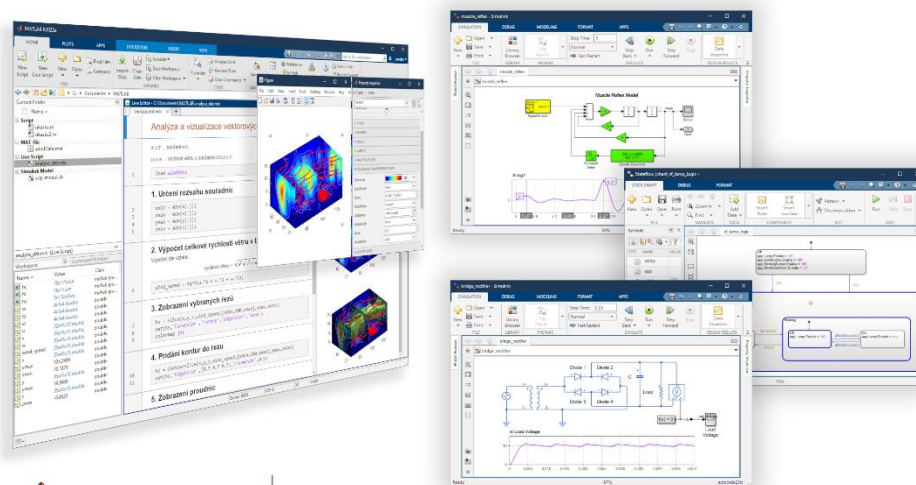


## AI jako součást metody Model-Based Design: Redukované modely



Michal Blaho

blaho@humusoft.sk

[www.humusoft.cz](http://www.humusoft.cz)

[info@humusoft.cz](mailto:info@humusoft.cz)

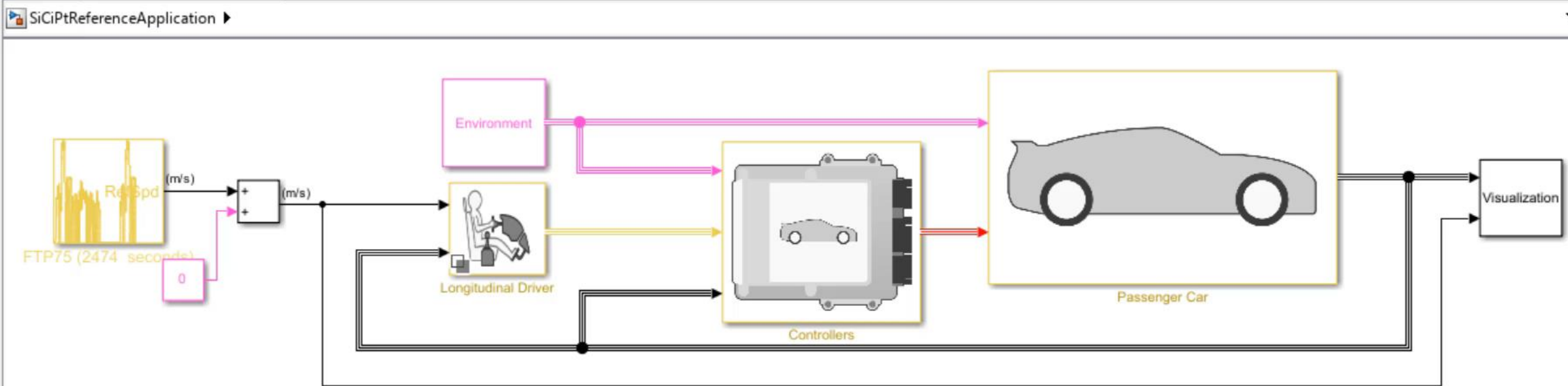
[www.mathworks.com](http://www.mathworks.com)

**SIMULATION**    **DEBUG**    **MODELING**    **FORMAT**    **APPS**

Project | New | Open | Save | Print | Library Browser | Log Signals | Add Viewer | Signal Table | Stop Time: 250 | Accelerator | Fast Restart | Step Back | Run | Step Forward | Stop | Data Inspector | Logic Analyzer | Bird's-Eye Scope

PROJECT    FILE    LIBRARY    PREPARE    SIMULATE    REVIEW RESULTS

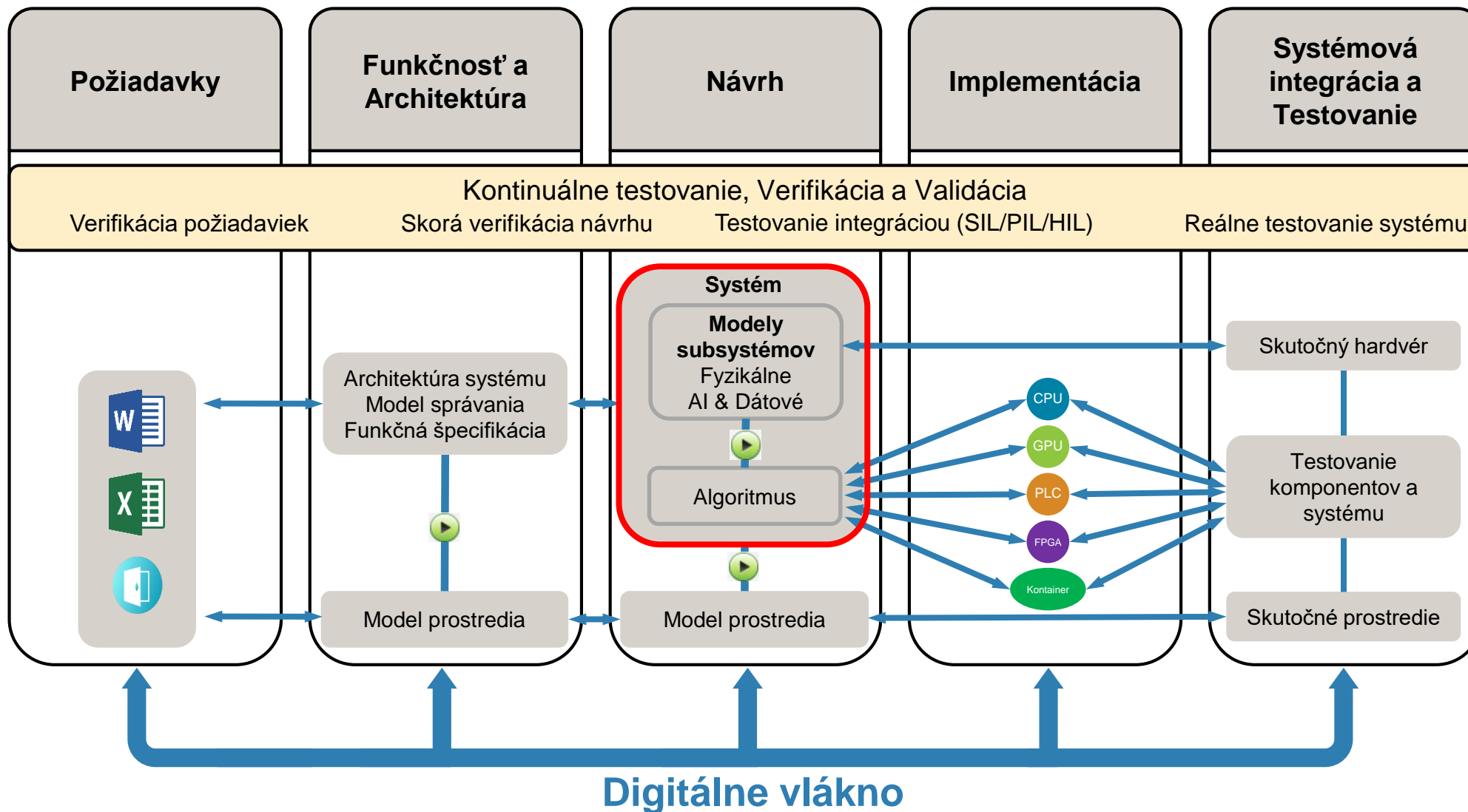
Model Browser: SiCiPtReferenceApplication



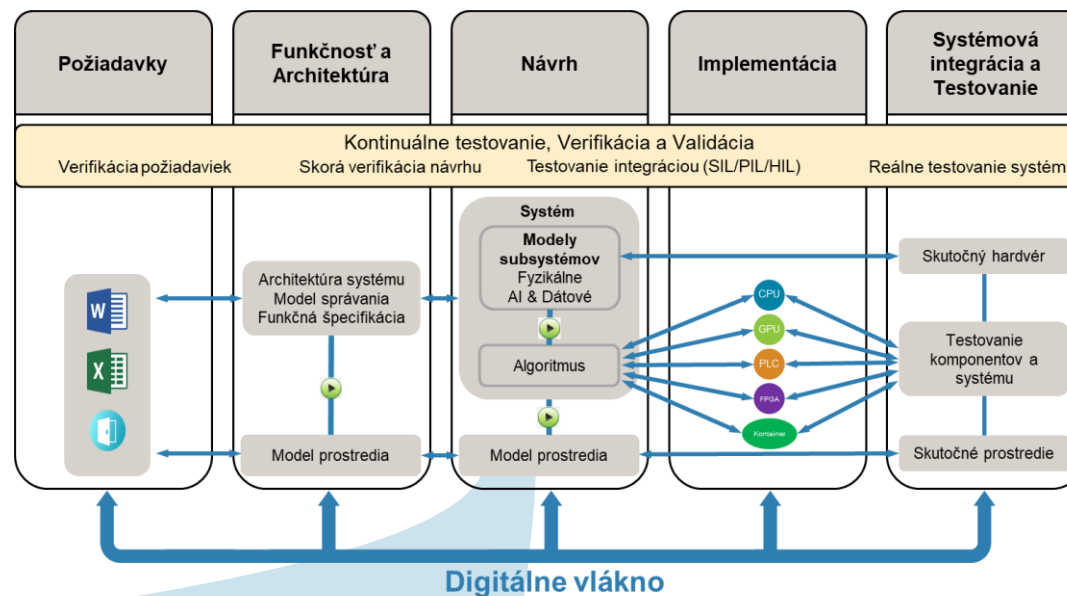
Analyze Power and Energy

Help

# Model-Based Design



# Integrácia modelov AI do metódy Model-Based Design



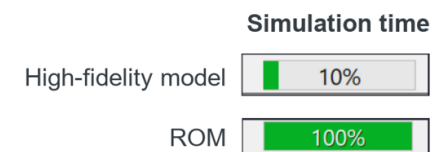
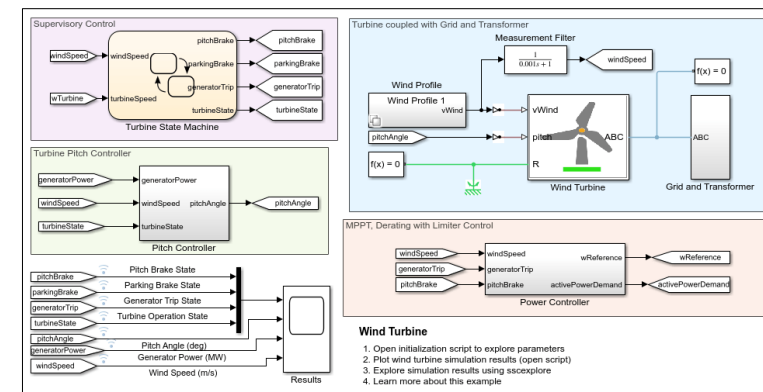
## AI pre komponenty

