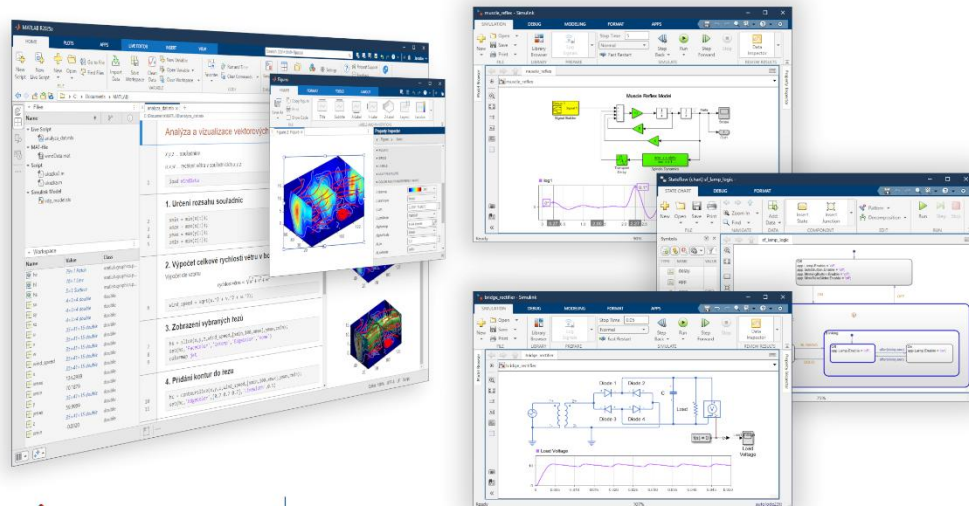


10.9.2026 Technical Computing Camp 2026

Návrh prediktivního řízení: Je to snazší než byste čekali



Jaroslav Jirkovský

jirkovsky@humusoft.cz

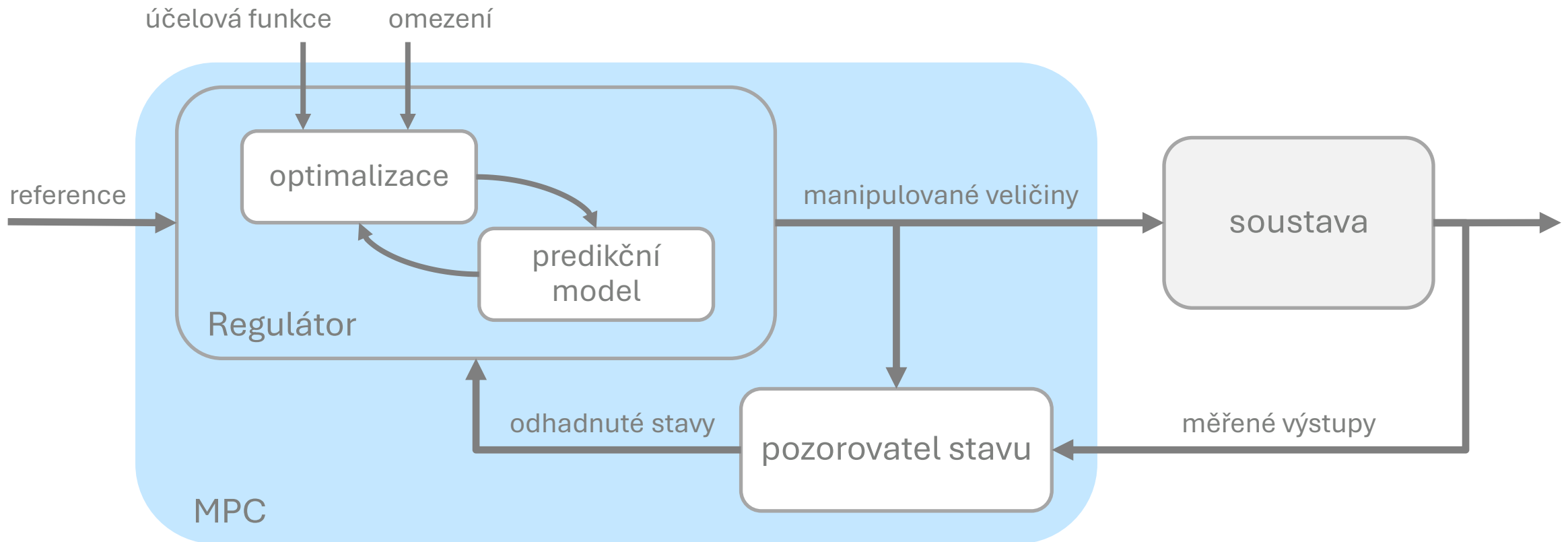
www.humusoft.cz

info@humusoft.cz

www.mathworks.com

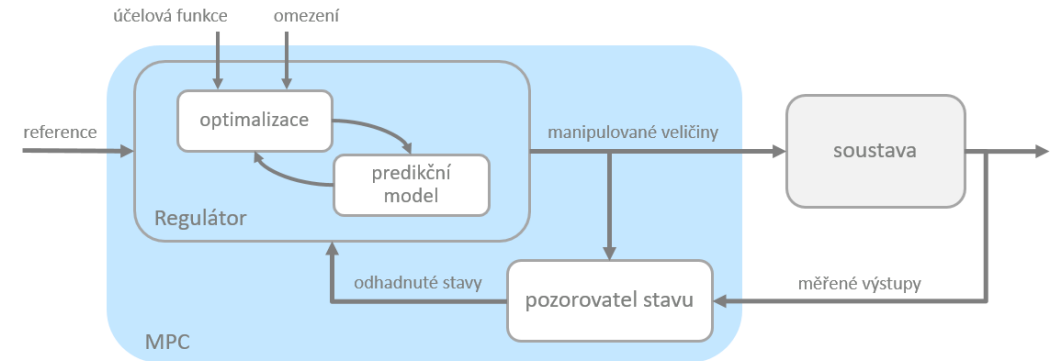
Model Predictive Control (MPC)

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



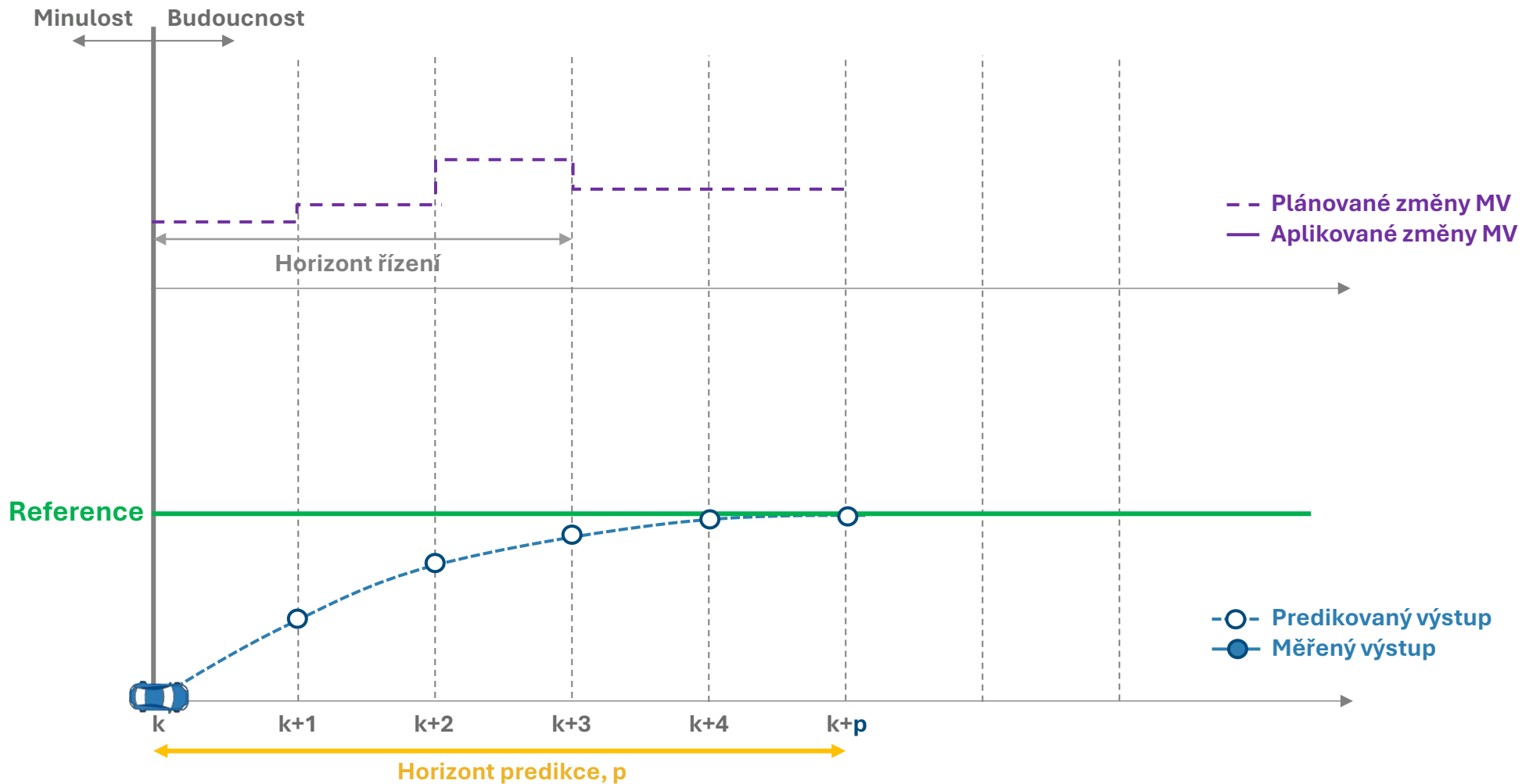
Model Predictive Control (MPC)

