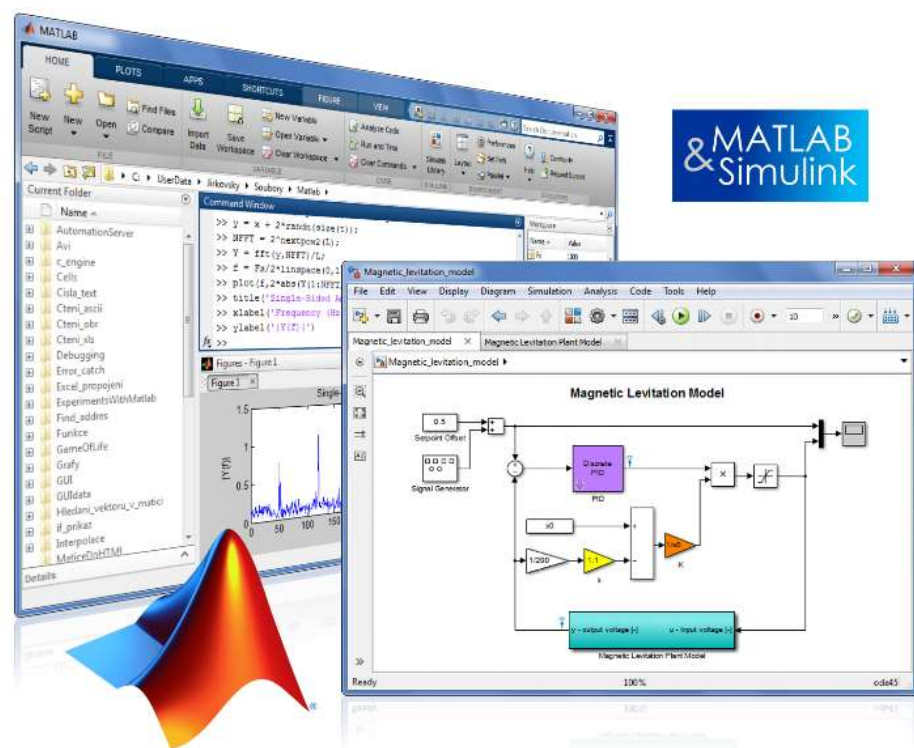


Novinky v prostředí MATLAB v roku 2018



Michal Blaho
blaho@humusoft.sk

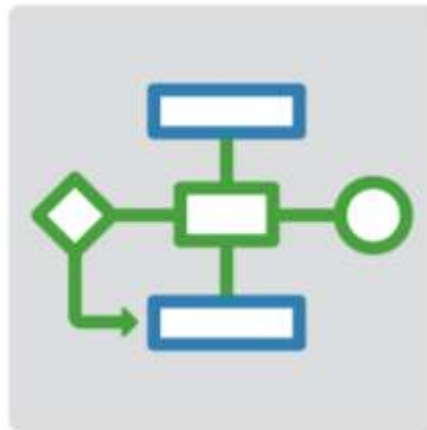
www.humusoft.cz
info@humusoft.cz
www.mathworks.com

Zvýšenie produktivity



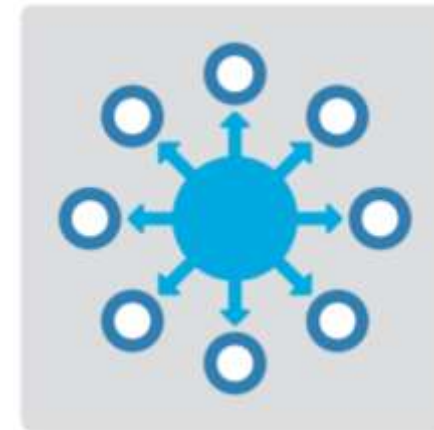
Zrýchlenie práce

Pracovné postupy



Podpora postupov

Tvorba systémov

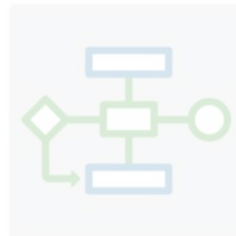


Produkty pre prácu

Zvýšenie produktivity



Pracovné postupy

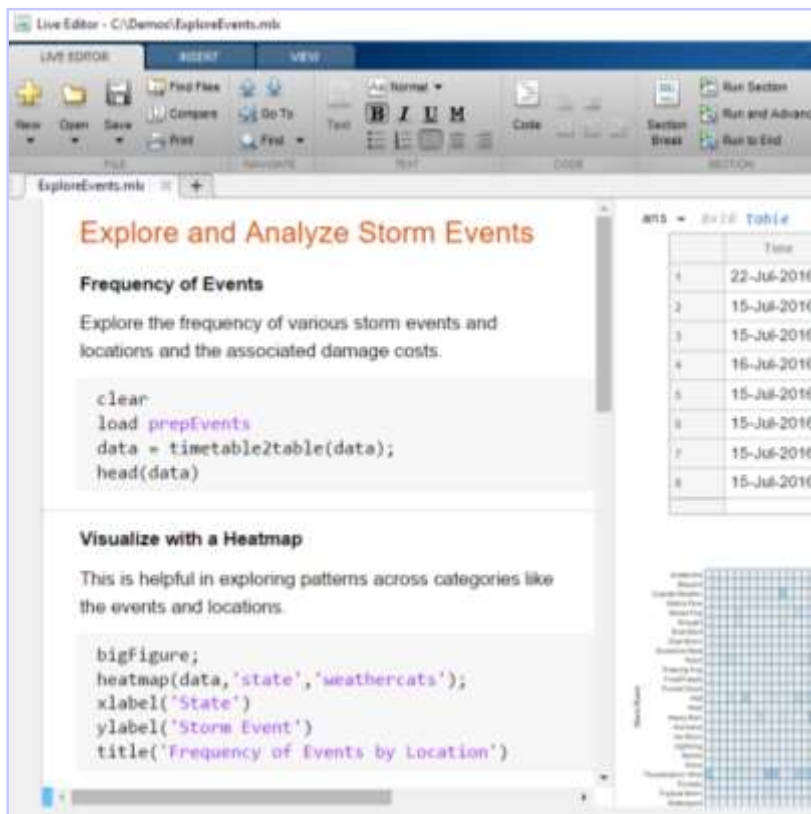


Tvorba systémov



- Rýchlejšia tvorba návrhu
- Jednoduchšia analýza
- Rýchlejšia simulácia a rast pri komplexných úlohách
- Spolupráca

Rýchlejšia tvorba návrhu



The screenshot shows the MATLAB Live Editor interface. The main window displays a script titled "Explore and Analyze Storm Events". The script includes the following code:

```

clear
load prepEvents
data = timetable2table(data);
head(data)

bigFigure;
heatmap(data,'state','weathercats');
xlabel('State')
ylabel('Storm Event')
title('Frequency of Events by Location')

```

The output of the script is shown on the right side of the editor. It includes a table with the following data:

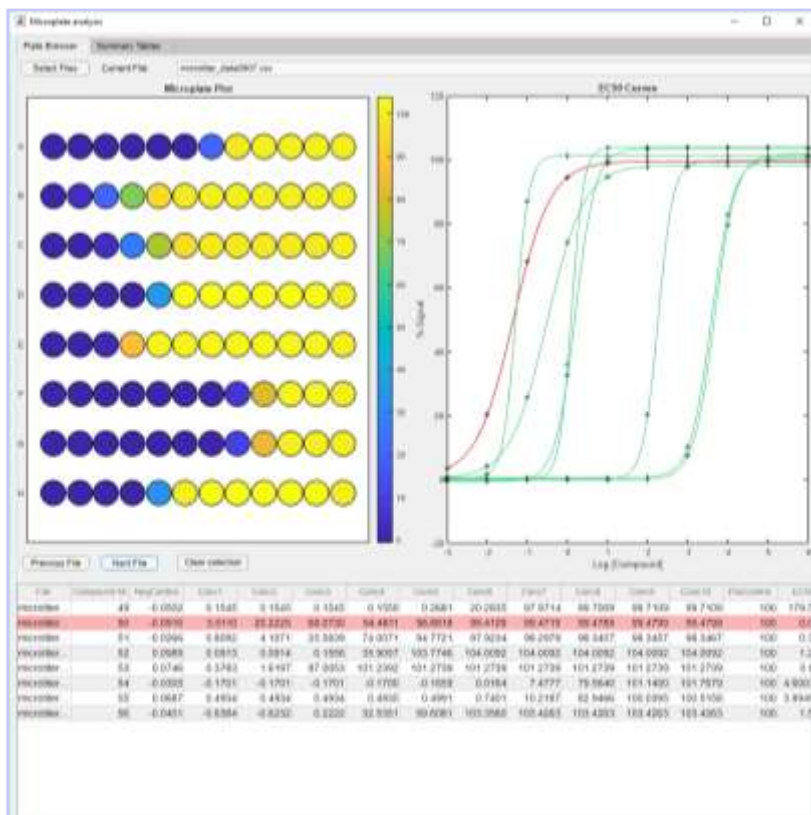
	Time
1	22-Jul-2016
2	15-Jul-2016
3	15-Jul-2016
4	16-Jul-2016
5	15-Jul-2016
6	15-Jul-2016
7	15-Jul-2016
8	15-Jul-2016

Below the table, a heatmap visualization is displayed, showing the frequency of events across different states and weather categories. The x-axis is labeled "State" and the y-axis is labeled "Storm Event". The title of the heatmap is "Frequency of Events by Location".

MATLAB

Live Editor

Rýchlejšia tvorba návrhu



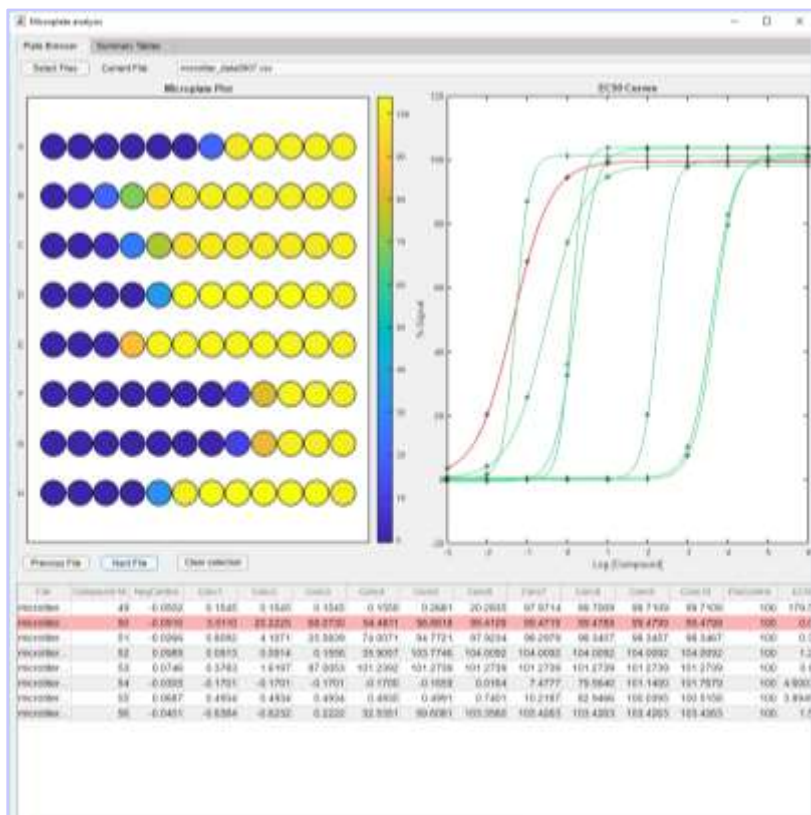
MATLAB

App Designer

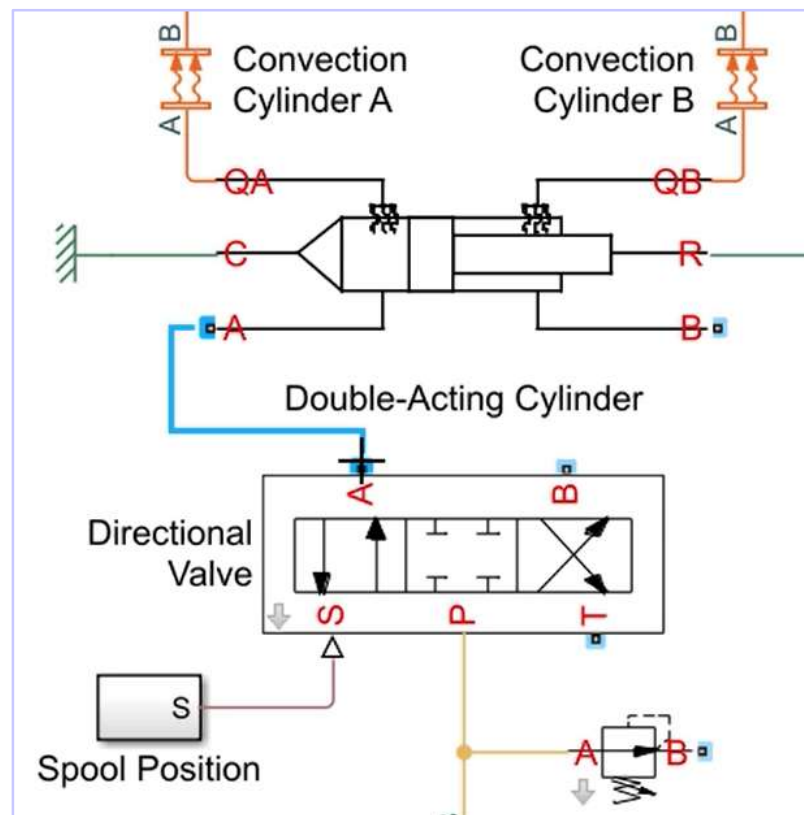


The screenshot shows the MathWorks File Exchange interface. The main heading is 'GUIDE to App Designer Migration Tool for MATLAB'. Below the heading, it states 'version 1.0 (15.1 KB) by MathWorks App Designer Team'. The description reads: 'Use the GUIDE to App Designer Migration tool to help transition your GUIDE apps to App Designer.'