- Zrýchlenie desktopovej a HIL simulácie
- Modelovanie dynamiky komponentov z dát, keď fyzikálne princípy nie sú známe

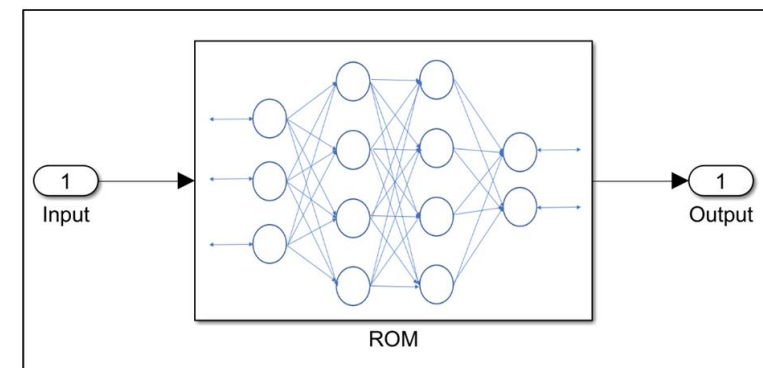
# Redukované modely

- ROM – Reduced Order Modeling
- Technika redukcie výpočtovej zložitosti alebo pamäťových nárokov počítačového modelu
  - zachováva očakávanú vernosť v rámci tolerovanej chyby
  - zachytáva základné chovanie systému
- Aplikácie
  - urýchlenie simulácie na systémovej úrovni
  - umožňuje simulácie na úrovni systémov
  - digitálne dvojča
  - návrh riadiacich systémov
  - testovanie simulácií typu hardware-in-the-loop

