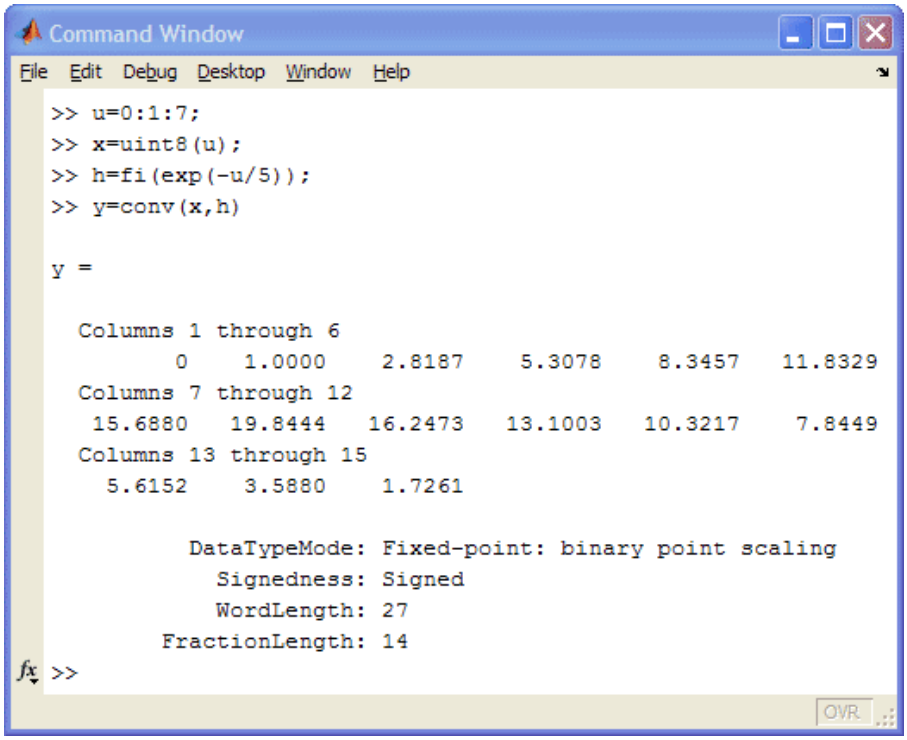
Fixed-Point Toolbox 3.0

Design and execute fixed-point algorithms and analyze fixed-point data

Fixed-Point Toolbox™ provides fixed-point data types and arithmetic in MATLAB®. The toolbox lets you design fixed-point algorithms using MATLAB syntax and execute them at compiled C-code speed. You can reuse these algorithms in Simulink® and pass fixed-point data to and from Simulink models, facilitating bit-true simulation, implementation, and analysis. The toolbox also enables you to generate test sequences for fixed-point software and hardware verification.

Key Features

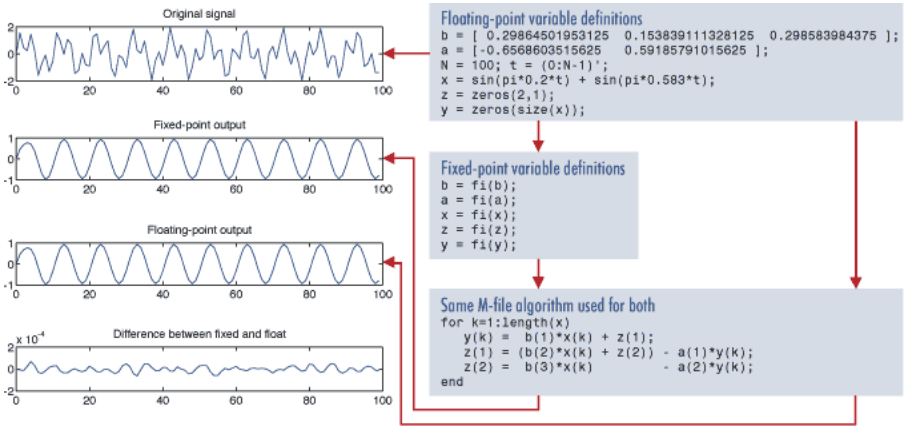
- Fixed-point data types in MATLAB with word lengths up to 65,535 bits
- Global and local settings for performing fixed-point arithmetic
- Logical and bitwise operators and native integers
- Fixed-point data types usable in both MATLAB and Simulink
- Data logging, data-type override, and other tools for floating-to-fixed-point conversion
- Accelerated execution of fixed-point algorithms in MATLAB



Example of fixed-point computation in MATLAB: Convolution of two fixed-point variables.

Fixed-Point Data Types

Together, Fixed-Point Toolbox and MATLAB provide an environment for developing, implementing, and verifying algorithms for fixed-point designs. The toolbox supports arbitrary word lengths from 2 to 65,535 bits for fixed-point data types, enabling you to capture bit-true fixed-point behaviors in the MATLAB workspace.



Plot of the algorithm output in both 16-bit fixed point and double-precision floating point. Using Fixed-Point Toolbox, you can develop data-type-independent algorithms and run them with fixed-point variables.

Fixed-Point Computation

Fixed-Point Toolbox uses MATLAB syntax to support fixed-point computation. The toolbox provides the following functions and operators:

- Basic arithmetic, relational, logical, and bitwise operations
- Square root and convolution functions
- Statistical functions, such as `min` and `max`
- User-selected overflow and rounding modes to govern all arithmetic

The toolbox lets you use global or local settings for performing fixed-point arithmetic. When using global settings, fixed-point variables in the MATLAB session share a single set of arithmetic attributes. Global settings facilitate the process of converting a design from floating-point to fixed-point. When using local settings, fixed-point variables have their own attached arithmetic attributes. Local settings provide the user with better control over the details of each arithmetic operation.

