Řízení a optimalizace
Stavové modely a model-prediktivní řízení

Matematické metody pro ITS (11MAMY)

Jan Přikryl

2. přednáška 11MAMY

úterý 22. března 2021

verze: 2022-03-29 12:03

Obsah

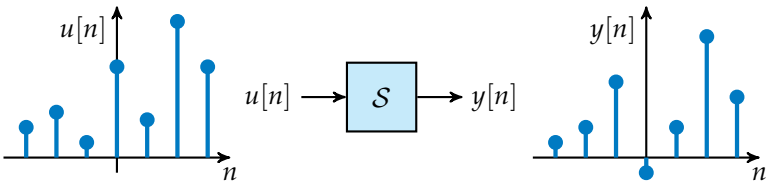
Tento text je do jisté míry experimentálním pískovištěm na odladění převodu textu prezentace vytvořené v LaTeXové třídě `beamer` do textu vysázeného pomocí `tufte-handout`. Obsah je oproti prezentaci mírně rozšířen o poznámky. Bude se ještě v průběhu semestru měnit, kontrolujte si prosím čas sestavení v záhlaví tohoto souboru.

Popis dynamického systému

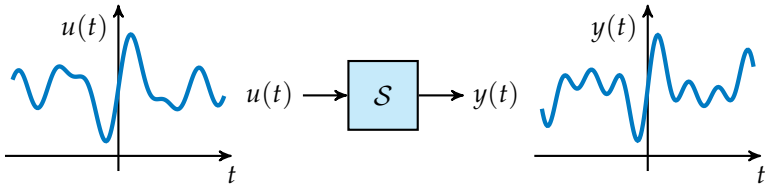
Připomeňme si z minulé přednášky, že dynamický systém lze popsat buď to pohledem zvenčí, bez znalosti vnitřních vazeb v sys-

tému – v takovém případě hovoříme o vnějším popisu – nebo pohledem zevnitř, pohledem na vnitřní fungování systému, na jeho vnitřní vazby a jeho **stav** – pak hovoříme o vnitřním popisu systému, v oblasti teorie řízení častěji o stavovém popisu.

Vnější popis systému – Opakování



Obrázek 1: Vnější popis obecného diskrétního systému



Obrázek 2: Vnější popis obecného spojitého systému

Vnitřní popis systému

Vnitřní popis dynamického systému je vztah mezi všemi veličinami systému, je to tedy relace mezi vstupními, stavovými a výstupními veličinami. Vnitřní popis je nejčastěji vyjádřen **stavovými rovnicemi**. Vnější popis, o němž jsme se bavili do této chvíle, je relace pouze mezi vstupními a výstupními veličinami, vyloučili jsme z něj veličiny stavové a systém popsaný vnějším popisem považujeme za černou skříňku (angl. *black box*).

Vnitřní a vnější popis systému spolu samozřejmě souvisí. Známe-li vnitřní popis -– stavové rovnice, snadno z něho jednodušší vnější popis odvodíme tak, že vyloučíme stavové proměnné. Obrácený postup, tedy určení vnitřního popisu z popisu vnějšího, již není tak jednoduchý. Vnitřní popis systému je bohatší a získáme jej z jednoduššího vnějšího popisu pouze za určitých předpokladů o struktuře systému. Z vnějšího popisu není totiž zřejmé, kolik má systém stavů, neboli jaká je dimenze stavového prostoru, ani jak zvolit jeho bázi. Určení vnitřního popisu z popisu vnějšího se nazývá problém realizace systému Štecha (2005).

Vnitřní popis nelineáního systému

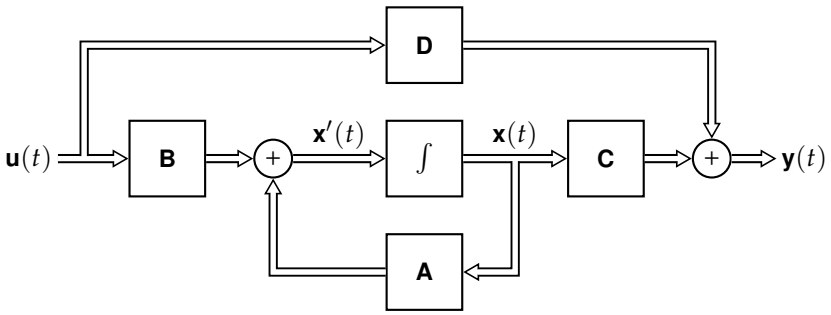
Spojitý systém	**Diskrétní systém**
vektor vstupních (řídicích) proměnných $\mathbf{u}(t)$	$\mathbf{u}[n]$
stavový vektor $\mathbf{x}(t)$	$\mathbf{x}[n]$
vektor výstupních proměnných $\mathbf{y}(t)$	$\mathbf{y}[n]$
$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$	$\mathbf{x}[n+1] = \mathbf{f}(n, \mathbf{x}[n], \mathbf{u}[n])$
$\mathbf{y}(t) = \mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t))$	$\mathbf{y}[n] = \mathbf{g}(n, \mathbf{x}[n], \mathbf{u}[n])$

Vnitřní popis lineárního systému

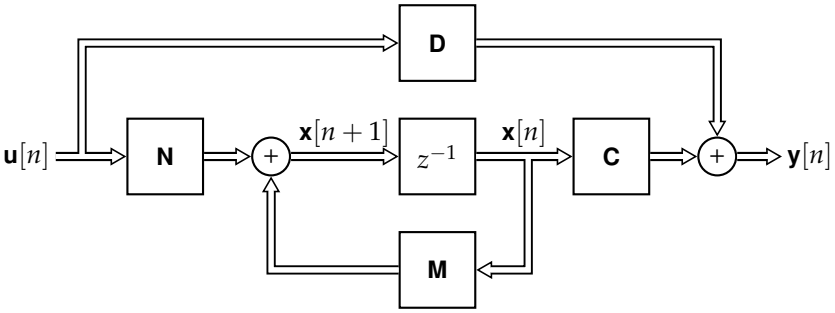
Nejprve obecnější nestacionární systém.

Spojitý systém	**Diskrétní systém**
$\mathbf{u}(t)$ ... vektor vstupních (řídicích) proměnných	$\mathbf{u}[n]$
$\mathbf{x}(t)$ ... stavový vektor	$\mathbf{x}[n]$
$\mathbf{y}(t)$ ... vektor výstupních proměnných	$\mathbf{y}[n]$
$\mathbf{x}'(t) = \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\mathbf{u}(t)$	$\mathbf{x}[n+1] = \mathbf{M}[n]\,\mathbf{x}[n] + \mathbf{N}[n]\,\mathbf{u}[n]$
$\mathbf{y}(t) = \mathbf{C}(t)\,\mathbf{x}(t) + \mathbf{D}(t)\,\mathbf{u}(t)$	$\mathbf{y}[n] = \mathbf{C}[n]\,\mathbf{x}[n] + \mathbf{D}[n]\,\mathbf{u}[n]$

Spojitý systém	**Diskrétní systém**
$\mathbf{u}(t)$ ... vstupní (řídicí) vektor	$\mathbf{u}[n]$ ... vstupní (řídicí) 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 $(m \times r)$



Obrázek 3: Blokové schéma spojitého LTI systému



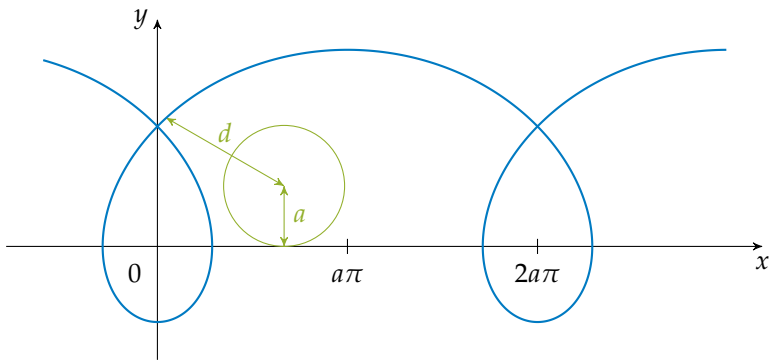
Obrázek 4: Blokové schéma diskrétního LTI systému

Příklady na stavový popis dynamických systémů

Cykloida

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

[1] Lze 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.