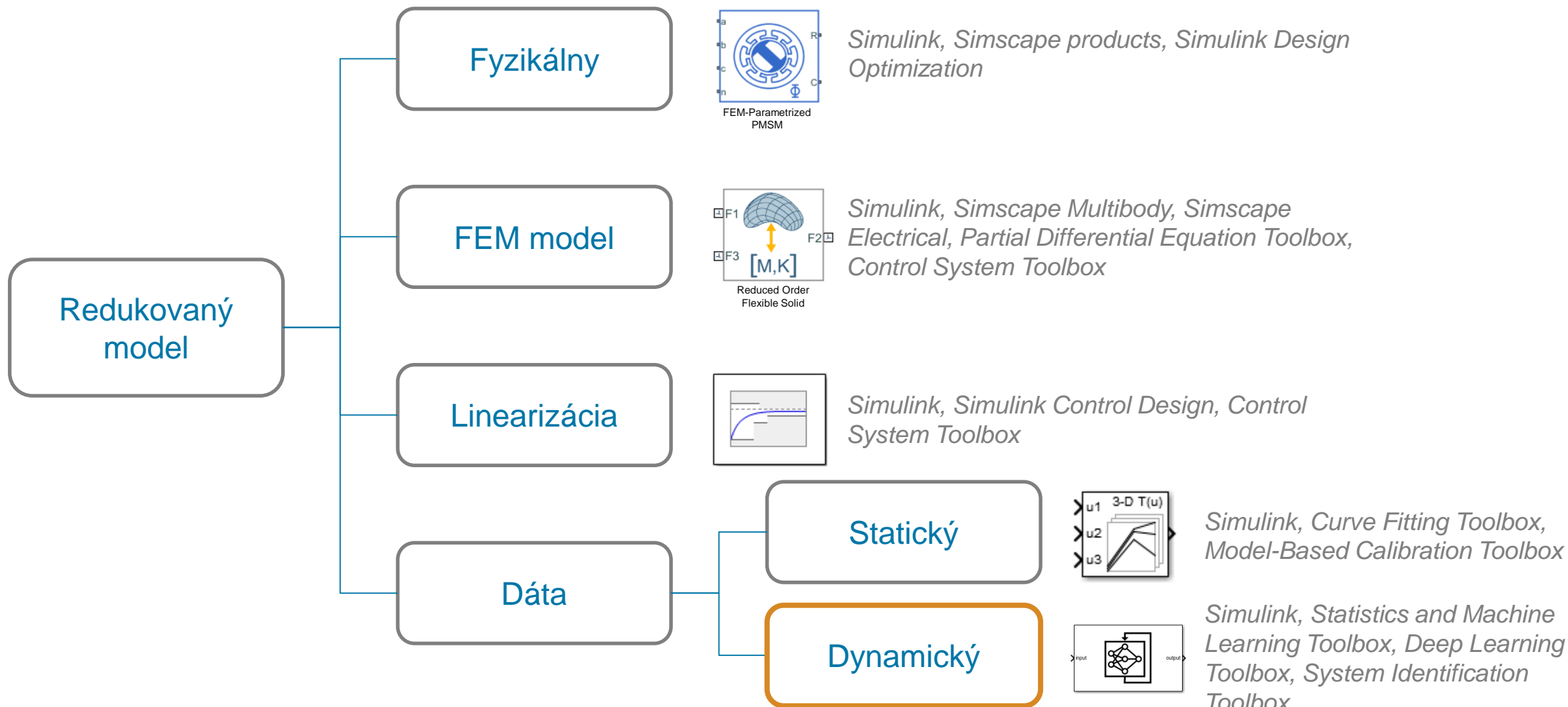
## Detailný model



## Reduced-Order Model (ROM)

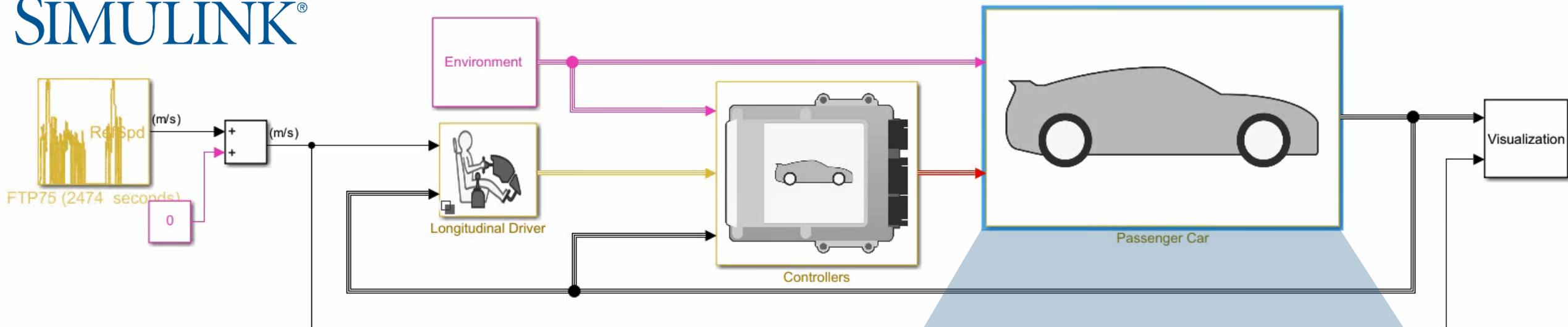


# Redukované modely



# Nahradenie modelu motora s ROM založenom na AI

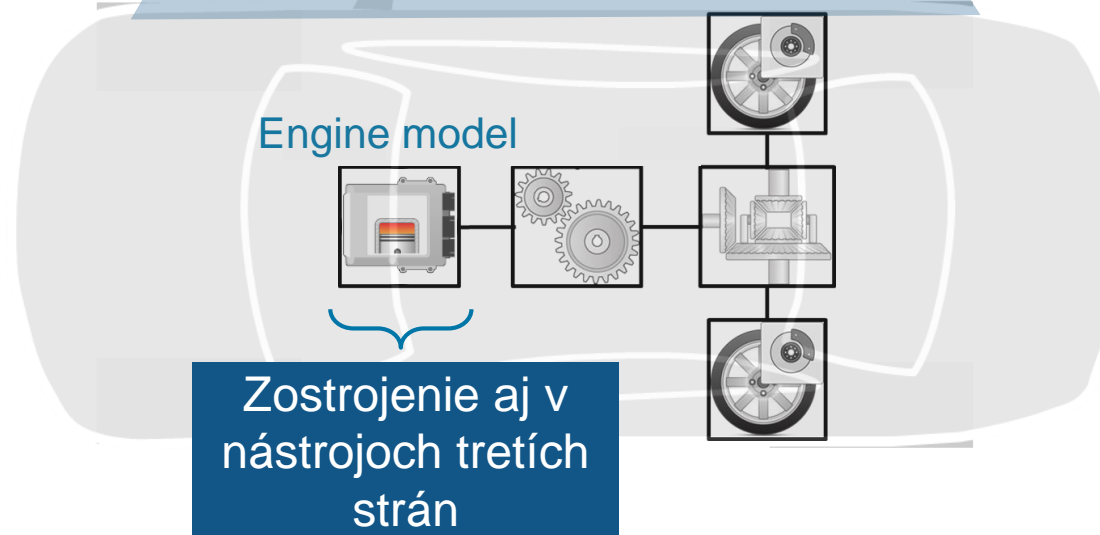
SIMULINK®



Vysoká presnosť

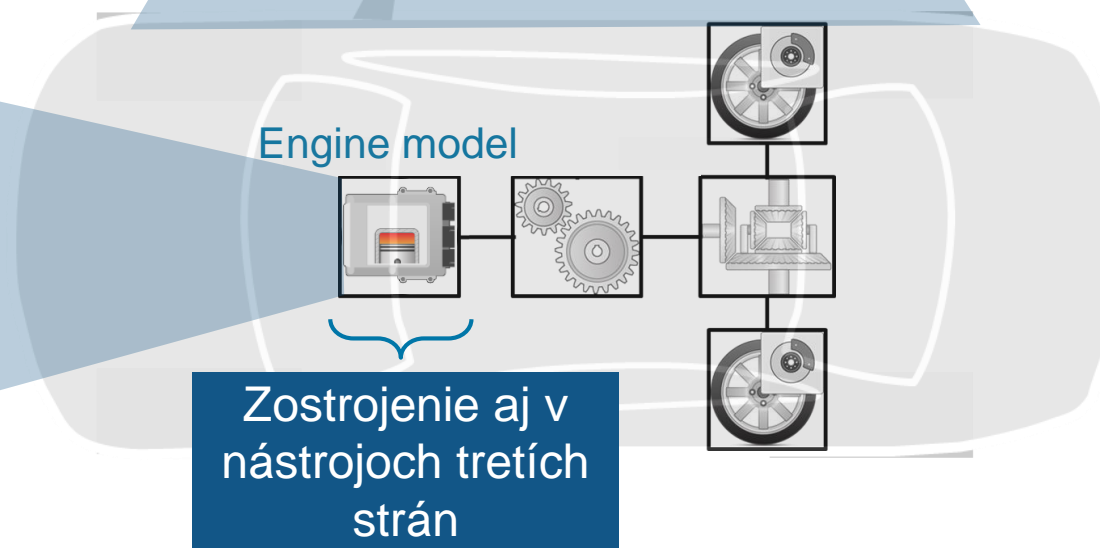
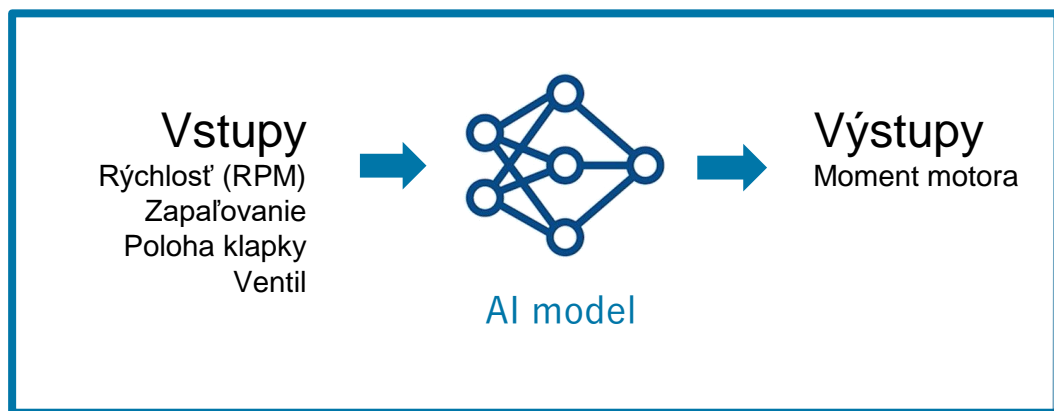
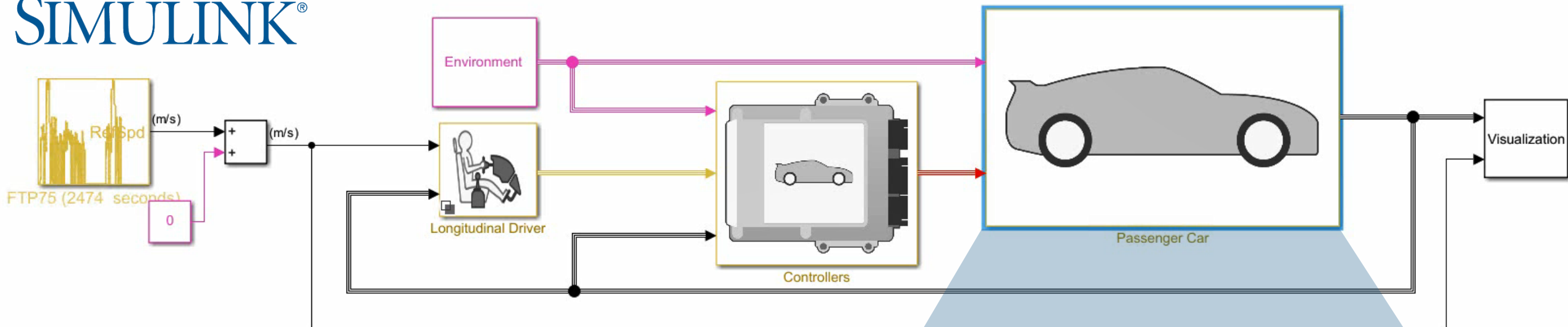
Komplexný model

Pomalá simulácia



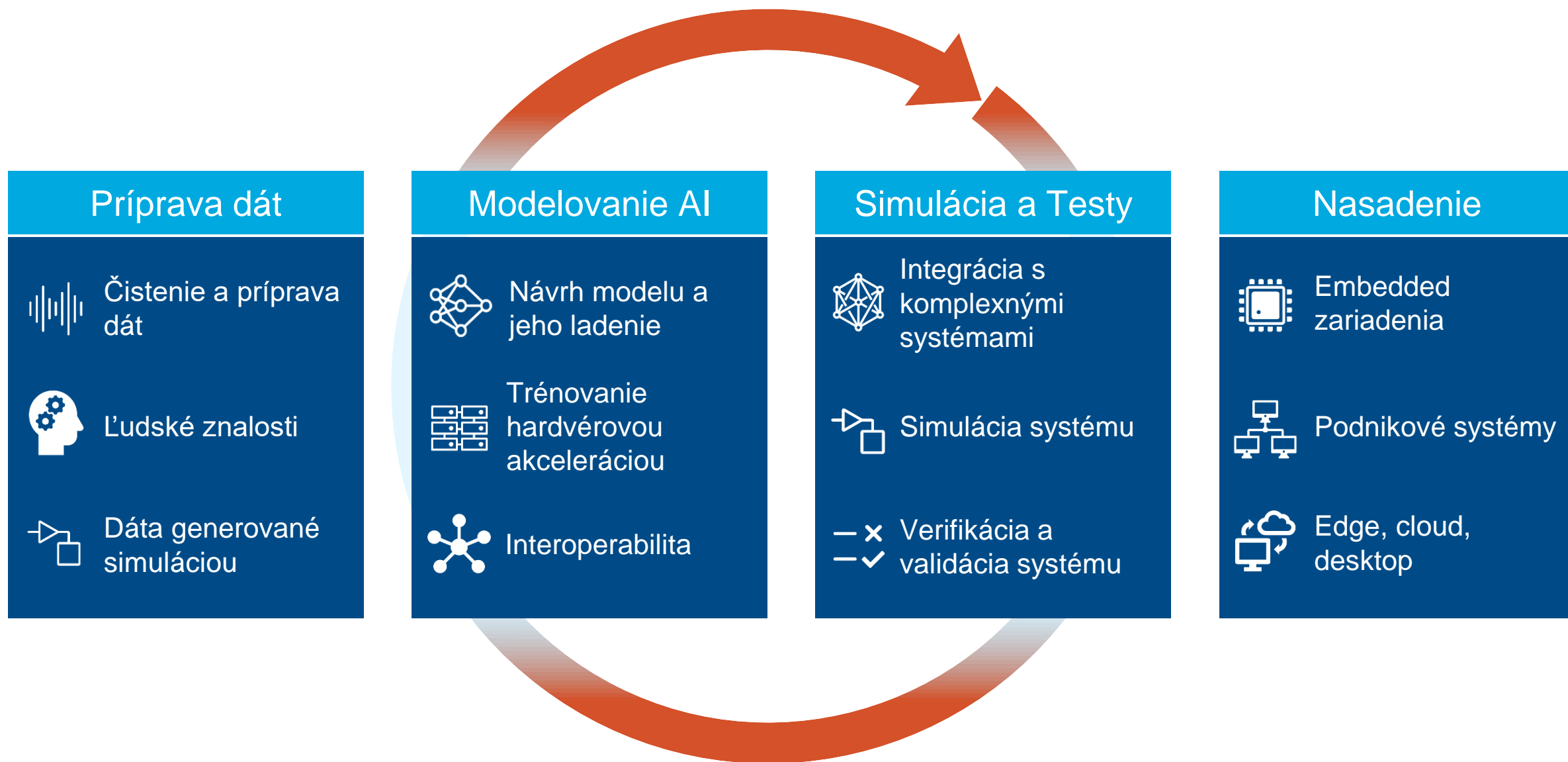
# Nahradenie modelu motora s ROM založenom na AI

SIMULINK®





# Návrh systémov založených na AI

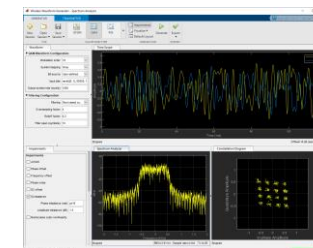


# Generovanie syntetických dát

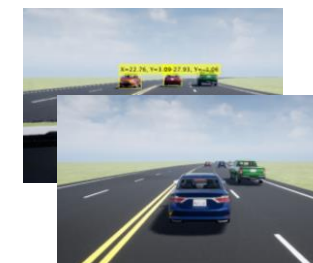
Iné prístupy:



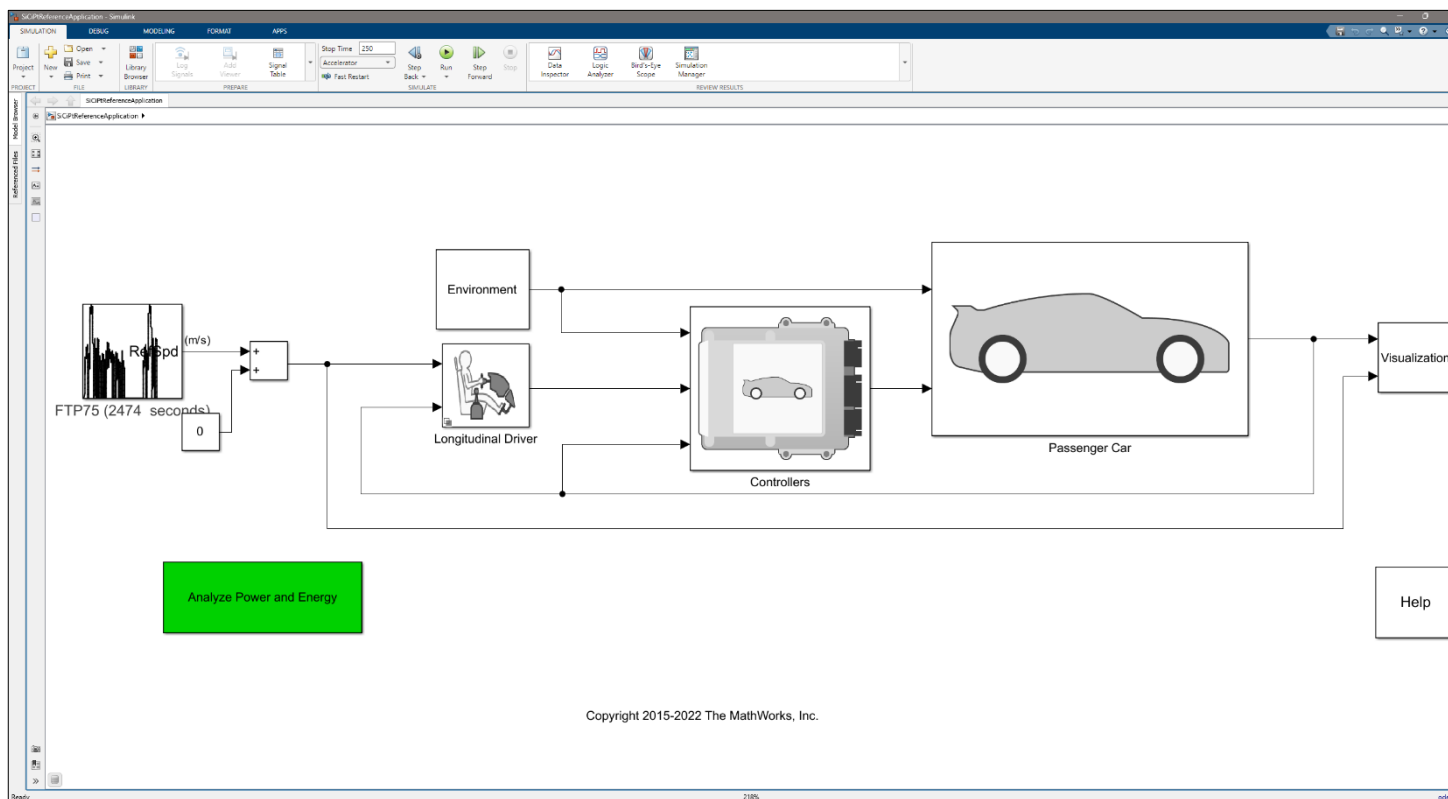
GANs



Wireless Waveform Generator



Unreal Engine®



Simulink/Simscape

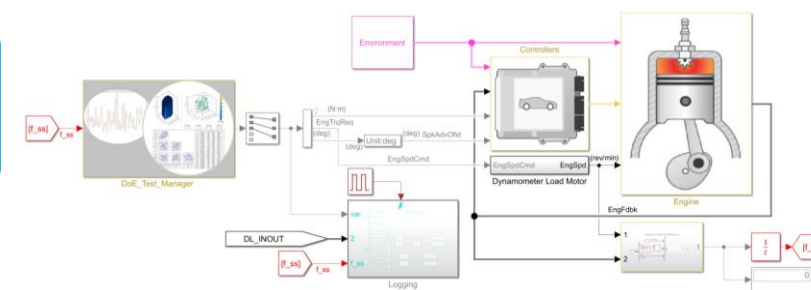
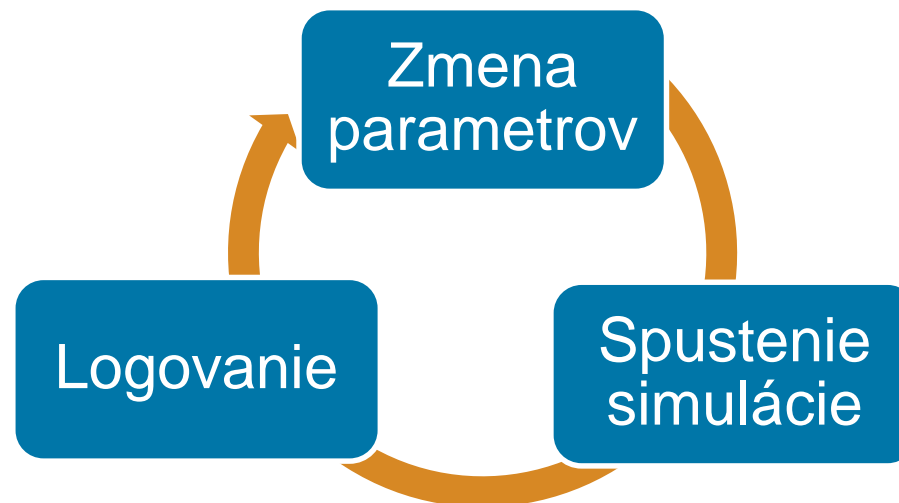
# Generovanie syntetických dát

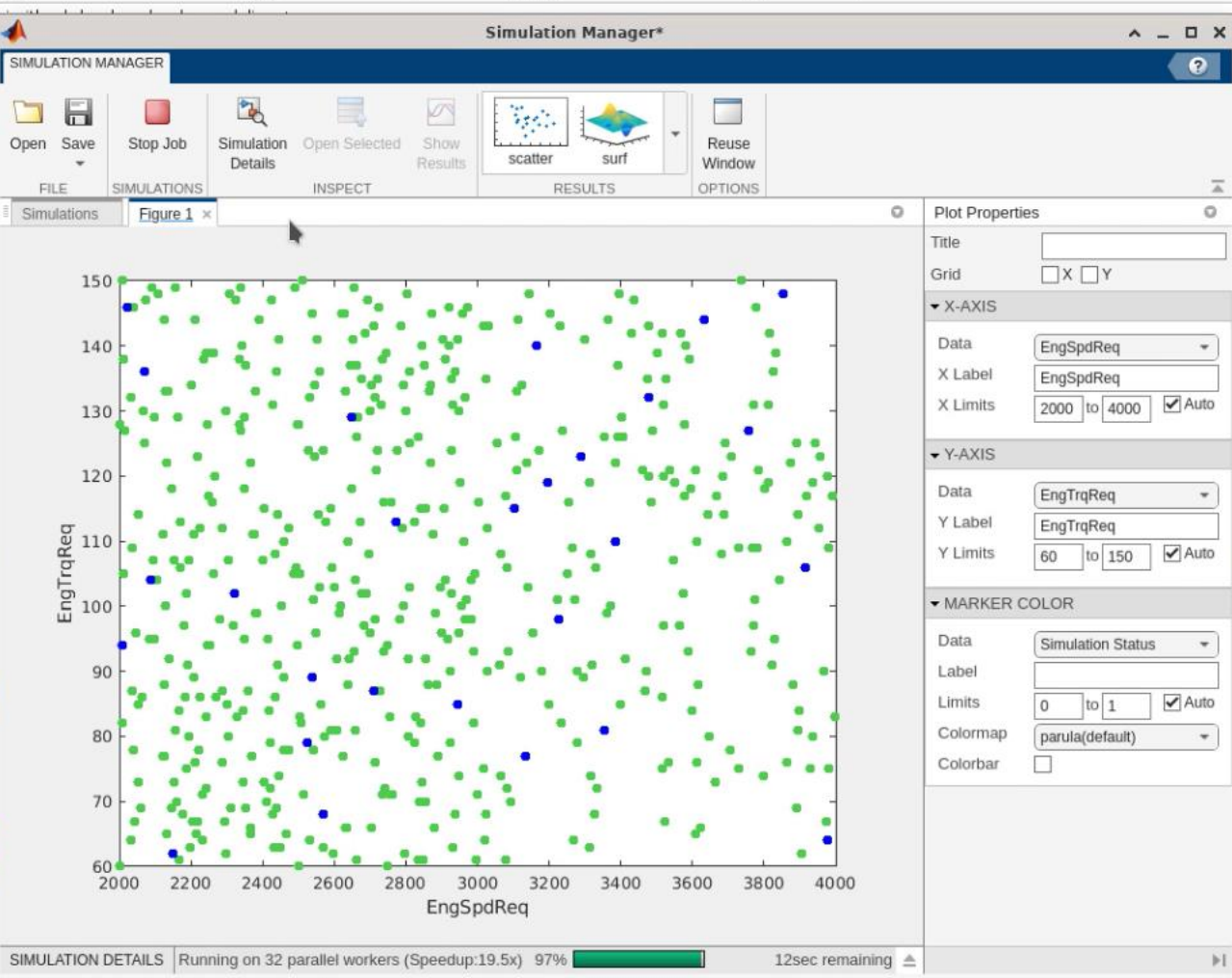
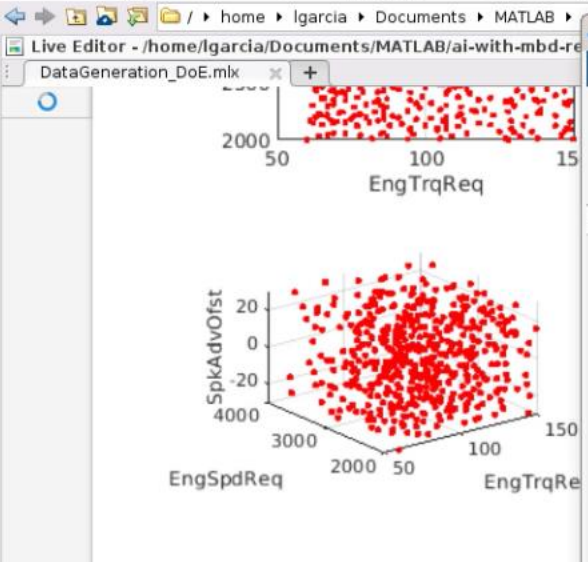
DoE = 512x3 table

	EngTrqReq	EngSpdR...	SpkAdvOfst
1	60	2000	-30
2	128	2500	15
3	94	2750	8
4	111	2875	-19
5	77	2625	-11
6	144	2125	4
7	85	2563	-21
8	119	3313	-28
9	68	2938	21

