

Řízení a optimalizace

Stavové modely a model-prediktivní řízení

Matematické metody pro ITS (11MAMY)

Jan Příkryl

2. přednáška 11MAMY

úterý 22. března 2021

verze: 2022-03-21 19:06

Ústav aplikované matematiky

ČVUT v Praze, Fakulta dopravní

Popis dynamického systému

Vnější popis systému – Opakování

Vnitřní popis systému

Vnitřní popis nelineárního systému

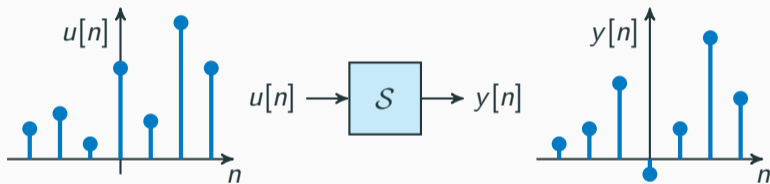
Vnitřní popis lineárního systému

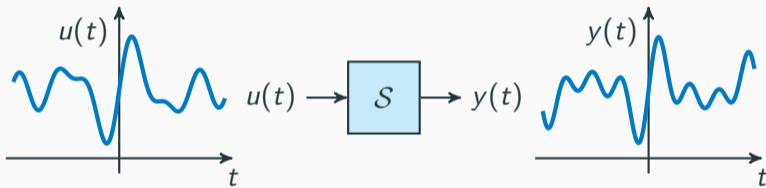
Příklady na stavový popis dynamických systémů

Úrovně modelování

Základy teorie řízení

Model-prediktivní řízení

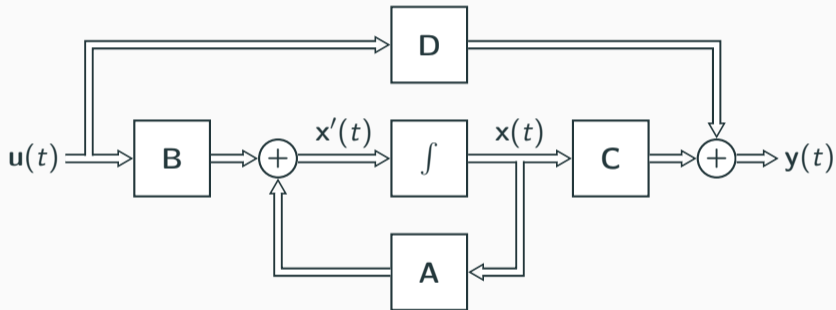


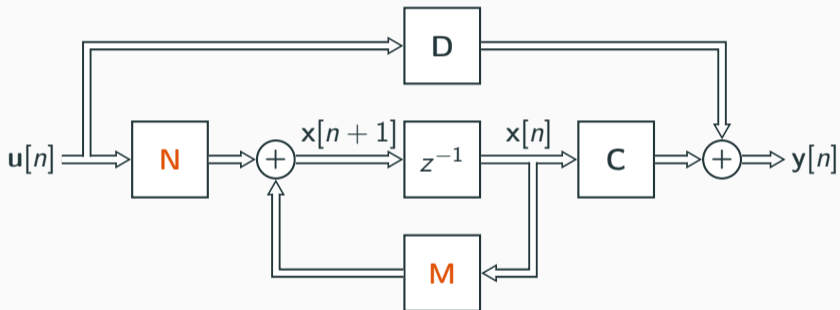


Spojité systém	Diskrétní systém
vstup $\mathbf{u}(t)$	$\mathbf{u}[n]$
stav $\mathbf{x}(t)$	$\mathbf{x}[n]$
výstup $\mathbf{y}(t)$	$\mathbf{y}[n]$
$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$	$\mathbf{x}[n + 1] = \mathbf{f}(n, \mathbf{x}[n], \mathbf{u}[n])$
$\mathbf{y}(t) = \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t))$	$\mathbf{y}[n] = \mathbf{g}(n, \mathbf{x}[n], \mathbf{u}[n])$

Spojitéý systém	Diskrétní systém
$\mathbf{u}(t)$... vstup	$\mathbf{u}[n]$
$\mathbf{x}(t)$... stav	$\mathbf{x}[n]$
$\mathbf{y}(t)$... výstup	$\mathbf{y}[n]$
$\mathbf{x}'(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$	$\mathbf{x}[n+1] = \mathbf{M}[n]\mathbf{x}[n] + \mathbf{N}[n]\mathbf{u}[n]$
$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t)$	$\mathbf{y}[n] = \mathbf{C}[n]\mathbf{x}[n] + \mathbf{D}[n]\mathbf{u}[n]$

Spojité systém	Diskrétní systém
$\mathbf{u}(t)$... vstupní (řídící) vektor	$\mathbf{u}[n]$... vstupní (řídící) vektor
$\mathbf{x}(t)$... stavový vektor	$\mathbf{x}[n]$... stavový vektor
$\mathbf{y}(t)$... výstupní vektor	$\mathbf{y}[n]$... výstupní vektor
$\mathbf{x}'(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t)$	$\mathbf{x}[n+1] = \mathbf{M} \mathbf{x}[n] + \mathbf{N} \mathbf{u}[n]$
$\mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{D} \mathbf{u}(t)$	$\mathbf{y}[n] = \mathbf{C} \mathbf{x}[n] + \mathbf{D} \mathbf{u}[n]$
\mathbf{A} je matice systému ($n \times n$)	\mathbf{M} je matice systému ($n \times n$)
\mathbf{B} je matice vstupů (řízení) ($n \times r$)	\mathbf{N} je matice vstupů (řízení) ($n \times r$)
\mathbf{C} je výstupní matice ($m \times n$)	\mathbf{C} je výstupní matice ($m \times n$)
\mathbf{D} je výstupní matice ($m \times r$)	\mathbf{D} je výstupní matice ($m \times r$)





