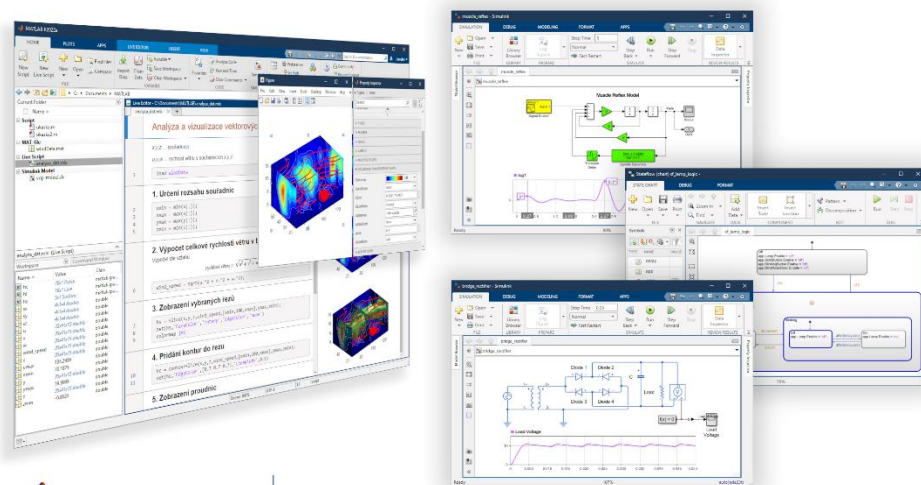


11.9.2025 Technical Computing Camp 2025

Nový MATLAB Desktop a ďalšie aktuality v prostredí MATLAB



Michal Blaho

blaho@humusoft.sk

www.humusoft.cz

info@humusoft.cz

www.mathworks.com



Turning Everyday Cell Phones
into Search-and-Rescue
Beacons



Chatty little marmosets call each
other by name



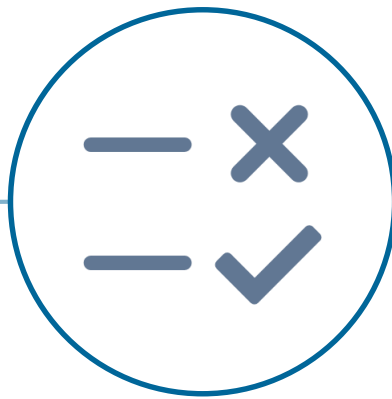
Zuri's VTOL Will Revolutionize
Regional Air Transportation



MATLAB®
& SIMULINK®



**Objavovanie a
tvorba**



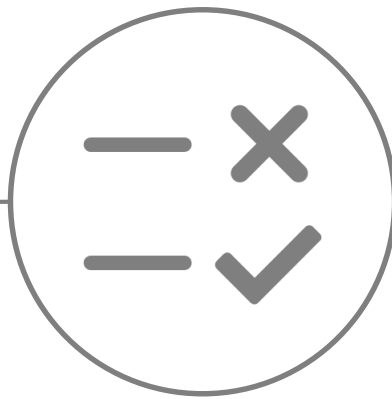
**Integrácia a
verifikácia**



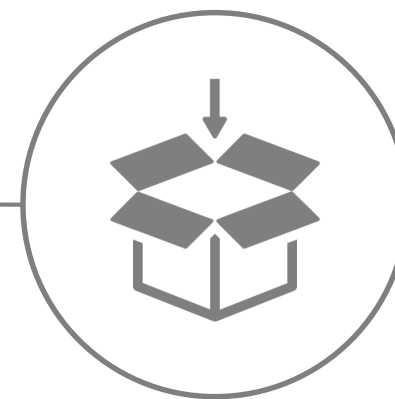
**Zdieľanie a
nasadenie**



**Objavovanie a
tvorba**

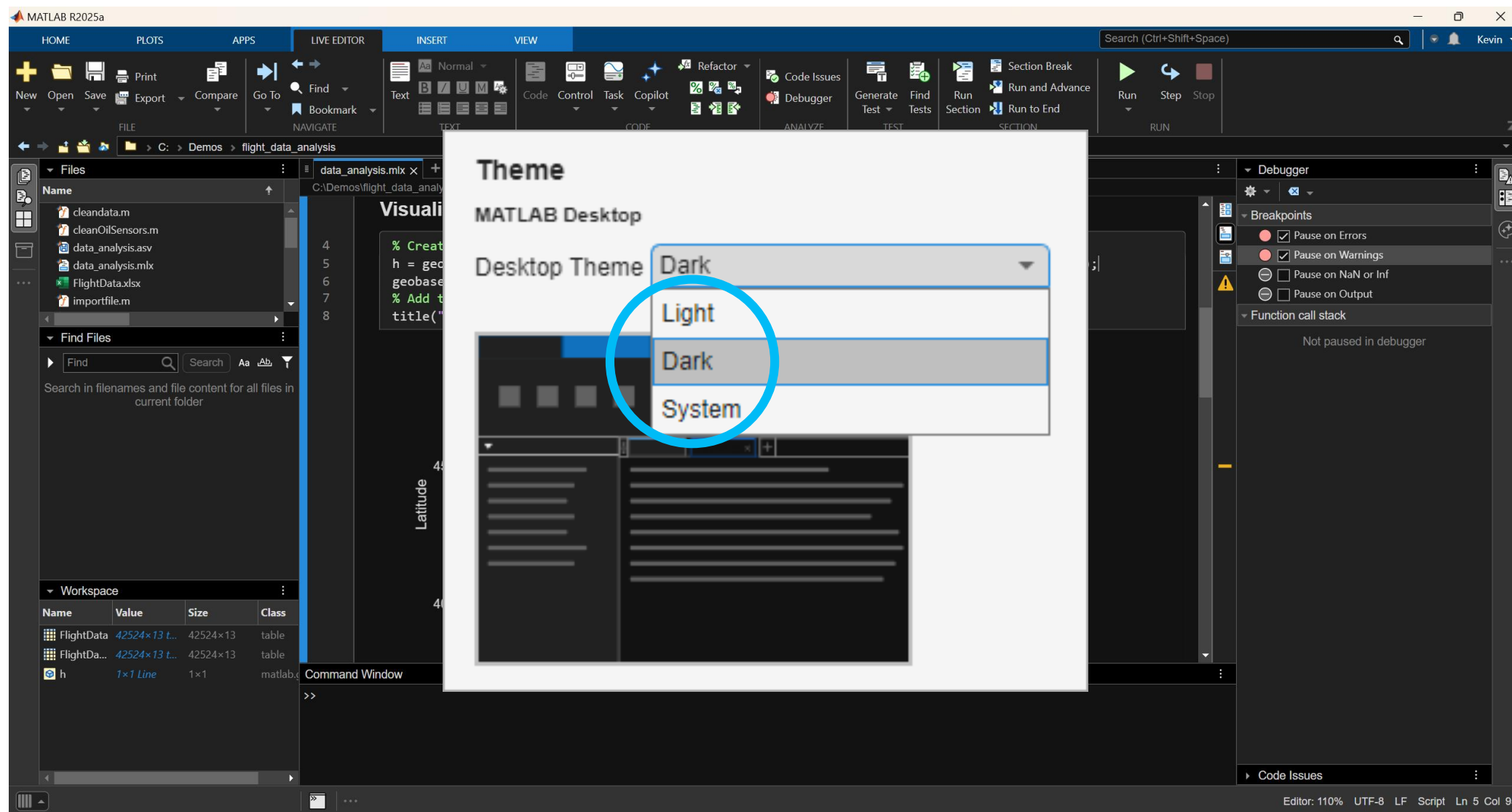


**Integrácia a
verifikácia**



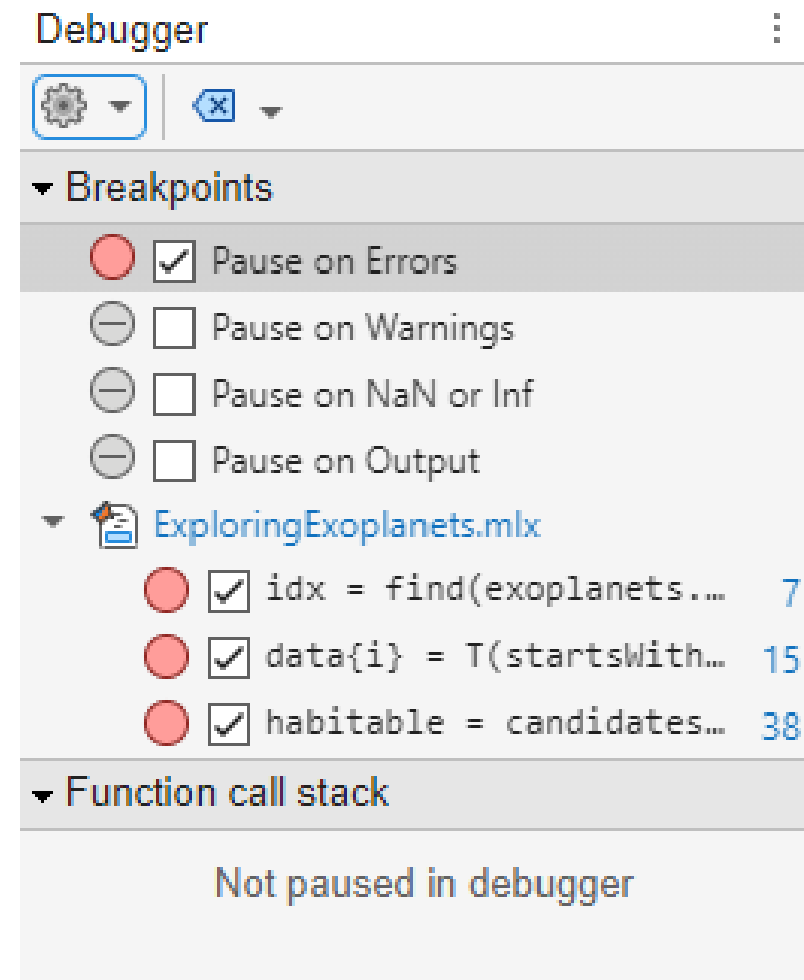
**Zdieľanie a
nasadenie**

Nové témy prostředí



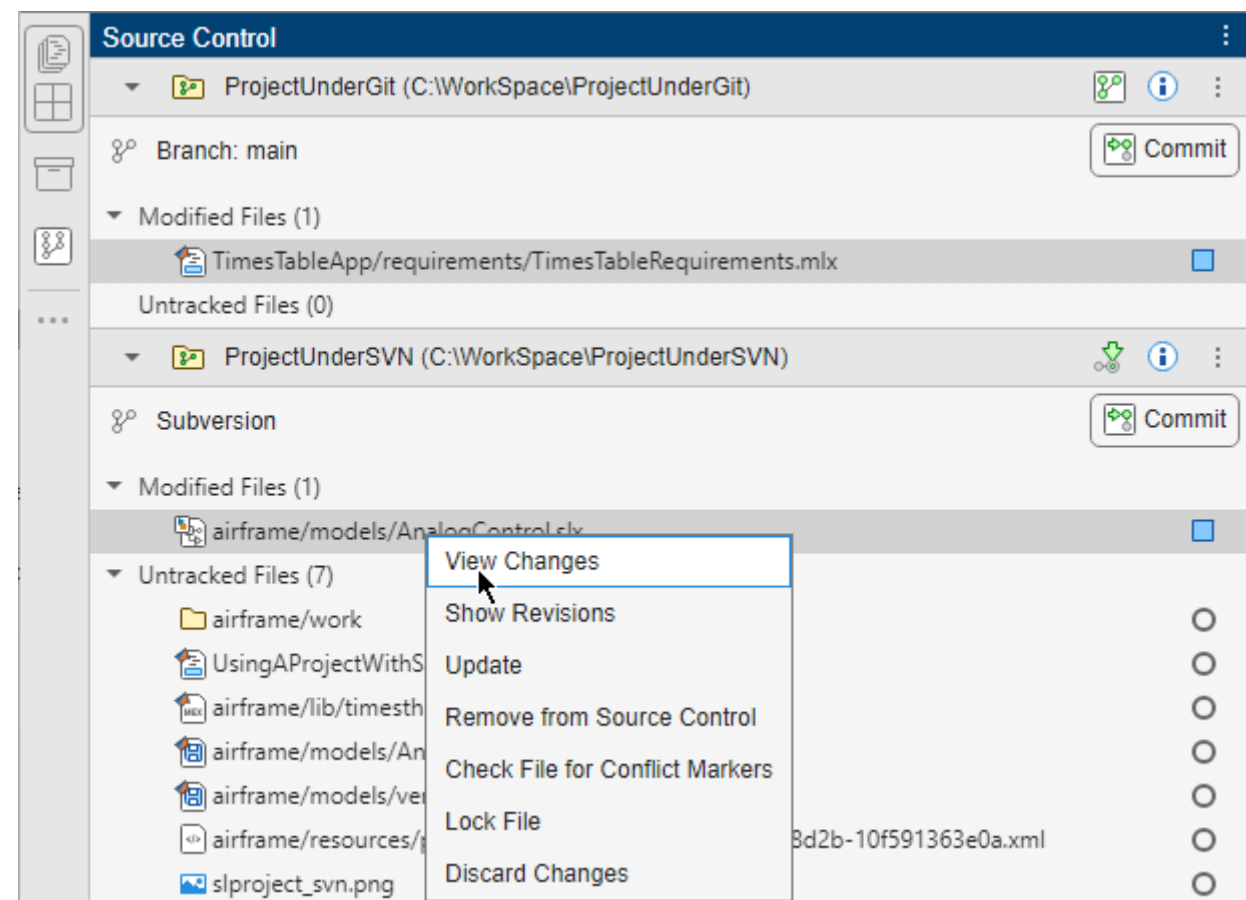
Nové témy prostredia umožňuje ľahký prístup k funkciám

- Nové panely pre rýchle programovanie a úlohy vývoja softvéru
 - Debugger



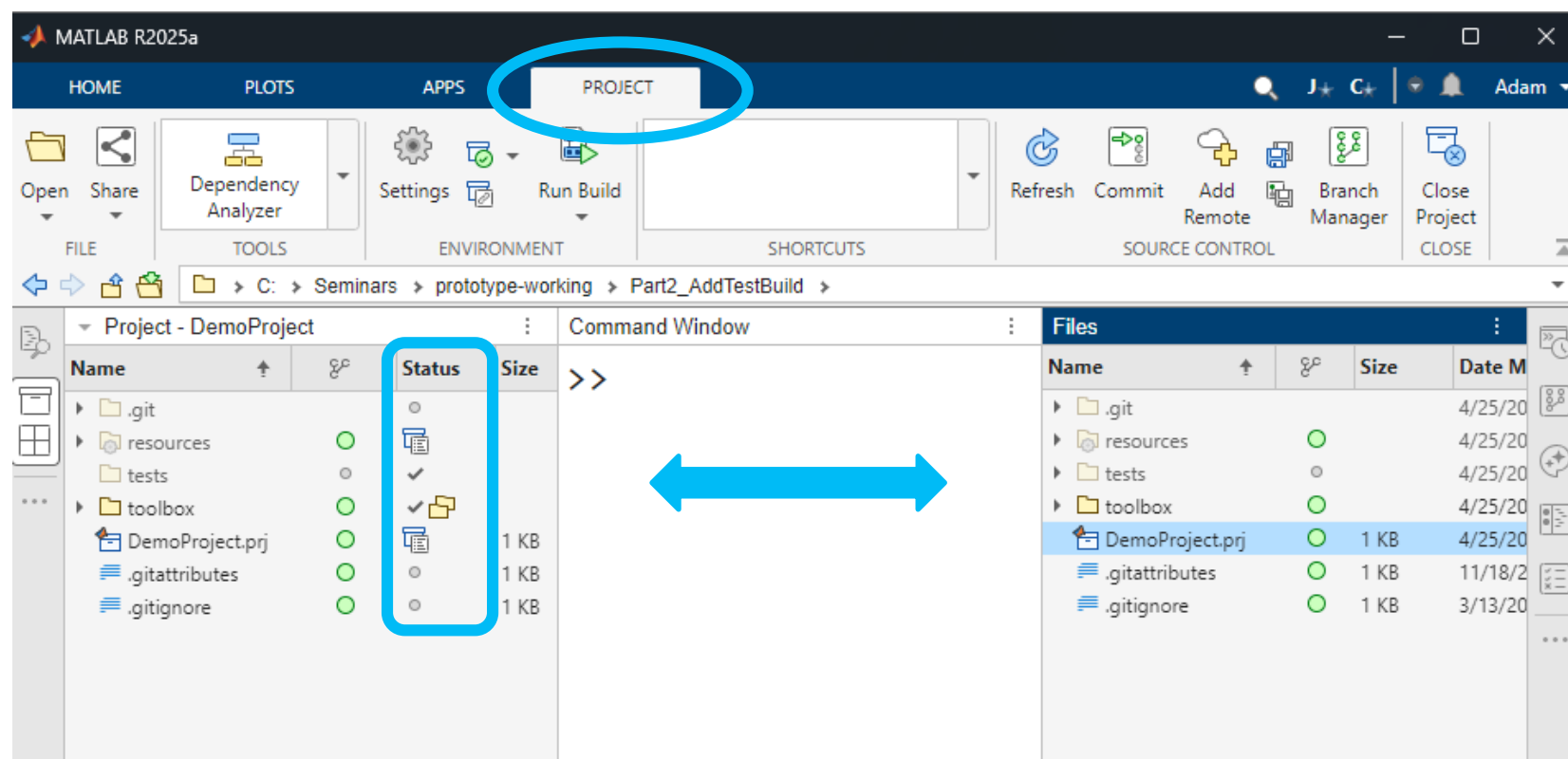
Nové témy prostredia umožňuje ľahký prístup k funkciám

- Nové panely pre rýchle programovanie a úlohy vývoja softvéru
 - Debugger
 - Správa zdrojov



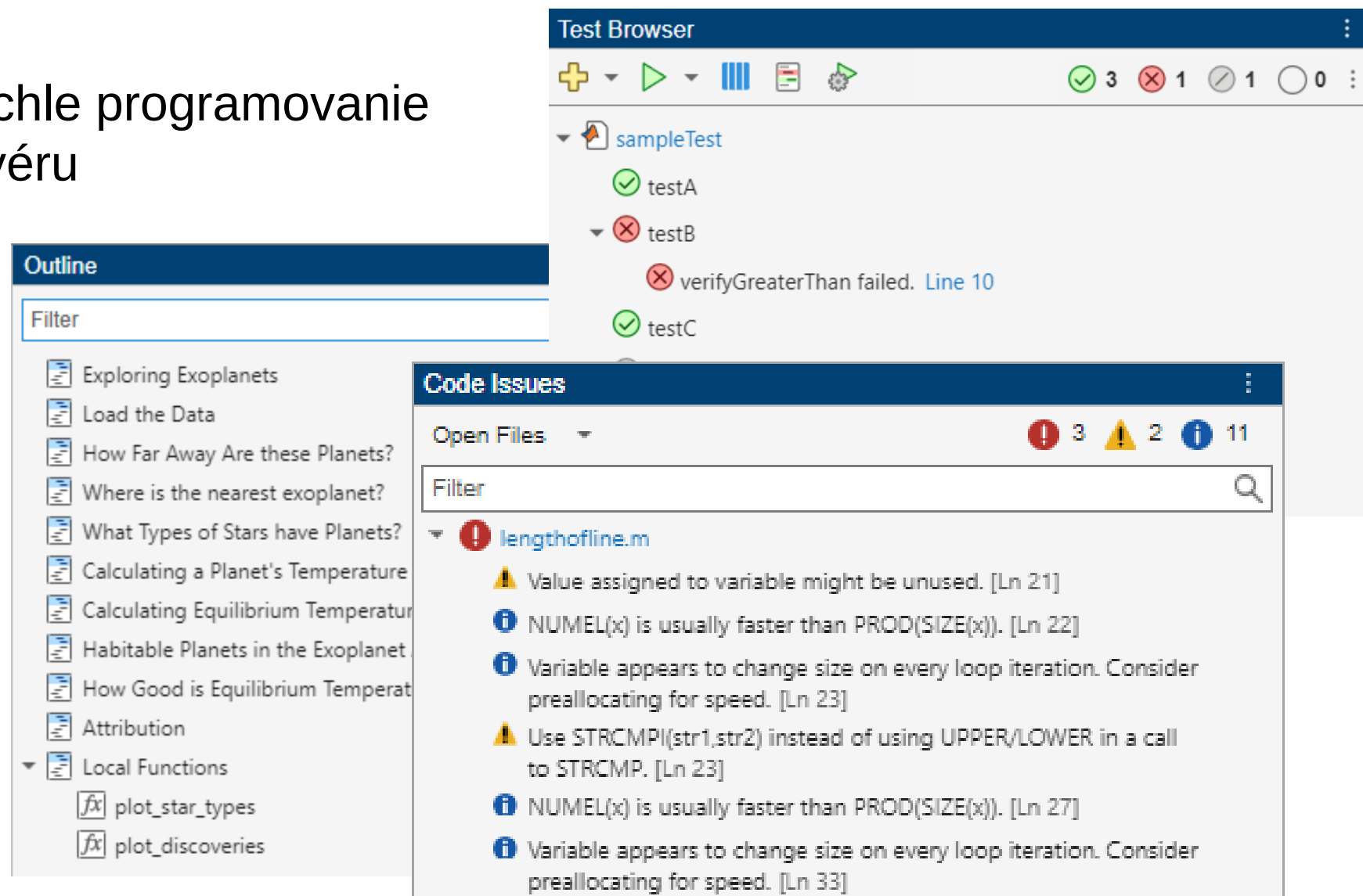
Nové témy prostredia umožňuje ľahký prístup k funkciám

- Nové panely pre rýchle programovanie a úlohy vývoja softvéru
 - Debugger
 - Správa zdrojov
 - Projekty



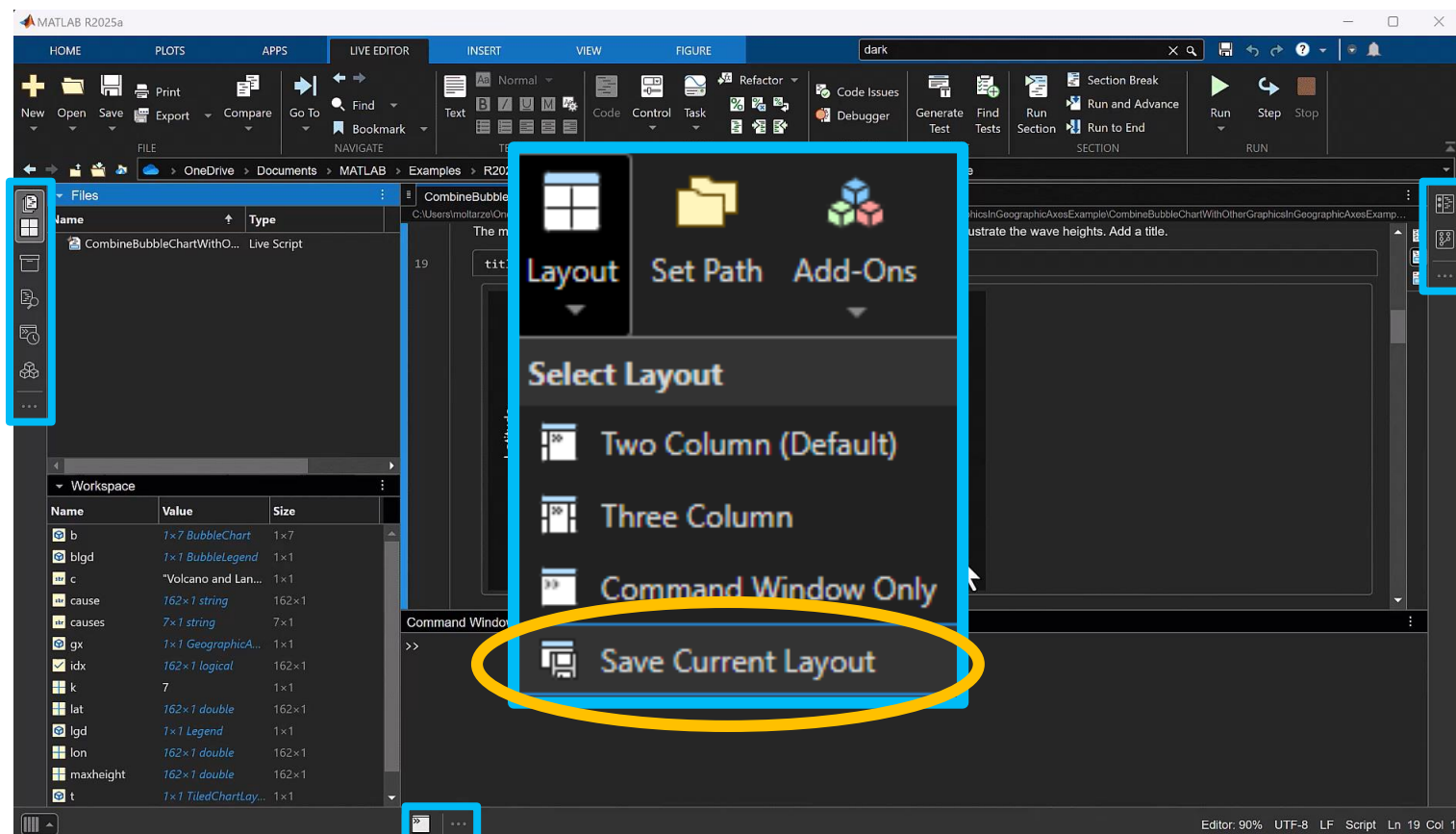
Nové témy prostredia umožňuje ľahký prístup k funkciám

- Nové panely pre rýchle programovanie a úlohy vývoja softvéru
 - Debugger
 - Správa zdrojov
 - Projekty
 - Test Browser
 - Outline
 - Code Issues