Rýchlejšia tvorba návrhu

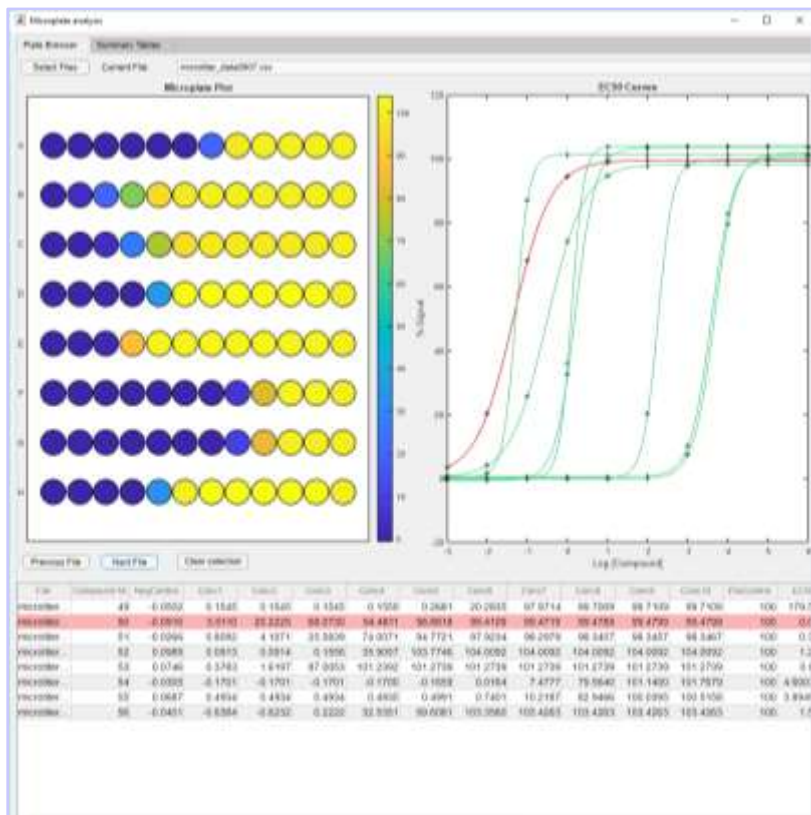


MATLAB

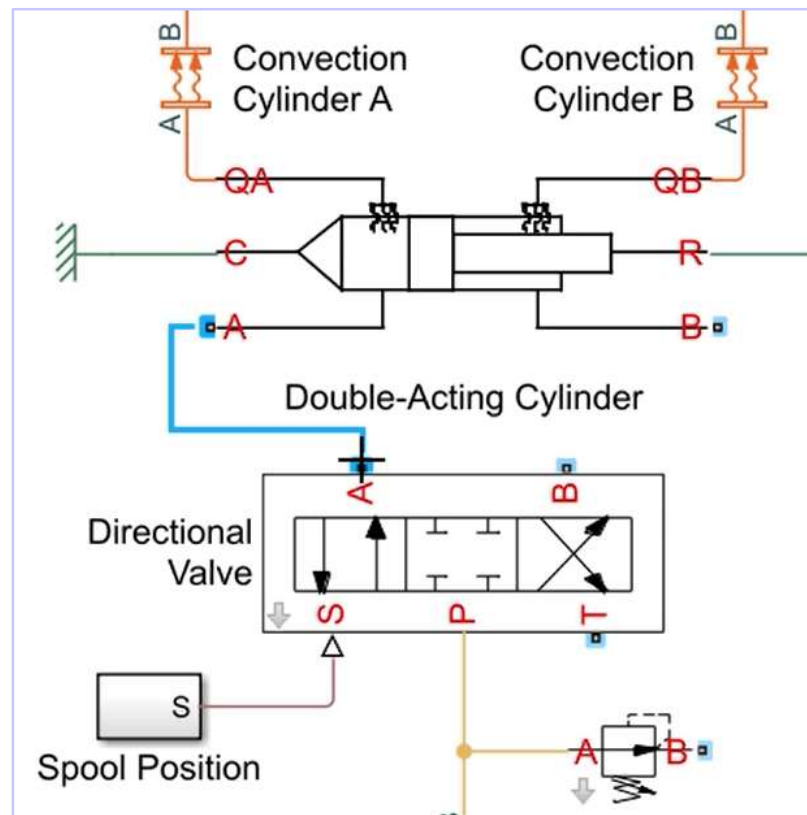


Simulink

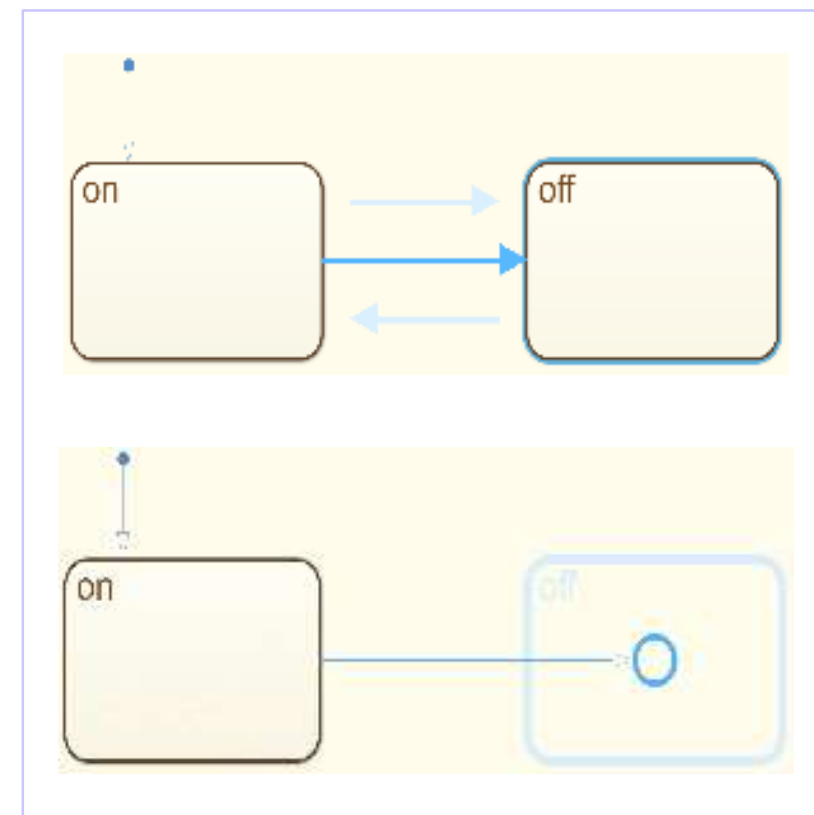
Rýchlejšia tvorba návrhu



MATLAB



Simulink

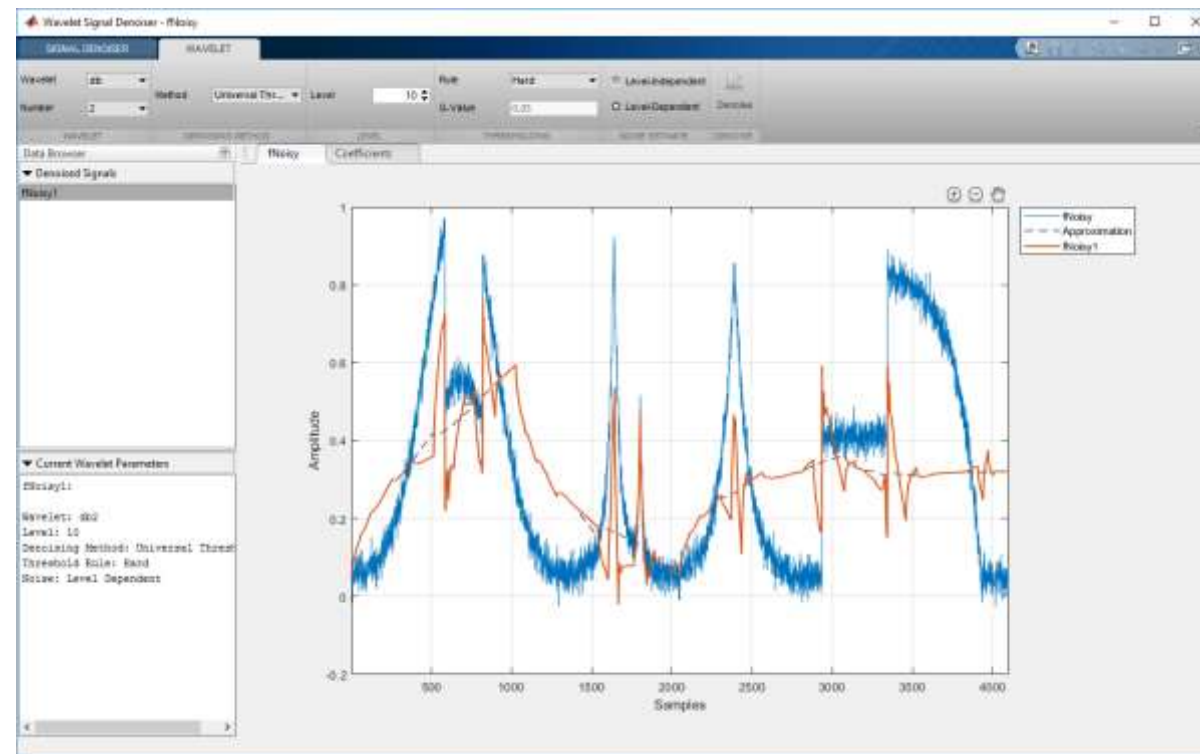


Stateflow

Zjednodušenie analýzy s aplikáciami

Interaktívne aplikácie automatizujú bežné výpočtové úlohy

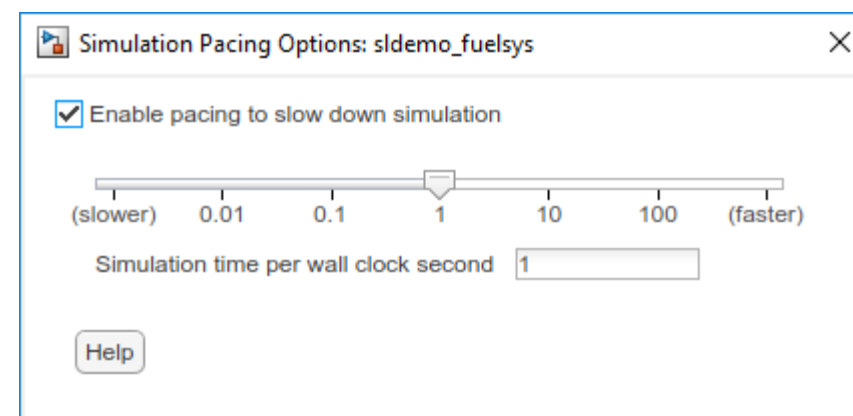
- Econometric Modeler app
Analýza časových postupností, testovanie špecifikácií, modelovanie diagnostika
- Analog Input Recorder app
Záznam a vizualizácia analógových vstupných signálov
- Wavelet Signal Denoiser app
Vizualizácia a odstránenie šumu



Zjednodušenie analýzy spomalením simulácie

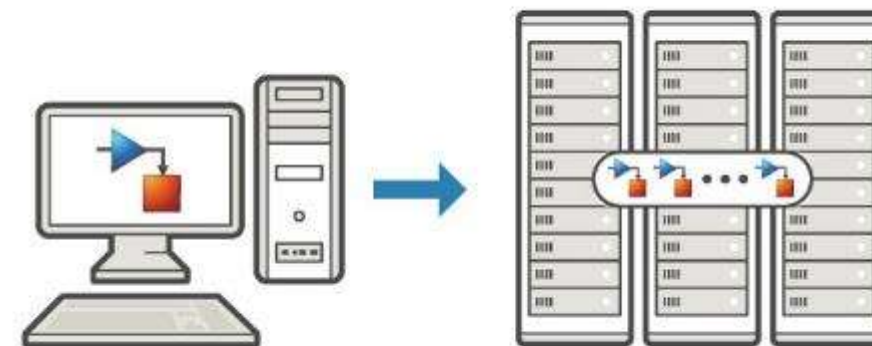
Spomalenie simulácie pre jednoduchšiu interakciu s modelom

- Výhodné v modeloch riadených a monitorovaných s blokmi Dashboard, Scope alebo Display
- Užitočné ak je model pripojený k hardvéru

