

Prediktivní řízení procesů s využitím prvků umělé inteligence

Ing. Jan Antoš, Ph.D.

Teze disertační práce



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Teze disertační práce

Prediktivní řízení procesů s využitím prvků umělé inteligence

Predictive Control of Processes with Utilization of Artificial Intelligence Elements

Autor: **Ing. Jan Antoš, Ph.D.**

Studijní program: P3902 Inženýrská informatika

Studijní obor: 3902V037 Automatické řízení a informatika

Školitel: doc. Ing. Marek Kubalčík, Ph.D.

Oponenti: prof. Dr. Ing. Miroslav Pokorný, VŠB Ostrava
prof. RNDr. Ing. Miloš Šeda, Ph.D., VUT Brno
doc. Ing. Renata Wagnerová, Ph.D., VŠB Ostrava

Zlín, červen 2019

© Jan Antoš, 2019

Vydala **Univerzita Tomáše Bati ve Zlíně** v edici **Doctoral Thesis Summary**.
Publikace byla vydána v roce 2019

Klíčová slova: *prediktivní řízení, MPC, diskrétní řízení, diskrétní dynamické modely, umělá inteligence, neuronové sítě, evoluční algoritmy, nelineární systémy, modelové systémy, nádrže, optimalizace, kvadratické programování, omezení*

Key words: *Predictive Control, MPC, Discrete Control, Discrete Dynamical Models, Artificial Intelligence, Neural Network, Evolutionary Algorithms, Models, Tanks, Optimization, Quadratic Programming, Constraints.*

Plná verze disertační práce je dostupná v Knihovně UTB ve Zlíně.

ISBN 978-80-7454-860-4

ABSTRAKT

Prediktivní řízení procesů je metoda regulace vhodná pro řízení různých typů systémů, která je založená na myšlence využití predikce budoucího chování systému a její optimalizace. Běžně se pro predikci chování využívá modelu systému, a proto je nutné pro správnou funkci prediktivního řízení provést jeho správný výběr a určit jeho parametry. Cílem této práce je aplikace některých prvků umělé inteligence ve vhodných oblastech prediktivního řízení, zejména využití jednoduchých evolučních algoritmů v rámci optimalizace a neuronových sítí jako nelineárních modelů. Je prokázáno, že kromě klasických optimalizačních algoritmů je možné použít i jednoduché evoluční algoritmy pro optimalizaci predikce, přičemž výpočetní náročnost může být srovnatelná. Dále se práce zabývá výběrem vhodných modelových systémů s pomalou dynamikou, jejich odvozením a vytvořením nelineárních modelů v podobě škálovatelných neuronových sítí. Potenciální výhodnost tohoto přístupu pro řízení systémů obtížně popsatelných či pro řízení systémů, jejichž matematicko-fyzikální popis není znám, byla v práci prokázána. Práce se také zabývá možností nasazení nalezených modelů na reálné systémy a stanovením nutných podmínek a požadavků pro jejich aplikaci.

ABSTRACT

Predictive control is a method of control process which is suitable for different types of systems. This method is based on the utilization of prediction of the future behaviour of a system and its optimization. A model of this system is mainly used for prediction, hence it is crucial to choose the model properly and set its parameters. The aim of this work is the application of some elements of artificial intelligence in proper fields of predictive control. It is focused especially on the utilization of simple evolutionary algorithms in the optimization process as well as using neural networks as models of the systems. It has been shown that, besides classical optimization algorithms, it is possible to apply simple evolutionary algorithms with similar computational demands. The process of choosing proper systems with slow dynamics, their derivations of mathematical formulas and the methodology of model creation in the form of scalable neural networks is discussed further. This approach can be convenient for controlling of systems which are difficult to be mathematically described or for systems whose description is not known at all. The possibility of application of these models to real systems, the definition of necessary conditions and the requirements for their applications are discussed as well.

OBSAH

1.	ÚVOD.....	6
2.	SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	6
3.	TEORETICKÁ ČÁST A ZVOLENÉ METODY	9
3.1	Prediktivní řízení	9
3.1.1	Modely.....	9
3.1.2	Lineární modely.....	9
3.1.3	Prediktor	10
3.1.4	Optimalizátor	11
3.1.5	Metody optimalizace	11
3.2	Umělá inteligence.....	12
3.2.1	Neuronové sítě.....	12
3.2.2	Neuronové sítě: učení RBF	13
3.2.3	Evoluční algoritmy	13
4.	CÍLE DISERTAČNÍ PRÁCE	15
4.1	Popis dílčích cílů a způsob jejich realizace.....	15
5.	EXPERIMENTÁLNÍ A METODICKÁ ČÁST	17
6.	VÝSLEDKY A DISKUSE.....	19
6.1	Předběžná analýza a ověření použití prvků umělé inteligence	19
6.1.1	Analýza účelové funkce a vlivu omezení.....	19
6.1.2	Ověření možnosti nasazení evolučních algoritmů	19
6.1.3	Ověření možnosti nasazení RBF jako modelu	20
6.2	Volba systému pro prediktivní řízení a vytvoření modelového systému	21
6.2.1	Modelové systémy – geometrické abstrakce.....	21
6.2.2	Modelové systémy: ověření správnosti	24

6.2.3	Modelové systémy: saturace, singularity.....	25
6.3	Volba a vytvoření modelu.....	26
6.3.1	Model systému popsany neuronovou sítí	26
6.3.2	RBF model systému.....	27
6.4	Identifikace soustavy a nalezení parametrů modelu.....	27
6.4.1	Trénovací množina.....	27
6.5	Vytvoření prediktoru.....	28
6.6	Optimalizace řízení	28
6.6.1	Řízení modelových systémů	29
6.7	Návrh přenosu získaných výsledků na reálný systém	32
7.	ZÁVĚR A PŘÍNOS PRÁCE PRO VĚDU A PRAXI	34
8.	ZÁVĚREČNÉ SHRNU TÍ	34
	SEZNAM POUŽITÉ LITERATURY	36
	SEZNAM OBRÁZKŮ A TABULEK	44
	SEZNAM POUŽITÝCH SYMBOLŮ.....	45
	SEZNAM ZKRATEK.....	47
	PUBLIKAČNÍ AKTIVITY AUTORA	48
	ODBORNÝ ŽIVOTOPIS	50