There are four modes for performing fixed-point computations:

- Fully automatic mode, enabling you to use up to 65,535 bits for the sum or product
- Fully specified mode, letting you indicate the word length and binary point location for the sum or product
- Two partially automatic modes, in which the binary point location is set automatically but you can specify the word length of the sum or product

Using the plotting and visualization functions in MATLAB, you can visualize fixed-point data objects created in Fixed-Point Toolbox and analyze the results of your fixed-point design.

Fixed-Point Data Exchange Between MATLAB and Simulink

The fixed-point operations in the toolbox complement the fixed-point capabilities of Simulink. You can use fixed-point data objects to pass fixed-point data between the MATLAB workspace and Simulink models. The toolbox also supports the design, analysis, and implementation of fixed-point digital filters with Filter Design

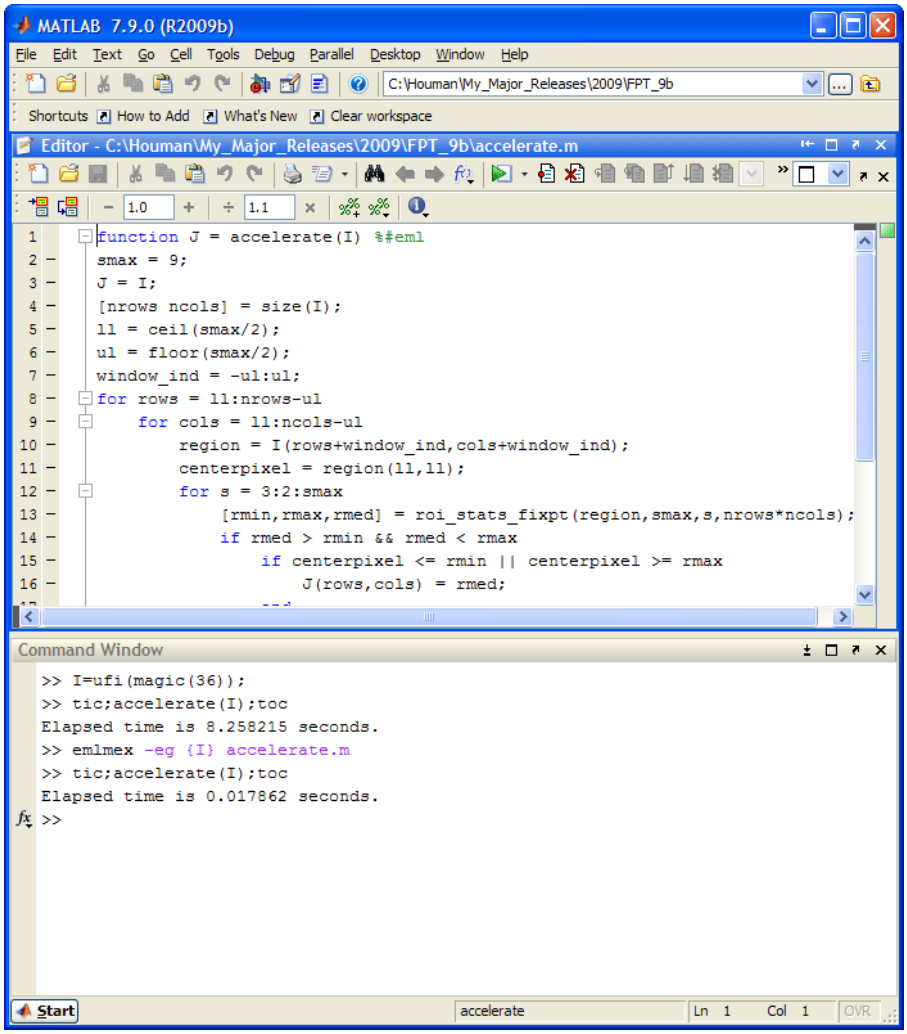
Toolbox™. You can quantize filter coefficients and data with Fixed-Point Toolbox and then use Filter Design Toolbox functions to construct appropriate filters to process the data.

Floating-to-Fixed-Point Conversion

Fixed-Point Toolbox provides analysis tools for efficiently converting a design from floating-point to fixed-point representation. The data logging tools let you record minimum and maximum data values and pinpoint when overflow and underflow occur during fixed-point operations. The data-type override tools let you analyze a fixed-point algorithm by switching the data type of variables between floating-point and fixed-point. With this analysis, you can observe the dynamic range of variables involved in your MATLAB code and ensure that the algorithm behaves consistently in floating-point and fixed-point representations.

Accelerated Execution of Fixed-Point Algorithms with Embedded MATLAB Subset

You can significantly accelerate the execution speed of fixed-point algorithms using the Embedded MATLAB™ language subset. The Embedded MATLAB subset consists of more than 270 MATLAB operators and functions and more than 100 Fixed-Point Toolbox functions. In the toolbox, the `emlmex` function generates a compiled MATLAB executable (MEX) version of your Embedded MATLAB code. The MEX version runs at compiled C-code speed, which is faster than the original MATLAB code. Using Real-Time Workshop®, you can then generate pure integer C code from your Embedded MATLAB code.



Embedded MATLAB function compiled using the `emlmex` function in the toolbox. Once compiled, the Embedded MATLAB code runs at compiled C-code speed.

Resources

Product Details, Demos, and System Requirements
www.mathworks.com/products/fixed

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