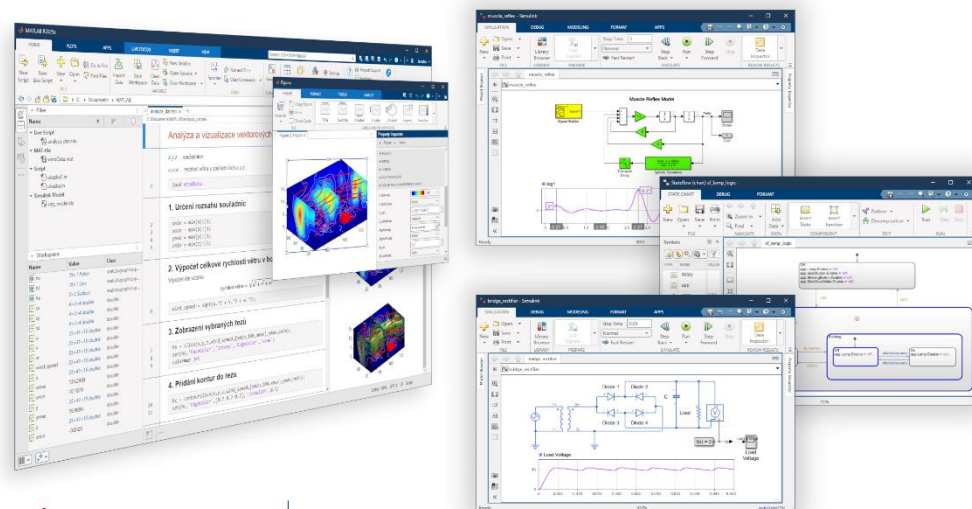


11.9.2025 Technical Computing Camp 2025

Návrh robustních, adaptivních a prediktivních řídicích systémů



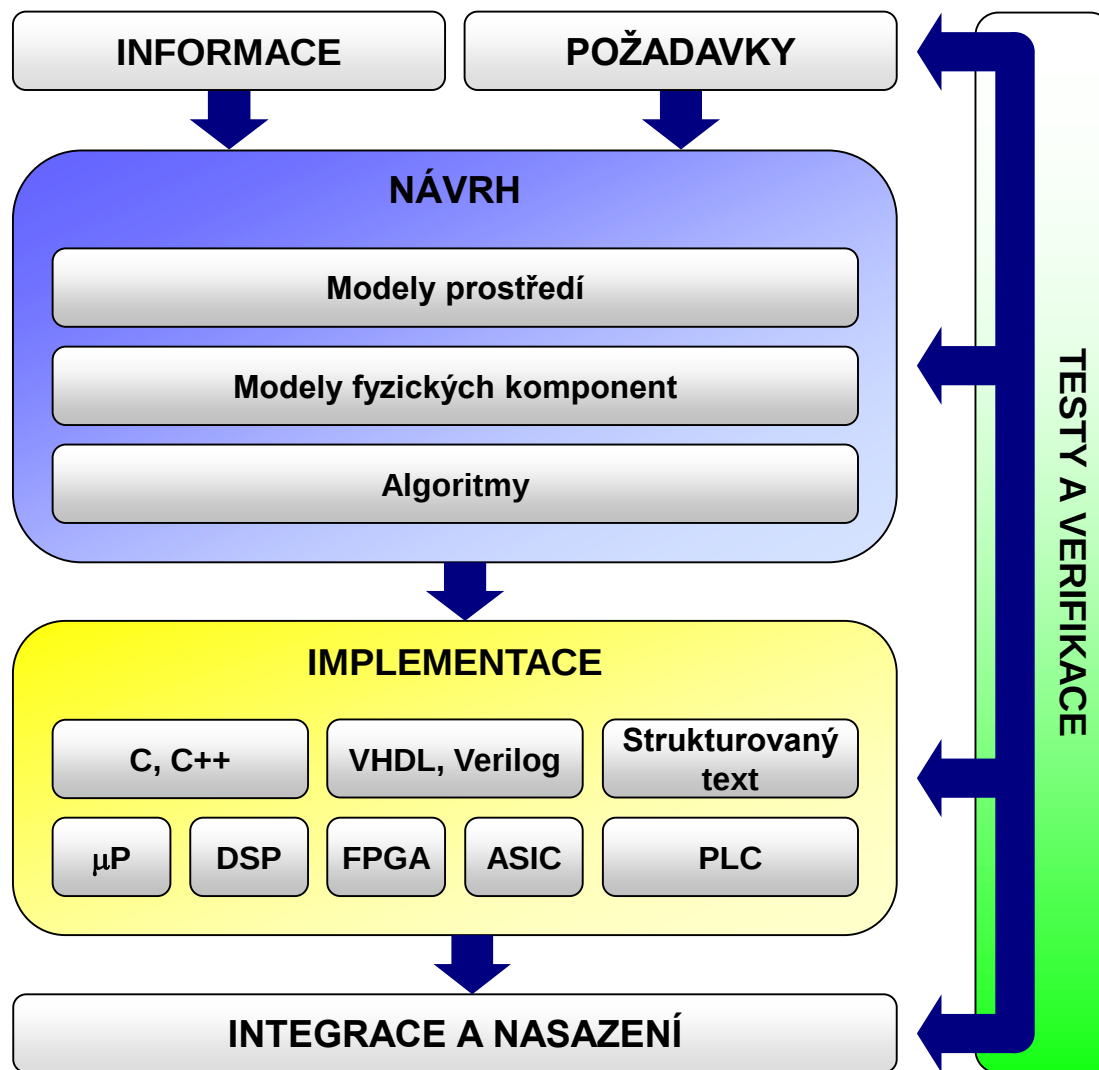
Jaroslav Jirkovský
jirkovsky@humusoft.cz

www.humusoft.cz

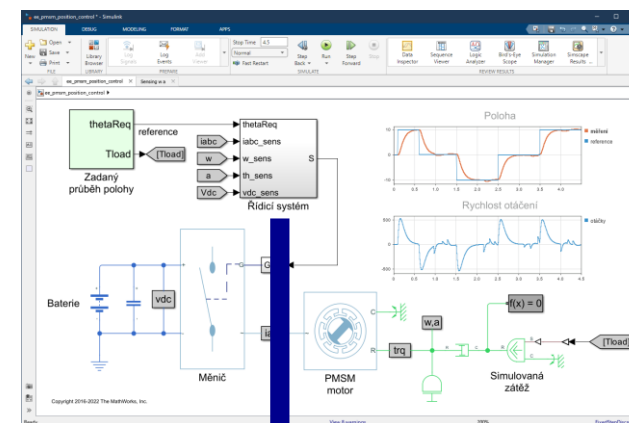
info@humusoft.cz

www.mathworks.com

Vývoj metodou Model-Based Design



Modelování, simulace a testování



Automatické generování kódu

```

Code
Control_System.c
Search

6  * Model version          : 6.4
7  * Simulink Code version  : 9.8 (R2023a) 19-Nov-2022
8  * C/C++ source code generated on : Tue Apr 25 11:36:04 2023
9
10 * Target selection: ert-tlc
11 * Embedded hardware selection: Texas Instruments->C2000
12 * Code generation objectives: Unspecified
13 * Validation result: Not run
14
15
16 #include "Control_System.h"
17 #include "math.h"
18 #include "rt_nonfinite.h"
19 #include "rtwtypes.h"
20 #include "string.h"
21
22 /* Block signals (default storage) */
23 #ifndef Control_System_Control_System_R;
24 #define Control_System_Control_System_R;
25
26 /* Block states (default storage) */
27 #ifndef Control_System_Control_System_Quark;
28 #define Control_System_Control_System_Quark;
29
30 /* External inputs (root input signals with default storage) */
31 #ifndef Control_System_Control_System_U;
32 #define Control_System_Control_System_U;
33
34 /* External outputs (root outputs fed by signals with default storage) */
35 #ifndef Control_System_Control_System_Y;
36 #define Control_System_Control_System_Y;
37
38 /* Real-time model */
39 static #ifndef Control_System_Control_System_R;
40 #define Control_System_Control_System_R;
41
42 /* Model step function */
43 void Control_System_step(void)
44 {
45     /* real32_T u; */
46     /* real32_T y; */
47
48     /* ... */
49 }
50
51 _ATMP1382134_MBOPodkladyControl_System_ert_rtwControl_System.c  Ln 6 Col 13

```

Nasazení

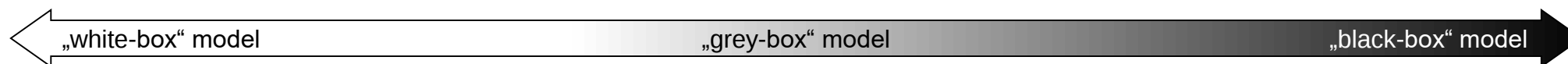


Modelování soustav : Přístupy k modelování

- Pro různé situace jsou vhodné různé přístupy

Fyzikální vztahy

Naměřená data



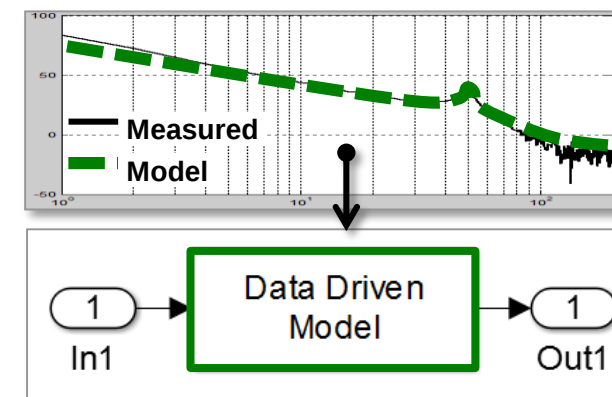
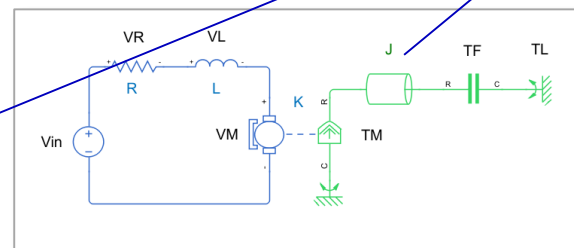
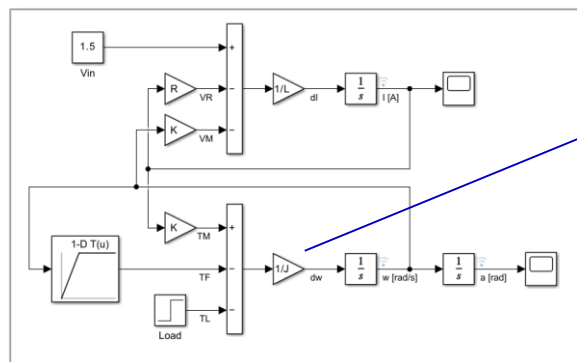
Modelování rovnic