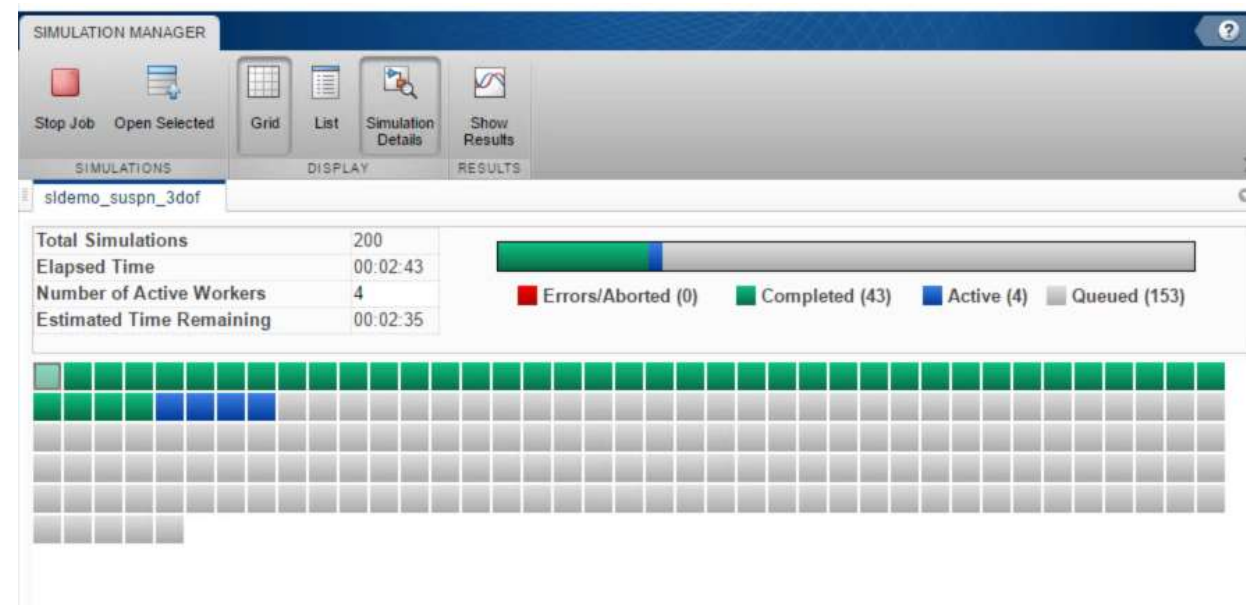


Komplexné riešenia

Využite paralelné výpočty na zrýchlenie behu viacerých simulácií



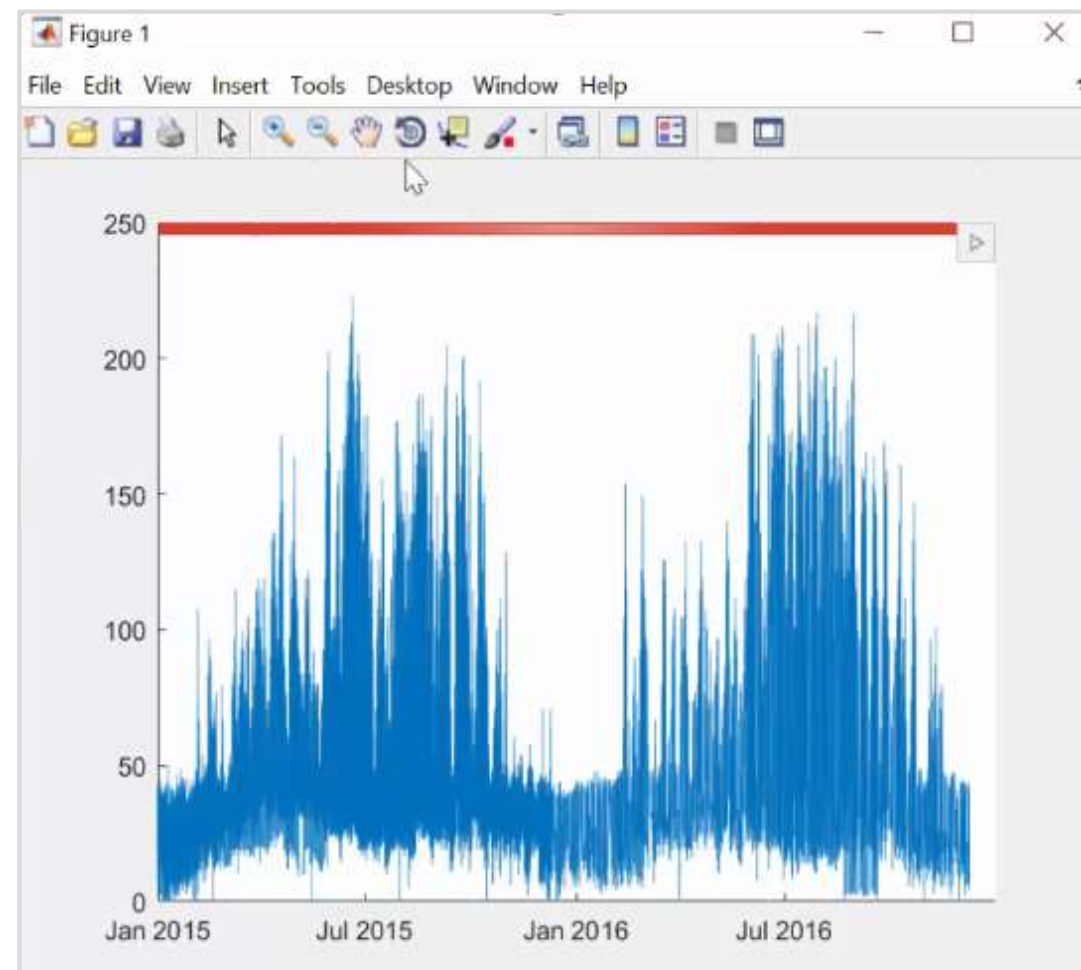
- Beh viacerých paraelných simulácií pomocou **parsim**
- Monitorovanie simulačného stavu a priebehu pomocou nástroja Simulation Manager



Komplexné riešenia

Využite tall arrays na manipuláciu a analýzu dát, ktoré sú príliš veľké pre pamäť

- Známe funkcie a syntax MATLABu
- Podpora stoviek funkcií
- Riešenia pre kláster Spark + Hadoop

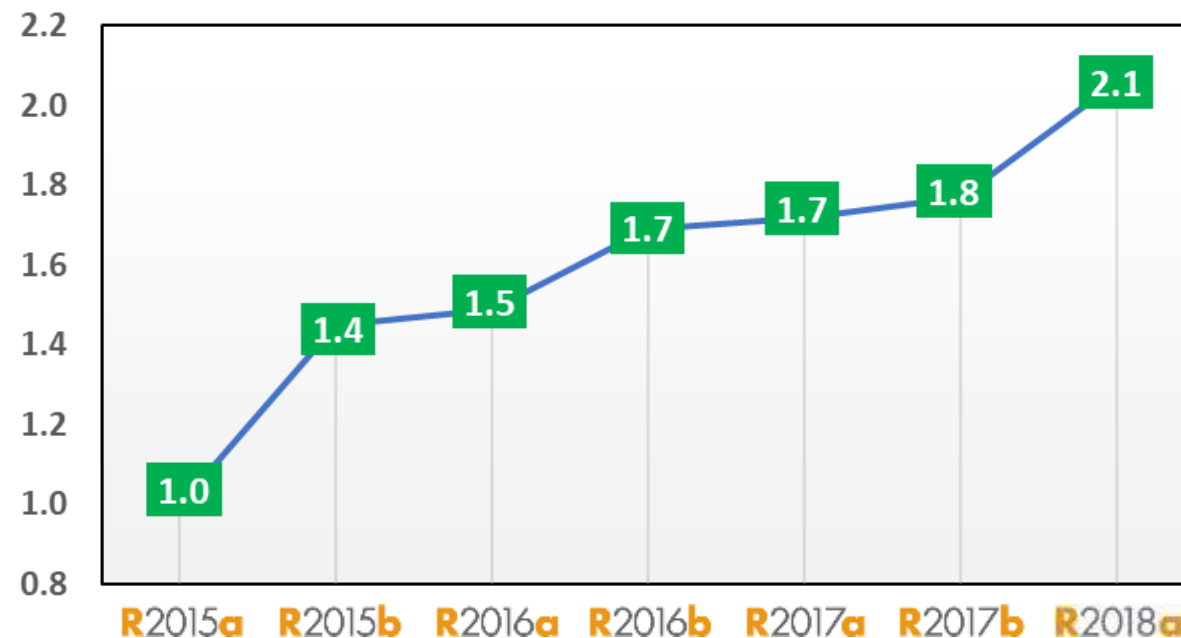


Rýchlejšia simulácia

Prerobené výpočtové jadro pre rýchlejší beh programu

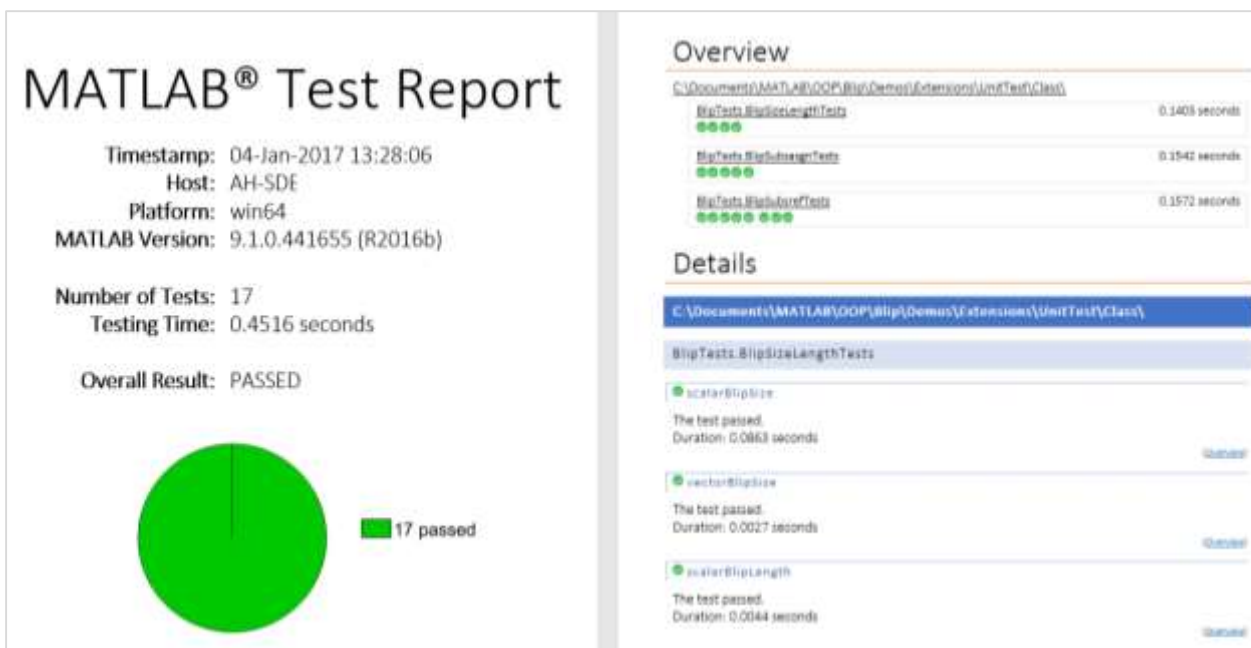
- Celý program MATLABu môže byť kompilovaný JIT
- MATLAB zbehne váš program dvakrát rýchlejšie ako pred troma rokmi
- Nemusíte meniť ani riadok kódu
- Zrýchlenie spustenia MATLABu v R2018a

Average Speedup in Customer Workflows



Tímová spolupráca

Pokročilé nástroje na vývoj softvéru a manažment, testovanie a integrácia kódu



Tímová spolupráca

Pokročilé nástroje na vývoj softvéru a manažment, testovanie a integrácia kódu

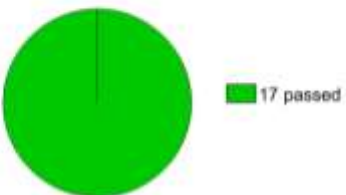
Zistenie rozdielov medzi časťami modelu, Stateflow a blokov MATLAB Function

MATLAB® Test Report

Timestamp: 04-Jan-2017 13:28:06
 Host: AH-SDE
 Platform: win64
 MATLAB Version: 9.1.0.441655 (R2016b)

Number of Tests: 17
 Testing Time: 0.4516 seconds

Overall Result: PASSED



Overview

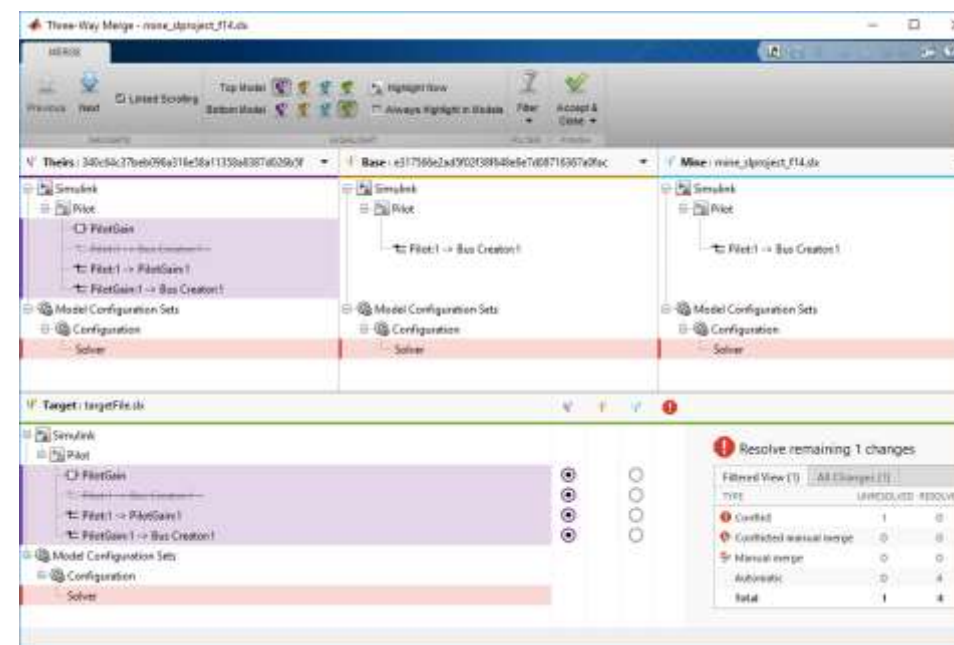
C:\Documents\MATLAB\OOP\Blip\Demus\Extensions\UnitTest\Class	0.3403 seconds
BlipTests.BlipSizeLengthTests	0.3542 seconds
BlipTests.BlipStartTests	0.1572 seconds