Popis dynamického systému

Příklady na stavový popis dynamických systémů

Cykloida

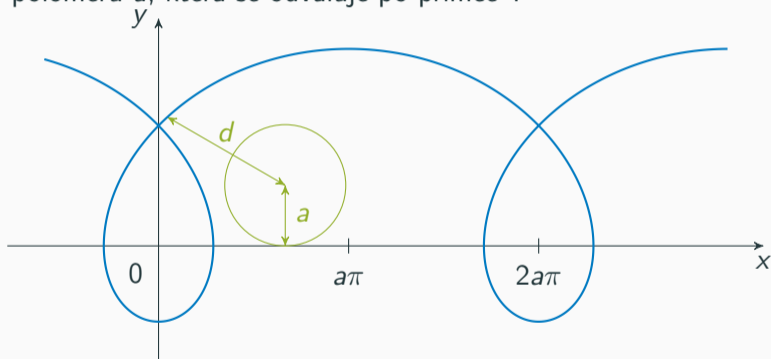
Modely typu predátor-kořist

Úrovně modelování

Základy teorie řízení

Model-prediktivní řízení

Cykloida je křivka, kterou opisuje „bod, pevně spojený ve vzdálenosti d od středu s kružnicí o poloměru a , která se odvaluje po přímce“.¹



¹Lze si ji možná lépe představit tak, že vezmete lahev od piva nebo mléka, na její dno přilepíte špejli a budete sledovat křivku, kterou opíše její konec.

V čase proměnnou polohu bodu $\mathbf{p}(t) = [x(t), y(t)]$ po cykloidě lze popsat parametrickou soustavou rovnic

$$x(t) = x_1(t) = at - d \sin t,$$

$$y(t) = x_2(t) = a - d \cos t,$$

která je pro počáteční podmínky²

$$x_1(0) = 0 \quad \text{a} \quad x_2(0) = a - d$$

dána řešením stavové rovnice

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ a \end{bmatrix} t.$$

²Bod je na počátku děje v dolní úvrati.

Poloha bodu je výstupem dynamického systému

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}t$$

$$\mathbf{p}(t) = \mathbf{x}(t)$$

popsaného maticemi

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ a \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \mathbf{0}.$$

Nelineární stavový model **vlci a ovečky**, který je znám v literatuře jako *Lotka-Volterra predator-prey model*, se týká populace ovcí popsané stavovou proměnnou $x_1(t)$ a populace vlků popsané stavovou proměnnou $x_2(t)$.

Tento dynamický model je autonomní a lze jej popsat nelineární soustavou stavových rovnic ve tvaru

$$\begin{aligned}\frac{d}{dt}x_1(t) &= a x_1(t) - b x_1(t)x_2(t), \\ \frac{d}{dt}x_2(t) &= -c x_2(t) + d x_1(t)x_2(t).\end{aligned}$$

Uvedený model můžeme snadno interpretovat: Žijí-li ovce a vlci odděleně, pro počet ovcí platí rovnice

$$\frac{d}{dt}x_1(t) = a x_1(t),$$

jejímž řešením je exponenciální růst

$$x_1(t) = x_1(0) e^{at},$$

zatímco bez potravy je přírůstek populace vlků záporný

$$\frac{d}{dt}x_2(t) = -c x_2(t)$$

a vlci hynou,

$$x_2(t) = x_2(0) e^{-ct}.$$

Počet sežraných ovcí a nasycených vlků je úměrný počtu jejich setkání – ten je dán součinem

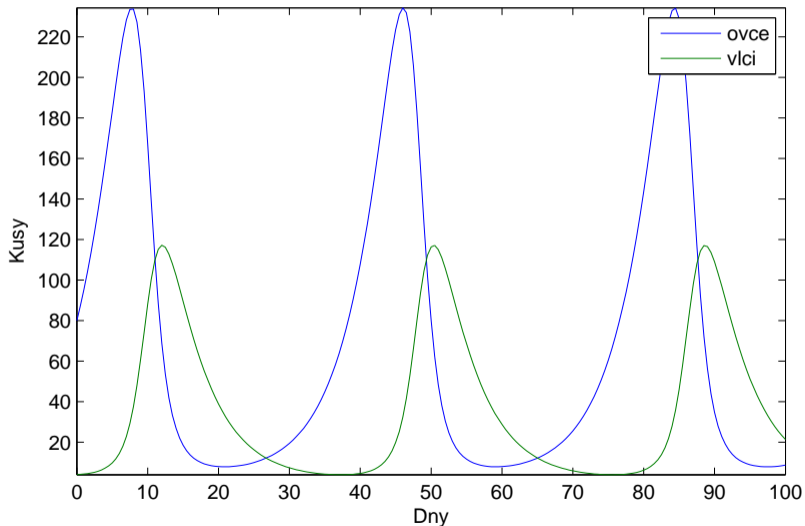
$$x_1(t)x_2(t).$$

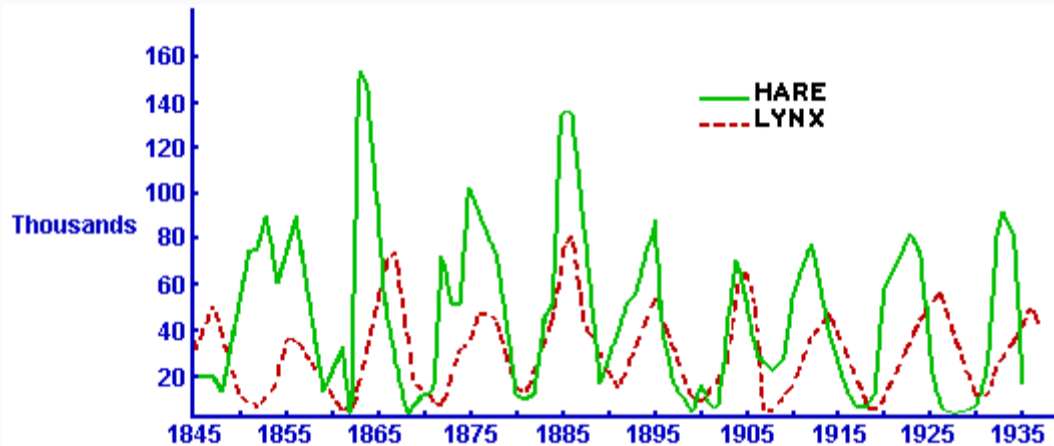
Počet ovcí klesá s rostoucí šancí, že ovce potká vlka:

$$-b x_1(t)x_2(t),$$

a vlci, hodující na ovcích, se mají dobře a jejich počet proto odpovídajícím způsobem stoupá:

$$d x_1(t)x_2(t).$$





Popis dynamického systému

Příklady na stavový popis dynamických systémů

Úrovně modelování

Přesný model systému

Model systému z naměřených dat

Gray-box

Základy teorie řízení

Model-prediktivní řízení

V některých případech jsme schopni přesně matematicky popsat chování celého systému.

