

Regionální inovační centrum elektrotechniky  
Fakulta elektrotechnická  
Západočeská univerzita v Plzni

## Modelově orientovaný návrh a prototypování aplikací v prostředí Matlab-Simulink

*Mikroprocesorové řízení pohonů 2*

Jakub Talla

## 1) Matlab a Simulink pro modelově orientované řízení

- ▶ Matlab a Simulink, funkce a použití
- ▶ Rapid prototyping SW v Matlab a Simulink

## 2) Jak konkrétně může vypadat modelově orientovaný návrh?

- ▶ V- diagram vývoje pohonu
- ▶ Návrh a testování řídicího embedded systému (HW, SW) pohonu

# Matlab a Simulink

## MATLAB A SIMULINK

- Nejpoužívanější SW pro modelově orientovaný návrh (Model-based design) je program MATLAB s grafickou nástavbou pro návrh a simulaci dynamických systémů Simulink s možností integrace celé řady knihoven pro řešení konkrétních typů úloh. (Další používaný SW například LabVIEW...).
- Matlab i Simulink jsou otevřené platformy umožňující tvorbu vlastních funkcí i implementaci funkcí (bloků) jiných firem (např. PLECS).

## Modelově orientovaný návrh umožňuje:

- Vývoj produktu (výrobku) různými projektovými týmy (HW, SW...) ve společném prostředí
- Transfer požadavků (zákazníka, normy) přímo do návrhu (modelu)
- Vývoj pomocí multi-doménových simulací (klasické dif. rovnice, stavové automaty, elektrické obvody, magnetické obvody, mechanika, teplo, konečné prvky, import 3d modelů apod.)
- Automaticky vygenerovat software (zdrojový kód C/C++, VHDL/Verilog) a dokumentaci
- Vývoj testovacích sad a procedur (MIL, SIL, PIL, HIL a další)
  
- Matlab a Simulink přirozeně implementuje V-diagram vývoje
- Podporuje řadu standardů bezpečnosti ASIL (Automotive Safety Integrity Levels): např. ISO 26262,
- Podpora AUTOSAR (AUTomotive Open System ARchitecture)

## MATLAB

MATLAB je inženýrský nástroj a interaktivní prostředí pro vědecké a technické výpočty, analýzu dat, vizualizaci a vývoj algoritmů. Jazyk MATLABu podporuje vektorové a maticové operace, které jsou zásadní pro řešení technických a vědeckých problémů. Příkazy mohou být spouštěny jeden po druhém a okamžitě vracet výsledky. To Vám umožní zkoumat více přístupů a dospět k optimálnímu řešení. Pro automatizaci a opakované využití Vaší práce lze v MATLABu vytvářet skripty a funkce. Lze tak budovat i komplexní programy a aplikace. Vývojové nástroje usnadní efektivní implementaci Vašich algoritmů a optimalizaci jejich výpočetního výkonu.

## Aplikační knihovny

aplikační knihovny, toolboxy, rozšiřují prostředí MATLABu o řešení úloh z nejrůznějších oblastí, jakými jsou:

- matematické výpočty, statistika a optimalizace
- návrh a analýza řídicích systémů
- zpracování signálu a komunikace
- zpracování obrazu a videa
- měření a testování
- finanční analýza a modelování
- výpočetní biologie
- modelování fyzikálních soustav

## SIMULINK

Simulink je nadstavba MATLABu pro simulaci a modelování dynamických systémů. Poskytuje uživateli možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat.

Modely mohou být popsány rovnicemi nebo mohou být sestavené z bloků reprezentujících prvky reálných systémů. Kromě modelů fyzikálních soustav je možné modelovat také algoritmy řídicích systémů včetně jejich automatického ladění, systémy pro zpracování signálu, komunikace a zpracování obrazu.

## Simulink a model based design

Středem návrhového procesu Model-Based Design je model systému v prostředí Simulink. Model je využíván od definice požadavků na finální zařízení, přes návrh dynamického systému a algoritmů, až po implementaci na cílovou platformu a testování prototypu. Implementace se opírá o automatické generování kódu v jazyce C nebo HDL z vytvořených modelů.

# Matlab + Simulink pro rapid-prototyping SW

MATLAB <sup>®</sup> PRODUCT FAMILY	SIMULINK <sup>®</sup> PRODUCT FAMILY
<b>MATLAB</b>	<b>Simulink</b>
<b>Parallel Computing</b> Parallel Computing Toolbox MATLAB Parallel Server	<b>Systems Engineering</b> System Composer
<b>Math and Optimization</b> Curve Fitting Toolbox Optimization Toolbox Global Optimization Toolbox Symbolic Math Toolbox Mapping Toolbox Partial Differential Equation Toolbox	<b>Event-Based Modeling</b> <b>Stateflow</b> Stateflow
<b>AI, Data Science, and Statistics</b> Statistics and Machine Learning Toolbox Deep Learning Toolbox Reinforcement Learning Toolbox Deep Learning HDL Toolbox Text Analytics Toolbox Predictive Maintenance Toolbox	<b>Physical Modeling</b> Simscape Simscape Driveline Simscape Electrical Simscape Fluids Simscape Multibody
<b>Code Generation</b> <b>MATLAB Coder</b> Embedded Coder HDL Coder HDL Verifier Filter Design HDL Coder Fixed-Point Designer GPU Coder	<b>Real-Time Simulation and Testing</b> Simulink Real-Time Simulink Desktop Real-Time
<b>Application Deployment</b> MATLAB Compiler MATLAB Compiler SDK MATLAB Production Server MATLAB Web App Server	<b>Code Generation</b> <b>Simulink Coder</b> <b>Embedded Coder</b> AUTOSAR Blockset Fixed-Point Designer Simulink PLC Coder Simulink Code Inspector DO Qualification Kit (for DO-178) IEC Certification Kit (for ISO 26262 and IEC 61508) HDL Coder HDL Verifier
<b>Database Access and Reporting</b> Database Toolbox MATLAB Report Generator	<b>Application Deployment</b> Simulink Compiler
	<b>Verification, Validation, and Test</b> Simulink Requirements Simulink Check Simulink Coverage Simulink Design Verifier Simulink Test Polyspace Bug Finder Polyspace Code Prover
	<b>Simulation Graphics and Reporting</b> Simulink 3D Animation Simulink Report Generator

# Matlab + Simulink pro rapid-prototyping SW

## Produkty firmy Mathworks+řada dalších např. PLECS

### Signal Processing

[Signal Processing Toolbox](#)  
[Phased Array System Toolbox](#)  
[DSP System Toolbox](#)  
[Audio Toolbox](#)  
[Wavelet Toolbox](#)

### Image Processing and Computer Vision

[Image Processing Toolbox](#)  
[Computer Vision Toolbox](#)  
[Lidar Toolbox](#)  
[Vision HDL Toolbox](#)

### Control Systems

[Control System Toolbox](#)  
[System Identification Toolbox](#)  
[Predictive Maintenance Toolbox](#)  
[Robust Control Toolbox](#)  
[Model Predictive Control Toolbox](#)  
[Fuzzy Logic Toolbox](#)  
[Simulink Control Design](#)  
[Simulink Design Optimization](#)  
[Reinforcement Learning Toolbox](#)  
[Motor Control Blockset](#)