Obrázek 5: Cykloida vznikne odvalováním bodu ve vzdálenosti d od středu kružnice s poloměrem a

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

$$\begin{aligned} x(t) &= x_1(t) = at - d\sin t, \\ y(y) &= x_2(t) = a - d\cos t, \end{aligned}$$

která je pro počáteční podmínky[2]

[2] Bod je na počátku děje v dolní úvrati.

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

dána řešením stavové rovnice

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

Poloha bodu je výstupem dynamického systému

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\,t \\ \mathbf{p}(t) &= \mathbf{x}(t) \end{aligned}$$

popsaného maticemi

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

Modely typu predátor-kořist

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

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} x_1(t) &= a\, x_1(t) - b\, x_1(t) x_2(t), \\ \frac{\mathrm{d}}{\mathrm{d}t} x_2(t) &= -c\, x_2(t) + d\, x_1(t) x_2(t). \end{aligned} \tag{1}$$

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

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

jejímž řešením je exponenciální růst populace ovcí nade všechny meze (neuvažujeme omezení zdrojů potravy, nemoci a tak dále)

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

zatímco bez potravy je přírůstek populace vlků záporný

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

a vlci hynou,

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

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

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

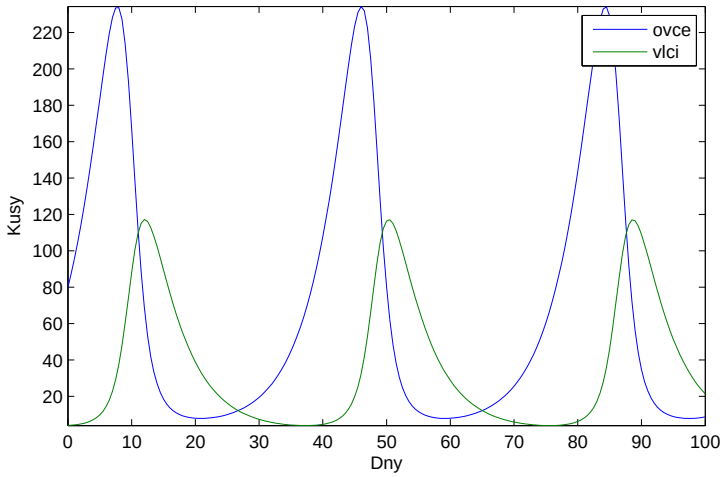
Počet ovcí klesá s rostoucí šancí, že ovce potká vlka:

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

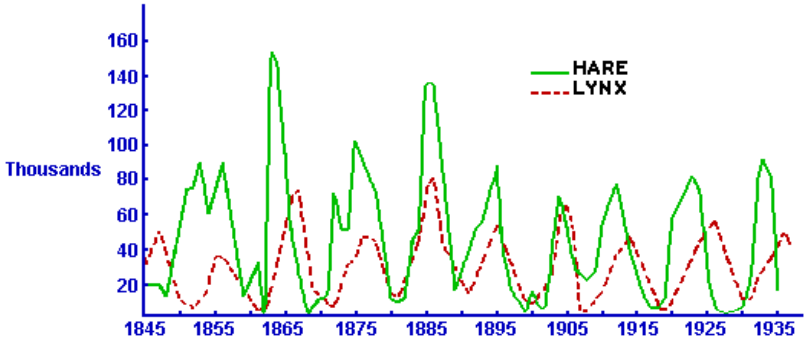
a vlci, hodující na ovcích, se mají dobře a jejich počet proto odpovídajícím způsobem stoupá:

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

Typický časový průběh předpovědi populací ovcí a vlků modelem podle rovnice (1) je znázorněn na obrázku 6. Pro srovnání můžeme nahlédnout na obrázek 7, uvádějící velikosti populace rysů a zajíců na přelomu 19. a 20. století v Kanadě. Podobnost obou závislostí je zřejmá.



Obrázek 6: Vývoj populací vlků a oveček s parametry $x_1(0) = 80$, $x_2(0) = 4$, $a = 0{,}2$, $b = 0{,}006$, $c = 0{,}2$, $d = 0{,}003$



Obrázek 7: Vývoj populací rysů a sněžných zajíců v Kanadě

Úrovně modelování

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

Přesný model systému

White-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, ...).

Model systému z naměřených dat

Black-box

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.

■ **Zde by slušel příklad, třeba regrese vs. Gaussovský proces, něco podobného, jako má Ghahramani:** `http://mlss.tuebingen.mpg.de/2015/slides/ghahramani/gp-neural-nets15.pdf` ■ Celá tato úvaha totiž směřuje k tomu, že v mnoha případech (neuvažujeme teď tak oblíbené neuronové sítě a deep learning) je pro modelování složitých datových vazeb nejvhodnější použít Bayesovské neparametrické datové modely, založené na Bayesově pravidle ve formě

$$P(f|\mathcal{D}) = \frac{P(f)P(\mathcal{D}|f)}{P(\mathcal{D})}.$$

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

Tabulka 1: Základní přehled parametrických a neparametrických metod matematického modelování systémů z naměřených dat. Jde vždy o metody statistického učení, blíže viz James (1999)

Dnes také velmi oblíbené neuronové sítě ⇒ BIG DATA!

■ **Toto je podáno velmi nahrubo a nesrozumitelné** ■

V ILSJames (1999) lze narazit na jednoduchou otázku porovnání neparametrických modelů (tj. například spline) s parametrickými modely (např. lineární regrese) podle různých scénářů. Otázkou je:

Jaké bude obecné očekávání lepší nebo horší výkonnosti parametrické metody statistického učení v porovnání s neparametrickou, pokud:

a) Počet prediktorů p je extrémně velký, a počet pozorování n je malý?

b) Rozptyl chybových členů[3], tj. $\sigma^2 = \mathrm{Var}(\mathbf{e})$, je velmi vysoký?

V těchto situacích srovnání flexibilních a neflexibilních modelů také závisí na

[3] Chybový člen je například $|\tilde{x} - x|$, kde $\tilde{x}$ je odhadovaná hodnota a x je reálná hodnota. Tvar členu závisí na chybové normě.

vztahu $y = f(x)$ – jak dalece je tento vztah lineární nebo naopak velmi nelineární,

tom, jak jsme naladili/omezili rozsah flexibility u neparametrického modelu při jeho identifikaci.

Pokud se hledaný vztah nachází v blízkosti lineární reprezentace a neomezíme nějak flexibilitu, lineární model by měl poskytnout lepší testovací chybu v obou výše uvedených případech, protože flexibilní model pravděpodobně v obou případech přetrénujeme (chytí se šumu, angl. *overfitting*).

Můžeme to vysvětlit následovně:

V obou případech údaje neobsahují dostatek informací o skutečném vztahu (v prvním případě vztahu je vysoká rozměrová složitost úlohy, p je vysoké, a nemáme dostatek dat; v druhém případě jsou data poškozena šumem)

Lineární model přináší některé externí apriorní informace o skutečné závislosti (omezujeme třídu hledaných modelů vestavěnými vazbami pouze na ty lineární) a že před info dopadá být správná (true vztah se nachází v blízkosti lineární). Neparametrický model neobsahuje žádné předběžné informace (je schopen se adaptovat na jakákoliv data), takže se bude držet šumu v datech tak dalece, jak mu to dovolíme.

Jestliže je však hledaný vztah velmi nelineární, je těžké říci, jaký model vyhraje; spíše lze asi tvrdit, že to bude plichta, protože ani jeden nebude fungovat zcela správně – ani neparametrický model nebude schopen přesně vystihnout nelineární závislost v systému.

Pokud naladíme/omezíme míru flexibility a uděláme to správným způsobem (například křížovou validací), potom s neparametrickým (flexibilním) modelem bychom ale měli uspět ve všech případech.

Gray-box

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 vyzoká 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

Tabulka 2: Základní rozdělení white-box, gray-box a black-box modelů

Základy teorie řízení

Následující text je velmi inspirován skriptem pánů Havleny a Štecha (Havlena, 1999).

Každý z nás mnohokrát denně dělá v nejrůznějších situacích různá rozhodnutí. Jsou to rozhodování typu kam půjdu, co udělám, co udělám nyní a co později a podobně. Jistě platí, že úspěch člověka podstatnou měrou závisí na jeho správných rozhodnutích, zvláště v klíčových situacích.