1. ÚVOD

Prediktivní řízení[1-5] je metoda regulace, jejíž hlavní myšlenkou je využití predikce chování systému pro optimalizaci akčního zásahu. Touto metodou se obvykle řídí spojité systémy s využitím diskrétních modelů[6, 7]. Dynamický spojité model systému je obvykle vyjádřen pomocí diferenciálních rovnic. Matematické operace s touto formou modelů jsou většinou obtížné, a proto se používají diskrétní modely, které využívají k popisu dynamického chování systému rovnice diferenční. Rozšířením tohoto modelu o budoucí hodnoty je pak možno získat predikci budoucího chování systému. Toto rozšíření se nazývá prediktor[8]. Tvar prediktoru je závislý na velikosti predikčního horizontu. Budoucí chování systému je poté možné optimalizovat tak, aby bylo dosaženo žádané hodnoty. Optimalizátor a prediktor společně tvoří regulátor[5]. Predikce chování vychází z modelu systému, a proto má tento model popisovat chování systému co nejpřesněji. Model systému se získá identifikací soustavy[9], a to buď matematicko-fyzikální analýzou, nebo identifikací experimentální[10, 11]. Pro systémy, jejichž přesný matematický model je obtížné nebo nemožné získat, je možno využít neuronových sítí[12, 13] jako modelu. Nevýhodou těchto modelů bývá jejich velká výpočetní náročnost, a proto jsou vhodné pro systémy s pomalou dynamikou. V rámci optimalizačního procesu se obvykle využívá kvadratického programování[14-16], neboť optimalizovaný problém je často kvadratického tvaru. Pro řešení optimalizačního procesu je často využito algoritmů simplexových[17, 18], je možné využít i některé druhy evolučních algoritmů[19] či algoritmů z nich vycházejících[20, 21]. Pokud jsou tyto algoritmy principiálně jednoduché, je možno s nimi dosáhnout výpočetní náročnosti srovnatelné s náročností klasických metod, přičemž tyto evoluční algoritmy je možno použít na celou řadu optimalizačních problémů[22].

2. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Pro nasazení prediktivního řízení se často využívá diskrétních dynamických modelů, kdy časté je nasazení modelů stavových[9, 23, 24]. Pro popis systémů je často využíváno lineárních modelů[17, 25-28], neboť matematická práce s těmito modely může být jednodušší než u modelů nelineárních. Mnoho prací se zabývá redukcí výpočetní náročnosti[29-33], zejména v oblasti optimalizátoru. Tato redukce dává smysl pro řízení procesů s velmi rychlou dynamikou (např. systémů elektronických). Existují však také systémy, kde jejich pomalá dynamika umožňuje nasazení výpočetně náročnějších metod.

V reálném řízení bývá často vzorkovací perioda fixní. Maximální kritická doba pro vyřešení optimalizačního problému je tedy dána touto vzorkovací periodou. Současná úroveň hardwaru umožňuje navíc řešit komplexní matematické problémy velmi efektivně např. masivní paralelizací[34-36] a použitím grafických procesorů[37], a to právě v oblasti prediktivního řízení[38-40]. Proto je možné v prediktivním řízení využívat i jinak velmi výpočetně náročných metod a prvků jako jsou prvky umělé inteligence[41, 42].

Prediktivní řízení lze tedy v současnosti úspěšně nasadit i na řízení nelineárních systémů[43, 44], dokonce je to výhodné, ovšem v rámci omezení uvedených výše a v oblastech, kde se jinak aplikace klasických metod optimalizace setkávají s obtížemi. Ve specifickém případě prvků umělé inteligence, např. evolučních algoritmů se pak nabízí využití takových algoritmů, jejichž výpočetní náročnost je srovnatelně nízká v porovnání s klasickými metodami[20, 45, 46]. Proto by měly být tyto algoritmy principiálně co nejjednodušší, a zároveň dostatečně sofistikované pro splnění kritérií kladených na prediktivní řízení, jako je dostatečná přesnost a splnění omezení signálů.

Pro popis nelineárních systémů se často používají modely stavové[44, 47], či částečně linearizované[48-50], popř. je nutné nasazení pokročilých metod řízení, v tomto kontextu zejména adaptivního prediktivního řízení[51-54]. Při využití neuronových sítí jako modelů[55-61] se uplatní jejich schopnosti aproximace široké škály matematických funkcí. Tento prvek umělé inteligence lze tedy využít pro popis různých druhů systémů s nelineární dynamikou, které jsou předmětem této práce.

Jednou z oblastí, kde se prediktivní řízení může vhodně uplatnit, je řízení hladiny nádrží netriviálního typu, tedy reprezentantů nelineárních systémů, pro které je možné, a za některých podmínek i vhodné, použití prvků umělé inteligence, jak je uvedeno výše.

Regulace výšky hladiny a průtoků kapaliny v nádržích je klíčová v mnoha nejrůznějších odvětvích. Souhrnem literární rešerše v této oblasti je možné uvést, že možnost nasazení prediktivního řízení výšky hladiny v nádržích byla již ověřena, a společně s použitím různě sofistikovaných přístupů k nelinearitě systémů bylo prediktivní řízení uplatněno především na systémech nádrží, jejichž horizontální průřez se s výškou nemění [62-67]. Nádrže jiné geometrie, než s neměnnou plochou hladiny, jsou studovány a v literatuře popisovány podstatně méně často, prakticky se jedná pouze o nádrže kulového a jednoduchého kuželového tvaru [68-73]. Zde je silným zdrojem nelinearity, a s ní spojených obtíží, již sám tvar, zatímco hydrostatický tlak je jen malým přispěvatelem. U nádrží kulového tvaru byly v literatuře nalezeny popisy

aplikace regulátorů odvozených od PID s limitovanými úspěchy [71, 72]. Nasazení prediktivního řízení nebylo zaznamenáno, a z prvků umělé inteligence byl využit algoritmus rojení částic pro odhad parametrů PID regulátoru. U nádrží konického tvaru byla v literatuře prokázána výhodnost prediktivního řízení ve srovnání s řízením pomocí regulátorů odvozených od PID, zejména v případě nelineárního prediktivního řízení [69, 70]. Ve všech relevantních případech se v uvedené literatuře potvrdil lepší výkon sofistikovanějších metod, avšak za cenu výpočetní náročnosti, což je nejpravděpodobnější příčinou doposud nedostatečného množství publikací, věnovaných aplikaci prvků umělé inteligence v prediktivním řízení nelineárních systémů reprezentovaných nádržemi netriviálních tvarů. Explicitní použití neuronových sítí pro řízení hladiny lze najít jen vzácně, zatím jen u soustav nádrží válcových[67] a konických[68]. Obtíže se zvládnutím nelinearity pravděpodobně limitovaly i dosavadní snahy jiných autorů o řešení problematiky spojené s řízením hladiny v nádržích dalších tvarů (mimo koule a kužel), o kterých se v literatuře nedá najít prakticky ani zmínka. Dalším významným deficitem relevantní oblasti literatury je omezenost všech publikací na konkrétní kazuistiky, vždy se jedná o řízení konkrétních systémů, ať už reálných nebo modelových, takže čtenář může využít publikované výsledky na základě analogie jako inspiraci pro svá vlastní řešení, avšak obecné výsledky schází.