Ladění neznámých parametrů

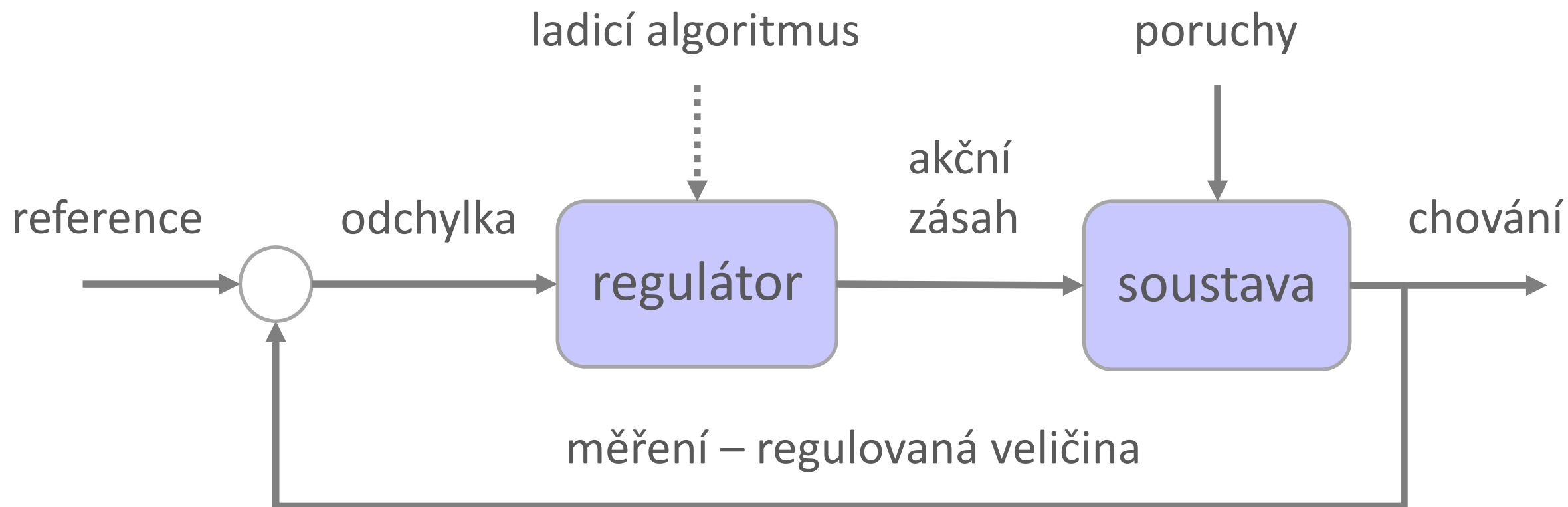
Identifikace soustav

Fyzikální sítě

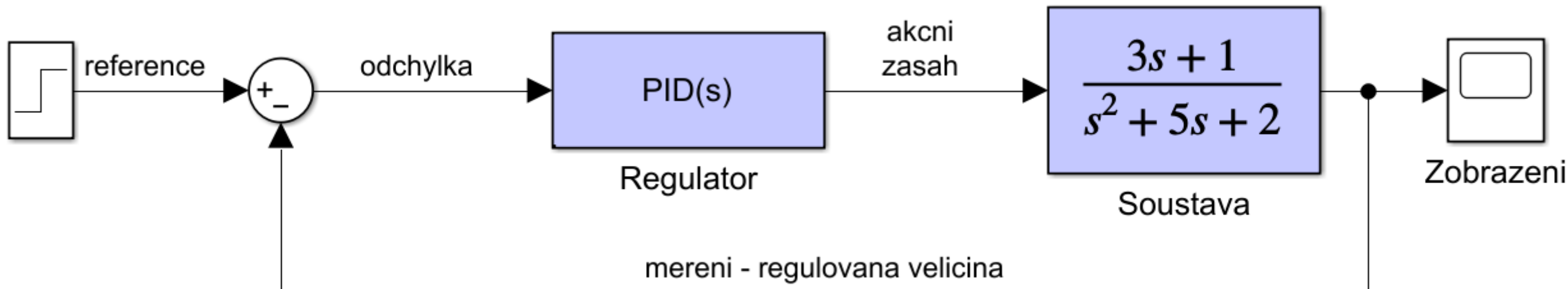
$$L \frac{dI}{dt} = V_{in} - RI - K\omega \quad J \frac{d\omega}{dt} = KI - f(\omega) - T_L$$



Základní schéma řídicího systému

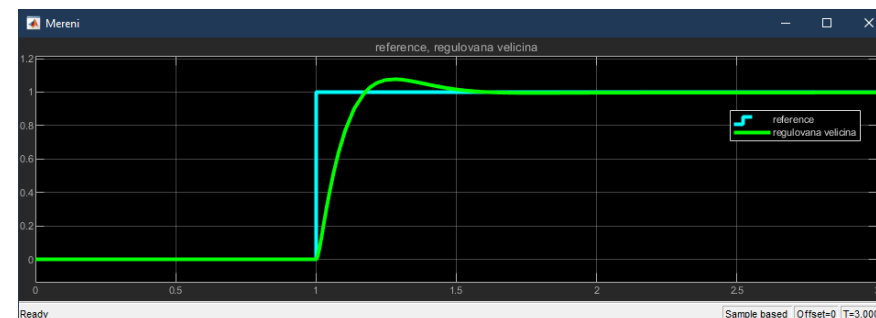
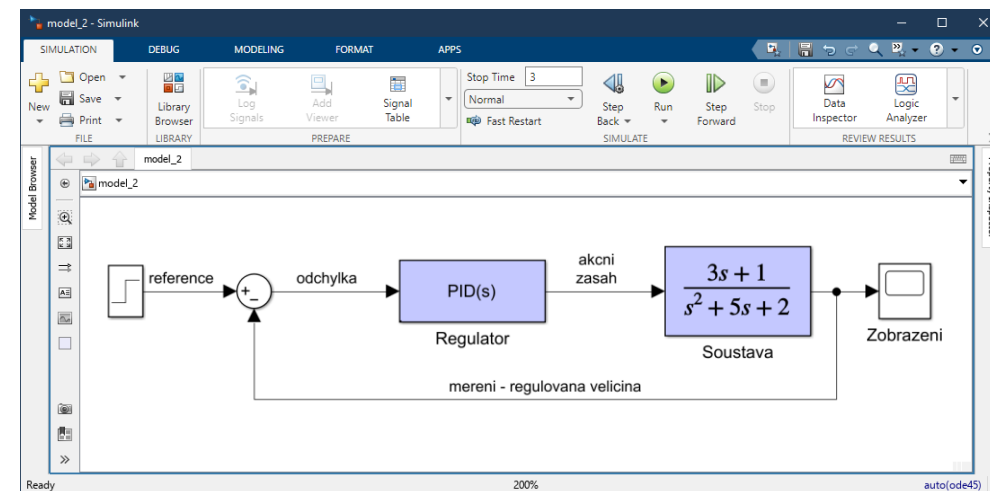


Základní schéma řídicího systému – Simulink



Návrh řídicích systémů v prostředí MATLAB a Simulink

- Propojení modelů soustav s regulátory
 - spojité a diskrétní prvky v jednom modelu
 - bohaté knihovny vstupních signálů
- Libovolná architektura řídicích systému
 - blok PID regulátoru v mnoha variantách
 - spojité, diskrétní, stavové regulátory
 - adaptivní a prediktivní řízení
- Nástroje pro ladění řídicích systémů



Rozdílné přístupy k návrhu řídicích systémů

Metody založené na modelu (~ tradiční)

PID, gain-scheduling, stavová vazba
H-infinity / Mu-syntéza
Návrh ve frekvenční oblasti
Model predictive control (MPC),
LQR / LQG, ...

Způsoby modelování

Model založený na
fyzikálních principech

Model založený na datech
získaný identifikací systému

(~ MPC s identifikovaným
modelem)

Metody založené na datech

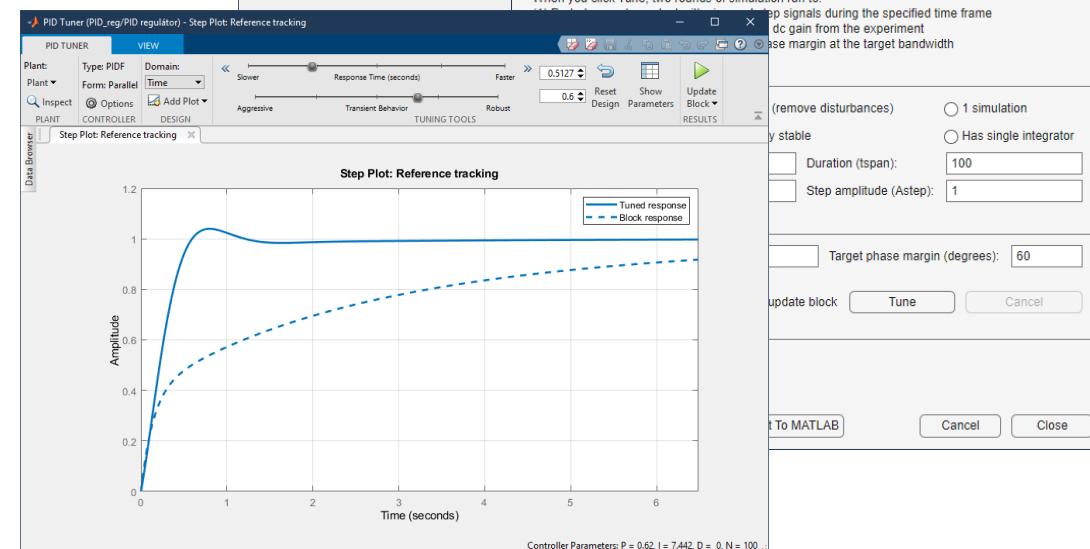
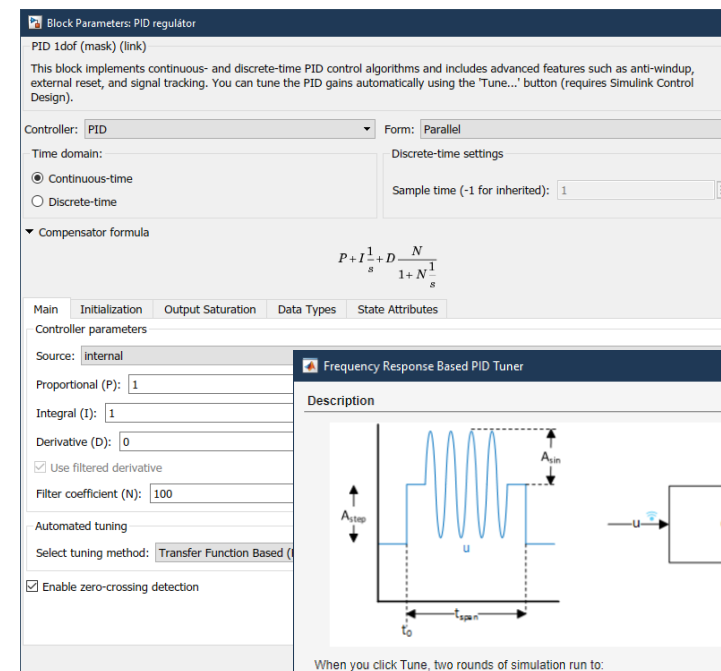
Active disturbance rejection control (ADRC)
Extremum seeking control (ESC)
Model reference adaptive control (MRAC)
Reinforcement Learning (RL)

Metody řízení založené na modelu se při návrhu řízení spoléhají na model systému.