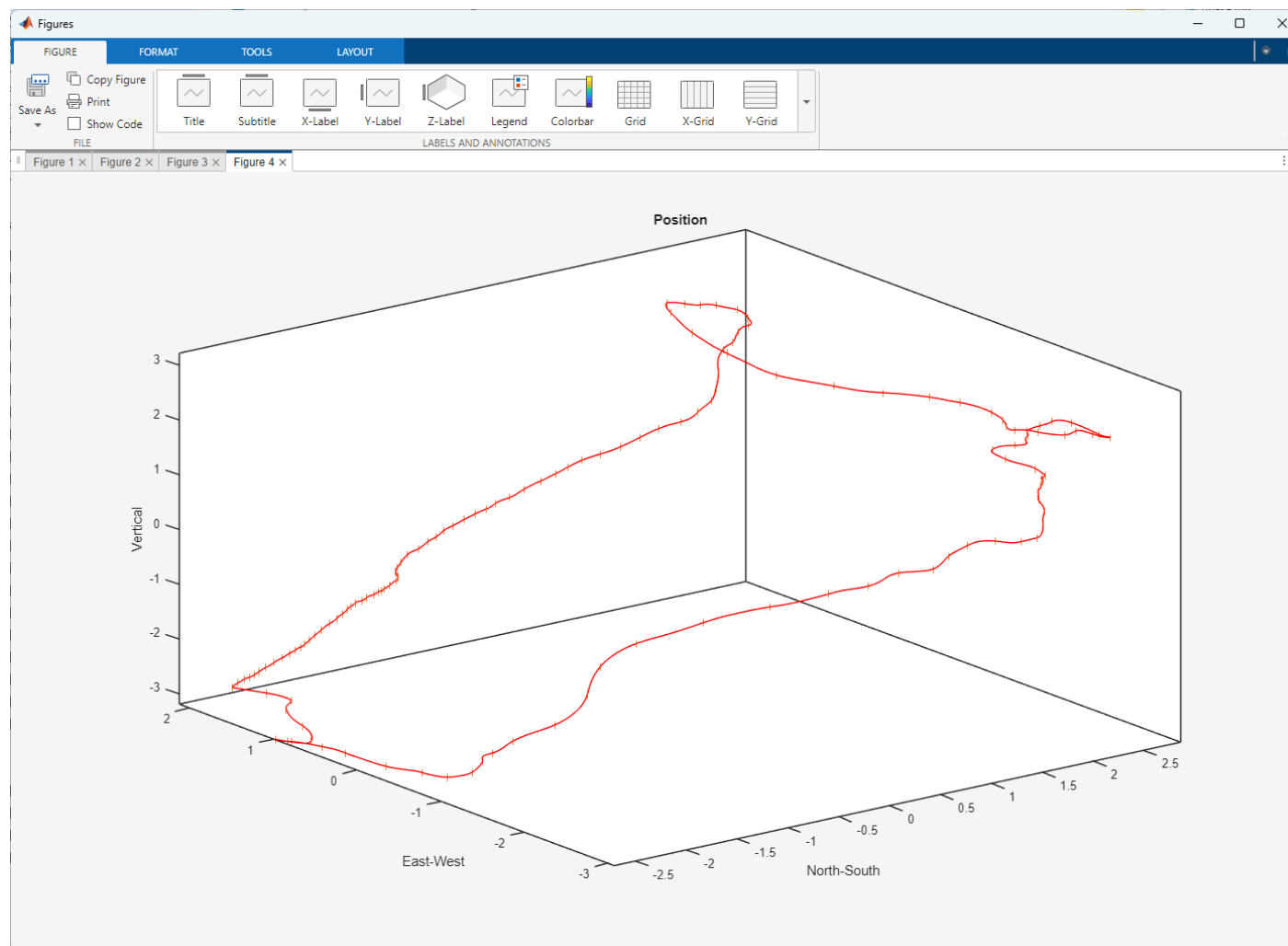


Nové témy prostredia umožňujú ľahký prístup k funkciám

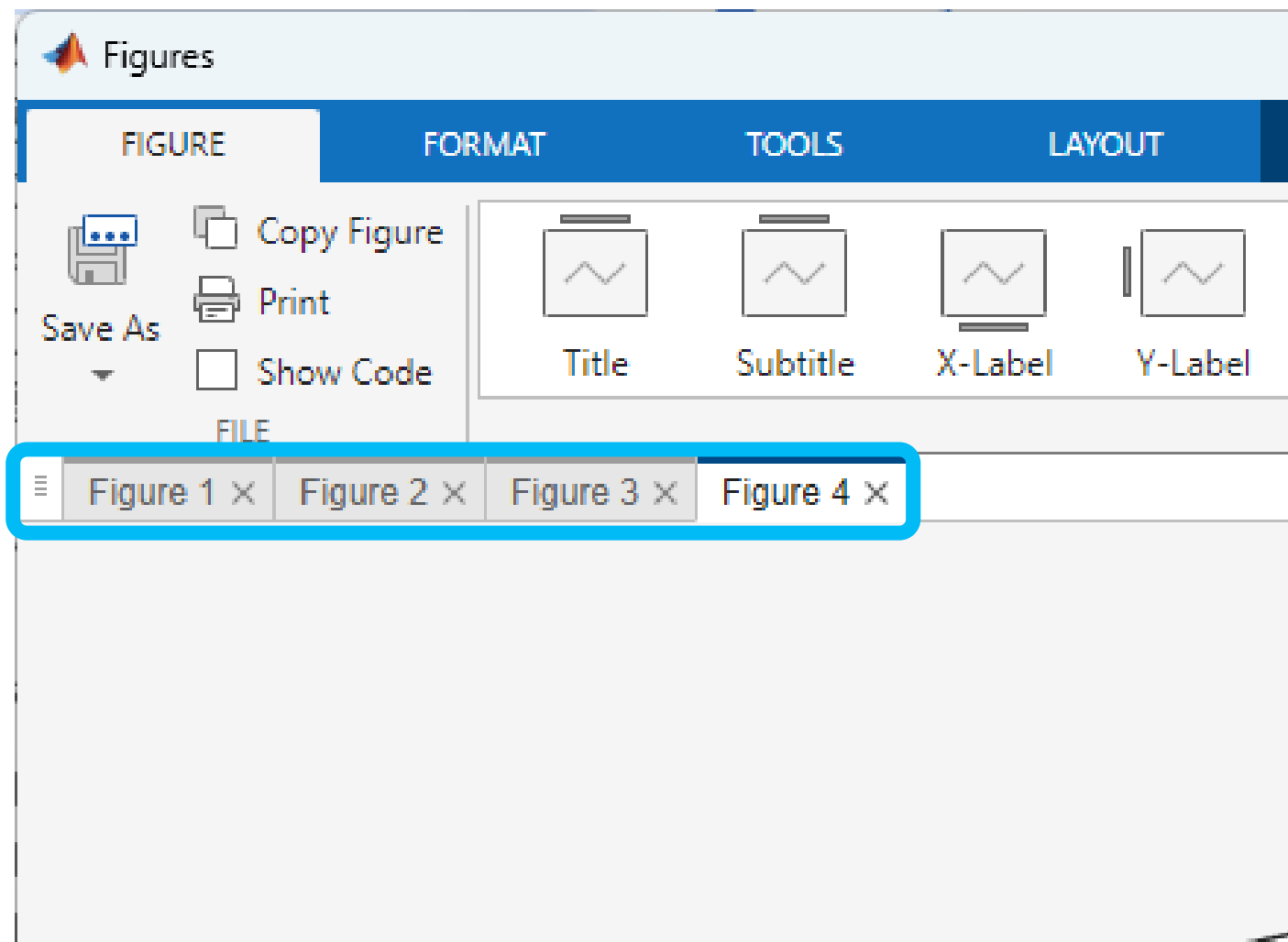
- Bočné panely poskytujú rýchly prístup k nástrojom a panelom na pracovnej ploche
- Vytvorte si prispôsobené rozloženia, ktoré obsahujú iba panely pre konkrétne úlohy



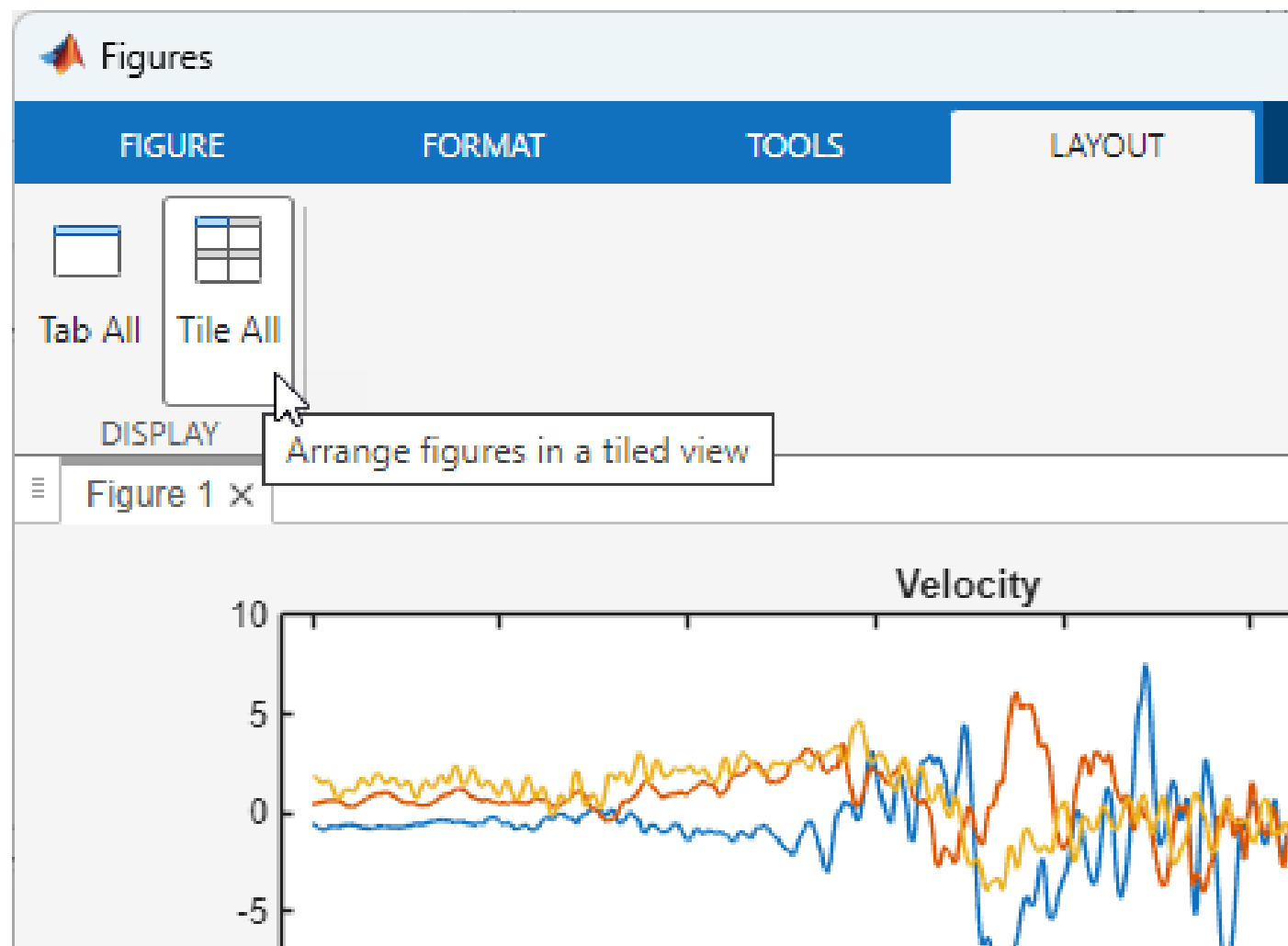
Grafy sa otvárajú v záložkách v jednom kontainery



Grafy sa otvárajú v záložkách v jednom kontainery



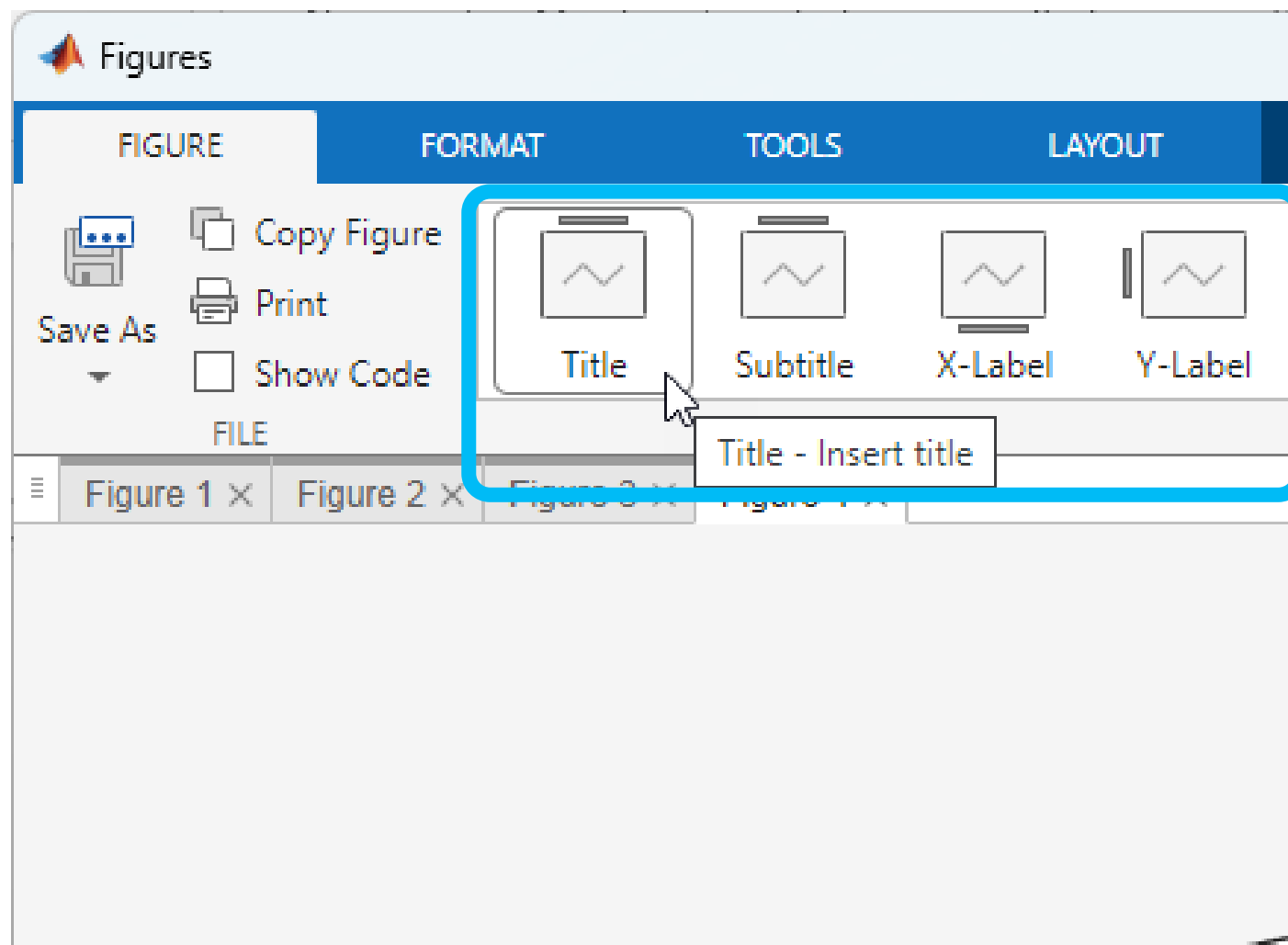
Prepínanie medzi kartami a dlaždicami



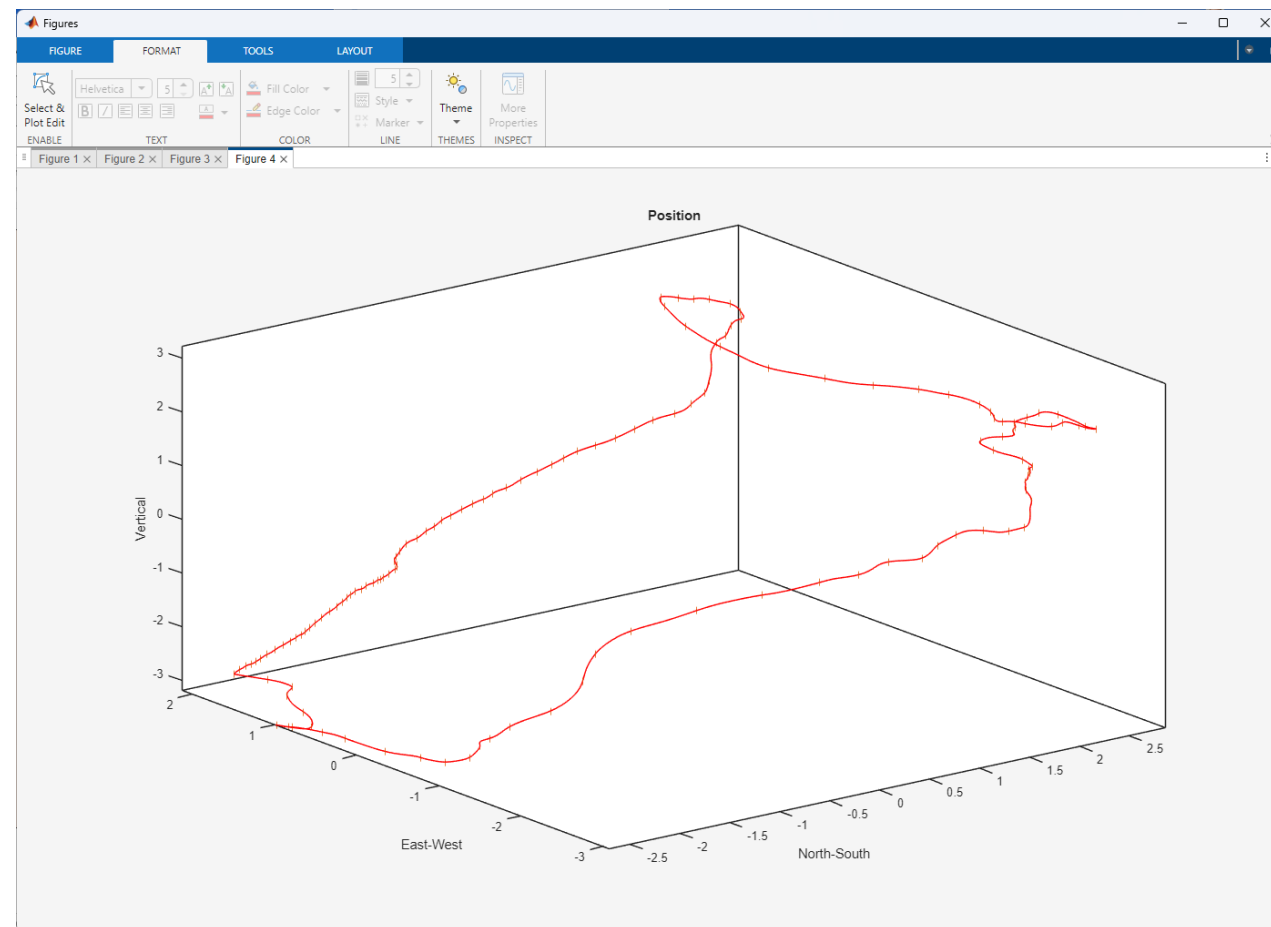
Prepínanie medzi kartami a dlaždicami



Interaktívna úprava grafov s novým toolstripom

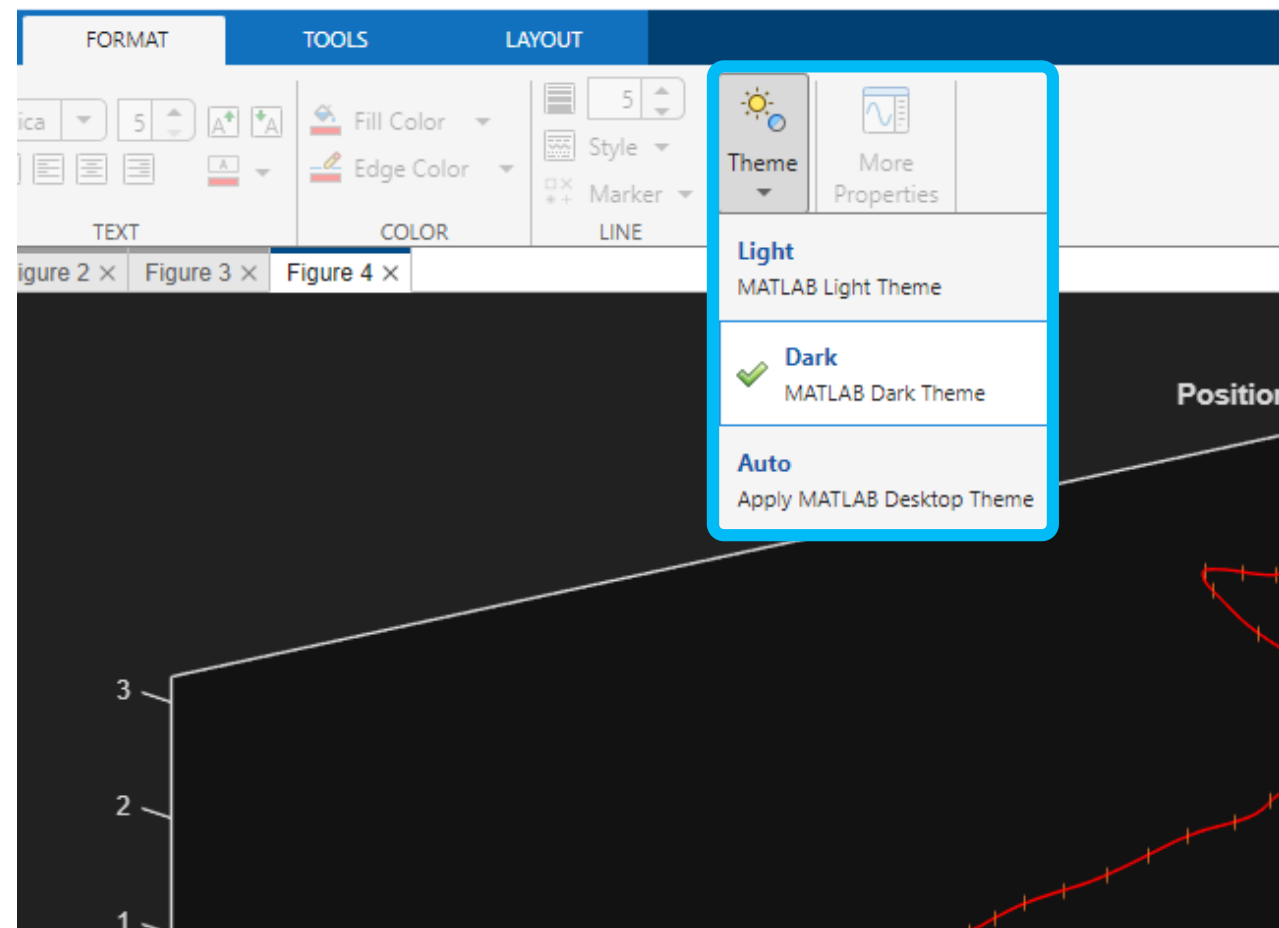


Ovládanie tém grafov oddelené od plochy



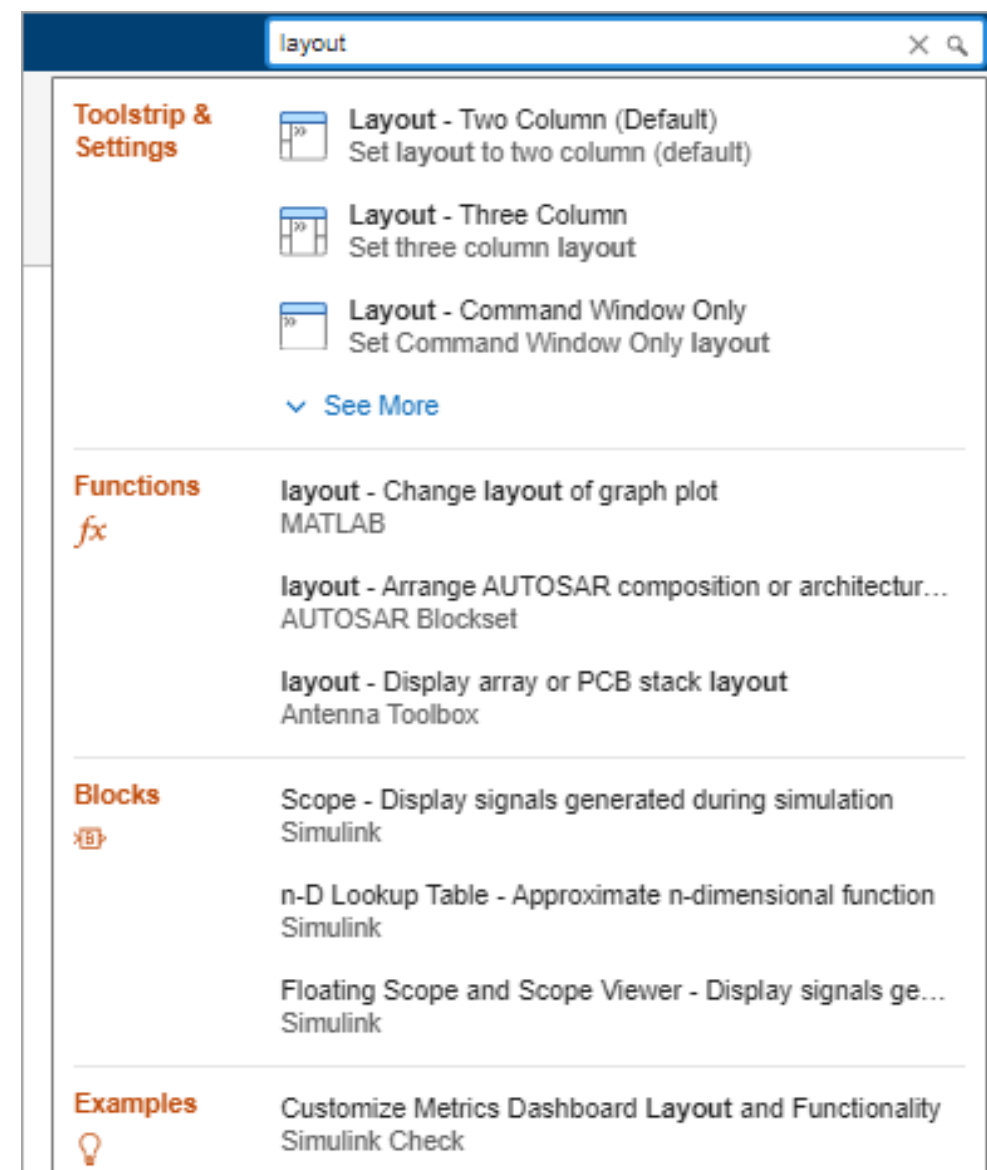
Ovládanie tém grafov oddelené od plochy

- Štandardne téma okien zodpovedá téme pracovnej plochy
- Môžete
 - Interaktívne vyberať inú tému na záložke Format
 - Explicitne nastaviť tému
 - funkcia theme
 - vlastnosť Theme objektu Figure



Vykonávajújte akcie priamo z vyhľadávania na pracovnej ploche

- Akcie Toolstripu
- Nastavenia
- Aplikácie
- Zdroje Help Centera , vrátane:
 - Dokumentácie
 - Príkladov
 - MATLAB Answers



A oveľa viac...

- Ukladanie live skriptov a funkcií pomocou textového formátu (.m)

```
fahrenheitconverter.m x +
C:\Work\ahrenheitconverter.m

Convert Fahrenheit to Celsius and Kelvin
This live script converts -40 degrees Fahrenheit to Celsius and Kelvin.

3 F = -40;
4 C = (F - 32) * 5/9;
5 fprintf("%.2f degrees Fahrenheit is %.2f degrees Celsius.",F,C);

-40.00 degrees Fahrenheit is -40.00 degrees Celsius.

6 K = C + 273.15;
7 fprintf("%.2f degrees Fahrenheit is %.2f Kelvin.",F,K);

-40.00 degrees Fahrenheit is 233.15 Kelvin.
```

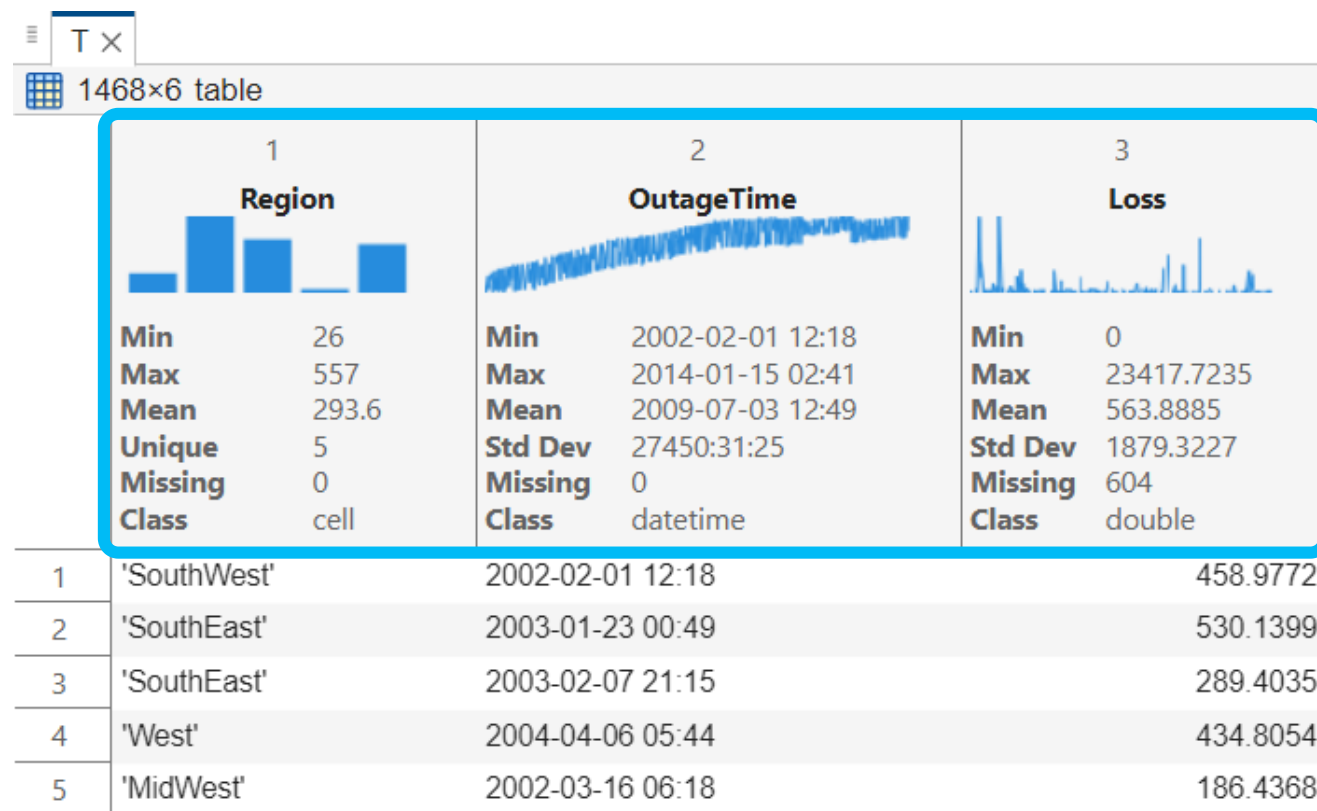


```
fahrenheitconverter.m x +
C:\Work\ahrenheitconverter.m

1 |[text] # Convert Fahrenheit to Celsius and Kelvin
2 |[text] This live script converts -40 degrees Fahrenheit to Celsius and Kelvin.
3 F = -40;
4 C = (F - 32) * 5/9;
5 fprintf("%.2f degrees Fahrenheit is %.2f degrees Celsius.",F,C); %[output:82b3b5a1]
6 K = C + 273.15;
7 fprintf("%.2f degrees Fahrenheit is %.2f Kelvin.",F,K); %[output:1d79611f]
8
9 |[appendix>{"version":"1.0"}
10 %---
11 |[metadata:view]
12 % data: {"layout":"inline"}
13 %---
14 %[output:82b3b5a1]
15 % data: {"dataType":"text","outputData":{"text":"-40.00 degrees Fahrenheit is -40.00
16 %---
17 %[output:1d79611f]
18 % data: {"dataType":"text","outputData":{"text":"-40.00 degrees Fahrenheit is 233.15
19 %---
```

A oveľa viac...