### Test and Measurement

[Data Acquisition Toolbox](#)  
[Instrument Control Toolbox](#)  
[Image Acquisition Toolbox](#)  
[OPC Toolbox](#)  
[Vehicle Network Toolbox](#)  
[ThingSpeak](#)

### RF and Mixed Signal

[Antenna Toolbox](#)  
[RF Toolbox](#)  
[RF Blockset](#)  
[Mixed-Signal Blockset](#)  
[SerDes Toolbox](#)

### Wireless Communications

[Communications Toolbox](#)  
[WLAN Toolbox](#)  
[LTE Toolbox](#)  
[5G Toolbox](#)  
[Wireless HDL Toolbox](#)

### Autonomous Systems

[Automated Driving Toolbox](#)  
[Robotics System Toolbox](#)  
[UAV Toolbox](#)  
[Navigation Toolbox](#)  
[ROS Toolbox](#)  
[Sensor Fusion and Tracking Toolbox](#)  
[RoadRunner](#)  
[RoadRunner Asset Library](#)  
[RoadRunner Scene Builder](#)

### FPGA, ASIC, and SoC Development

[HDL Coder](#)  
[HDL Verifier](#)  
[Deep Learning HDL Toolbox](#)  
[Wireless HDL Toolbox](#)  
[Vision HDL Toolbox](#)  
[Filter Design HDL Coder](#)  
[Fixed-Point Designer](#)  
[SoC Blockset](#)

### Automotive

[Model-Based Calibration Toolbox](#)  
[Powertrain Blockset](#)  
[Vehicle Dynamics Blockset](#)  
[Automated Driving Toolbox](#)  
[IEC Certification Kit \(for ISO 26262 and IEC 61508\)](#)  
[Vehicle Network Toolbox](#)  
[AUTOSAR Blockset](#)  
[RoadRunner](#)  
[RoadRunner Asset Library](#)  
[RoadRunner Scene Builder](#)

### Aerospace

[Aerospace Blockset](#)  
[Aerospace Toolbox](#)  
[UAV Toolbox](#)  
[DO Qualification Kit \(for DO-178\)](#)

### Computational Finance

[Econometrics Toolbox](#)  
[Financial Toolbox](#)  
[Datafeed Toolbox](#)  
[Database Toolbox](#)  
[Spreadsheet Link \(for Microsoft Excel\)](#)  
[Financial Instruments Toolbox](#)  
[Trading Toolbox](#)  
[Risk Management Toolbox](#)

### Computational Biology


[Bioinformatics Toolbox](#)  
[SimBiology](#)

### Code Verification

[Polyspace Bug Finder](#)  
[Polyspace Bug Finder Access](#)  
[Polyspace Bug Finder Server](#)  
[Polyspace Code Prover](#)  
[Polyspace Code Prover Access](#)  
[Polyspace Code Prover Server](#)  
[Polyspace Products for Ada](#)

# Matlab + Simulink pro rapid-prototyping SW


**HUMUSOFT**® Vždy aktuální ceny naleznete na  
[www.humusoft.cz/matlab/pricing](http://www.humusoft.cz/matlab/pricing)


**MathWorks**

**Ceník produktů systému MATLAB - individuální licence** platný od 1.1.2021  
 Časově neomezené licence, ceny zahrnují předplatné nových verzí a technickou podporu na 12 měsíců!

Název produktu	Verze	Standard	Školní
<b>MATLAB</b>	9.9	<b>54 980,-</b>	<b>13 980,-</b>
<b>SIMULINK</b>	10.2	<b>82 980,-</b>	<b>13 980,-</b>
<b>STATEFLOW</b>	10.3	<b>78 980,-</b>	<b>13 980,-</b>

## Volitelné nadstavby systémů MATLAB a Simulink:

### Analýza dat, statistika a optimalizace

Statistics and Machine Learning Toolbox	12.0	27 980,-	5 580,-
Curve Fitting Toolbox	3.5	27 980,-	5 580,-
Optimization Toolbox	9.0	31 980,-	5 580,-
Global Optimization Tbx (vyžaduje Optimization Tbx)	4.4	27 980,-	5 580,-
Deep Learning Toolbox	14.1	31 980,-	5 580,-
<b>Deep Learning HDL Toolbox (vyžaduje Deep Learning Toolbox)</b>	<b>1.0</b>	<b>137 980,-</b>	<b>13 980,-</b>
Text Analytics Toolbox (vyžaduje Statistics and Machine Learning Tbx)	1.6	31 980,-	5 580,-
Bioinformatics Toolbox (vyžaduje Statistics and Machine Learning Tbx)	4.15	27 980,-	5 580,-
SimBiology	6.0	82 980,-	13 980,-
Partial Differential Equation Toolbox	3.5	31 980,-	5 580,-
Symbolic Math Toolbox	8.6	27 980,-	5 580,-
Mapping Toolbox	5.0	27 980,-	5 580,-

# Matlab + Simulink pro rapid-prototyping SW

## Automatické generování kódu, Rapid Prototyping a HIL simulace

MATLAB Coder	5.1	164 980,-	13 980,-
Simulink Coder (vyžaduje Simulink a MATLAB Coder)	9.4	82 980,-	13 980,-
Embedded Coder (vyžaduje MATLAB Coder)	7.5	137 980,-	13 980,-
AUTOSAR Blockset (vyžaduje Simulink)	2.3	54 980,-	5 580,-
GPU Coder (vyžaduje MATLAB Coder a Parallel Computing Tbx)	2.0	137 980,-	13 980,-
HDL Coder (vyžaduje MATLAB Coder a Fixed-Point Designer)	3.17	261 980,-	13 980,-
Simulink PLC Coder (vyžaduje Simulink, pouze pro Win)	3.3	261 980,-	13 980,-
Simulink Desktop Real-Time (vyž. SL a SL Coder pro Ext.M., jen Win a Mac)	5.11	54 980,-	13 980,-
Simulink Real-Time (vyž. MATLAB Coder a Simulink Coder, pouze Win)	7.0	137 980,-	13 980,-

## Zpracování signálu a obrazu

Signal Processing Toolbox	8.5	27 980,-	5 580,-
DSP System Toolbox (vyžaduje Signal Processing Tbx)	9.11	34 980,-	5 580,-