V metodách založených na datech se regulátor nebo model učí z naměřených dat.

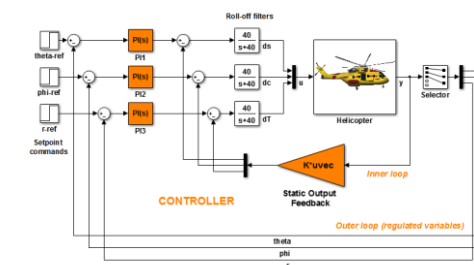
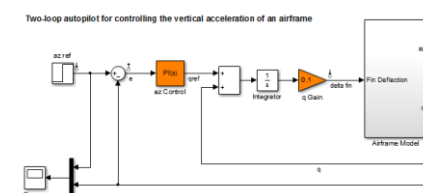
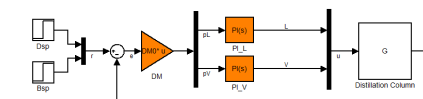
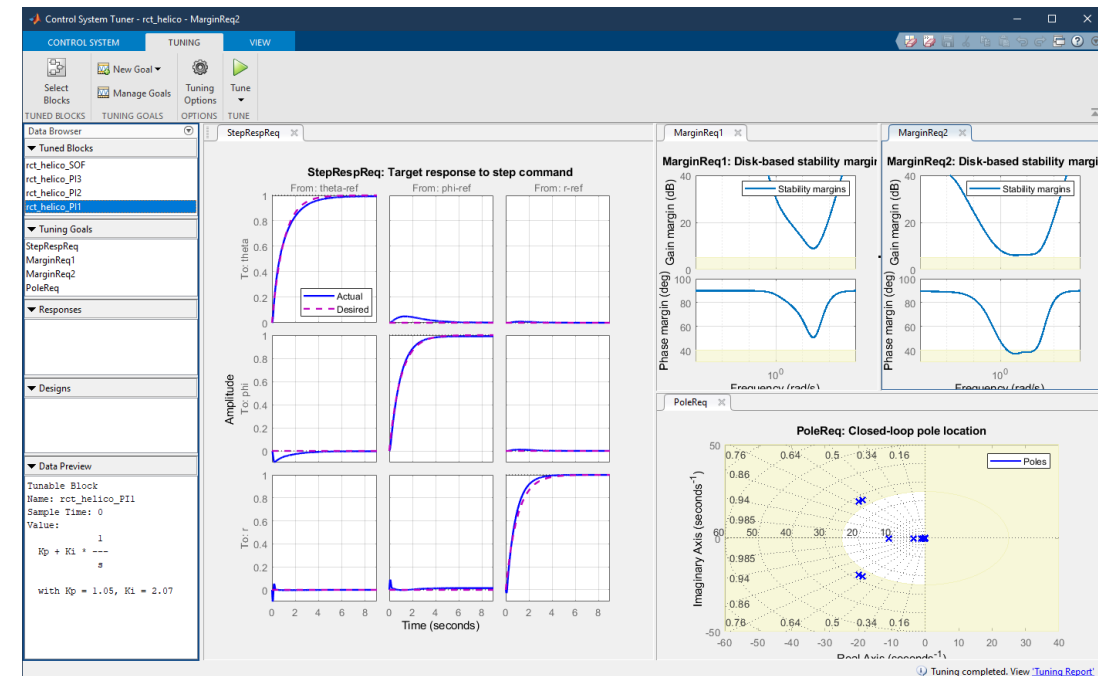
Ladění PID regulace

- Pro jednosmyčkovou PID regulaci
 - více smyček lze ladit jednu po druhé
- Využívá linearizaci soustavy
 - možná volba pracovního bodu
- Grafická aplikace *PID Tuner*
 - ladění na základě přechodové odezvy
- *Frequency Response Based PID Tuner*
 - pro soustavy, které nelze linearizovat



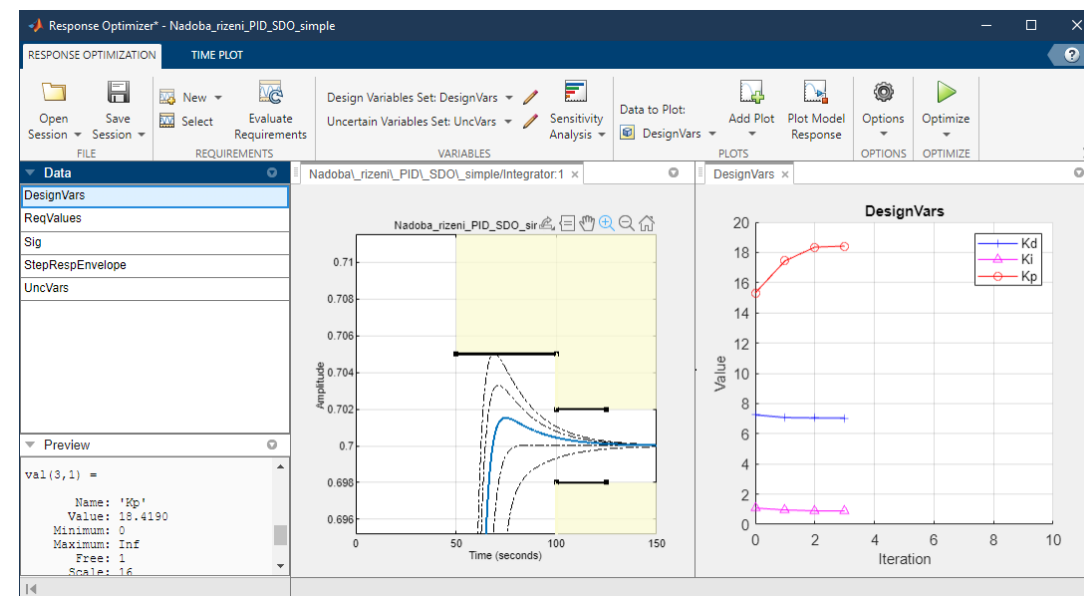
Ladění systémů s více vstupy a výstupy (MIMO)

- Libovolná struktura řídicího systému
 - společné ladění všech prvků
 - linearizace modelu a nehladká optimalizace
- Knihovna nastavitelných kritérií
 - v časové i frekvenční oblasti
 - kritéria stability
- *Control System Tuner*
 - grafická aplikace pro nastavení a ladění



Ladění na základě optimalizace

- Využívá nelineární model v Simulinku
 - nastavení podmínek pro zvolené signály
- Libovolná struktura řídicího systému
 - výběr parametrů pro optimalizaci
- Různé optimalizační algoritmy
- Podpora paralelních výpočtů
- *Response Optimizer*
 - grafická aplikace pro nastavení a ladění



Evoluce řídicích algoritmů v prostředí MATLAB a Simulink

2018 - 2023

2023

2024

2025

Ladění PID

- Closed-Loop PID Autotuner (R18a)
- Field Oriented Control Autotuner (R20a)
- Gain-Scheduled PID Autotuner (R24a)
- Start-Stop Generator (R24a)

Pokročilé řízení

- Model Reference Adaptive Control (MRAC) (R21a)
- Extremum Seeking Control (ESC) (R21a)
- Active Disturbance Rejection Control (ADRC) (R22b)
- MRAC with Single Hidden Layer Neural Network (R23a)
- Extended State Observer (R24a)
- Disturbance Compensator (R24a)
- Sliding Mode Control (R24b)
- Iterative Learning Control (R24b)
- Linear Sliding Mode Control
- Virtual Reference Feedback Tuning
- Ultra-Local Model for Disturbance Compensation

Omezení zásahu

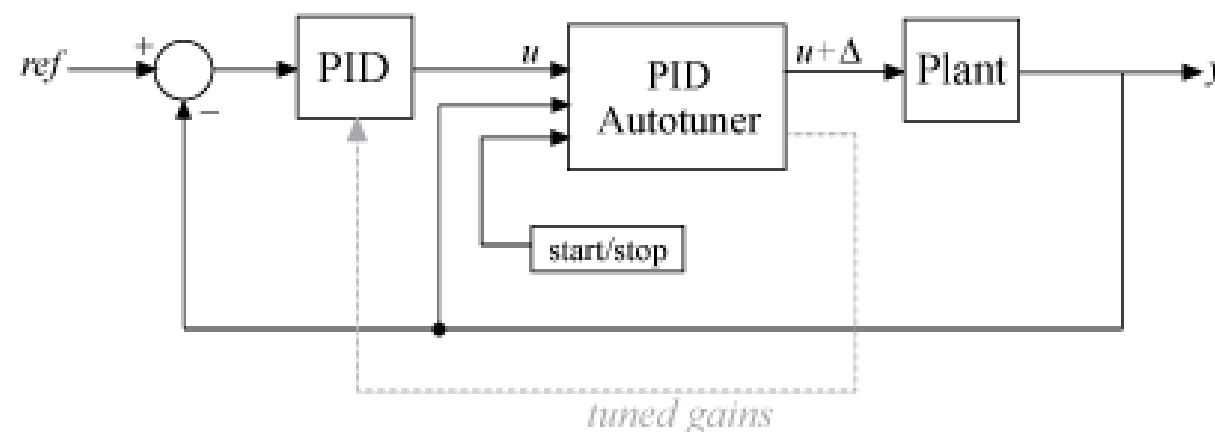
- Constraint Enforcement block (R21a)
- Barrier Certificate Enforcement Block (R22a)
- Passivity enforcement block (R23a)

Odhad frekvenční odezvy

- Frequency Response Estimator block (R19a)
- Frequency Response Estimator block enhancements:
 - Support for PRBS signals (R23a)
 - Set PRBS parameters automatically (R23b)
- Sinestream Signal Generator (R24a)
- PRBS Signal Generator (R24a)
- Start-Stop Generator (R24a)
- Superposition Signal Generator (R24b)

