

MATLAB Coder

Generate C and C++ code from MATLAB code

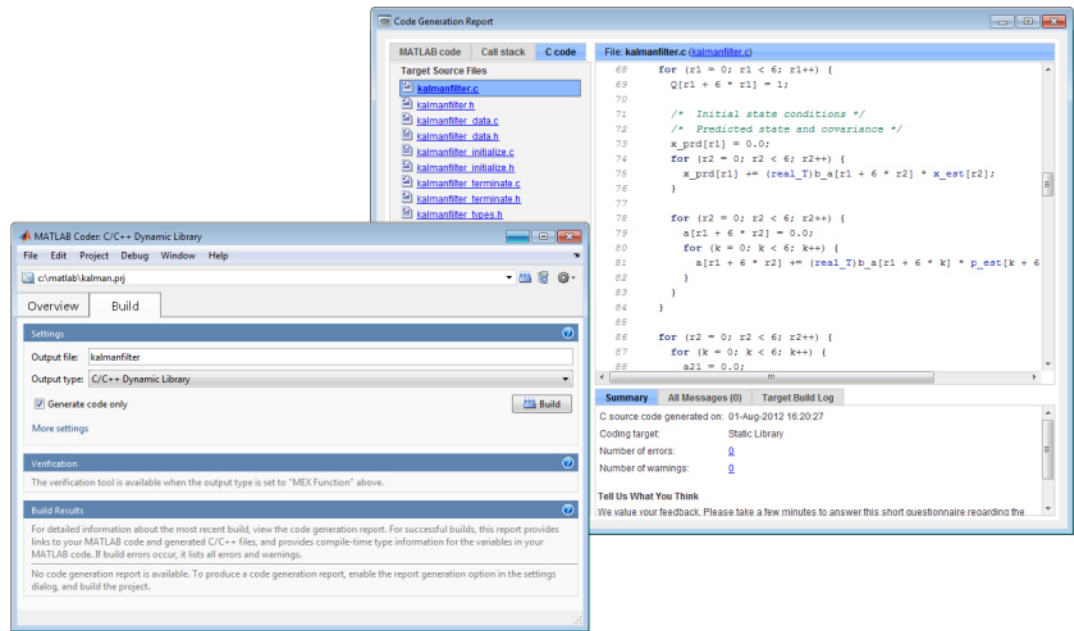
MATLAB Coder™ generates standalone C and C++ code from MATLAB® code. The generated source code is portable and readable. MATLAB Coder supports a subset of core MATLAB language features, including program control constructs, functions, and matrix operations. It can generate MEX functions that let you accelerate computationally intensive portions of MATLAB code and verify the behavior of the generated code.

Key Features

- ANSI/ISO compliant C and C++ code generation
- [MEX function generation](#) for fixed-point and floating-point math
- Project management tool for specifying entry points, input data properties, and other code-generation configuration options
- Static or dynamic memory allocation for variable-size data
- Code generation support for many functions and System objects™ in [Communications System Toolbox™](#), [Computer Vision System Toolbox™](#), [DSP System Toolbox™](#), and [Phased Array System Toolbox™](#)
- Support for [common language matlab features](#), including matrix operations, subscripting, program control statements (if, switch, for, while), and structures

MATLAB Coder works with [Simulink Coder™](#) and [Embedded Coder®](#) to generate C code from [Simulink®](#) models that contain MATLAB code. The generated C code can be used for:

- Standalone execution
- Integration with other software
- Accelerating MATLAB algorithms
- Embedded implementation



MATLAB Coder project manager interface (left) and code generation report (right) showing generated C code.

MATLAB Language Subset for Code Generation

MATLAB Coder generates code from a [subset of the MATLAB language](#) typically used by design engineers for developing algorithms as components of larger systems. The supported subset includes [more than 400 operators and functions](#) from MATLAB.

A limited set of functions are supported for code generation from [Aerospace Toolbox](#), [Fixed-Point Designer™](#), [Image Processing Toolbox™](#), [Signal Processing Toolbox™](#) and [Statistics Toolbox™](#). MATLAB Coder also supports code generation from many [System objects™](#) found in the [Communications System Toolbox™](#), [Computer Vision System Toolbox™](#), [DSP System Toolbox™](#), and [Phased Array System Toolbox™](#).

MATLAB Coder supports a [broad range of MATLAB language attributes](#) to develop algorithms. These include:

Matrices and arrays, such as persistent variables, global variables, matrix operations, N-dimensional arrays, subscripting, and frames

Classes and data types, such as complex numbers, integer match, double/single precision, fixed-point arithmetic, characters, structures, numeric classes, and variable-sized data

Programming constructs, such as MATLAB program control statements (if, for, while, switch) and arithmetic, relational, and logical operators

Functions, such as function handles, sub-functions, the subset of MATLAB functions, variable length input and output argument lists, and the ability to call MATLAB functions

See the full list of [supported MATLAB language attributes for code generation](#).

Use Cases for MATLAB Coder

MATLAB Coder lets design engineers developing algorithms in [MATLAB](#), generate [readable and portable C/C++ code](#). With this generated code, you can:

- Prototype your MATLAB algorithms as a [standalone executable](#) on PCs and communicate them as design specifications to your software engineers
- Accelerate parts of your simulation by [replacing the MATLAB code with an automatically generated MATLAB executable \(MEX function\)](#) that calls the compiled C or C++ code
- Integrate your MATLAB algorithms as a compiled library component into other software such as a custom simulator
- Implement and verify your MATLAB algorithms on an embedded processor



MATLAB Coder use cases.