Details

C:\Documents\MATLAB\OOP\Blip\Demus\Extensions\UnitTest\Class

BlipTests.BlipSizeLengthTests

- scalarBlipSize: The test passed. Duration: 0.0063 seconds
- vectorBlipSize: The test passed. Duration: 0.0027 seconds
- vectorBlipLength: The test passed. Duration: 0.0044 seconds



The screenshot shows the 'Three-Way Merge' interface in Simulink. It compares three models: 'This', 'Base', and 'Mine'. The 'This' model is highlighted in purple, 'Base' in yellow, and 'Mine' in light blue. The interface shows a hierarchical tree of blocks including Simulink, Plot, FilterGain, and Solver. A 'Resolve remaining 1 changes' dialog is visible in the bottom right corner, showing a table of changes:

TYPE	UNRESOLVED	RESOLVED
Conflict	1	0
Conflicted manual merge	0	0
Manual merge	0	0
Automatic	0	4
Total	1	4

Zvýšenie produktivity



Pracovné postupy



Tvorba systémov

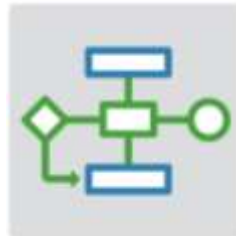


- Rýchlejšia tvorba návrhu
- Jednoduchšia analýza
- Rýchlejšia simulácia a rast pri komplexných úlohách
- Spolupráca

Zvýšenie
produktivity



Pracovné
postupy



Tvorba
systémov



- **Nasadenie algoritmov a aplikácií MATLABu**
- **Generovanie kódu z modelov Simulinku**
- **Verifikácia a validácia**

Nasadenie algoritmov a aplikácií MATLABu

Prístup k dátam

 **Senzory** **Súbory** **Databázy**

Analýza dát

 **Skúmanie dát** **Spracovanie** **Algoritmy**

Vývoj

 **Modely AI** **Vývoj algoritmov** **Modelovanie a simulácia**

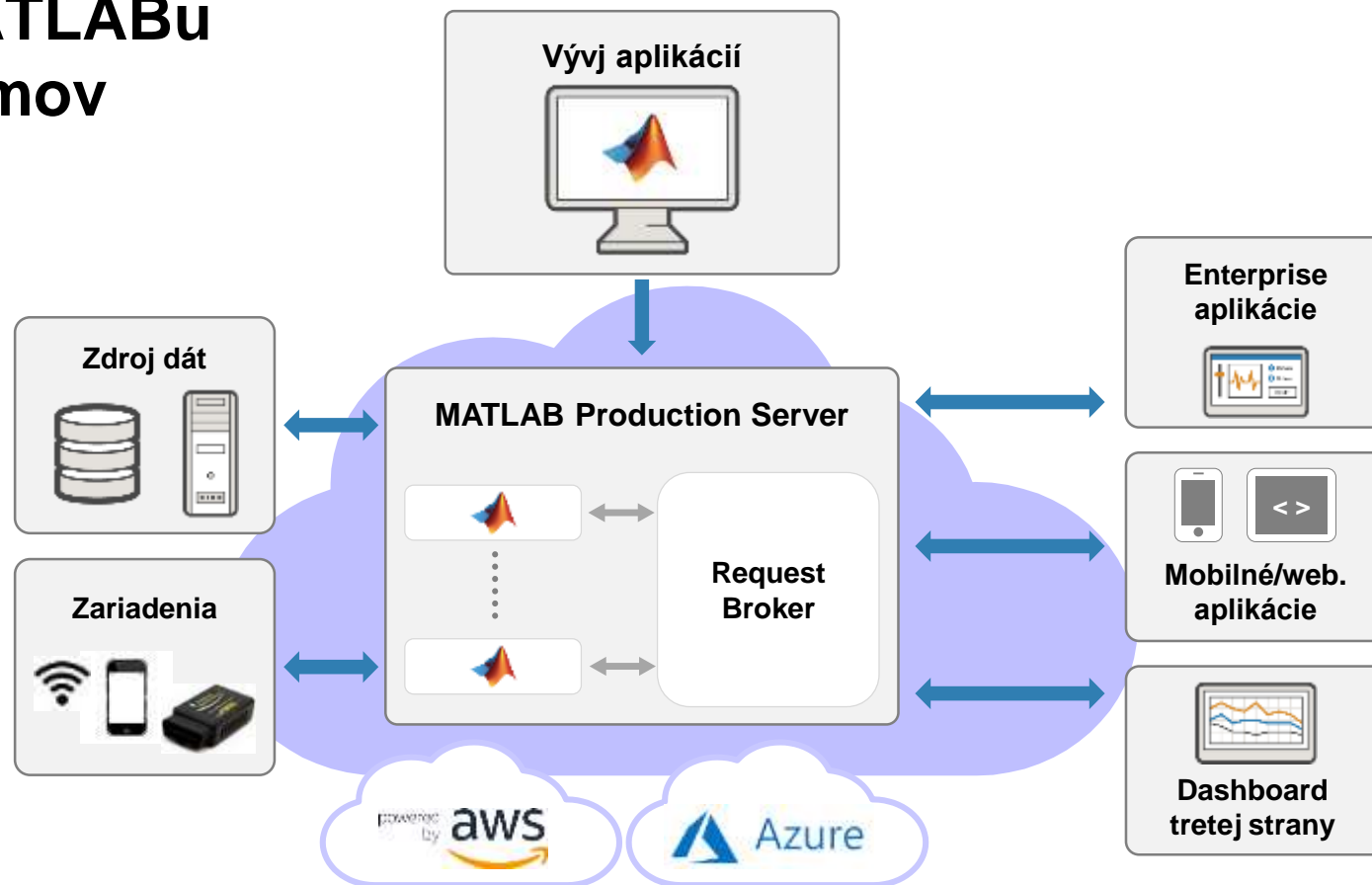
Nasadenie

 **Aplikácie** **Enterprise systémy** **Embedded zariadenia**

Nasadenie algoritmov a aplikácií MATLABu

Zdieľajte vašu prácu mimo MATLABu bez preprogramovania algoritmov

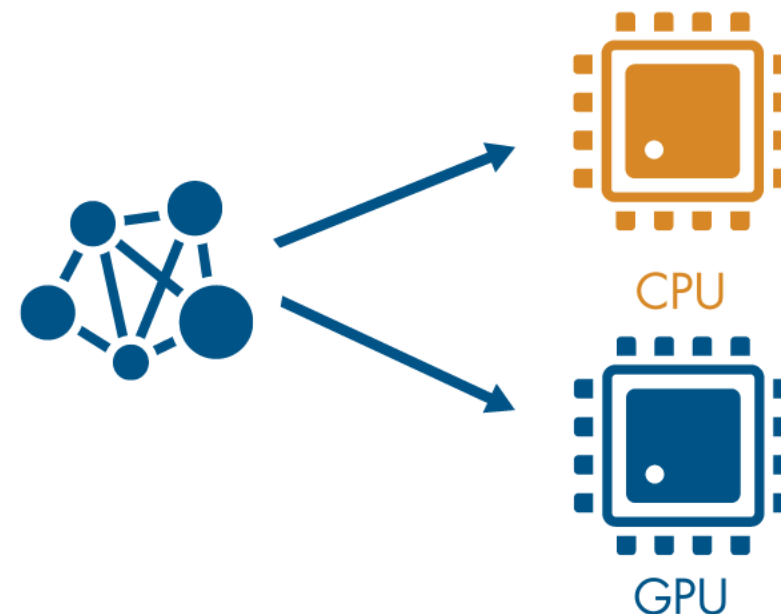
- Samostatné desktopové aplikácie
- Doplnky pre Microsoft Excel
- Softvérové komponenty pre integráciu s inými jazykmi (*C/C++*, *.NET*, *Python*, *Java*)
- Softvérové komponenty pre web a enterprise aplikácie



Nasadenie algoritmov

Nasadenie modelov strojového učenia a deep learningu s využitím automaticky generovaného kódu

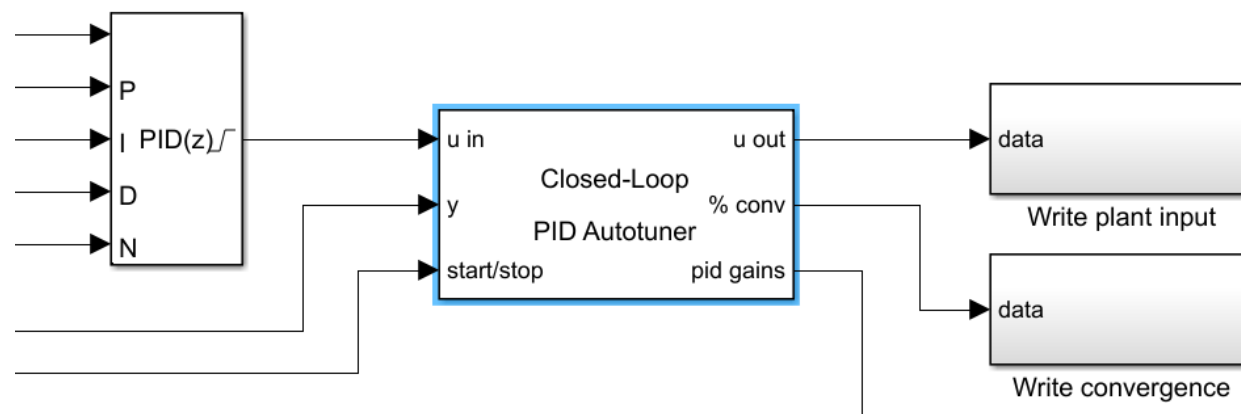
- Generovanie C kódu pre prediktívne strojové učenie a modely deep learningu
- Generovanie optimalizovaného CUDA kódu pre deep learning, počítačové videnie a autonómne systémy



Ladenie PID regulátorov

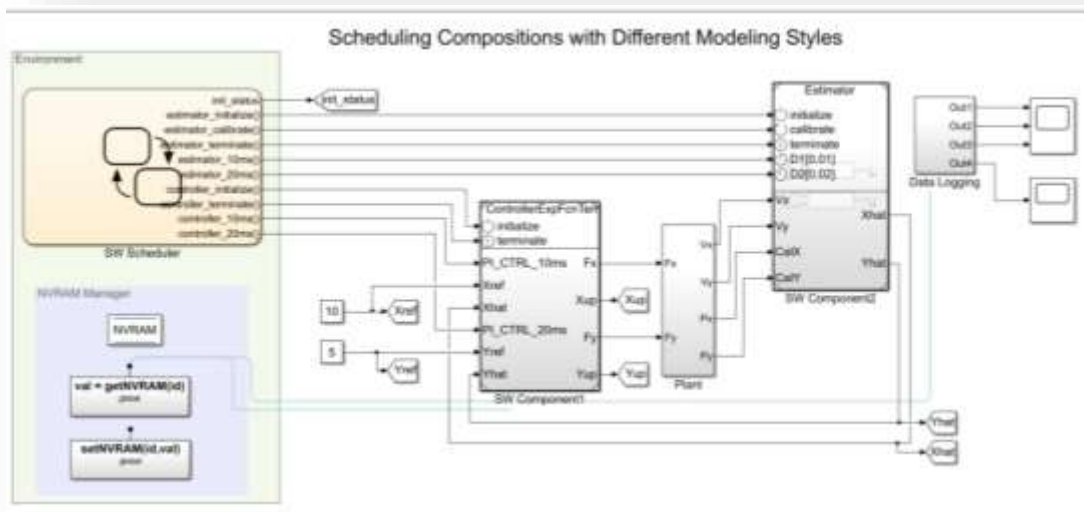
Implementácia samonastavujúceho sa PID algoritmu

- Automatické ladenie zosilnení PID regulátora v reálnom čase so sústavou
- Nepotrebujeme model dynamiky systému
- Nasadenie ladiaceho algoritmu na embedded zariadenie s využitím automaticky generovaného kódu



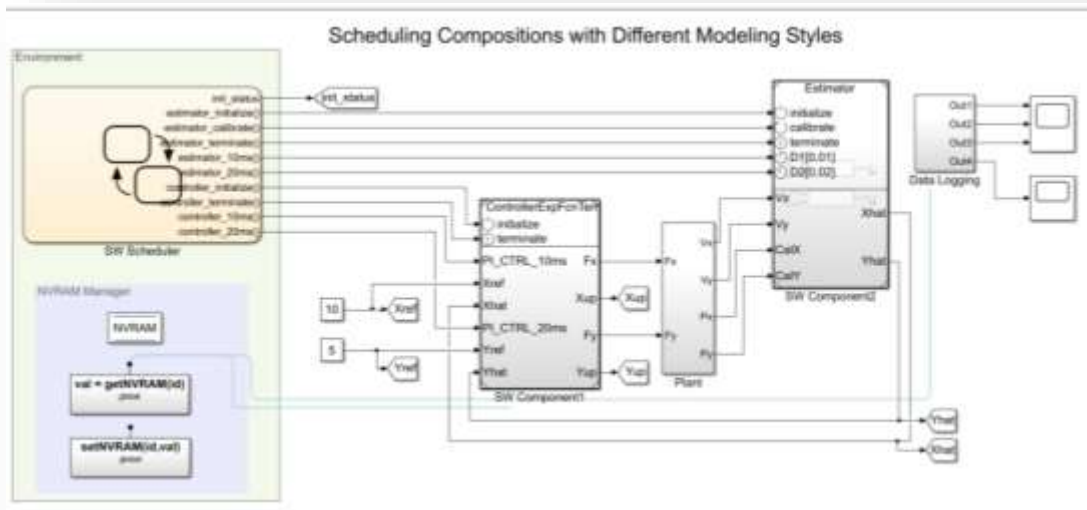
Príprava modelu na generovanie kódu

Príprava komponentov modelu na generovanie kódu



Príprava modelu na generovanie kódu

Príprava komponentov modelu na generovanie kódu

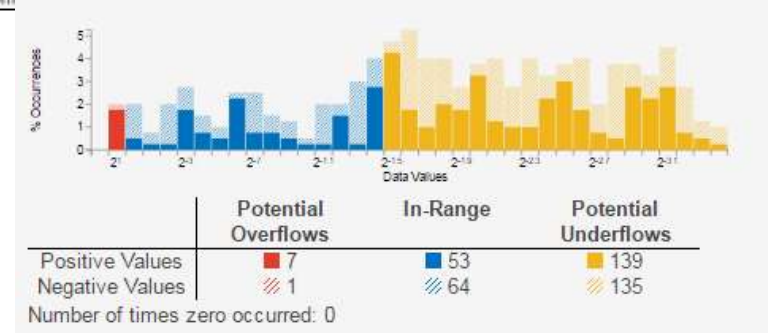


Príprava dát modelu na generovanie kódu

The screenshot shows the FIXED-POINT TOOL interface. The 'Simulation Ranges' and 'Derived Ranges' buttons are highlighted with blue boxes. A green arrow points from the 'Simulation Ranges' button to the 'Propose Data Types' button, which is also highlighted with a blue box. Another green arrow points from the 'Propose Data Types' button to the 'Compare Results' button, which is also highlighted with a blue box. The 'MODEL HIERARCHY' pane shows a tree structure with 'Simulink Root' at the top, followed by 'Data Objects', 'mpSRP_sub_codegen_M', 'Subsystem1', 'Subsystem2', 'Subsystem', and 'Subsystem'. The 'Results' table below shows the following data:

Name	Run	CompiledDT	SpecifiedDT	ProposedDT	Accept Sim
SRP Subsystem1	Ranges(Double)	double	fixdt(1,13,11)	locked	0
SRP Subsystem2	Ranges(Double)	double	fixdt(1,13,11)	locked	0
SRP Subsystem3	Ranges(Double)	double	fixdt(1,17,15)	fixdt(0,17,22)	0

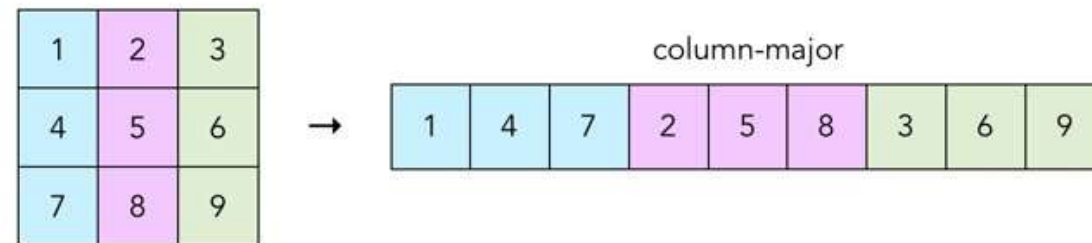
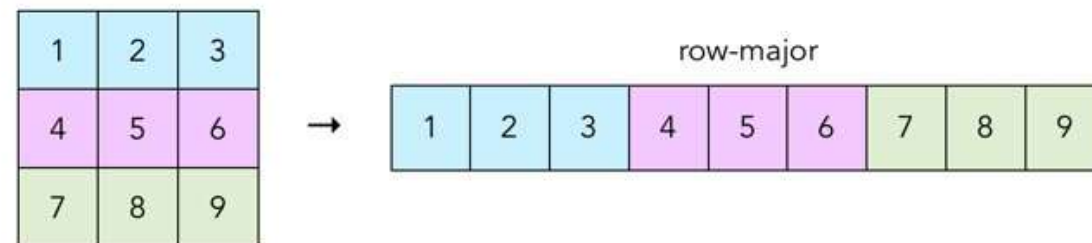
Visualization of Simulation Data



Generovanie kódu z modelov Simulinku

Prístup a definovanie všetkých informácií v modely týkajúcich sa generovanie kódu

- Náhľad a definovanie dát na jednom mieste
- Náhľad implementačných detailov bez detailov modelu
- Zlepšenie výkonu a implementácie s iným C kódom



**Možnosť rozloženia v pamäti
row-major**

Prepojenie návrhu s hardvérom

Priame prepojenie s hardvérom pomocou podporných balíčkov

- Streamovanie z a do hardvéru
- Spustenie modelov Simulinku na lacnom HW ako je Arduino, Raspberry Pi alebo LEGO
- Automatické generovanie kódu a spustenie na mikrokontroléroch FPGA a mnoho ďalších



Arduino



ARM Cortex



Raspberry Pi



Microsemi FPGA

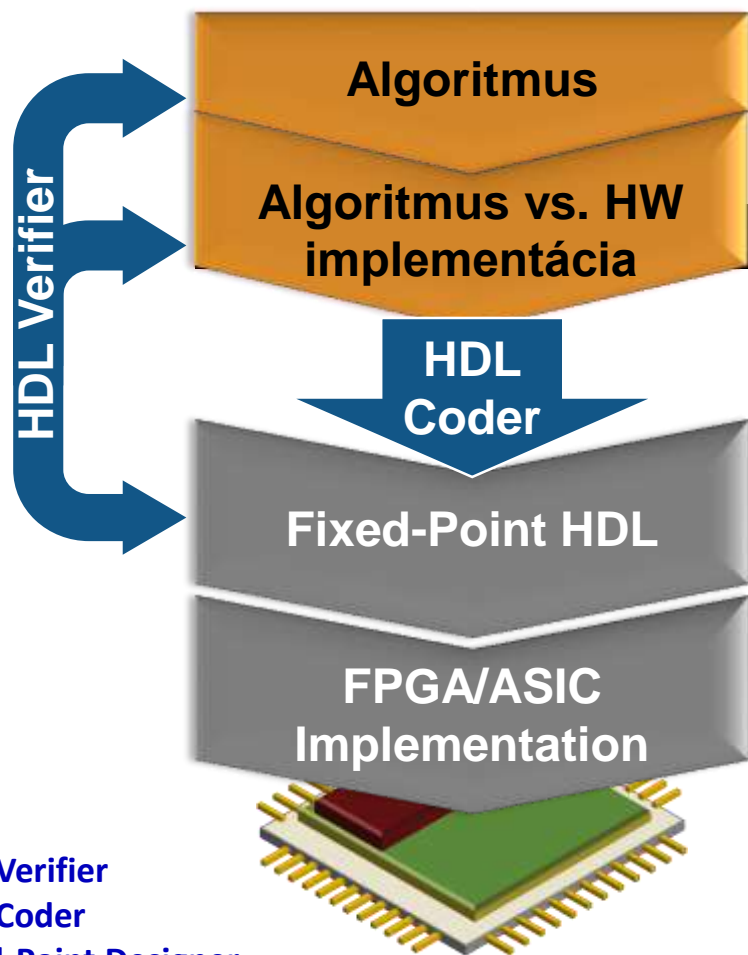


LEGO

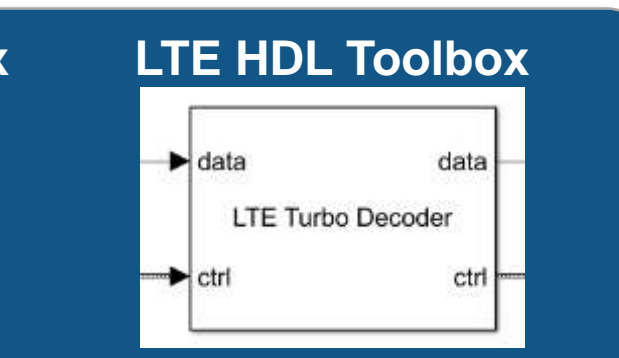
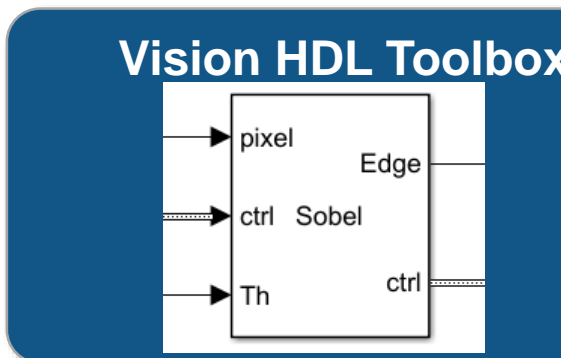
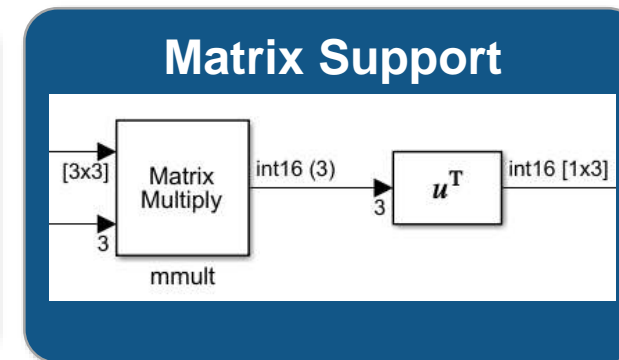
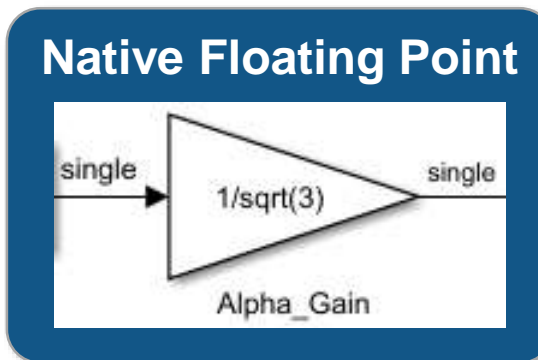


ADALM-PLUTO

Nasadenie na FPGA a ASIC



HDL Verifier
HDL Coder
Fixed-Point Designer
Vision HDL Toolbox
LTE HDL Toolbox



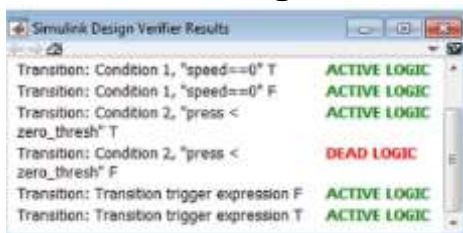
Verifikácia a validácia

Kompletná sada produktov

Simulink Requirements R2017b



Simulink Design Verifier



Simulink Check R2017b

- Modeling Standards for Secure Coding (CERT C, CWE, ISO/IEC TS 17961)
 - Check configuration parameters for secure coding standards
 - Check for blocks not recommended for C/C++ production code deployment
 - Check for blocks not recommended for secure coding standards
 - Check usage of Assignment blocks
 - Check for switch case expressions without a default case
 - Check for bitwise operations on signed integers
 - Check for equality and inequality operations on floating-point values
 - Check integer word lengths

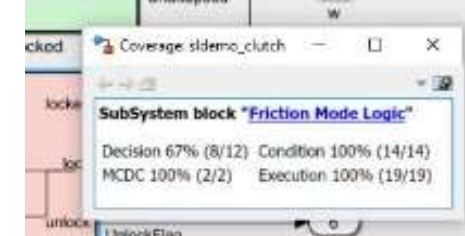
Polyspace

```

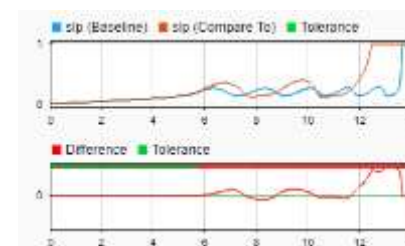
25  INTEGER_DIVIS
30  -----
31  int intdiv(int p)
32  {
33      int i;
34      int j = 1;
35
36      i = 1024 / (j - p);
37      return i;
    
```

Probable cause for 'Integer division by zero':
 intdiv(1);
 operator / on type int 32
 left: 1024
 right: 0
 result: [-1024 .. 1024]

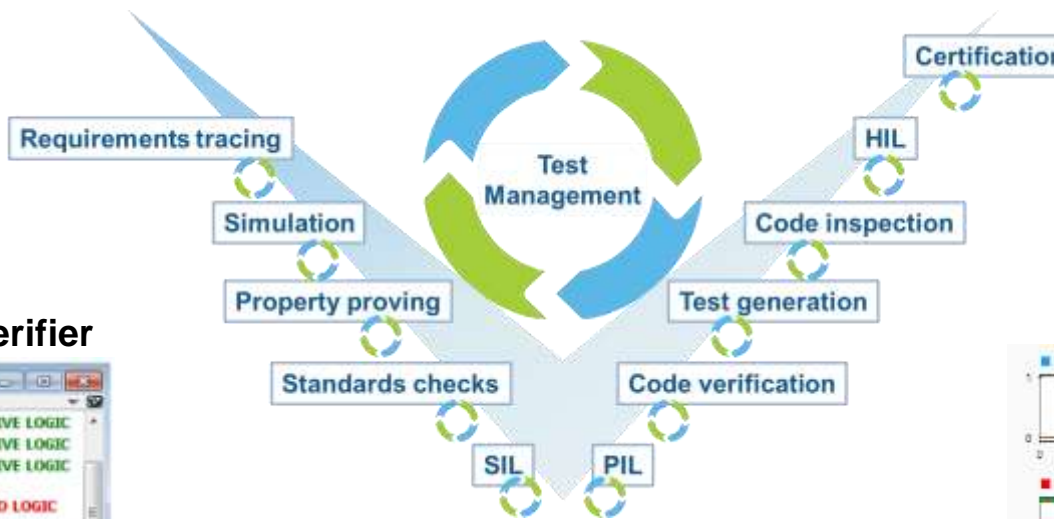
Simulink Coverage R2017b



Simulink Test



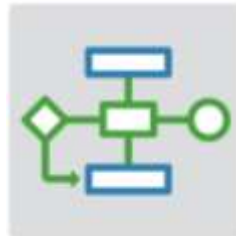
podpora
AUTOSAR
 R2018a



Zvýšenie
produktivity



Pracovné
postupy



Tvorba
systémov



- **Nasadenie algoritmov a aplikácií MATLABu**
- **Generovanie kódu z modelov Simulinku**
- **Verifikácia a validácia**