Automatické ladění PID regulátorů v reálném čase

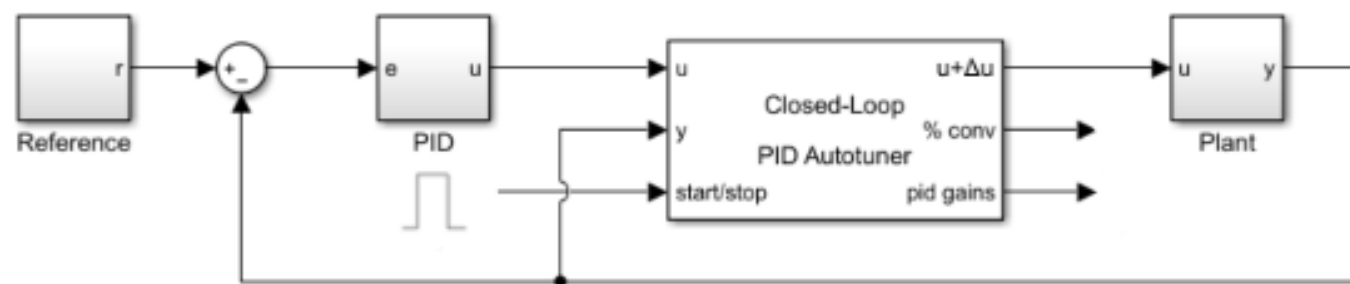
- Umožňuje ladit PID regulátory v reálném čase vůči k reálné soustavě*
 - nevyžaduje model soustavy
 - ladění je spouštěno zadaným signálem přímo v nasazeném systému
- Dostupné bloky
 - Open-Loop PID Autotuner
 - Closed-Loop PID Autotuner
 - Gain-Scheduled PID Autotuner
 - Field Oriented Control Autotuner
- Podpora generování kódu
 - implementace ladicího algoritmu na hardware



Dva způsoby zapojení experimentu pro ladění PID

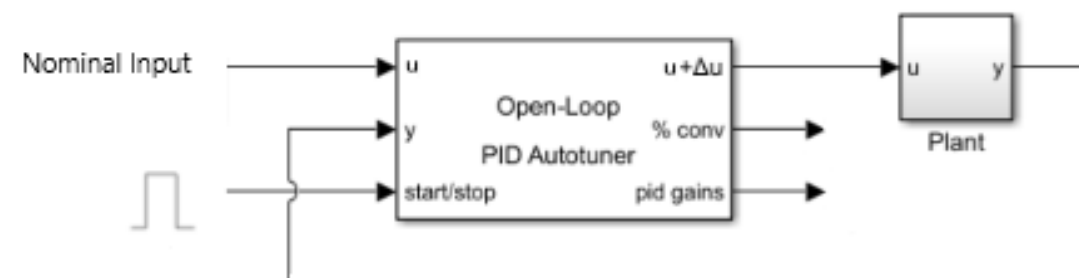
- Ladění probíhá na základě experimentálního odhadu frekvenční odezvy soustavy
- Experiment v uzavřené smyčce

– blok *Closed-Loop PID Autotuner*



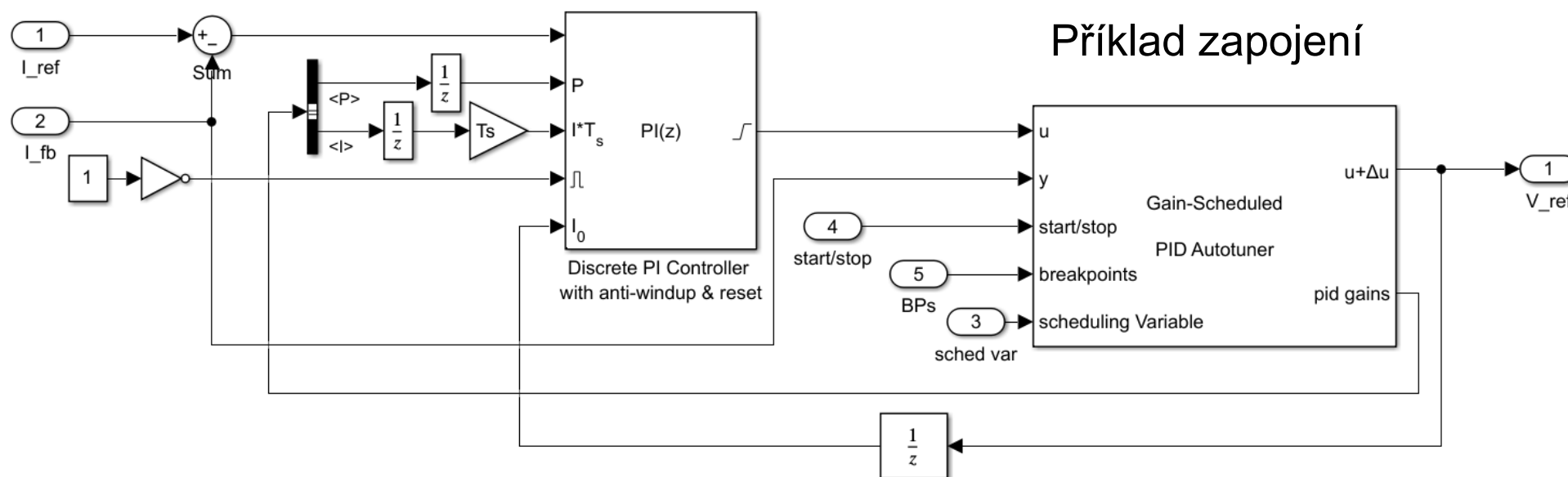
- Experiment v otevřené smyčce

– blok *Open-Loop PID Autotuner*



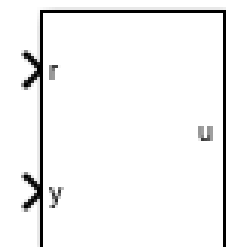
Ladění gain-scheduled PID

- Automatické ladění zesílení PID regulátoru v několika provozních bodech
 - experimentální odhad frekvenčních odezev soustavy
 - experiment v uzavřené smyčce, sekvenční změna pracovních bodů
 - blok *Gain-Scheduled PID Autotuner*

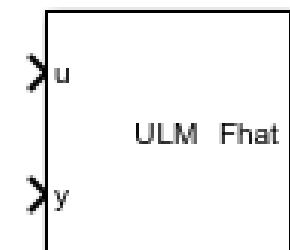


Kompenzace poruch

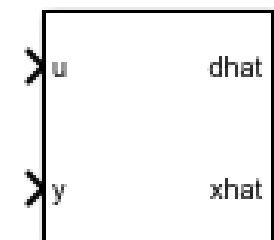
- Odhad a kompenzace poruch a neznámé dynamiky lineárních systémů
 - umožňuje odhadnout stavy a poruchy zařízení na základě jeho vstupů a výstupů
- Dostupné metody
 - Active Disturbance Rejection Control
 - Extended State Observer
 - Disturbance Compensator
 - Ultra Local Model



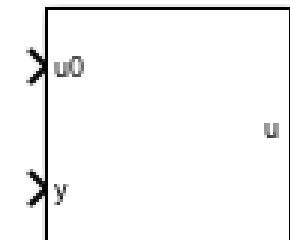
Active Disturbance Rejection Control



Ultra-Local Model



Extended State Observer



Disturbance Compensator

Active Disturbance Rejection Control (ADRC)

- Řízení soustav s neznámou dynamikou a vnitřními a vnějšími poruchami
 - regulace s robustním potlačením poruch a minimálním překmitem

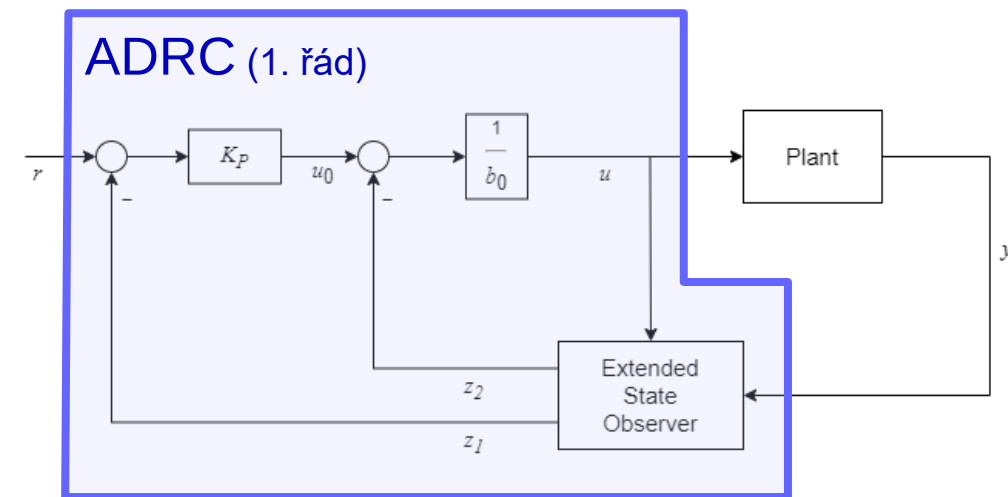
- Blok *Active Disturbance Rejection Control*

- Princip ADRC

- nevyžaduje model soustavy
- využívá jednoduchou aproximaci známé dynamiky

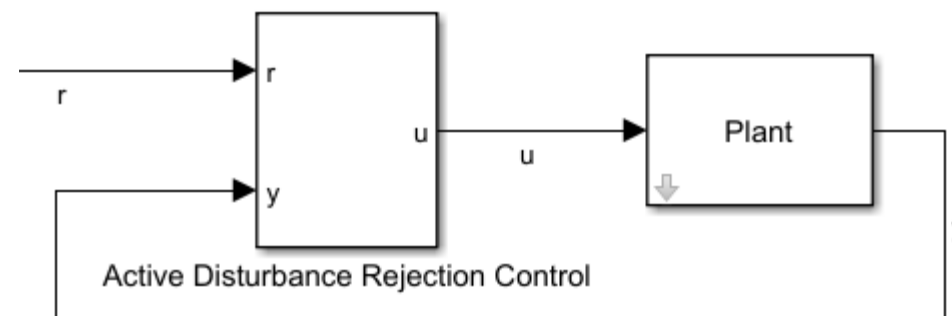
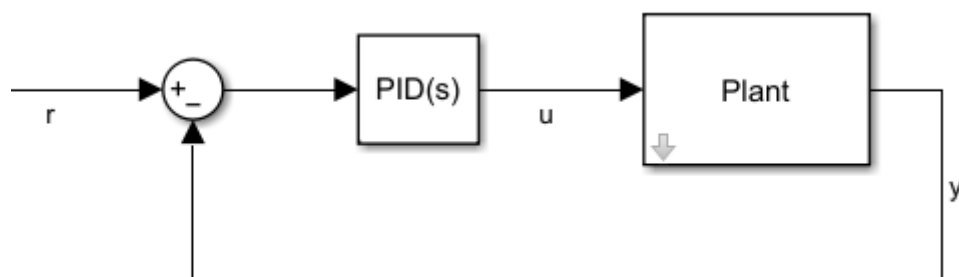
$$\dot{y}(t) = b_0 u(t) + f(t) \quad \text{nebo} \quad \ddot{y}(t) = b_0 u(t) + f(t)$$

- neznámá dynamika a poruchy $f(t)$ modelovány jako rozšířený stav soustavy

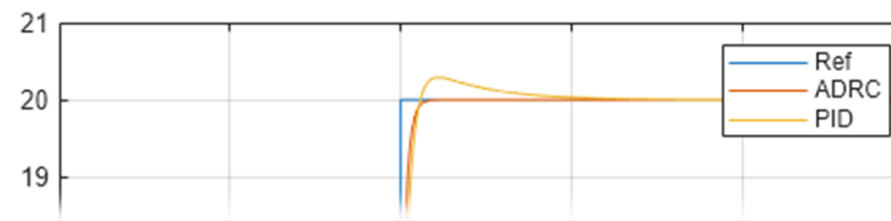


Active Disturbance Rejection Control (ADRC)