Předmětem výzkumu v této práci bude tedy nasazení prvků umělé inteligence v některých částech prediktivního řízení, kdy nasazení evolučních algoritmů bude provedeno alespoň na dílčím systému, a tento algoritmus bude porovnán s klasickými metodami. Dále budou vytvořeny modelové systémy nádrží s různými geometriemi, které odpovídají zobecněním nejčastějších tvarů reálných systémů vyskytujících se v praxi nebo v přírodě. Pro aplikovatelnost a obecnost je nutné odvození modelů klasickým postupem s non-dimenzializací veličin[74-76]. Současně bude pomínut efekt hydrostatického tlaku, který je stejný v nádrži jakéhokoliv tvaru a závisí jen a pouze na výšce, a je zpětně implementovatelný do regulátoru při přechodu na reálný systém. Pro tyto modelové systémy budou vytvořeny obecné modely neuronových sítí, které budou využity v prediktoru regulátoru. Takto získané modely a regulátory by tedy měly být škálovatelné a jejich možnosti nasazení v konkrétní aplikaci široké, vždy pro reálnou nádrž daného typu tvaru. Eventuální dopad výpočetní náročnosti při online optimalizaci se vztahuje ke vzorkovací periodě, tj. ve vztahu k dynamice řízeného systému.

3. TEORETICKÁ ČÁST A ZVOLENÉ METODY

3.1 Prediktivní řízení

Model prediktivní řízení[1-5] MPC (Model Predictive Control) je metoda regulace, která využívá pro predikci chování systému jeho modelu. Predikci lze poté optimalizovat tak, aby bylo dosaženo žádaného stavu systému, a případně splněny další požadavky řízení. Regulátor[8] je pak tedy založen na využití prediktoru a optimalizátoru. Hlavní výhodou prediktivního řízení je jeho schopnost vypořádat se s omezeními[4, 14-16] akčních, řízených i stavových veličin. Další výhodou je jeho aplikovatelnost na systémy, které jsou jinými metodami obtížně říditelné. Jedná se zejména o mnohorozměrové systémy, systémy neminimálně fázové či systémy s dopravním zpožděním. Nevýhodami jsou větší výpočetní náročnost a nutnost návrhu vhodného prediktoru a optimalizátoru.

3.1.1 Modely

V další části se budeme zabývat pouze modely matematickými, zejména modely vstupně-výstupními[3, 17, 26]. Pro získání modelu mohou být použity dva přístupy: matematicko-fyzikální analýza a experimentální identifikace.

Při matematicko-fyzikální analýze je chování systému popsáno pomocí rovnic založených na fyzikálních a chemických zákonech, zejména bilanci hmoty a energie. Využitím tohoto přístupu je poté získán matematický popis systému ve formě algebraických (statické systémy) či diferenciálních rovnic (dynamické systémy) a jeho koeficienty mají fyzikální význam. Nevýhodou pak může být obtížnost nalezení analytického řešení, a proto se tyto výpočty často provádějí numerickými metodami[77, 78].

Při experimentálním přístupu jsou využívány informace o zkoumaném systému získané během experimentu, zejména naměřené hodnoty vstupních a výstupních signálů. Vnitřní struktura systému se nebere v úvahu a systém je považován za „černou skříňku – black box“. Na základě naměřených vstupních a výstupních signálů je zkoumána vstupně-výstupní relace a je určen příslušný matematický model, ovšem koeficienty modelu nemají fyzikální význam, a práce s těmito modely v rámci simulace může být jednodušší. Pokud model kombinuje data a alespoň částečný fyzikální model, je považován za tzv. „grey box“,[28] čistě teoretický model je pak tzv. „white box“.[26, 79]

3.1.2 Lineární modely

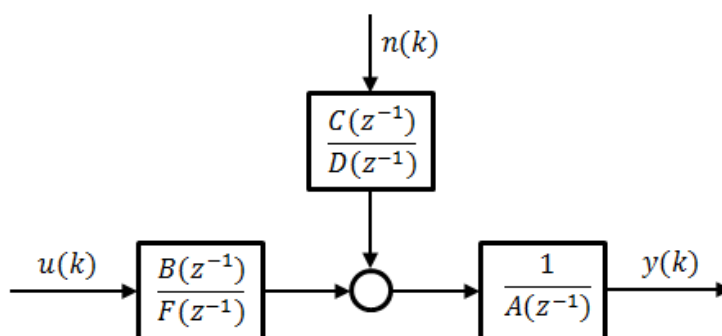
V další části budou uvedeny některé diskrétní lineární modely[17, 26, 80-82], které se pro prediktivní řízení spojitého systému používají.

Impulsní funkce systému popisuje reakci diskrétního systému na jednotkový (Diracův) impuls. **Přechodová funkce** vyjadřuje odezvu systému na jednotkový (Heavisideův) skok. Model systému je popsán podílem polynomů. **Přenos** systému lze zapsat ve tvaru:

$$G(z^{-1}) = \frac{Y(z^{-1})}{U(z^{-1})} = \frac{B(z^{-1})}{A(z^{-1})} \quad (3.1)$$

Obecné lineární modely

Lineární modely jsou modely popsány polynomy $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$, $D(z^{-1})$ a $F(z^{-1})$ a zahrnují kromě vstupu také poruchu systému. Obecné uspořádání těchto modelů je zobrazeno na Obr. 3.1. kde $n(k)$ je bílý šum.