- Ukladanie live skriptov a funkcií pomocou textového formátu (.m)
- Zobrazenie grafov a štatistík pre dáta v tabuľkách



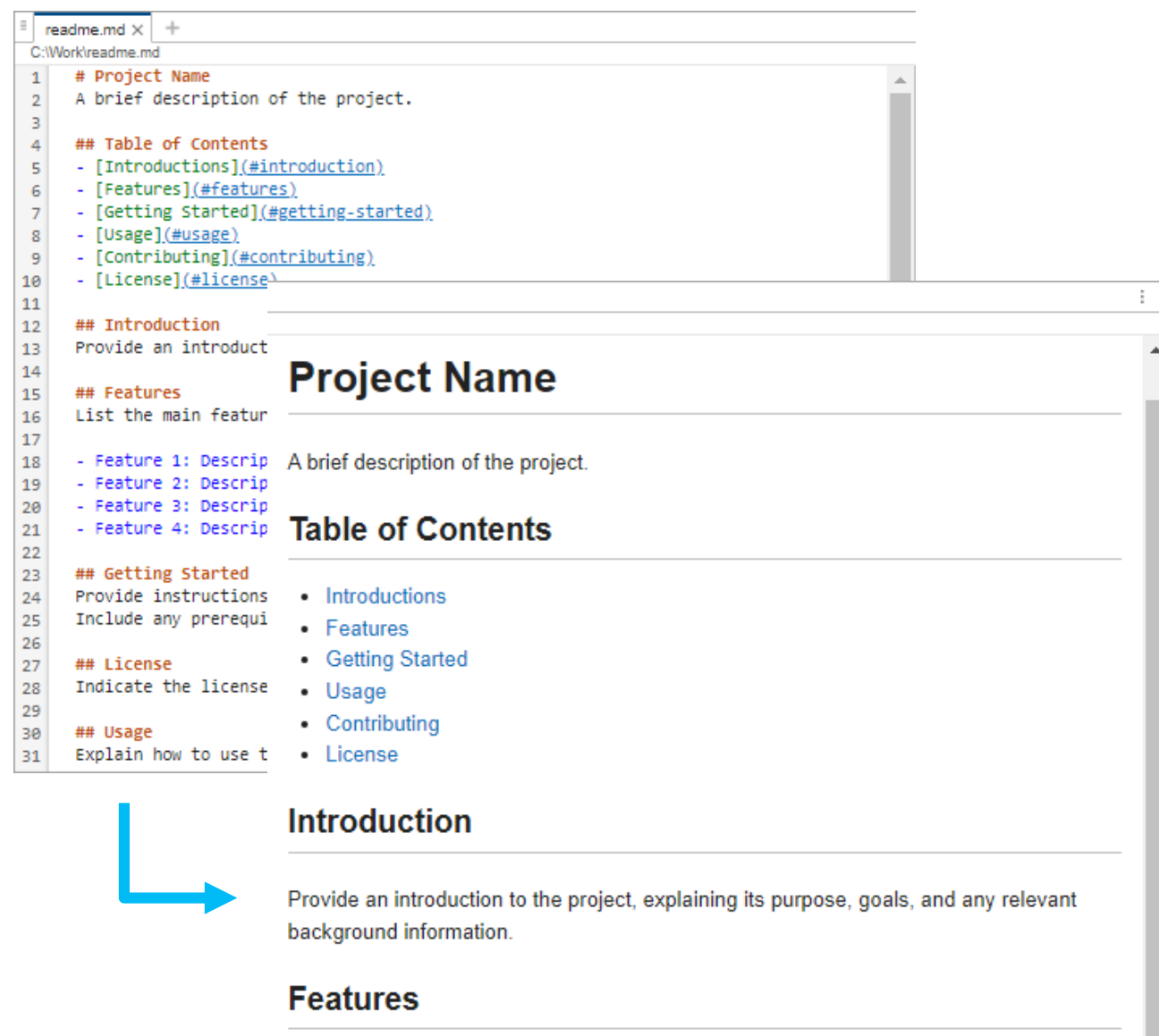
A oveľa viac...

- Ukladanie live skriptov a funkcií pomocou textového formátu (.m)
- Zobrazenie grafov a štatistík pre dáta v tabuľkách
- Videá v live skriptoch



A oveľa viac...

- Ukladanie live skriptov a funkcií pomocou textového formátu (.m)
- Zobrazenie grafov a štatistík pre dáta v tabuľkách
- Videá v live skriptoch
- Zobrazenie Markdown súborov v editore



```
1 # Project Name
2 A brief description of the project.
3
4 ## Table of Contents
5 - [Introductions](#introduction)
6 - [Features](#features)
7 - [Getting Started](#getting-started)
8 - [Usage](#usage)
9 - [Contributing](#contributing)
10 - [License](#license)
11
12 ## Introduction
13 Provide an introduct
14
15 ## Features
16 List the main featur
17
18 - Feature 1: Descrip
19 - Feature 2: Descrip
20 - Feature 3: Descrip
21 - Feature 4: Descrip
22
23 ## Getting Started
24 Provide instructions
25 Include any prerequi
26
27 ## License
28 Indicate the license
29
30 ## Usage
31 Explain how to use t
```

Project Name

A brief description of the project.

Table of Contents

- [Introductions](#)
- [Features](#)
- [Getting Started](#)
- [Usage](#)
- [Contributing](#)
- [License](#)

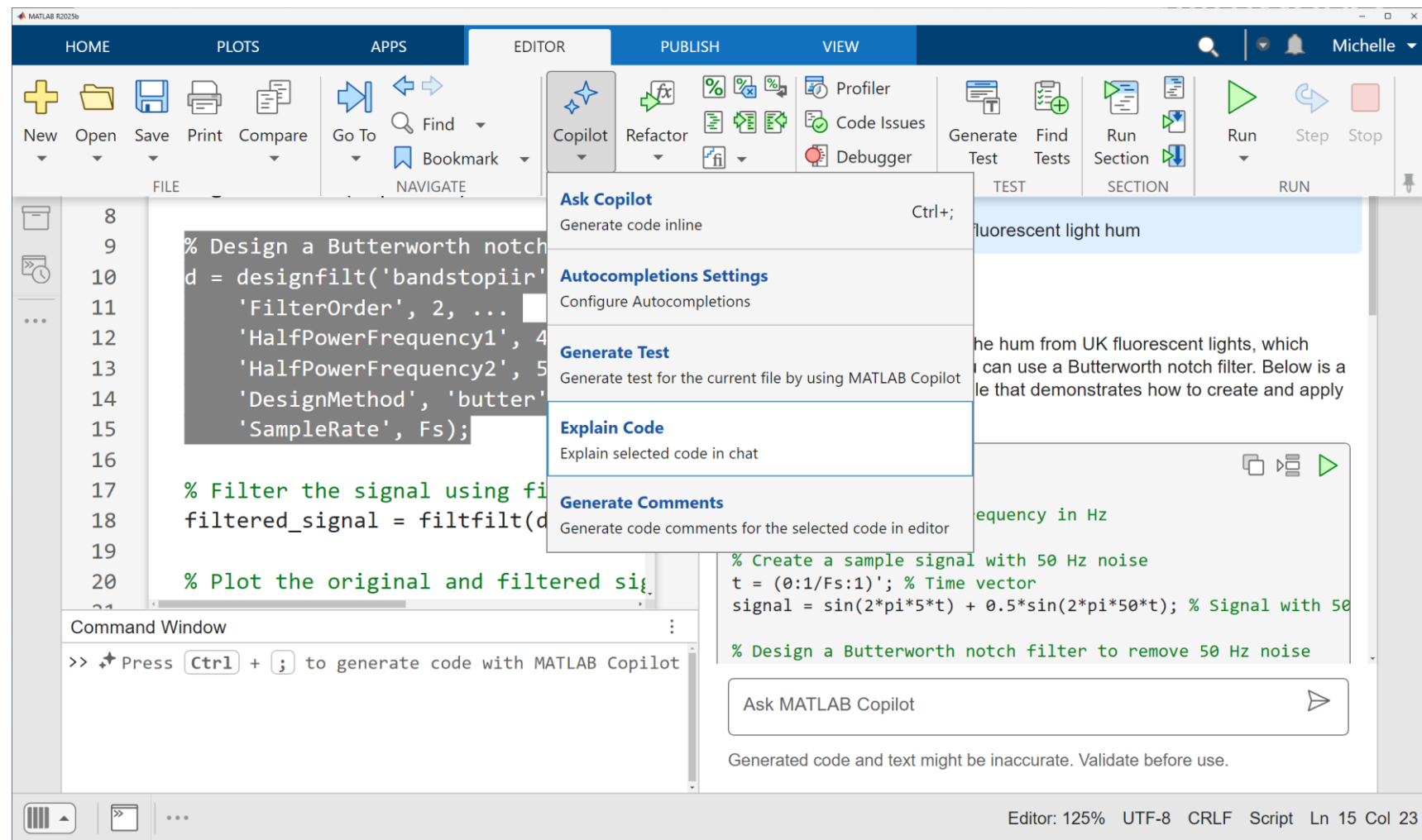
Introduction

Provide an introduction to the project, explaining its purpose, goals, and any relevant background information.

Features

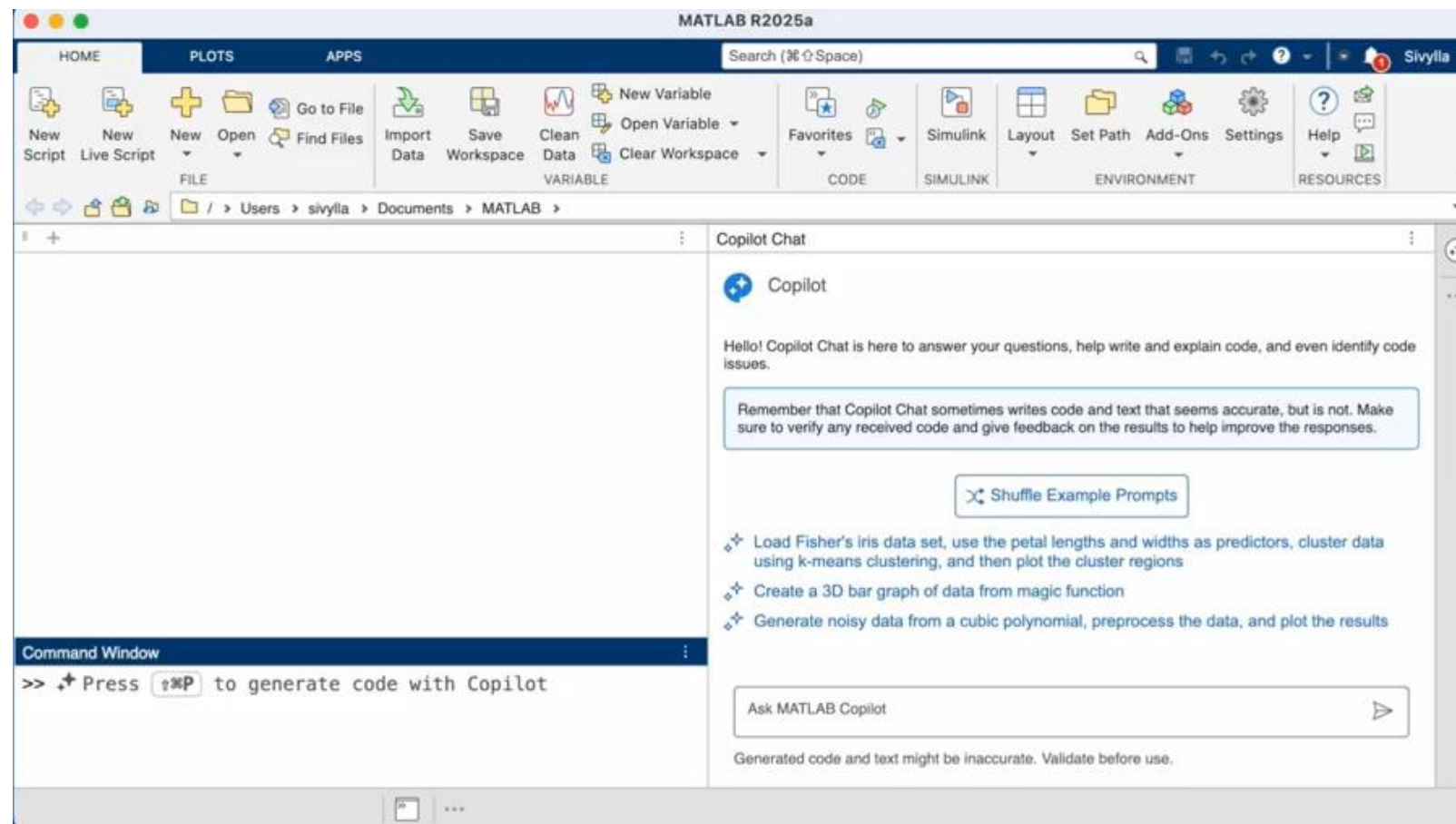
MATLAB Copilot – AI asistent optimalizovaný pre MATLAB

- Nový produkt



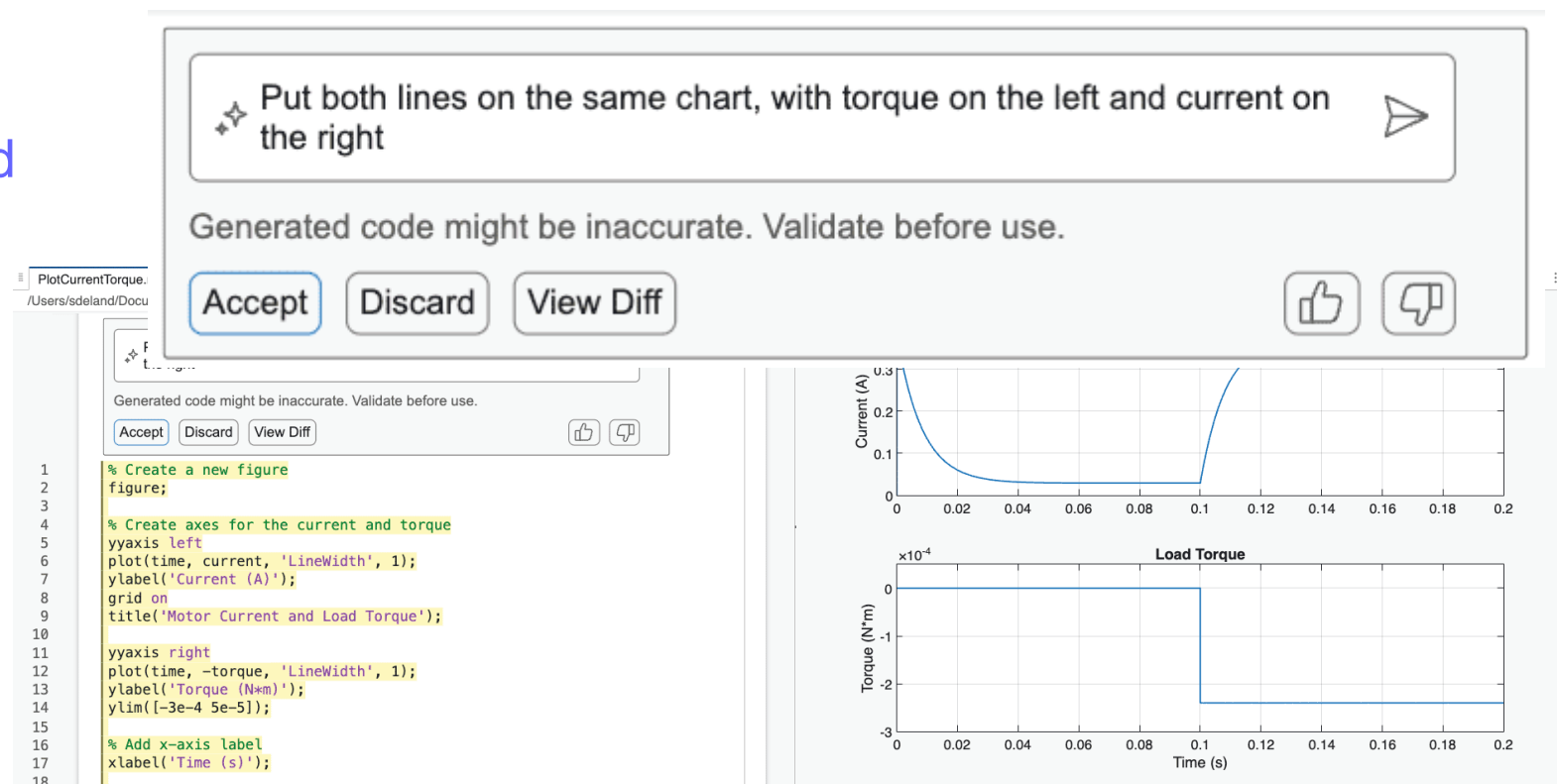
MATLAB Copilot – AI asistent optimalizovaný pre MATLAB

- Nový produkt
- Umožňuje
 - komunikovať s Copilotom



MATLAB Copilot – AI asistent optimalizovaný pre MATLAB

- Nový produkt
- Umožňuje
 - komunikovať s Copilotom
 - generovať a modifikovať kód



MATLAB Copilot – AI asistent optimalizovaný pre MATLAB

- Nový produkt
- Umožňuje
 - komunikovať s Copilotom
 - generovať a modifikovať kód
 - vysvetlenie kódu a chýb

 Copilot

Brief Overview: This code sets up and solves a differential-algebraic equation (DAE) model for a transistor amplifier circuit, simulating the relationship between input and output voltages over time.

Summary: The code begins by defining a small constant and initializing a mass matrix for the system. It then defines a function representing the system of equations governing the amplifier's behavior. Initial conditions for the voltages are set, and the ode23t solver is used to solve the DAE. Finally, it plots the input and output voltages over time, providing a visual representation of the amplifier's response.

Walkthrough:

Copilot Chat

SP Sivylla

Explain Error

Unrecognized function or variable 'sine'.

Error in mysine (line 3)
y = sine(x);
^^^^^^^^^^^^

 Copilot

Probable Explanation: This error indicates that MATLAB cannot find a function or variable named 'sine'. It usually occurs when the function is not defined or misspelled.

Troubleshooting Recommendation: Check if you have defined the function 'sine' in your code or if it is a built-in function. If it's a custom function, ensure it's in the current folder or on the MATLAB path. If you meant to use the sine function, use `sin(x)` instead.

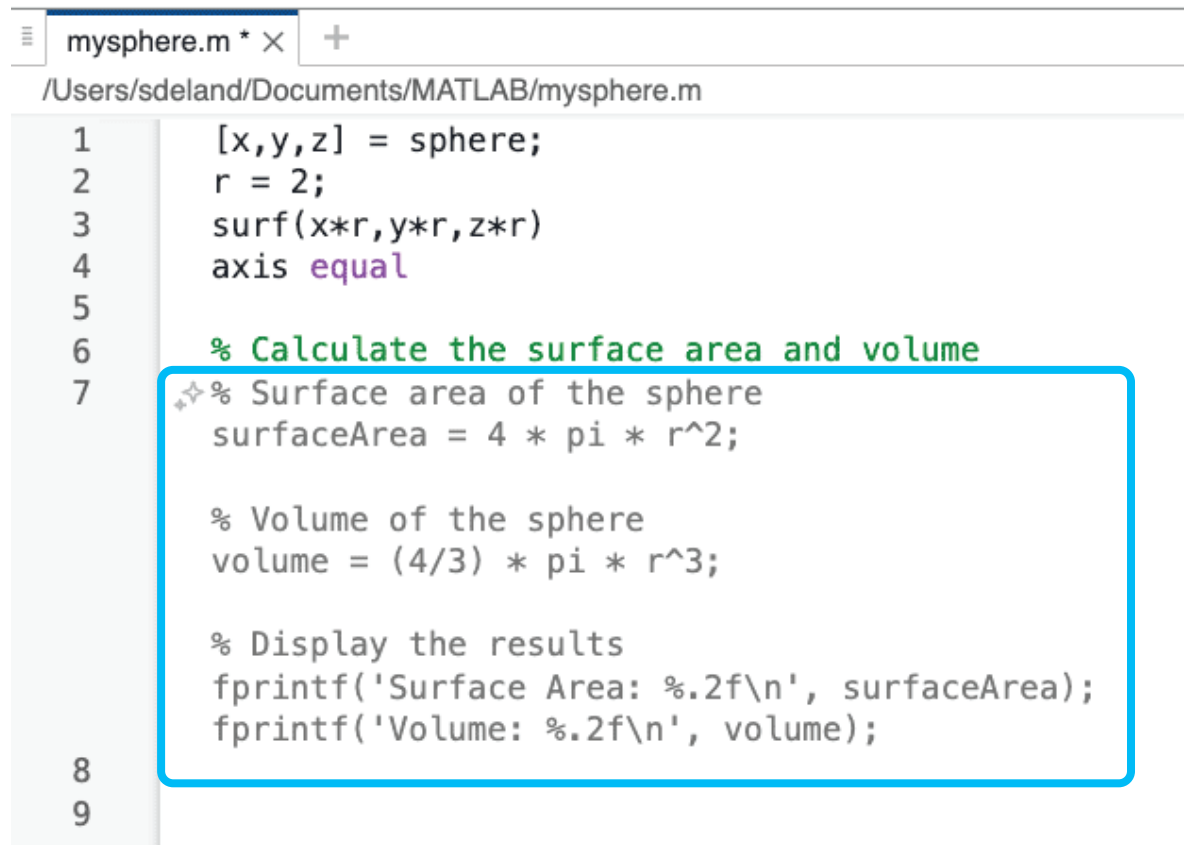




small constant c
AE solver.

MATLAB Copilot – AI asistent optimalizovaný pre MATLAB

- Nový produkt
- Umožňuje
 - komunikovať s Copilotom
 - generovať a modifikovať kód
 - vysvetlenie kódu a chýb
 - doplnenie kódu

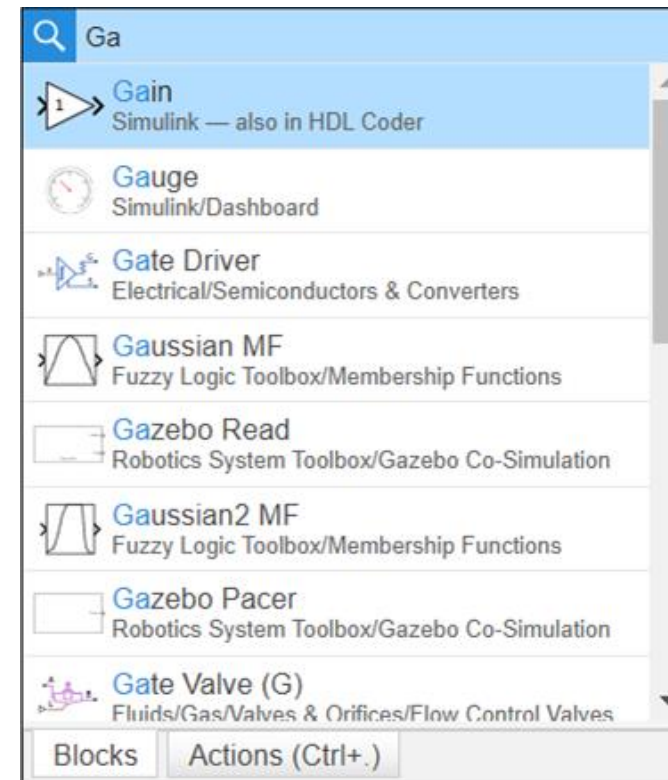


```
mysphere.m * × +
/Users/sdeland/Documents/MATLAB/mysphere.m

1 [x,y,z] = sphere;
2 r = 2;
3 surf(x*r,y*r,z*r)
4 axis equal
5
6 % Calculate the surface area and volume
7 ✦% Surface area of the sphere
  surfaceArea = 4 * pi * r^2;
8
  % Volume of the sphere
  volume = (4/3) * pi * r^3;
9
  % Display the results
  fprintf('Surface Area: %.2f\n', surfaceArea);
  fprintf('Volume: %.2f\n', volume);
```

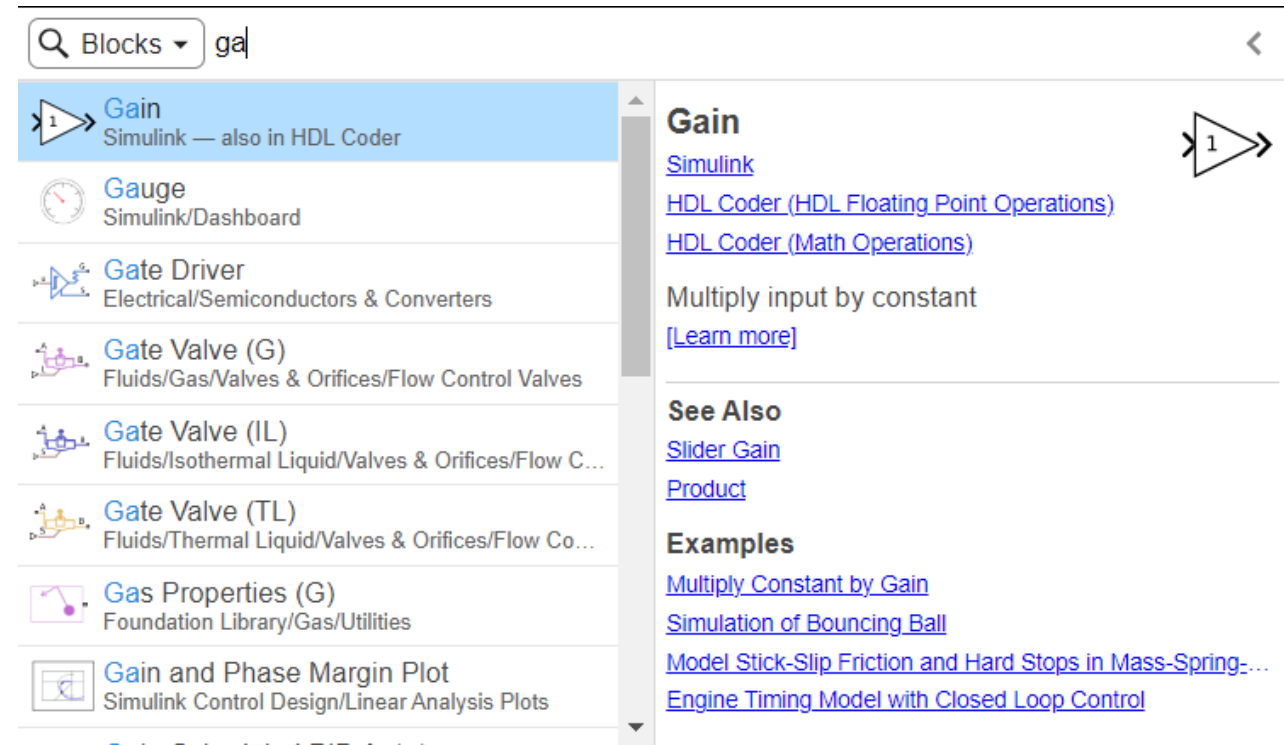
Zobrazenie podrobnosti blokov a zdrojov

- Vložte bloky priamo do Simulinku pomocou rýchleho vkladania



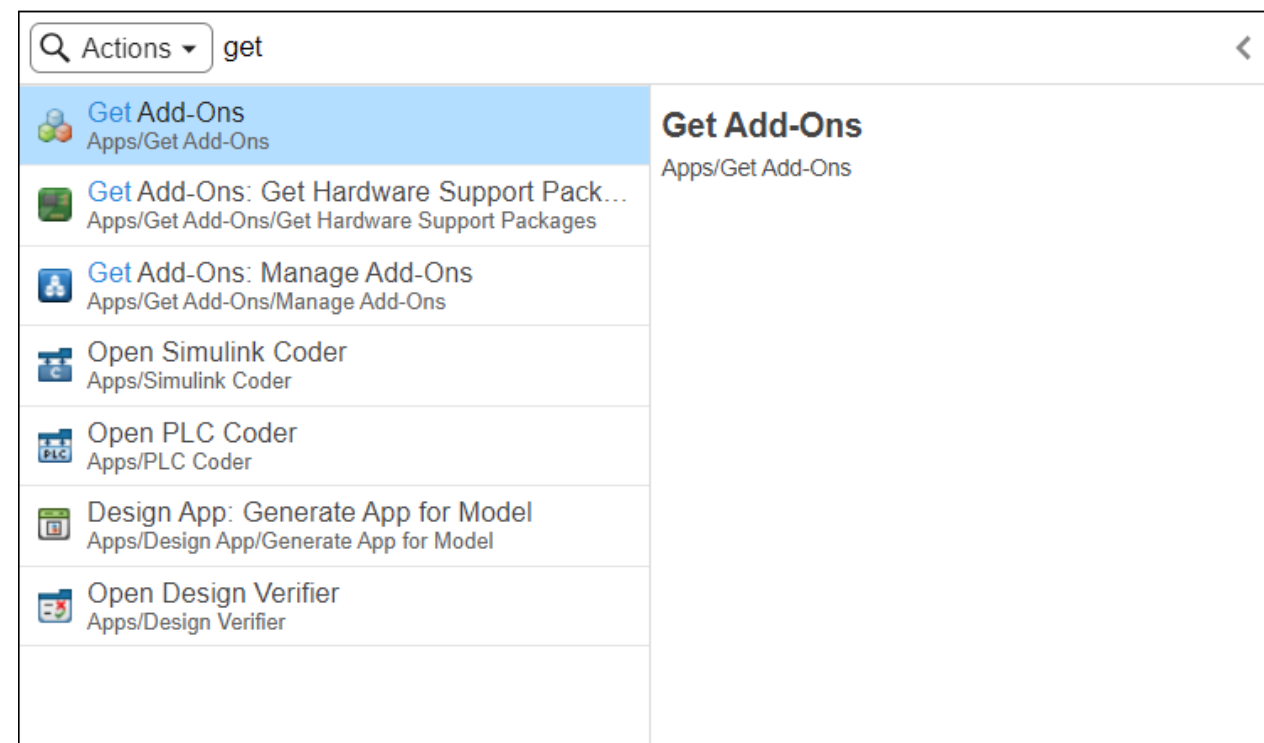
Zobrazenie podrobnosti blokov a zdrojov

- Vložte bloky priamo do Simulinku pomocou rýchleho vkladania
- Zobrazenie podrobnejších informácií o bloku v ponuke