Zvýšenie produktivity



Pracovné postupy

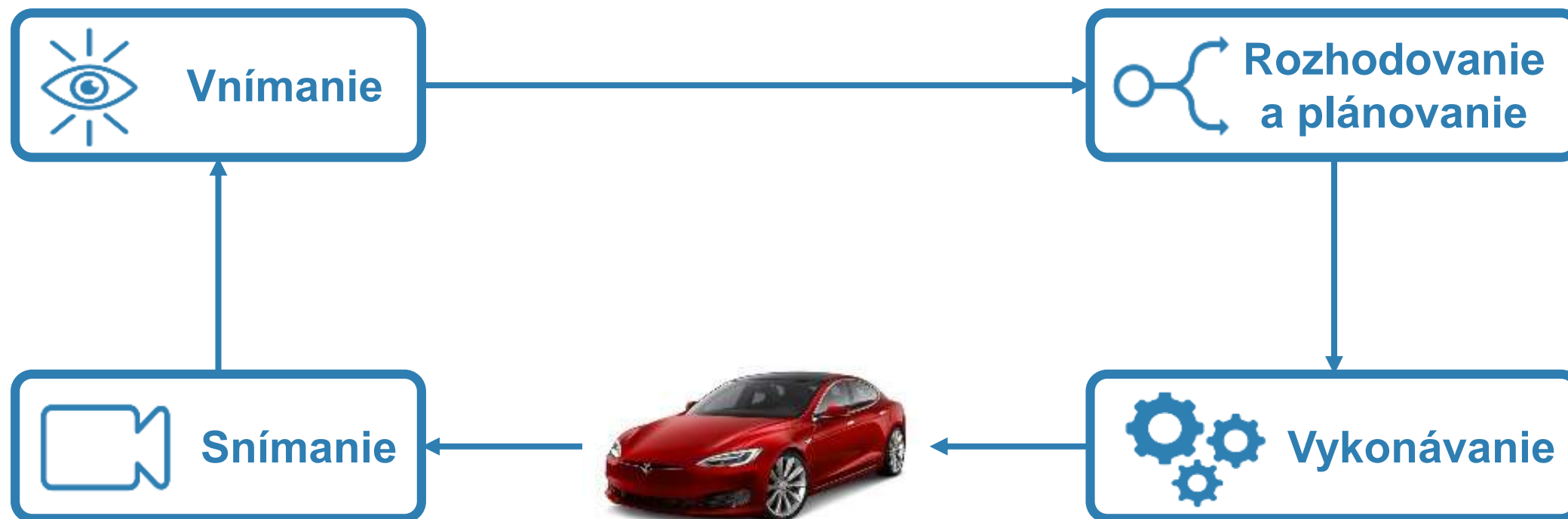


Tvorba systémov



- **Autonómne systémy**
- **Bezdrôtové komunikácie**
- **Umelá inteligencia (AI)**

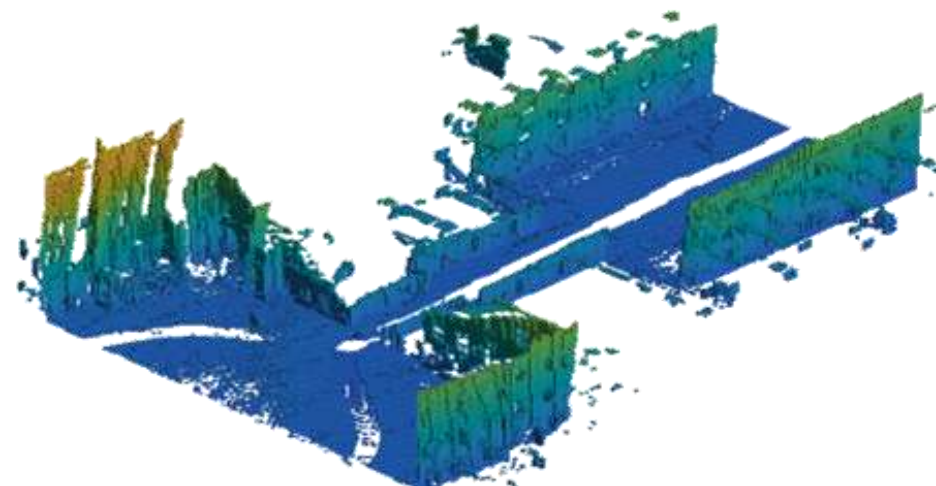
Návrh autonómnych systémov



Návrh autonómnych systémov

Mapovanie prostredia s využitím senzorových dát

- Segmentácia a registrácia lidarových point cloudov
- SLAM pomocou lidar: Lokalizácia robotov a tvorba máp prostredia



Návrh autonómnych systémov

Pochopenie prostredia pomocou techník počítačového videnia a deep learningu

- Detekcia objektov a ich sledovanie
- Sémantická segmentácia pomocou deep learningu

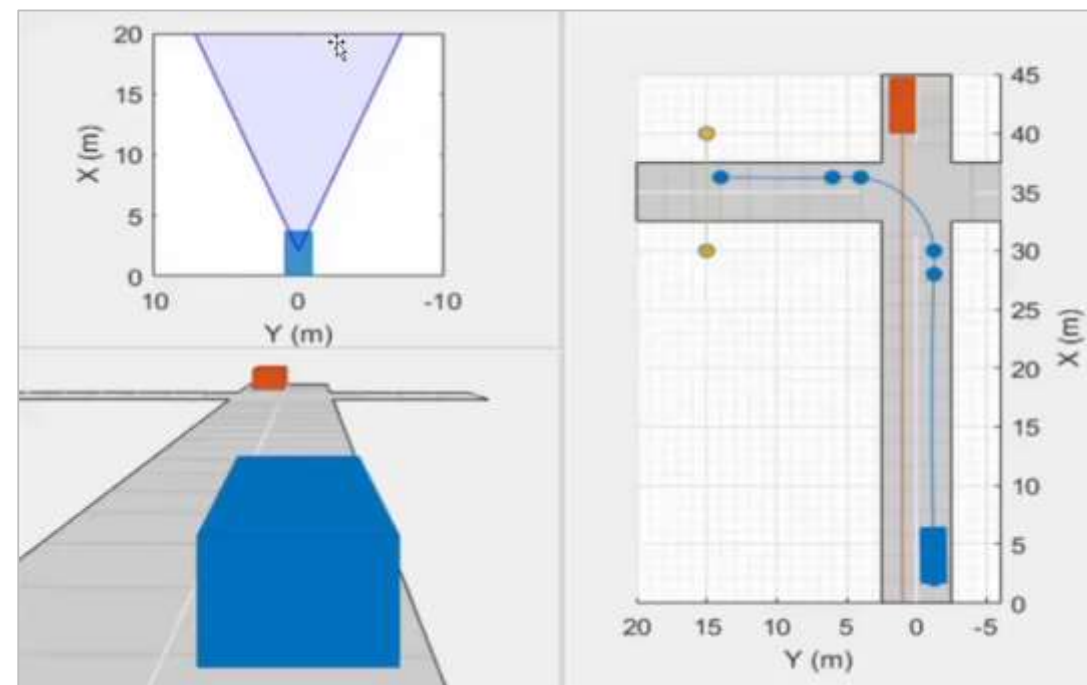


CamVid Database: Brostow, Gabriel J., Julien Fauqueur, and Roberto Cipolla. "Semantic object classes in video: A high-definition ground truth database." *Pattern Recognition Letters* Vol 30, Issue 2, 2009, pp 88-97.

Návrh autonómnych systémov

Návrh syntetických scenárov na testovanie reguátorov a algoritmov senzorickej fúzie

- Interaktívny návrh syntetických scenárov skladajúcich sa z ciest a účastníkov (*vozidlá, chodci, atď.*)
- Generovanie vizuálnych a radarových detekcií účastníkov

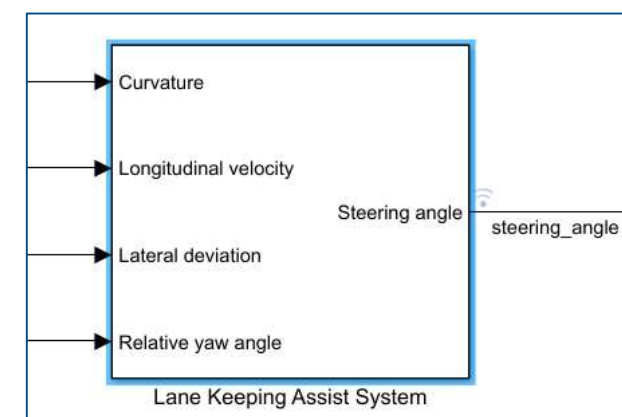
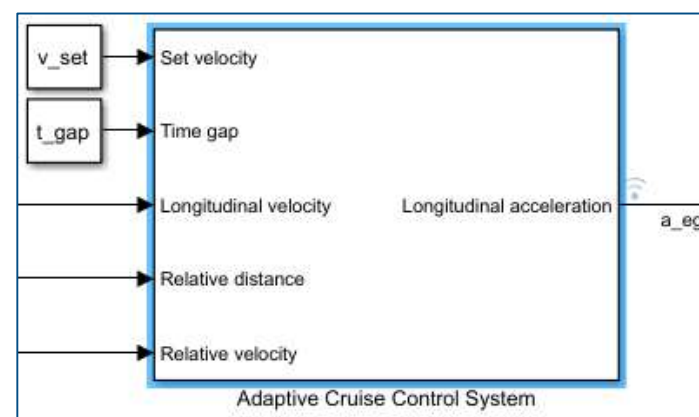


Driving Scenario Designer App

Návrh autonómnych systémov

Prediktívne riadenie na algoritmy adaptívnych tempomatov a udržania jazdného pruhu

- Predpripravené bloky na ľahký začiatok
- Jednoduché rozhrania na konfiguráciu prediktívneho riadenia
- Flexibilita nastavení pre rôzne aplikácie



Simulácia vozidla



Jazdné manévry



Podvozok/zavesenie

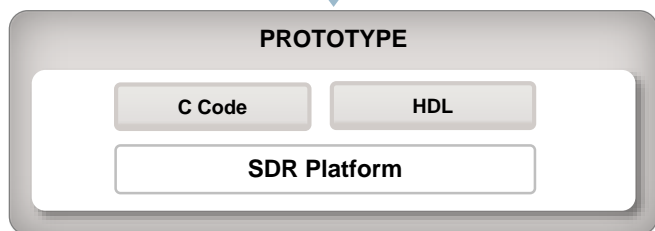
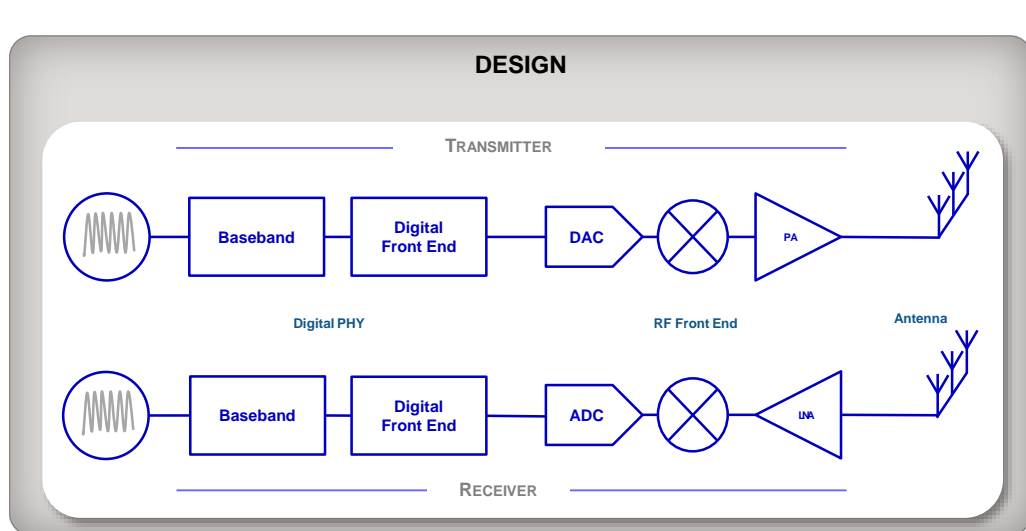


Autonómne riadenie

Vývoj s aktuálními bezdrôtovými štandardami



Model-based design v bezdrátových komunikáciach



- **Návrh a verifikácia algoritmov**
- **Návrh RF a antén**
- **Verifikácia a testovanie systémov**
- **Rapid Prototyping a produkcia**

Code Generation and Verification
Fixed-Point Designer
HDL Coder
HDL Verifier
LTE HDL Toolbox R2017b
Embedded Coder

Návrh a prototypovanie RF a antén

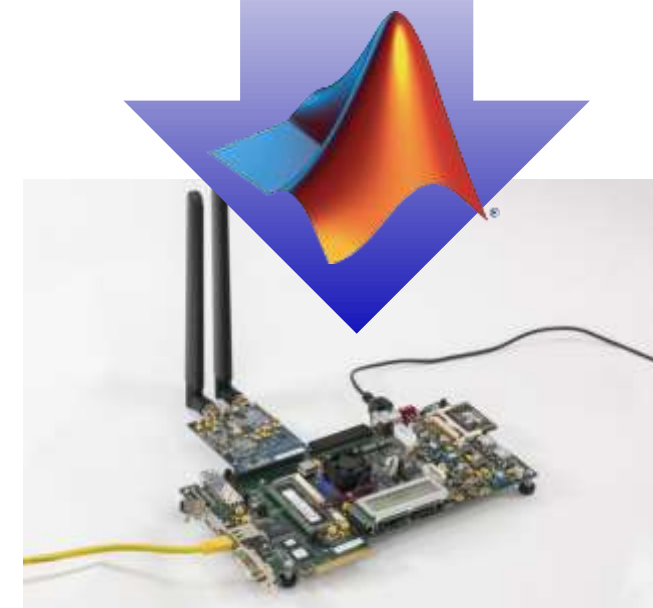
Využitie modelov RF a antén počas celého vývojového cyklu

- Návrh RF pomocou RF Budget Analyzer app
- Adaptívno-hybridný beamforming a modelovanie MIMO systémov
- Modelovanie RF zosilňovačov a DPD linearizácia
- Šírenie RF a 3D vizualizácia terénu
- Návrh a stavba tlačných (PCB) antén

Antenna Toolbox
RF Toolbox
RF Blockset



Od nápadu ...