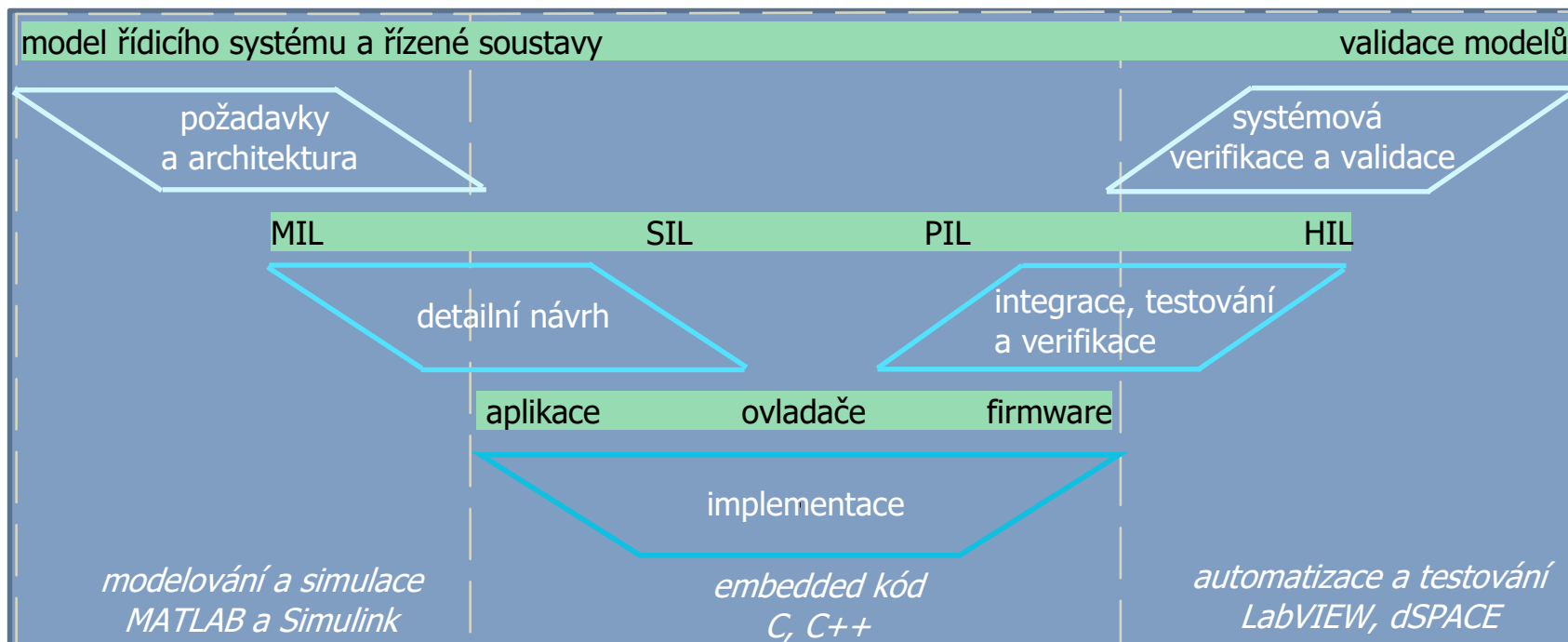
## Návrhy řídicích systémů

Control System Toolbox	10.9	31 980,-	5 580,-
------------------------	------	----------	---------

# Modelově orientovaný návrh v prostředí Matlab a Simulink

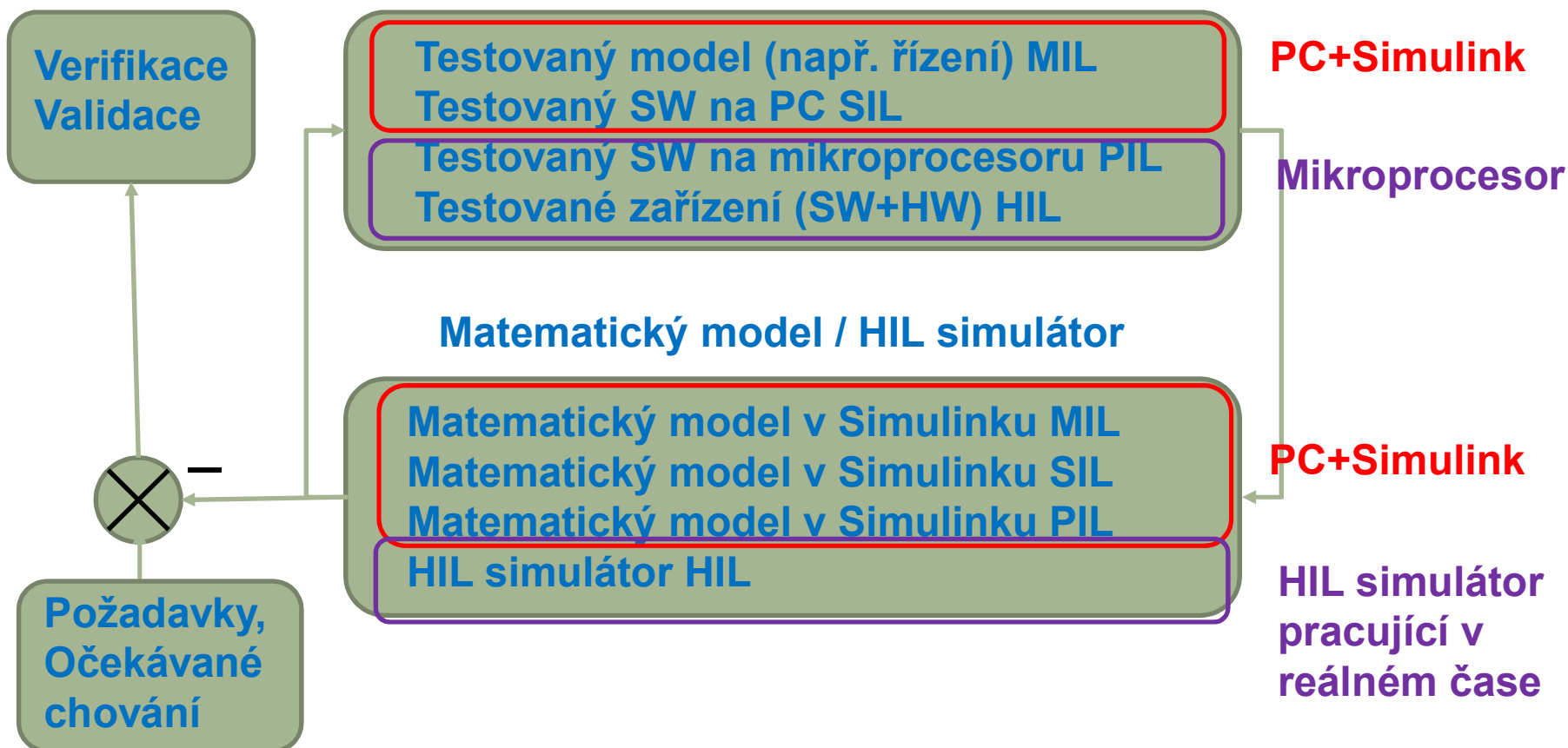
## V-diagram modelově orientovaného vývoje v prostředí Matlab+Simulink

- **MIL** = Model In the Loop – Model ve smyčce
- **SIL** = Software In the Loop – Software ve smyčce (pozor může znamenat i Safety Integrity Level)
- **PIL** = Processor In the Loop – Procesor ve smyčce
- **HIL** = Hardware In the Loop – Hardware ve smyčce



**Základy testování ve smyčkách – In the Loop testing**  
 (Testy mohou probíhat v mnoha cyklech pro různé funkční celky)

**D.U.T. (Device Under Test) Testovaný objekt**



## Základní porovnání metod SIL, PIL a HIL pro verifikaci a validaci algoritmu navrženého metodou MIL

Parametr / faktor	SIL	PIL	HIL
účel	verifikovat zdrojový kód	verifikovat objektový kód	verifikovat systémovou funkcionalitu
reálný čas (RT)	není RT	je RT v rámci simulačního kroku	plně RT
použití a prostředí (řídící systém)	bez cílového HW, pouze Simulink na PC	cílový procesor	cílová elektronická jednotka
řízená soustava	model v Simulinku na PC (-x86/64)	model v Simulinku na PC (-x86/64)	model v reálném čase (např. na procesoru nebo HIL test stand)
věrnost a přesnost oproti finálnímu produktu	různé bitové šířky, různá přesnost (fixed a floating point aritmetika)	stejná bitová šířka, stejná přesnost, měřené hodnoty simulovány	stejná bitová šířka, stejná přesnost, reálné měření