- Může sloužit jako alternativa PID regulace
 - zejména pro výkonovou elektroniku a řízení motorů
 - obdobné výpočetní nároky jako PID



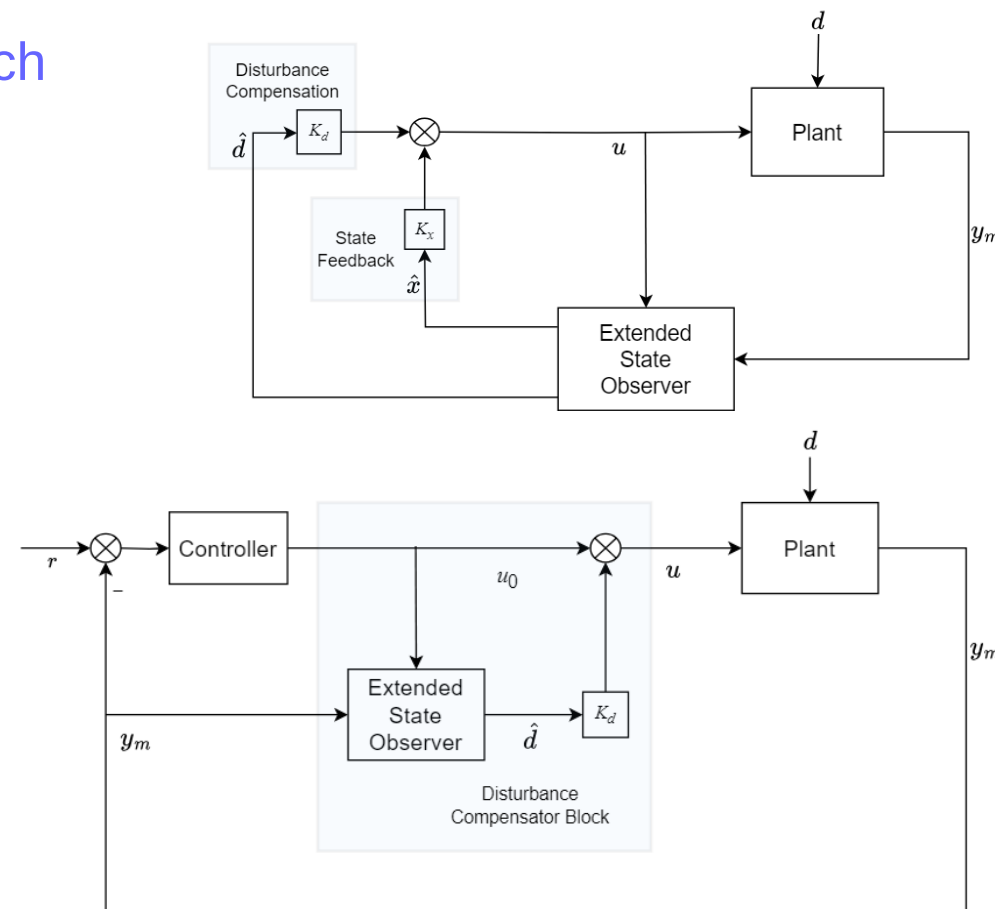
- Jednoduchý postup ladění
 - pro širokou škálu provozních podmínek



Extended State Observer a Disturbance Compensator

- Odhad a kompenzace poruch systémů
 - včetně neznámé dynamiky, vnitřních i vnějších poruch
- Blok *Extended State Observer*
 - odhad stavu a poruch
- Blok *Disturbance Compensator*
 - úprava nominálního akční zásahu regulátoru
 - kompenzace zjištěných poruch
- Bloky lze využít společně s dalšími regulátory
 - zlepšení potlačení poruch v daném řídicím systému

Typická zapojení



Ultra Local Model (ULM)

- Technika odhadu umožňující aproximaci nelineární soustavy

- jako systém s jedním nebo dvěma integrátory

$$\dot{y}(t) = F(t) + \alpha u(t) \quad \text{nebo} \quad \ddot{y}(t) = F(t) + \alpha u(t)$$

- afinity člen $F(t)$ zachycuje neznámou dynamiku a poruchy, je trvale aktualizován

- Odhadnutý model je platný

- pouze v okolí lokálního pracovního bodu

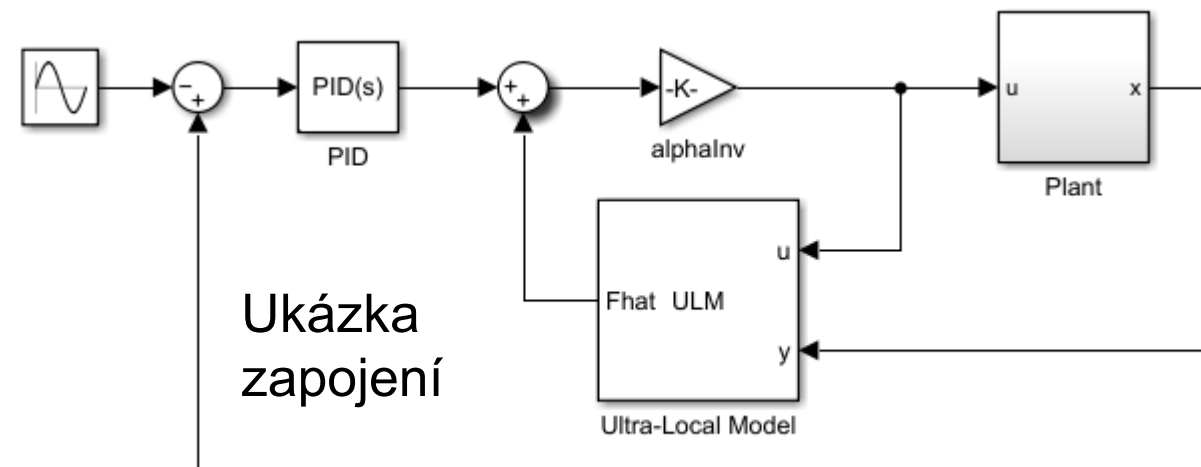
- v krátkém časovém úseku

- Blok *Ultra-Local Model*

- Použití

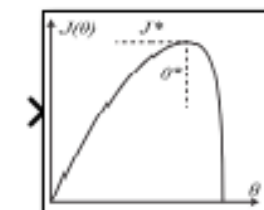
- pro samostatný odhad poruch a predikci výstupu

- pro zlepšení chování nominálního regulátoru (~PID) pomocí odhadu poruch

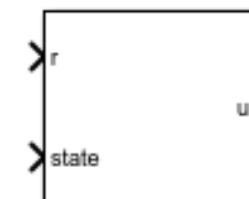


Adaptivní řídicí systémy

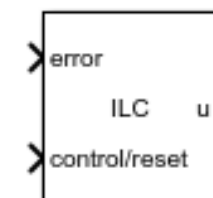
- Cílem je přizpůsobení se změnám soustavy
 - nepředvídané vlivy okolí, poruchy, změna dynamiky
- Dostupné metody
 - Extremum Seeking Control
 - Model Reference Adaptive Control
 - Iterative Learning Control
 - Sliding Mode Control
 - Virtual Reference Feedback Tuning



Extremum Seeking Control



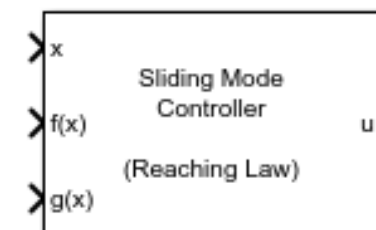
Model Reference Adaptive Control



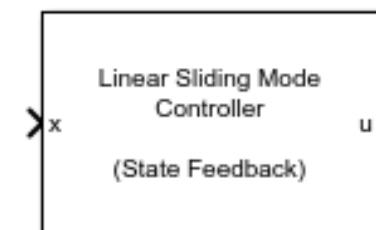
Iterative Learning Control



Virtual Reference Feedback Tuning



Sliding Mode Controller
(Reaching Law)



Linear Sliding Mode Controller
(State Feedback)

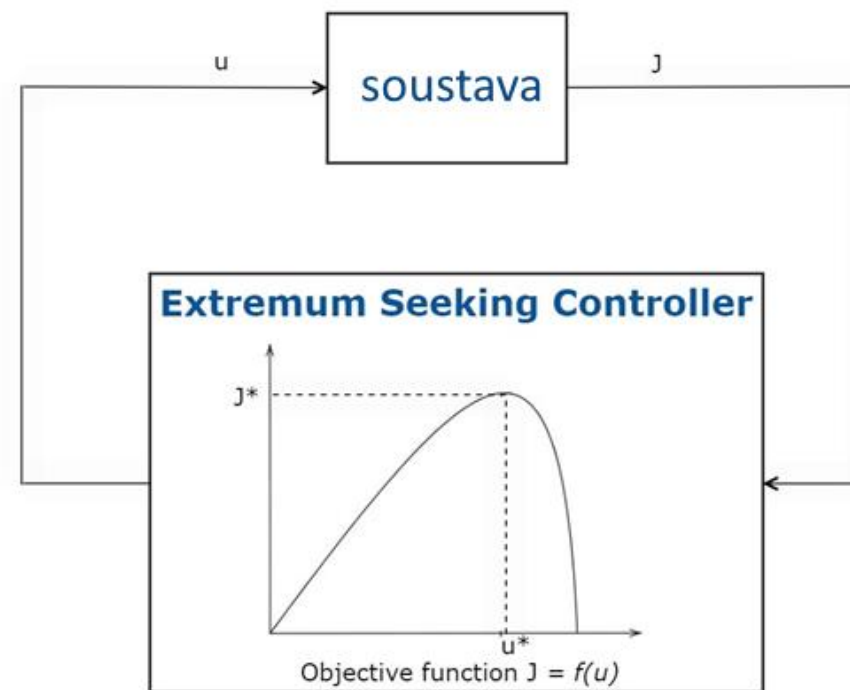
Proč použít adaptivní řídicí systémy?