V případě složitých systémů s mnoha neurčitými vazbami to však tak jednoduché není.

Podle toho, kolik informací o modelovaném systému máme, lze modely dělit do tří skupin:

- white-box
- gray-box
- black-box

Zcela odvozen ze základních zákonů fyziky, chemie, ekonomie, . . . (first-principle models).

Všechny rovnice a parametry lze určit na teoretické úrovni.

Také kombinace modelů, i v případě, že některé parametry odhadnuty z dat.

Charakteristická vlastnost: nezávislé na datech, parametry přímo interpretovatelné jako základní veličiny (hmotnost, rychlost, národní důchod, . . .).

Zcela závislé na měřených datech.

Jak **struktura modelu**, tak i **parametry** jsou odvozeny experimentálně.

Nevyužíváme žádnou apriorní informaci o modelovaném systému.

Parametry typicky nemají žádný vztah k základním veličinám.

Algoritmy strojového učení s učitelem: Známe vstupní a výstupní data, použijeme generický model, jehož parametry nastavíme.

Vyžaduje trénovací a testovací data

Velmi rozšířené v inženýrské praxi

Příklad: AR (autoregresní modely), ANN (neuronové sítě), SVM (metoda podpůrných vektorů, angl. *support vector machines*), GP (Gaussovské procesy)

Dělíme na

- parametrické
- neparametrické

Parametrické modely předpokládají existenci **konečné množiny parametrů** θ . Jakmile jednou najdeme parametry, budoucí výstup modelu pro vstupy x nezávisí na množině právě pozorovaných dat \mathcal{D} , model lze tedy reprezentovat pravděpodobnostním rozdělením odpovídajícím

$$P(x|\theta, \mathcal{D}) = P(x|\theta).$$

Složitost modelu je ohraničená, daná θ , i když množství pozorovaných dat ohraničené není.

Tyto modely **nejsou flexibilní**, všechna informace o \mathcal{D} je obsažena v θ .

Neparametrické modely nepředpokládají, že data lze reprezentovat distribuční funkcí založenou na konečné množině parametrů. Mnohdy je ale definujeme ze předpokladu, že θ má nekonečnou dimenzi. Typicky je θ nějaká funkce.

V tomto případě může θ obsáhnout rostoucí objem informace o datech tak, jak \mathcal{D} roste.

Tyto modely jsou složitější, ale zato **jsou flexibilní** – dokážou brát v úvahu i evidenci obsaženou v nově přicházejících datech.

Jednoduchý (a často jediný použitelný) nástroj na modelování složitých vztahů v datech.

Příklady black-box modelů

Parametrické	Neparametrické	Aplikace
polynomiální regrese	Gaussovský proces	aproximace fcí
logistická regrese	GP klasifikátor	klasifikace
směsové modely, k -means	směs Dirichletovských procesů	shlukování
skryté Markovské modely (HMM)	nekonečné HMM	časové řady
faktorová analýza, PCA, PMF	nekonečné latentní FM	hledání příznaků

Dnes také velmi oblíbené neuronové sítě \Rightarrow BIG DATA!

Kompromis / kominace mezi white-box a black-box modelem. Možné jsou všechny možné kombinace.

Jeho popis může obsahovat také kvalitativní informace o modelovaném systému, např. popis chování systému ve formě pravidel.

Charakteristika: slučuje všechny možné snadno dostupné zdroje informací o systému.

Struktura modelu bývá odvozena z expertních znalostí, **parametry** modelu jsou ale určeny z dat.

	White-box	Gray-box	Black-box
Zdroje informací	základní principy znalosti	kvalitativní znalosti pravidla částečné znalosti a data	experimenty data
Vlastnosti	dobrá extrapolace dobré pochopení vzoká spolehlivost škálovatelnost		krátká doba vývoje nevyžaduje expertní znalosti lze i pro neznámé procesy
Nevýhody	časově náročné vyžaduje znalosti znalosti omezují přesnost pouze pro známé procesy		nelze extrapolovat není škálovatelné přesnost omezena daty málo pochopení
Aplikační oblasti	plánování konstrukce, design spíše jednoduché procesy		pouze pro existující procesy spíše složité procesy

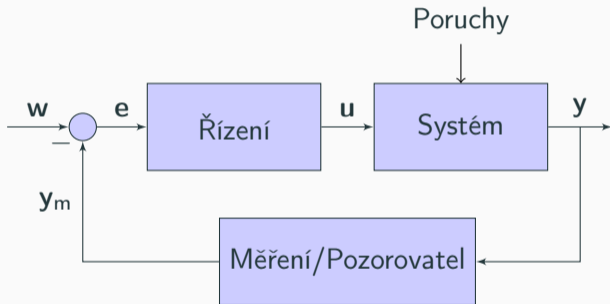
Popis dynamického systému

Příklady na stavový popis dynamických systémů

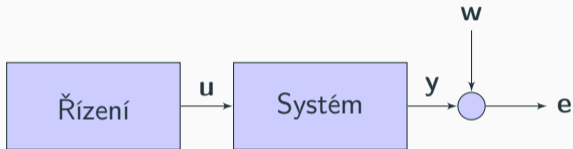
Úrovně modelování

Základy teorie řízení

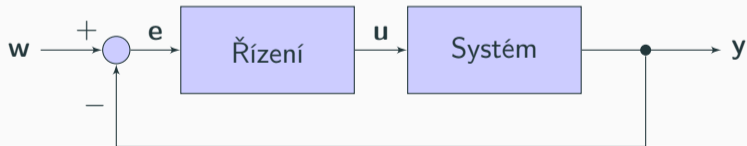
Model-prediktivní řízení



Ovládání:



Regulace:



Definice (Dosažitelnost)

Stav x je dosažitelný, existuje-li řízení $u(t)$, které za konečný čas převede počáteční stav $x(t_0) = 0$ do stavu x . Jsou-li všechny stavy systému dosažitelné, říkáme, že **systém je dosažitelný**.