Vstup  
 Rýchlosť (RPM)  
 Zapaľovanie  
 Poloha klapky  
 Ventil

Výstup  
 Moment motora





### Execute Simulation

For this demo, the engine model is brought from

- <https://www.mathworks.com/help/aut>
- <https://www.mathworks.com/help/aut>

Default setting: Simulation Type => "Rapid Ac"

```

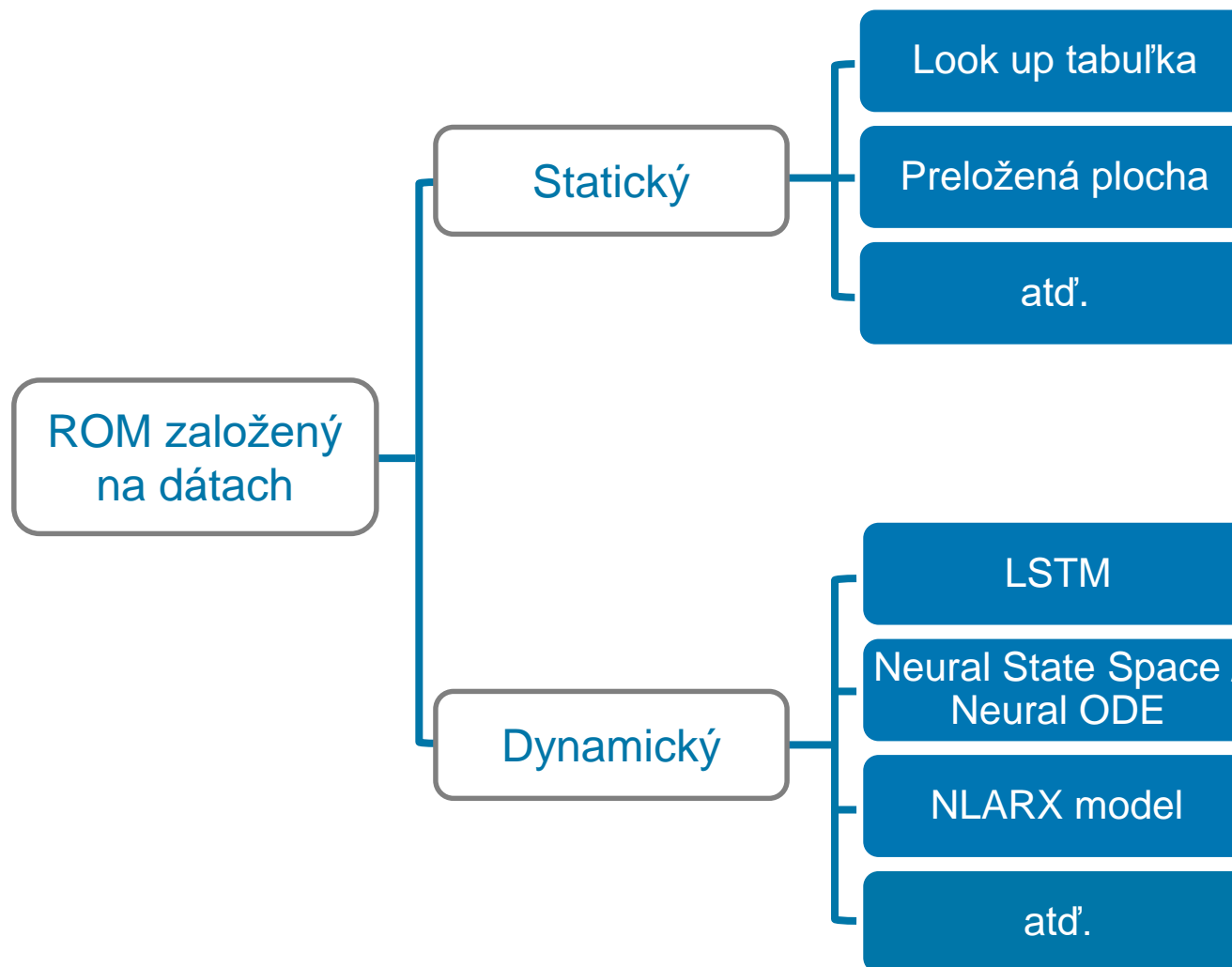
36 open_system('DoE_Engine_Test.slx')
37 % Run parallel simulations, 300s
38 simout = helper.simulation(DoE, 'Rapid Acquisition', 300);
    
```

```

Starting parallel pool (parpool) using local 'parpool'
Connected to the parallel pool (number of workers: 32)
[10-Jun-2022 03:33:09] Checking for availability of parallel pool...
[10-Jun-2022 03:33:09] Starting Simulink on parallel workers...
[10-Jun-2022 03:33:56] Loading project on parallel workers...
[10-Jun-2022 03:33:56] Configuring simulation cache folder on parallel workers...
[10-Jun-2022 03:34:18] Transferring base workspace variables used in the model to parallel workers...
[10-Jun-2022 03:34:20] Loading model on parallel workers...
[10-Jun-2022 03:34:53] Running simulations...
    
```

### Check and Save Simulation Result

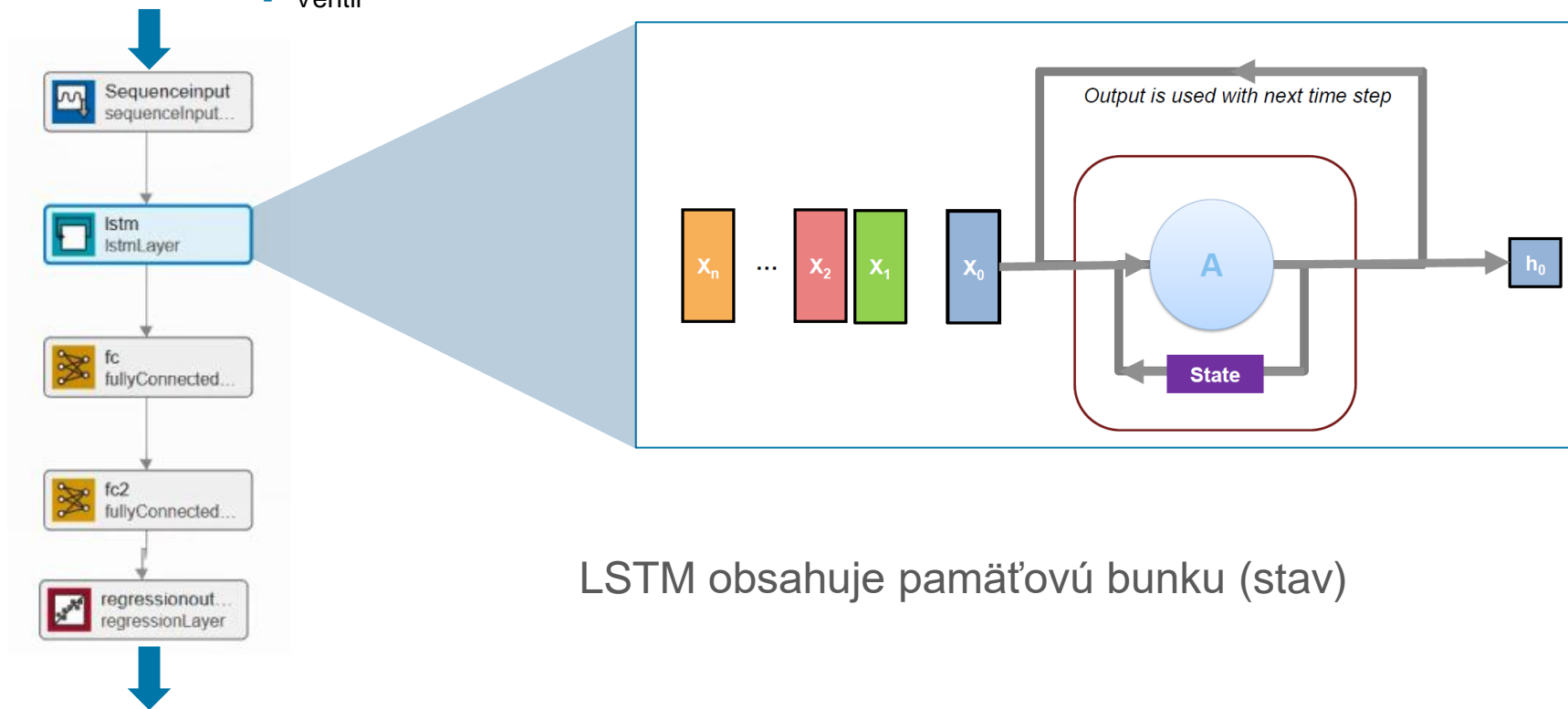
# Redukované modely založené na datech



# Redukovaný model pomocou LSTM

**Vstupy**

- Rýchlosť (RPM)
- Zapaľovanie
- Poloha klapky
- Ventil



LSTM obsahuje pamäťovú bunku (stav)

**Výstupy**

- Moment motora

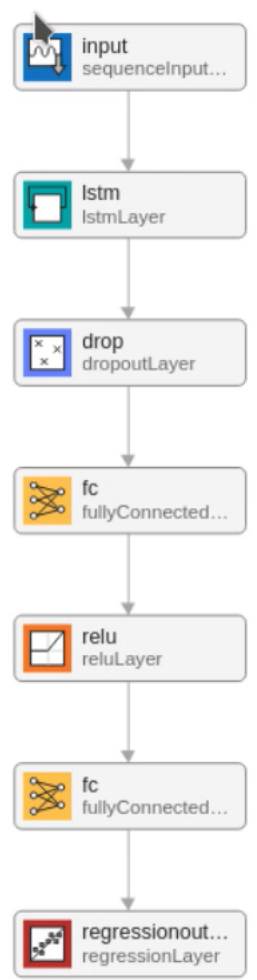
# Deep Network Designer

**DESIGNER**

New Duplicate Cut Copy Paste Fit to View Zoom In Zoom Out Auto Arrange Analyze Export

NETWORK BUILD NAVIGATE LAYOUT ANALYSIS EXPORT

- Layer Library
- Filter layers...
- resize2dLayer (for reference input)
  - resize3dLayer (for scale)
  - resize3dLayer (for output size)
  - resize3dLayer (for reference input)
- OUTPUT
- softmaxLayer
  - sigmoidLayer
  - classificationLayer
  - regressionLayer
  - rpnSoftmaxLayer
  - rcnnBoxRegressionLayer
  - rpnClassificationLayer
  - pixelClassificationLayer
  - dicePixelClassificationLayer
  - yolov2OutputLayer
  - focalLossLayer



Properties

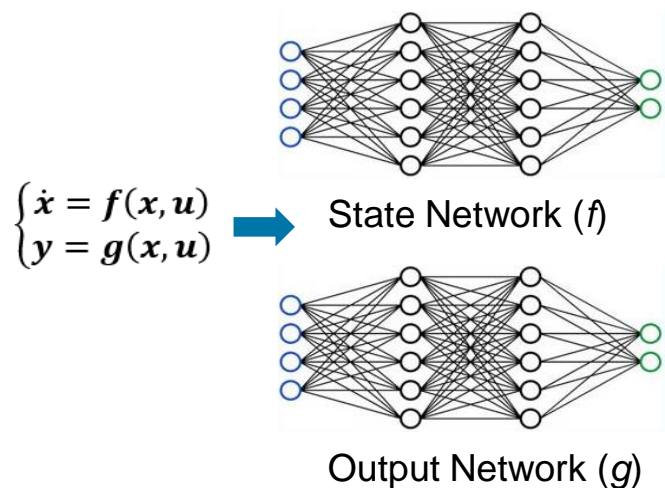
Input type	Sequence
Output type	Regression
Number of layers	7
Number of connections	6



48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69

validationData, {xval: UNormalized, yval: UNormalized}, ...