- 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



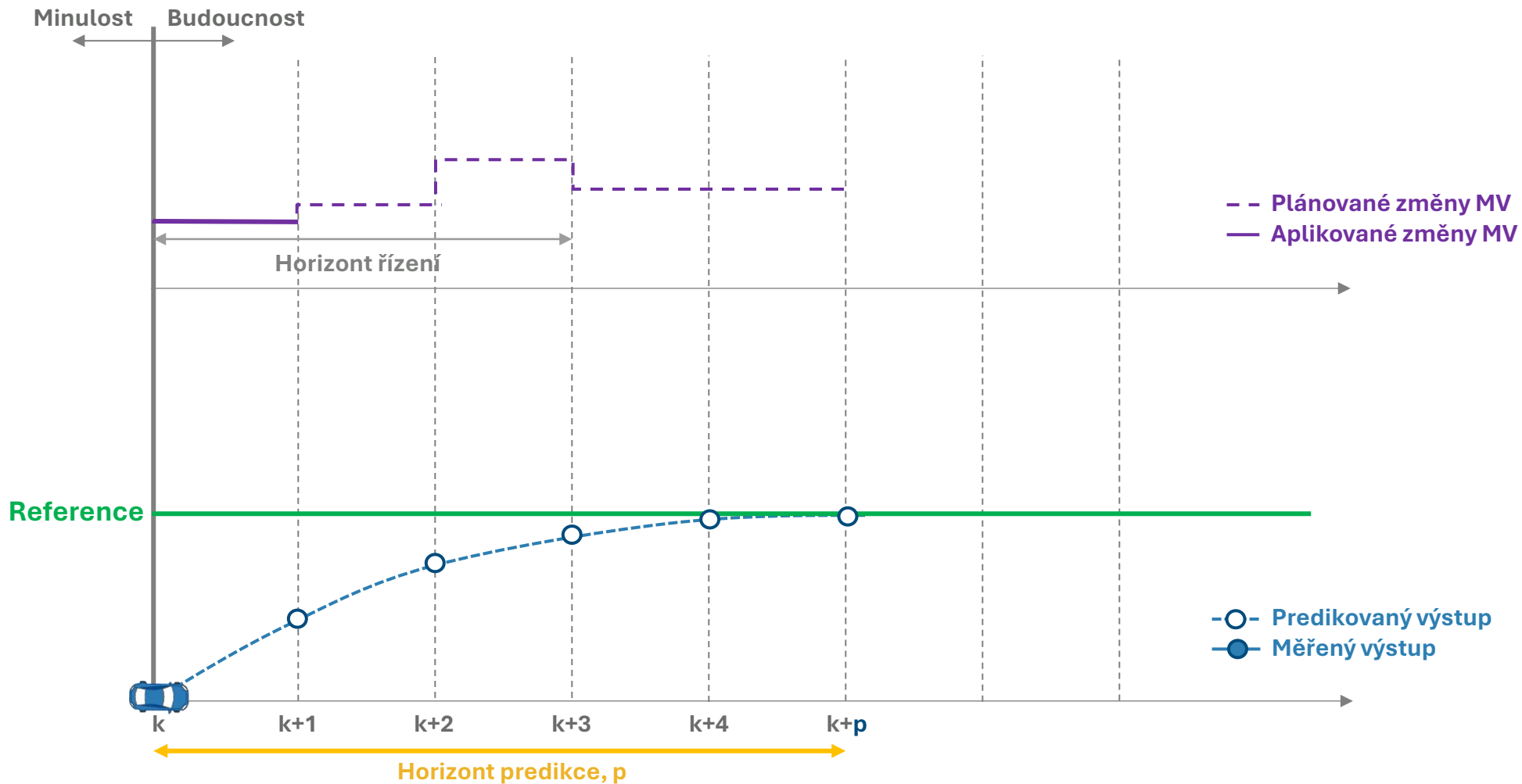
Jak MPC funguje

- Řeší optimalizační úlohu v kroku k , jaký optimální akční zásah provést k dosažení cíle