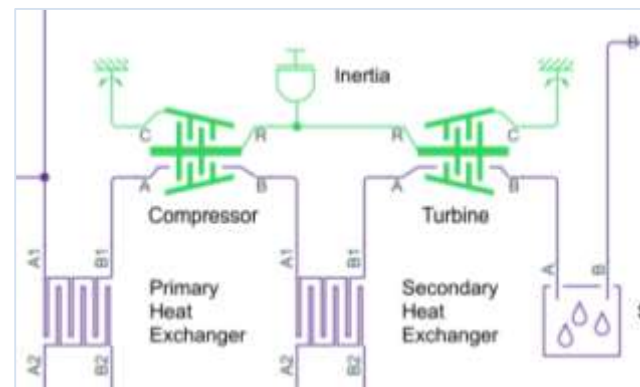


... po implementácii

Modelovanie systémov vlhkého vzduchu

Modelovanie HVAC a radiacích systémov prostredia

- Modelovanie a simulácia HVAC pre systémy ako sú budovi, autá alebo lietadlá
- Knižnica obsahuje komory, nádrže, miestne obmedzenia, energetické prevodníky, zdroje a snímače
- Zabezpečenie dovolených teplôt, tlaku, vlhkosti alebo kondenzácií v prostredí
- Poznámka: Beh simulácie vie ísť až 5x rýchlejšie s lokálnym solverom



Umelá inteligencia

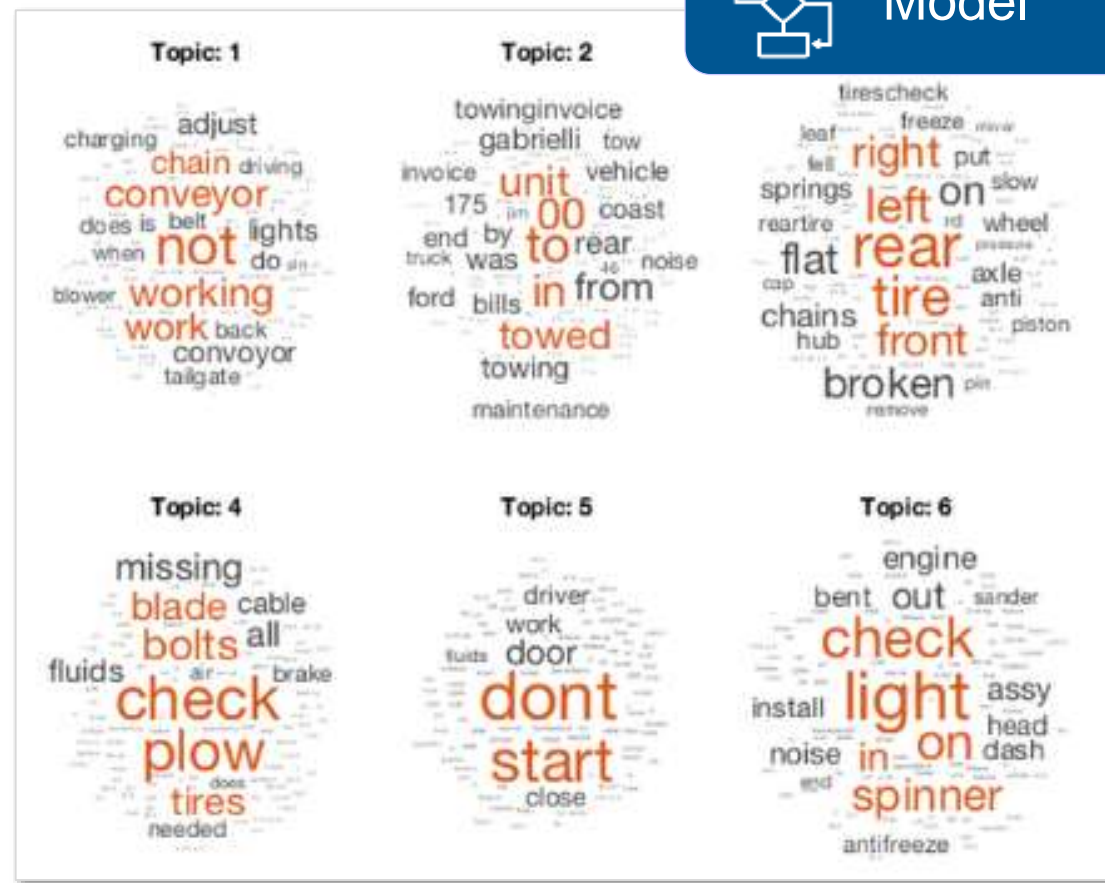


Textová analytika

 **Dáta**

```
repairNotes = 617x1 string array
"PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE"
"SERVICEROB,EXT,5604"
"NEED 4 PLOW PINS"
"INSTALL SPINNER ASSY"
"DONT START"
"DOG BONE PIN BROKEN"
"NEED SERVICE, CHECK BRAKES"
"HYD CAP CHECK ENGINE LIGHT ON"
"TARP VALVE STICKINGRIGHT SIDE MIRROR BRACKET I"
"HANDLES IN CAB LOOSE"
"NO PLOW LIGHTS"
"WILL NOT START"
```

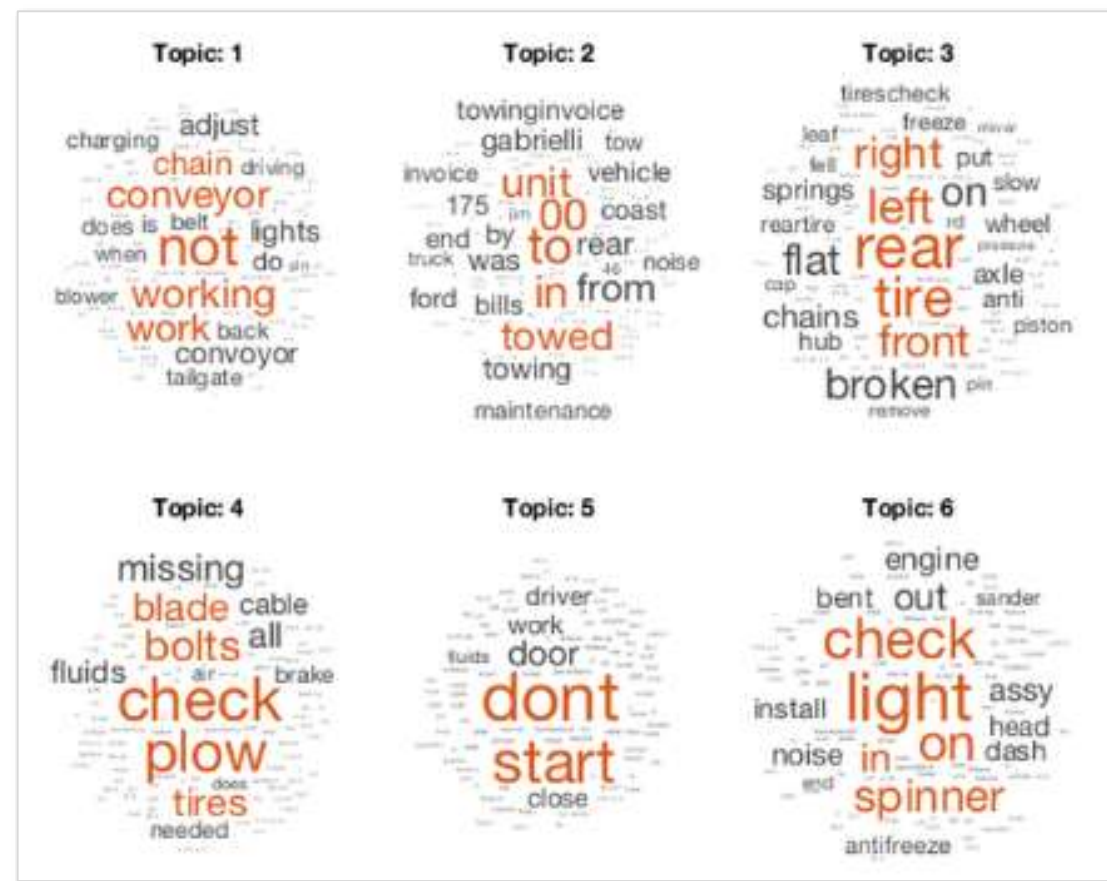
 **Model**



Textová analytika

Práca s textom zo záznamov a správ

- Predspracovanie dát – výber, filtrovanie a rozdelenie
- Vizualizácia textu pomocou world clouds a textových scatter grafov
- Návrh prediktívnych modelov s využitím vstavaných algoritmov strojového učenia (LDA, LSA, word2vec)

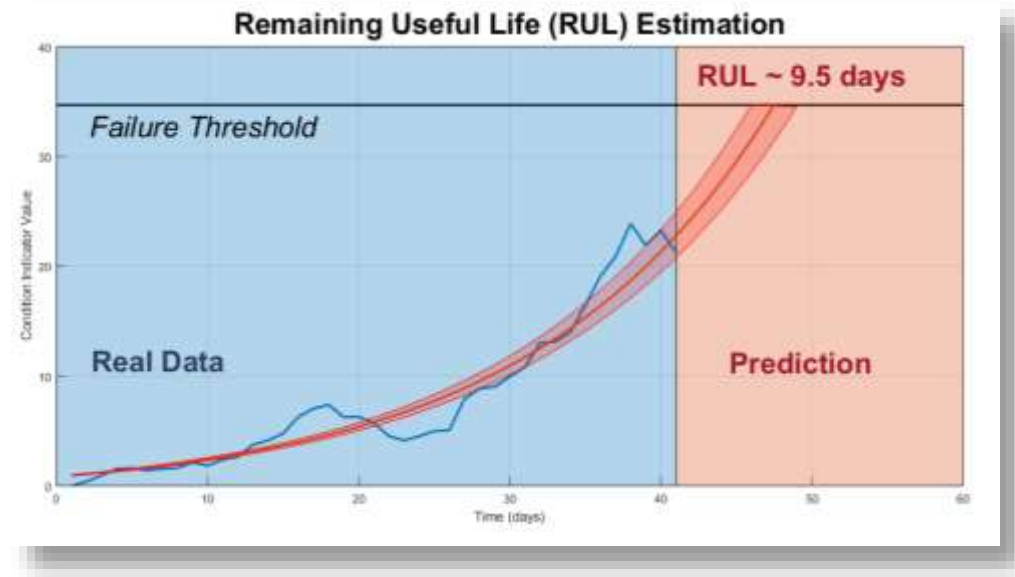
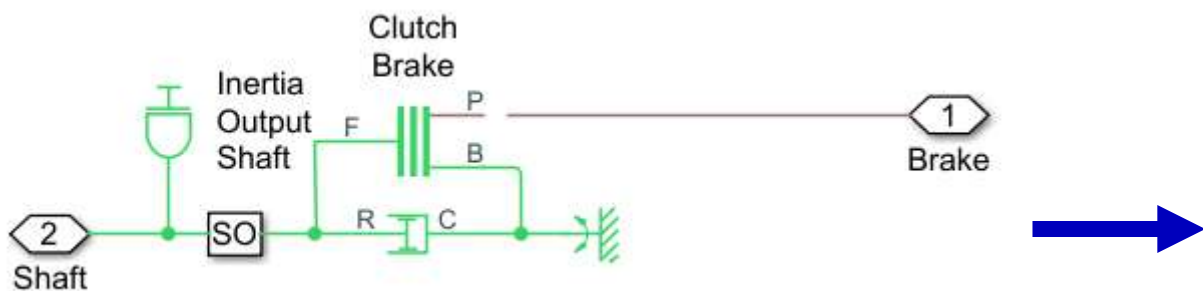