Abychom se rozhodovali správně, vytváříme si, vědomě či nevědomě, ve své mysli modely situací a podle nich vážíme důsledky různých variant, které máme při rozhodování k dispozici. Rozhodneme se samozřejmě pro tu variantu, jejíž důsledek je pro nás nejpříznivější. Skutečný důsledek našeho rozhodnutí poznáme ale až později, při rozhodování možné důsledky odhadujeme (predikujeme) pouze pomocí modelu dané situace. Model si tedy vytváříme za účelem predikce budoucích důsledků našich možných rozhodnutí. Čím lepší model situace si vytvoříme, tím, po zvážení všech důsledků, máme lepší možnost vybrat si dobrou variantu pro naše rozhodnutí.

Naše rozhodnutí závisí tedy na tom, jak věrný model situace jsme si schopni vytvořit. Při tom se samozřejmě uplatňuje naše zkušenost, intuice a vědomosti. Často si model situace vytváříme při rozhovoru s přáteli a porovnáváme jejich názor (jejich model situace a jeho hodnocení) s názorem svým. Naše rozhodnutí závisí také na tom, jaké kritérium pro vážení různých důsledků si vybereme. Ani to není mnohdy věc jednoduchá. Při hodnocení rozhodnutí provedených v minulosti a jejich skutečných důsledků si často uvědomujeme, jak nevhodné kritérium jsme si v minulosti zvolili.

Při rozhodování se někdy uplatňuje časové omezení. V dané si-

tuaci je třeba se rychle rozhodnout, ale pro správné rozhodnutí je třeba vytvořit dobrý model situace, což vyžaduje určitý čas. Nutnost rychlého rozhodnutí stojí v přímém protikladu k časově náročnému procesu vytváření modelu dané situace. Při tom se uplatňuje také složitost rozhodovacího algoritmu, kterou je také třeba někdy omezit. Proto se často pod tlakem času musíme rozhodovat podle velmi zjednodušených modelů dané situace, které zahrnují pouze nejpodstatnější jevy.

Naopak nejsme-li při rozhodování v časové tísni, často provádíme různé testy, abychom lépe porozuměli rozhodovací situaci. Tak na příklad nakoupíme vzorek zboží, abychom poznali jeho kvalitu; partnera podrobíme různým zkouškám, abychom lépe poznali jeho povahu a podobně.

Diskrétní řízení dynamických systémů je vlastně posloupnost rozhodování o volbě velikosti řídicí veličiny v různých časových okamžicích. Proto při řízení dynamických systémů postupujeme principiálně stejným způsobem jako při volbě našich rozhodnutí v nejrůznějších životních situacích.

Dynamický systém

Řízením působíme na reálný svět. Tu část reality, kterou řídíme, nazýváme objekt. Abychom nějaký objekt dobře řídili, je třeba si vytvořit dobrý model objektu. To znamená, že na objektu si definujeme systém. Dynamický systém si vytváříme pro predikci chování objektu v budoucnosti. Existují v podstatě dva způsoby tvorby systému na daném objektu či procesu, a to analytický a experimentální.

Při prvním způsobu využíváme různé fyzikální, biologické, ekonomické a jiné zákonitosti a na jejich základě hledáme vztahy mezi veličinami, které nás zajímají. Tomuto způsobu vytváření modelu objektu se říká matematicko - fyzikální analýza. Dynamický systém vzniklý na základě matematicko-fyzikální analýzy je často složitý, a proto abychom ho mohli použít, je nezbytné ho zjednodušit. V tom je další úskalí. Ve vztazích mezi veličinami se vyskytují různé konstanty, jejichž velikost se mnohdy určuje obtížně.

Druhý způsob tvorby systému je založen na měření provedeném na skutečném objektu, rozboru změřených dat a tím určení vztahů mezi veličinami. Tomuto způsobu vytvářen modelu objektu se říká experimentální identifikace. Systém si při tomto způsobu představujeme jako černou skříňku (black box). Při tomto způsobu tvorby modelu je třeba vzít v úvahu to, že měření má vždy omezenou přesnost, a proto je model objektu vzniklý tímto způsobem velmi často stochastický systém.

Samozřejmě nejlepší cesta je kombinace analytického a experimentálního způsobu tvorby systému. Pomocí analytického přístupu určíme například strukturu systému. Potom při experimentální identifikaci přistupujeme k systému jako k šedé skříňce (gray box),

o které již máme částečné znalosti z analytického rozboru. Modelem situace nebo objektu je tedy systém, a protože budeme sledovat časový průběh veličin, bude se jednat o dynamický systém. Systém je znázorněn na obr. 1.1.

Veličiny, které na systém působí z okolí a jsou na systému nezávislé, nazýváme vstupní veličiny. Tyto veličiny rozdělujeme na dvě skupiny. Jedna skupina vstupních veličin jsou takové veličiny, které můžeme měnit podle potřeby. Ty nazýváme řídicí veličiny systému a značíme je u. Řídicí veličiny systému se také nazývají akční veličiny. Druhou skupinou vstupních veličin jsou takové veličiny, které působí na systém nezávisle na nás a nemáme možnost je měnit. Takové veličiny nazýváme poruchové veličiny a značíme je v. Pro účely řízení je třeba ještě poruchové veličiny rozdělit na poruchové veličiny měřitelné a neměřitelné.

Veličiny, které produkuje systém a které měříme, nazýváme výstupní veličiny systému. Veličiny, které chceme řídit nebo regulovat, nazýváme regulované veličiny a značíme je y. Výstupní veličiny systému nemusí být totožné s veličinami regulovanými. Požadavky kladené na řízení

Teorie automatického řízení zkoumá metody jak působit na systém, jak ho řídit, aby se řízený systém choval podle našich požadavků. Požadavky kladené na řízení mohou být různé.

Kompenzace vlivu poruchových veličin (Disturbance Rejection)

Na systém vždy působí celá řada poruchových veličin, jejichž vliv na regulovanou veličinu je mnohdy nežádoucí. Někdy je poruchová veličina měřitelná a potom lze její vliv kompenzovat účinně tím způsobem, že signál od měřitelné poruchy zpracujeme v řídicím členu. Tak lze někdy vliv poruchy kompenzovat beze zbytku. Pokud se tímto způsobem porucha na výstupu systému vůbec neprojeví (řídicí člen její vliv úplně kompenzuje), pak říkáme, že řízený systém je invariantní vzhledem k poruše.

Problém regulátoru (Regulator Problem)

Dynamické vlastnosti samotného systému jsou někdy nevyhovující (například systém je nestabilní) a účelem řízení je navrhnout takovou strukturu řízení, aby celý systém měl potom vyhovující dynamické vlastnosti. Často chceme tímto způsobem systém stabilizovat.

Problém sledování (Tracking Problem)

Někdy požadujeme, aby výstupní veličina co nejvěrněji sledovala nějaký průběh určený referenční veličinou w, chceme tedy, aby byla co nejmenší takzvaná regulační odchylka e, která je rozdílem mezi požadovanou veličinou a skutečným výstupem ze systému (čili $e = w - y$). Často je nutné respektovat různá omezení na velikost řídicí veličiny u či omezení na velikost celkové řídicí energie.

Optimální řízení (Optimal Control)

V moderní teorii řízení jsou všechny požadavky na řízení shrnuty do kritéria kvality řízení a problém řízení je převeden na optimalizační problém minimalizace kritéria kvality řízení. Při tomto přístupu k řešení problému řízení existují dva zásadní problémy. Prvním problémem je vhodná volba kritéria kvality řízení, která by zahrnula všechny naše požadavky na kvalitu řízení. Druhým problémem je řešitelnost takto formulovaného optimalizačního problému. Nejvíce používaným kritériem kvality řízení je kvadratické kritérium, které pro lineární systémy vede na lineární zákon řízení.

Teorie řízení má dva hlavní kořeny: regulaci a optimalizaci trajektorie. Ten první, regulace, je ten důležitější a technicky zaměřený. Druhý kořen, optimalizace trajektorie, je založen na matematice. Jak však uvidíme, tyto kořeny se ve druhé polovině dvacátého století do značné míry spojily.

Úkolem regulace je navrhnout mechanismy, které udržují určité regulované proměnné na konstantních hodnotách proti vnějším poruchám, které působí na regulované zařízení, nebo proti změnám jejích vlastností. Systém, který je řízen, se obvykle označuje jako zařízení, což je obecný termín, který může označovat například fyzikální nebo chemický systém. Může to být také ekonomický nebo biologický systém, ale v takovém případě by se nepoužíval technický výraz „zařízení“.