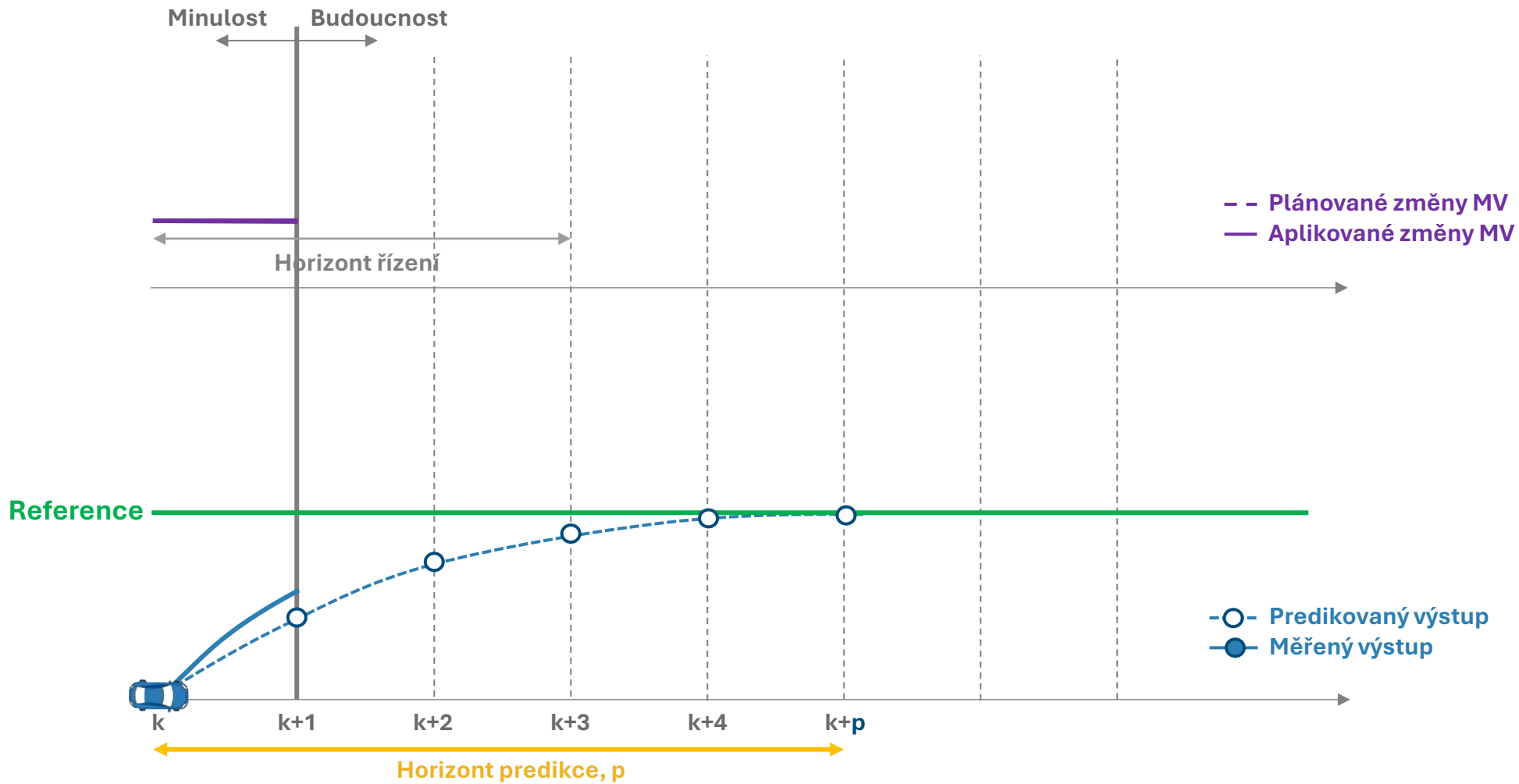
Jak MPC funguje

- Provede první krok řízení (otevřená smyčka), vyřadí zbytek



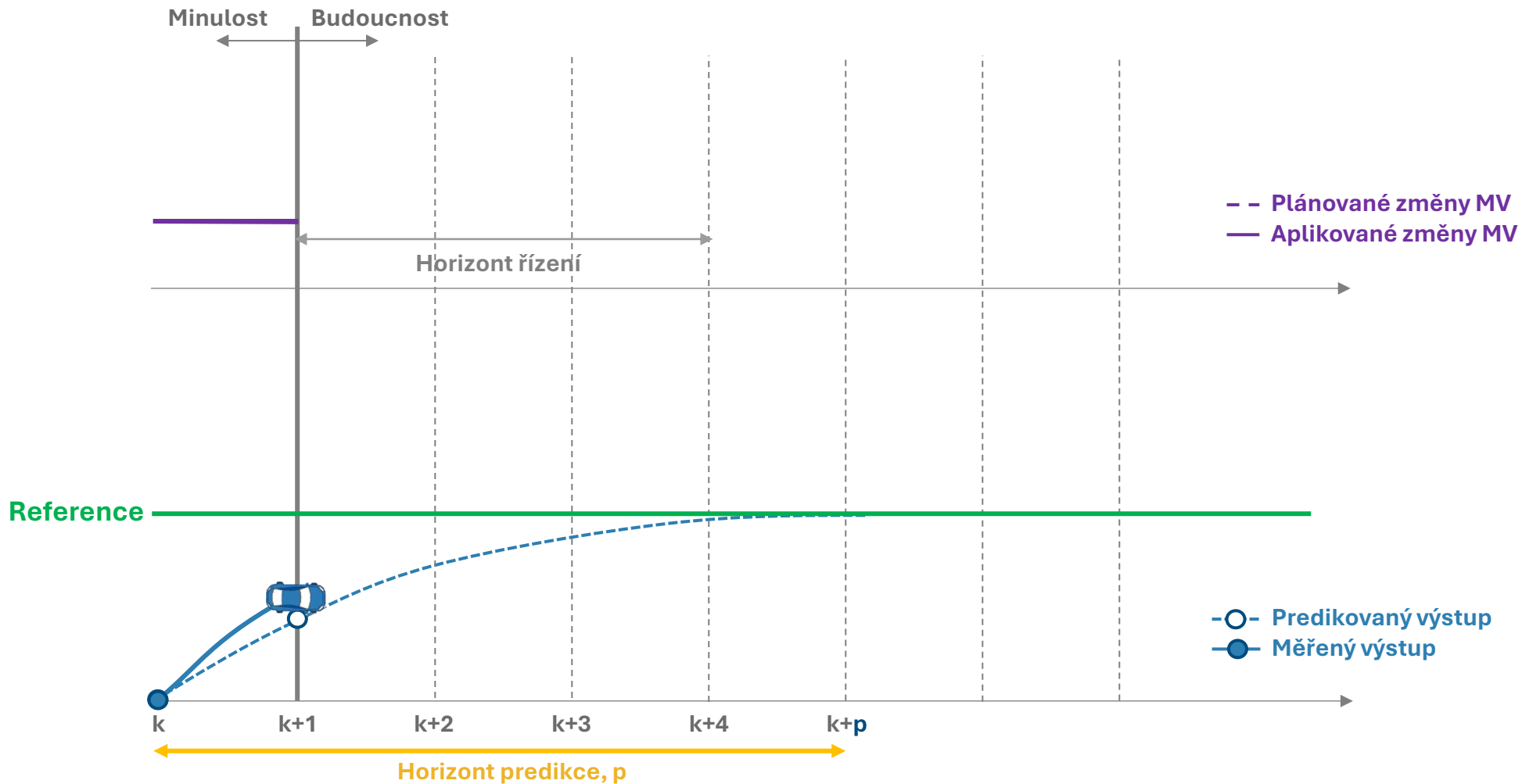
Jak MPC funguje

- Počká na reakci systému



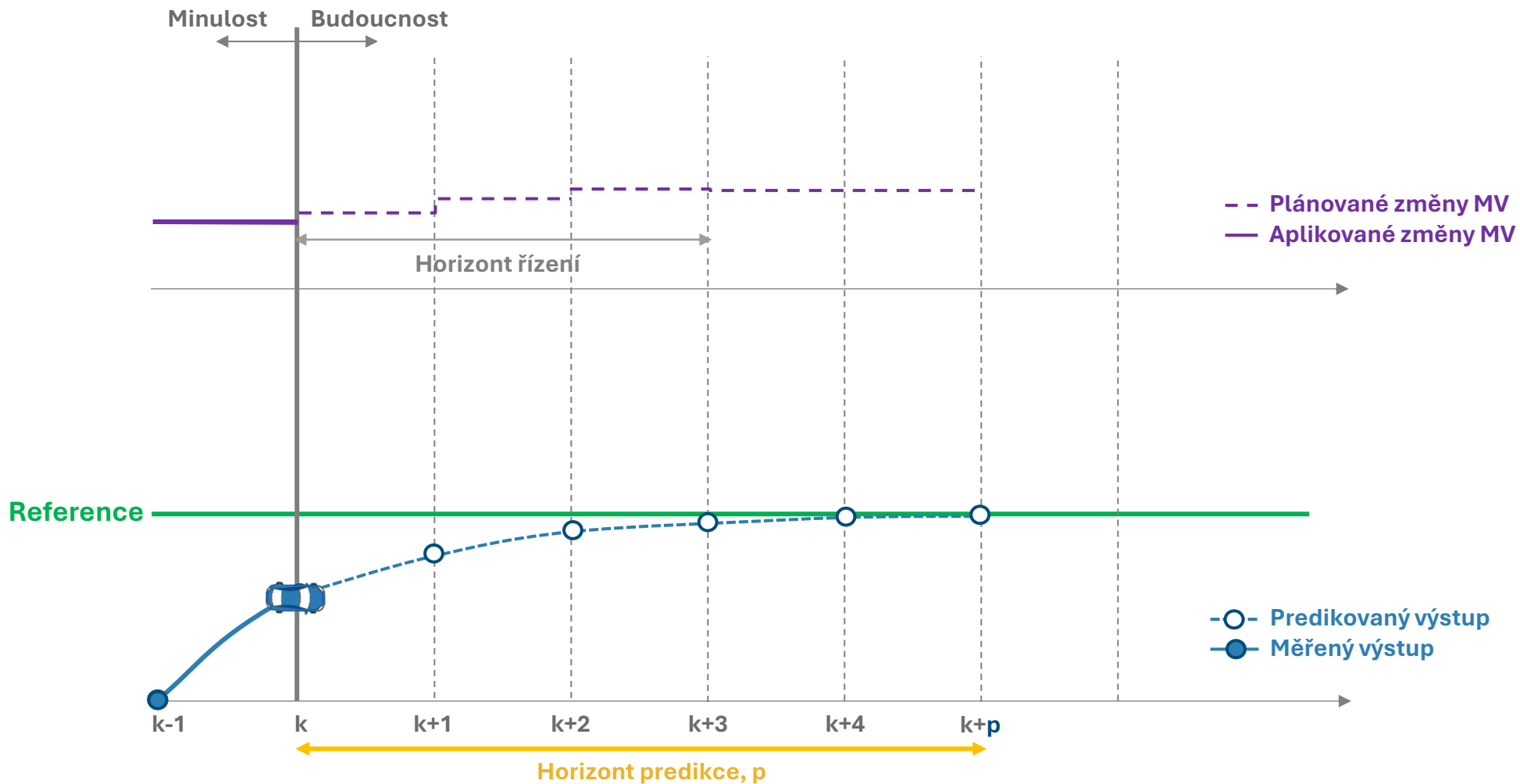
Jak MPC funguje

- Posune časový horizont, změní aktualizované výstupy



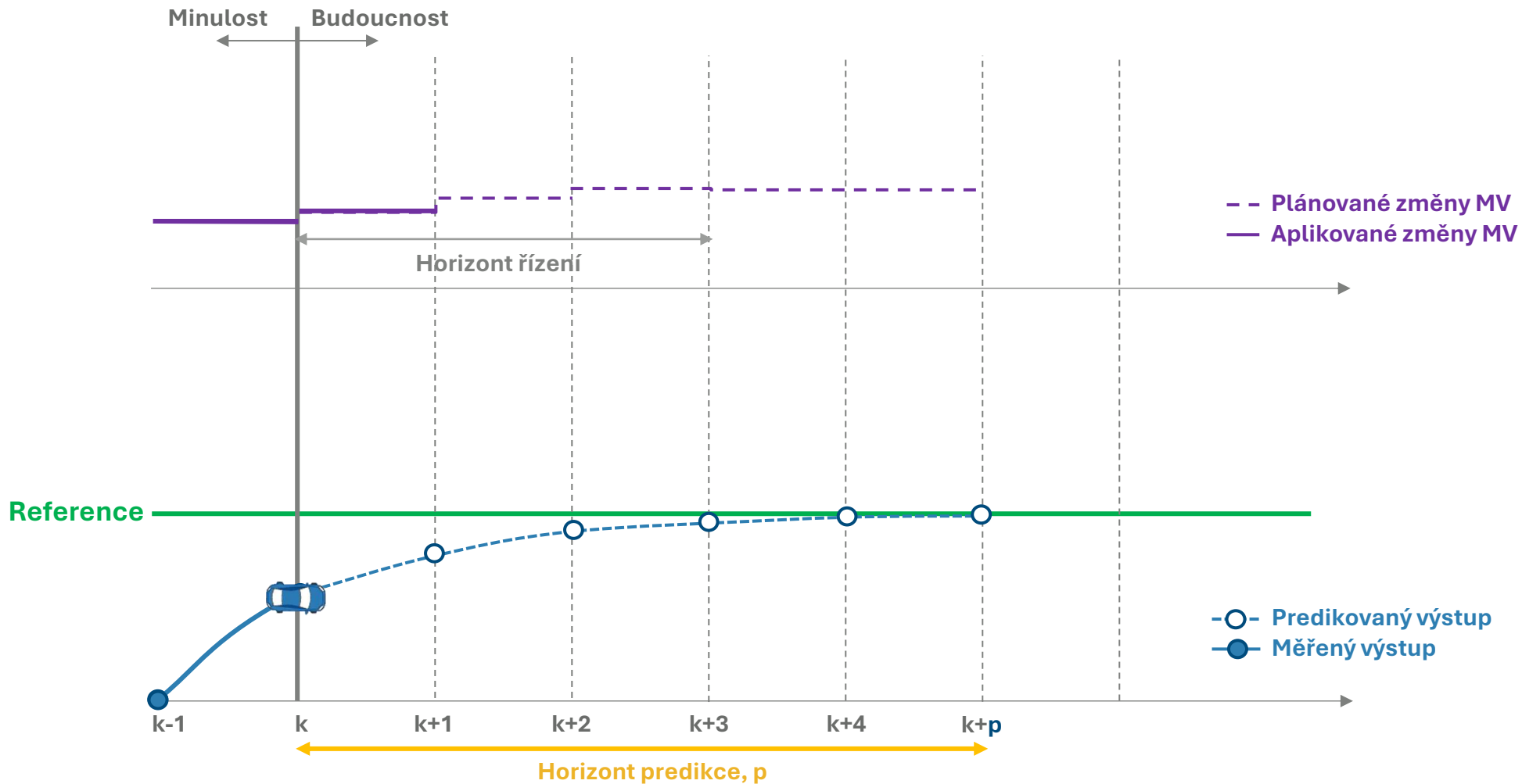
Jak MPC funguje

- Řeší optimalizační úlohu v kroku k



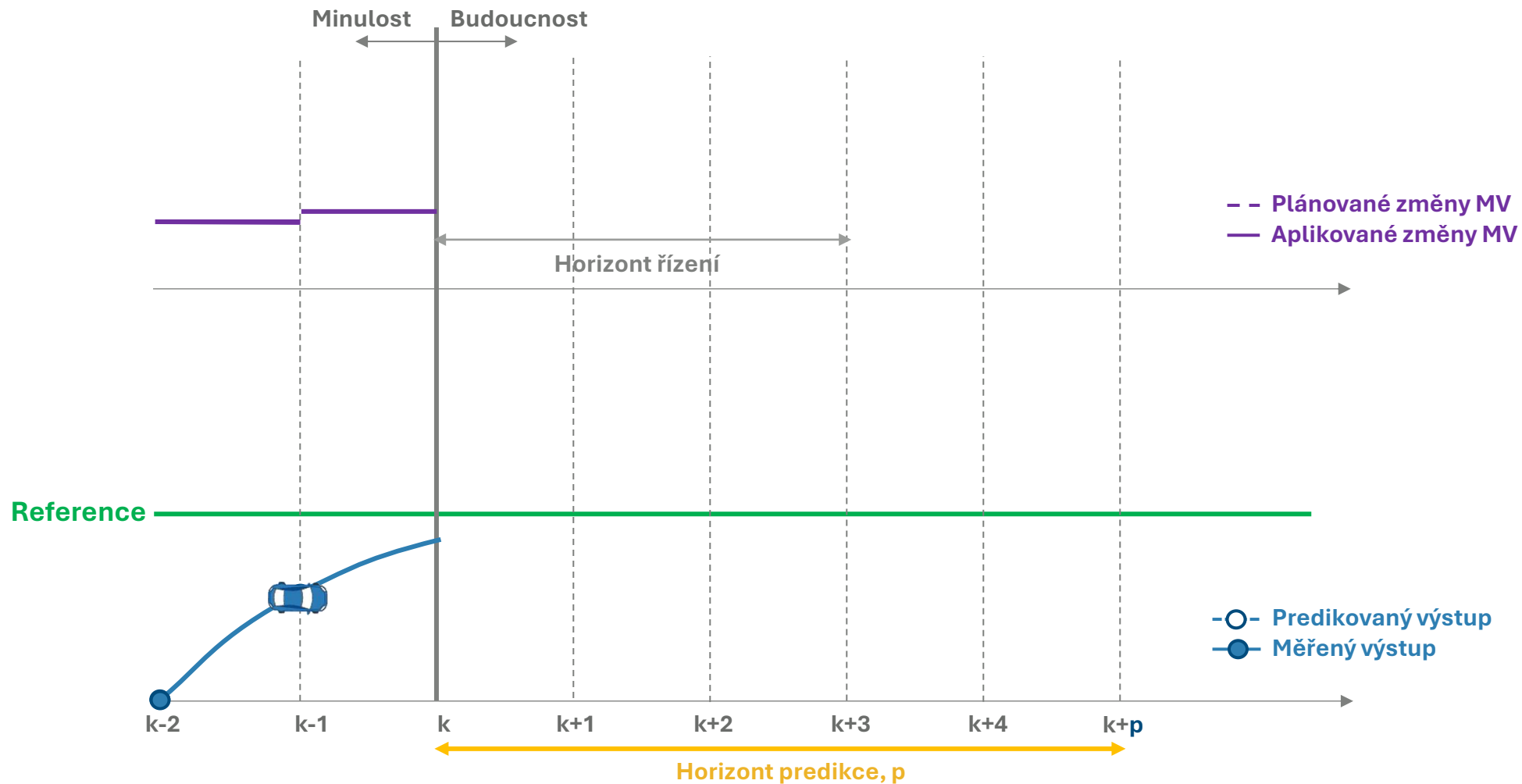
Jak MPC funguje

- Provede první krok řízení (otevřená smyčka), vyřadí zbytek



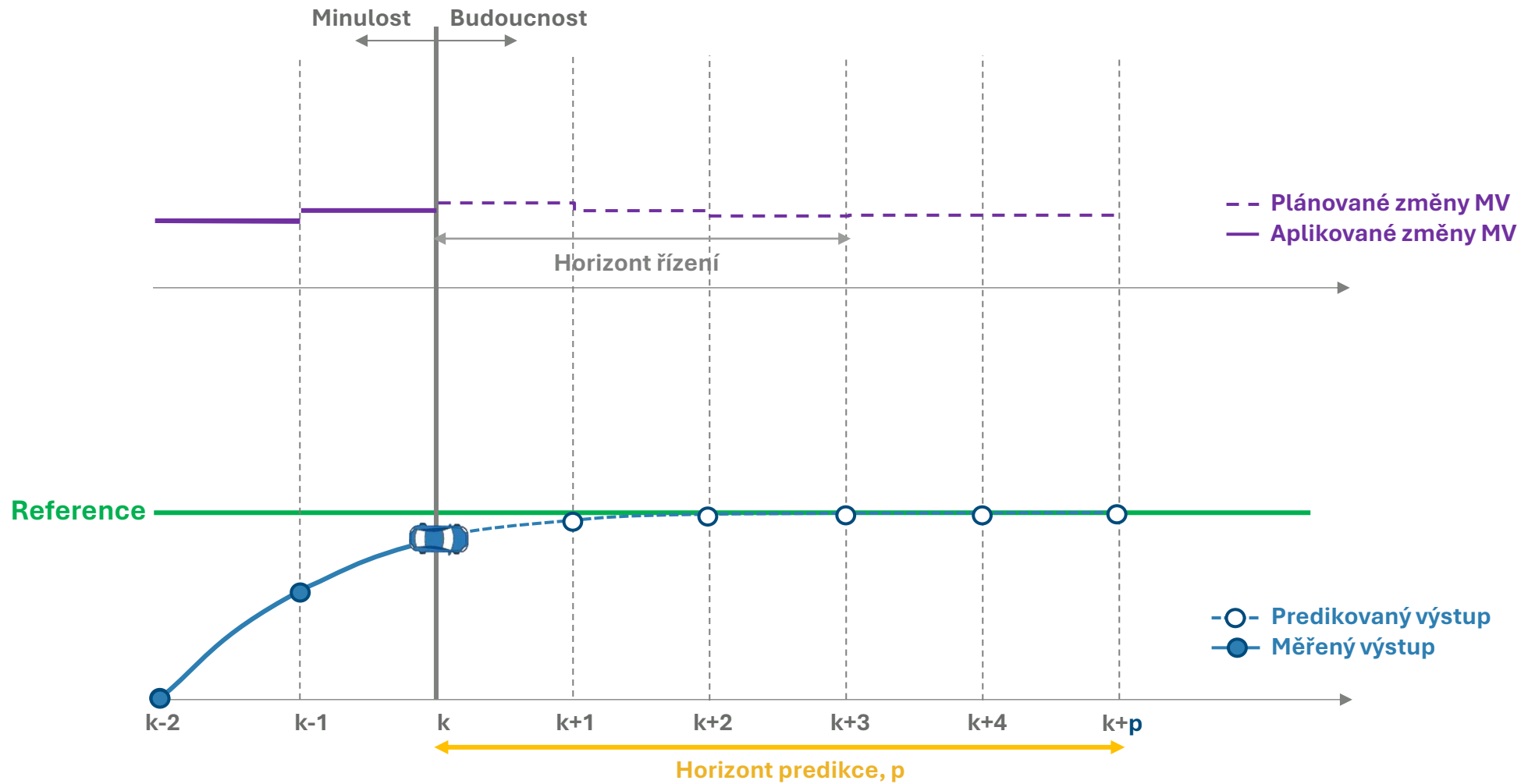
Jak MPC funguje