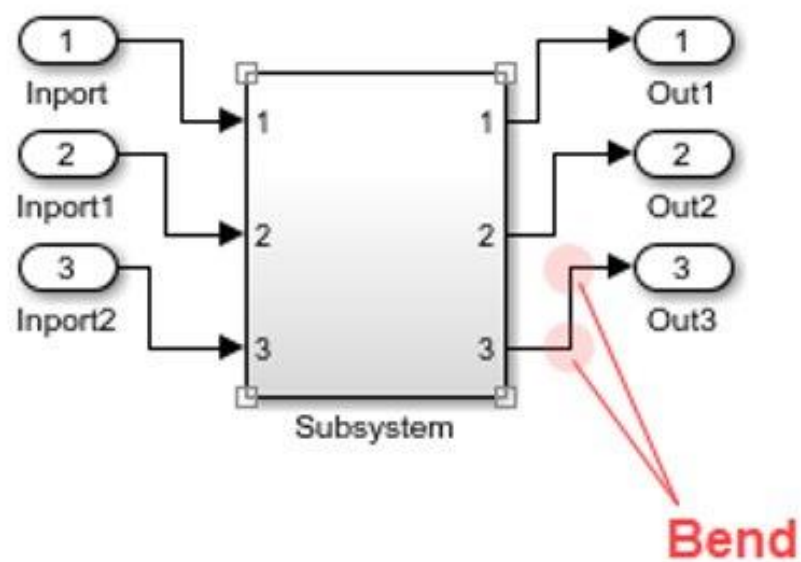


Zobrazenie podrobnosti blokov a zdrojov

- Vložte bloky priamo do Simulinku pomocou rýchleho vkladania
- Zobrazenie podrobnejších informácií o bloku v ponuke
- Vyhľadanie akcií na získanie doplnkov alebo otvorenie aplikácií

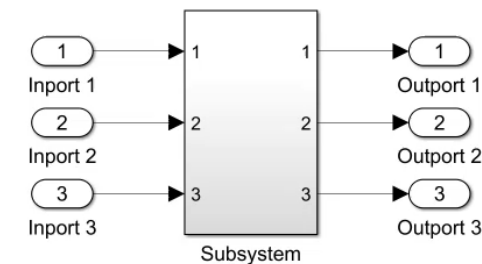
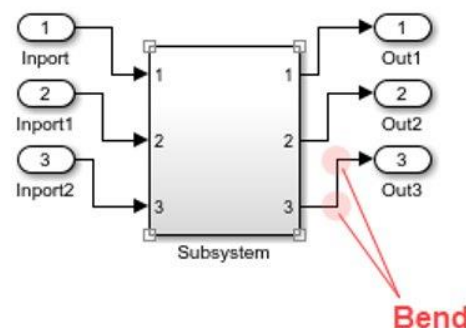


Zjednodušenie čiar signálov



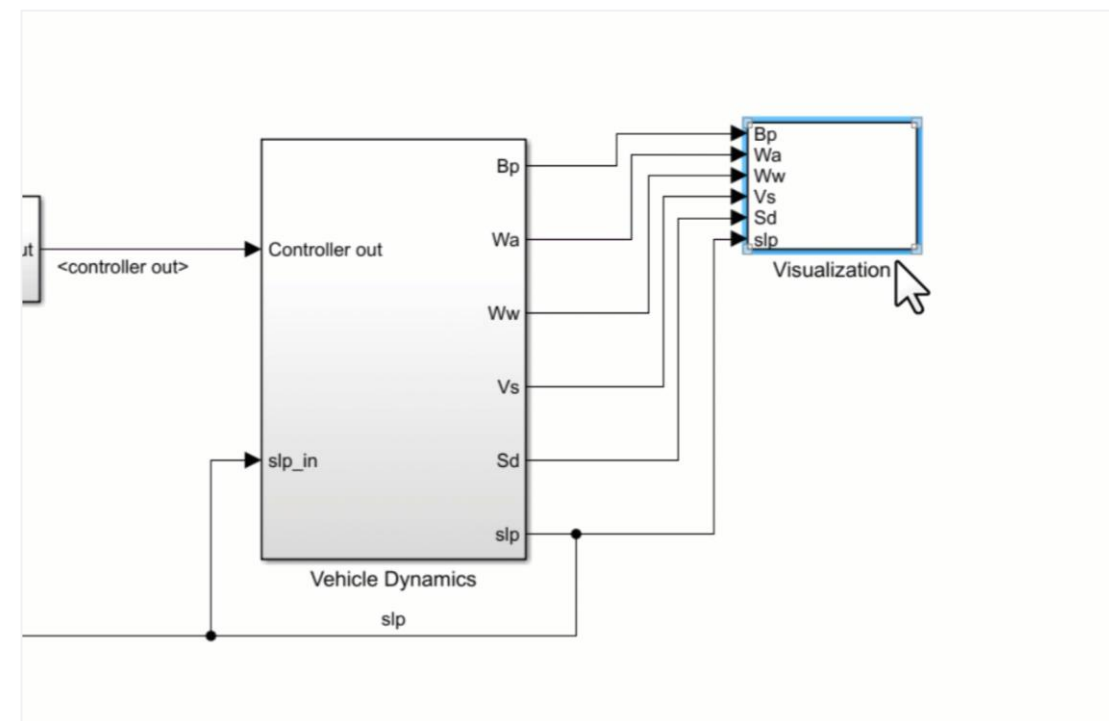
Zjednodušenie čiar signálov

- Zachovanie tvaru signálov pri presúvaní a zmene veľkosti blokov



Zjednodušenie čiar signálov

- Zachovanie tvaru signálov pri presúvaní a zmene veľkosti blokov
- Vyrovnanie čiar signálov automaticky pokiaľ to je možné



Vizualizácia dát a simulačných výstupov

- Zobrazenie výstupov simulácií vo vylepšenom bloku Scope

- nový toolstrip
- zoskupovanie viacerých Scope
- zlepšený výkon a odozva



Návrh systémov s ohľadom na komplexnosť

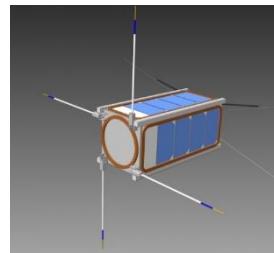
Controls



Signal
Processing



Wireless



Vision

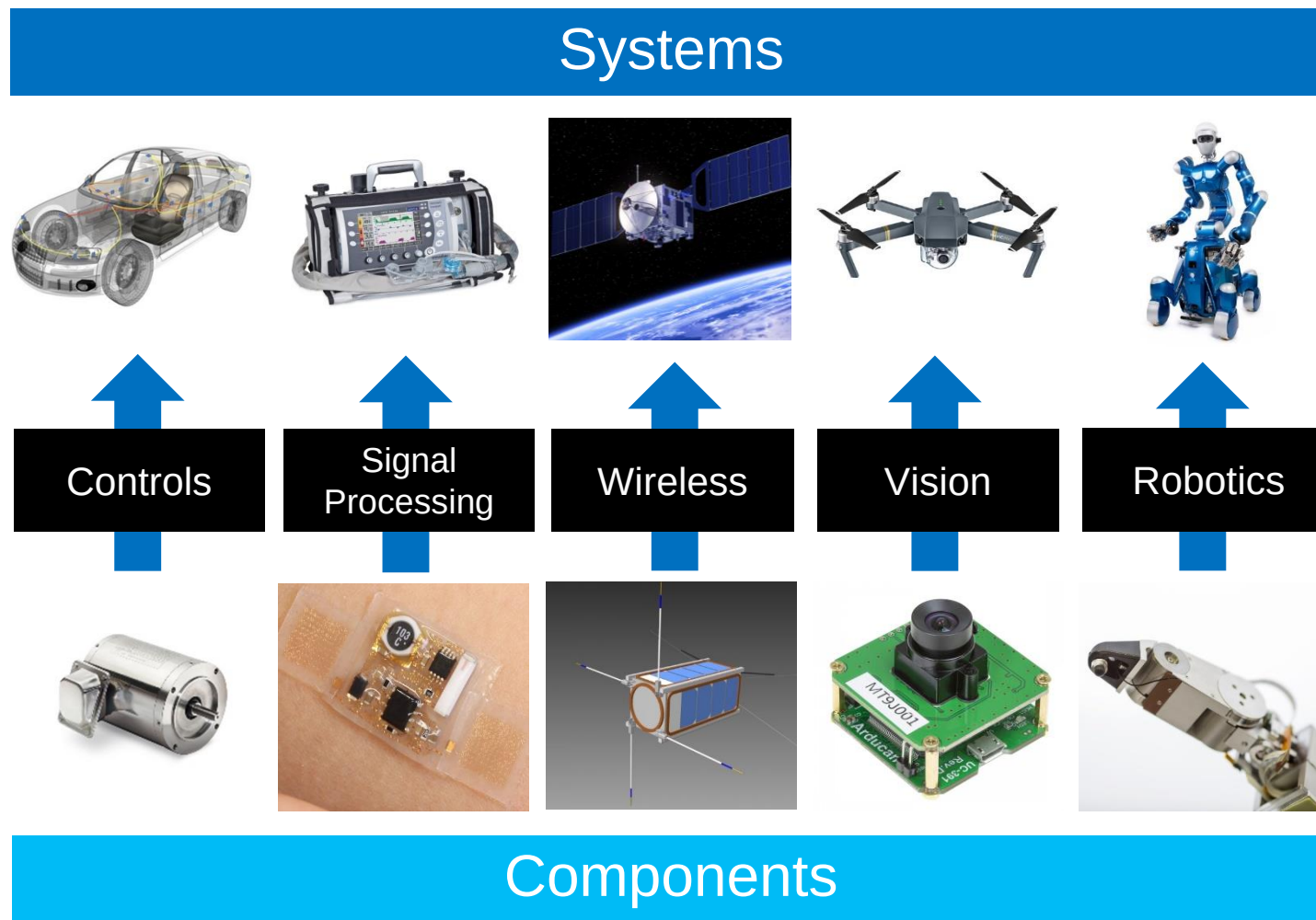


Robotics



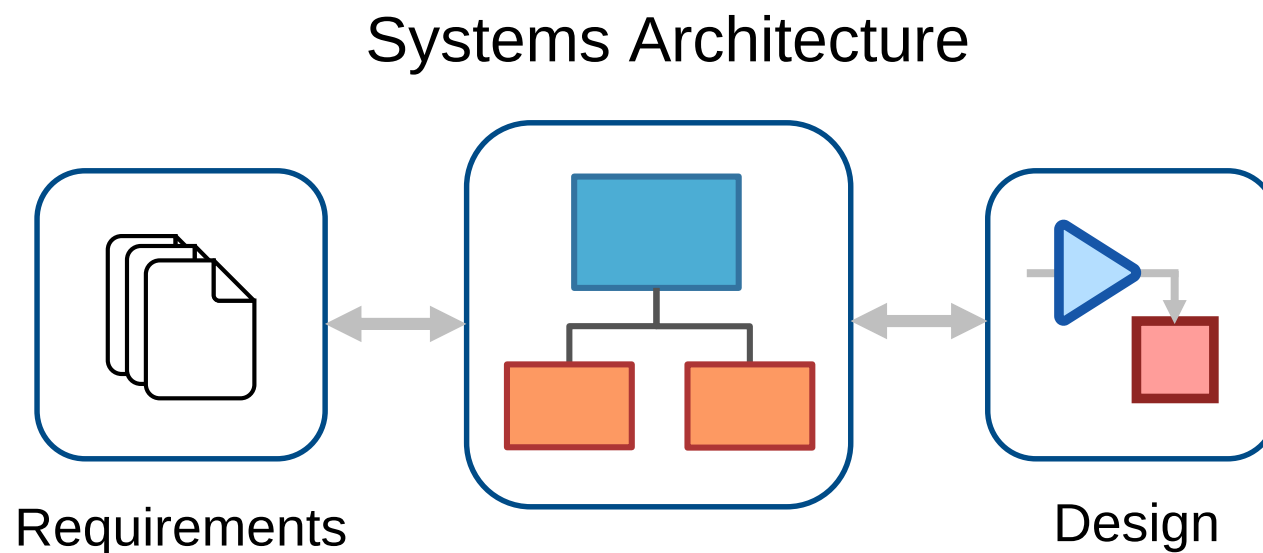
Components

Návrh systémov s ohľadom na komplexnosť



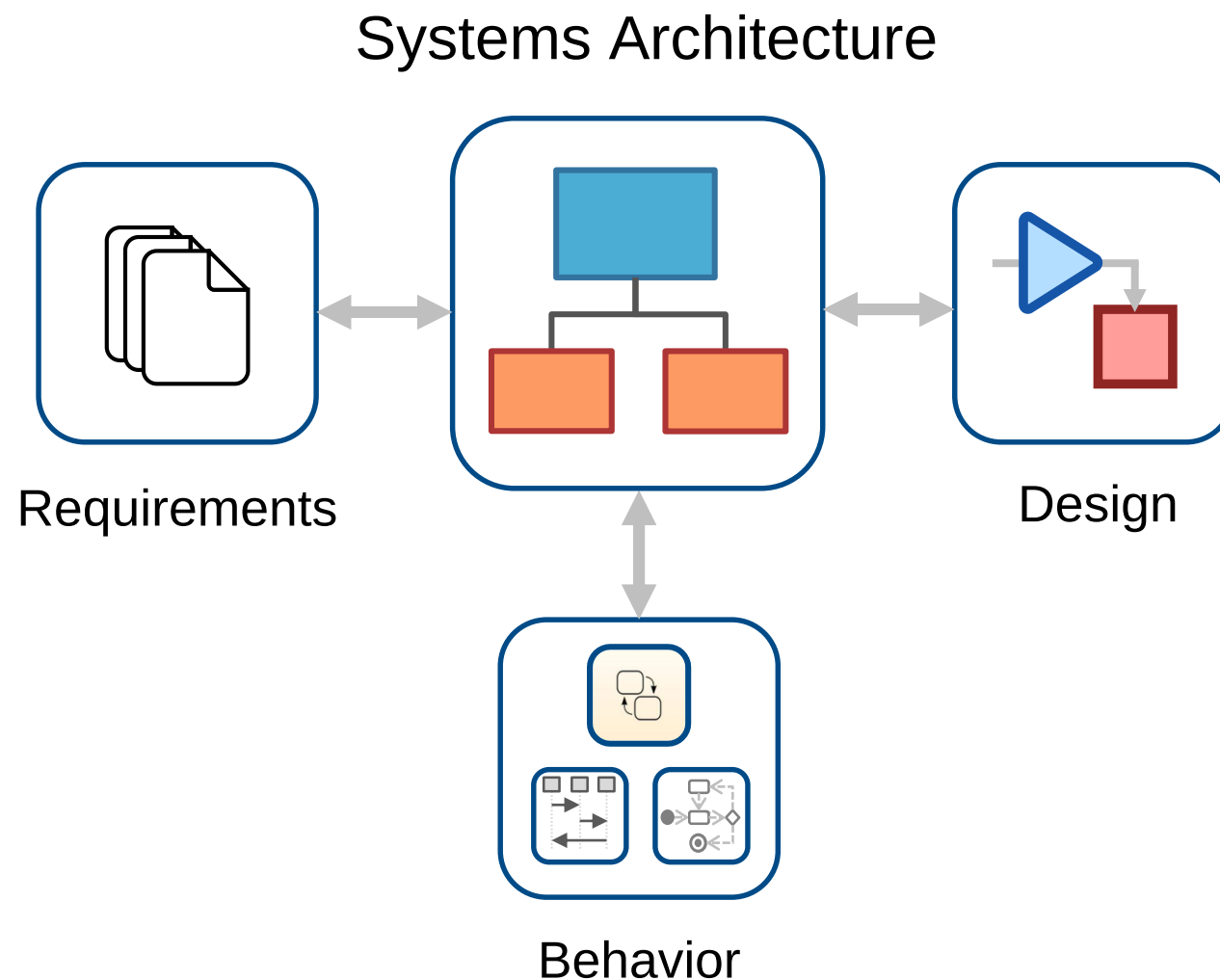
Návrh systémov s ohľadom na komplexnosť

- Tvorba architektúry systému v System Composer, ktorá prepája požiadavky a návrh



Návrh systémov s ohľadom na komplexnosť

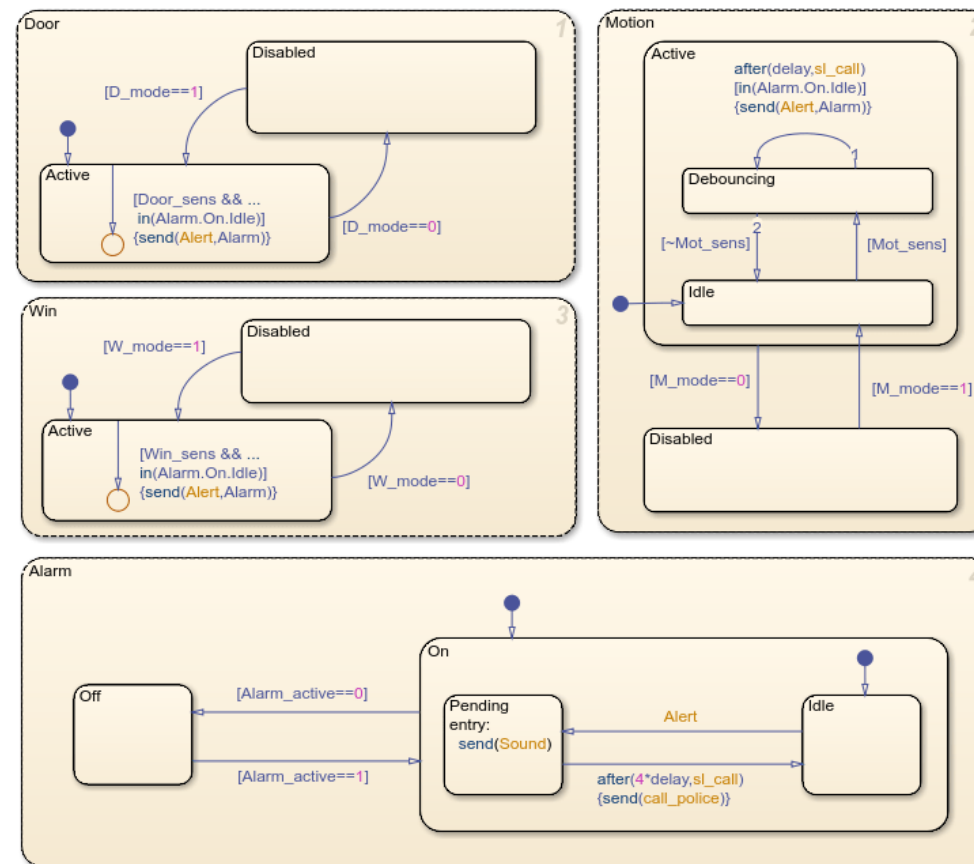
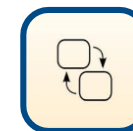
- Tvorba architektúry systému v System Composer, ktorá prepája požiadavky a návrh
- Popis správania medzi systémovými komponentami



Návrh systémov s ohľadom na komplexnosť

- Tvorba architektúry systému v System Composer, ktorá prepája požiadavky a návrh
- Popis správania medzi systémovými komponentami

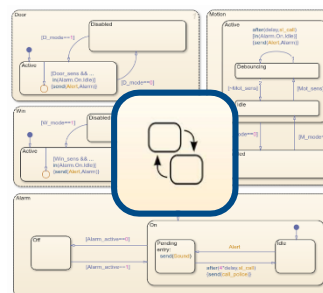
State Charts



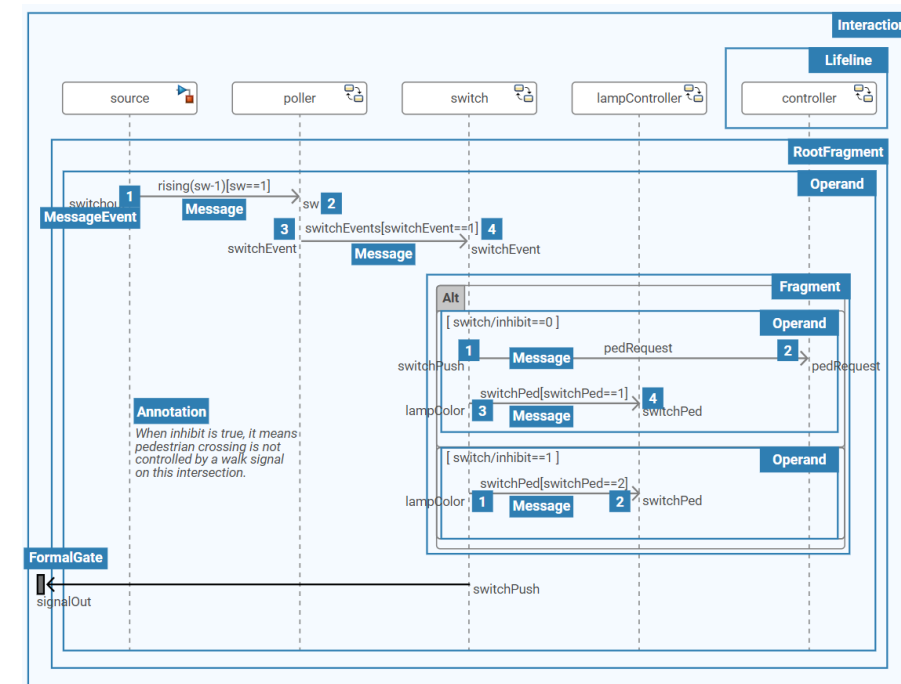
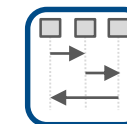
Návrh systémov s ohľadom na komplexnosť

- Tvorba architektúry systému v System Composer, ktorá prepája požiadavky a návrh
- Popis správania medzi systémovými komponentami
- Definovanie výmeny správ v sekvenčných diagramoch