Definice (Říditelnost stavu)

Stav x je říditelný, existuje-li řízení $u(t)$, které v konečném čase převede tento stav do počátku (do nulového stavu). Jsou-li všechny stavy systému říditelné, říkáme, že **systém je říditelný**.

Definice (Říditelnost na výstupu)

Systém je říditelný na výstupu, pokud existuje funkce řízení $u(t)$ taková, že převede výstup $y(t_0)$ na $y(t_1)$ v konečném čase $t_1 - t_0$.

Definice (Pozorovatelnost)

Systém je pozorovatelný, když změřením vstupu a výstupu na konečném časovém intervalu je možno určit hodnotu stavu systému na počátku měření.

Měřením lze zjistit vlastnosti pouze u pozorovatelné části systému.

Definice (Pozorovatel stavu)

Pozorovatel stavu je systém, který poskytuje odhad vnitřního stavu daného reálného systému z měření vstupu a výstupu reálného systému.

Pozorovatel je obvykle implementován počítačem a poskytuje základ pro řadu praktických aplikací.

Typické úlohy

- Kompenzace vlivu poruchových veličin
(angl. *Disturbance Rejection*)
- Problém regulátoru
(angl. *Regulator Problem*)
- Problém sledování
(angl. *Tracking Problem*)
- Optimální řízení
(angl. *Optimal Control*)

Na systém vždy působí celá řada poruchových veličin.

Poruchová veličina **měřitelná** \Rightarrow její vliv lze kompenzovat v řídicím členu. Někdy lze vliv poruchy kompenzovat beze zbytku.

Řízený systém je invariantní vzhledem k poruše: porucha se na výstupu systému vůbec neprojeví.

Dynamické vlastnosti samotného systému jsou někdy nevyhovující.

Účelem je navrhnout takovou strukturu řízení, aby celý systém měl vyhovující dynamické vlastnosti.

Často jde o **stabilizaci** systému, jenž je nestabilní.

Požadujeme, aby $y(t)$ co nejlépe sledovalo referenční průběh $w(t)$.

Minimalizace regulační odchylky $e(t)$

$$e(t) = w(t) - y(t)$$

Dodatečná omezení:

- velikost řídicí veličiny,
- velikost celkové řídicí energie, ...

V moderní teorii řízení jsou všechny požadavky na řízení shrnuty do **kritéria kvality řízení** $J = f(\mathbf{x}, \dots)$.

Návrh řídicího zásahu \rightarrow optimalizační problém minimalizace hodnoty J .

Dva zásadní problémy:

- volba kritéria kvality řízení,
- řešitelnost takto formulovaného optimalizačního problému.

Nejvíce používaným kritériem kvality řízení je **kvadratické kritérium**

$$J = \mathbf{x}^T \mathbf{R} \mathbf{x} + \dots$$

Pro lineární systémy vede na **lineární zákon řízení** $\mathbf{u} = -\mathbf{K} \mathbf{x}$, kde \mathbf{K} je matice, kterou lze spočítat z tzv. *Ricattiho rovnice*.

Optimalizace: Snaha dosáhnout určitého cíle (minimalizace nákladů, maximalizace výtěžku, minimalizace cestovní doby)

Definice (Bellmanův princip optimality)

Optimální řídicí strategie má tu vlastnost, že bez ohledu na počáteční stav a původní rozhodnutí představují ostatní rozhodnutí opět optimální strategii s ohledem na stav, vyplývající z prvního rozhodnutí.

Lze tedy postupovat zpětně (tzv. angl. *backtracking* nebo angl. *rollout*)

Bellmanův princip optimality vede na strategii zvanou **dynamické programování**:

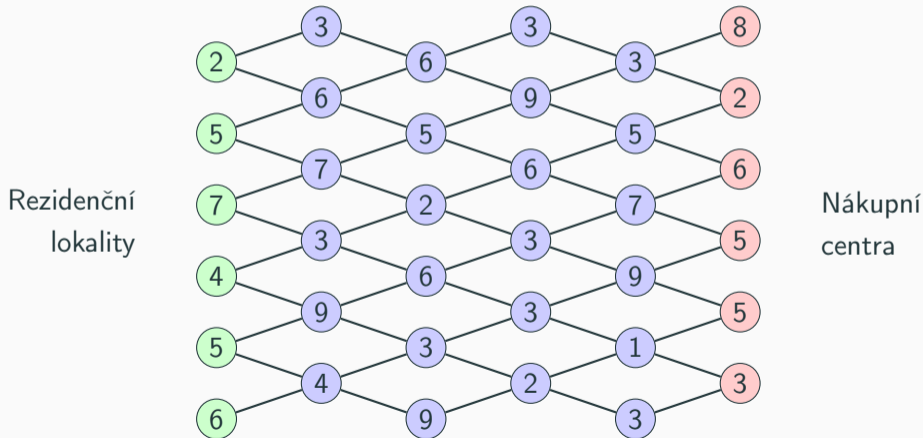
Definice (Dynamické programování)

Dynamické programování (DP) je postup pro řešení úloh, jejichž podúlohy se překrývají, t.j. řeší stejné podúlohy opakovaně. Metody DP si hodnoty dříve vyřešených podúloh zapamatuje a při jejich znovuobjevení je již neřeší znovu.