Generating Code from MATLAB

Translating MATLAB algorithms to C code involves specifying implementation requirements. MATLAB Coder guides you through this iterative process while enabling you to stay within the MATLAB language. You ensure that your algorithm is compliant for code generation by [generating a MEX function](#) that wraps the compiled code for execution back within MATLAB. MATLAB Coder produces a report that identifies any errors which must be fixed to make your MATLAB algorithm compliant for code generation. You iterate between fixing errors and regenerating a MEX function until your MATLAB algorithm is suitable for code generation.

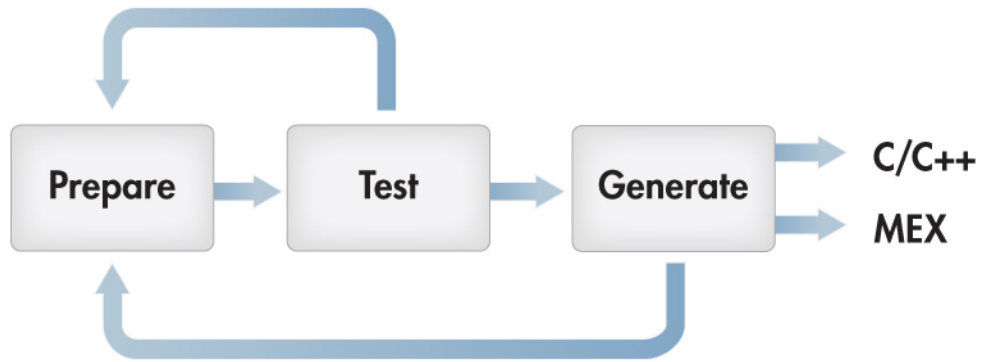
You can now generate either C or C++ source code or a MEX function tuned for performance from your MATLAB algorithm.

C or C++ source code enables you to:

- Prototype on PCs
- Create a library
- Implement as embedded code

The MEX function can be called in place of the original MATLAB code to:

- Verify the compiled code back in MATLAB
- Accelerate the execution



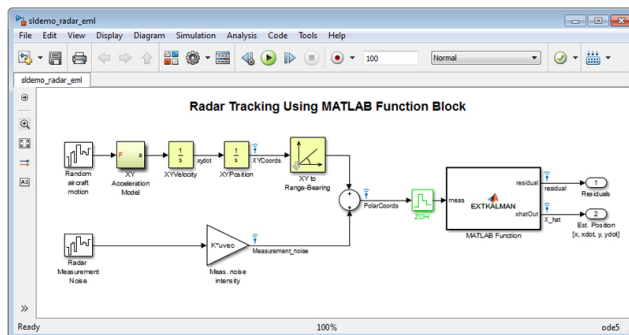
Three-step iterative workflow for generating code from MATLAB using MATLAB Coder.

MATLAB Coder with Simulink Coder and Embedded Coder

The MATLAB Function block for simulation and code generation lets you add MATLAB algorithms written in the MATLAB subset for integrating MATLAB code into Simulink models. Simulink Coder lets you generate code from these Simulink models that contain MATLAB code.

```

28 % Radar update time deltat is inherited from model workspace
29
30 % 1. Compute Phi, Q, and R
31 Phi = [1 deltat 0 0; 0 1 0 0; 0 0 1 deltat; 0 0 0 1];
32 Q = diag([0 .005 0 .005]);
33 R = diag([300^2 0.001^2]);
34
35 % 2. Propagate the covariance matrix:
36 P = Phi*P*Phi' + Q;
37
38 % 3. Propagate the track estimate::
39 xhat = Phi*xhat;
40
41 % 4 a). Compute observation estimates:
42 Rangehat = sqrt(xhat(1)^2+xhat(3)^2);
43 Bearinghat = atan2(xhat(3),xhat(1));
44
45 % 4 b). Compute observation vector y and linearized measurement m
46 yhat = [Rangehat;
47         Bearinghat];
48 M = [ cos(Bearinghat)      0 sin(Bearinghat)      0
49       -sin(Bearinghat)/Rangehat 0 cos(Bearinghat)/Rangehat 0 ];
50
51 % 4 c). Compute residual (Estimation Error)
52 residual = meas - yhat;
  
```



Radar tracking model in Simulink. The model implements a Kalman filtering algorithm that is written in MATLAB and called using the MATLAB Function block.

Embedded Coder generates code for supported embedded processors, on-target rapid prototyping boards, and microprocessors used in mass production. It extends MATLAB Coder and Simulink Coder by providing configuration options and advanced optimizations for fine-grain control of the generated code's functions, files, and data. Embedded Coder improves code efficiency and facilitates integration with legacy code, data types, and calibration parameters used in production.

Code generated with Embedded Coder can be exported into third-party development environments, enabling you to automate the creation of an executable to deploy on your embedded system. The generated code can also be executed on processors to verify behavioral performance and gather resource utilization metrics through processor-in-the-loop (PIL) and profiling techniques.

Resources

Product Details, Examples, and System Requirements

www.mathworks.com/products/matlab-coder

Trial Software

www.mathworks.com/trialrequest

Sales

www.mathworks.com/contactsales

Technical Support

www.mathworks.com/support

Online User Community

www.mathworks.com/matlabcentral

Training Services

www.mathworks.com/training

Third-Party Products and Services

www.mathworks.com/connections

Worldwide Contacts

www.mathworks.com/contact