Příkladů regulačních problémů z našeho bezprostředního prostředí je mnoho. Domy jsou regulovány termostaty tak, aby vnitřní teplota zůstala konstantní bez ohledu na změny vnějších povětrnostních podmínek nebo změny situace v domě: dveře, které mohou být otevřené nebo zavřené, počet osob přítomných v místnosti, aktivita v kuchyni atd. Motory v pračkách, sušičkách a v mnoha dalších domácích spotřebičích jsou řízeny stálými otáčkami, nezávisle na zatížení. Moderní automobily mají desítky zařízení, která regulují různé proměnné. Ve skutečnosti je možné pohlížet také na odpružení automobilu jako na regulační zařízení, které pohlcuje nerovnosti vozovky za účelem zlepšení pohodlí a bezpečnosti cestujících. Regulace je skutečně velmi důležitým aspektem moderní technologie. Z mnoha důvodů, jako je účinnost, kontrola kvality, bezpečnost a spolehlivost, vyžadují průmyslové výrobní procesy regulaci, aby bylo zaručeno, že určité klíčové proměnné (teploty, směsi, tlaky atd.) Budou udržovány na příslušných hodnotách. Faktory, které brání dosažení těchto požadovaných hodnot, jsou vnější rušení, jako například vlastnosti surovin a úrovně zatížení nebo změny vlastností zařízení, například v důsledku stárnutí zařízení nebo selhání některých zařízení. Problémy s regulací se vyskytují také v jiných oblastech, jako je ekonomie a biologie.

Jedním z ústředních konceptů v řízení je zpětná vazba: hodnota jedné proměnné v zařízení se měří a používá (přivádí se zpět), aby

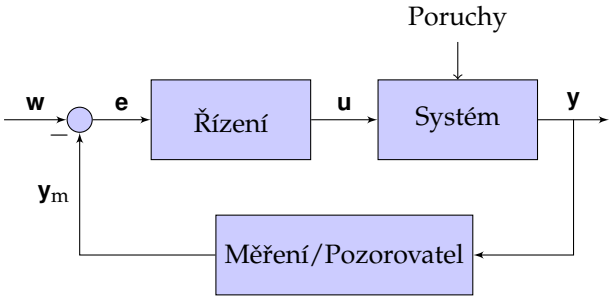
bylo možné provést příslušnou akci prostřednictvím řídicí proměnné v jiném bodě zařízení. Dobrým příkladem zpětnovazebního regulátoru je termostat: snímá teplotu v místnosti, porovnává ji s nastavenou hodnotou (požadovanou teplotou) a vrací výsledek zpět do kotle, který se poté zapne nebo vypne podle toho, zda je teplota příliš nízká nebo příliš vysoká.

Pole regulace považované za regulaci zůstalo hlavně technologií poháněnou během první poloviny dvacátého století. V tomto období došlo k dvěma velmi důležitým vývojům, které měly na oblast trvalý vliv. Nejprve zde byl vynález řadiče proporcionálního - integrálního - diferenciálního (PID). PID regulátor produkuje řídicí signál, který se skládá z váženého součtu tří členů (PID regulátor se proto často nazývá tříčlenný regulátor). P-člen produkuje signál, který je úměrný chybě mezi skutečnou a požadovanou hodnotou proměnné, která má být řízena. Dosahuje základní kontroly kompenzace zpětné vazby, což vede k řídicímu vstupu, jehož účelem je, aby se proměnná, která má být řízena, zvýšila, když je příliš nízká, a snížila, když je příliš vysoká. I-termín vrací zpět integrál chyby. Výsledkem tohoto výrazu je velmi velký korekční signál, kdykoli se tato chyba nesblíží na nulu. Za tu chybu tedy platí, Jdi na nulu nebo poprsí! Při správném naladění tento termín dosahuje robustnosti, dobrého výkonu nejen pro nominální zařízení, ale také pro zařízení, která jsou mu blízká, protože I-termín má tendenci vynutit chybu na nulu pro širokou škálu parametrů zařízení. Termín D působí na derivaci chyby. Výsledkem je řídicí korekční signál, jakmile se chyba začne zvětšovat nebo zmenšovat, a lze tedy očekávat, že tato předběžná akce povede k rychlé odezvě. PID regulátor měl a stále má velmi velký technologický dopad, zejména v oblasti řízení chemických procesů.

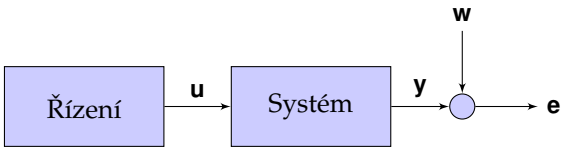
Nyní budeme sledovat další historický kořen kontroly, optimalizaci trajektorie. Problémem přenosu trajektorie je otázka určení drah dynamického systému, které přenášejí systém z daného počátečního do předepsaného koncového stavu. Často se hledají cesty, které jsou v určitém smyslu optimální. Na problém nalezení optimálních trajektorií se pohlíželo jako na kontrolní problém až ve druhé polovině dvacátého století. To se však změnilo v roce 1956 zveřejněním maximálního principu Pontryagin. Princip maximální sestává z velmi obecné sady nezbytných podmínek, které musí řídicí vstup, který generuje optimální cestu, splňovat. Tento výsledek je důležitým zobecněním klasických problémů variačního počtu. Nejen, že umožňuje řešit mnohem větší třídu problémů, ale je také důležité nastolit problém optimálního výběru vstupu (na rozdíl od optimálního výběru trasy) jako ústřední otázky optimalizace trajektorie.

Přibližně ve stejné době, kdy se objevil princip maxima, bylo zjištěno, že (optimální) vstup lze implementovat také jako funkci státu. To znamená, že místo hledání řídicího vstupu jako funkce času je možné zvolit (optimální) vstup jako funkci zpětné vazby stavu. Tato

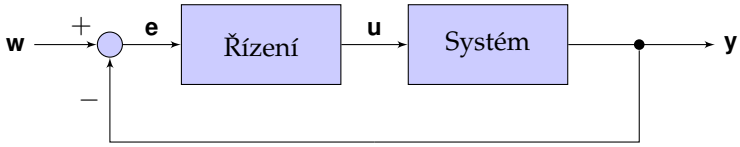
myšlenka je základem pro dynamické programování, které formuloval Bellman na konci 50. let a které bylo okamžitě publikováno v mnoha existujících časopisech o aplikované matematice. S pochopením získaným pomocí dynamického programování je rozdíl mezi regulací založenou na zpětné vazbě optimalizace trajektorie (založená na výběru vstupu) byla rozmazaná.



Ovládání:



Regulace:



Definice 1 (Dosažitelnost)**.** Stav **x** je dosažitelný, existuje-li řízení $\mathbf{u}(t)$, které za konečný čas převede počáteční stav $\mathbf{x}(t_0) = 0$ do stavu **x**. Jsou-li všechny stavy systému dosažitelné, říkáme, že *systém je dosažitelný*.

Při vyšetřování dosažitelnosti tedy vycházíme z nulového počátečního stavu a ptáme se na existenci řízení $\mathbf{u}(t)$ s uvedenými vlastnostmi.

Definice 2 (Řiditelnost stavu)**.** Stav **x** je řiditelný, existuje-li řízení $\mathbf{u}(t)$, které v konečném čase převede tento stav do počátku (do nulového stavu). Jsou-li všechny stavy systému řiditelné, říkáme, že *systém je řiditelný*.

Při vyšetřování řiditelnosti je stav **x** počátečním stavem a požadujeme konečný čas převodu.

Definice 3 (Řiditelnost na výstupu)**.** Systém je řiditelný na výstupu, pokud existuje funkce řízení $\mathbf{u}(t)$ taková, že převede výstup $\mathbf{y}(t_0)$ na $\mathbf{y}(t_1)$ v konečném čase $t_1 - t_0$.

Definice 4 (Pozorovatelnost)**.** Systém je pozorovatelný, když změřením vstupu a výstupu na konečném časovém intervalu je možno určit hodnotu stavu systému na počátku měření.

Nemůžeme-li rozborem změřených hodnot vstupu a výstupu jednoznačně určit počáteční stav systému, pak systém obsahuje nepozorovatelné stavy. Jsou to takové stavy systému, které se na výstupu systému vůbec neprojeví.