- Počká na reakci systému, posune časový horizont, změří aktualizované výstupy



Jak MPC funguje

- Řeší optimalizační úlohu v kroku k

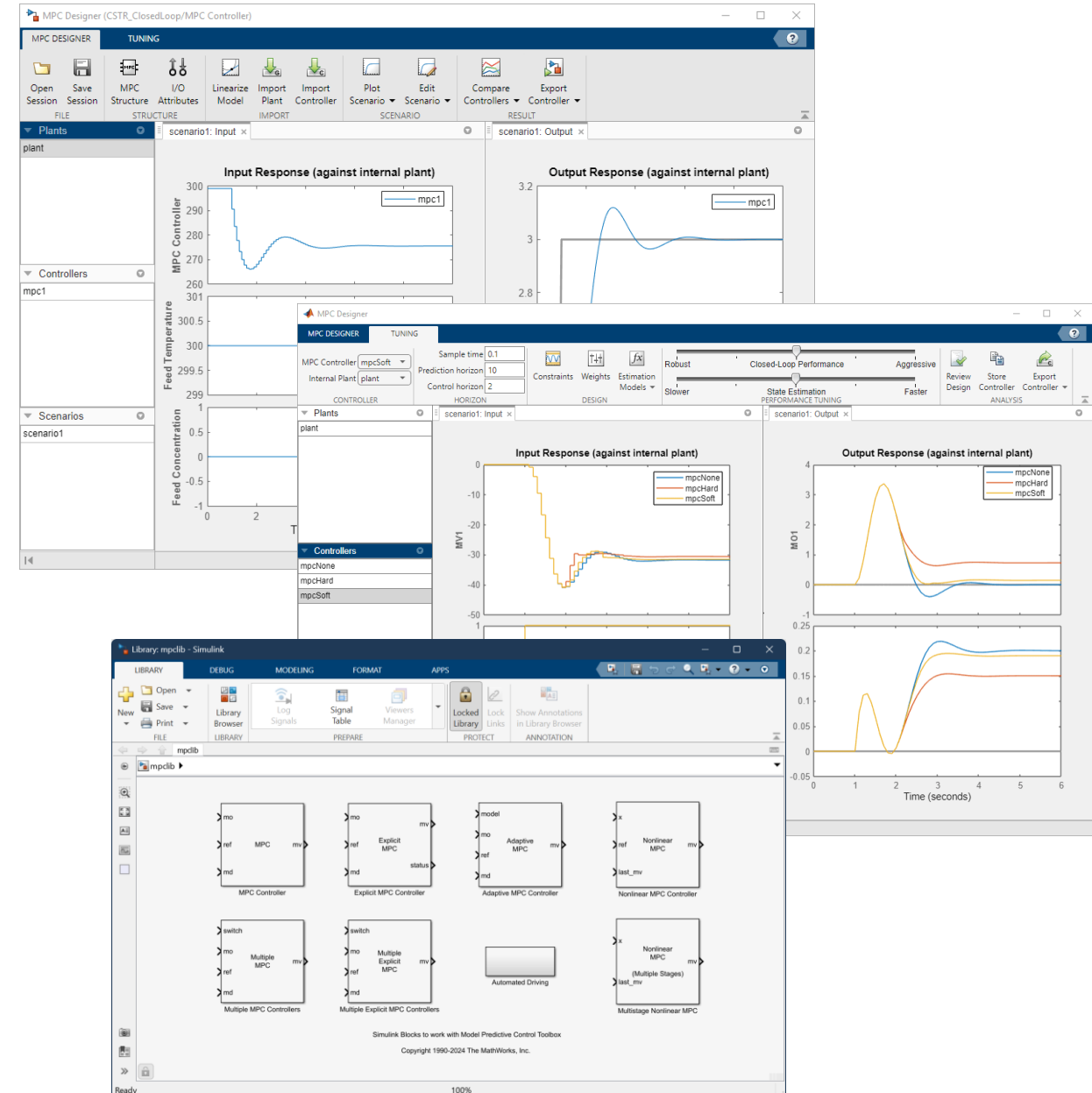


Princip MPC řízení

- MPC algoritmus – co řeší v každém kroku?
 - aktualizace výstupu modelu soustavy dle reálného stavu
 - predikce budoucího výstupu soustavy na základě modelu (na několik kroků dopředu)
 - výpočet optimální hodnoty manipulovaných veličin (QP úloha) pro další kroky
 - cíl: aby predikovaný výstup dosáhnul referenční hodnoty „optimálním způsobem“
 - (další krok = opakování postupu)
- Parametry MPC
 - výpočetní perioda
 - horizont predikce – kolik kroků dopředu počítá predikci