Obr. 3.1: Obecný lineární model, podle ref. [26, 28]

CARIMA (Controlled AutoRegressive Integrated Moving Average)

Vyjdeme-li z ARMAX[26] (AutoRegressive-Moving-Average model with eXogenous input) modelu a zahrneme-li do této rovnice $\Delta = 1 - z^{-1}$ (ΔC zapíšeme opět jako C), dostaneme rovnici CARIMA modelu[3, 26, 83].

$$\Delta Ay(k) = B\Delta u(k) + Cn(k) \quad (3.2)$$

Pro výpočet prediktoru však není vhodné uvádět rovnici v tomto tvaru, neboť $u(k)$ je neznámá veličina, kterou je nutné predikovat. Je proto vhodné použít přímo $u(k - 1)$ a polynom B zapsat s nenulovým absolutním koeficientem.

$$Ay(k) = Bu(k - 1) + \frac{C}{\Delta}n(k) \quad (3.3)$$

3.1.3 Prediktor

Prediktor je nástroj pro předpovídání budoucího chování systému. Délka intervalu predikcí je dána velikostí predikčního horizontu.[1, 3, 4, 84] Rozšířením modelu lze získat prediktor, který je funkcí nejenom minulých, ale také i budoucích hodnot.

$$\vec{y} = f_p(\vec{u}, \vec{y}, \vec{u}, \theta_p) \quad (3.4)$$

kde f_p je rovnice prediktoru a θ_p jsou parametry prediktoru, \vec{u} a \vec{y} jsou minulé hodnoty vstupu a výstupu, \vec{u} a \vec{y} jsou hodnoty budoucí (predikované).

3.1.4 Optimalizátor

Optimalizátor slouží k nalezení sub-optimálního řešení v rámci aktuální řídicí periody. Pro tyto účely se využívá prediktoru. V rámci optimalizace je definována účelová funkce[4] obvykle kvadratického charakteru

$$J = (\hat{y} - \vec{w})^T (\hat{y} - \vec{w}) + \lambda \tilde{u}^T \tilde{u} \quad (3.5)$$

kde \hat{y} je výstup prediktoru, \vec{w} je žádaná hodnota v rámci výstupního horizontu, \tilde{u} je vstup prediktoru a λ volný váhový parametr.

Pro účely aplikace na reálný systém je možné v prediktivním řízení aplikovat ještě různé typy omezení[4, 14, 16]. Jsou-li omezení intervalového typu, lze omezení zapsat jako soustavu nerovnic, kde f_c je zvolená funkce omezení.

$$f_c(\tilde{u}) \leq 0 \quad (3.6)$$

Problém optimalizace se řeší pro aktuální periodu řízení. V rámci celého cyklu řízení je aplikován princip posuvného horizontu[2, 85, 86]. Velikost predikce výstupní veličiny je dána výstupním horizontem $\langle N_1, N_2 \rangle$ (možno zahrnout dopravní zpoždění). Z tohoto vyplývá, že žádaná hodnota \vec{w} musí být v rámci tohoto horizontu známá, či alespoň předvídatelná. Velikost řídicího horizontu je pak N_u .

3.1.5 Metody optimalizace

Kvalita regulace je závislá na výsledku optimalizace, pro kterou se využívají numerické metody[22, 78], které sice nemusí poskytnout řešení globálně optimální, nicméně i sub-optimální řešení může být natolik dobré, aby bylo dosaženo potřebné kvality regulace.

Jednou ze základních tříd jsou algoritmy gradientní[22], které se hodí pro nasazení na unimodální funkce. Základní myšlenkou gradientního algoritmu je fakt, že lokální extrém se nachází ve směru gradientu¹.

Jednou z vlastností prediktivního řízení je možnost zahrnout omezení v rámci optimalizace. V případě lineárního problému je možné zahrnout tato omezení použitím lineárního programování[16-18, 22]. Pro řešení nelineárních problému je pak nutné použít některou z metod nelineárního

¹ Pro hledání minima se postupuje ve směru negativního gradientu.

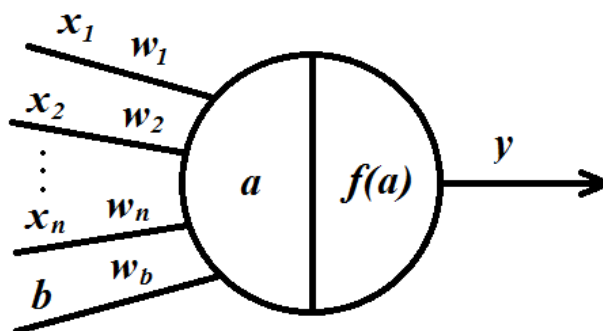
programování[16, 87]. V prediktivním řízení se pak používá kvadratické programování[14-16], které využívá komplementární algoritmy, založené na principu simplexových algoritmů.

3.2 Umělá inteligence

Pojmem umělá inteligence[13, 42, 88] se označují systémy nebo stroje, které vykazují určitý stupeň inteligentního chování. Obor umělé inteligence (UI) je však velmi široký a proto se tato práce zabývá pouze dvěma oblastmi UI a to neuronovými sítěmi[13, 88-90] (NN) a evolučními algoritmy[19-21, 91]. Jedním z důvodů proč se zabývat možností nasazení prvků UI v prediktivním řízení je fakt, že tyto systémy mohou řešit problémy složité či obecné. Jedním z hlavních limitů nasazení UI pak může být často vyšší výpočetní náročnost v porovnání s klasickými metodami. Proto byly zvoleny takové oblasti, které jsou pro toto nasazení vhodné.

3.2.1 Neuronové sítě

Neuronové sítě jsou nástrojem (prvkem) umělé inteligence, jejichž uplatnění lze nalézt v mnoha různých oblastech jako je např. strojový překlad, autonomní systémy či zpracování multimediálních dat. Základem sítě je elementární jednotka zvaná neuron (Obr. 3.2).



Obr. 3.2: Schéma neuronu, volně podle ref.[13, 88]

Neuron obvykle obsahuje množinu vstupů x a jejich příslušných vah w . Někdy také mohou obsahovat další vstup b (případně i jeho váhu w_b), kterému se říká práh. Tento práh má v přírodních sítích obvykle excitační či tlumící funkci.