# Modelově orientovaný návrh v prostředí Matlab a Simulink – příklad

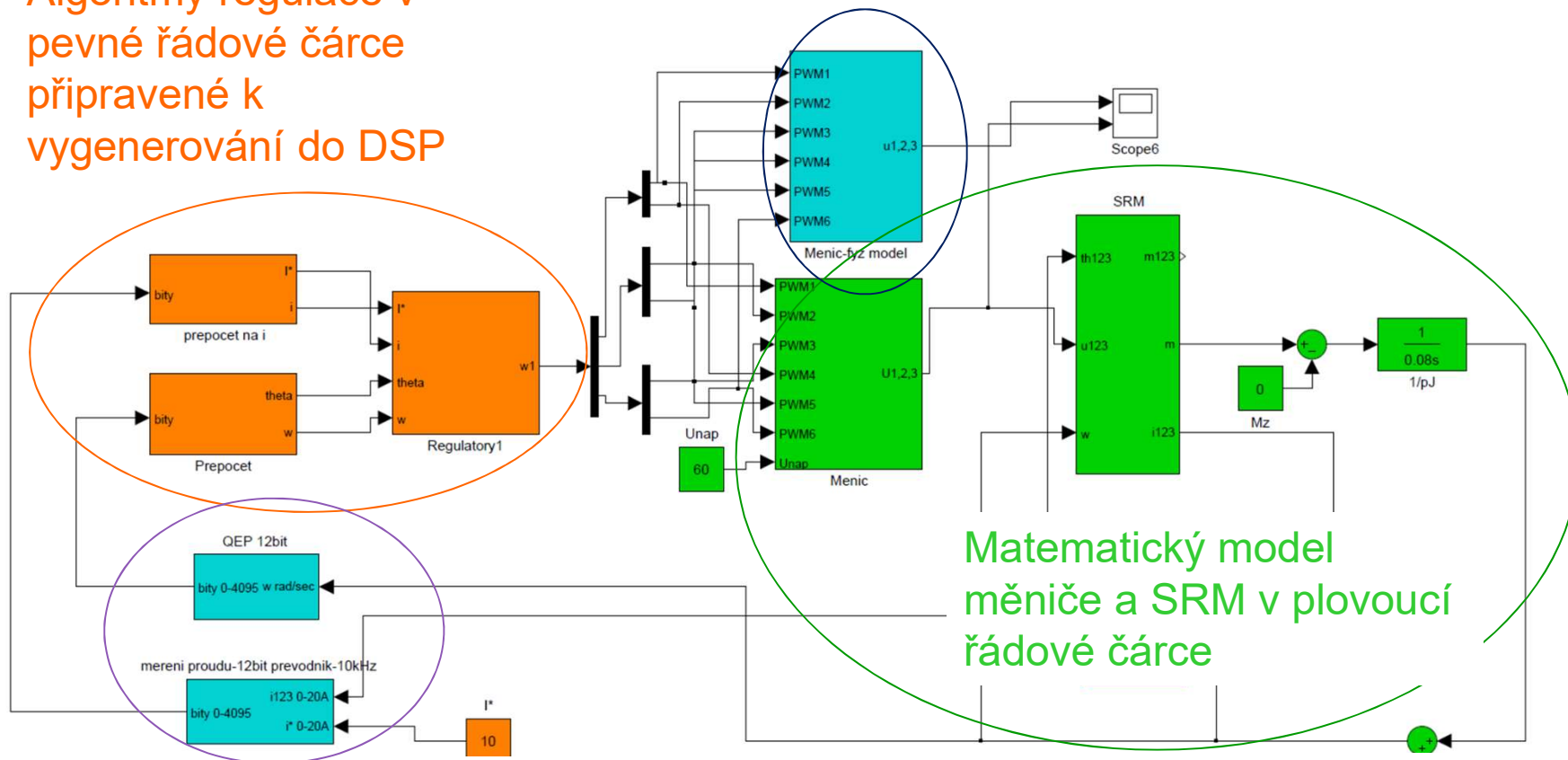
## Možné zadání (požadavky):

Vyvinout pohon se spínaným reluktančním motorem o daných parametrech (požadavky na moment, otáčky, rozměry, napětí měniče.....cenu),

Podúkoly: návrh (výběr) motoru, návrh (výběr) výkonového měniče (topologie, prvky...), vývoj řídicí platformy, (vývoj DPS, výběr mikroprocesoru, čidel), vývoj řídicích algoritmů a SW

Algoritmy regulace v pevné řádové čarce připravené k vygenerování do DSP

Model výkonového měniče



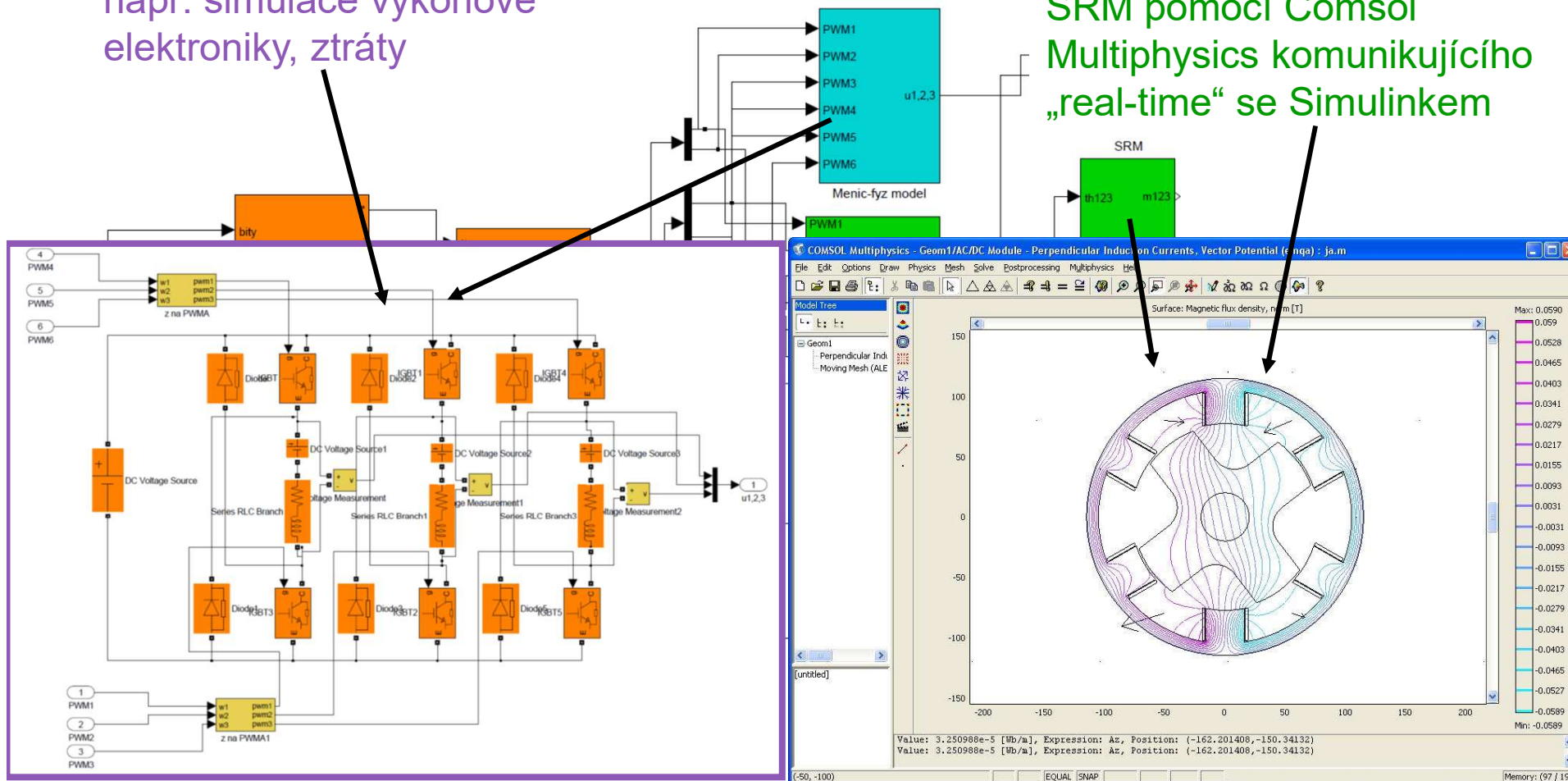
Matematický model vstupních HW periférií mikrokontroléru