Měřením lze zjistit vlastnosti pouze u pozorovatelné části systému.

Definice 5 (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 $\mathbf{y}(t)$ co nejlépe sledovalo referenční průběh $\mathbf{w}(t)$.

Minimalizace regulační odchylky $\mathbf{e}(t)$

$$\mathbf{e}(t) = \mathbf{w}(t) - \mathbf{y}(t)$$

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*

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

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 výdělku, minimalizace cestovní doby)

Definice 6 (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*)

Dynamické programování označuje velmi rozsáhlou třídu algoritmů. Hlavní myšlenkou tohoto postupu je rozdělit komplikovaný problém (pokud je to možné) na jednodušší dílčí úlohy a postupně nalézat (optimální) řešení těchto dílčích problémů na dílčích podmnožinách dat, s nimiž pracuje původní úloha. Pokud lze DP použít, lze výsledky řešení těchto dílčích problémů postupně slučovat, aniž by byla dotčena již v minulosti vypočtená optimální řešení dílčích problémů. Postupným zpracováním předešlých výsledků nakonec obsáhneme a zpracujeme všechna vstupní data a získáme odpověď na původní problém.

Častým případem DP je jev, kdy pro průběžné zpracování dílčích úloha potřebujeme nakonec znát více informací, než kolik jich je potřeba v poslením kroku algoritmuy (jeden z oblíbených příkladů je DP úloha, řešící výpočet hodnoty n-tého členu Fibonacciho posloupnosti v lineární čase: V každé fázi výpočtu si algoritmus poamatuje nejméně dvě hodnoty, fib(i-1) a fib(i); v okamžiku, když i nabude hodnoty n, je potřeba pouze jedna z těchto hodnot, fib(i) = fib(n)). Je to však držení dvou hodnot, které umožňují algoritmus lineárního času.

Existují dvě typické implementace dynamického programování: přístup zdola nahoru a shora dolů.

Dynamické programování shora dolů není nic jiné, než obyčejná rekurze. Je ovšem vylepšeno zapamatováním řešení pro dílčí problémy. Když na určitý dílčí problém narazíme podruhé (potřetí,

počtvrté, . . .), není třeba tento problém znovu řešit od začátku, ale lze použít řešení, jež jsme si uložili po prvním vyřešení úlohy. Tato technika je známá pod názvem **memoizace** (angl. .memoization, bez 'r' před 'i').

To je vlastně to, co jsme si ukazovali v příkladu s Fibonacciho sekvencí: Stačí použít rekurzivní vzorec pro Fibonacciho posloupnost, ten by ale byl bez memoizace velmi pomalý. Pokud si navíc ale sestavíme tabulku hodnot fib(i) a postupně ji plníme, dostáváme algoritmus DP implementovaný přístupem shora dolů (pokud například potřebujete vypočítat hodnotu fib(5) podruhé, získáte ji z tabulky – kam jste si ji uložili po prvním výpočtu – místo jejího opětovného počítání).

Při dynamickém programování přístupem zdola nahoru je postup založen také na ukládání dílčích řešení do paměti, ale úlohu řešíme v opačném pořadí, tedy začínáme rovnou od nejmenších problému, jejichž řešení postupně slučujeme v řešení těch větších. Výsledná struktura algoritmu není rekurzivní. Klasickým příkladem tohto přístupu je výpočet nejdelší společné sekvence (LCS).

Algoritmy DP zdola nahoru jsou obvykle efektivnější, ale obecně je obtížnější (a někdy i nemožné) je vytvořit. Není totiž vždy jednoduché předvídat, jaké primitivní dílčí problémy budete muset vyřešit k tomu, abyste dokázali obsáhnout celý původní problém, a jakou cestou musíte postupovat od malých dílčích problémů, abyste se dostali ke konečnému řešení tím nejefektivnějším způsobem.

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

Definice 7 (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.

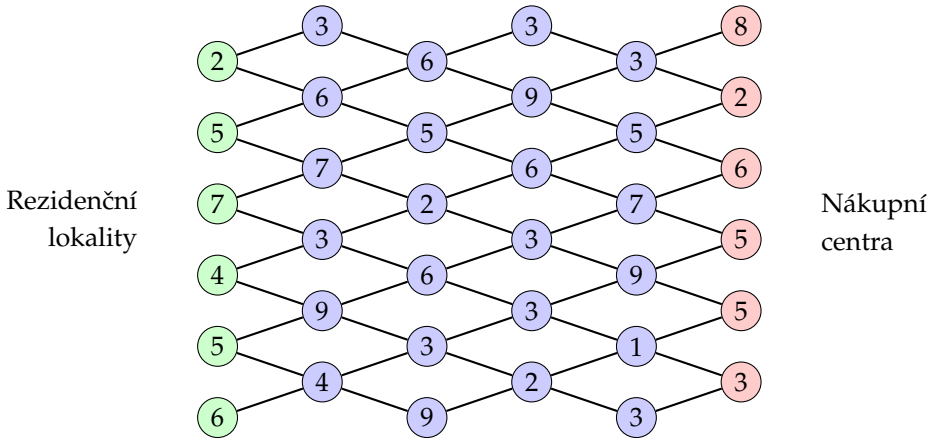
Přístupy:

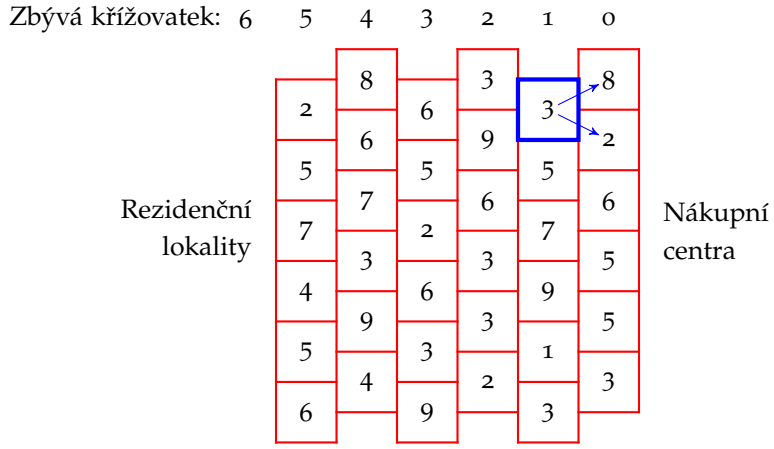
shora dolů (rekurze)