Výstup neuronu y je pak dán přenosovou funkcí $f(a)$, která může být různého typu, v závislosti na zvoleném typu sítě a řešeném problému. Jelikož různé prameny používají různá značení, budeme dále používat pro libovolný vstup symbol u . Rovnice aktivační funkce poté tedy bude nabývat následujícího tvaru.

$$a = \sum_{i=1}^n u_i w_i \quad (3.7)$$

Neuronová síť je pak množina těchto neuronů, které jsou vzájemně propojené v jeden funkční celek.

Jedním typem sítě, která může být vhodná pro nasazení v prediktivním řízení, je síť **Radial Basis Function**[88] (RBF). Používají se dva typy funkcí Gaussovská

$$f(x) = e^{-\lambda x^2} \quad (3.8)$$

a Inverzní multi-kvadratická.

$$f(x) = \frac{1}{\sqrt{x^2 + \lambda^2}} \quad (3.9)$$

Skalár x i -tého neuronu skryté vrstvy, je určen jako střední vzdálenost vstupního vektoru od vektoru středů

$$x_i = \|\mathbf{u} - \mathbf{c}_i\| = \sqrt{(\mathbf{u} - \mathbf{c}_i)^T \boldsymbol{\Sigma}_i (\mathbf{u} - \mathbf{c}_i)} \quad (3.10)$$

kde \mathbf{u} je vektor vstupů, \mathbf{c}_i je vektor středů a $\boldsymbol{\Sigma}_i$ je norma.

Jednou z výhod této sítě je dobrá interpretovatelnost vlivu[88] nastavení jednotlivých parametrů na výstup této sítě.

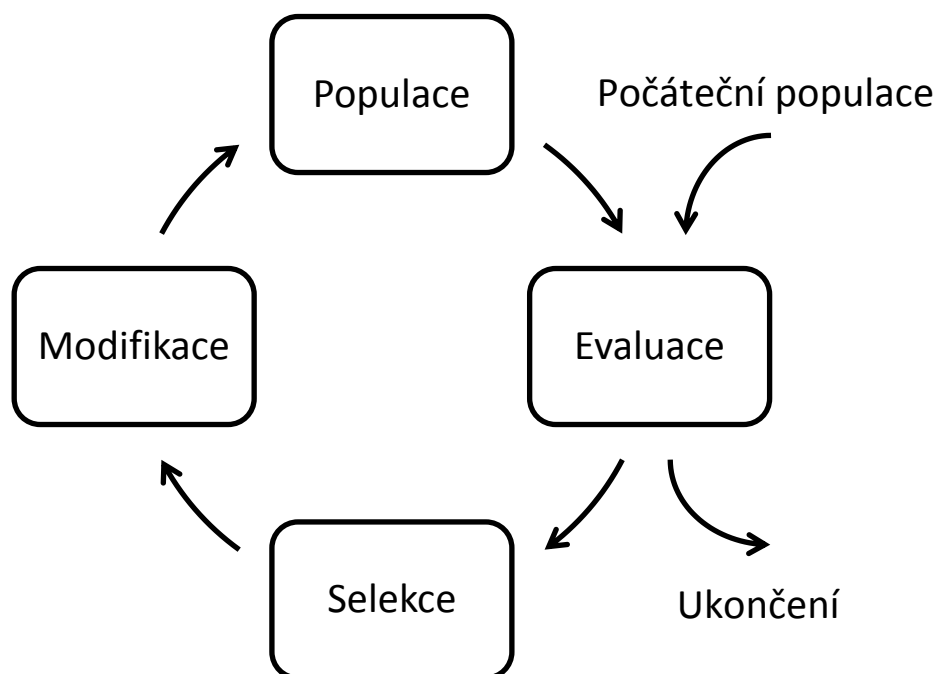
3.2.2 Neuronové síť: učení RBF

Určení parametrů sítě je optimalizační problém, kdy je třeba nalézt parametry modelu tak, aby energetická funkce sítě byla minimální. Protože nelineární optimalizace[88] může být náročná a často není nalezení optimálního řešení nutné, je často vhodné použít některý postup, který vede k řešení sub-optimálnímu. Základní metodou pro výběr středů je metoda k -means[88], jejíž základní myšlenkou je rozdělení vstupních dat na k oblastí, k nimž příslušná data vykazují podobné vlastnosti. Jednotlivé středy jsou poté přiřazeny k nejbližší k -oblasti. Jednotlivé parametry sítě jsou poté upravovány tak dlouho, dokud není dosaženo ukončovacích podmínek.

3.2.3 Evoluční algoritmy

Evoluční algoritmy[19] (EA) je oblast umělé inteligence zabývající se optimalizací[19, 20]. Výhodou těchto algoritmů může být fakt, že mohou řešit komplexní funkce jako např. funkce multimodální, multidimenzionální, multikriteriální či funkce s jistými patologiemi[21]. Nevýhodou pak může být

vyšší výpočetní náročnost, v rámci zvoleného algoritmu, jeho nastavení a typu řešeného problému. EA jsou inspirovány přírodním procesem evoluce[21] a přirozeného výběru. Evoluční vývoj probíhá v cyklech (Obr. 3.3).



Obr. 3.3: Evoluční vývoj, nakresleno volně podle ref.[91-93]

Jako vhodné pro aplikaci v regulaci se jeví algoritmy typu *Hill climbing* a jeho modifikace jako *Tabu search* a *Simulated annealing*. Jako velmi nevhodné se jeví algoritmy, které pracují s velkými generacemi či s velkým počtem cyklů jako *SOMA* a *diferenciální evoluce*.

4. CÍLE DISERTAČNÍ PRÁCE

Hlavním cílem disertační práce je aplikovat prvky umělé inteligence ve vhodné oblasti prediktivního řízení. Konkrétně se jedná o neuronové sítě s případným využitím evolučních algoritmů. Jako vhodnou oblast pro jejich nasazení lze považovat prediktivní řízení systémů, které nejsou (ani snadno, ani úplně) popsitelné matematickým modelem, případně obtížně říditelné pomocí, byť i nelineárních, klasických metod (např. systémy s modely nelineárními, časově variantními, neminimálně fázovými). Tento hlavní cíl se z hlediska praktické realizace rozpadá do následujících dílčích cílů.