Vytvoření matematických modelů řízených systémů , návrh algoritmů a jejich simulace

## Modelově orientovaný návrh příklad – vývoj

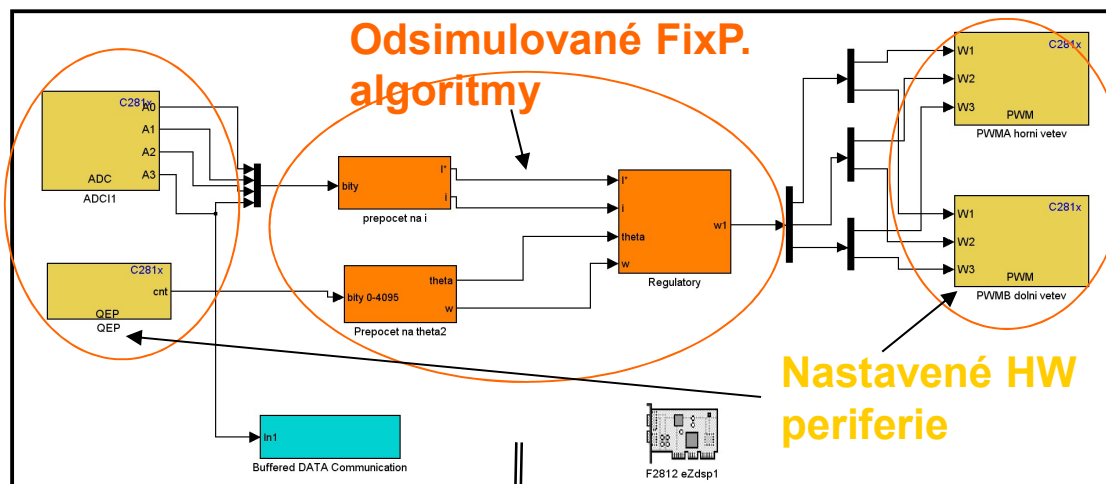
Aplikační blocksety – např. simulace výkonové elektroniky, ztráty

Linky – FEM modelování SRM pomocí Comsol Multiphysics komunikujícího „real-time“ se Simulinkem



**Ukázka vzájemné provázanosti systémů, možnost implementace FEM multifyzikálních úloh do prostředí Simulinku, vytváření komplexních systémů**

# Modelově orientovaný návrh příklad – vývoj



**Další možné výstupy:**

**Podklady pro návrh či nákup motoru**

**Podklady pro návrh měniče**

**Podklady pro návrh DPS**

.....

```

1  File: c2812adcpwmtest_main.c
2  * Real-Time Workshop code generated for Simulink model c2812adcpwmtest.
3
4  * Model Version : 160
5  * Real-Time Workshop file version : 2 (R2008b) 04-Aug-2008
6  * Real-Time Workshop file generated on : Wed Feb 18 07:49:06 2009
7  * TIC version : 2 (Aug 5 2008)
8  * C/C++ source code generated on : Wed Feb 18 07:49:07 2009
9
10
11
12
13 #include "c2812adcpwmtest.h"
14 #include "rtwtypes.h"
15 #include "c2812adcpwmtest_private.h"
16 #include "c2000_main.h"
17 #include "DSP281x_Device.h"
18 #include "DSP281x_Examples.h"
19 #include <stdlib.h>
20 #include <stdio.h>
21
22 void in1_board(void);
23 void enable_interrupts(void);
24 void config_schedulerTimer(void);
25 void disable_interrupts(void);
26 volatile int isrOverrun = 0;
27 boolean_T isRateRunning[] = { 0, 0 };
28
29 boolean_T need2runFlags[] = { 0, 0 };
30
31 /* Function: rt_OneStep
32
33 * Abstract:
34 * Perform one step of the model. Multi-tasking implementation.
35 */
36 void rt_OneStep(void)
37 {
38     boolean_T eventFlags[];
39
40     /* Check base rate for overrun */
41     // disabling interrupts
42     asm(" SEC INTRM");
43     if (isRateRunning[0]++) {
44         isrOverrun = 1;
45         isRateRunning[0]--;
46         // allow future iterations to succeed
47     }
48     // enabling interrupts
49     asm(" CLRC INTRM");