 - horizont řízení – pro kolik kroků dopředu propočítává manipulované veličiny (akční zásah)
 - omezení vstupu a výstupu daná fyzickými parametry systému

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

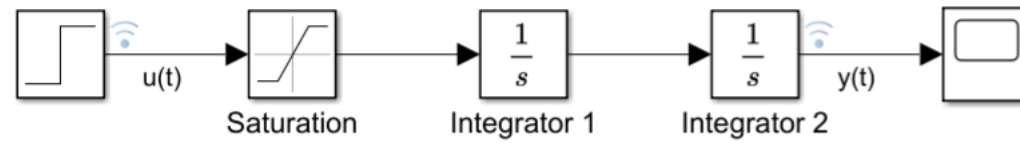


Ukázka: Návrh lineárního implicitního MPC

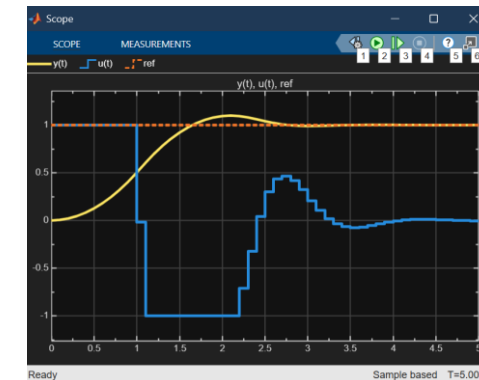
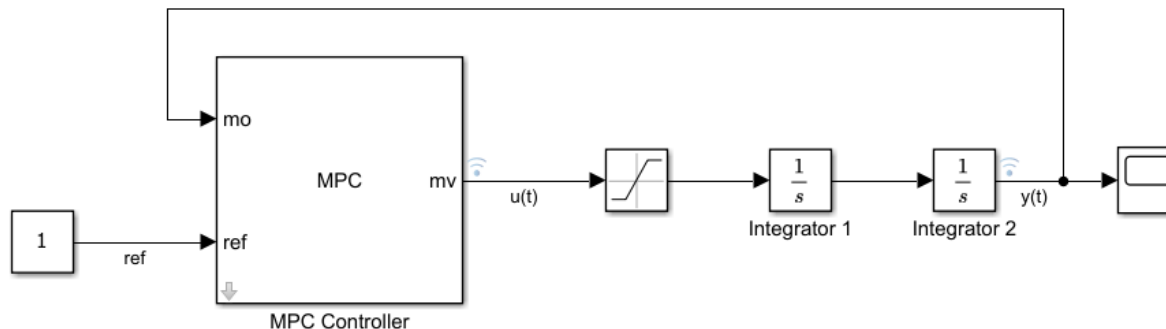
- Řízený systém je popsán diferenciální rovnicí druhého řádu a rozsahem vstupu:

$$\frac{d^2}{dt^2} y(t) = u(t) \quad -1 \leq u(t) \leq 1$$

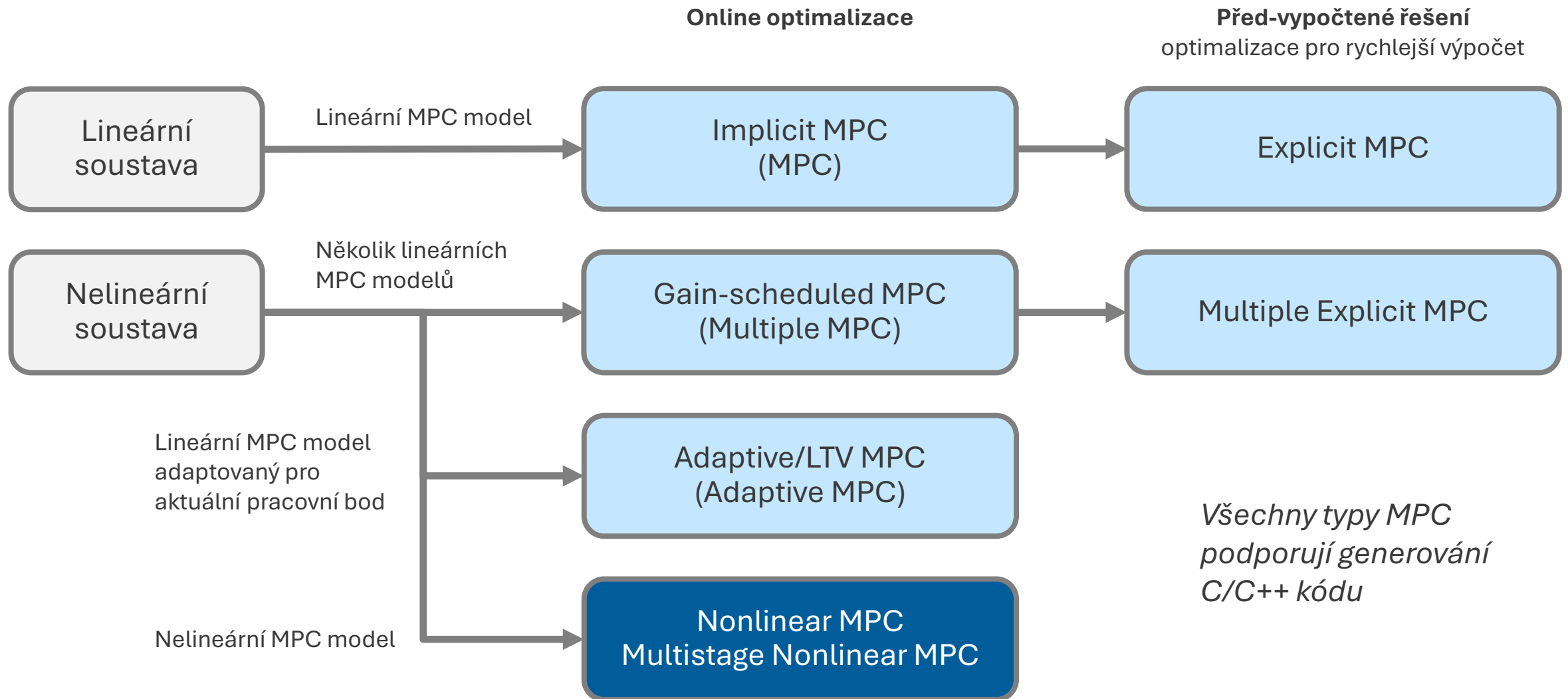
- Model systému v Simulinku: dvojitý integrátor a saturace vstupu