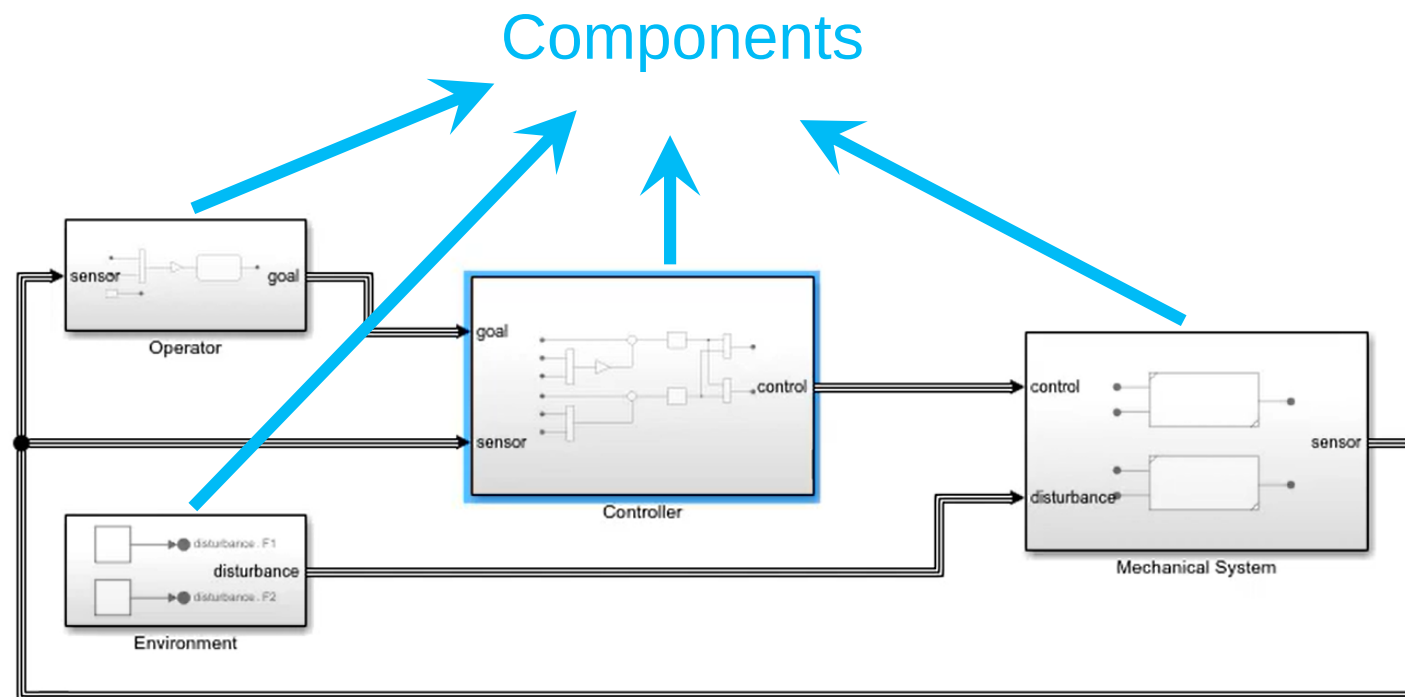
State Charts



Sequence Diagrams

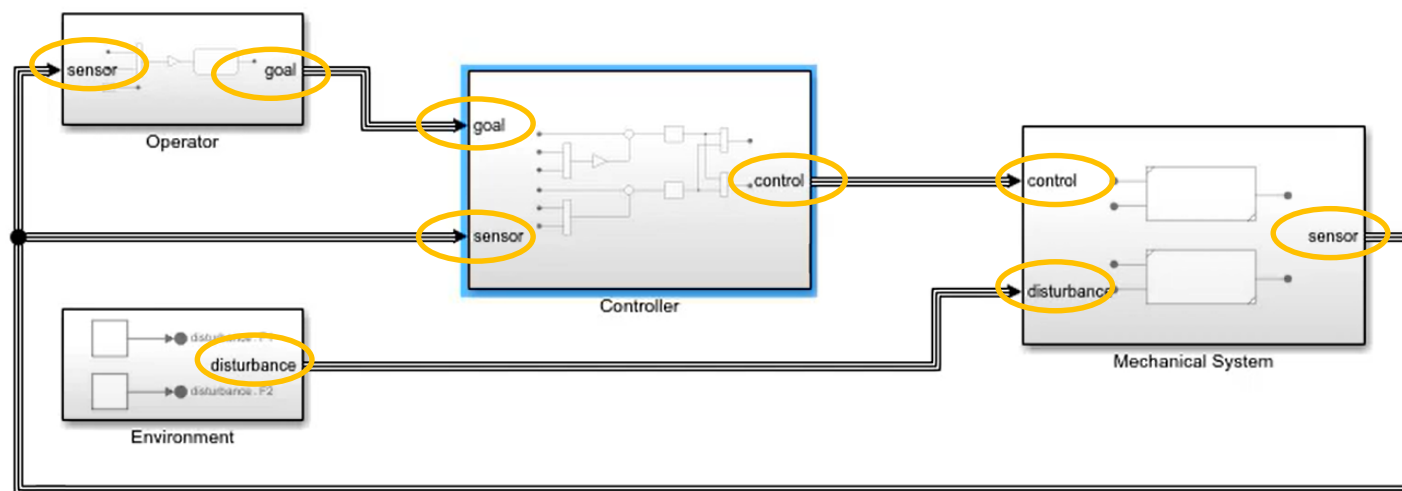


Sledovanie dátových rozhraní medzi komponentmi

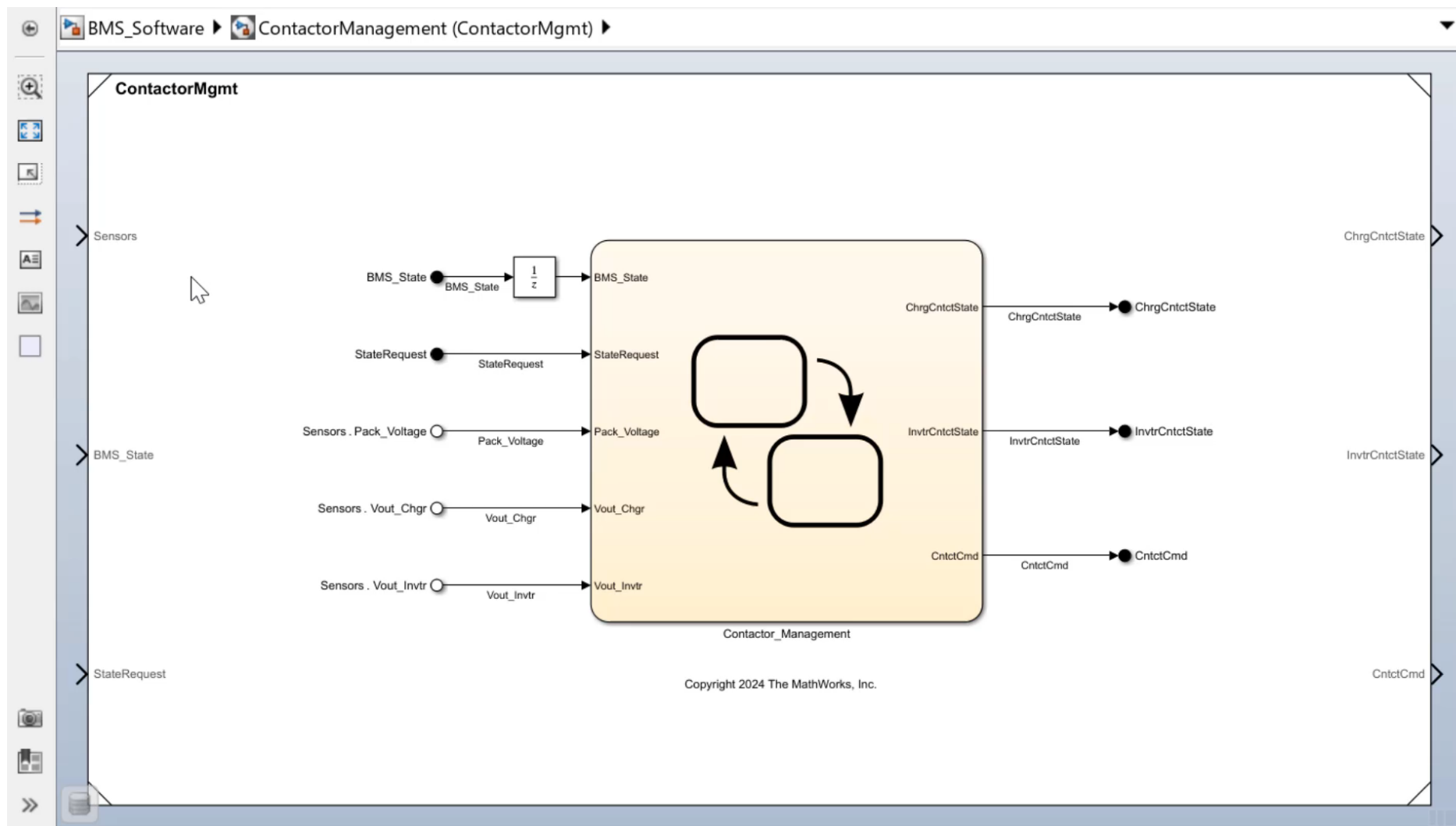


Sledovanie dátových rozhraní medzi komponentmi

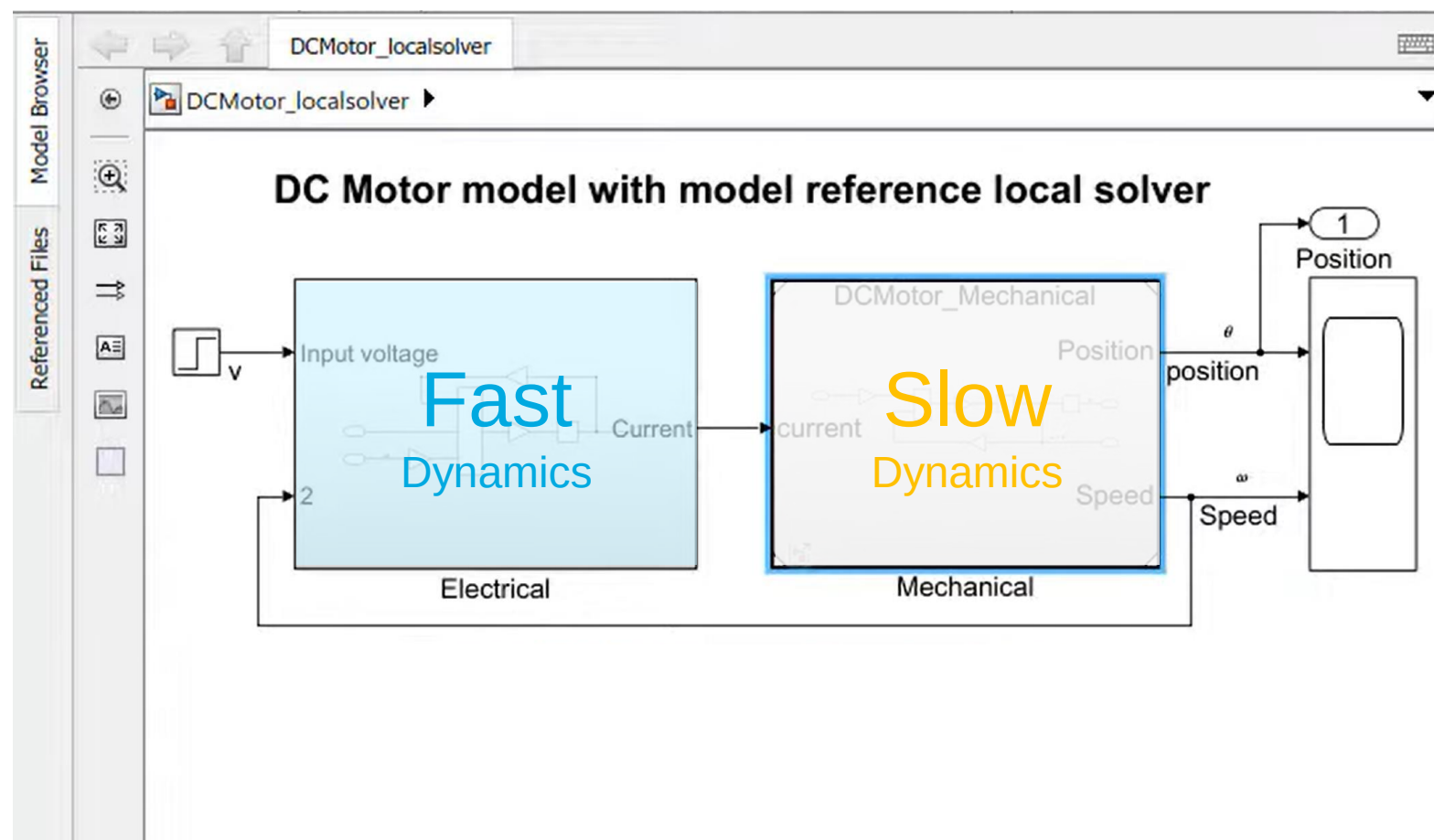
Interfaces



Sledovanie dátových rozhraní medzi komponentmi

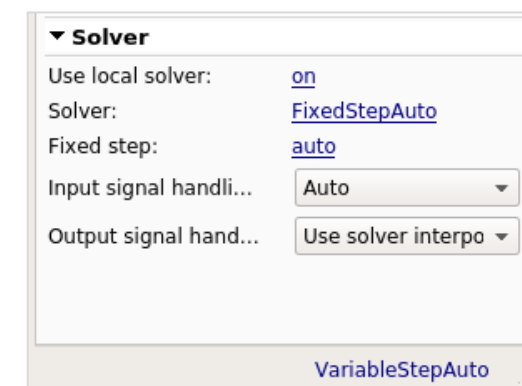
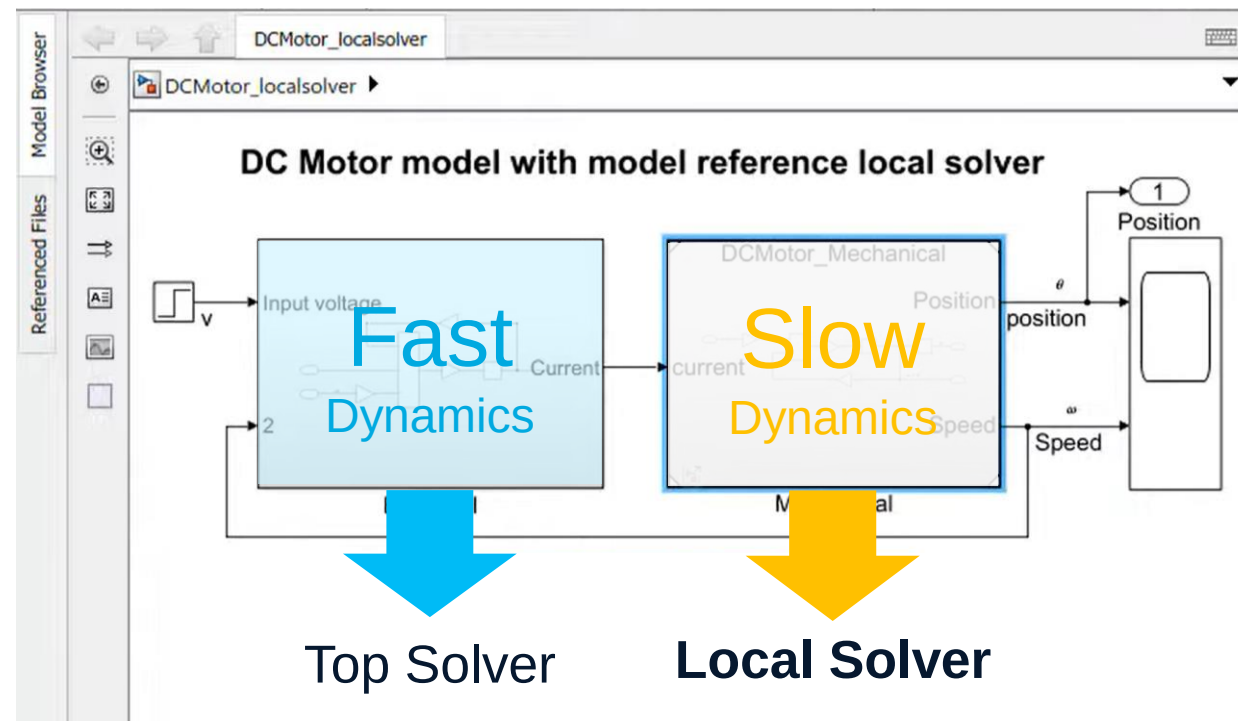


Výber riešiča zodpovedajúcemu potrebám komponentov



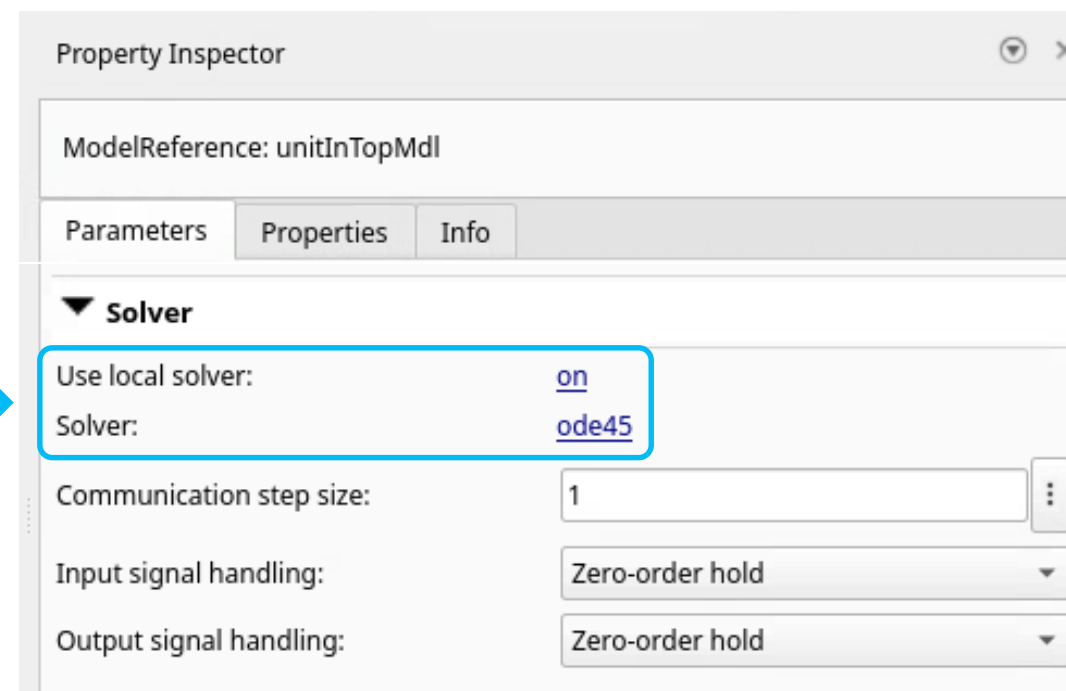
Výber riešiča zodpovedajúcemu potrebám komponentov

- Priradíte lokálne riešiče komponentom pre optimalizáciu výkonu simulácie



Výber riešiča zodpovedajúcemu potrebám komponentov

- Priradíte lokálne riešiče komponentom pre optimalizáciu výkonu simulácie
- Konfigurácia lokálnych riešičov pre referencie
 - riešiče s premenlivým a pevným krokom
 - detekcia prechodu nulou



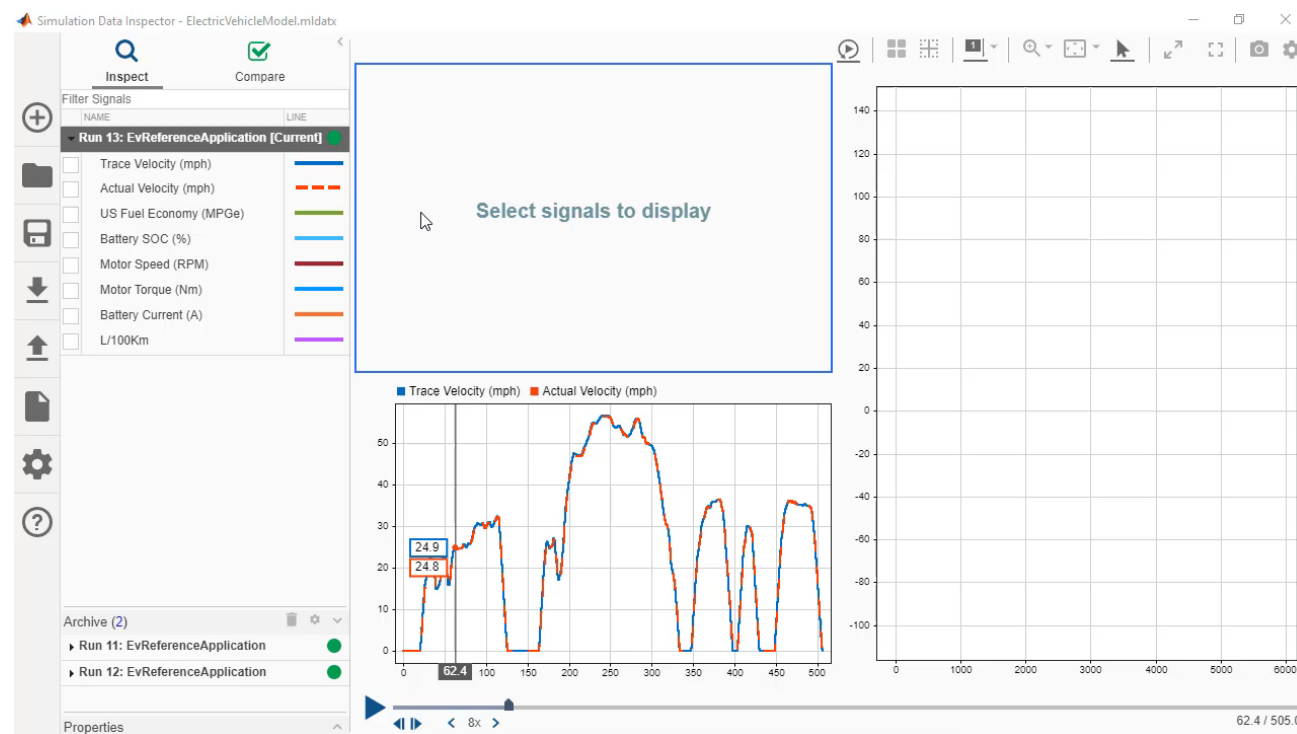
Spúšťajte simulácie a analyzujte dáta ešte rýchlejšie

- Fast Restart s Rapid Accelerator na spustenie viacerých simulácií bez nutnosti opätovnej kompilácie

```
1  %% Parallel Simulations Using Parsim: Rapid Accelerator + Fast Restart
2  mdl = 'sldemo_suspn_3dof';
3  isModelOpen = bdIsLoaded(mdl);
4  open_system(mdl);
5
6  % Set Up Multiple Simulations Using SimulationInput Objects
7  Mb_sweep = Mb*(0.5:1:45.5);
8  numSims = length(Mb_sweep);
9
10 % Define simulation inputs
11 for i = numSims:-1:1
12     in(i) = Simulink.SimulationInput(mdl);
13     in(i) = in(i).setModelParameter('SimulationMode', 'rapid');
14     in(i) = in(i).setVariable('Mb', Mb_sweep(i));
15 end
16
17 % Run simulations in parallel with Fast Restart
18 out = parsim(in, 'ShowProgress', 'on', 'UseFastRestart', 'on');
19
20
```

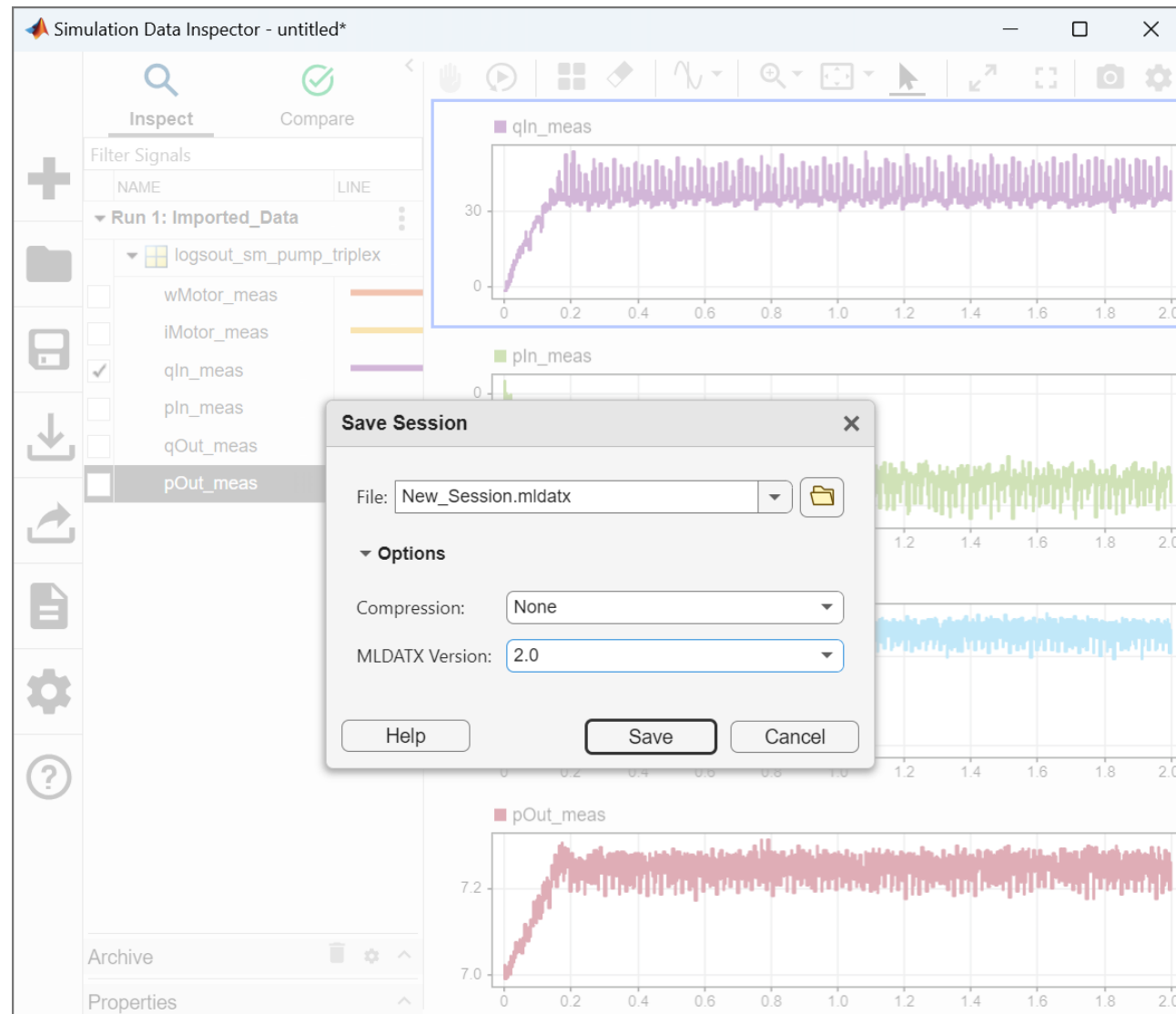
Spúšťajte simulácie a analyzujte dáta ešte rýchlejšie

- Fast Restart s Rapid Accelerator na spustenie viacerých simulácií bez nutnosti opätovnej kompilácie
- Zobrazenie údajov z viacerých simulácií v Simulation Data Inspector



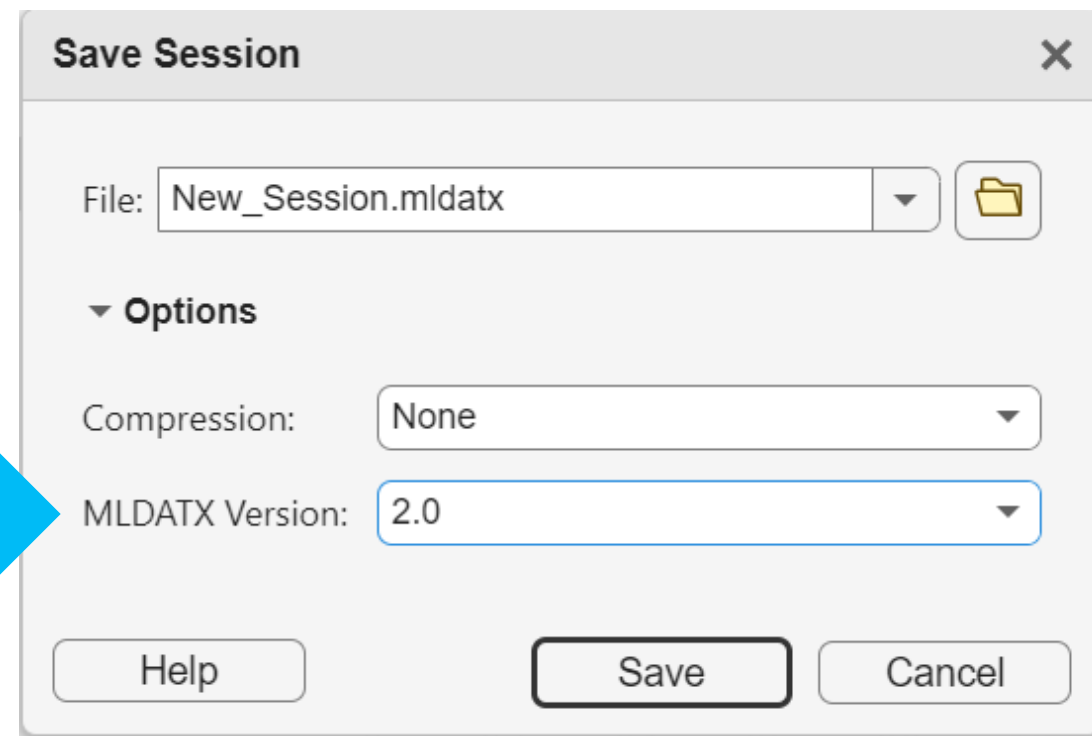
Spúšťajte simulácie a analyzujte dáta ešte rýchlejšie

- Fast Restart s Rapid Accelerator na spustenie viacerých simulácií bez nutnosti opätovnej kompilácie
- Zobrazenie údajov z viacerých simulácií v Simulation Data Inspector
- Rýchle ukladanie a načítavanie veľkých súborov údajov



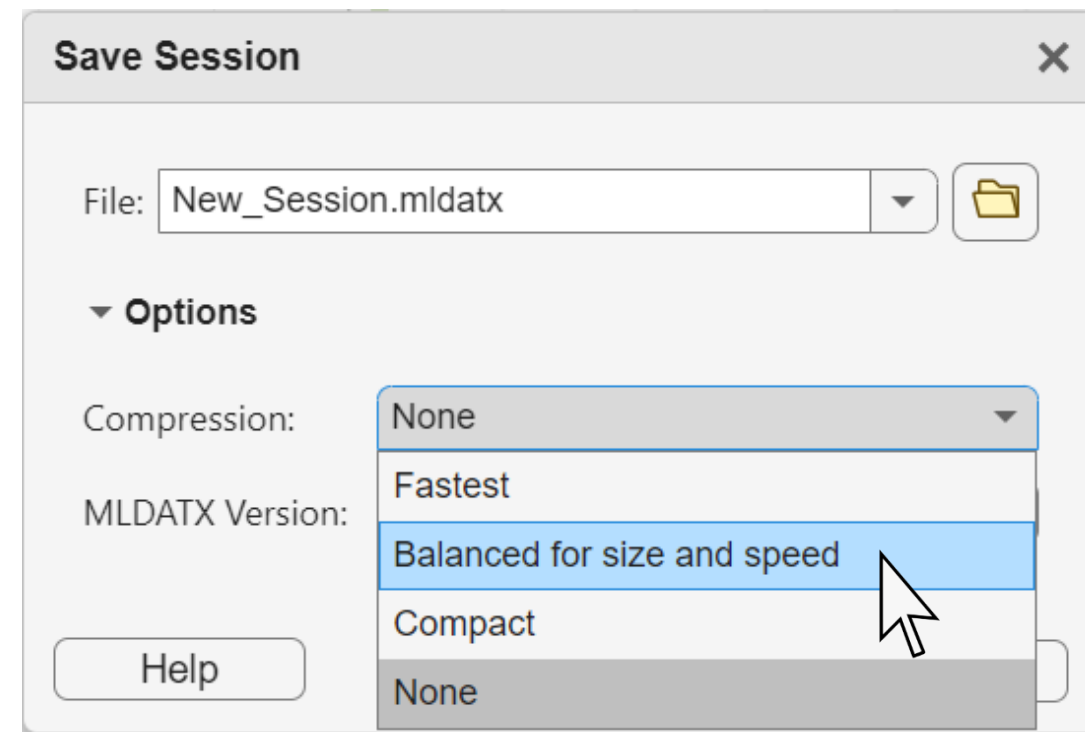
Spúšťajte simulácie a analyzujte dáta ešte rýchlejšie

- Fast Restart s Rapid Accelerator na spustenie viacerých simulácií bez nutnosti opätovnej kompilácie
- Zobrazenie údajov z viacerých simulácií v Simulation Data Inspector
- Rýchle ukladanie a načítavanie veľkých súborov údajov



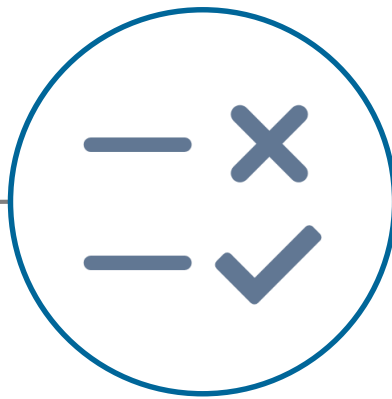
Spúšťajte simulácie a analyzujte dáta ešte rýchlejšie