- Adaptivní řídicí systémy mohou řešit náročné problémy nelineárního řízení, které jsou pro tradiční techniky obtížně řešitelné



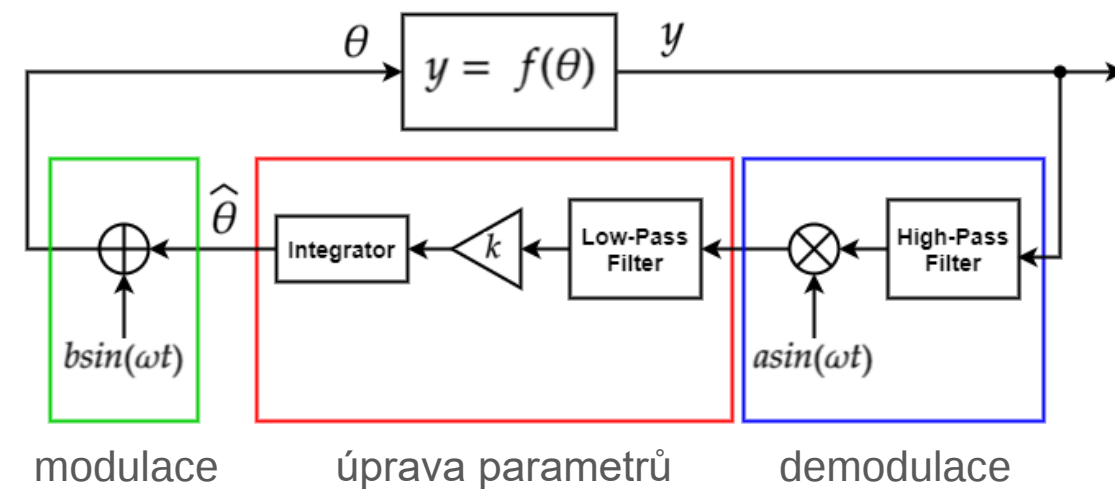
Extremum Seeking Control

- Optimální řízení soustav se změnami chování
- Princip: úprava parametrů řídicího systému pro maximalizaci účelové funkce
 - řešení optimalizační úlohy v reálném čase
 - modulace parametrů sinusovým signálem
- Nevyžaduje model soustavy
- Vhodné pro
 - systémy s pomalou změnou parametrů
 - stabilní systémy tolerující šum v řízení
 - malý počet adaptujících se parametrů
- Blok *Extremum Seeking Control*

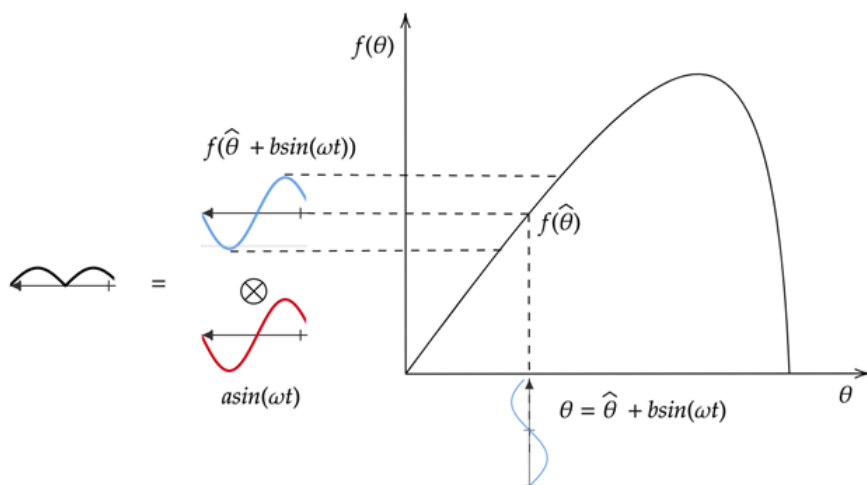


Extremum Seeking Control

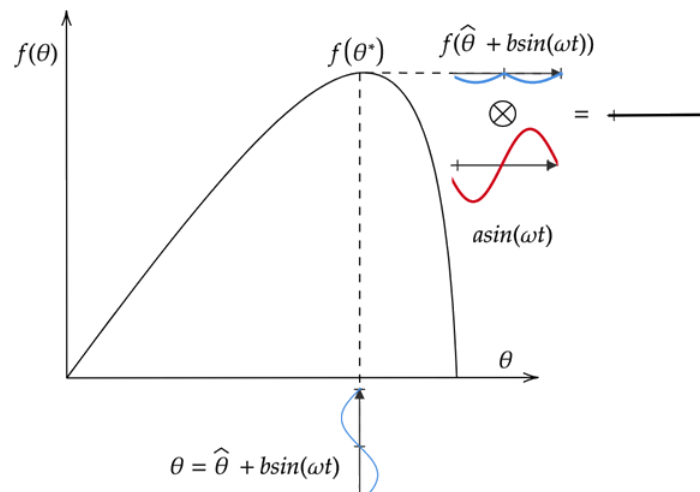
- Hledání optimálních parametrů



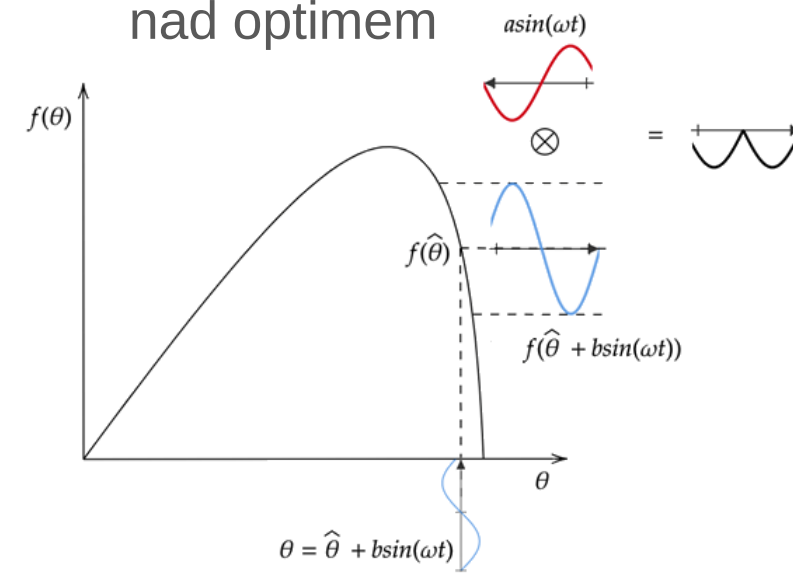
pod optimem



v optimu

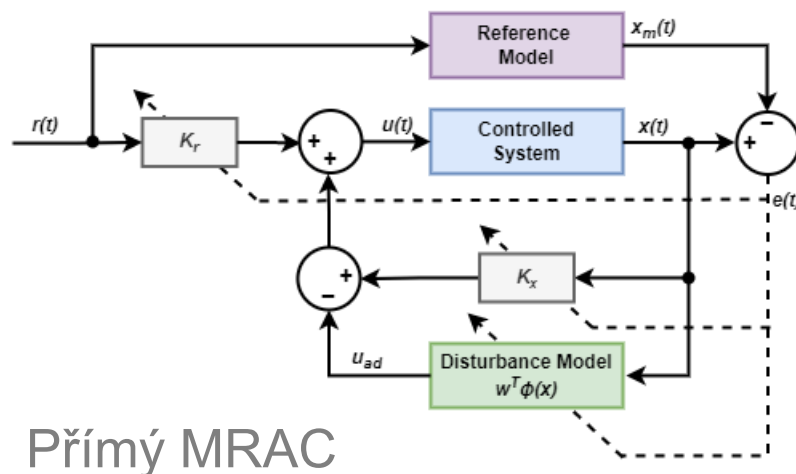


nad optimem

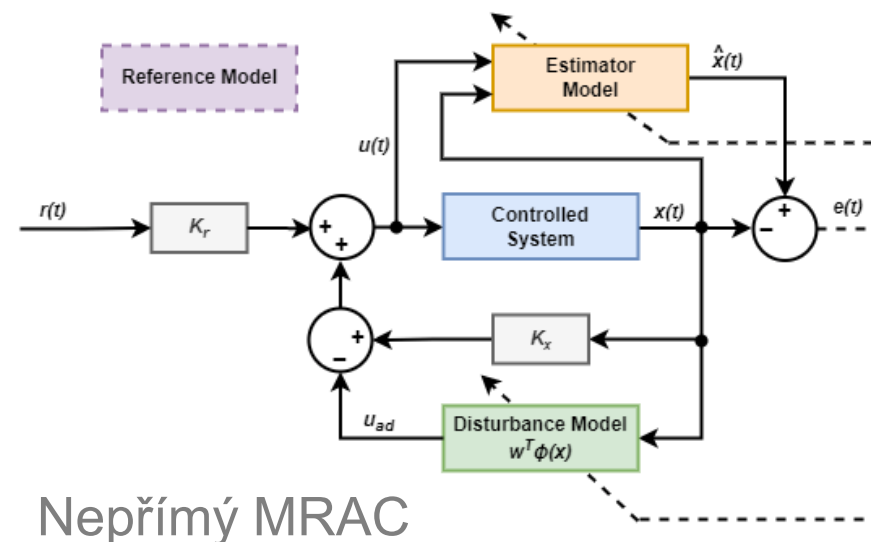


Model Reference Adaptive Control (MRAC)

- Umožní řízení systémů v případě neočekávaných změn
- Princip: adaptace, aby řízený systém sledoval chování referenčního modelu
 - odhad externích poruch a nejistých parametrů systému
 - model poruch: kromě „klasického“ může být i neuronová síť s jednou skrytou vrstvou
- Blok *Model Reference Adaptive Control*
- Dva typy:



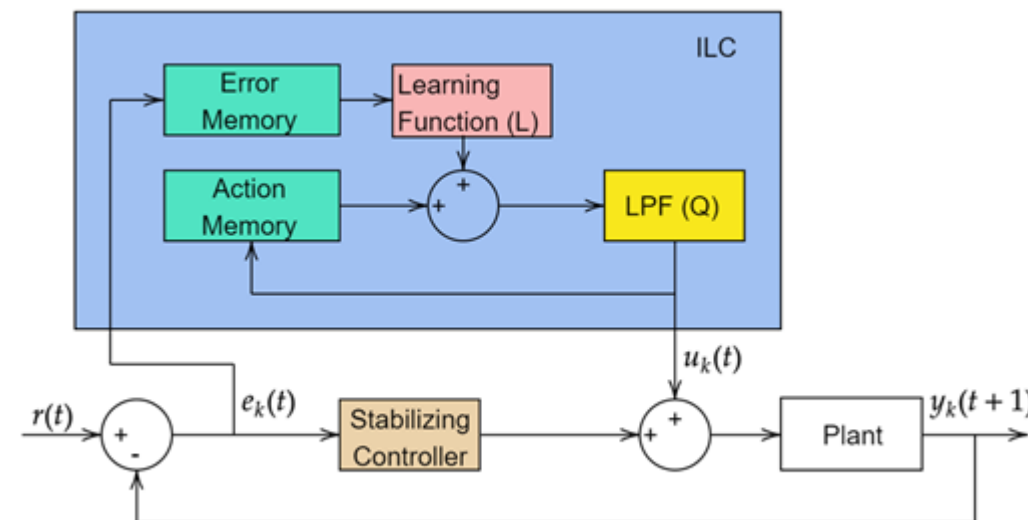
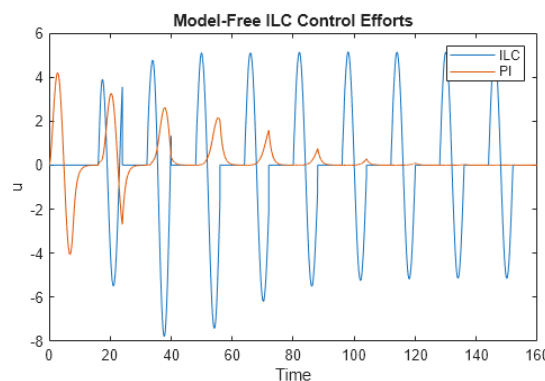
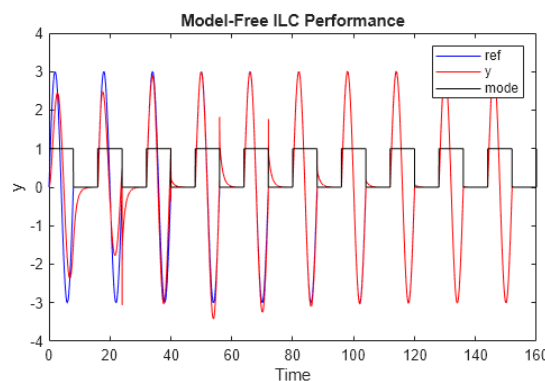
Přímý MRAC