- Zapojení řídicího systému s MPC

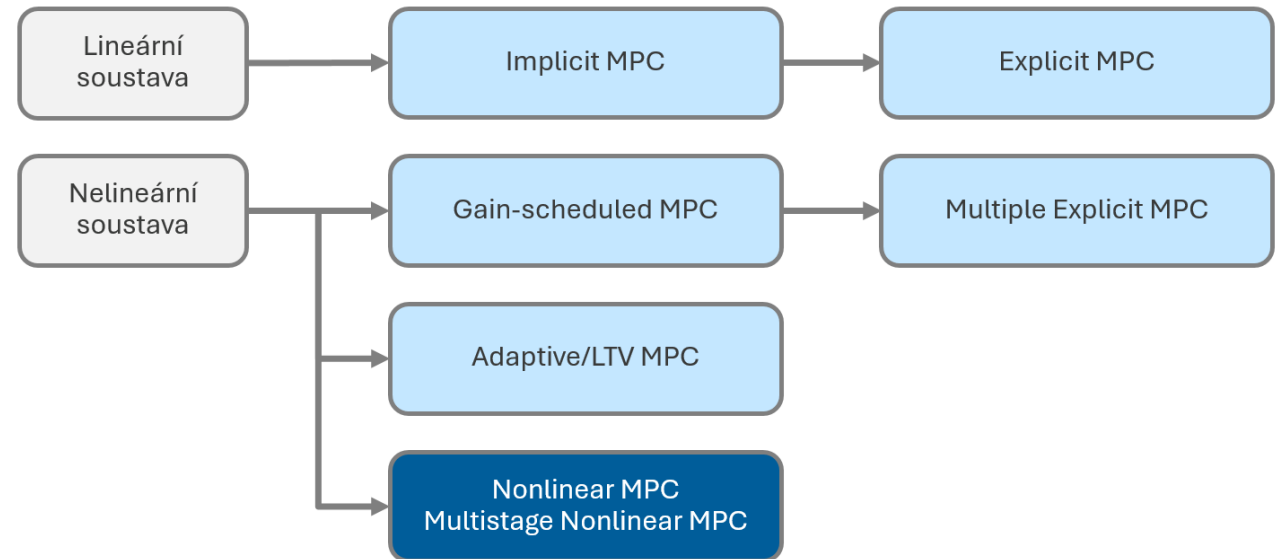


Typy MPC v prostředí MATLAB

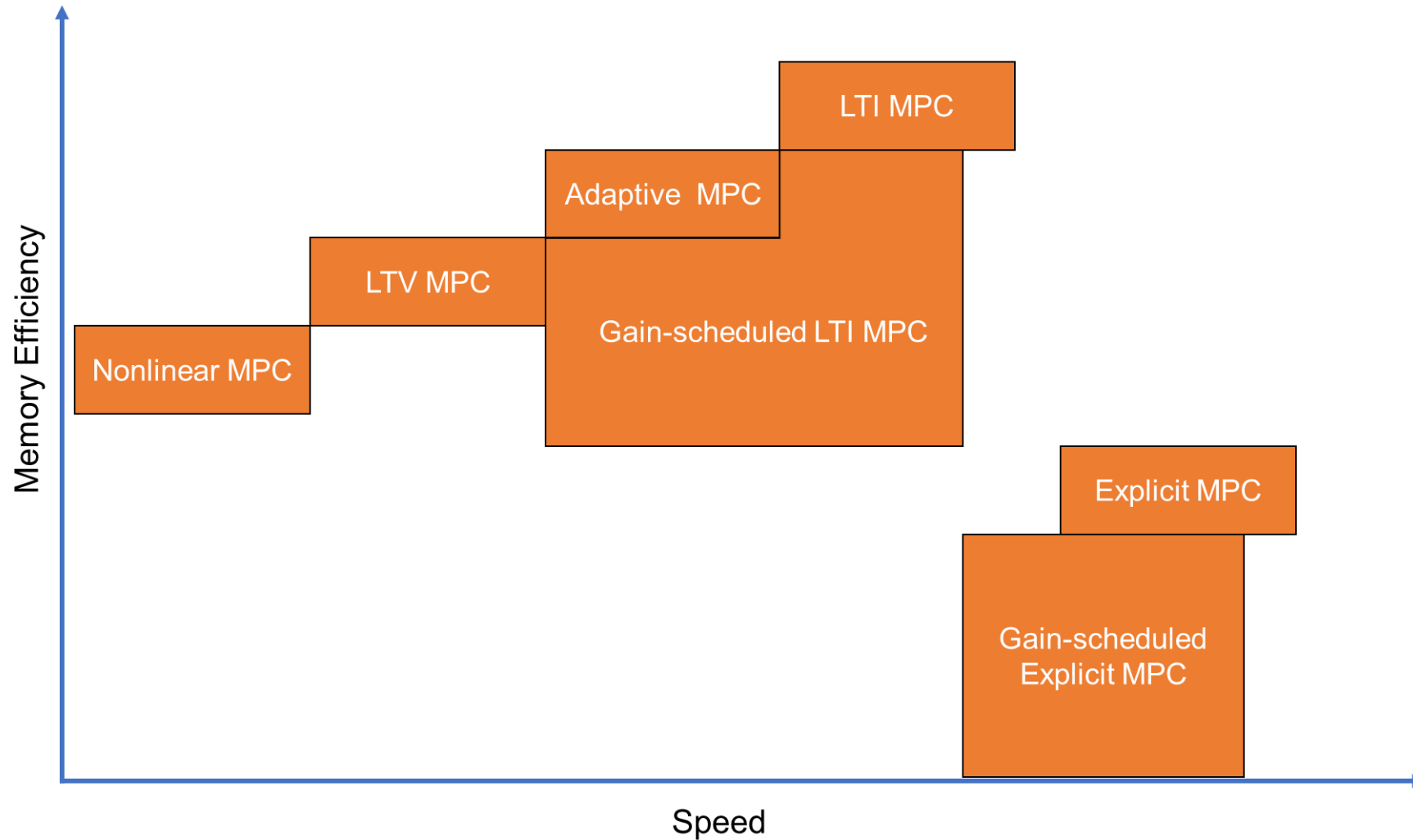


Volba typu MPC dle požadavků

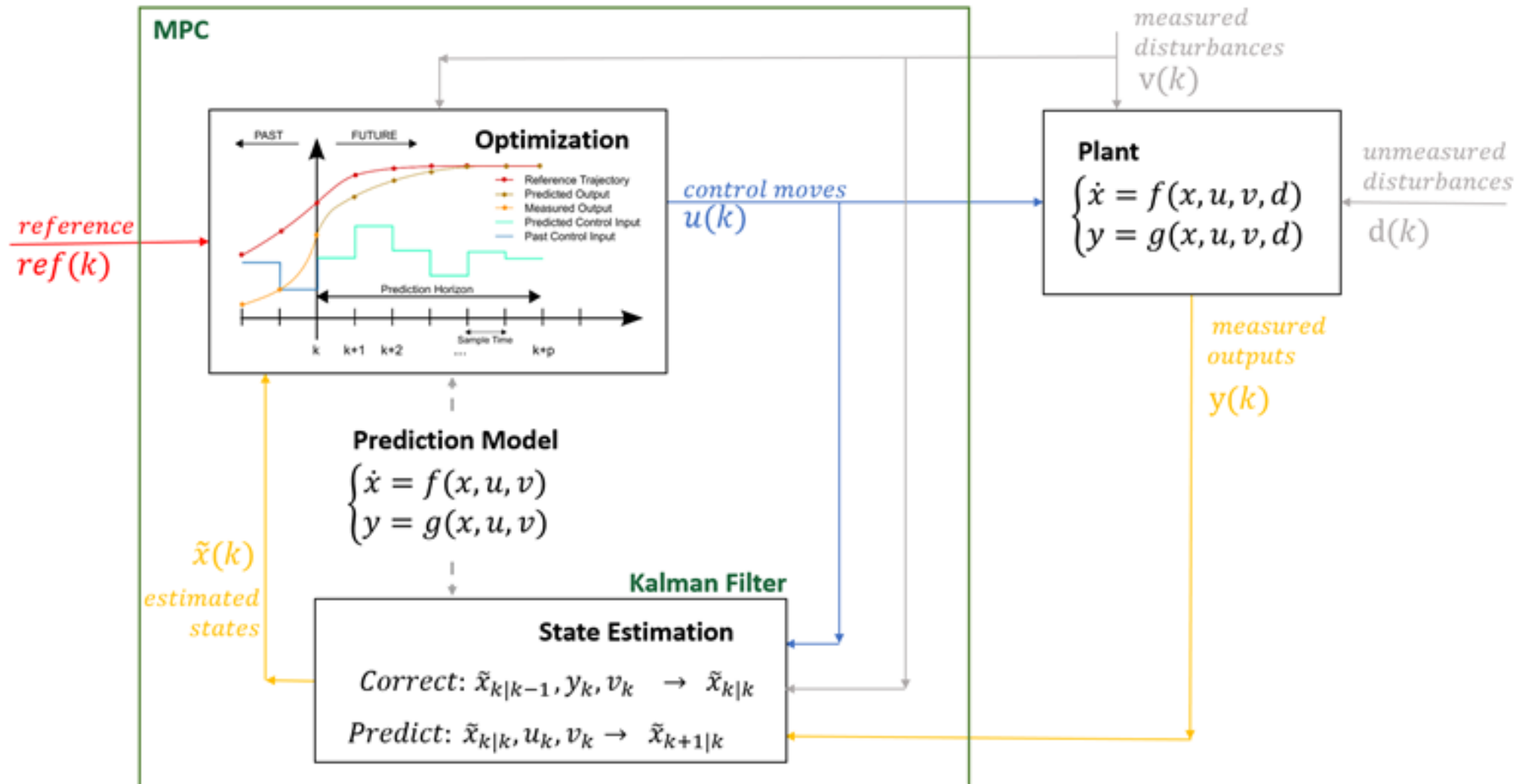
- Model soustavy
 - lineární / nelineární
- Omezení
 - lineární / nelineární
- Účelové funkce
 - kvadratická / ne-kvadratická
- Hardwarové zdroje
 - požadovaná výpočetní perioda, dostupná paměť