- Fast Restart s Rapid Accelerator na spustenie viacerých simulácií bez nutnosti opätovnej kompilácie
- Zobrazenie údajov z viacerých simulácií v Simulation Data Inspector
- Rýchle ukladanie a načítavanie veľkých súborov údajov
- Vyváženie veľkosti a rýchlosti pri kompresii dátových súborov

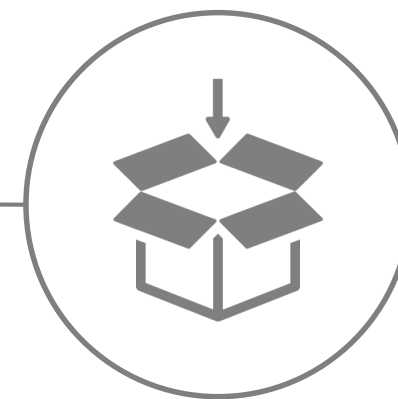




**Objavovanie a
tvorba**

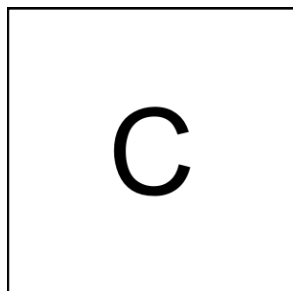


**Integrácia a
verifikácia**

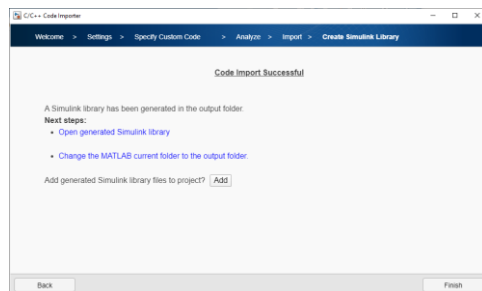


**Zdieľanie a
nasadenie**

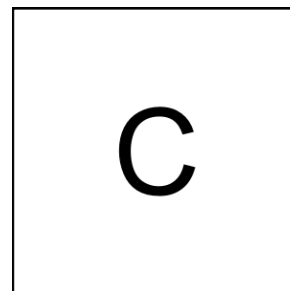
Integrácia vlastného kódu do Simulinku



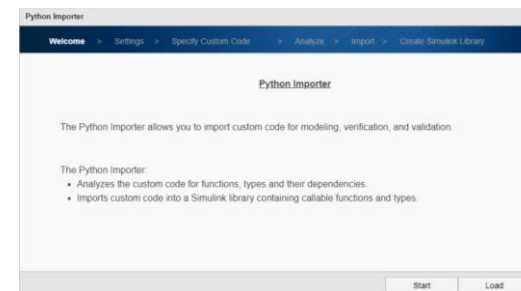
C Function



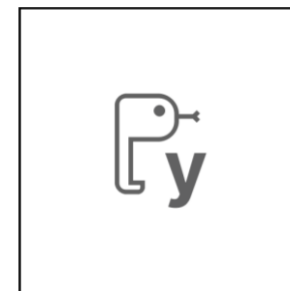
Code Importer



C Function
Supports C++



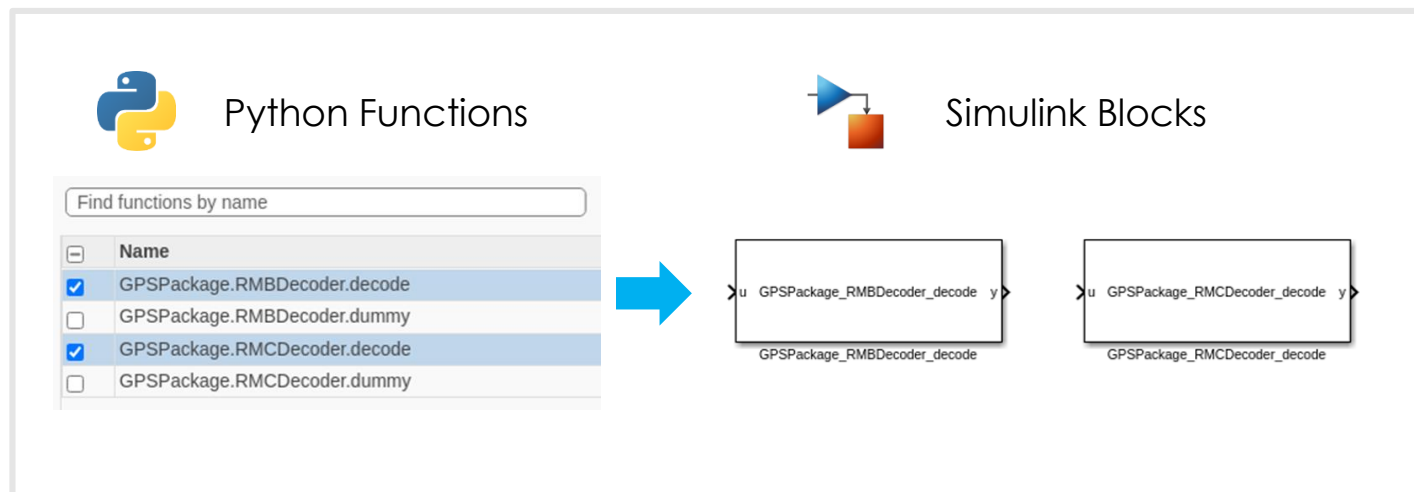
Python Importer



Python Code
Block

Integrácia vlastného kódu do Simulinku

- Tvorba blokov Simulinku z funkcií Pythonu



Integrácia vlastného kódu do Simulinku

- Tvorba blokov Simulinku z funkcií Pythonu

```
class room:

    def __init__(self, length, breadth, height):
        self.length = length
        self.breadth = breadth
        self.height = height

    def volume(self):
        result = self.length * self.breadth * self.height
        return result

    def wallarea(self):
        result = 2 * (self.length * height + self.breadth * height)
        return result
```

Python class definition

Integrácia vlastného kódu do Simulinku

- Tvorba blokov Simulinku z funkcií Pythonu
- Využite Python Importer na postupný import funkcií definovaných v triedach Pythonu

Python Importer

Welcome > Settings > Specify Custom Code > Analyze > **Import** > Create

Edit the block port specifications for the previously selected functions.

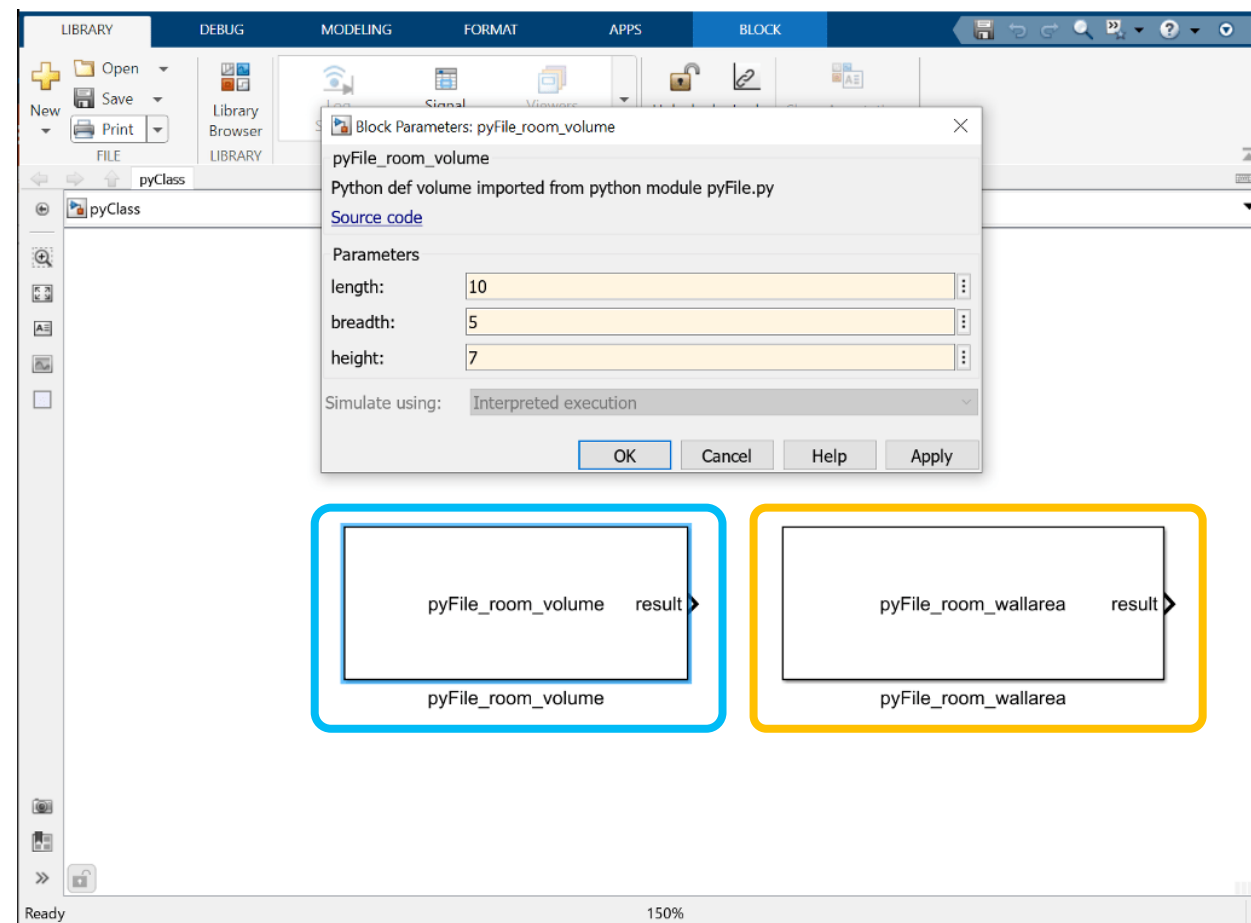
Search

Name	Scope	Label	Type	Complexity	Size
fx pyFile.room.room					
self	Parameter	self	ClassObject	real	[1 1]
length	Parameter	length	double	real	[1 1]
breadth	Parameter	breadth	double	real	[1 1]
height	Parameter	height	double	real	[1 1]
fx pyFile.room.volume					
result	Output	result	double	real	[1 1]
self	Parameter	self	ClassObject	real	[1 1]
fx pyFile.room.wallarea					
result	Output	result	double	real	[1 1]
self	Parameter	self	ClassObject	real	[1 1]

Back

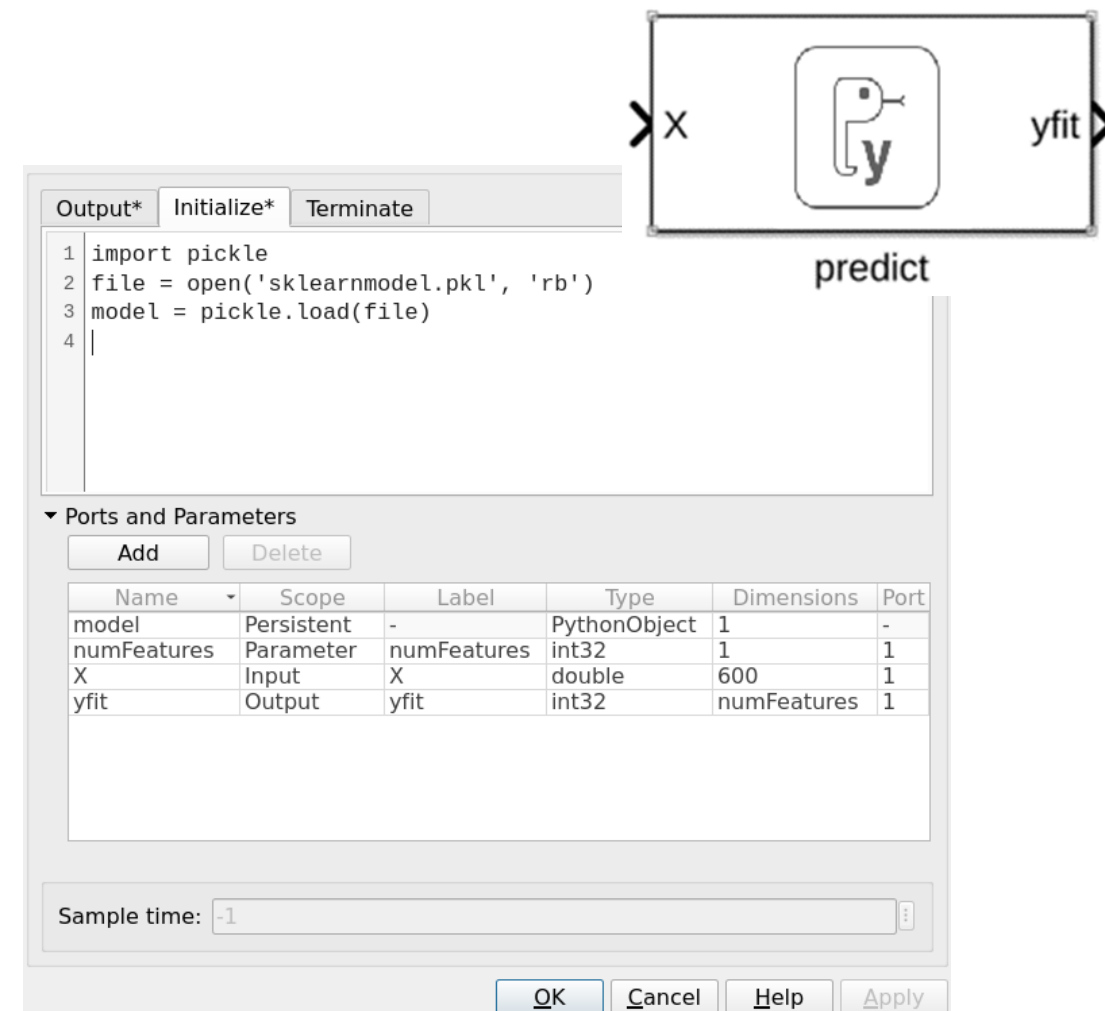
Integrácia vlastného kódu do Simulinku

- Tvorba blokov Simulinku z funkcií Pythonu
- Využite Python Importer na postupný import funkcií definovaných v triedach Pythonu



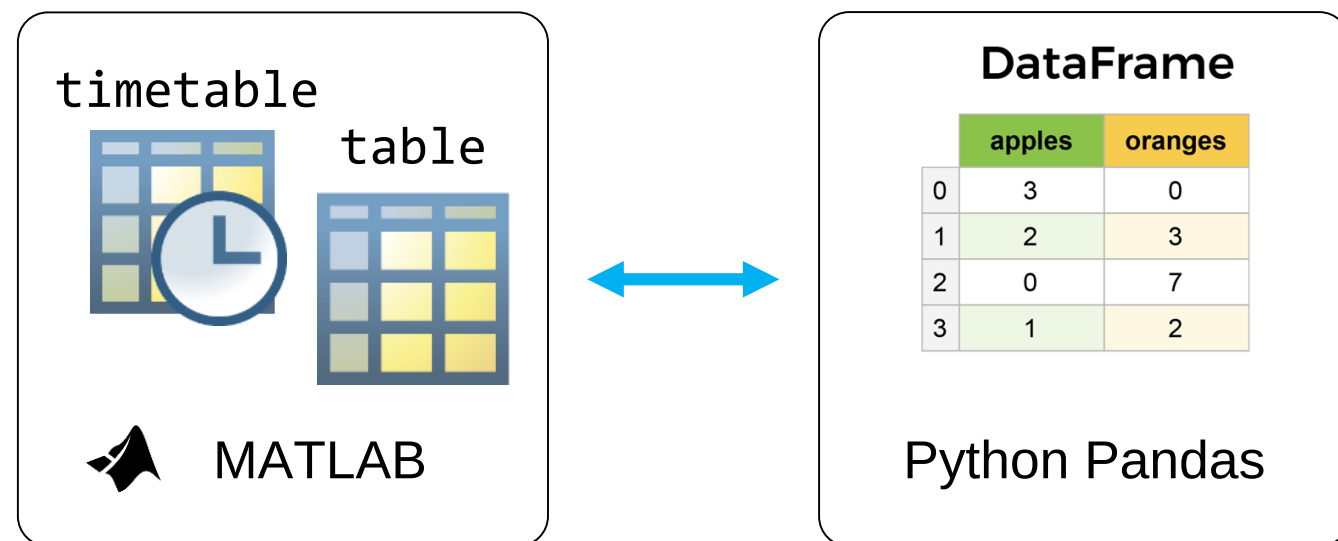
Integrácia vlastného kódu do Simulinku

- Tvorba blokov Simulinku z funkcií Pythonu
- Využite Python Importer na postupný import funkcií definovaných v triedach Pythonu
- Píšte natívny kód Pythonu v dialógovom okne bloku Simulinku



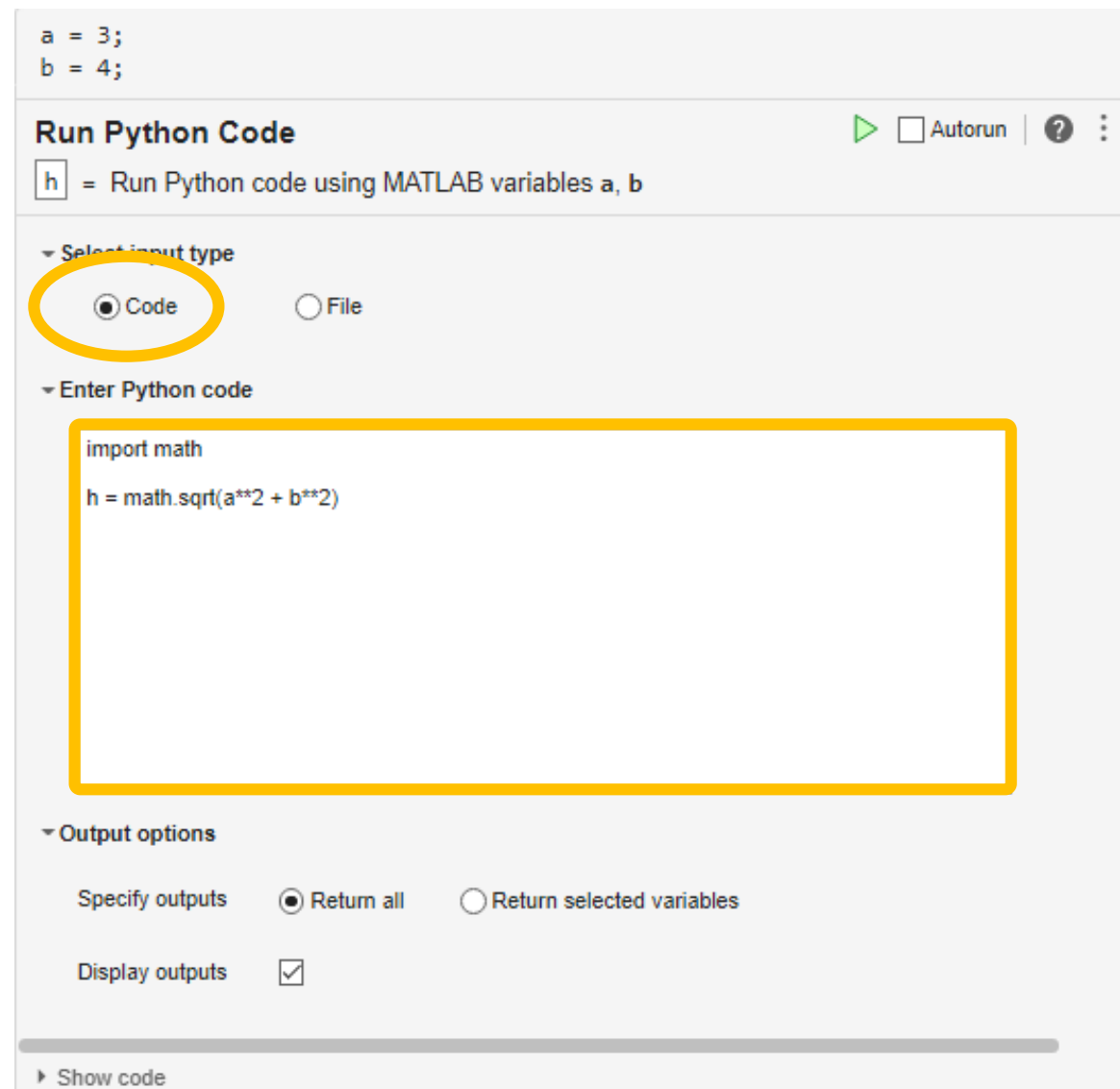
Výměna dát – MATLAB & Python

- Automatická konverzia medzi:
 - MATLAB tabuľky / časové tabuľky a Python Pandas Dataframes
 - MATLAB datetime / duration typy a relevantné dátové typy Pythonu



Interaktívne spúšťanie kódu a funkcií Pythonu

- Písanie a spúšťanie Python kódu
- Spúšťanie Python kódu v súboroch
- Vrátenie dát z Pythonu do pracovného priestoru MATLABu
- Posielanie dát z MATLABu do Python kódu
- Špecifikácia argumentov



The image shows the 'Run Python Code' dialog box in MATLAB. At the top, it displays MATLAB code: `a = 3;` and `b = 4;`. Below this, the title 'Run Python Code' is followed by a green play button, an 'Autorun' checkbox, and a help icon. A text field contains the command `h = Run Python code using MATLAB variables a, b`. Under the 'Select input type' section, the 'Code' radio button is selected and circled in yellow, while the 'File' radio button is unselected. The 'Enter Python code' section, also outlined in yellow, contains the Python code: `import math` and `h = math.sqrt(a**2 + b**2)`. The 'Output options' section at the bottom has 'Specify outputs' unselected, 'Return all' selected, 'Return selected variables' unselected, and 'Display outputs' checked. A 'Show code' link is at the very bottom.

```
a = 3;
b = 4;
```

Run Python Code  ☐ Autorun  

`h` = Run Python code using MATLAB variables a, b

▼ Select input type

☒ Code ☐ File

▼ Enter Python code

```
import math
h = math.sqrt(a**2 + b**2)
```

▼ Output options

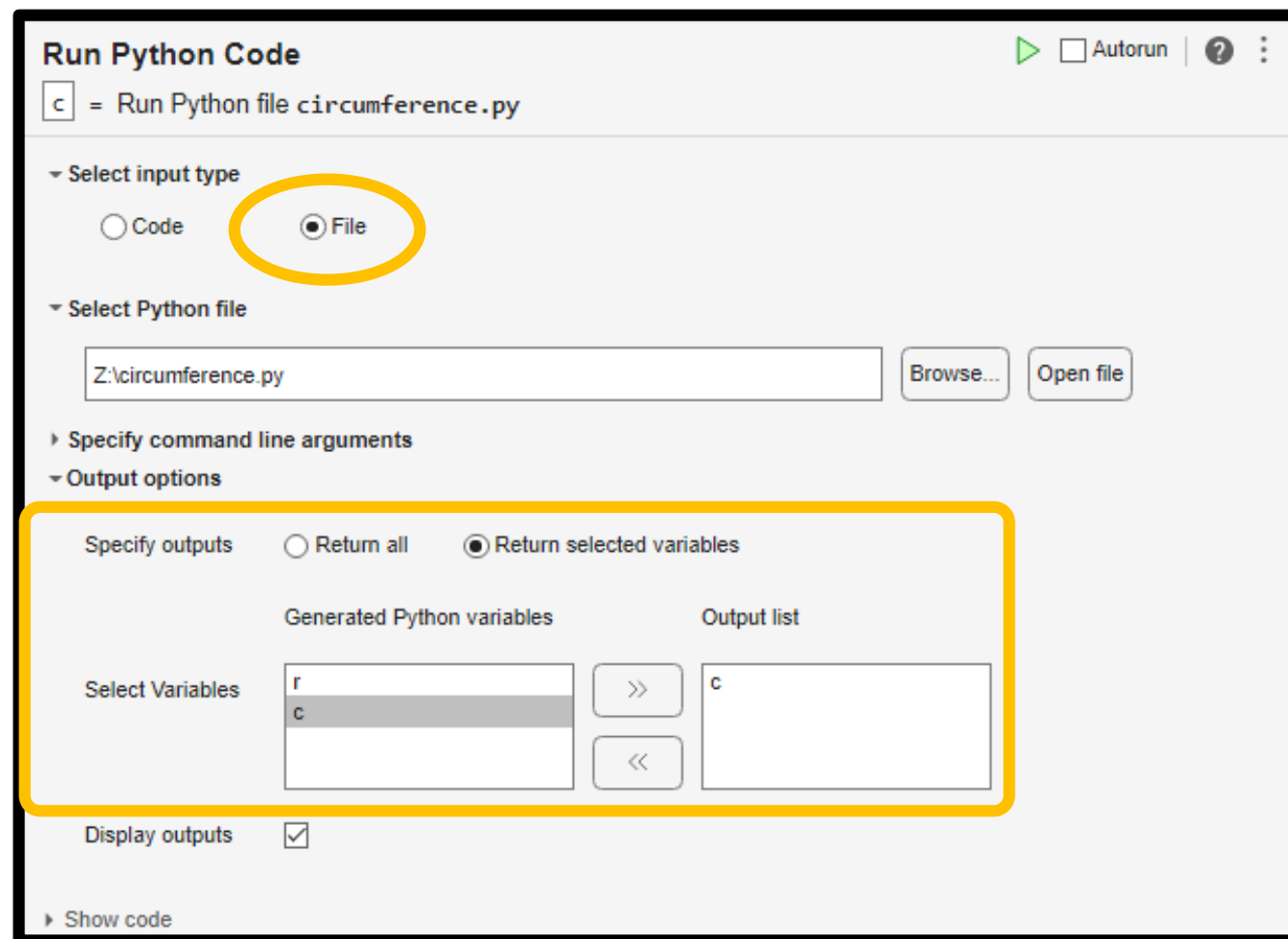
Specify outputs ☐ Return all ☐ Return selected variables

Display outputs ☒

► Show code

Interaktívne spúšťanie kódu a funkcií Pythonu

- Písanie a spúšťanie Python kódu
- Spúšťanie Python kódu v súboroch
- Vrátenie dát z Pythonu do pracovného priestoru MATLABu
- Posielanie dát z MATLABu do Python kódu
- Špecifikácia argumentov



Run Python Code ▶ ☐ Autorun ? ⋮

☒ = Run Python file `circumference.py`

▼ Select input type

☐ Code ☒ File

▼ Select Python file

Browse... Open file

► Specify command line arguments

▼ Output options

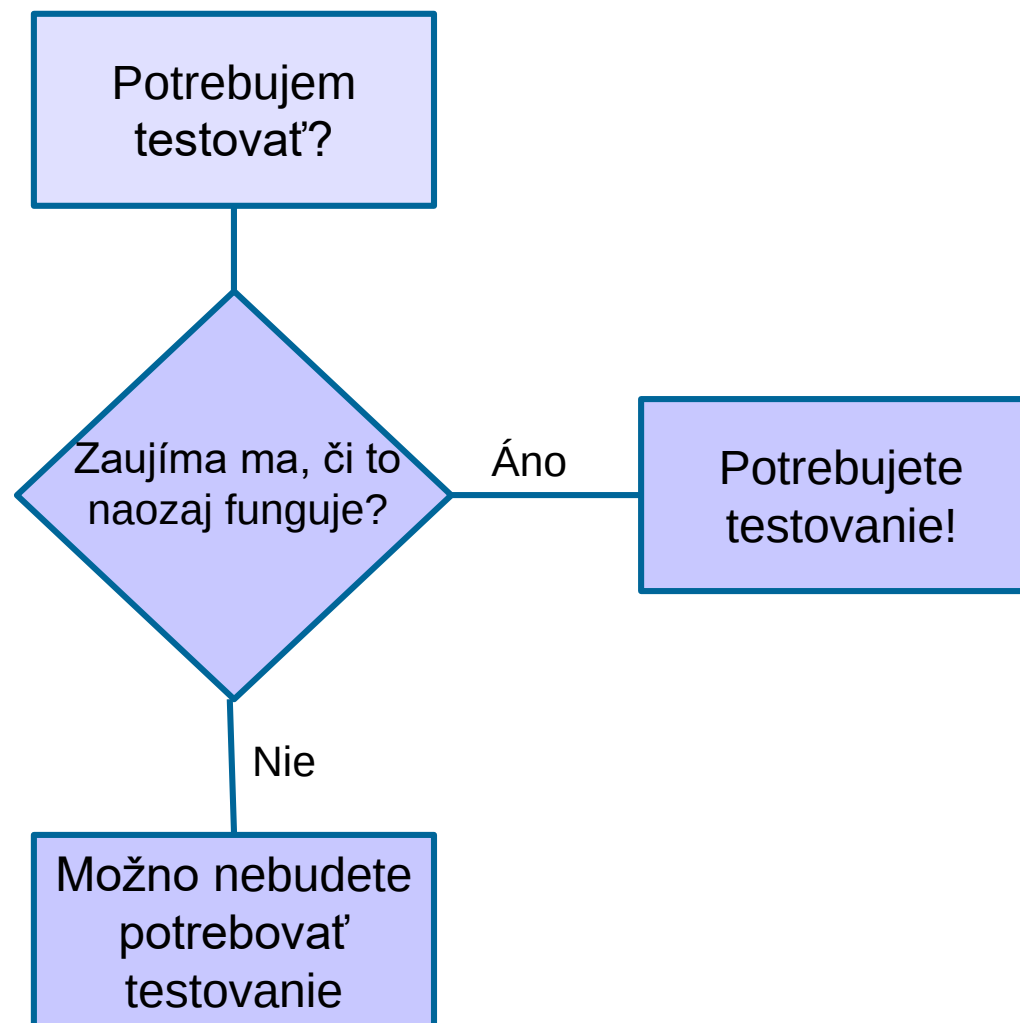
Specify outputs ☐ Return all ☒ Return selected variables