Nepřímý MRAC

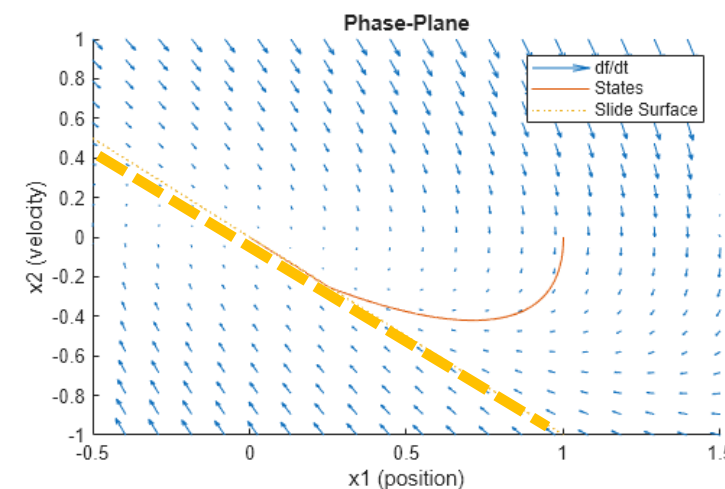
Iterative Learning Control (ILC)

- Cíl: Zlepšení výkonu opakujících se řídicích úkolů
 - průmyslové systémy, robotika, cyklické trajektorie UAV, ...
- Zapojuje se ve spojení se základním regulátorem (např. PID)
 - ILC se postupně učí z průběhu chyby v jedné iteraci a optimalizuje akční zásah pro následující
 - postupně zvyšuje svůj podíl na akčním zásahu oproti nominálnímu regulátoru
- Blok *Iterative Learning Control*



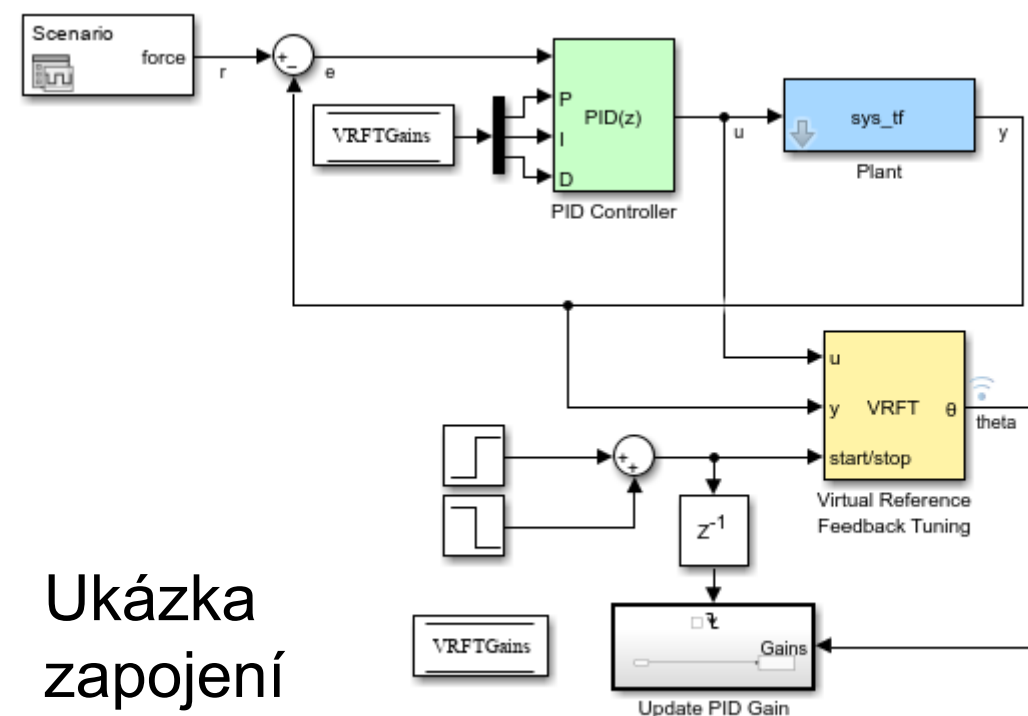
Sliding Mode Control (SMC)

- Robustní strategie řízení
 - trakční pohony, výkonová elektronika, robotika, autonomní řízení
- Přivádí trajektorii stavů systému na určenou plochu a udržuje ji tam
 - výhodné pro systémy vyžadující vysokou úroveň přesnosti a robustnosti
 - zejména za přítomnosti poruch a nejistot
- Blok *Sliding Mode Controller (Reaching Law)*
 - pro obecný nelineární model systému
 - sliding surface je třeba definovat manuálně
- Blok *Linear Sliding Mode Controller (State Feedback)*
 - založeno na nominálním lineárním modelu systému
 - sliding surface je možné navrhnout automatizovaně pomocí poskytnutých metod



Virtual Reference Feedback Tuning (VRFT)

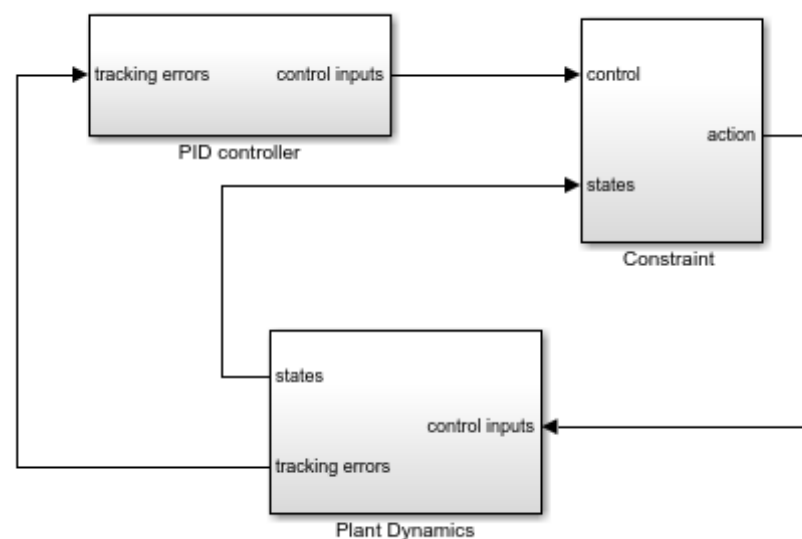
- Automatické ladění lineárních regulátorů na základě vstupně-výstupních dat
 - PID, FIR filtry, lineární kombinace regulátorů
- Výhoda:
 - pro ladění vyžaduje jediný experiment
 - online nebo offline přístup
- Blok *Virtual Reference Feedback Tuning*



Ukázka
zapojení

Omezení akčního zásahu

- Omezení akčního zásahu na základě zadaných omezení
 - modifikovaný akční zásah u co nejbližší požadovanému u_0
 - řeší optimalizační úlohu (QP) v reálném čase
- Připravené bloky
 - Constraint Enforcement, Barrier Certificate Enforcement, Passivity Enforcement



Blok Constraint Enforcement řeší:

$$\min_u |u - u_0|^2$$

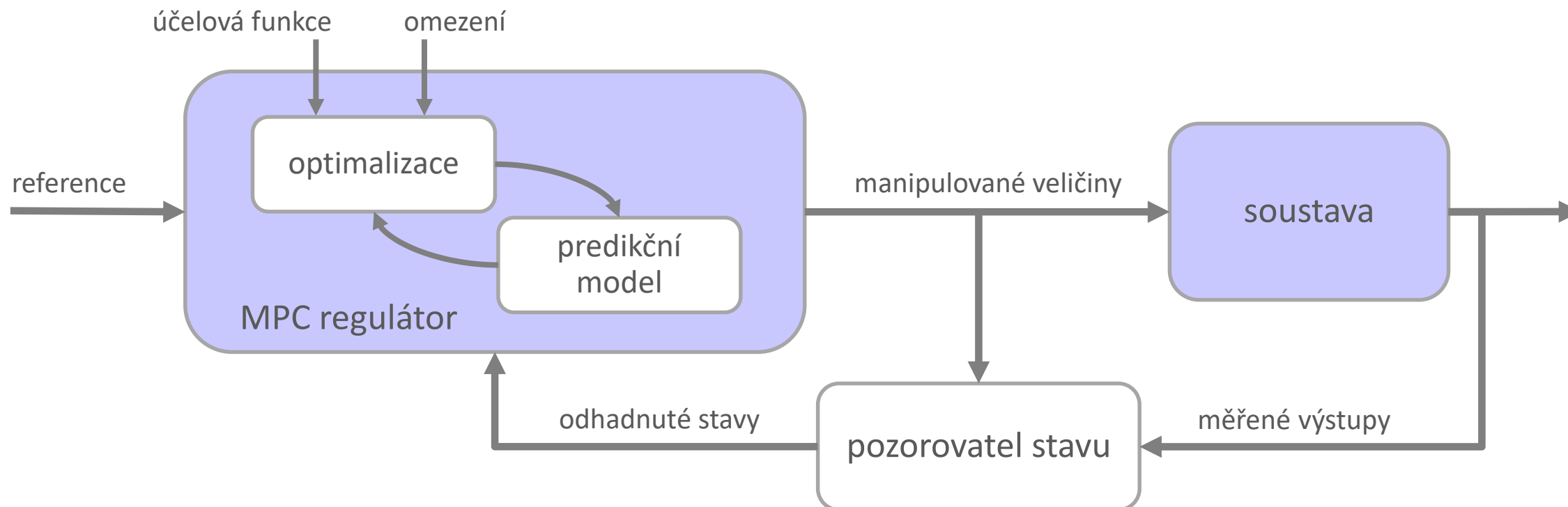
$$f_x + g_x u \leq c$$

$$u_{min} \leq u \leq u_{max}$$

omezení
limity akčního zásahu

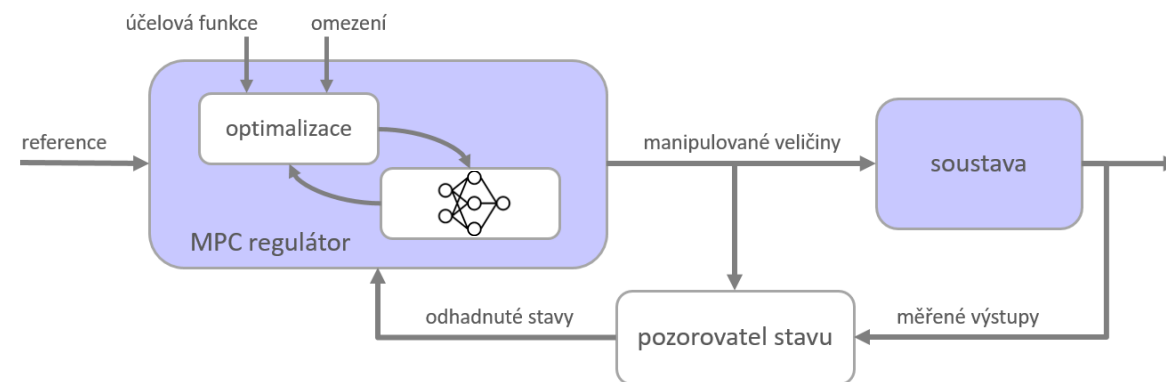
Model Predictive Control (MPC)