Rychlost a paměťové nároky různých typů MPC



Model Predictive Control: pod pokličkou



Lineární MPC = úloha kvadratického programování s omezeními

lineární predikční model

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

kvadratická účelová funkce

$$J = (ref_y - y)^T W_y (ref_y - y) + (ref_u - u)^T W_u (ref_u - u) + (\Delta u)^T W_u (\Delta u)$$

lineární omezení

$$y_{min} \leq y \leq y_{max}, u_{min} \leq u \leq u_{max}, \Delta u_{min} \leq \Delta u \leq \Delta u_{max}$$

kvadratické programování (QP)

$$\min_z \left\{ \frac{1}{2} z^T H z + f^T z \right\}$$

$$A z \leq b$$

$$A_{eq} z = b_{eq}$$

$$lb \leq z \leq ub$$

Nelineární MPC = úloha nelineárního programování s omezeními

nelineární predikční model

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x, u) \end{cases}$$

obecná nelineární účelová funkce

$$J = cost(x, u)$$

obecná nelineární omezení

$$c = ineq(x, u), ceq = eq(x, u)$$

nelineární programování (NLP)

$$\min_z f(z)$$

$$g_i(z) \leq 0, i \in \{1, \dots, m\}$$

$$h_j(z) = 0, j \in \{1, \dots, p\}$$

$$z \in Z$$

Kdy použít MPC vs LQR vs PID

Typ řízení	Založeno na optimalizaci	MIMO	Zahrnut model	Omezení stavu/akčního zásahu	Možnost „náhledu do budoucna“	Nároky na paměť	Výpočetní nároky na běh
MPC	ano	ano	ano	ano	ano	vysoké	vysoké (vyjma Explicit MPC)
LQR/LQG	ano	ano	ano	ne	ne	nízké	nízké
PID	ne	ne	ne	ne	ne	Nízké	nízké

AI model pro predikci: Neural State-Space

- Nelineární MPC může využívat Neural State-Space model

- Co je Neural State-Space

- standardní state-space model

$$\dot{x}_t = f(x_t, u_t)$$

$$y_t = g(x_t, u_t)$$

x ... stav soustavy

y ... měřený výstup

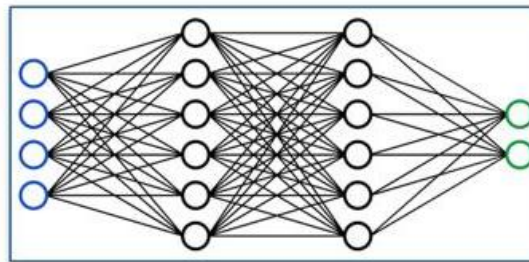
u ... vstup

f, g ... nelineární funkce

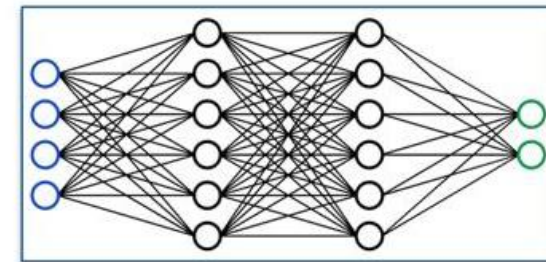
- neural state-space model má funkce f a g reprezentovány neuronovými sítěmi

$$\dot{x}_t = f(x_t, u_t)$$

$$y_t = g(x_t, u_t)$$



State Network (f)



Output Network (g)

AI model pro predikci: Neural State-Space

- Neural State-Space může být i diskretní

- standardní diskretní state-space model

$$x_{k+1} = f(x_k, u_k)$$

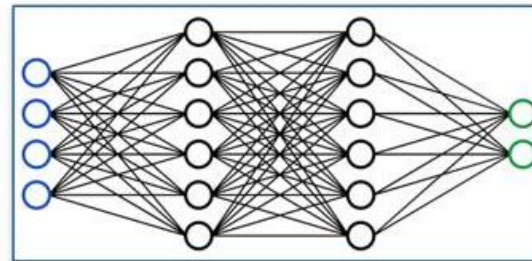
$$y_k = g(x_k, u_k)$$

x ... stav soustavy
 y ... měřený výstup
 u ... vstup
 f, g ... nelineární funkce

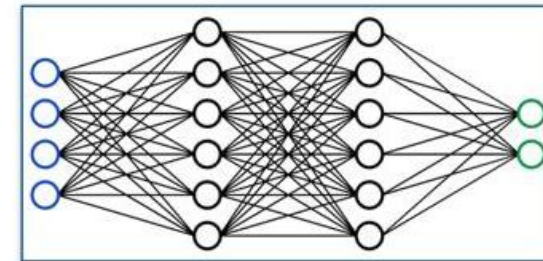
- diskretní neural state-space model má funkce f a g reprezentovány neuronovými sítěmi

$$x_{k+1} = f(x_k, u_k)$$

$$y_k = g(x_k, u_k)$$



State Network (f)



Output Network (g)

Neural State-Space model

- Typická síť pro spojitý NSS

- pro jeden stav, jeden vstup, výstup = stav

```
nss = idNeuralStateSpace(1, NumInputs=1)
```

```
nss =
```

Continuous-time Neural ODE in 1 variables

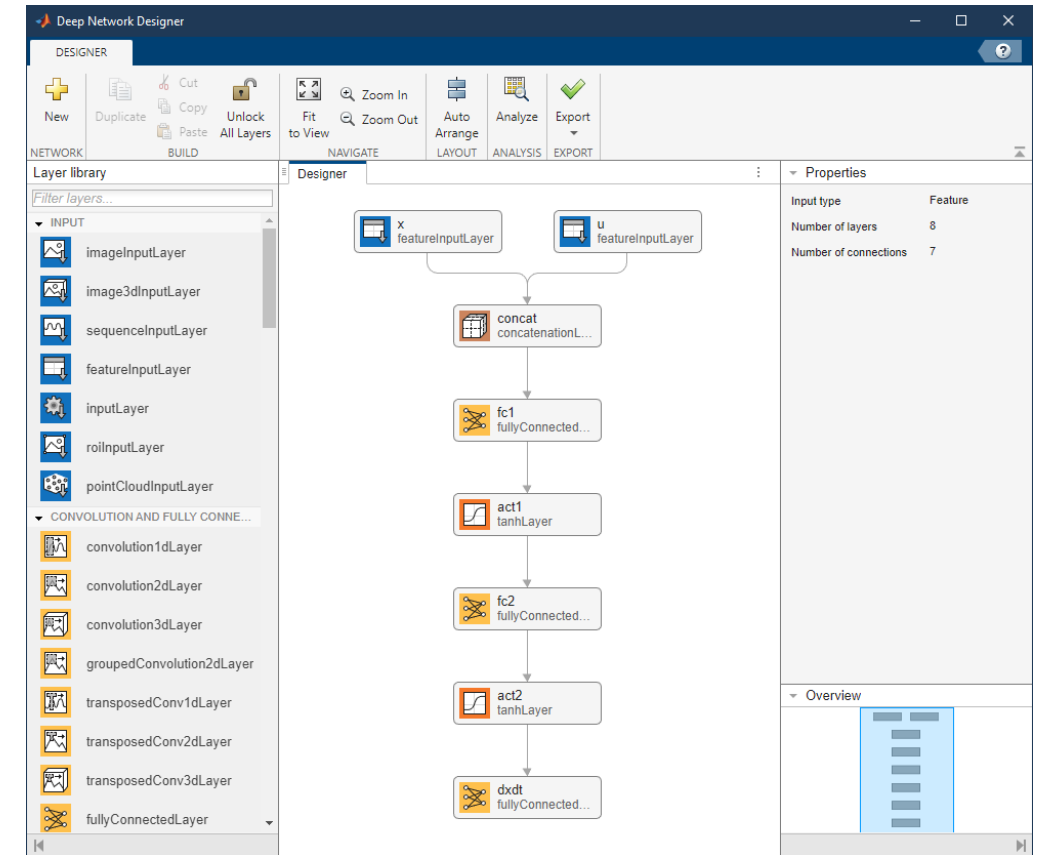
$$dx/dt = f(x(t), u(t))$$

$$y(t) = x(t) + e(t)$$

$f(\cdot)$ network:

Deep network with 2 fully connected, hidden layers

Activation function: Tanh



- Automatizované vytvoření sítí s uživatelskými parametry: `createMLPNetwork()`

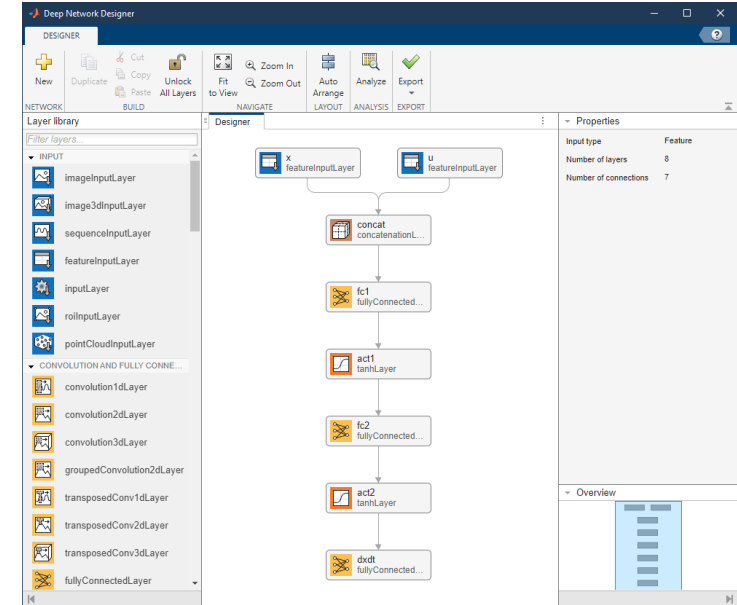
Identifikace Neural State-Space modelu a propojení s MPC

■ Postup identifikace NSS modelu

- získat data z reálné soustavy vstup-výstup: `iddata()`
- vytvořit neural state-space objektu: `idNeuralStateSpace()`
- vytvořit neuronové sítě pro funkce f a g : `createMLPNetwork()`
- nastavit předvolby pro učení sítí: `nssTrainingOptions()`
- identifikovat model (= učení sítí): `nlssest()`

■ Propojení s nelineárním MPC:

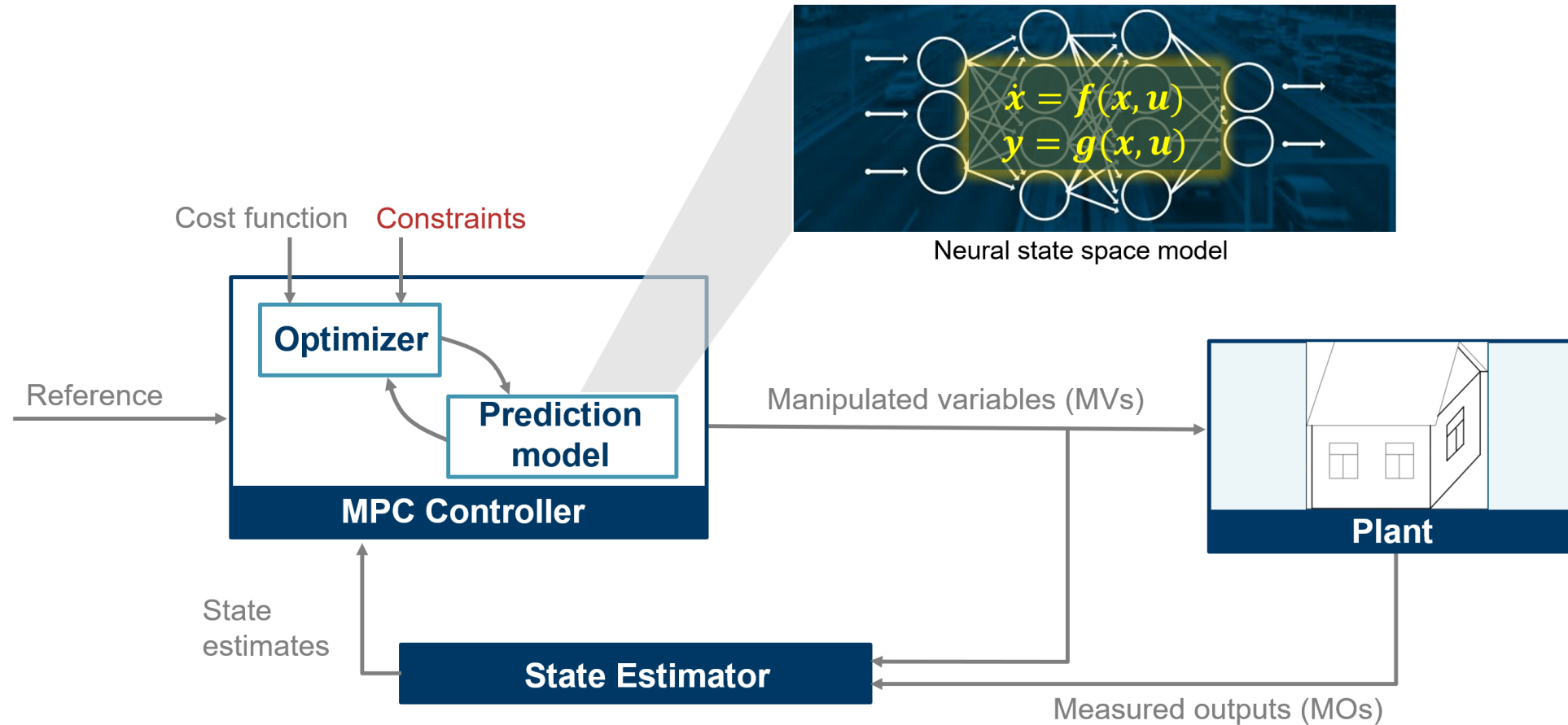
- generovat z NSS funkce pro odhad stavu a výpočet Jacobiánu: `generateMATLABFunction()`
- použít funkce v návrhu MPC



Návrh nelineárního MPC

- Vytvoření objektu nelineárního MPC
 - obecný nelineární MPC: `n1mpc()`
 - vícestupňový nelineární MPC: `n1mpcMultistage()`
- Specifikace predikčního modelu
 - zadání funkcí f a g pro výpočet stavu a výstupu
 - pro lepší výkon specifikovat funkce pro výpočet jejich Jakobiánu
- Zadání optimalizačního kritéria, omezení, řešiče a dalších parametrů
- Volání MPC
 - volání objektu MPC z prostředí MATLAB: `n1mpcmove()`
 - simulace v prostředí Simulink: bloky pro volání MPC různých typů
- Generování kódu z MPC algoritmu a nasazení na cílovou platformu

Ukázka: Řízení systému metodou MPC s AI modelem



Vývoj Model Predictive Control v prostředí MATLAB

≤ 2017	2018	2019	2020	2021	2022	2023
Linear MPC LTI MPC Adaptive MPC LTV MPC Online constraints Online weights Simulink blocks Code Generation MATLAB Coder Simulink Coder Embedded Coder Simulink PLC Coder App MPC Designer	Linear MPC Run-time time-varying weights Nonlinear MPC Generic Nonlinear MPC ADAS Blocks Automatic Cruise Control Lane Keeping Assistant Code Generation ACC block LKA block	Linear MPC Run-time horizons Nonlinear MPC Simulate as Adaptive and LTV MPC ADAS Blocks Path Following Control Code Generation PFC block	Linear MPC Interior-Point QP Solver Run-time time-varying constraints Code Generation Generic Nonlinear MPC Plug in FORCESPRO QP solver	Linear MPC Support “quadprog” Nonlinear MPC Multi-stage nonlinear MPC Hybrid MPC Finite Control Set (FCS) MPC Code Generation Multi-stage nonlinear MPC GPU Coder support ISO 26262 compliance Plug in FORCESPRO NLP solver	Nonlinear MPC Support neural networks for prediction model Passivity-based constraints ADAS Blocks Vehicle Path Planning Code Generation VPP block MISRA C compliance	Nonlinear MPC Automatic differentiation for multistage nonlinear MPC New NLP solver: C/GMRES Code Generation ISO 26262 compliance for multistage nonlinear MPC

V minulých letech byla do MPC Toolboxu přidána řada příkladů, zejména v oblastech nelineárního MPC, ADAS a generování kódu.

Děkuji za pozornost.

Otázky?

Kontakty



- Pro technické dotazy využijte naši technickou podporu
 - email: support@humusoft.cz
 - tel.: +420 602 231 500
 - <https://www.humusoft.cz/matlab/support/>
- Všeobecné dotazy
 - email: info@humusoft.cz
 - tel.: +420 284 011 730
- Nebo můžete využít náš kontaktní formulář
 - <https://www.humusoft.cz/contact/#contact>

Adresa:

Pobřežní 20
186 00 Praha 8
Česká republika

Web:

www.humusoft.cz