# Redukovaný model pomocou Neural State Space



Multi-layer Perceptron (feedforward) networks

Vyžaduje Deep Learning Toolbox

Nelineárna stavová funkcia  $f$  a nelineárna výstupná funkcia  $g$  sú dopredné neurónové siete, ktoré sa učia z dát

(známy aj ako Neural ODE)

```

% Define a neural state sapce model
obj = idNeuralStateSpace(1,NumInputs=4); % no output Y in this case

%% Configure state network
obj.StateNetwork = createMLPNetwork(obj,'state',LayerSizes=[128 128], ...
    WeightsInitializer="glorot",BiasInitializer="zeros", Activations='tanh');

%% Specify training options for state network
StateOpt = nssTrainingOptions('adam');
StateOpt.MaxEpochs = 90;
StateOpt.MinibatchSize = 100;
StateOpt.InputInterSample = 'pcchip';

%% Train the system
obj = nlssest(Ucell,Xcell,obj,StateOptions=StateOpt);
    
```



HOME PLOTS APPS LIVE EDITOR INSERT VIEW

New Open Save Compare Print Export FILE  
Go To Find Bookmark NAVIGATE  
Text Normal B I U M CODE  
Code Control Task Refactor SECTION  
Run Section Run and Advance Run to End RUN  
Run Step Stop

/ > home > lgarca > Documents > MATLAB > ai-with-mbd-reduced-order-modeling >

Live Editor - /home/lgarca/Documents/MATLAB/ai-with-mbd-reduced-order-modeling/Part 1 - AI Modeling/ROM\_2a\_SysID\_NeuralStateSpace.mlx \*

ROM\_2a\_SysID\_NeuralStateSpace.mlx \* AI\_ROM\_1\_Simulation.mlx +

## 2. Design and estimate Neural State Space Model

```
79 doEstimation = true; %#ok<*UNRCH>
80
81 if doEstimation
82 % Define a neural state space model
83 nssobj = idNeuralStateSpace(1,NumInputs=4); % no output Y in this case
84
85 %% Configure state network
86 nssobj.StateNetwork = createMLPNetwork(nssobj,'state',LayerSizes=[16 16], ...
87     WeightsInitializer="glorot",BiasInitializer="narrow-normal",Activations='tanh');
88
89 %% Specify training options for state network
90 StateOpt = nssTrainingOptions('adam');
91 StateOpt.LearnRate = 0.005;
92 StateOpt.MaxEpochs = 90;
93 StateOpt.MinibatchSize = 200;
94 StateOpt.InputInterSample = 'pchip';
95
96 %% Train the system
97 tic;
98 nssobj = nlssest(Ucell,Ycell,nssobj,StateOpt);
99 toc;
```

# Neural State Space interaktívne

- Estimate Neural State-Space Model
  - Live Editor task
  - interaktívna estimácia a validácia
- Návrh modelu
  - tvorba štruktúry
  - parametrizácia modelu
  - parametrizácia riešiča
- Tvorba automatizovaného skriptu
  - Generovanie MATLAB kódu

**Estimate Neural State-Space Model** ▶  Autorun | ?

`sys` = Estimated continuous neural state-space model for "tt1" with 1 states

▼ Select data

Data type:  ▼

Estimation data: Timetable  ▼

	u	y
Input (u)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Output (y)	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Validation data: Timetable  ▼

▼ Specify model structure

Number of states:  ▼

Time invariant Time domain:  ▼  Feedthrough

$\dot{x}(t) = F(x(t), u(t))$   
 $y(t) = x(t)$

▼ State network

Activation function:  ▼ Number of layers:  ▼ Layer size:

Weights initializer:  ▼ Bias initializer:  ▼

› ODE Solver options

▼ Specify training options

Training algorithm:  ▼ Gradient decay factor:  Squared gradient decay factor:

Loss function:  ▼ Learn rate:  Maximum number of epochs:  ▼

Size of mini-batch:  ▼ Input intersample:  ▼

Show fit to validation data during training Validation data fit frequency:  ▼

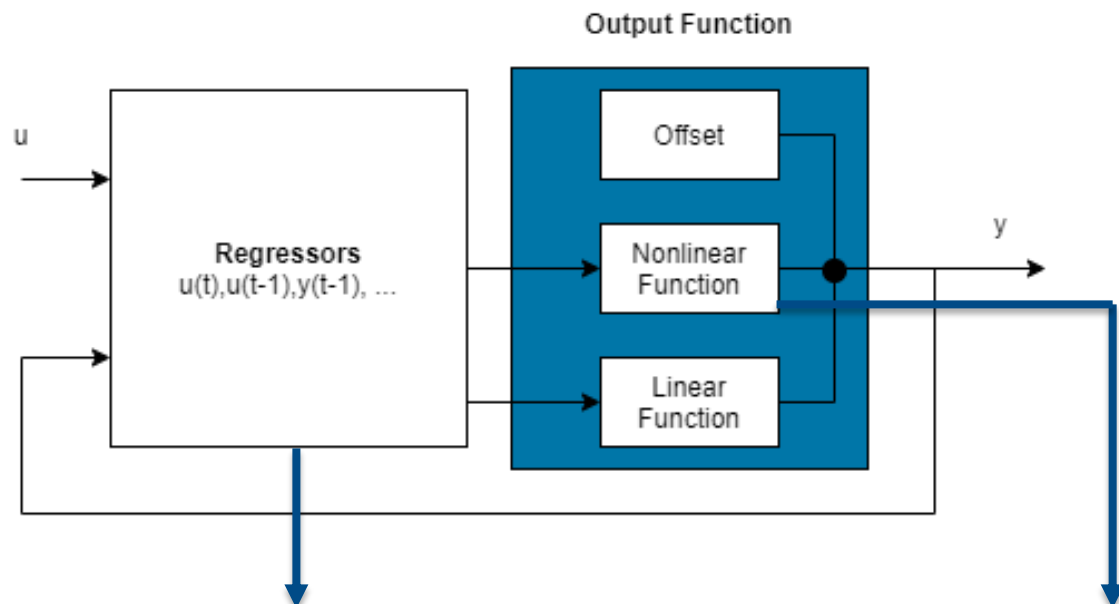
Show training loss plot

▼ Display results

Show fit to estimation data  Show fit to validation data

▶ Show code

# Redukovaný model pomocou Nonlinear ARX



Rozširuje lineárne modely a modeluje komplexné nelineárne správanie pomocou flexibilných nelineárnych funkcií

- Linear regressors
  - Polynomial regressors
  - Periodic regressors
  - Custom regressors
- Ex:  $\max(\min(u(t-1), 100) - 100)$

- Wavelet Network
- Sigmoid Network
- .....
- Gaussian Process
- Support Vector Machine
- Regression Tree Ensemble

Všeobecné estimátory nelineárnych funkcií

Algoritmy strojového učenia na zachytenie nelinearít

Vyžaduje Statistics and Machine Learning Toolbox

Nonlinear ARX Model of SI Engine Torque Dynamics

## 1.2. Prepare estimation and validation data

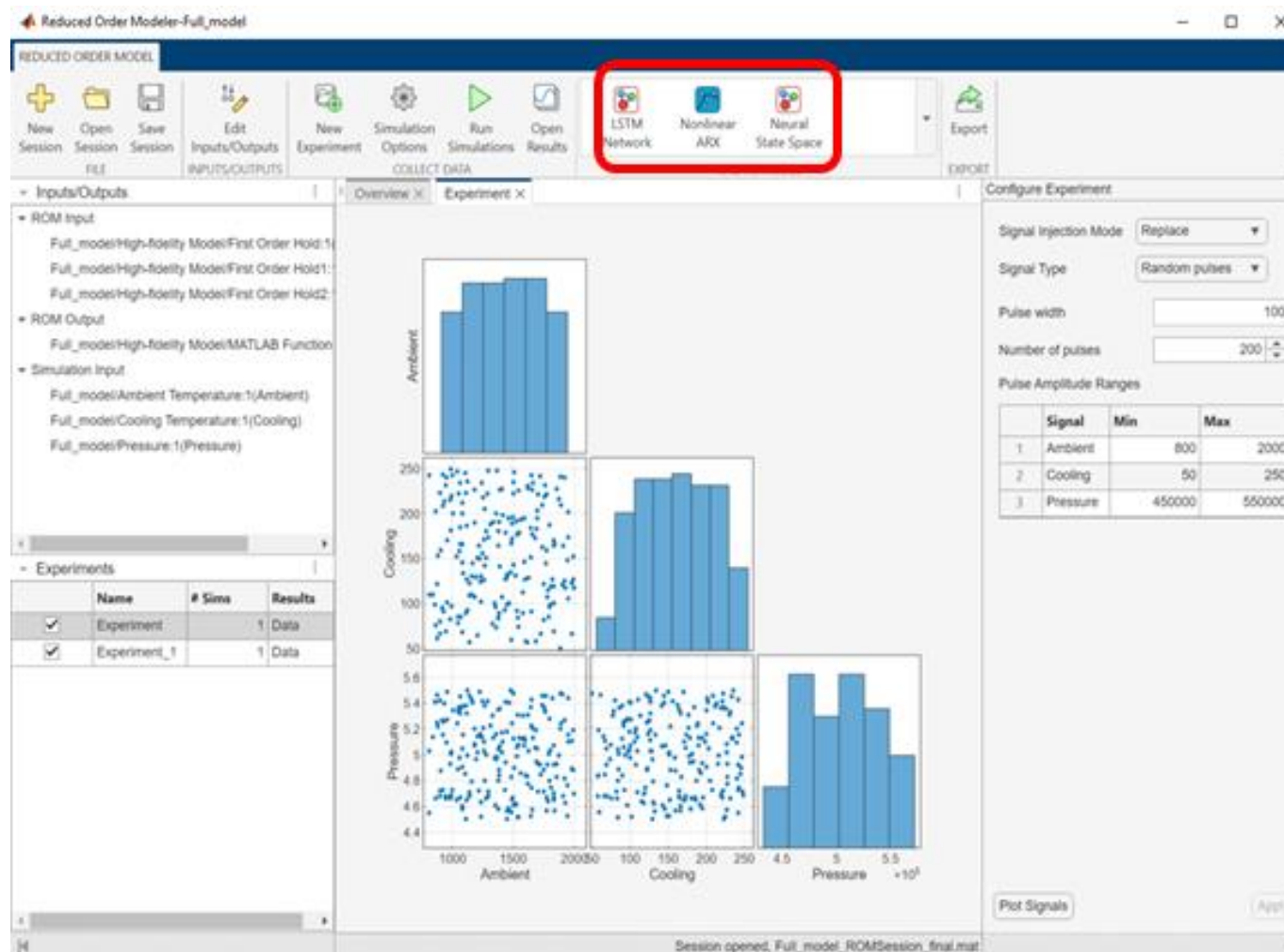
```

29 % Remove startup transient
30 nSamplesTransient = 500;
31 U(:,1:nSamplesTransient) = [];
32 Y(1:nSamplesTransient) = [];
33
34 holdOut = 0.2;
35 percentValidation = round(length(U)*holdOut);
36 UTrain = U(:,1:end-percentValidation);
37 UVal = U(:,end-percentValidation+1:end);
38
39 YTrain = Y(1:end-percentValidation);
40 YVal = Y(end-percentValidation+1:end);
41
42 meanX = mean(UTrain,2);
43 meanY = mean(YTrain,2);
44
45 stdX = std(UTrain,0,2);
46 stdY = std(YTrain,0,2);
47
48 normalize = @(x,mu,sigma) (x - mu) ./ sigma;
49
50 UTrainNormalized = normalize(UTrain,meanX,stdX);
51 YTrainNormalized = normalize(YTrain,meanY,stdY);
52
53 UValNormalized = normalize(UVal,meanX,stdX);
54 YValNormalized = normalize(YVal,meanY,stdY);
55
56 % Encapsulate input and output measurement data
57 zEst = iddata(YTrainNormalized,UTrainNormalized,'Ts',Ts, ...
58     'InputName',{'ThrottlePosition';'WastegateValve';'EngineSpeed';'SparkTiming'},...
59     'OutputName','EngineTorque');
60 zEst.Name = "EstimationData";
61
62 zVal = iddata(YValNormalized,UValNormalized,'Ts',Ts, ...
63     'InputName',{'ThrottlePosition';'WastegateValve';'EngineSpeed';'SparkTiming'},...
64     'OutputName','EngineTorque');
65 zVal.Name = "ValidationData";

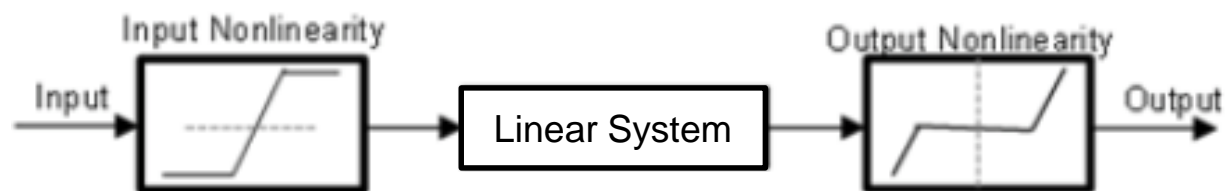
```