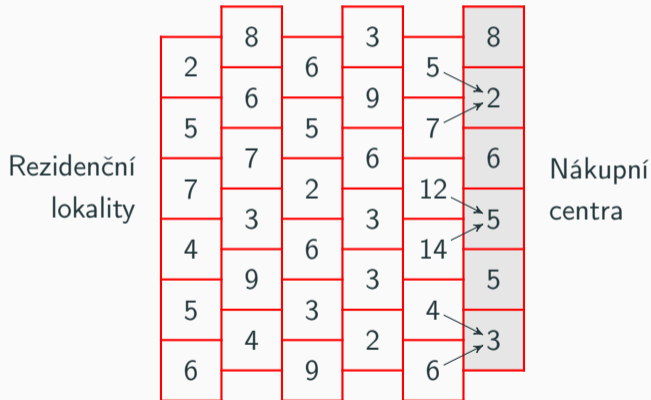
Přístupy:

- shora dolů (rekurze)
- zdola nahoru

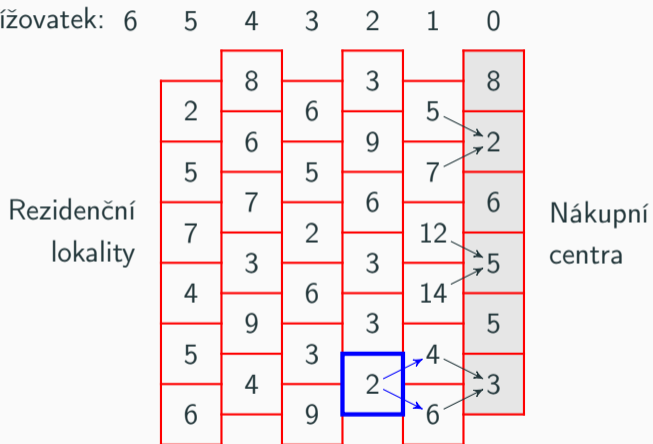
Hledáme nejrychlejší cesty z domů vlevo k nákupním střediskům vpravo. Uzly grafu jsou křižovatky, ohodnocení uzlů je zpoždění.



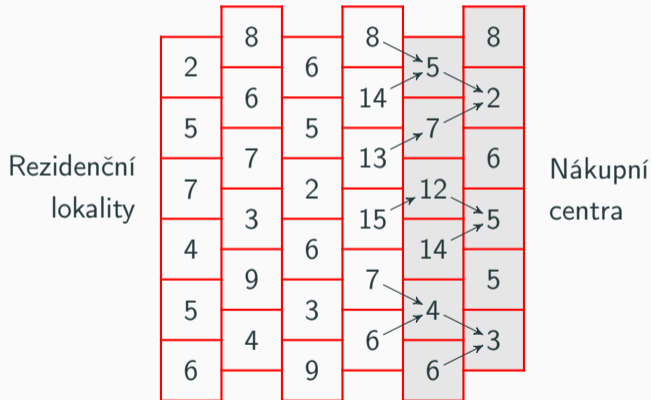
Zbývá křížovatek: 6 5 4 3 2 1 0



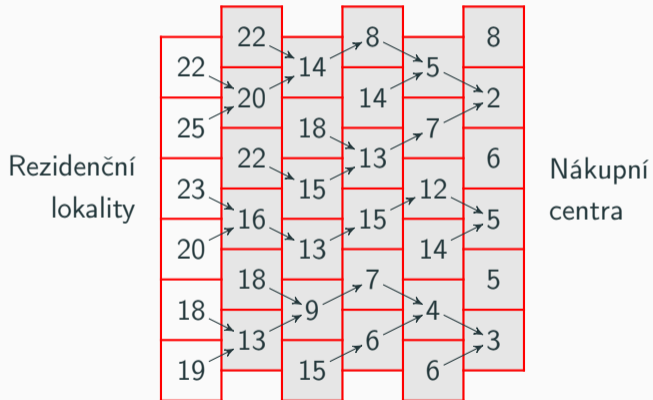
Zbývá křížovatek: 6



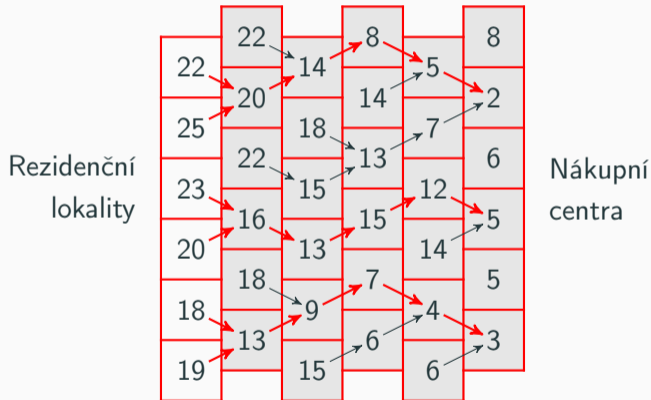
Zbývá křížovatek: 6 5 4 3 2 1 0



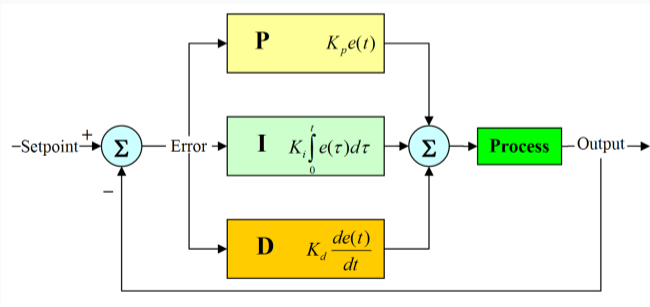
Zbývá křížovatek: 6 5 4 3 2 1 0



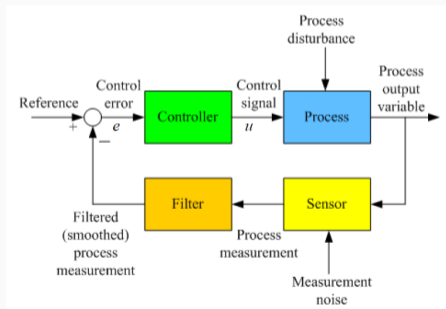
Zbývá křížovatek: 6 5 4 3 2 1 0



$$u(t) = u_0 + \underbrace{K_p e(t)}_{u_p} + \underbrace{\frac{K_p}{T_i} \int_0^t e(\tau) d\tau}_{u_i} + \underbrace{K_p T_d \frac{d}{dt} e(t)}_{u_d}$$



$$u(t) = u_0 + \underbrace{K_p e(t)}_{u_p} + \underbrace{\frac{K_p}{T_i} \int_0^t e(\tau) d\tau}_{u_i} + \underbrace{K_p T_d \frac{d}{dt} e(t)}_{u_d}$$



Popis dynamického systému

Příklady na stavový popis dynamických systémů

Úrovně modelování

Základy teorie řízení

Model-prediktivní řízení

Idea: Použijeme model systému k posouzení budoucích dopadů současných akcí.