Prediktívna údržba

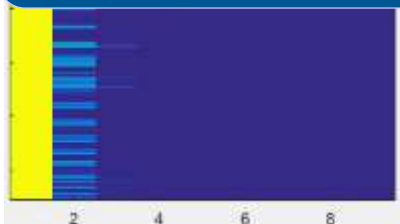
Dáta

Senzory

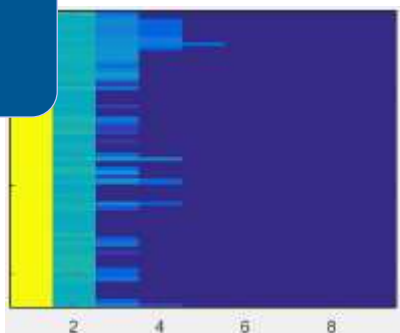
Model



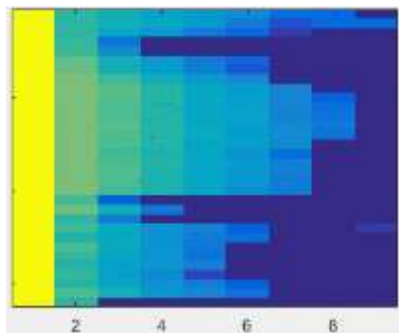
Výstup



Bežná prevádzka



Monitorovať

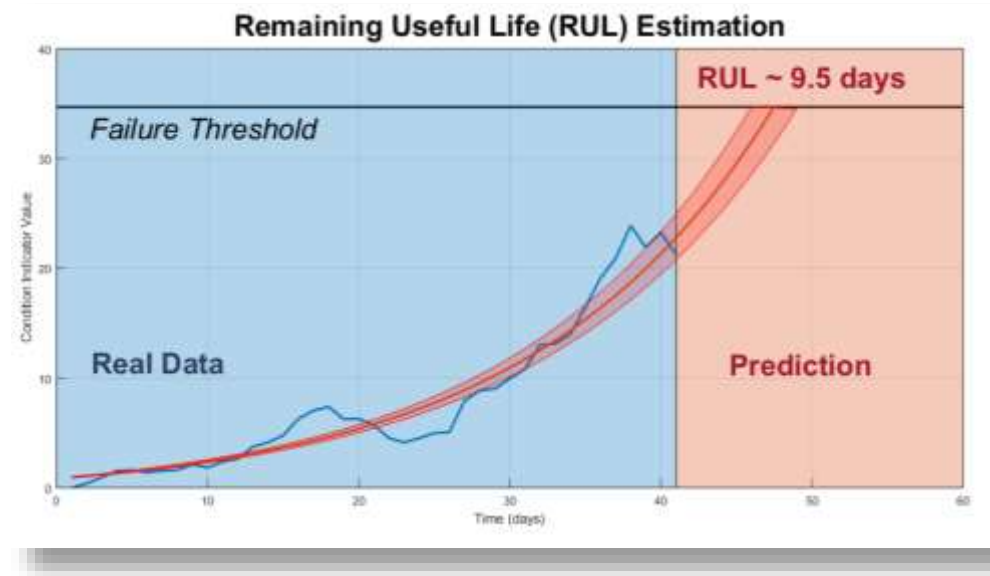


Potrebná údržba

Prediktívna údržba

Návrha testovanie algoritmov na sledovanie stavu a prediktívnej údržby

- Import dát senzorov zo súborov a cloudu (*Amazon S3, Windows Azure Blob Storage, and Hadoop HDFS*)
- Využitie simulovaných dát z modelov Simulinku
- Odhad zostavajúcej životnosti stroja (RUL)
- Príklady pre ľahké začatie (*motory, prevodovky, batérie a iné zariadenia*)



Deep Learning

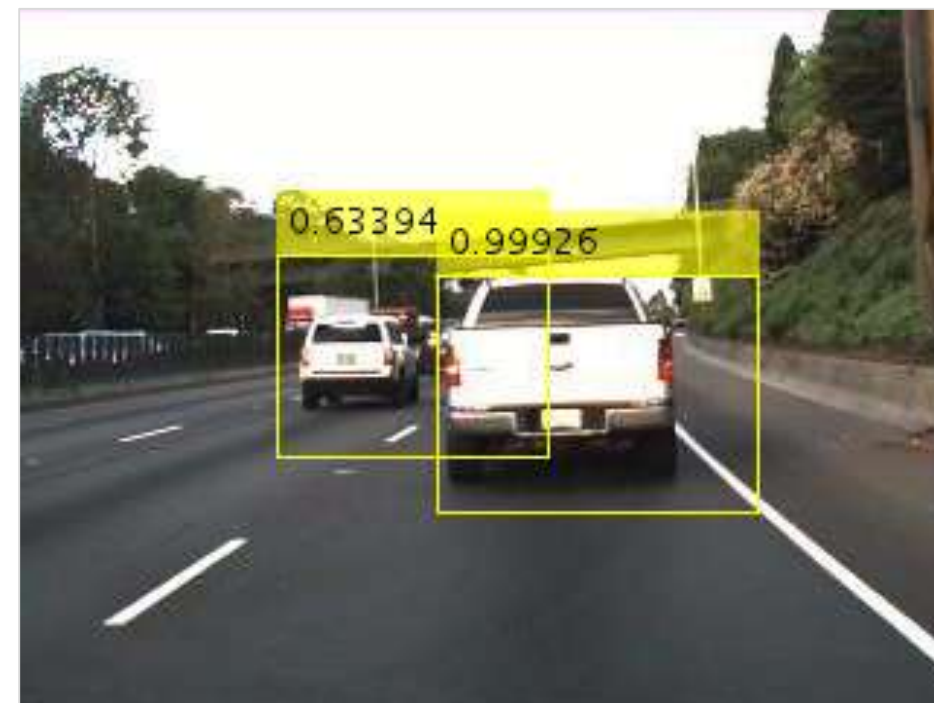
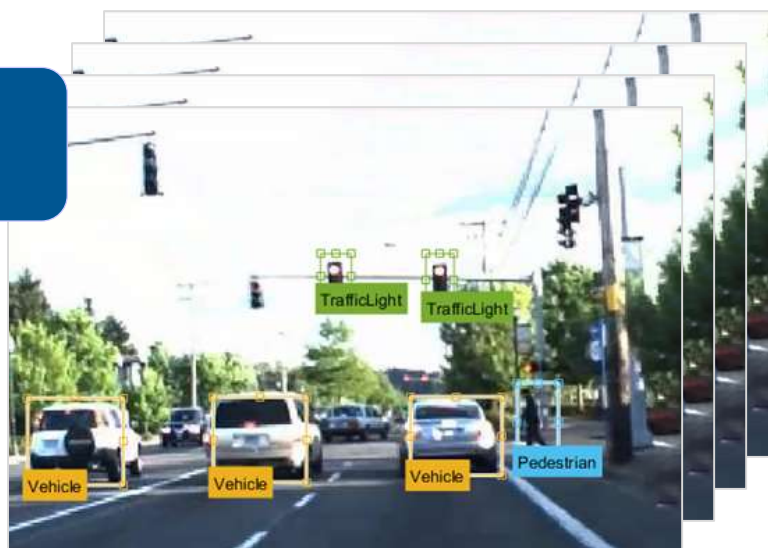
 **Dáta**



 **Model**



 **Výstup**



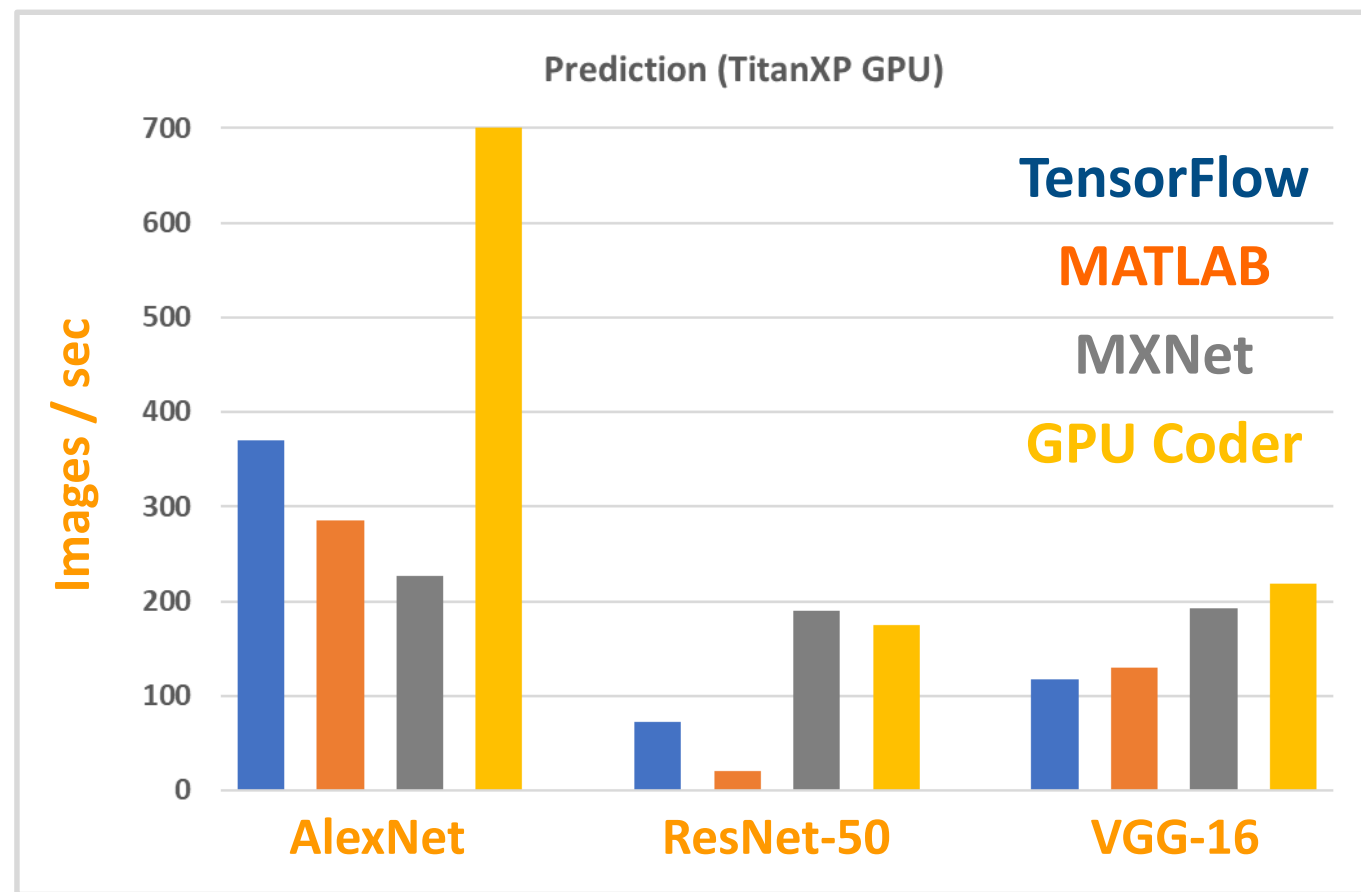
Neural Network Toolbox
Computer Vision System Toolbox
GPU Coder

Deep learning

Návrh, tvorba a vizualizácia konvolučných neuronových sietí

- Prístup k najnovším modelom
- Import predtrénovaných modelov a využitie dotrénovania (transfer learning)
- Automatický popis s aplikáciami
- Návrh vlastných modelov
- Využitie NVIDIA GPU na tréovanie
- Automatické generovanie optimalizovaného CUDA kódu pre nasadenie

Neural Network Toolbox
 Computer Vision System Toolbox
 GPU Coder



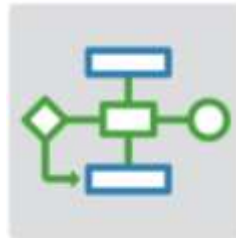
Čo je nové v MATLABe a Simulinku

Zvýšenie produktivity



- Rýchlejšia tvorba
- Analýza
- Simulácia, Komplexnosť
- Spolupráca

Pracovné postupy



- Nasadenie
- Generovanie kódu
- Verifikácia a validácia

Tvorba systémov



- Autonómne systémy
- Bezdrôtové komunikácie
- Umelá inteligencia (AI)

Aktualizácia kódu MATLABu a modelov Simulinku

Web Browser - (3 Errors) Code Compatibility Report

(3 Errors) Code Compatibility Report

Code Compatibility Report Top 3 Errors 1 Warning 304 Checks 2 Files

Analysis Date: 05-Sep-2017 14:32:08
MATLAB Version: R2017b

Incompatibility and Syntax Errors

Row	Filename	Line	Description
1	classifyBloodPressure.m	18	TREEFIT has been removed
2	classifyBloodPressure.m	21	TREEDISP has been removed. Use TREEVIEW methods instead.
3	classifyBloodPressure.m	24	TREEVAL has been removed. Use TREEVIEW methods instead.

Warnings and Other Recommendations

Row	Filename	Line	Description
1	classifyBloodPressure.m	Z	RAND or RANDN with 'seed' is not recommended. Use RANDI or RANDJ instead.

Upgrade Advisor - sf_climate_control

File Edit Run Settings Help

Find: [] Disable Upgrade Notifications

Upgrade Project Report

100% Passed

Models: 7, Libraries: 1, MATLAB Code: 8

Need attention: -

Show: All Files All Results

File	Check Name	Result
AnalogControl.mdl	Check model settings for migration to simplified initialization mode	Passed
analyzeModelFiles.m	Check that the model is saved in SLX format	Passed
billOfMaterials.m	Check usage of function-call connections	Passed with fixes
checkCodeProblems.m	Check and set embedded target model to use rt.tlc system target file	Need attention
DigitalControl.slx	Check and update masked blocks in library to use promoted parameters	Passed
f14_airframe.slx	Check and update mask image display commands with unnecessary imread() function calls	Passed
f14_airframe_test.m	Check and update mask to affirm icon drawing commands dependency on mask workspace	Passed
find_top_models.m	Check and update model to use toolchain approach to build generated code	Passed
LinearActuator.slx		Passed
NonLinearActuator.mdl		Passed
rebuild_s_functions.m		Passed
runUnitTest.m		Passed
siproject_f14.slx		Passed
upgrade_project.m		Passed
vertical_channel.slx		Passed
wind_gust_lib.slx		Passed

Check model settings for migration to simplified initialization mode [Learn more](#)

Check for model level messages
This check finds and reports model level messages for migrating to simplified initialization mode.

See Also

- Check model settings for migration to simplified initialization mode
- Underspecified initialization detection

Checks run on 02/01/2018 10:44

Publish Report Close

Identify Variant Model blocks and convert those to Vari

Analysis

Upgrade Variant Model blocks to Variant Subsystems contain offers enhanced capabilities while maintaining equivalent fun variant models will be removed in a future release.

Run This Check

Result: Passed

Identify Variant Model blocks at model level.

Passed
No Variant Model blocks found.

Ďakujeme