```

**Okomentovaný vygenerovaný  
embedded C kód, provázaný  
hypertextem s modelem v Simulinku**

**Vygenerovaná dokumentace**

### V průběhu vytváření modelu průběžné testování MIL (Model In the Loop), postupné zpřesňování modelu

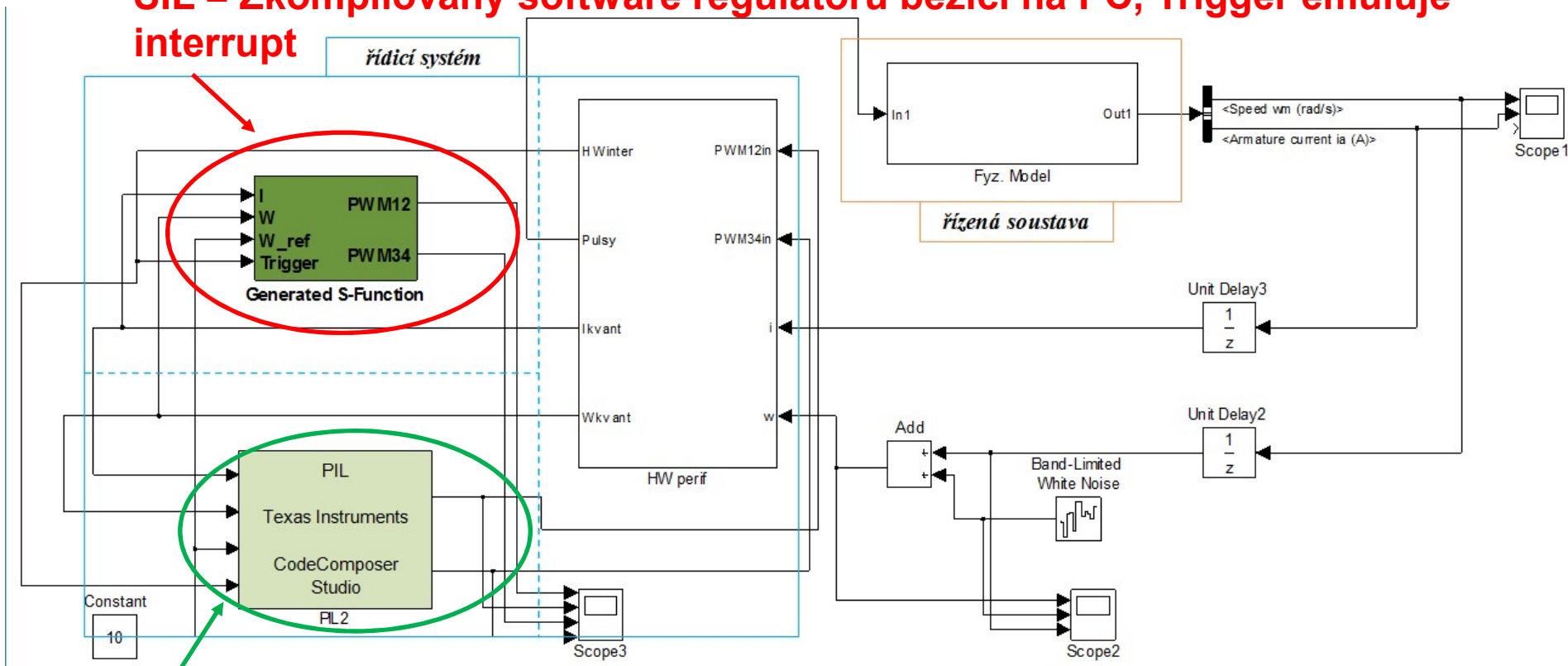
Model In the Loop – návrh SW PI regulátorů proudu a otáček

Některé kroky návrhu:

- lineární model v aritmetice plovoucí řádové čárky – řešení přenosových funkcí a návrh regulátoru pomocí frekvenčních charakteristik a jejich ověření
- diskretizace
- fyzikální omezení řídicího systému
- fyzikální model řízené soustavy z knihovny Simulinku – Power Electronics
- zavedení mrtvých časů (ochranu polovodičových prvků měniče)
- napěťové úbytky na polovodičových prvcích měniče
  
- finální model v aritmetice fixed point, uvažovány reálná zpoždění a reálné šумы – vytvořen blok SW
  - blok SW, který je beze změny použit také v metodách SIL, PIL a HIL a pro generování výsledné aplikace (SW produktu)

**SIL+PIL vygenerovaný SW je testován pomocí SIL a poté PIL**

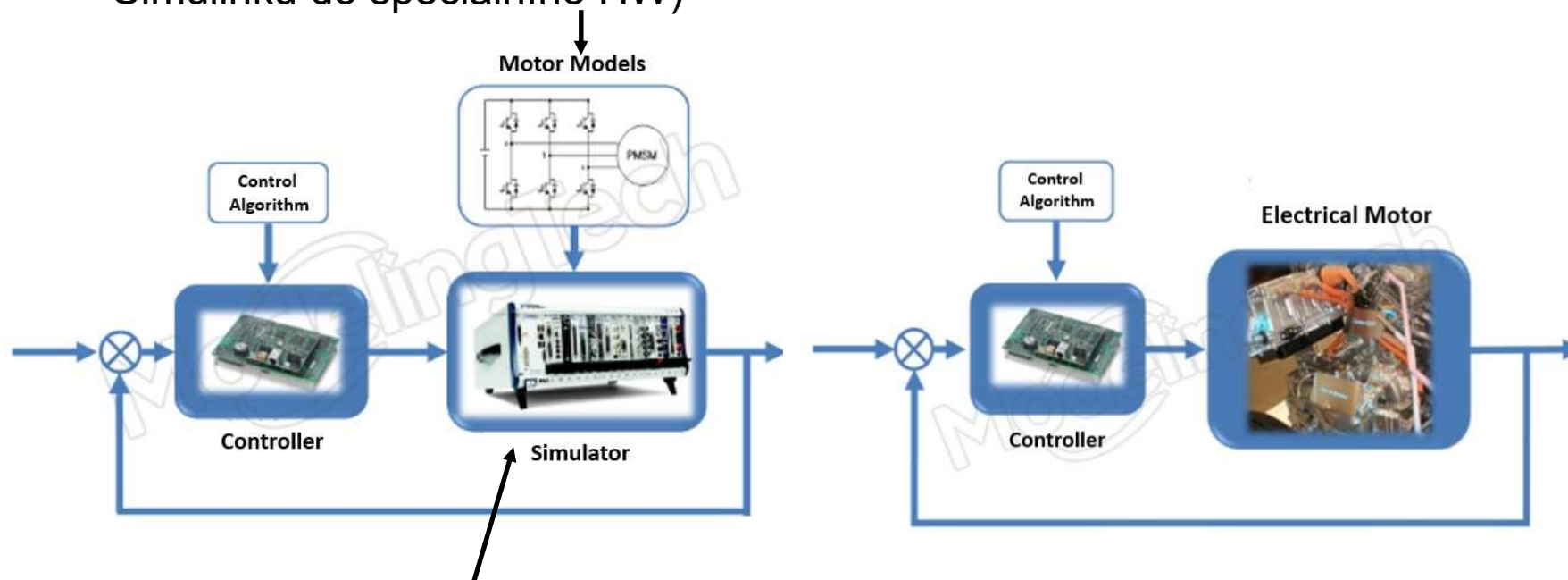
**SIL = Zkompilovaný software regulátoru běžící na PC, Trigger emuluje interrupt**