- dnes velmi rozvinuté regulační paradigma
- základ: 1970tá léta, Shell Oil, petrochemie
- důvod: úspora nákladů, lze se více přiblížit mezním parametrům

V každém okamžiku vzorkování k používáme

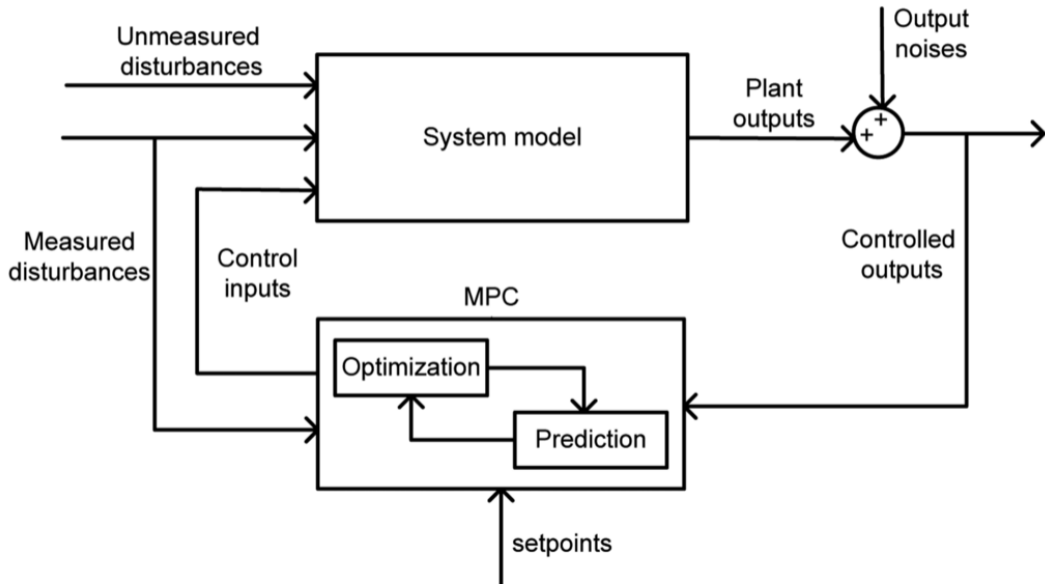
- a) matematický model řízené soustavy
dynamický model, omezující podmínky
- b) (konečnou) historii hodnot regulovaných veličin až po k -tou
- c) (konečnou) historii hodnot řízení
- d) požadovaný průběh regulovaných veličin
v rámci uvažovaného horizontu predikce h

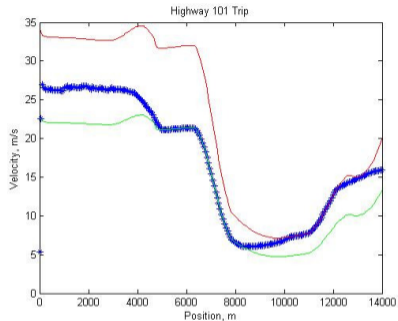
Máme:

- stavový vektor v kroku k , $\mathbf{x}[k]$,
- diskrétní model
$$\begin{cases} \mathbf{x}[k+1] = f(k, \mathbf{x}[k], \mathbf{u}[k]) \\ \mathbf{y}[k] = g(k, \mathbf{x}[k], \mathbf{u}[k]) \end{cases}$$

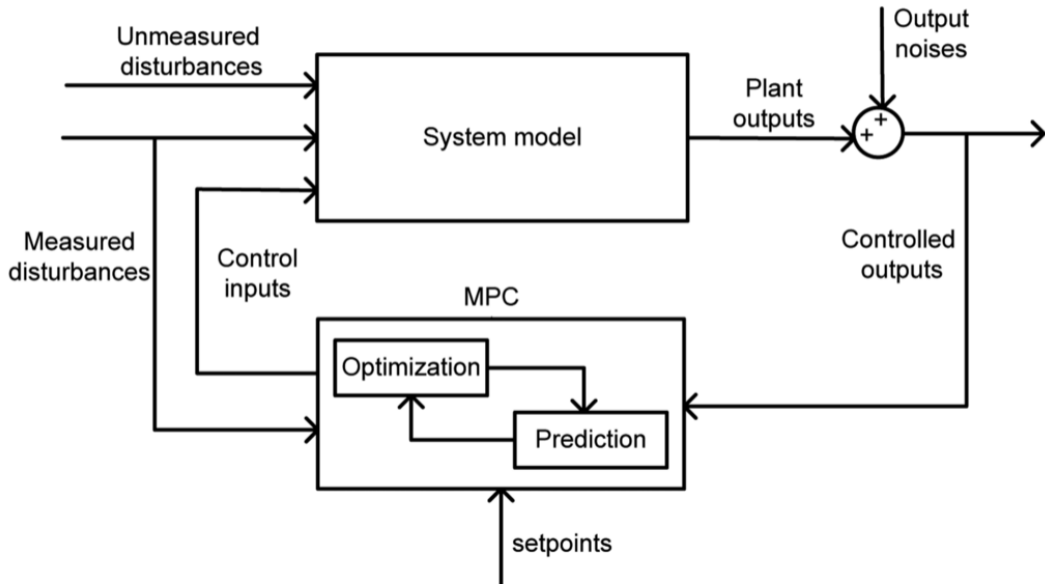
Vektor vstupů $\mathbf{u}[k]$ minimalizuje ztrátovou funkci $J(\mathbf{x}[k])$