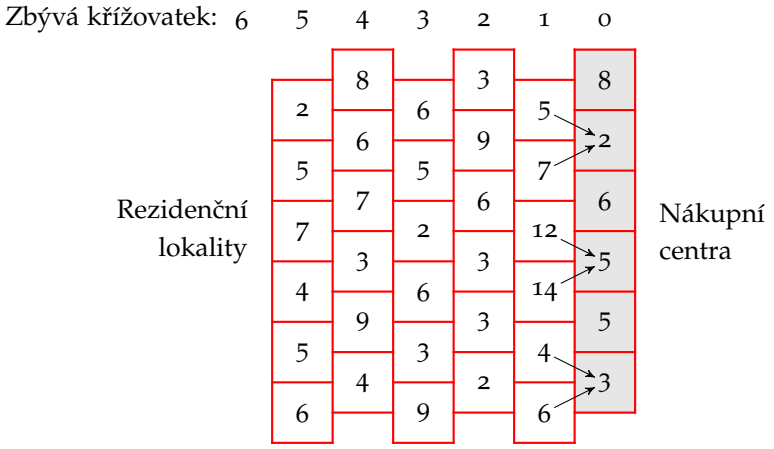
zdola nahoru

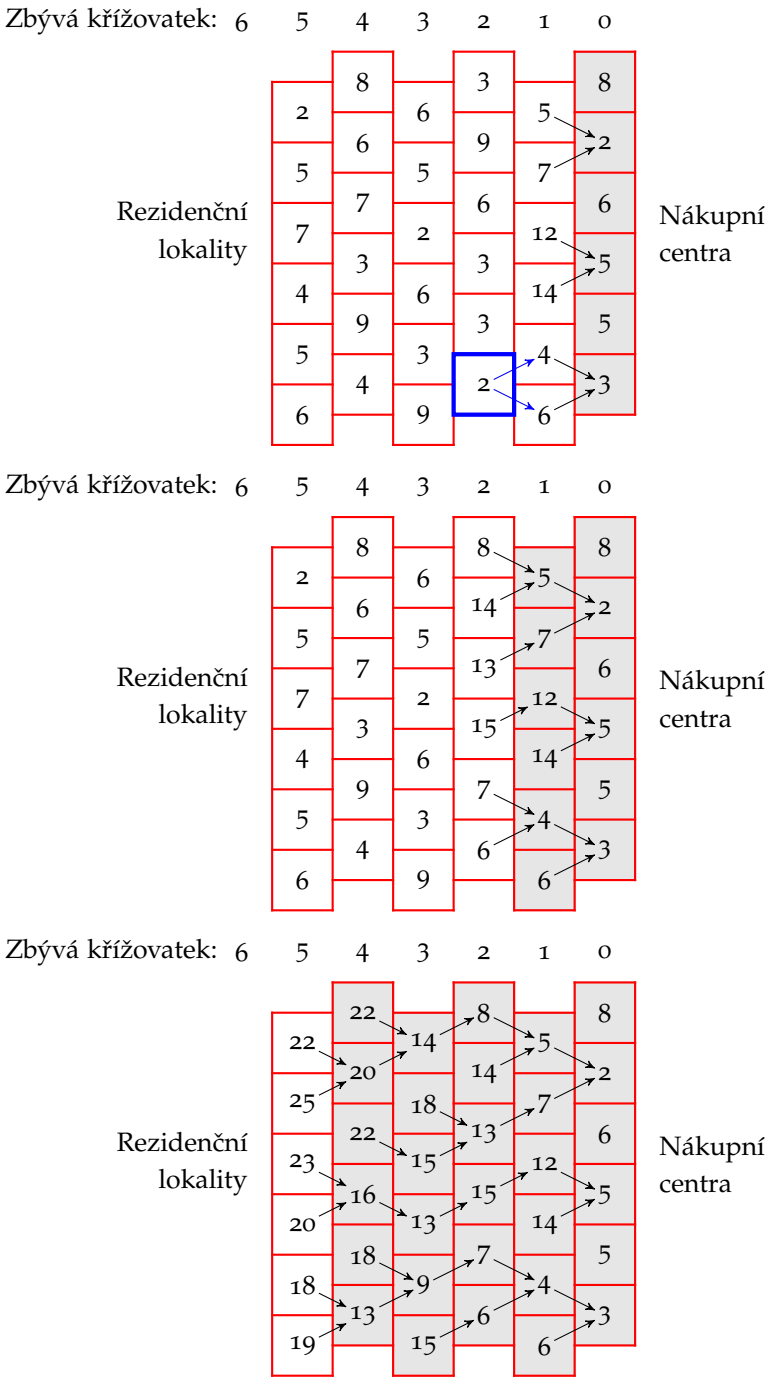
Hledáme nejrychlejší cesty z domů vlevo k nákupním střediskům vpravo. Uzly grafu jsou křižovatky, ohodnocení uzlů je zpož-

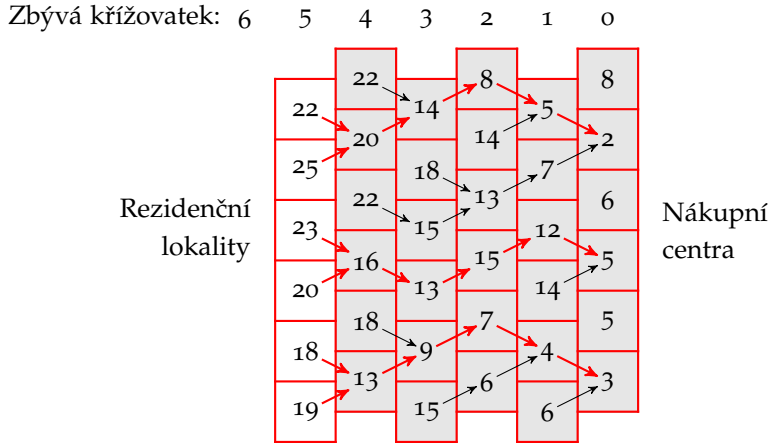
dění.


Rezidenční lokality
Nákupní centra


Zbývá křížovatek: 6 5 4 3 2 1 0
Rezidenční lokality
Nákupní centra


Zbývá křížovatek: 6 5 4 3 2 1 0
Rezidenční lokality
Nákupní centra


Zbývá křížovatek: 6 5 4 3 2 1 0
Rezidenční lokality
Nákupní centra
Zbývá křížovatek: 6 5 4 3 2 1 0
Rezidenční lokality
Nákupní centra
Zbývá křížovatek: 6 5 4 3 2 1 0
Rezidenční lokality
Nákupní centra



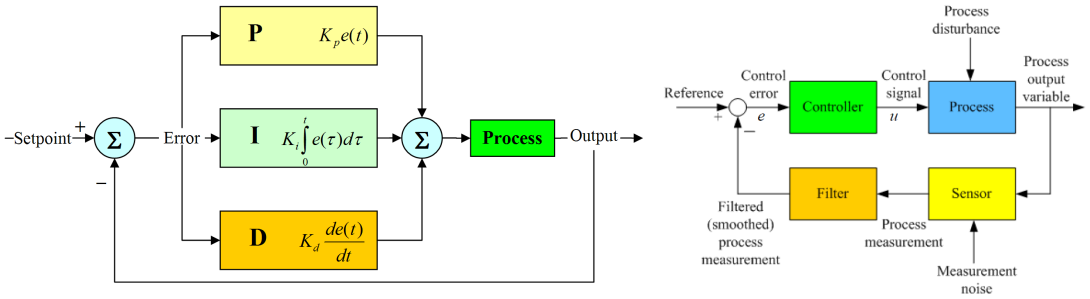
PID regulátor se používá více než sto let v různých formách. Těšil se popularitě jako čistě mechanické zařízení, jako pneumatické zařízení a dnes jako elektronické zařízení. Digitální PID regulátory založené na mikroprocesorech nalézají v dnešní době široké uplatnění v různých odvětvích průmyslu i jako součást spotřebního zboží.

PID znamená „proporcionální, integrální, derivační.“ Tyto tři pojmy popisují základní prvky PID regulátoru. Každý z těchto prvků plní jiný úkol a má jiný vliv na fungování systému. V typickém regulátoru PID jsou tyto prvky ovládány kombinací ovlázacího signálu a zpětnovazebního signálu ze systému (objektu), který je řízen. Zpětnovazební signál je odvozen od výstupu systému.

Integrální složka řízení slouží k přidání dlouhodobé přesnosti do řídící smyčky. Je téměř vždy používáno ve spojení s proporcionálním řízením, protože samotné integrální řízení snižuje stabilitu a systém má tendenci stále ocilovat okolo přednastavené cílové hodnoty výstupu. Stav integrátoru „si pamatuje“ vše, co se dělo předtím, což dovoluje řídící jednotce potlačit jakékoli dlouhodobé poruchy na výstupu. Tato paměť ale také přispívá k nestabilitě – řídící jednotka reaguje vždy pozdě, poté, co systém dosáhla rychlosti. Chcete-li stabilizovat dva předchozí systémy, potřebujete trochu jejich současné hodnoty, kterou získáte z proporcionálního výrazu.

Víme, že proporcionální řízení je odvozeno od současného chování zařízení a že integrální řízení zohledňuje minulé chování. Pokud bychom měli k dospozici nějaký prvek, který předpovídá chování zařízení do budoucnosti, pak bychom jej mohli použít ke stabilizaci zařízení. Tento úkol používáme diferenciátor. Diferenciální řízení je velmi mocné, ale jeho nastavování je také nejproblematičtější. Tři problémy, s nimiž se budete pravděpodobně zajímat, jsou nestejnoměrné vzorkování, šum a kmitání na vysokých kmitočtech.

$$u(t) = u_0 + \underbrace{K_\mathrm{p} e(t)}_{u_\mathrm{p}} + \underbrace{\frac{K_\mathrm{p}}{T_\mathrm{i}} \int_0^t e(\tau)\,\mathrm{d}\tau}_{u_\mathrm{i}} + \underbrace{K_\mathrm{p} T_\mathrm{d} \frac{\mathrm{d}}{\mathrm{d}t} e(t)}_{u_\mathrm{d}}$$



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

Hlavní účel, proč jsou tyto modely sestavovány, je možnost předpovědět, jak se za daných podmínek chová námi sledovaný systém. Z toho můžeme vyvodit hodnoty řídicích proměnných tak, abychom dosáhli optimálního stavu. Tato metoda se nazývá *prediktivní řízení* (anglicky *Model Predictive Control*, zkráceně MPC[4]).