- Prediktivní řízení založené na modelu systému a optimalizaci v reálném čase
 - počítá optimální akční zásah podřízený zadaným omezením



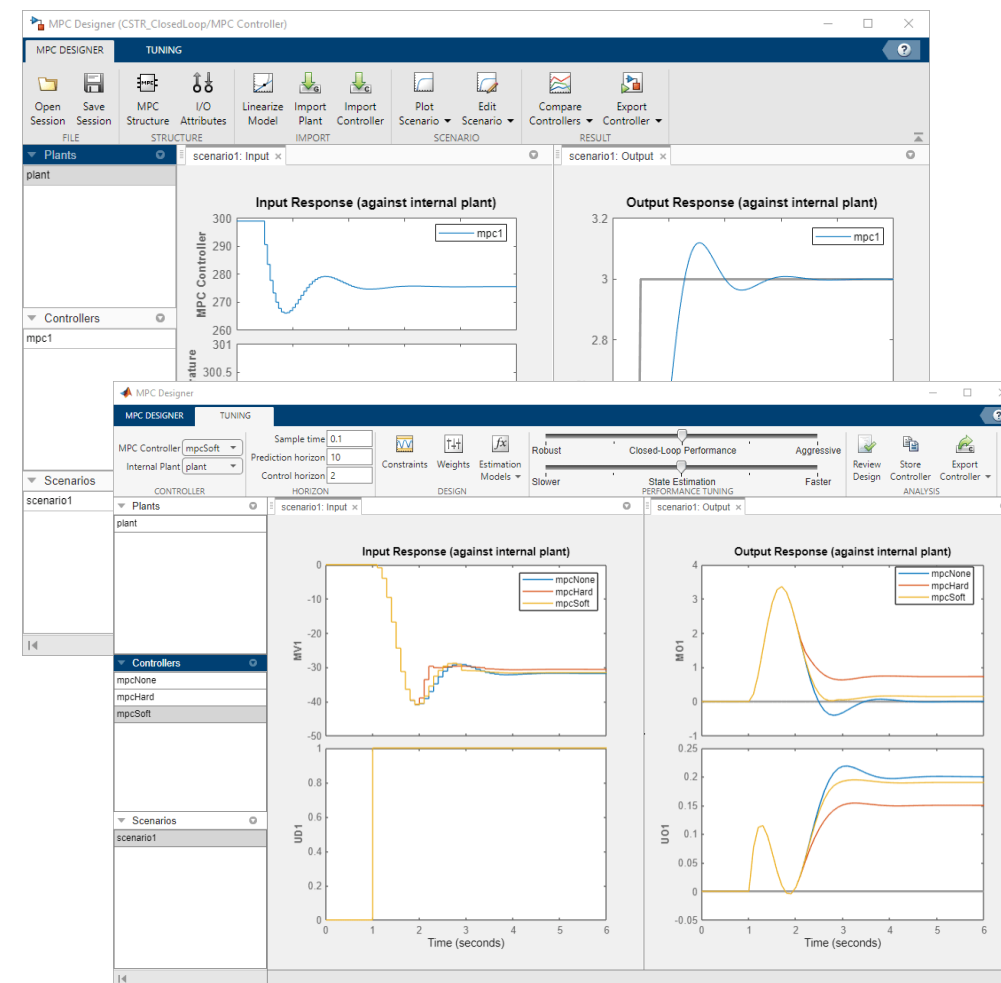
Model Predictive Control (MPC)

- Pro predikci využívá vnitřní model(y)
 - lineární
 - nelineární
- MPC + AI
 - predikční model může být ve formě neural state space modelu
- Typické využití MPC
 - řízení technologických procesů
 - autonomní řízení vozidel
 - robotika



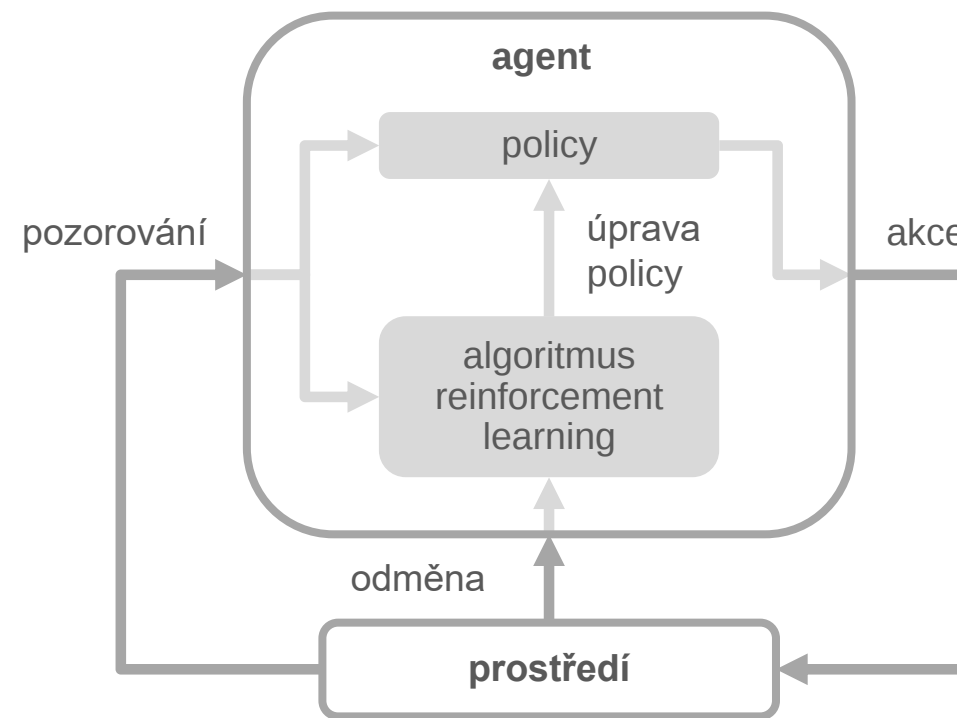
Tvorba MPC

- Připravené funkce
 - pro různé typy MPC
- Grafická aplikace MPC Designer
 - pro lineární implicitní MPC
- Bloky v prostředí Simulink
 - obecné MPC různých typů
 - specializované pro konkrétní úlohy



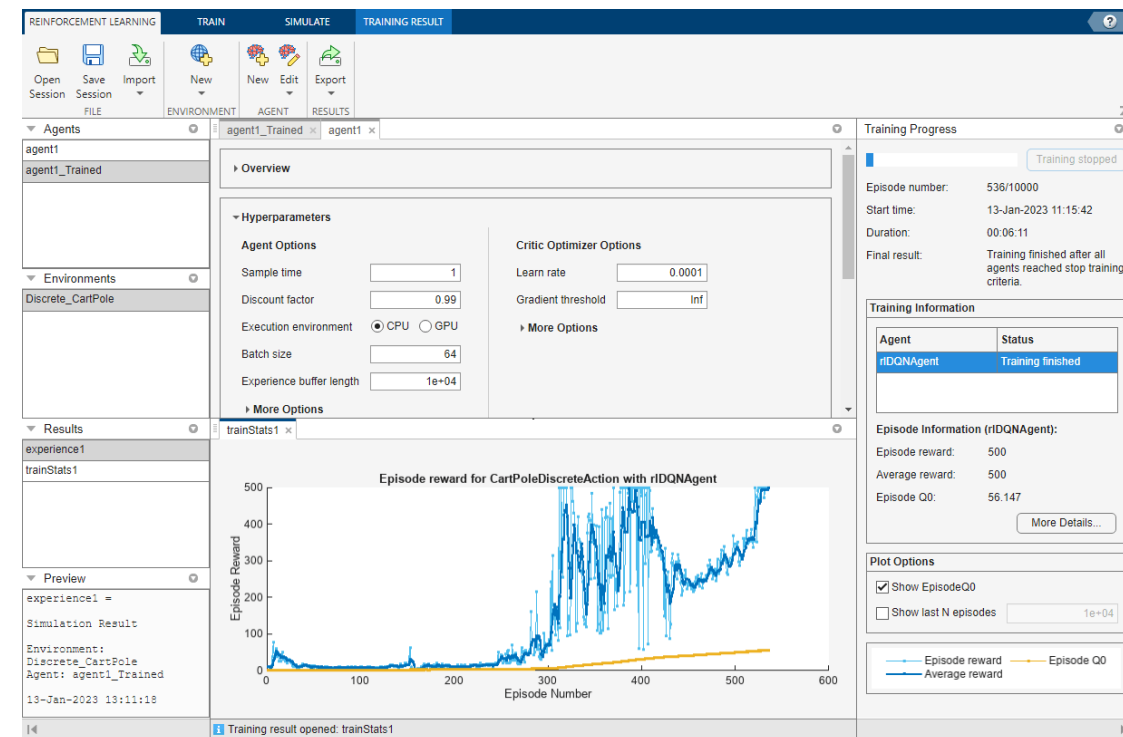
Reinforcement learning (RL)

- Metoda strojového učení
 - počítačový agent se učí optimálnímu chování opakovanou interakcí s dynamickým prostředím
- Cíl: maximalizovat odměnu v dlouhodobém časovém horizontu
- Algoritmus policy
 - hluboká neuronová síť (nejčastěji)
 - implementuje: regulátory, rozhodovací algoritmy
- Pro komplexní systémy
 - kde jsou tradiční metody obtížně formulovatelné



Nástroje pro reinforcement learning

- RL „prostředí“
 - vytvořené v MATLABu
 - model systému v Simulinku
- Návrh, učení a simulace agenta
 - připravené funkce
 - Reinforcement Learning Designer app
- Simulink: připravené bloky
 - RL Agent – simulace a učení agenta
 - Policy – simulace a nasazení naučené funkce policy
- Reinforcement Learning Toolbox



Automatické generování kódu pro aplikace v reálném čase

- Prototypování a testování systému – PC s měřicí kartou, real-time simulátor
- Generování optimalizovaného embedded kódu pro zvolenou cílovou platformu



Děkuji za pozornost