	Generated Python variables		Output list
Select Variables	r	>> <<	c
	c		

Display outputs ☒

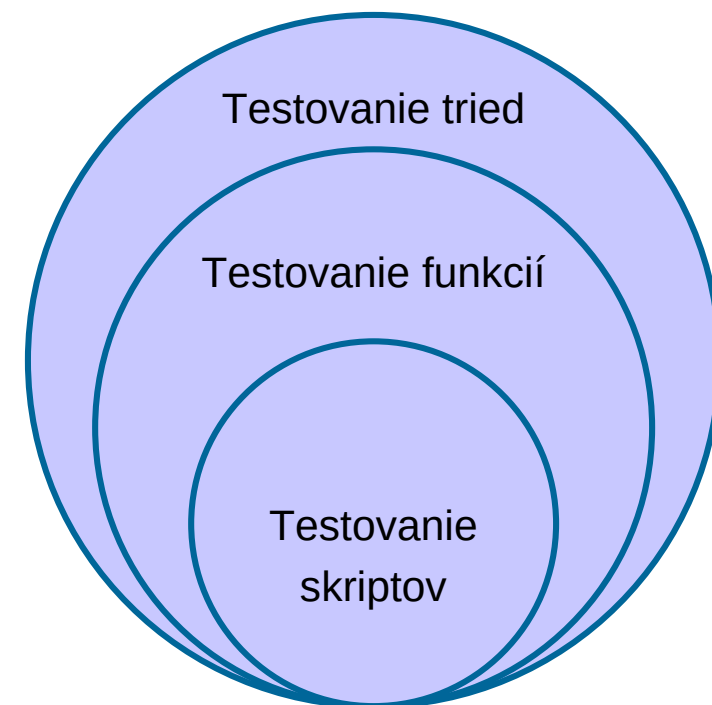
► Show code

Testovanie



Využite vstavané testovacie možnosti

- Unit testing framework
 - testy založené na skriptoch
 - testy založené na funkciách
 - testy založené na triedach
- Performance testing framework
 - Merania
 - štart kódu
 - zohľadnenie šumu
 - Využívanie rozhrania unit testovania
- Šablóny pre rýchly začiatok

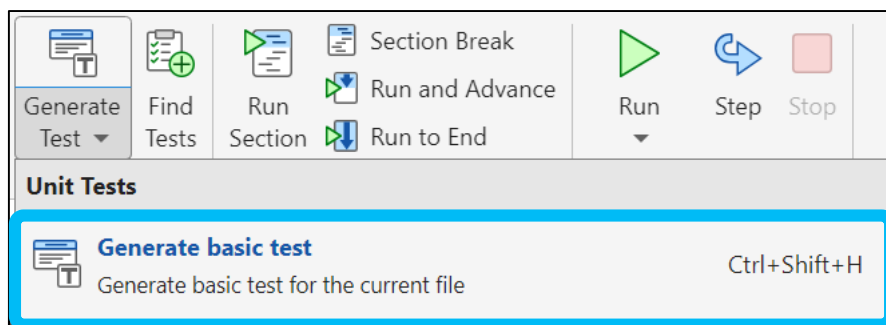


MATLAB Test dokáže generovať testy

- Základný test pre súbor

```
function r = quadraticSolver(a,b,c)
% quadraticSolver returns solutions to the
% quadratic equation a*x^2 + b*x + c = 0.

r = sort((-b + sqrt(b^2 - 4*a*c)) / (2*a); ...
        (-b - sqrt(b^2 - 4*a*c)) / (2*a));
end
```



```
% This is an autogenerated sample test for file quadraticSolver.m
classdef testquadraticSolver < matlab.unittest.TestCase

    methods (Test)

        function test quadraticSolver(testCase)

            % Specify the input(s) of quadraticSolver
            a =
            b =
            c =

            % Specify the expected output(s) of quadraticSolver
            expected_r =

            % Exercise the function quadraticSolver
            actual_r = quadraticSolver(a, b, c);

            testCase.verifyEqual(actual_r, expected_r);

        end
    end
end
```

MATLAB Test dokáže generovať testy

- Základný test pre súbor
- Test z histórie príkazov

Suppose that you executed these commands.

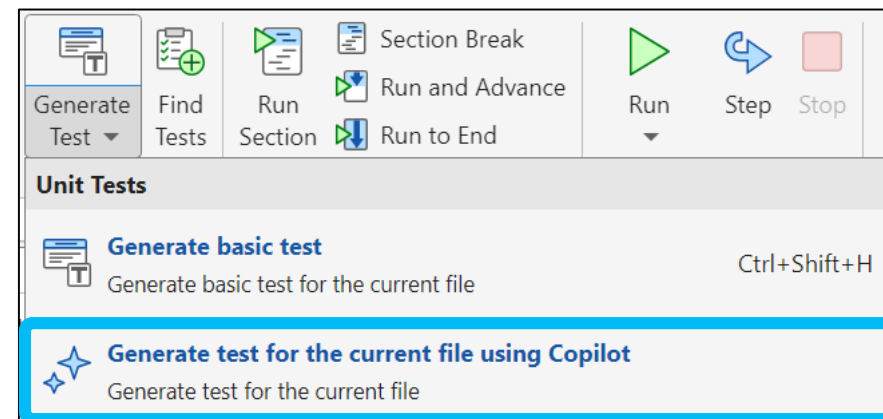
```
>> a = 2;  
>> b = 2;  
>> c = -4;  
>> r = quadraticSolver(a,b,c);
```



```
% This is an autogenerated sample test for file quadraticSolver.m  
classdef testquadraticSolver < matlab.unittest.TestCase  
    methods (Test)  
        function test_quadraticSolver(testCase)  
            % Specify the input(s) of quadraticSolver  
            a = 2;  
            b = 2;  
            c = -4;  
  
            % Exercise the function quadraticSolver.m  
            actualOutput_r = quadraticSolver(a,b,c);  
  
            % Specify the expected output(s) of quadraticSolver  
            expectedOutput_r =  
  
            testCase.verifyEqual(actualOutput_r, expectedOutput_r);  
        end  
    end  
end
```

MATLAB Copilot na generovanie testov

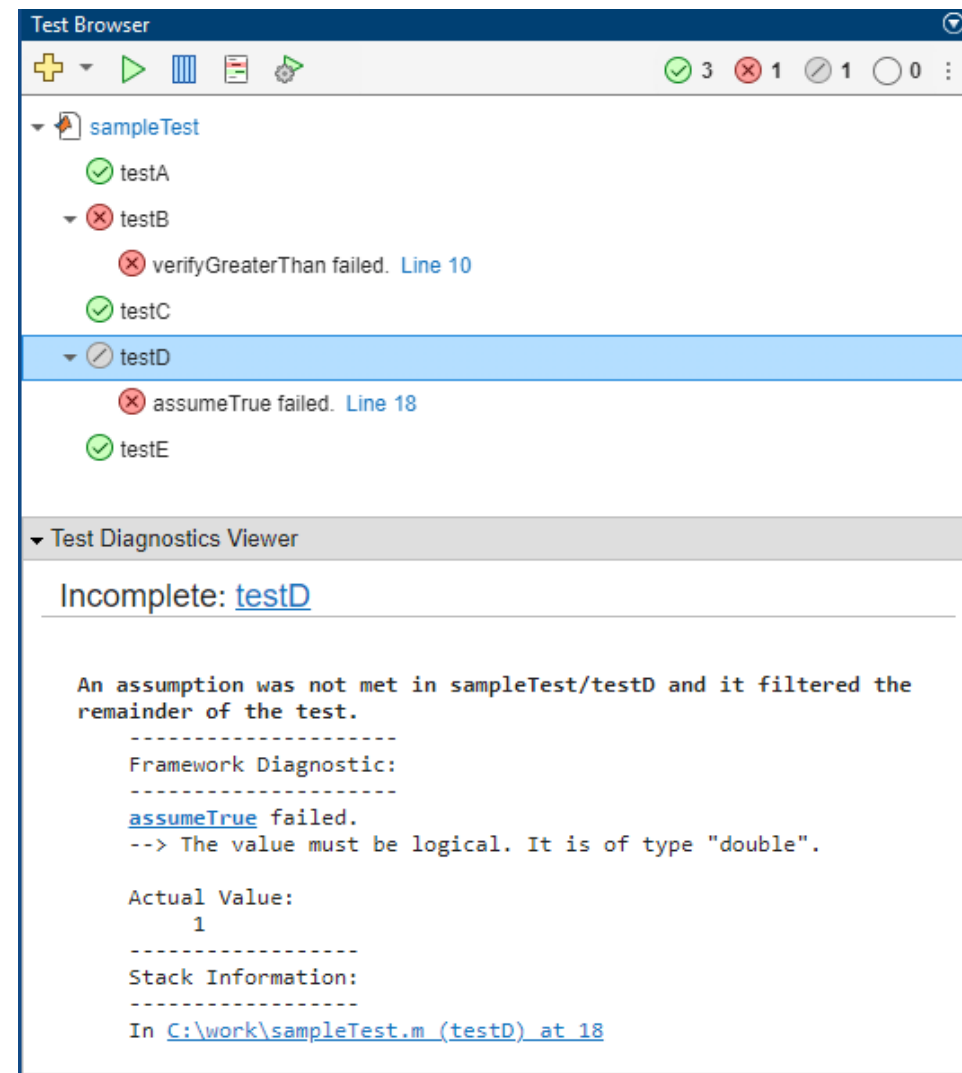
- Testovací súbor sa generuje automaticky
 - Môže obsahovať jeden alebo viac testov
- V generovaných testoch sú špecifikované vstupné aj očakávané výstupné hodnoty



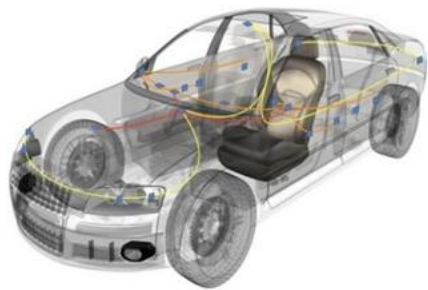
Pred použitím overte vygenerované testy a uistite sa, že testy vykonávajú správanie, ktoré chcete otestovať

Test Browser na spúšťanie a prezieranie výsledkov

- Otvorí sa ako nový panel v novom desktope
- Nové možnosti zahŕňajú:
 - opakované testovanie z posledného behu
 - identifikácia posledných výsledkov testu

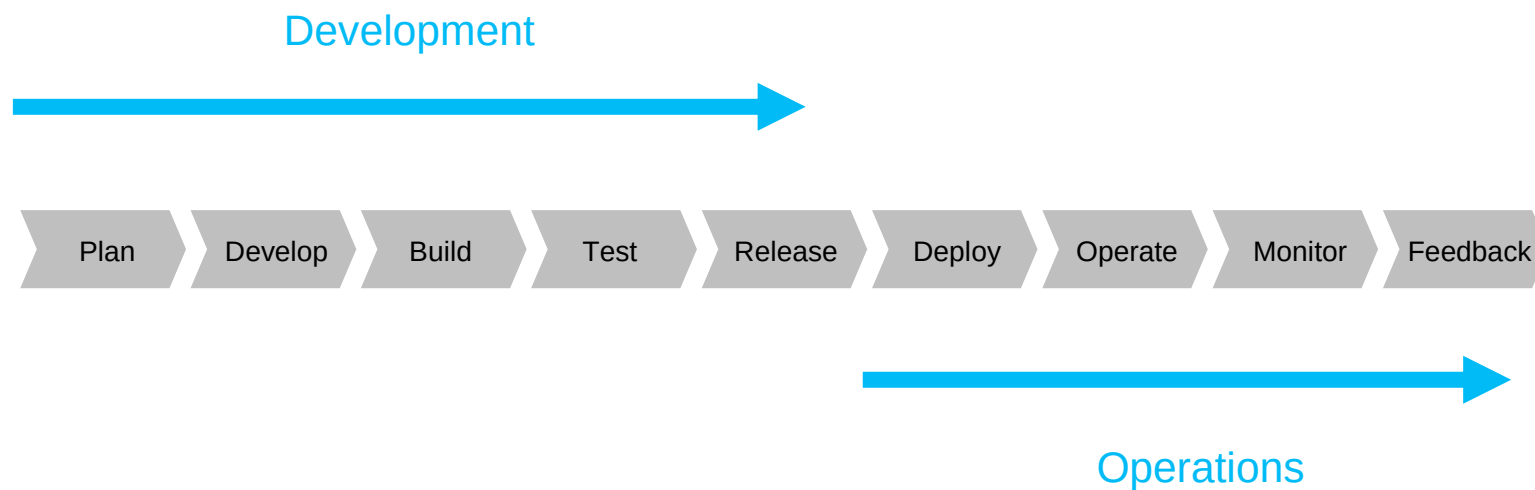


Transformácia vývoja a prevádzky systémov

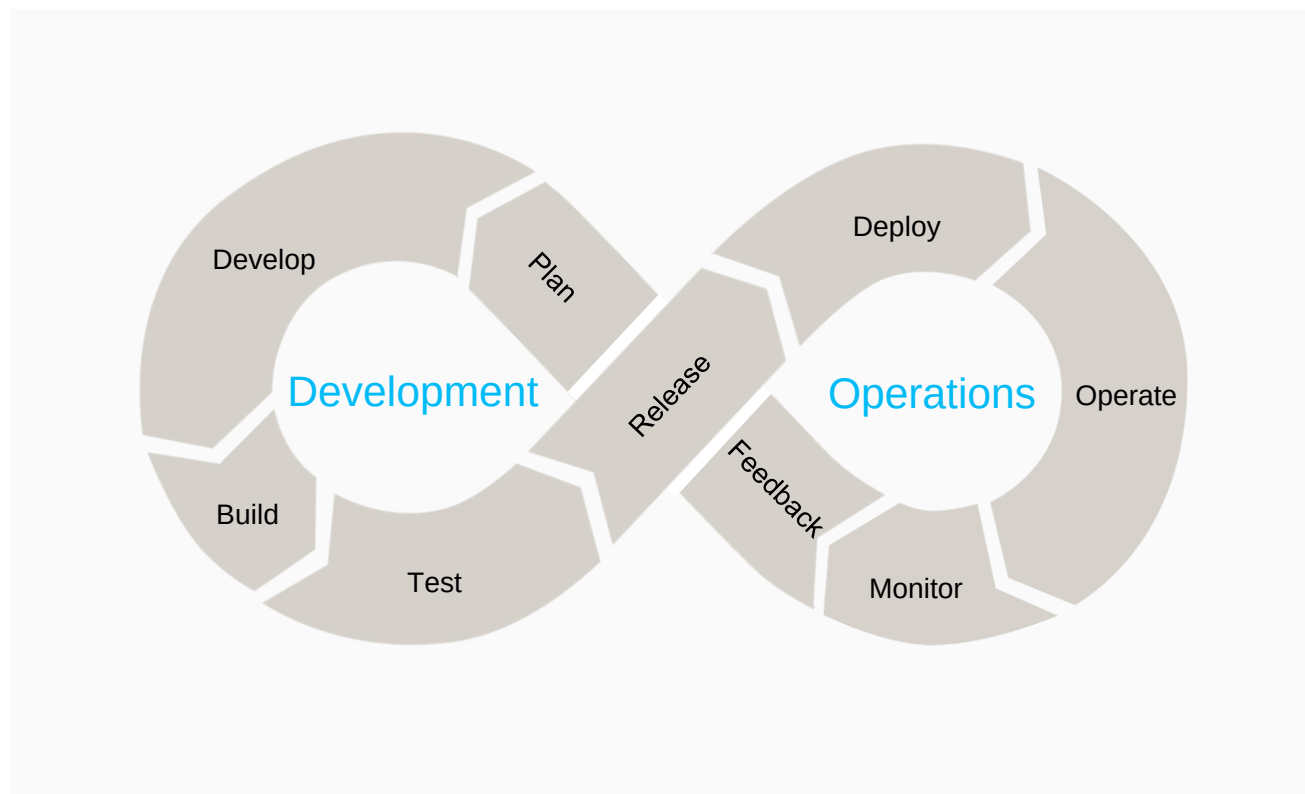


Software-Enabled Systems

Transformácia vývoja a prevádzky systémov

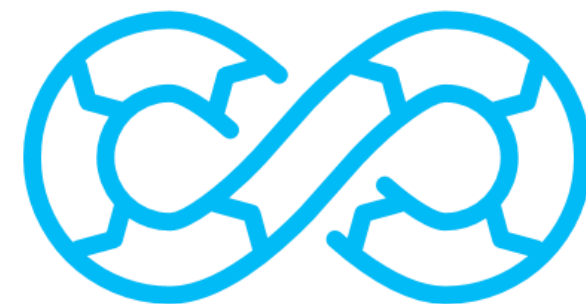


Transformácia vývoja a prevádzky systémov



Transformácia vývoja a prevádzky systémov

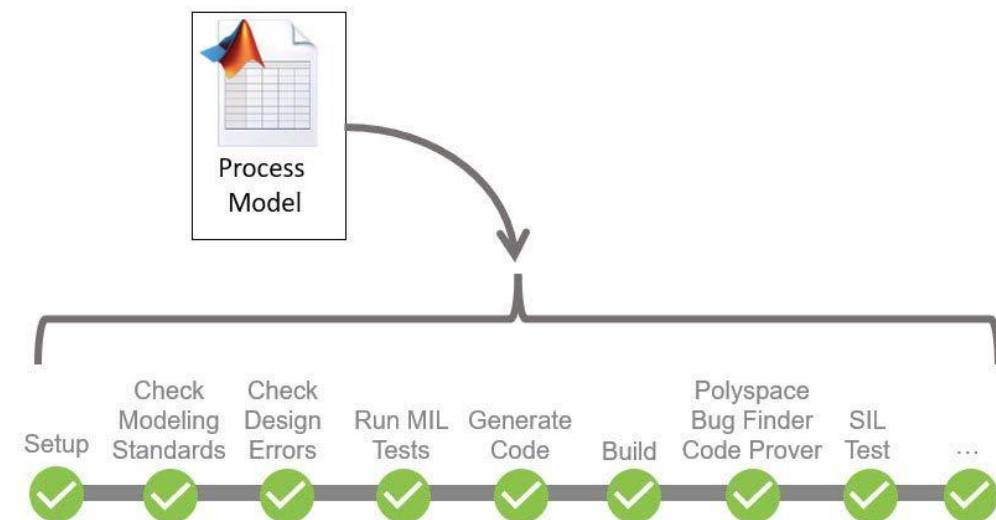
- Automatizované zostavovanie a overovanie MATLAB kódu a modelov Simulinku



CI Support Package
for Simulink

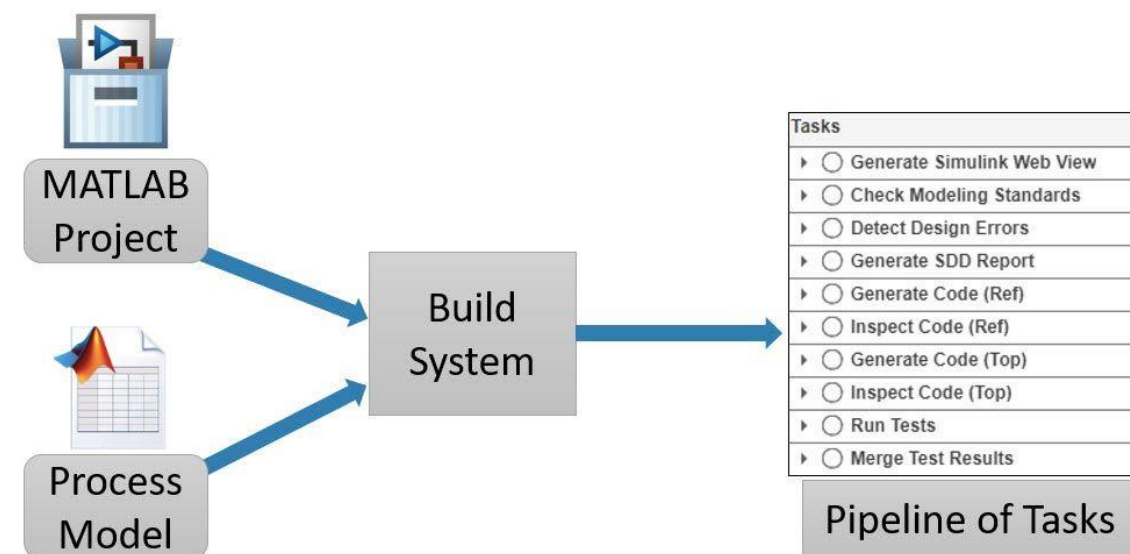
Transformácia vývoja a prevádzky systémov

- Automatizované zostavovanie a overovanie MATLAB kódu a modelov Simulinku
- Schopnosti:
 - definícia procesov zostavovania a overovania



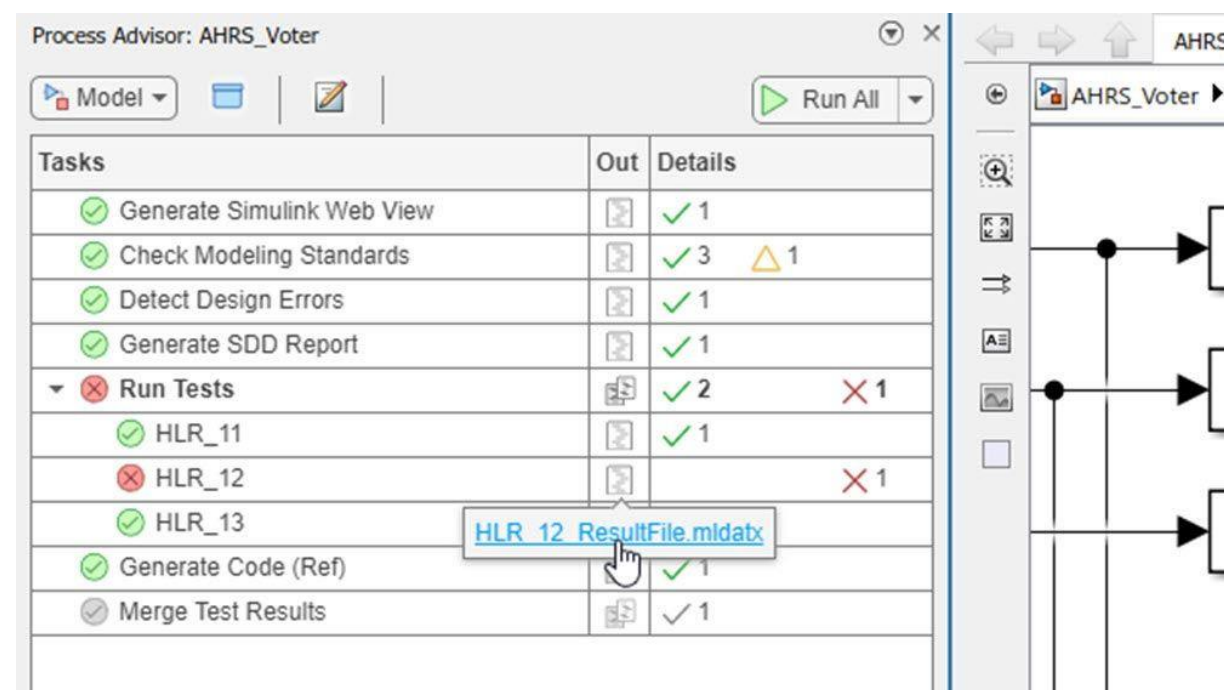
Transformácia vývoja a prevádzky systémov

- Automatizované zostavovanie a overovanie MATLAB kódu a modelov Simulinku
- Schopnosti:
 - definícia procesov zostavovania a overovania
 - generovanie postupov a ich postupné zostavovanie



Transformácia vývoja a prevádzky systémov

- Automatizované zostavovanie a overovanie MATLAB kódu a modelov Simulinku
- Schopnosti:
 - definícia procesov zostavovania a overovania
 - generovanie postupov a ich postupné zostavovanie
 - kvalifikácia lokálnych zmien



Process Advisor: AHRs_Voter

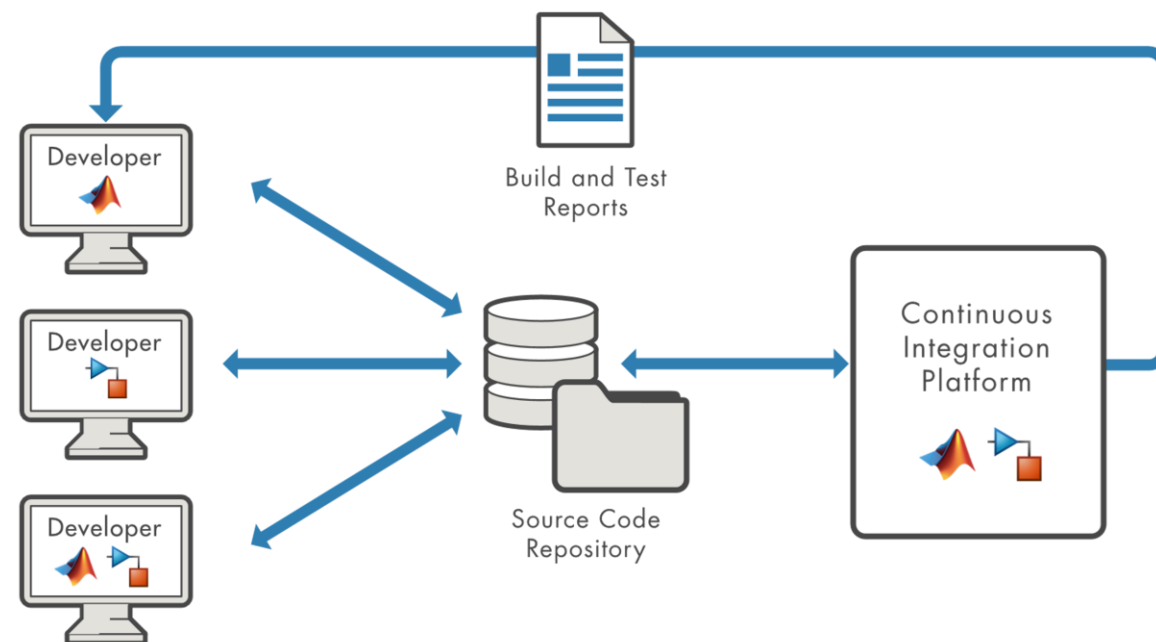
Model | Run All

Tasks	Out	Details
✓ Generate Simulink Web View	✓ 1	
✓ Check Modeling Standards	✓ 3	△ 1
✓ Detect Design Errors	✓ 1	
✓ Generate SDD Report	✓ 1	
✗ Run Tests	✓ 2	✗ 1
✓ HLR_11	✓ 1	
✗ HLR_12		✗ 1
✓ HLR_13		
✓ Generate Code (Ref)	✓ 1	
✓ Merge Test Results	✓ 1	

HLR_12 ResultFile.mldatx

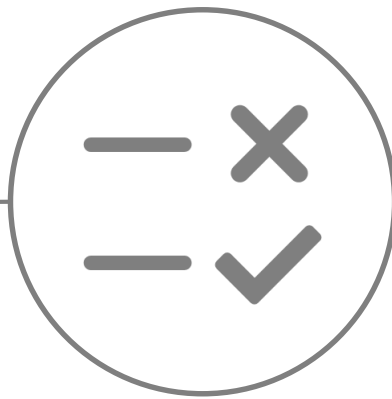
Transformácia vývoja a prevádzky systémov

- Automatizované zostavovanie a overovanie MATLAB kódu a modelov Simulinku
- Schopnosti:
 - definícia procesov zostavovania a overovania
 - generovanie postupov a ich postupné zostavovanie
 - kvalifikácia lokálnych zmien
 - integrácia so systémami CI/CD