[4]

Prediktivní řízení je stále ještě poměrně nová (používá se od 80. let 20. století) metoda zpětnovazebního řízení, založená na dostatečně přesném matematickém modelu řízeného procesu, jenž který se požívá k predikci budoucího chování systému pro současné hodnoty řídicích, poruchových a referenční veličin. Použitím složité optimalizační metody, často umožňující zahrnout i různá omezení a nelinearity, se vypočte nová posloupnost řídicích zásahů tak, aby byla optimální pro celý časový horizont konečné délky. Z něj se pak ale typicky využije jen první krok — řízení v nejbližší periodě vzorkování. Po jeho provedení se změří následky a využijí se pro aktualizaci modelu systému. V příštím kroku se vše opakuje: predikce, optimalizace, využití, aktualizace.

Míra optimality stavu systému $\mathcal{S}$ je většinou vyjádřena kritériem $J(\mathcal{S})$, funkcí přiřazující množině stavových a řídicích proměnných reálnou hodnotu. Optimalizace odpovídá minimalizaci tohoto kritéria. Dosažení optima v jednom časovém bodě může evidentně připravit špatné počáteční podmínky pro budoucí vývoj a vést tak k nestabilitě. K odstranění tohoto efektu optimalizujeme v rámci delšího časového horizontu[5]. Minimalizace neprobíhá pouze pro následující časový krok $k+1$, ale přes horizont — stanovený počet budoucích časových kroků $k+1, k+2, \ldots, k+h$. K řízení je vždy použita pouze první hodnota optimálních řídicích proměnných $u[k]$.

[5]; and

Nadále bude označovat $\mathbf{s}[h|k]$ vektor odhadů všech stavových i řídicích proměnných od časového kroku k na horizontu délky h, tedy po časový krok $k+h$. Pokud je $\mathbf{x}[k] \in \mathbb{R}^l$ a $\mathbf{u}[k] \in \mathbb{R}^m$, bude mít vektor $\mathbf{s}[h|k]$ tvar

$$\mathbf{s}[h|k] = (\hat{x}_1[k|k], ..., \hat{x}_l[k|k], \hat{u}_1[k|k], ..., \hat{u}_m[k|k], ..., \tag{2}$$

$$\hat{x}_1[k+h|k], ..., \hat{x}_l[k+h|k], \hat{u}_1[k+h-1|k], ..., \hat{u}_m[k+h-1|k])^{\mathrm{T}} . \tag{3}$$

Symbol $\hat{x}_i[l|k]$ označuje odhad proměnné $x_i[l]$ v čase t_l na základě stavu systému v čase t_k. Evidentně platí $\hat{x}_i[k|k] = x_i[k]$.

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 $\begin{cases} \mathbf{x}[k+1] = f(k, \mathbf{x}[k], \mathbf{u}[k]) \\ \mathbf{y}[k] = g(k, \mathbf{x}[k], \mathbf{u}[k]) \end{cases}$

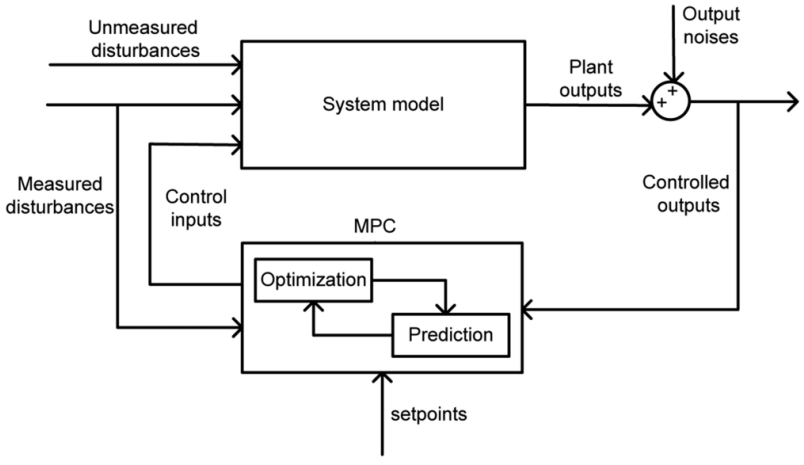
Vektor vstupů $\mathbf{u}[k]$ minimalizuje **ztrátovou funkci** $J(\mathbf{x}[k])$

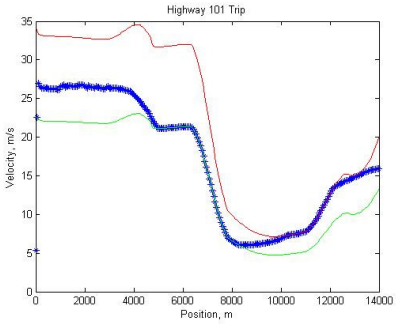
jednokroková minimalizace $J(\mathbf{x}[k])$ není globálně optimální

použijeme model systému pro h-krokovou predikci $\mathbf{x}[k+1], \ldots, \mathbf{x}[k+h]$

rozšíříme $\mathbf{x}[k]$ na $\mathbf{x}[h|k] = [\mathbf{x}[k]; \mathbf{x}[k+1]; \ldots; \mathbf{x}[k+h]]$

počítáme $J(\mathbf{x}[h|k])$ pro h kroků



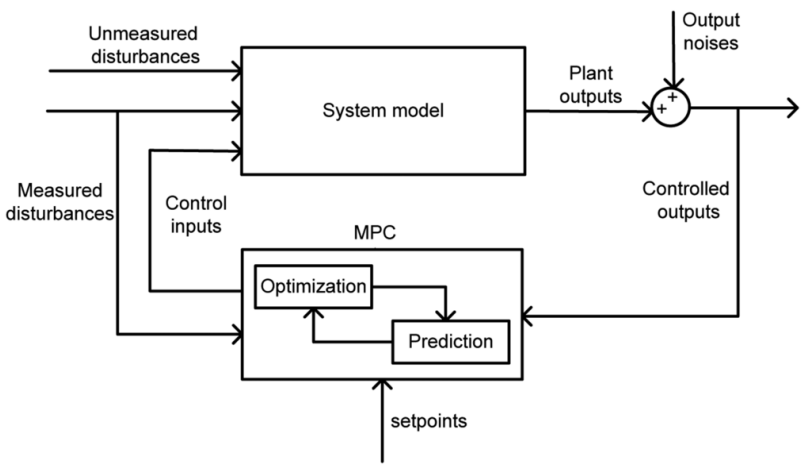


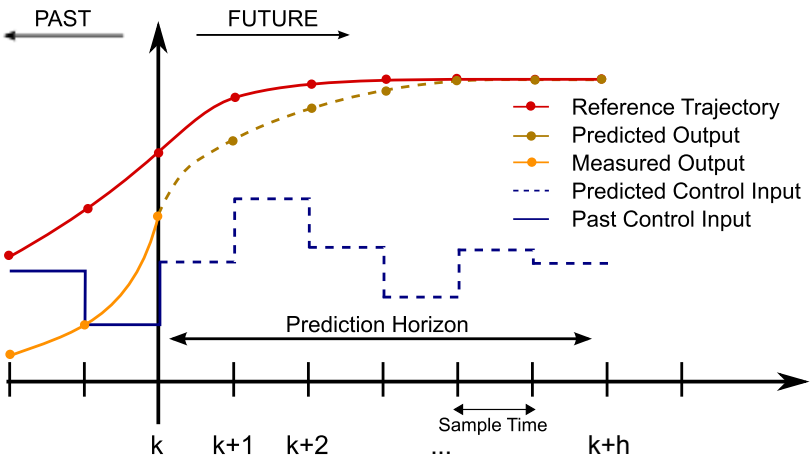
Návrh MPC řadiče nad ACC, regulujícího požadovanou rychlost tak, aby vozidlo jelo co nejhospodárněji