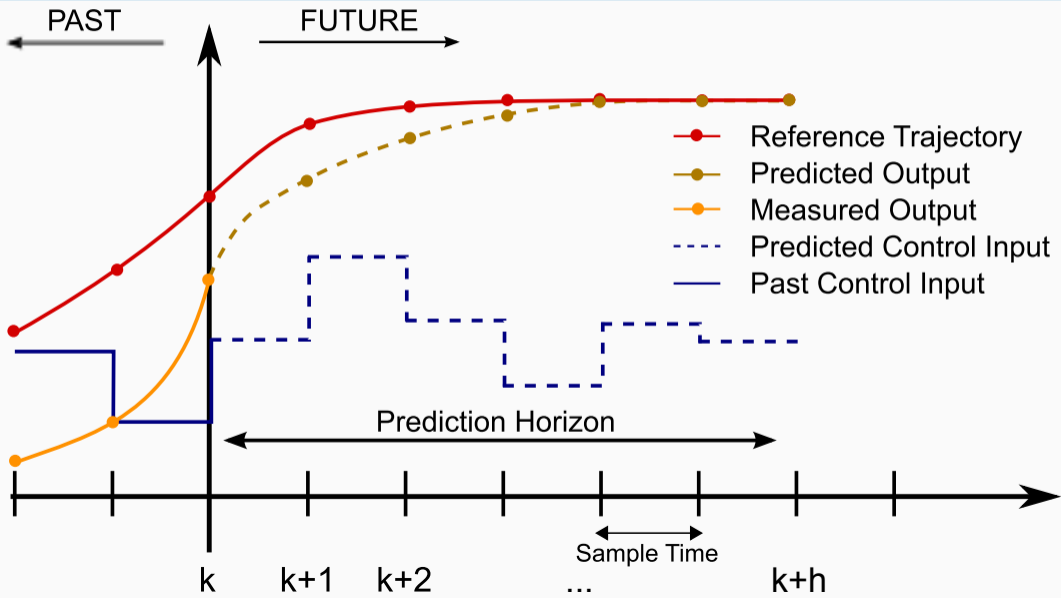
- jednokroková minimalizace $J(\mathbf{x}[k])$ není globálně optimální
- použijeme model systému pro h -krokovou predikci $\mathbf{x}[k+1], \dots, \mathbf{x}[k+h]$
- rozšíříme $\mathbf{x}[k]$ na $\mathbf{x}[h|k] = [\mathbf{x}[k]; \mathbf{x}[k+1]; \dots; \mathbf{x}[k+h]]$
- počítáme $J(\mathbf{x}[h|k])$ pro h kroků

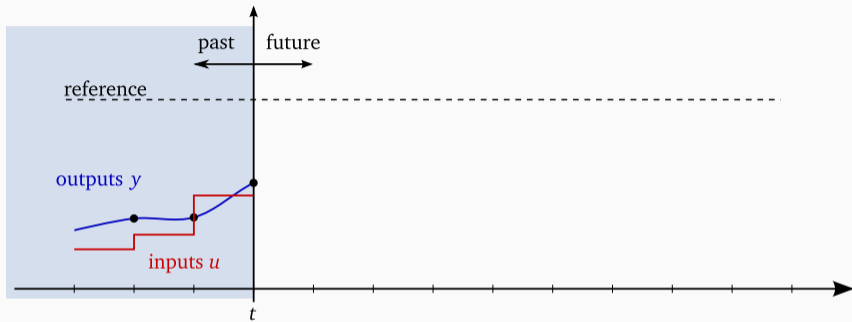




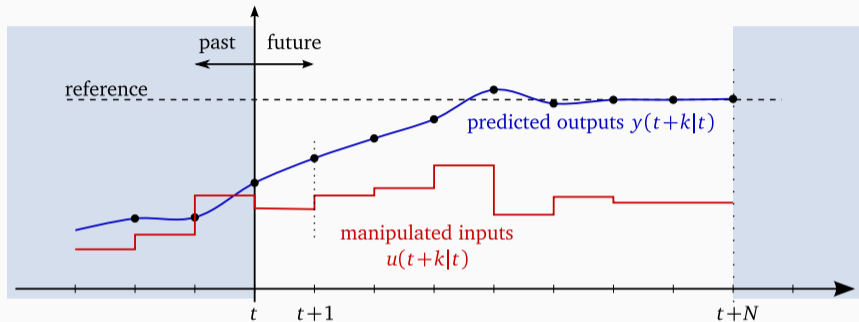
- Návrh MPC řadiče nad ACC, regulujícího požadovanou rychlost tak, aby vozidlo jelo co nejekonomičtěji
- Predikce: maximální + minimální rychlost dopravy, sklon
- Omezení: maximální + minimální rychlost dopravy a vozidla



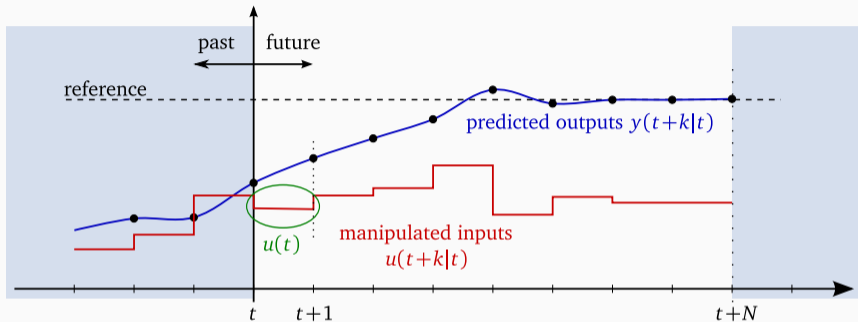




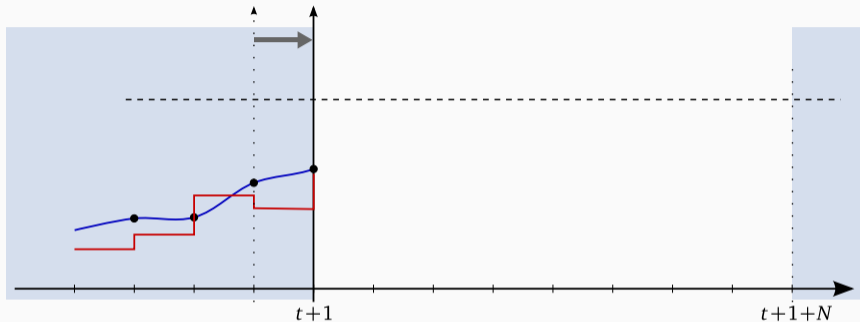
- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$