# Reduced Order Modeling Support Package (beta)

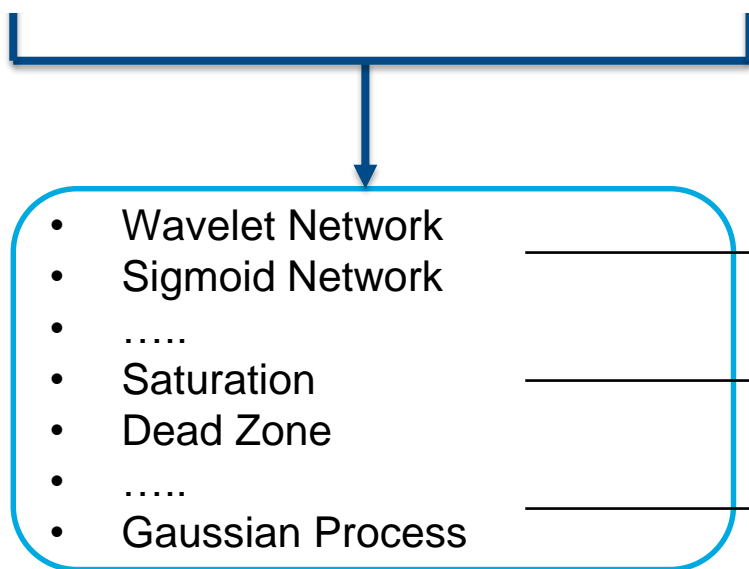
- Návrh a spustenie experimentov, tréovanie modelov, optimalizácia



# Redukovaný model pomocou Hammerstein Wiener modelov



Rozširuje lineárne modely sériovým zapojením statických nelineárnych modifikátorov



Všeobecné estimátory nelineárnych funkcií

Fyzikálne inšpirované nelineárne estimátory

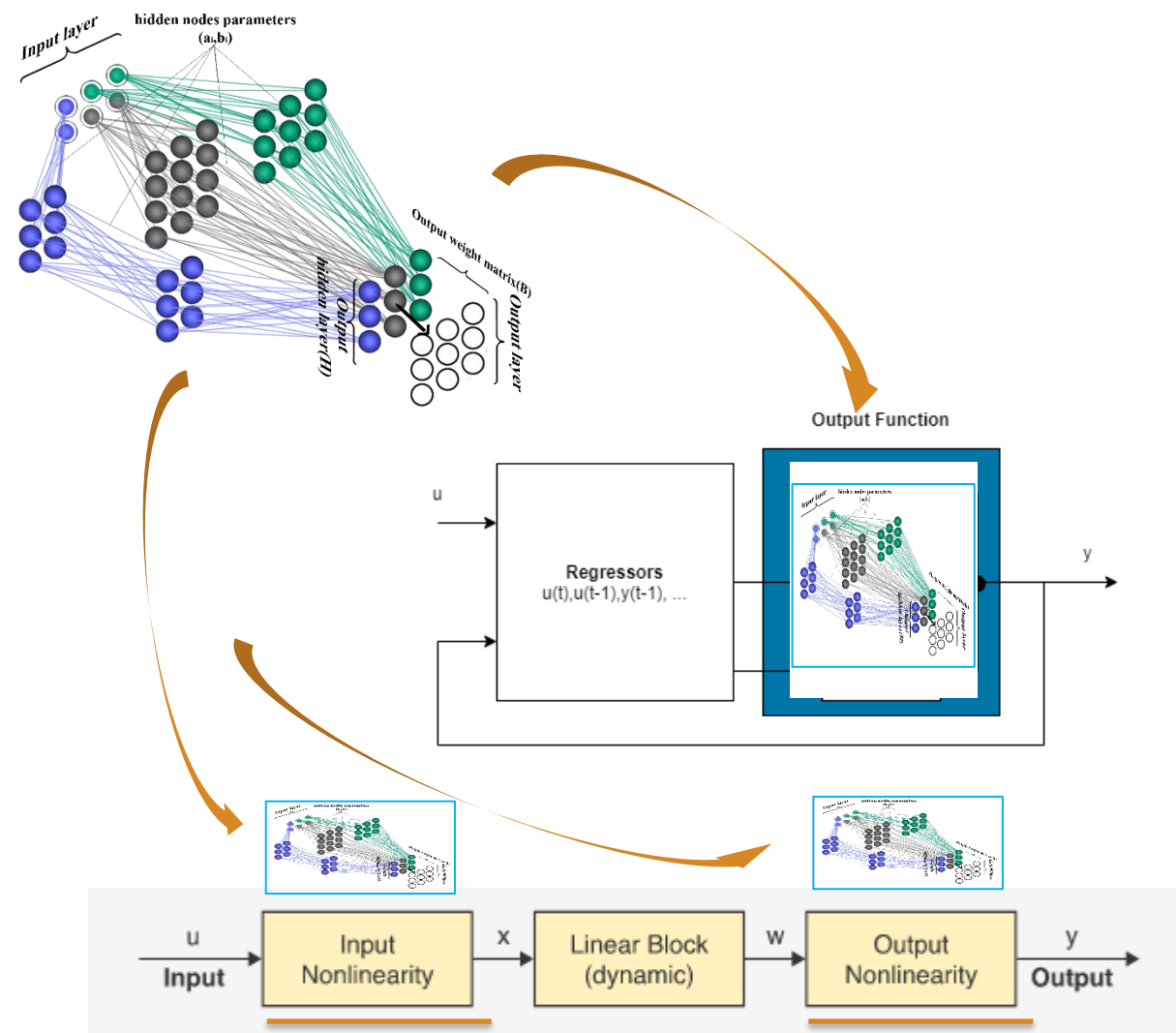
Algoritmy strojového učenia na zachytenie nelinearít

↓  
Vyžaduje **Statistics and Machine Learning Toolbox**

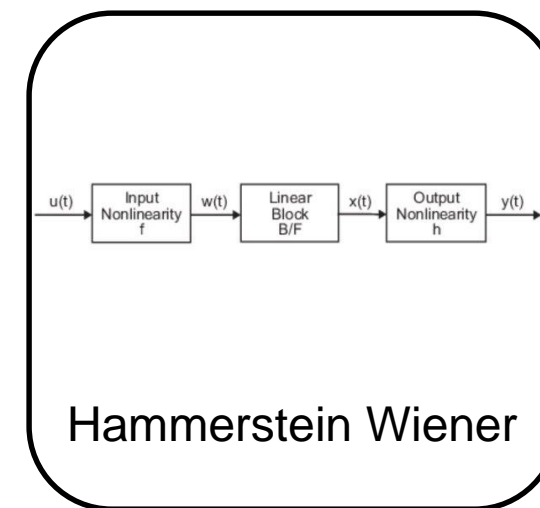
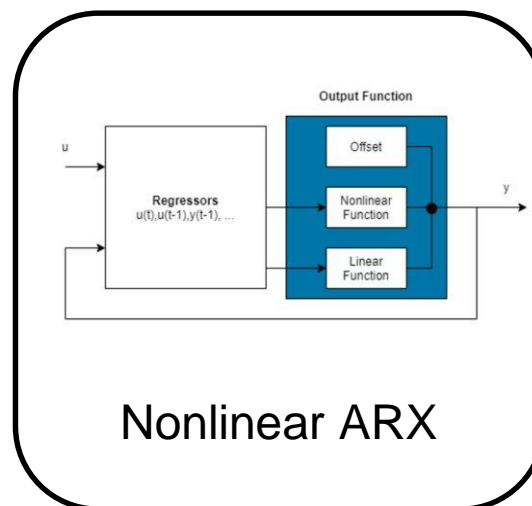
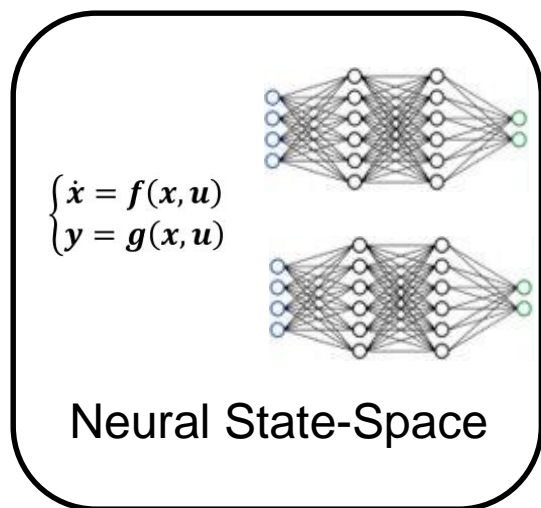
Hammerstein-Wiener Model of SI Engine Torque Dynamics

# Neurónové siete na reprezentáciu nelinearít

- Využitie neurónových sietí na tvorbu modelov Nonlinear ARX a Hammerstein-Wiener
- Neurónové siete
  - Deep Learning Toolbox (@dlnetwork, @network)
  - Statistics and Machine Learning Toolbox (@RegressionNeuralNetwork, fitrnet)
- Kombinácia neurónovej siete s lineárnymi komponentami založenými na fyzikálnych znalostiach



# Identifikácia nelineárnych systémov pomocou AI



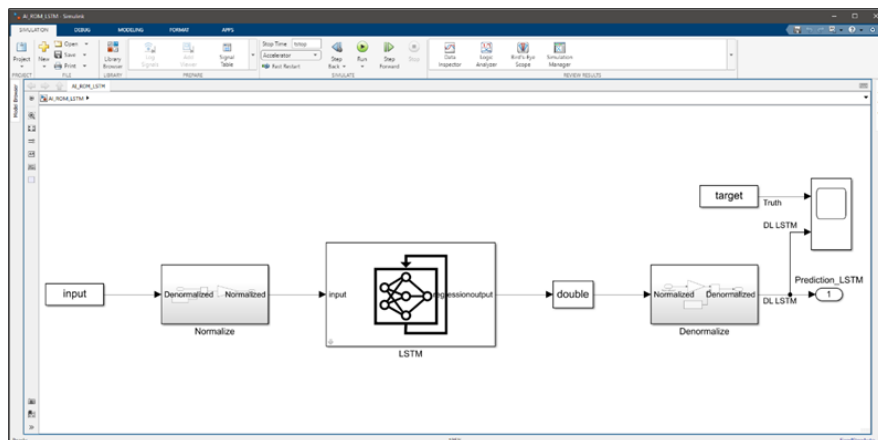
Využitie prístupov AI bez znalostí z oblasti AI