**PIL = Zkompilovaný software regulátoru běžící na mikrokontroléru, (komunikace přes JTAG)**

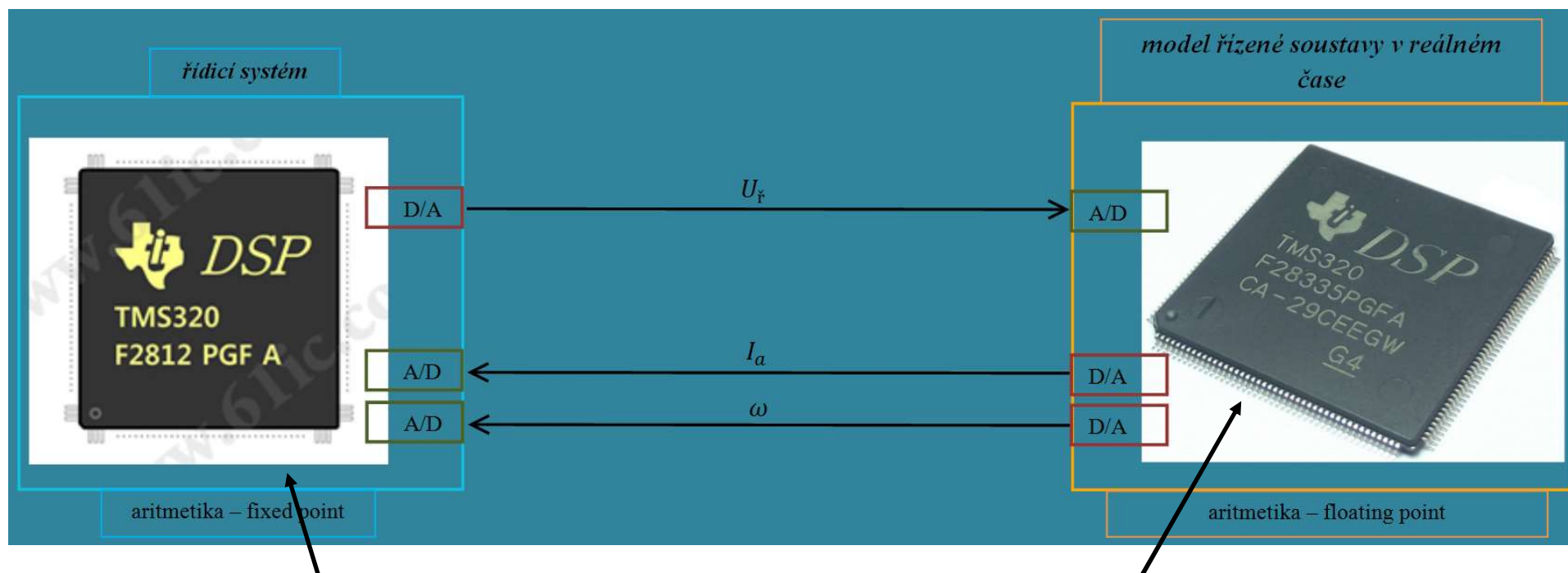
## HIL testování

Model pohonu běžící v reálném čase  
(nejčastěji vygenerovaný kód ze  
Simulinku do speciálního HW)



**HIL simulátor- nejčastěji dSPACE  
nebo NI, ale i řada dalších firem**

## HIL testování (levné řešení pro řídicí systém pohonu s DC motorem)

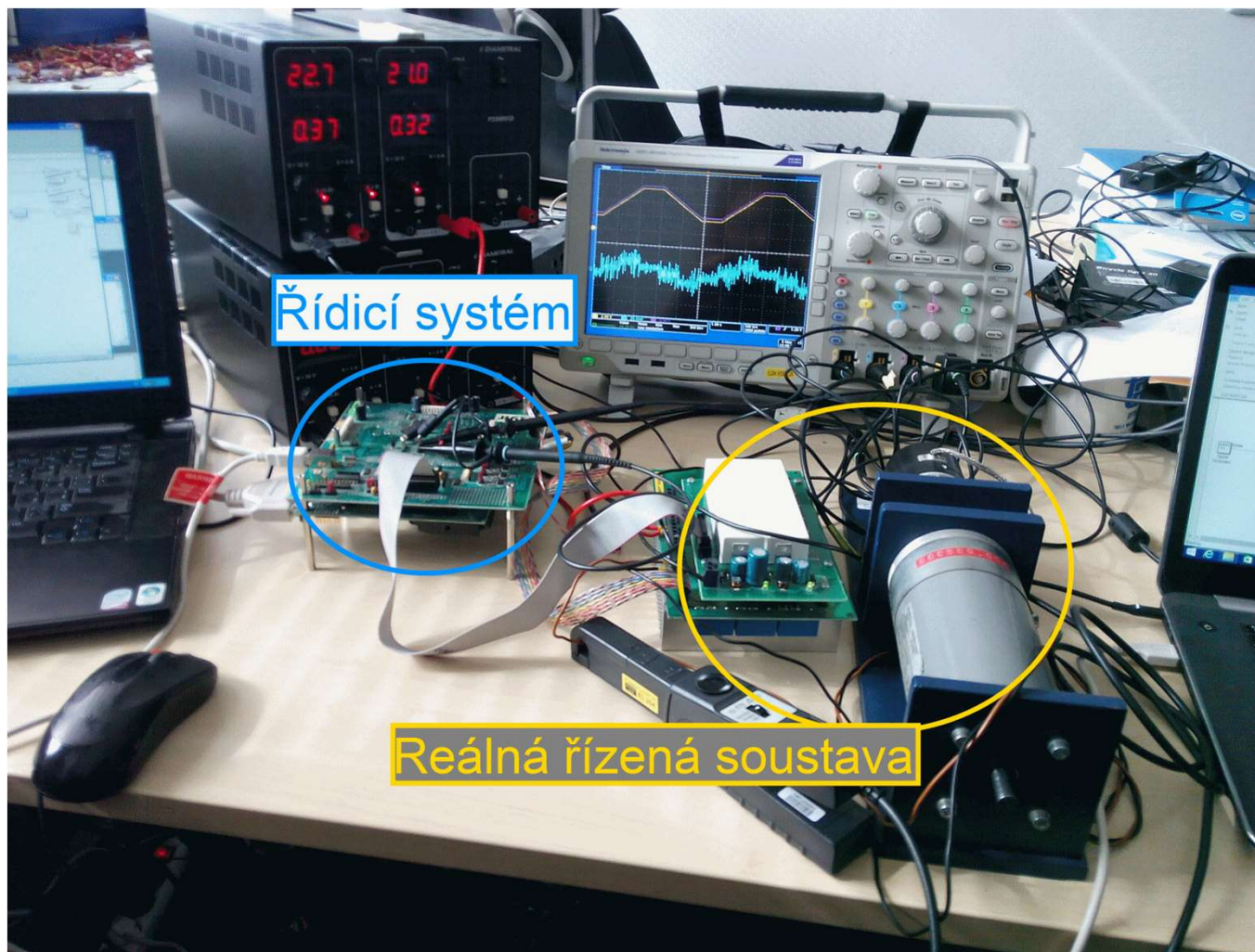


Navrhovaný řídicí embedded systém  
 - Algoritmy řízení pohonu DC motoru  
 v pevné řádové čárce vygenerované  
 ze Simulinku

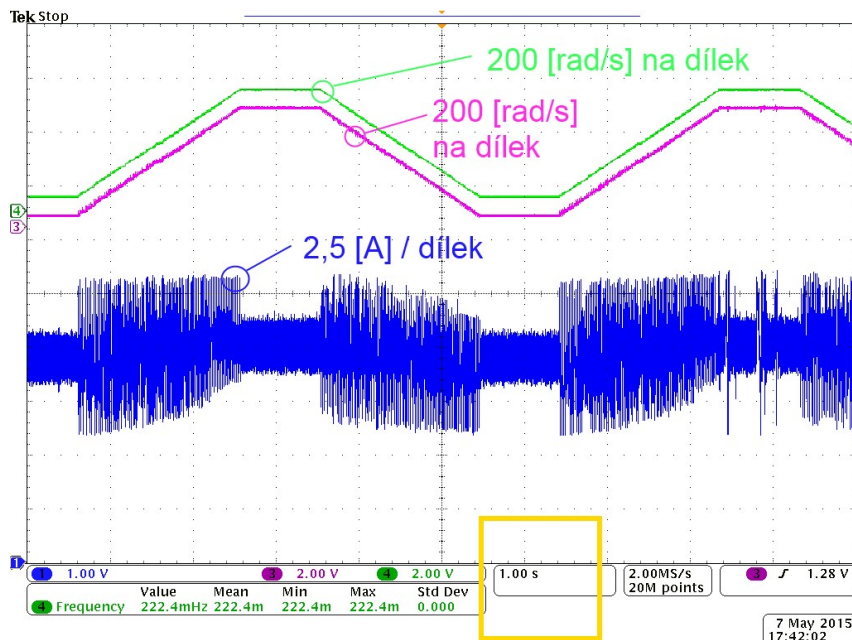
HIL simulátor - Model pohonu běžící  
 v reálném čase (vygenerovaný kód  
 modelu pohonu s DC motorem ze  
 Simulinku v plovoucí řádové čárce)