4.1 Popis dílčích cílů a způsob jejich realizace

1. Předběžná analýza a ověření použití prvků umělé inteligence

Pro předběžné ověření a prokázání schůdnosti aplikace zvolených prvků řešení na alespoň dílčím příkladu je prvním úkolem je analýza tvaru účelové funkce a vlivu omezení na dosažitelnou oblast. Dalším úkolem je prokázání možností a limitů nasazení evolučních algoritmů v optimalizační části. Třetím předběžným úkolem je pak ověření možnosti nasazení neuronových sítí jako modelu systému na dobře známém fyzikálním modelovém systému.

2. Volba systému pro prediktivní řízení a vytvoření modelového systému

Jako vhodný testovací systém pro prediktivní řízení se jeví průtočná nádrž na kapalinu, ve které se reguluje výška hladiny. S ohledem na potřebnou obecnost hledaného řešení a abstrakci od konkrétního fyzikálního modelu z předběžné studie bude zvoleno šest základních geometrií.

Nejjednodušším modelovým systémem je vertikální válcová nádrž. Dále bude studována nádrž ve tvaru kuželu ve dvou variantách V prvním případě je dno nádrže tvořeno vrcholem kuželu, a tedy, pokud se nádrž plní konstantním přítokem, rychlost zvedání hladiny se zpomaluje, zatímco v případě kuželové nádrže, kde dno je tvořeno podstavou, se při rovnoměrném plnění vzestup hladiny zrychluje. Kombinací obojího jsou situace s nádrží, která se ve střední výšce zužuje, nebo naopak rozšiřuje – dvoj-kužel nebo koule. V realitě pak můžeme obdobné tvary nalézt v různých rezervoárech kapalin, vodních a jiných nádržích, odkalištích, poldrech, přehradách, kanálech a podobně. Efekt hydrostatického tlaku na rychlost výtoku kapaliny z nádrže je pomínut tím, že akčním zásahem je přímo (od)tok kapaliny z nádrže. Matematicko-fyzikální analýzou odvozené modelové systémy budou reprezentovány pomocí S-Funkce v prostředí SIMULINK. Pro dosažení škálovatelnosti získaných výsledků bude nutné modely realizovat pomocí vhodně zvolených bezrozměrných veličin.

3. Volba a vytvoření modelu

System nádrže obecného tvaru bude vykazovat nelineární vlastnosti, a proto se jako vhodné jeví nasazení modelu neuronové sítě, která má schopnost učení se a nevyžaduje dostupnost matematického popisu modelu. Z neuronových sítí připadají v úvahu 2 typy: multi-layer perceptron a síť založené na funkcích radiální báze (RBF). Vzhledem k lepší interpretovatelnosti vlivu parametrů na výstupní funkci se jako vhodný model jeví druhý případ.

4. Identifikace soustavy a nalezení parametrů modelu

Vzhledem k charakteru modelového systému bude zvolena typická vstupně-výstupní sada, která bude získána simulací modelového systému v prostředí SIMULINK. Vzhledem ke kumulační povaze systému je testování pomocí testovacích množin nevhodné. Testovat se bude kvalita regulace, což je výhodné, neboť již v průběhu učení lze vybírat nejlepší kombinaci parametrů.

5. Vytvoření prediktoru

V případě nasazení RBF sítě jako modelu, lze prediktor vytvořit pomocí rekurzivního přístupu, kdy rovnici modelu rozšíříme o budoucí hodnoty. Výhodou je snadná změna velikosti horizontů a možnost zachovat původní model. Vzhledem k pomalé dynamice systému nádrže není zvýšená výpočetní náročnost na závadu.

6. Optimalizace řízení

Tvar účelové funkce bude použit jako v řešení prvního dílčího cíle práce, a vliv požadavků řízení vyplývá z omezení veličin. Vhodný optimalizační algoritmus bude určen empiricky zkoušením dostupných reprezentativních typů algoritmů. Poté budou provedeny experimenty s modelovými systémy a vyhodnocena úspěšnost řízení.

7. Návrh přenosu získaných výsledků na reálný systém

Budou stanovena pravidla a proveden návrh, jak ověřit nebo porovnat řízení reálného systému dosavadními metodami a pomocí nejlepší sítě (sítí). Pokud by byla k dispozici data z vhodného reálného systému, lze toto ověření provést virtuálně.

5. EXPERIMENTÁLNÍ A METODICKÁ ČÁST

Pro dosažení dílčích cílů práce nebylo zapotřebí žádných reálných experimentů.

Byl využit SW Matlab/Simulink ve verzi R2012b a SW Wolfram Mathematica v aktuálních verzích po dobu řešení práce, za využití následující literatury pro tento software. [4, 5, 17, 26, 28, 77, 78, 89]

1. Úkol: Předběžná analýza a ověření použití prvků umělé inteligence

Pro řešení předběžných úkolů analýzy účelové funkce byl využit vlastní kód v SW Wolfram Mathematica, pro ověření nasazení evolučních algoritmů byla využita funkce `quadprog()` a dva vlastní kódy s algoritmem typu „hill climbing“ a „evolučně-gradientním“ algoritmem v SW Matlab, a výsledky byly zpracovány v SW Wolfram Mathematica. Pro ověření nasazení neuronové sítě byl vytvořen vlastní kód neuronové sítě v SW Matlab a pro učení sítě byla využita nelineární optimalizace `NMinimize()` v SW Mathematica.

2. Úkol: Volba systému pro prediktivní řízení a vytvoření modelového systému

Průtočná nádrž na kapalinu byla jako vhodný testovací systém pro prediktivní řízení pojednána pomocí abstrakce a zredukována na základní geometrické tvary – modelové systémy – reprezentující zvolené situace, fyzikální faktory byly taktéž maximálně redukovány s ohledem na potřebnou obecnost hledaného řešení. Vytvoření modelových systémů proběhlo nejprve návrhem na papíru, poté byly vytvořeny jako S-funkce v SW Simulink, ke kterým byly přidány korektivní S-funkce řešící problémy singularit, saturací a podobně, tak aby simulace v SW Simulink poskytovala realistický model v rámci daných omezení a zobecnění.

3. Úkol: Volba a vytvoření modelu

Byl zvolen typ neuronové sítě RBF, typ MLP byl vyloučen. Oproti původnímu záměru vlastní konstrukce „*de novo*“ neuronové sítě typu RBF, bylo s výhodou využito funkce `newrb()` v SW Matlab. Byla zvolena struktura modelu, počet vstupů a výstupů. Vnitřní struktura modelu je předmětem pozdějších variací, zatímco počet vstupů (4) a výstupů (1) je fixní.