Kombinácia znalostí a fyzikálnej podstaty systému s prístupmi AI

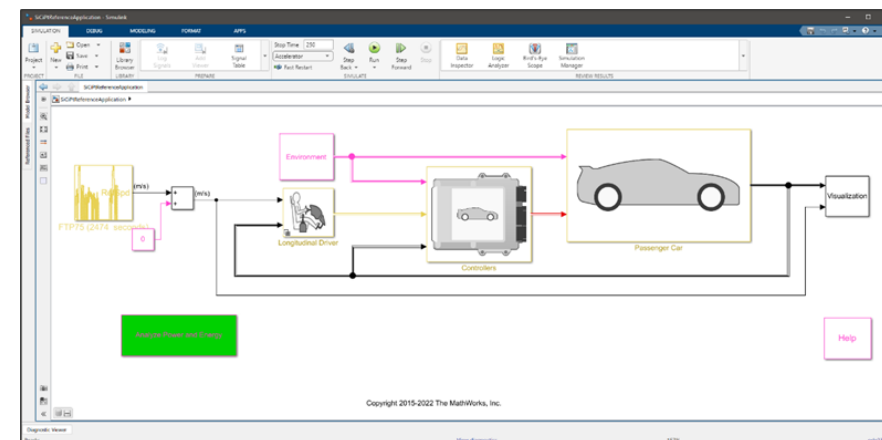


# Integrácia modelu AI pre simuláciu a testovanie

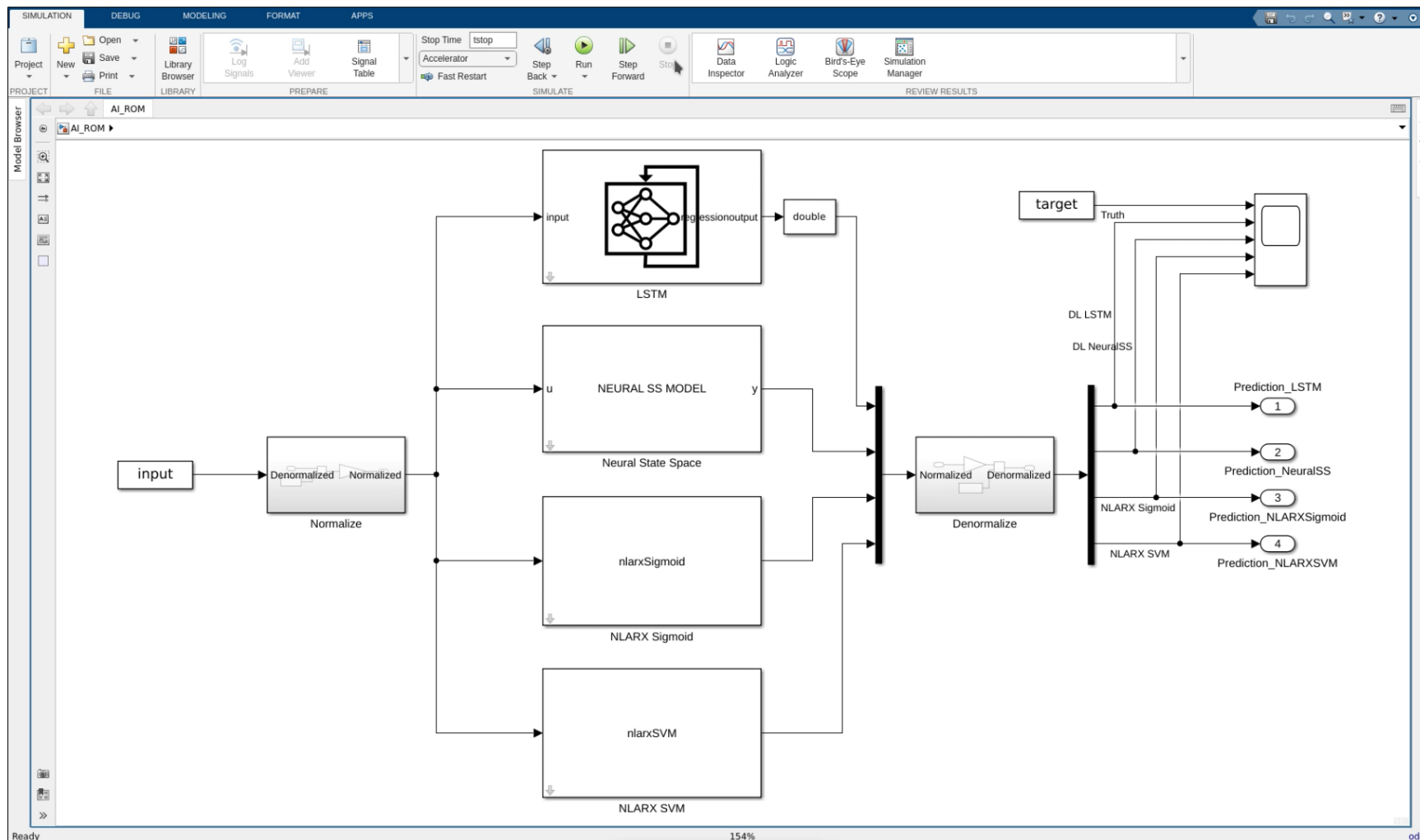
## Integrácia modelu AI do Simulinku





## Simulácia na systémovej úrovni



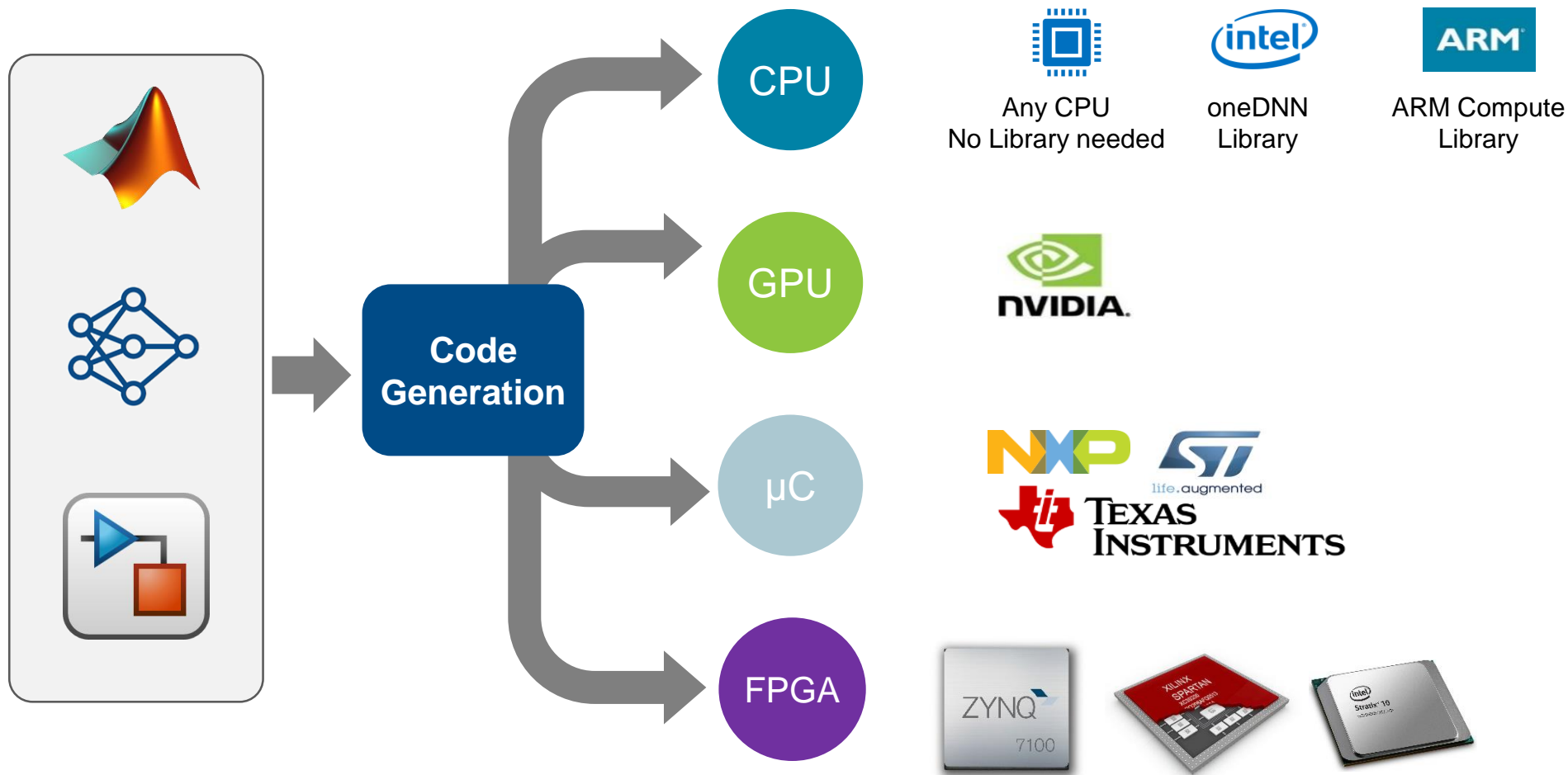
# Príklad – Integrácia modelu AI do Simulinku























# Simulink Profiler

Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
AI_ROM		49.440	45.732	142760
> LSTM		2.643	0.000	0
> NLARX Sigmoid		0.284	0.000	0
> Neural State Space		0.195	0.000	0
Scope		0.188	0.188	23795
From Workspace2		0.161	0.161	23794
Demux		0.128	0.128	95184
From Workspace1		0.054	0.054	23794
Prediction_LSTM		0.040	0.040	23794
Prediction_NeuralSS		0.006	0.006	23794
Prediction_NLARXSigmoid		0.005	0.005	23794
Prediction_NLARXSVM		0.004	0.004	23794
> NLARX SVM		0.001	0.000	0
> Normalize		0.000	0.000	0
Cast To Double		0.000	0.000	3
> Denormalize		0.000	0.000	0

# Nasadenie algoritmov generovaním kódu




# Kompromisy návrhu modelu AI

	<b>LSTM</b> Long Short-Term Memory Network	<b>Neural SS</b> Neural State Space (Neural ODE)	<b>NLARX Sig</b> Nonlinear ARX Sigmoid	<b>NLARX SVM</b> Nonlinear ARX Support Vector Machine
Rýchlosť tréovania	 *			
Interpretovateľnosť				
Rýchlosť inferencie				
Veľkosť modelu				
Presnosť (RSME)				

Výsledky vzhľadom na ROM model z príkladu



\*  ak sa trénuje pomocou GPU. Testované s GPU NVIDIA A100

Ďakujem za pozornosť