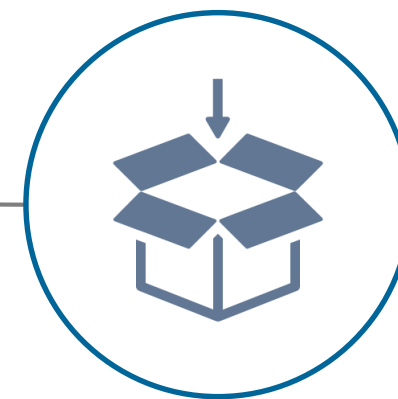




**Objavovanie a
tvorba**

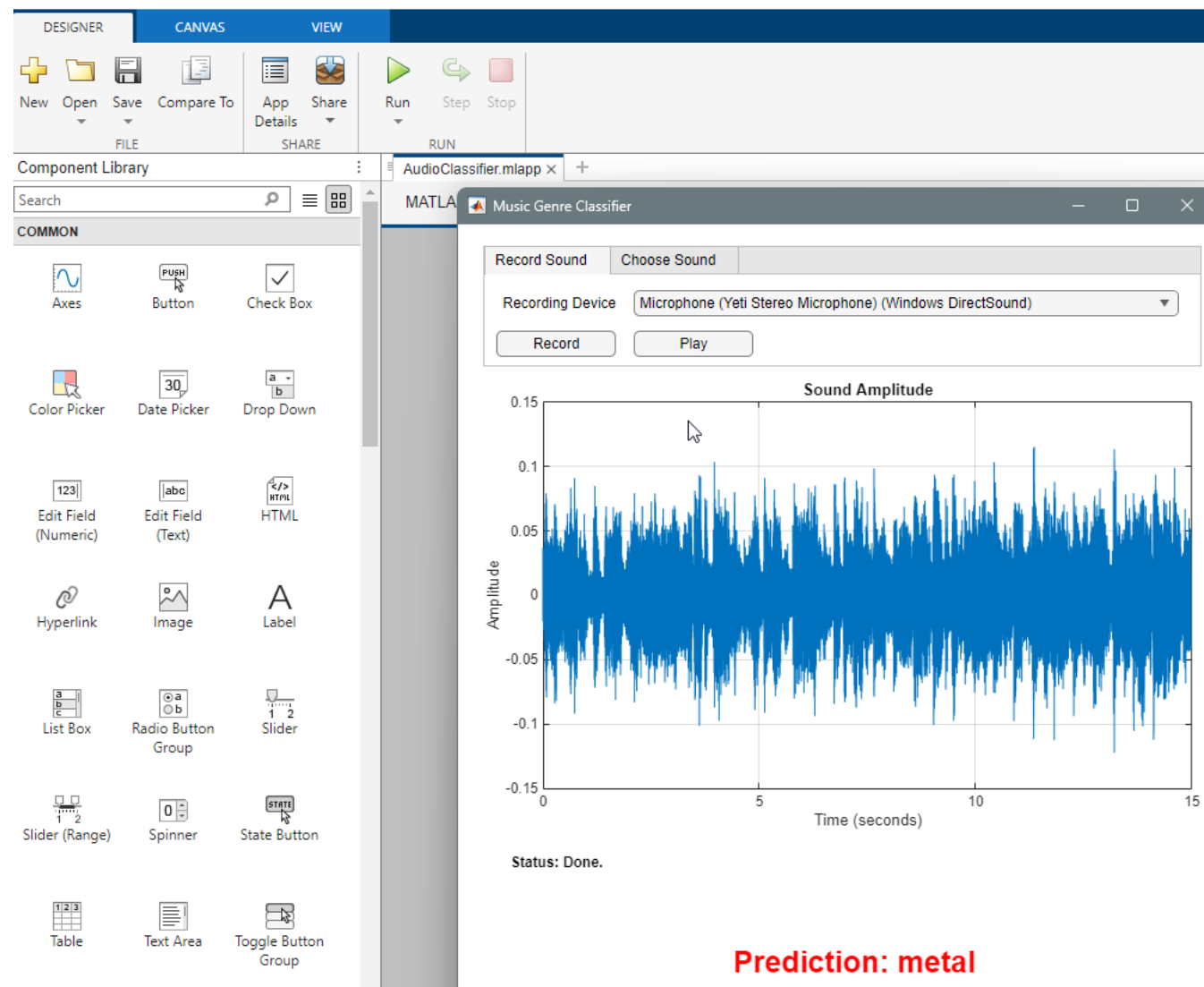


**Integrácia a
verifikácia**

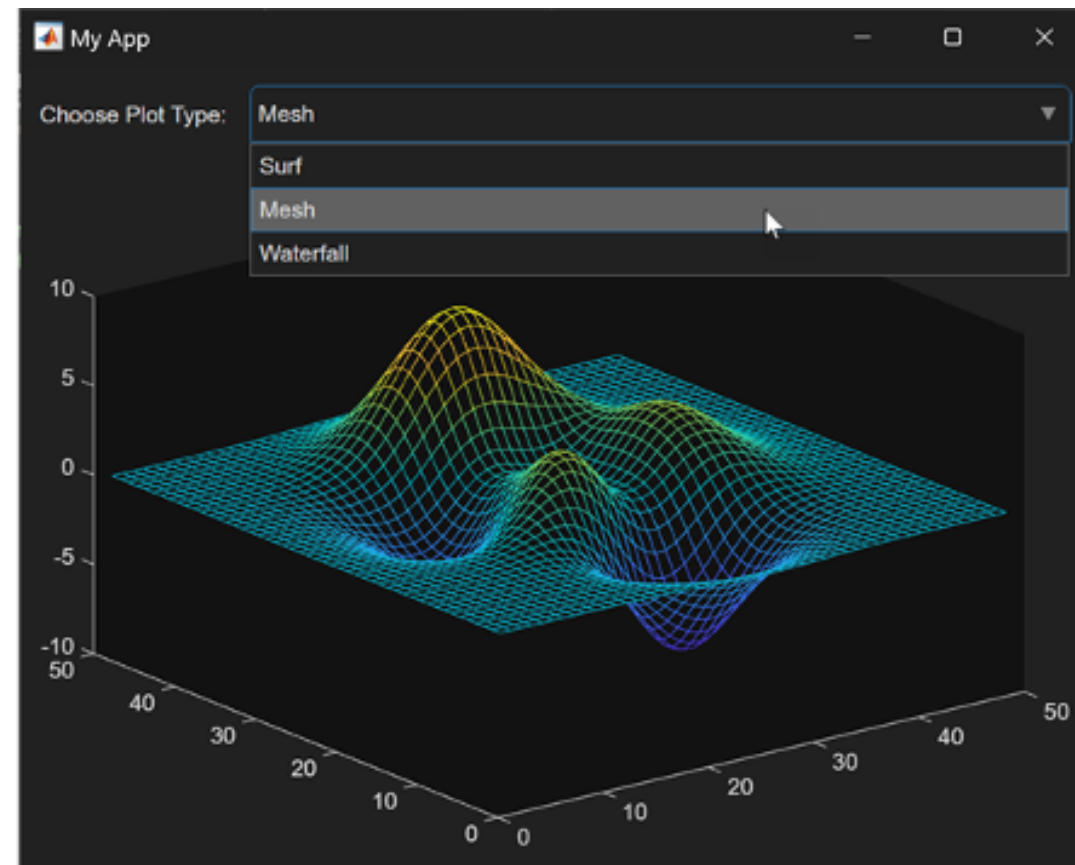
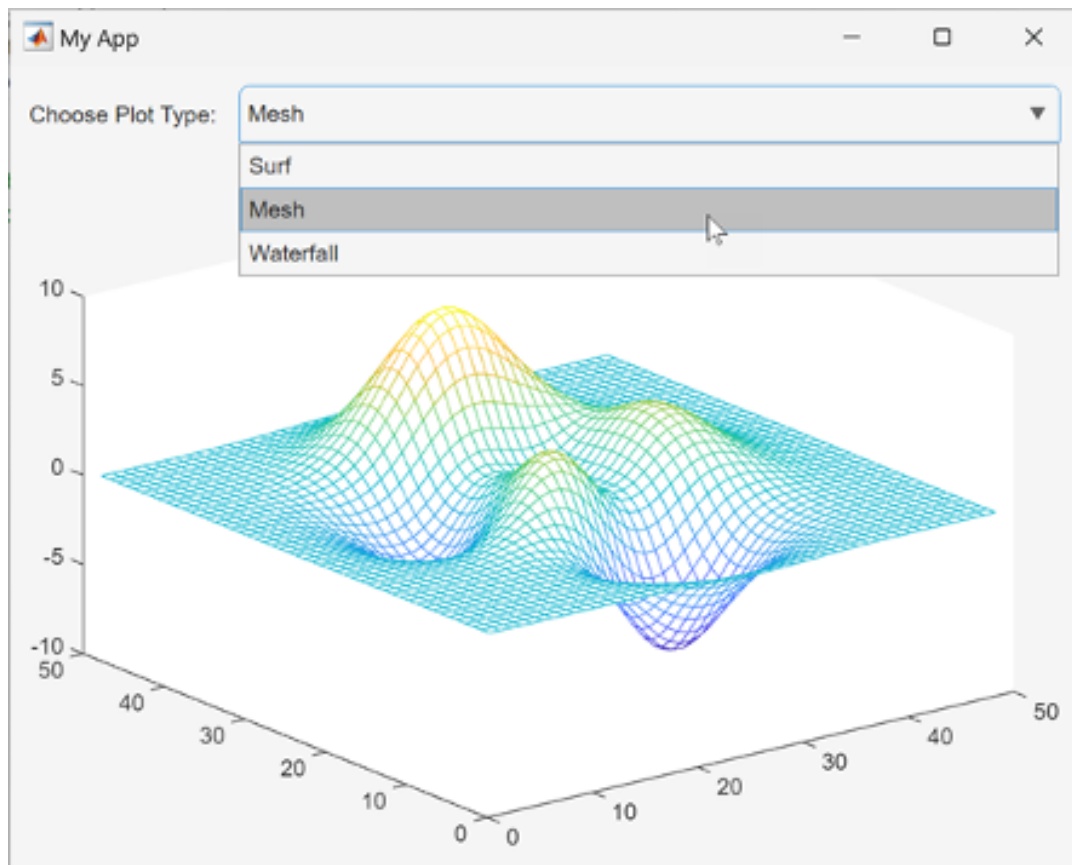


**Zdieľanie a
nasadenie**

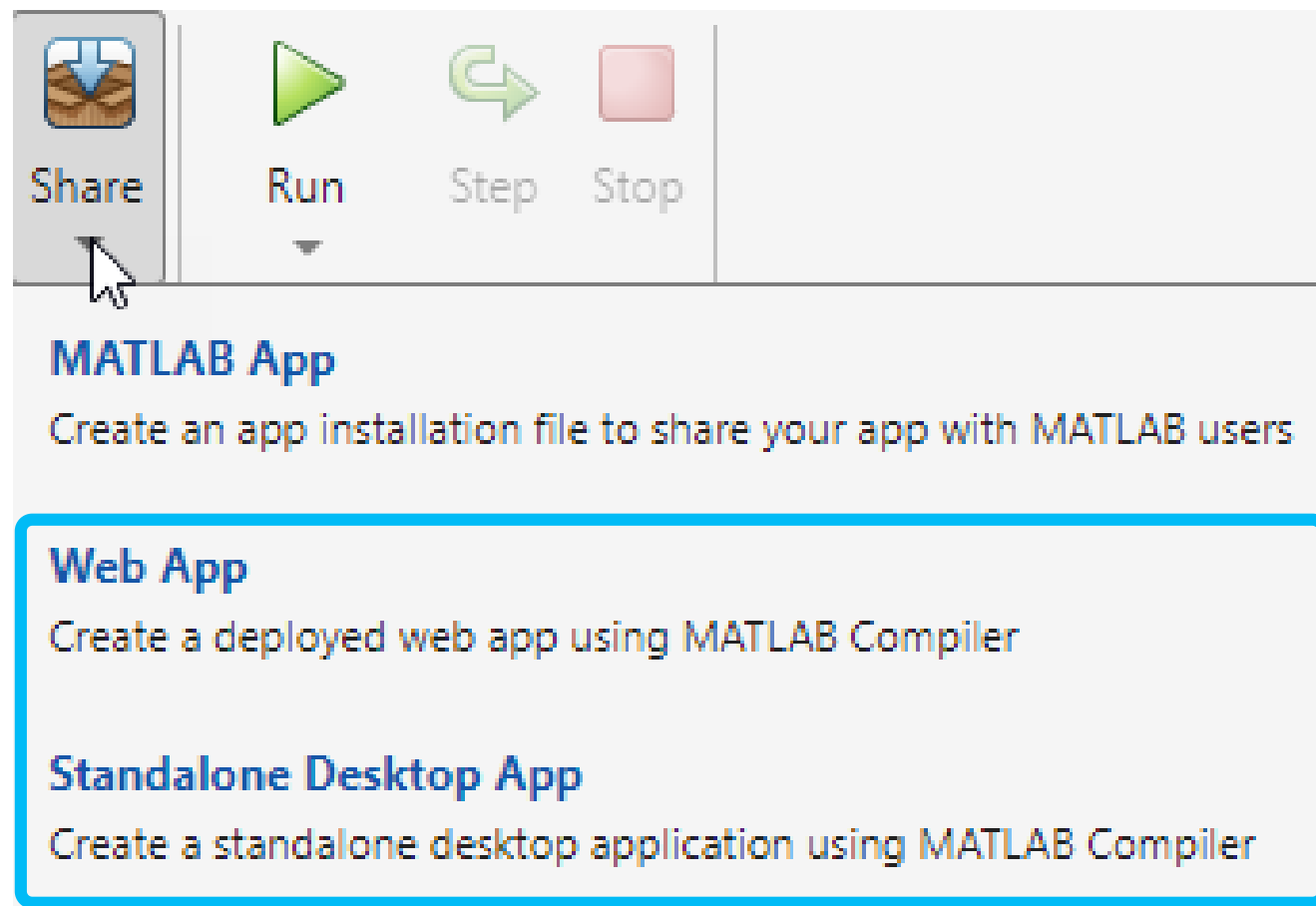
App Designer na tvorbu interaktívnych aplikácií



Tvorba aplikácií so svetlou alebo tmavou témou



Zdieľanie ako desktopové alebo webové aplikácie



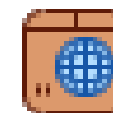
Zjednodušené pracovné postupy nasadzovania aplikácií

- MATLAB Compiler obsahuje špecifické aplikácie
- Aktualizovaný dizajn zjednodušuje pracovné postupy nasadenia
- Integrácia s MATLAB projektmi pre lepšiu organizáciu a správu závislostí

APPLICATION DEPLOYMENT

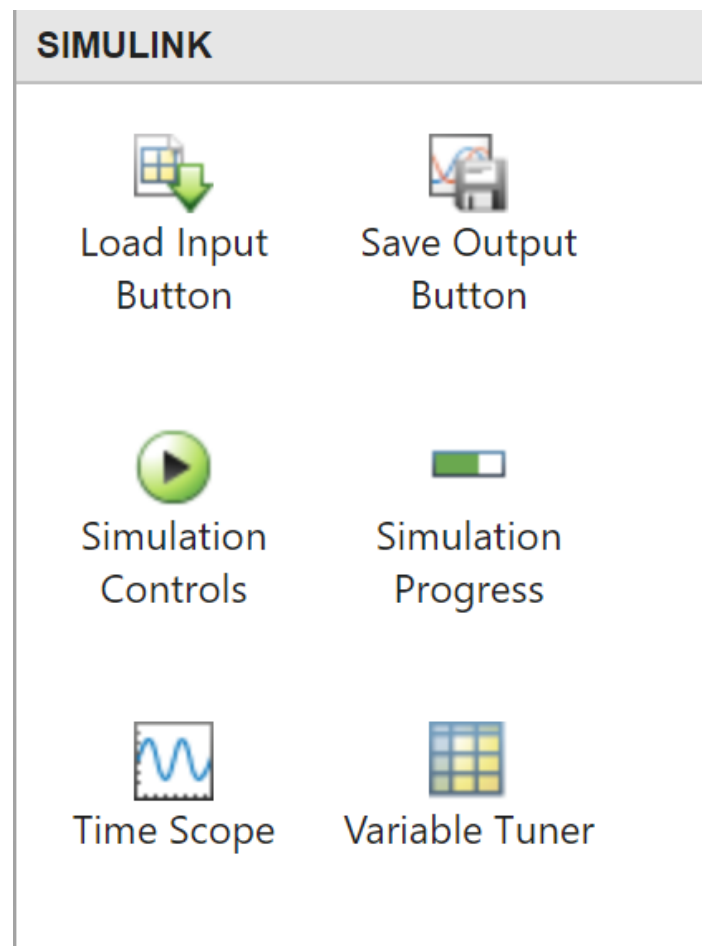


Standalone
Application ...

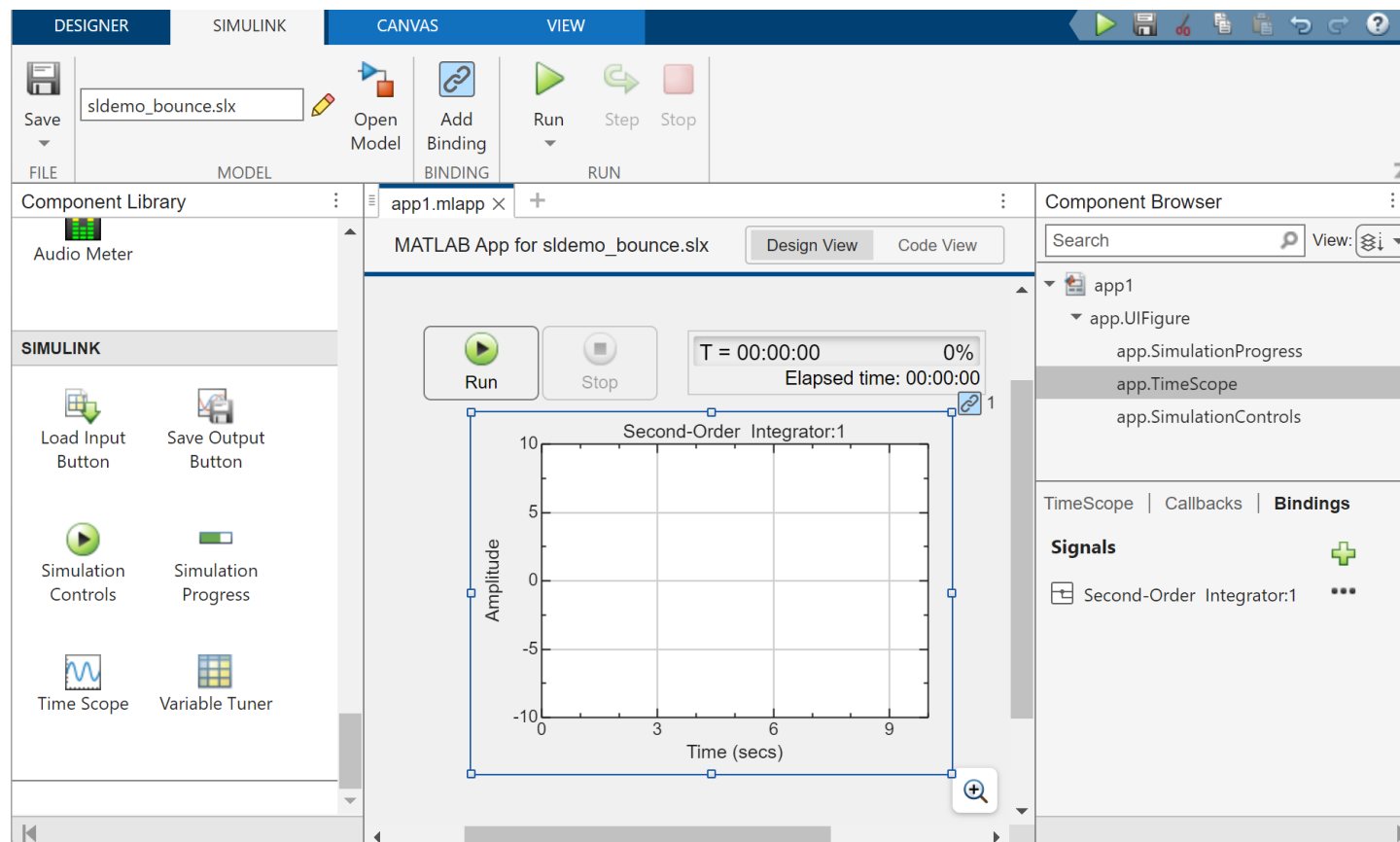


Web App
Compiler

Tvorba aplikácií s komponentami Simulinku

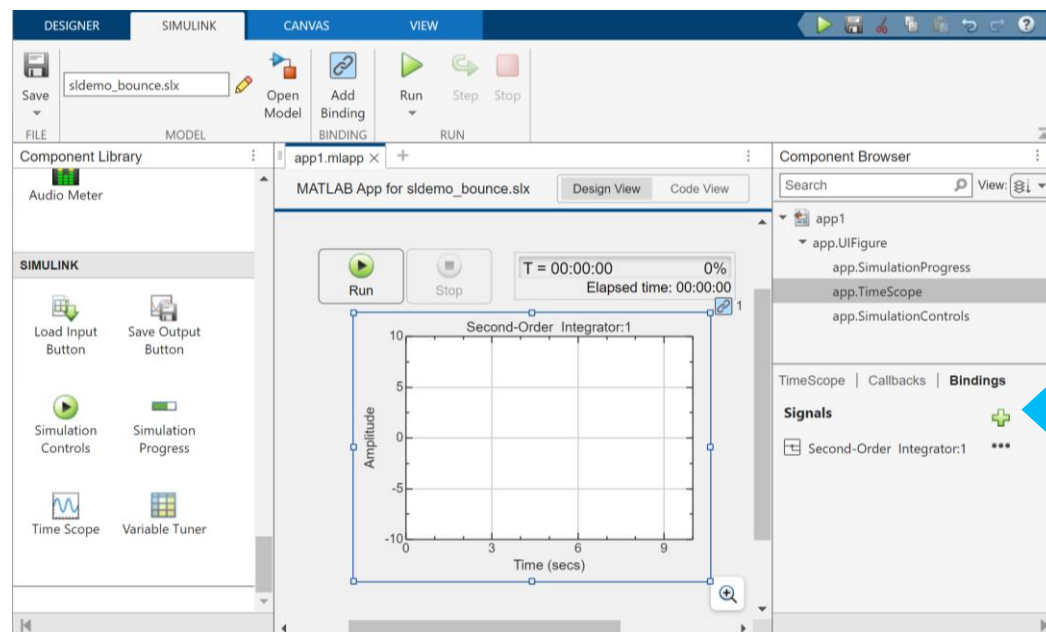


Tvorba aplikácií s komponentami Simulinku

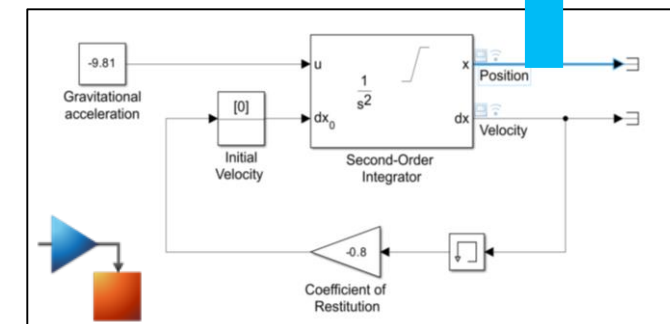


Tvorba aplikácií s komponentami Simulinku

- Prepojte komponenty so signálmi v modeli pomocou zreťazenia

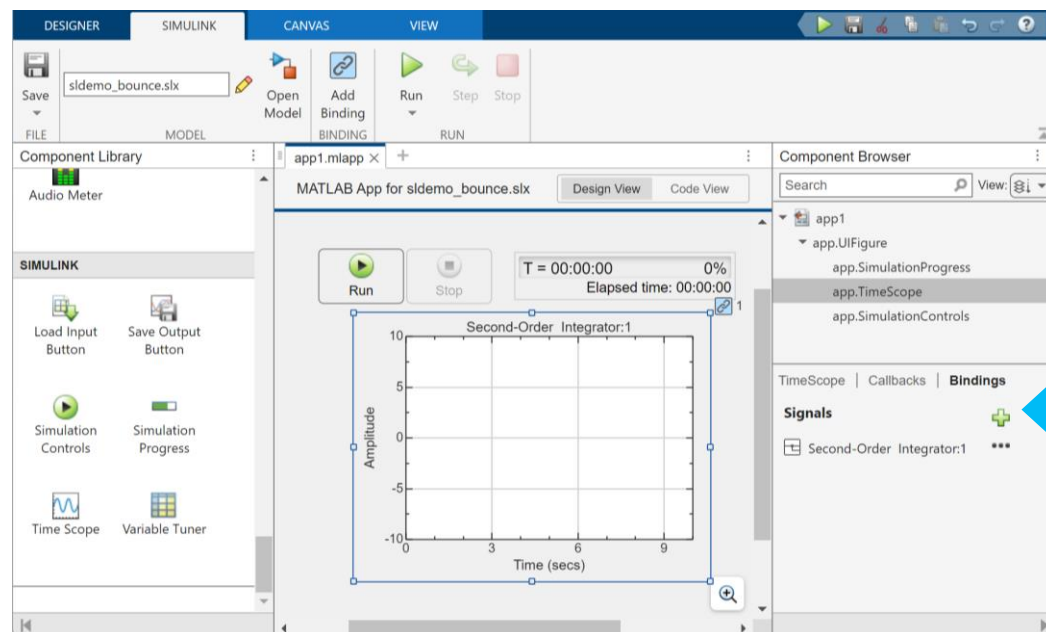


Signal binding

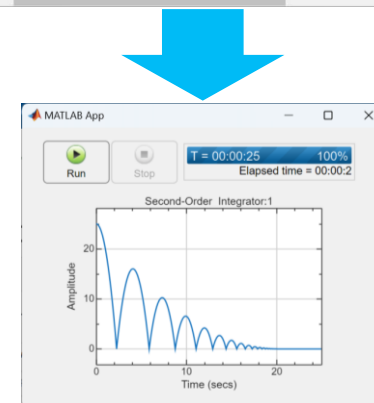


Tvorba aplikácií s komponentami Simulinku

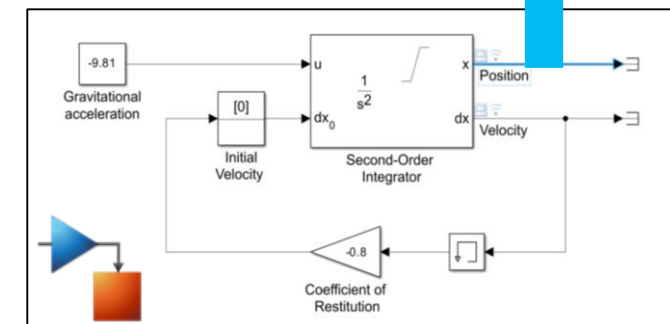
- Prepojte komponenty so signálmi v modeli pomocou zreťazenia
- Generovanie simulačných aplikácií



Signal binding



Simulation app



R2025a Release Highlights – MATLAB and Simulink

Discover What's New

Get more out of MATLAB and Simulink by downloading the latest release.

[Download now](#)



R2025a
Release Highlights

1:37

New Products

- **MATLAB Copilot** – AI assistant optimized for MATLAB.

Release Notes

R2025a

New Products and Major Updates

Antenna Toolbox

Design, analyze, and visualize antenna elements and antenna arrays

HDL Verifier

Find RTL bugs and generate testbenches for ASICs or FPGAs

MATLAB

The Language of Technical Computing

MATLAB Coder

Generate C and C++ code from MATLAB® code

MATLAB Copilot

Create, refine, and debug MATLAB code with generative AI

New in R2025a

Phased Array System Toolbox

Design and simulate phased array and beamforming systems

Radar Toolbox

Design and analyze monostatic, bistatic, and multifunction radar systems

Risk Management Toolbox

Develop risk models and perform risk simulation

RoadRunner

Design 3D scenes for automated driving simulation

RoadRunner Scenario

Create and play back scenarios for automated driving simulation

Sensor Fusion and Tracking Toolbox

Design, simulate, and test multisensor tracking and positioning systems

Explore Release Notes

Select a Product ▼

Resources

[Installation and Licensing Changes \(R2025a\)](#)

[System Requirements](#)

[Bug Reports](#)

[Bug Fixes](#)

MATLAB a Simulink vlastním tempom

Self-Paced Online Courses



Courses and Learning Paths

FILTER BY

Type

Course	76
Onramp (Introductory Course)	24
Learning Path	14

Using MATLAB

All MATLAB	41
Get Started with MATLAB	5
Language Fundamentals	14
Data Import and Analysis	16

Sort by Featured

< Results 1 - 10 of 90 >



New to MATLAB? Start here:

MATLAB Onramp

2 hours | [Languages](#)

Get started quickly with the basics of MATLAB.



New to Simulink? Start here:

Simulink Onramp

2 hours | [Languages](#)

Get started quickly with the basics of Simulink.

<https://matlabacademy.mathworks.com>

Ďakujem za pozornosť!