4. Úkol: Identifikace soustavy a nalezení parametrů modelu

Pro identifikaci soustavy a určení parametrů modelu bylo nutno provést časové přeškálování modelových systémů a určení vzorkovací periody. Dále byla provedena úprava modelových systémů tak, aby vstupní signál byla akumulace namísto odtoku. Toto další zobecnění je možné při předpokládané znalosti přítoku a systém se tím stává nezávislým na hodnotě přítoku.

Nevýhodou je pak ztráta aktuální integrální informace. Vzhledem k nevhodnosti využití testovací množiny pro kumulační systém byla trénovací množina testována pomocí kvality regulace pilotních experimentů, a postupně vylepšována ve zpětné vazbě po provedení optimalizace (úkol 6). Počet cyklů zpětné vazby byl <10 . Velikost trénovací množiny je cca 12 000 vzorků. Trénovací množina byla (v každém cyklu zpětné vazby) použita k nalezení parametrů modelu určeného funkcí `newrb()`, což probíhalo vždy ve dvou fázích. Nejprve byla měněna hodnota `spread` (rozptyl) až došlo ke konvergenci hodnoty energetické funkce. Onset tohoto ustálení potom určuje minimální efektivní počet neuronů. Poté byl omezen maximální počet neuronů na tuto efektivní hodnotu a byl natrénován a uložen RBF model.

5. Úkol: Vytvoření prediktoru

Prediktor byl vytvořen pomocí rekurzivního přístupu a byl realizován jako procedura v rámci S-Funkce regulátoru v simulačním prostředí Simulink, Regulátor si soubor s naučenou sítí načte a předává prediktoru jako parametr.

6. Úkol: Optimalizace řízení

Obecně kvadratický tvar účelové funkce byl použit jako v řešení prvního dílčího cíle práce, avšak zde bylo využito funkce `fmincon()` v SW Matlab, která umožňuje nelineární optimalizaci s omezením veličin. Byly zkoušeny následující optimalizační algoritmy: `interior-point`, `sqp-legacy`, `active-set` a `trust-region-reflective`, a byl vybrán nejlepší. V průběhu práce byly taktéž upravovány parametry účelové funkce a funkce omezení. Po odladění regulátoru byly s pomocí nejlepšího optimalizačního algoritmu provedeny experimenty s modelovými systémy a vyhodnocena úspěšnost řízení. Pro dosažení shody ustálených hodnot výstupu s žádanou veličinou byla zvolen přístup korekce žádané hodnoty pomocí korekční funkce získané kalibrací žádané hodnoty proti změřenému výstupu. Následně byla korekční funkce zavedena do regulátoru a ověřena funkčnost navrženého řešení.

7. Úkol: Návrh přenosu získaných výsledků na reálný systém

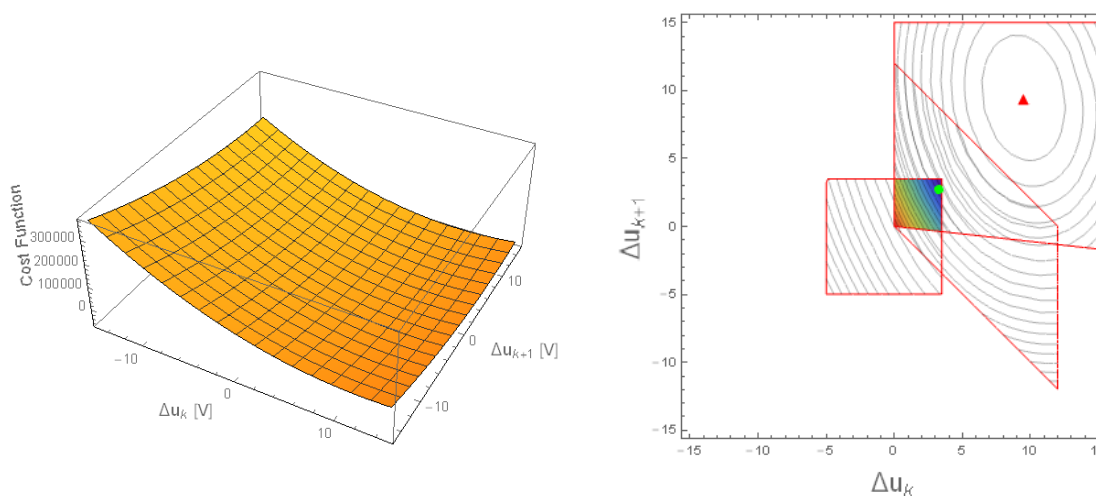
Data z reálného systému nebyla k dispozici, ale i přesto bylo možné stanovit postupy a pravidla pro přenos získaných poznatků na reálný systém. Byla využita škálovatelnost modelu, dále zkušenosti s chováním použitých modelů a procesů, a bylo postupováno ve směru konkretizace a zpětné extrapolace k fyzikálním vlivům, od nichž se na počátku při volbě modelových systémů abstrahovalo.

6. VÝSLEDKY A DISKUSE

6.1 Předběžná analýza a ověření použití prvků umělé inteligence

6.1.1 Analýza účelové funkce a vlivu omezení

Prvotní experimenty byly provedeny na modelu popisujícím tepelnou soustavu, která byla inspirována reálným systémem pece[94], který byl vybrán pro svou jednoduchost. Pro účely prediktivního řízení budeme uvažovat CARIMA model[3] (viz kapitola 3.1.1). Perioda byla získána odhadem – experimentálním změřením odezvy systému na jednotkový skok. Tvar účelové funkce bez omezení a s omezením je zobrazen na Obr. 6.1.



Obr. 6.1: Účelová funkce (vlevo), b) dosažitelná oblast (vpravo), vlastní zdroj[94].

6.1.2 Ověření možnosti nasazení evolučních algoritmů

U stejného modelu jako v předchozí kapitole provedeme ověření možnosti nasazení evolučního algoritmu jako optimalizátoru.[95]. Pro řešení optimalizačního problému byl vybrán algoritmus Hill climbing (kapitola 3.2.3). Kromě klasické verze algoritmu byl aplikován i vylepšený algoritmus, který je založen na principu generování populace pouze ve směru gradientu. Po dosažení hranice je opět využita základní verze Hill climbing. Princip obou algoritmů je zobrazen na Obr. 6.2.