Predikce: maximální + minimální rychlost dopravy, sklon

Omezení: maximální + minimální rychlost dopravy a vozidla

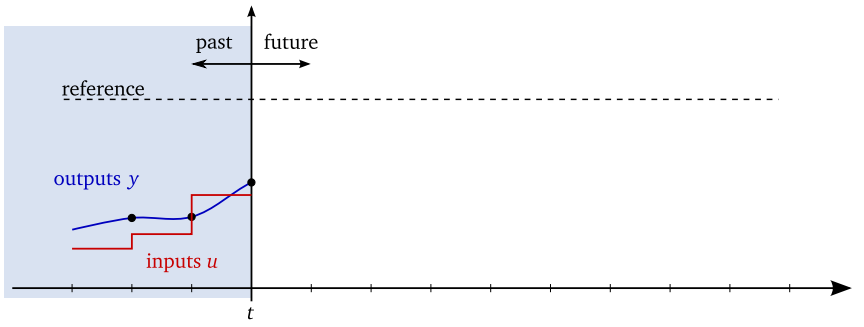
Převzato z Borelli: ME290E a Balandat: EE128

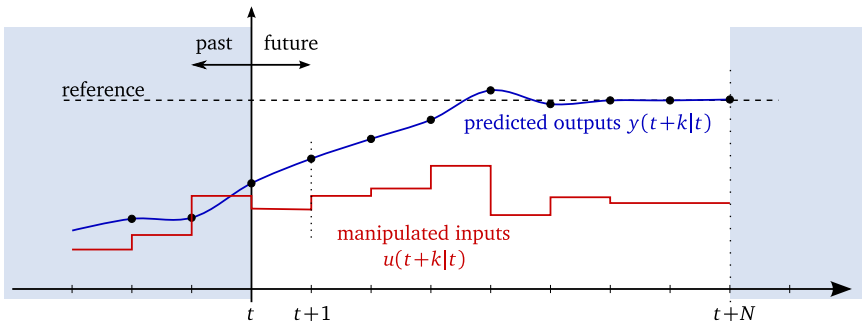


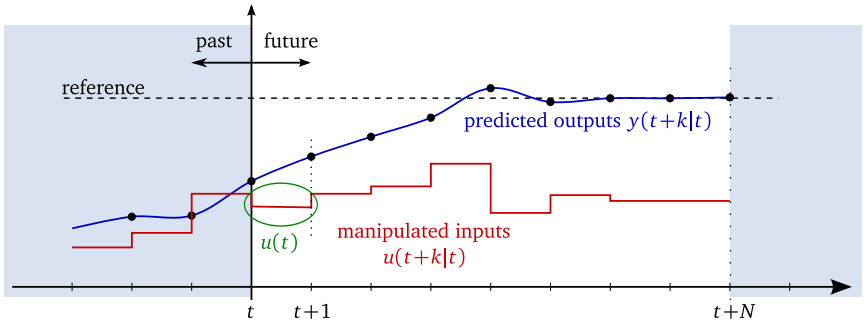


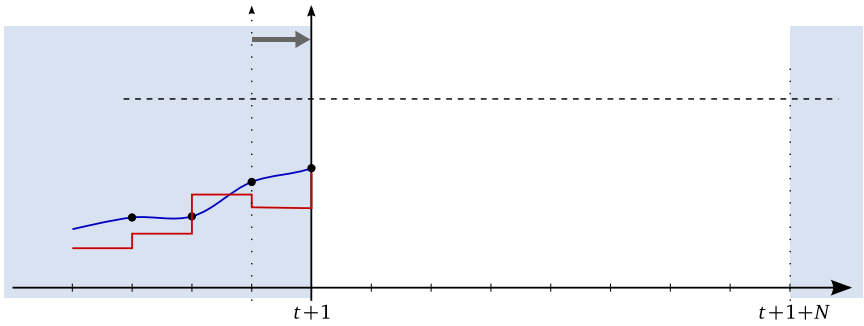
Podle $x(t)$ optimalizujeme vstupy na horizontu délky N

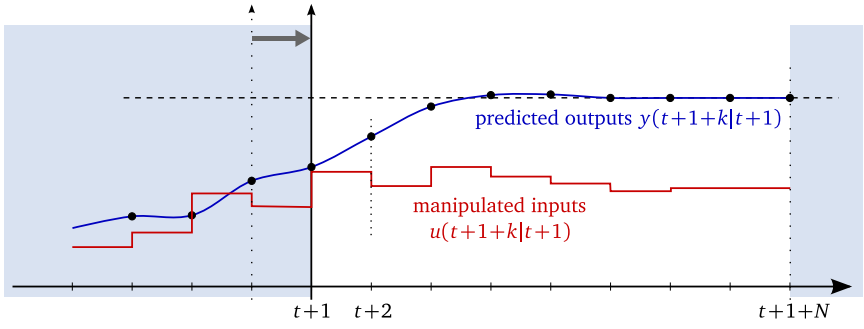
Z celé predikce použijeme pouze první optimální krok $u(t)$

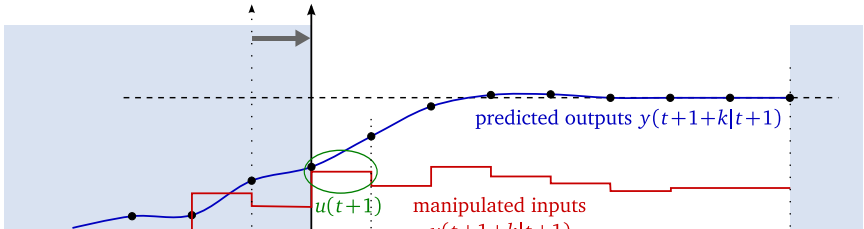

past
future
reference
outputs y
inputs u
t


past
future
reference
predicted outputs $y(t+k|t)$
manipulated inputs
$u(t+k|t)$
t
$t+1$
$t+N$


past
future
reference
predicted outputs $y(t+k|t)$
$u(t)$
manipulated inputs
$u(t+k|t)$
t
$t+1$
$t+N$


$t+1$
$t+1+N$


predicted outputs $y(t+1+k|t+1)$
manipulated inputs
$u(t+1+k|t+1)$
$t+1$
$t+2$
$t+1+N$


predicted outputs $y(t+1+k|t+1)$
$u(t+1)$
manipulated inputs

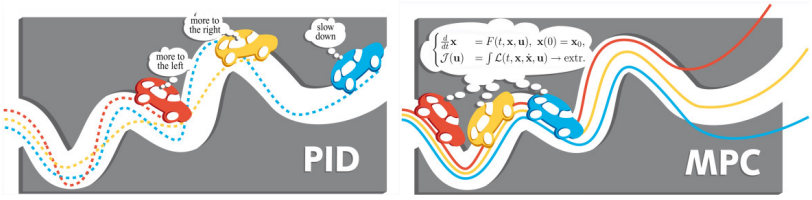
Postoupíme o jeden časový krok a posuneme horizont

Vše opakujeme v čase $t+1$

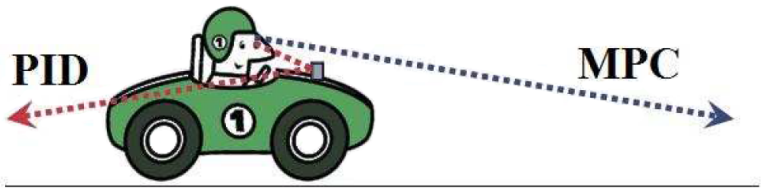
Optimalizace používá současná měření $\rightarrow$ **zpětná vazba**

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:

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

Omezení hodnot přírůstků akčních veličin:

$$\left|\Delta u[k+p|k]\right| \leq \Delta u_{\max}, \qquad p = 0, 1, \ldots, h$$

Omezení hodnot regulovaných veličin:

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

mohou být obecně také proměnná v čase,

u regulovaných veličin $\rightarrow \pm$ *prázdná množina přípustných řešení*

Reference

HAVLENA, Vladimír a Jan ŠTECHA. *Moderní teorie řízení*. Skriptum ČVUT FEL. Praha : Ediční středisko ČVUT, 1999, 297 s.

JAMES, Garreth a Deborah WITTEN a Trevor HASTIE a Robert TIBSHIRANI. *An introduction to statistical learning*. New York : Springer, 2003, 000 s.

ŠTECHA, Jan a Vladimír HAVLENA. *Teorie dynamických systémů*. Skriptum ČVUT FEL. Praha : Ediční středisko ČVUT, 2005, 254 s.

HOMOLOVÁ, Jitka. *?????*. TODO.