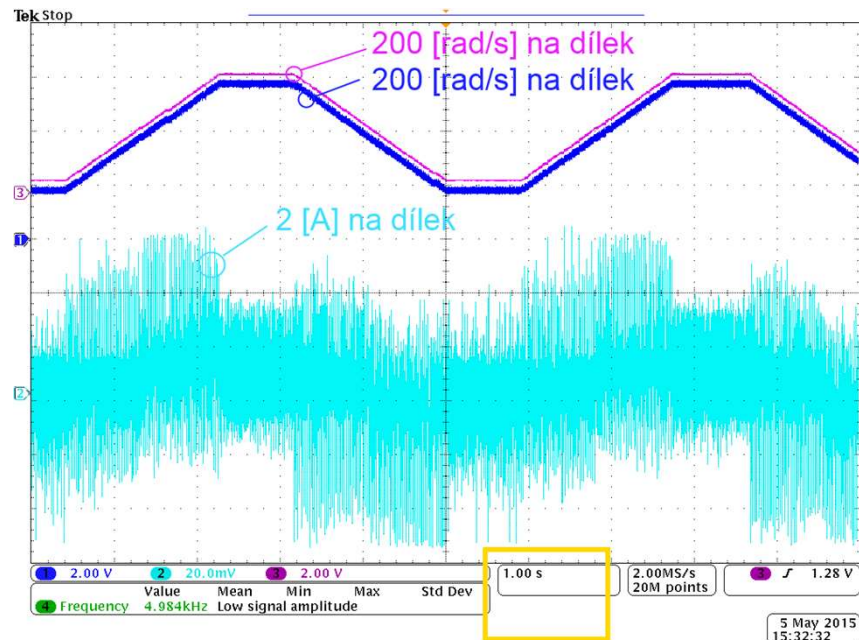
## Experiment s reálným pohonem



## HIL testování



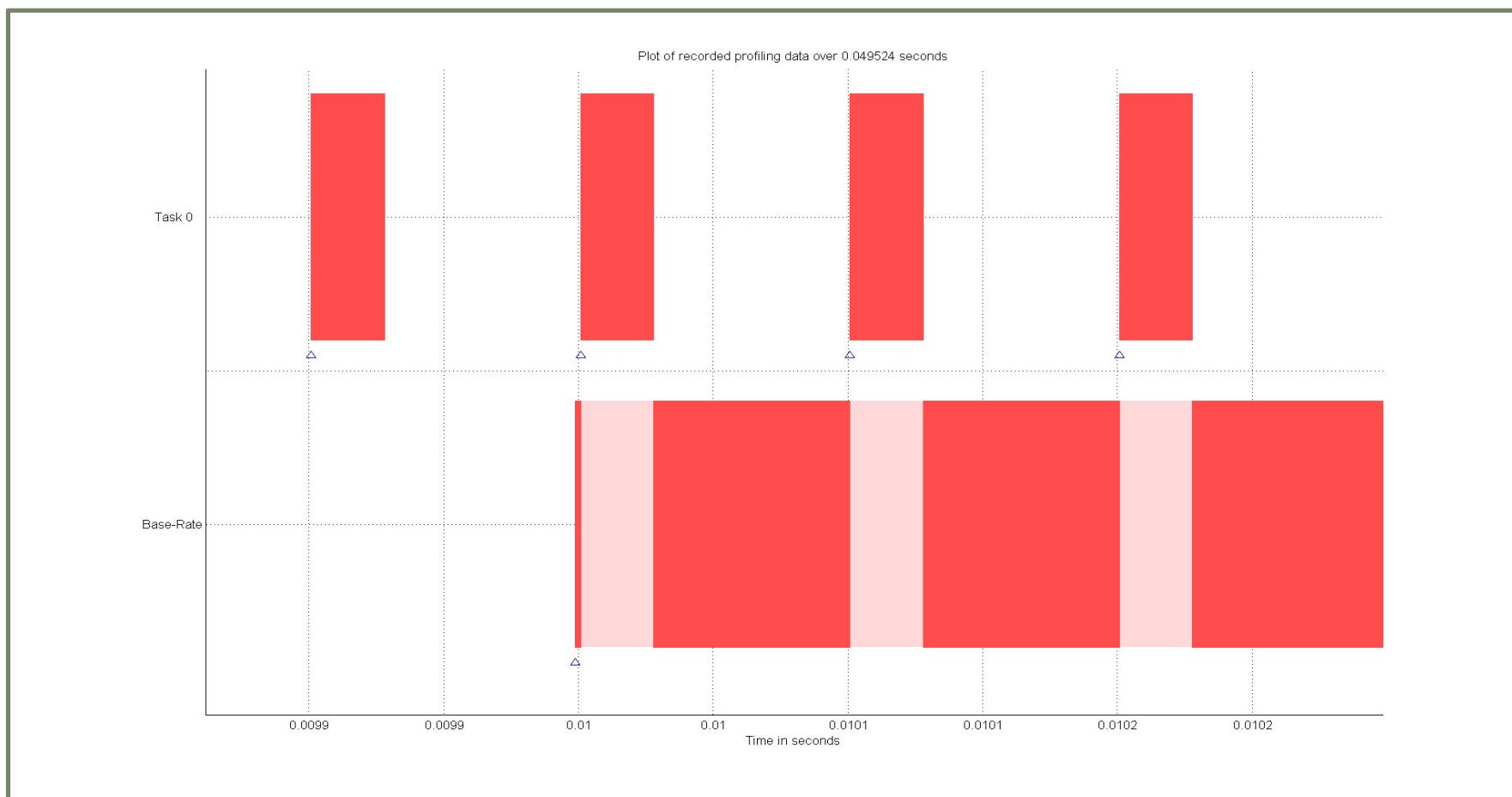
## Reálný pohon



## Modelově orientovaný návrh příklad – testování

### Hardware In the Loop – profiling (měření vytížení procesoru)

- Task 0: cca 27,1  $\mu$ s ( $T=100 \mu$ s), Base-Rate: cca 4,5 ms ( $T=10$  ms)



Regionální inovační centrum elektrotechniky  
Fakulta elektrotechnická  
Západočeská univerzita v Plzni

## Příští přednáška

# Základy návrhu SW v prostředí Matlab a Simulink

Jakub Talla

Regionální inovační centrum elektrotechniky  
Fakulta elektrotechnická  
Západočeská univerzita v Plzni

## Děkuji za pozornost!

**Adresa:** Univerzitní 26  
306 14 Plzeň  
Česká republika

**Tel:** +420 377 634 443  
**Fax:** +420 377 634 402

**Email:** [talic@kev.zcu.cz](mailto:talic@kev.zcu.cz)

[www.rice.zcu.cz](http://www.rice.zcu.cz)