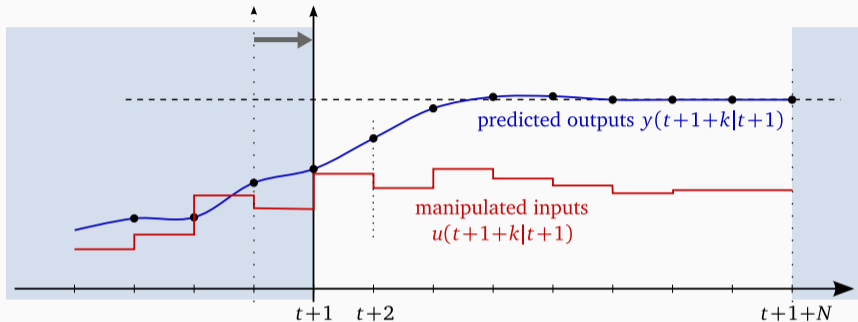
- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$



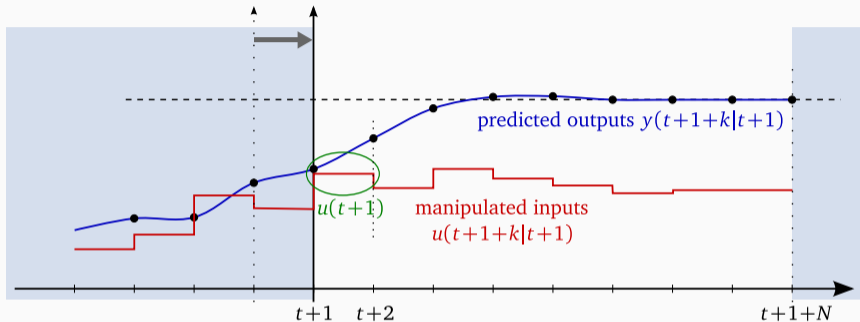
- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$



- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$



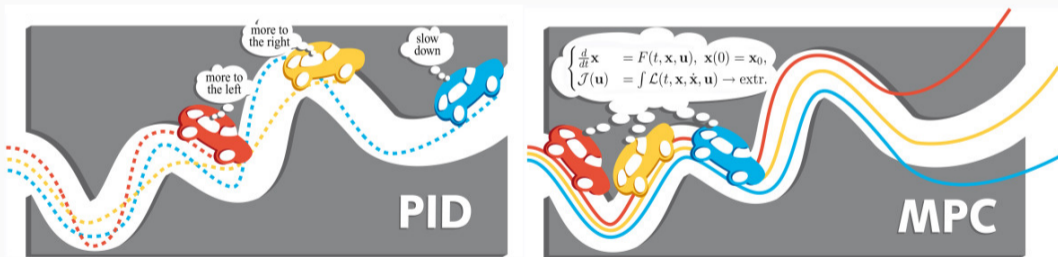
- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$



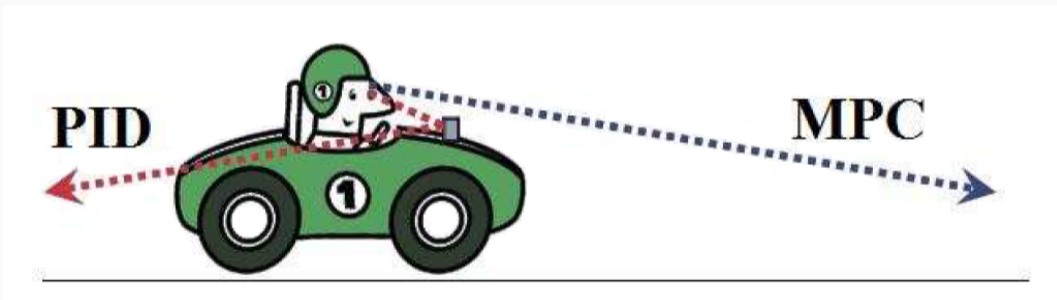
- Podle $x(t)$ optimalizujeme vstupy na horizontu délky N
- Z celé predikce použijeme pouze první optimální krok $u(t)$
- Postoupíme o jeden časový krok a posuneme horizont
- Vše opakujeme v čase $t + 1$

MPC reaguje na predikovanou budoucí hodnotu regulačních odchylek

PID reaguje jen na současné a minulé hodnoty.



Používat PID je jako řídit auto na základě pohledu do zpětného zrcátka (Hlava, 2007).



Omezení hodnot akčních veličin:

$$u_{\min} \leq u[k + p|k] \leq u_{\max}, \quad p = 0, 1, \dots, h$$

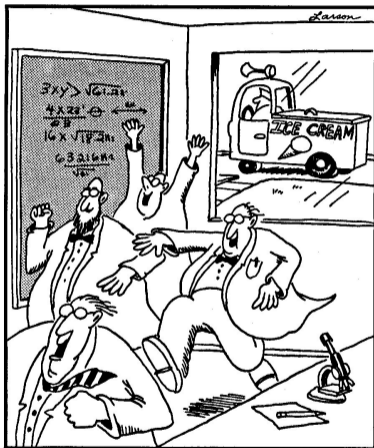
Omezení hodnot přírůstků akčních veličin:

$$|\Delta u[k + p|k]| \leq \Delta u_{\max}, \quad p = 0, 1, \dots, h$$

Omezení hodnot regulovaných veličin:

$$y_{\min} \leq y[k + p|k] \leq y_{\max}, \quad p = h, n + 1, \dots, N$$

- mohou být obecně také proměnná v čase,
- u regulovaných veličin $\rightarrow \pm$ prázdná množina přípustných řešení



Děkuji za pozornost. Až budete utíkat, prosím opatrně.