Tailoring Tools for Application-Oriented Development Using Model-Based Design

Tom Erkkinen
MathWorks
The Origins of MATLAB

By Cleve Moler, MathWorks

In the late 1970s, following Wirth’s methodology, I used Fortran and portions of LINPACK and EISPACK to develop the first version of MATLAB. The only data type was “matrix.” The HELP command listed all of the available functions, with their names abbreviated.

<table>
<thead>
<tr>
<th>ABS</th>
<th>ANS</th>
<th>ATAN</th>
<th>BASE</th>
<th>CHAR</th>
<th>CHOL</th>
<th>CHOP</th>
<th>CLEA</th>
<th>COND</th>
<th>CONJ</th>
<th>COS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET</td>
<td>DIAG</td>
<td>DIAR</td>
<td>DISP</td>
<td>EDIT</td>
<td>EIG</td>
<td>ELSE</td>
<td>END</td>
<td>EPS</td>
<td>EXEC</td>
<td>EXIT</td>
</tr>
<tr>
<td>EXP</td>
<td>EYE</td>
<td>FILE</td>
<td>FLOP</td>
<td>FLPS</td>
<td>FOR</td>
<td>FUN</td>
<td>HESS</td>
<td>HILB</td>
<td>IF</td>
<td>IMAG</td>
</tr>
<tr>
<td>INV</td>
<td>KRON</td>
<td>LINE</td>
<td>LOAD</td>
<td>LOG</td>
<td>LONG</td>
<td>LU</td>
<td>MACR</td>
<td>MAG</td>
<td>NORM</td>
<td>ONES</td>
</tr>
<tr>
<td>ORTH</td>
<td>PINV</td>
<td>PLOT</td>
<td>POLY</td>
<td>PRIN</td>
<td>PROD</td>
<td>QR</td>
<td>RAND</td>
<td>RANK</td>
<td>RCON</td>
<td>RAT</td>
</tr>
<tr>
<td>REAL</td>
<td>RETU</td>
<td>RREF</td>
<td>ROOT</td>
<td>ROUN</td>
<td>SAVE</td>
<td>SCHU</td>
<td>SHOR</td>
<td>SEMI</td>
<td>SIN</td>
<td>SIZE</td>
</tr>
<tr>
<td>SQRT</td>
<td>STOP</td>
<td>SUM</td>
<td>SVD</td>
<td>TRIL</td>
<td>TRIU</td>
<td>USER</td>
<td>WHAT</td>
<td>WHIL</td>
<td>WHO</td>
<td>WHY</td>
</tr>
</tbody>
</table>

There were only 80 functions. There were no M-files or toolboxes.

The bald and not excessively bald and not excessively smart hamster obeyed a terrified and not excessively terrified hamster.

www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html
Early MathWorks Landscape

- Simulink
- MATLAB
- Blocksets (external)
- Toolboxes (external)
Today’s Landscape

Event-Based Modeling

Physical Modeling

Simulation Graphics and Reporting

System, and Test

Code Generation

Embedded Coder

Deployment

Database Access and Reporting

MathWorks Toolboxes and Products

- Statistics and Machine Learning Toolbox
- Computer Vision System Toolbox
- Robotics System Toolbox
- Simscape
- HDL Coder

MathWorks Toolboxes and Products

- Filter by Source
  - MATLAB: 24,003
  - Simulink: 3,694
  - Polyspace: 10

Community Toolboxes

- GUI Layout Toolbox
- Deep Neural Network
- JSONlab: a toolbox to encode/decode JSON files
- PIVlab: time-resolved particle image velocimetry (PIV) tool
- Numerical Computing with MATLAB

Toolbox containing filters and apps from Numerical Computing with MATLAB.
Today’s Landscape – Prolific
Creating Apps

![MATLAB R2017a](https://www.mathworks.com/products/matlab/

**Package App**

C:\Users\terkkine\MATLAB\Projects\alexamples\ConVehRefApp\ConVeh\Untitled1.png

**Pick main file**  Describe your app

**Package into installation file**

Output folder

C:\Users\terkkine\MATLAB\Projects\alexexamples\ConVehRefApp\ConVeh\Untitled1.png

**Package**
User Apps

The Adoption of MATLAB Apps and Toolboxes at Jaguar Land Rover

Engineering teams across Jaguar Land Rover rely on MATLAB based apps and toolboxes to develop a wide range of advanced vehicles. This session discusses...

Evolving Landscape

User Apps

Application-Oriented Products

Simulink

MATLAB

Blocksets (external)

Toolboxes (external)

Toolboxes

Blocksets
Application-Oriented Products

Why Application Oriented?

- Basic blocks and functions no longer suffice
- Starting reference point is necessary
- System models are important but hard
- Let your experts focus on their expertise
  - Focus on own component, leverage elsewhere

Cost savings of nearly $2 million per year
Introducing … Products for Application-Oriented Development

Powertrain Blockset
Capabilities
- Model a Powertrain System
  Use fully assembled reference applications and parameterize the components
  ➤ Learn more
- Design and Test a Controller Model
  Use built-in controller models, user-defined controller models, and embedded estimators
  ➤ Learn more
- Perform System Design Tradeoff
  Perform powertrain design tradeoff studies, such as emissions, fuel economy, and performance.
  ➤ Learn more
- Deploy for Hardware-in-the-Loop Testing
  Perform HIL testing on your controller model.
  ➤ Learn more

Automated Driving System Toolbox
Capabilities
- Ground Truth Labeling
  Automatically label ground truth data, and compare output from the algorithm under test with ground-truth data.
  ➤ Learn more
- Sensor Fusion and Tracking
  Perform multisensor fusion using multijoint tracking framework with Kalman filters.
  ➤ Learn more
- Vision System Design
  Develop computer vision algorithms for vehicle and pedestrian detection, lane detection, and classification.
  ➤ Learn more
- Scenario Generation
  Generate traffic scenarios and simulate radar and camera sensor outputs to test sensor fusion and control algorithms.
  ➤ Learn more
Reference Applications – Starting Point

Powertrain Blockset
Reference Applications – Starting Point

Automated Driving System Toolbox
Workflow Integration - Design

Powertrain Blockset

Automated Driving System Toolbox
Workflow Integration - Code

Powertrain Blockset

Automated Driving System Toolbox

```
%% Code Generation for Tracking and Sensor Fusion
function [Vx,Vy] = calculateGroundSpeed(Vxi,Vyi,egoSpeed)
% Inputs
% (Vxi,Vyi) : relative object speed
% egoSpeed : ego vehicle speed
% Outputs
% [Vx,Vy] : ground object speed

Vx = Vxi + egoSpeed;  % calculate longitudinal ground speed
theta = atan2(Vyi,Vxi);  % calculate heading angle
Vy = Vx * tan(theta);  % calculate lateral ground speed
end
```
Workflow Integration - Verification

Powertrain Blockset

Automated Driving System Toolbox
Open and Customizable - Tailoring

Powertrain Blockset

Automated Driving System Toolbox
We have other application products
Workflow integration – Your workflow
Your Landscape

The Origins of Your ???

By You

>>Why