyze and compare the performance of multi-object tracking systems using `trackAssignmentMetrics`.

Orientation, Rotations, and Representation Conversions

The quaternion data type enables efficient representation of orientation and rotations. Sensor Fusion and Tracking Toolbox provides the following functions for use with the quaternion data type:

Rotations	
<code>rotateframe</code>	Quaternion frame rotation
<code>rotatepoint</code>	Quaternion point rotation

Representation Conversion	
<code>rotmat</code>	Convert quaternion to rotation matrix
<code>rotvec</code>	Convert quaternion to rotation vector (radians)
<code>rotvecd</code>	Convert quaternion to rotation vector (degrees)
<code>parts</code>	Extract quaternion parts
<code>euler</code>	Convert quaternion to Euler angles (radians)
<code>eulerd</code>	Convert quaternion to Euler angles (degrees)
<code>compact</code>	Convert quaternion array to N-by-4 matrix

Metrics and Interpolation	
dist	Angular distance in radians
norm	Quaternion norm
meanrot	Quaternion mean rotation
slerp	Spherical linear interpolation

Initialization and Convenience Functions	
ones	Create quaternion array with real parts set to one and imaginary parts set to zero
zeros	Create quaternion array with all parts set to zero
classUnderlying	Class of parts within quaternion
normalize	Quaternion normalization

Mathematics	
times, .*	Element-wise quaternion multiplication
mtimes, *	Quaternion multiplication
prod	Product of a quaternion array
minus, -	Quaternion subtraction
uminus, -	Quaternion unary minus
conj	Complex conjugate of quaternion
ldivide, .\	Element-wise quaternion left division
rdivide, ./	Element-wise quaternion right division
exp	Exponential of quaternion array
log	Natural logarithm of quaternion array
power, .^	Element-wise quaternion power

Array Manipulation	
ctranspose, '	Complex conjugate transpose of quaternion array
transpose, .'	Transpose of quaternion array

Sensor Fusion and Tracking Examples

The release of Sensor Fusion and Tracking Toolbox includes the following examples.

Applications
"Air Traffic Control"
"Multiplatform Radar Detection Fusion"
"Passive Ranging Using a Single Maneuvering Sensor"
"Tracking Using Distributed Synchronous Passive Sensors"
"Search and Track Scheduling for Multifunction Phased Array Radar"
"Extended Object Tracking"
"Visual-Inertial Odometry Using Synthetic Data"
"IMU and GPS Fusion for Inertial Navigation"

Multi-Object Trackers
"Multiplatform Radar Detection Fusion"
"Tracking Closely Spaced Targets Under Ambiguity"
"Tracking Using Distributed Synchronous Passive Sensors"
"Extended Object Tracking"
"Introduction to Using the Global Nearest Neighbor Tracker"
"Introduction to Track Logic"

Estimation Filters
"Tracking Maneuvering Targets"
"Tracking with Range-Only Measurements"
"Passive Ranging Using a Single Maneuvering Sensor"

Inertial Sensor Fusion
"Estimate Orientation Through Inertial Sensor Fusion"
"IMU and GPS Fusion for Inertial Navigation"
"Estimate Position and Orientation of a Ground Vehicle"

Inertial Sensor Fusion
“Estimate Orientation and Height Using IMU, Magnetometer, and Altimeter”
Sensor Models
“Inertial Sensor Noise Analysis Using Allan Variance”
“Simulating Passive Radar Sensors and Radar Interferences”
“Introduction to Simulating IMU Measurements”
“Introduction to Tracking Scenario and Simulating Radar Detections”
“Scanning Radar Mode Configuration”
Trajectory and Scenario Generation
“Introduction to Tracking Scenario and Simulating Radar Detections”
“Benchmark Trajectories for Multi-Object Tracking”
“Multiplatform Radar Detection Generation”
Quaternion Representation
“Rotations, Orientation and Quaternions”
“Lowpass Filter Orientation Using Quaternion SLERP”