Obr. 6.2: Algoritmus hill climbing (vlevo), algoritmus evolučně-gradientní (vpravo), vlastní zdroj[95].

Řízení pomocí EG algoritmu a pomocí vestavěné funkce quadprog (QP) v Matlabu je pak porovnáno z hlediska výpočetního času, potřebného pro optimalizaci v daný časový okamžik. Maximální výpočetní čas EG algoritmu byl v tomto případě o něco málo lepší, než maximální výpočetní čas algoritmu kvadratického programování. Dále bylo zjištěno, že největší výpočetní náročnosti je dosaženo v oblastech s velkou změnou žádané hodnoty. V ostatních případech je použito analytické řešení. I přes snahu vytvořit porovnatelné podmínky, mohou být tyto výsledky závislé např. na nastavení algoritmů, HW a SW konfiguraci a jiných aspektech. Bylo proto provedeno několik opakování a výsledky porovnání jsou uvedeny v následující tabulce Tab. 1 (menší číslo znamená lepší). Všechny hodnoty jsou podobné, odlišují se až v jednotkách procent. Algoritmy byly také porovnány v závislosti na nastavení řídicího horizontu, a tedy dimenzionality řešení optimalizačního problému a bylo zjištěno, že mají podobný trend.

Tab. 1: Porovnání algoritmů optimalizátoru pro řízení s CARIMA modelem, vlastní zdroj[95]

	Výpočetní. čas [ms]	SSE
EG algoritmus	2,10	234,67
HC algoritmus	2,28	238,29
QP algoritmus	2,01	232,00

Uvedené výsledky ukázaly, že existují i evoluční algoritmy, které mohou být srovnatelné s klasickými metodami a nakolik mohou v rámci vzorkovací periody vyřešit optimalizační problém, tak byla možnost nasazení evolučních algoritmů v prediktivním řízení potvrzena.

6.1.3 Ověření možnosti nasazení RBF jako modelu

Stejně jako u evolučních algoritmů, je i pro použití neuronových sítí v prediktivním řízení nutné provést ověření možnosti nasazení tohoto prvku. Pro tyto účely bude použit modelový systém, který je svými vlastnostmi podobný modelovým systémům uvedených v cílech práce tvořený nádrží, čerpadlem, ventilem a výpustním ventilem, Systém je popsáný rovnicí:

$$S \frac{dh(t)}{dt} = Q(t) - k_1 \sqrt{h(t)} - k_2 \sqrt{h(t)} \quad (6.1)$$

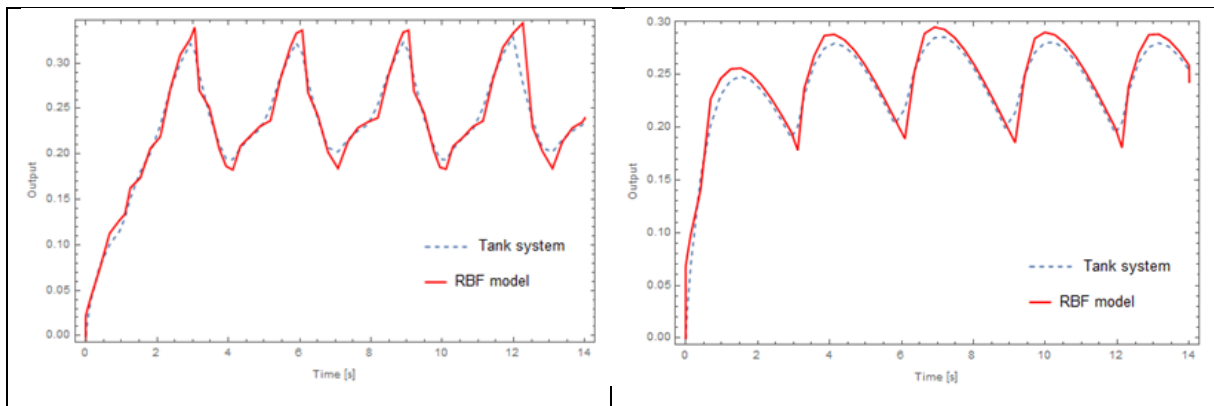
kde Q je objemový průtok vstupního čerpadla, h je výška hladiny v nádrži, S je průřez nádrže, V_1 reprezentuje výstupní ventil a V_2 simuluje odpouštění

kapaliny, a $k_{1,2}$ jsou koeficienty jednotlivých ventilů. V prostředí SIMULINK je tento systém reprezentován S-Funkcí. Pro systém 2. řádu lze systém zapsat jako funkci minulých hodnot (a parametrů).

$$y(k) = f(y(k-1), y(k-2), u(k-1), u(k-2), \Theta(k)) \quad (6.2)$$

Toto lze popsat modelem RBF sítě se 4 vstupy a 1 výstupem. Počet neuronů skryté vrstvy je prvním stupněm volnosti, předpokládejme ale, že pro popis systému bude stačit 5 skrytých neuronů.

Pro účely učení sítě byla v simulaci reálného systému v prostředí SIMULINK vytvořena trénovací množina a sada testovacích množin. Výstupem systému je pak h a vstupem je Q .



Obr. 6.3: Trénování (vlevo) a testování (vpravo) RBF sítě, vlastní zdroj.

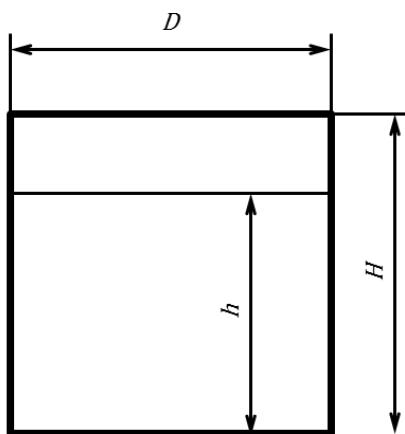
Existuje několik strategií pro trénování RBF sítě [88], v tomto řešeném případě však byla strategie vytvořena vlastní kombinací zvolených postupů. Trénování i testování sítě je zobrazeno na Obr. 6.3. Jak je patrné, RBF síť dobře odpovídá chování reálného systému ve zvolené pracovní oblasti ($R^2 \approx 0.96$). Možnost nasazení neuronové sítě jako modelu byla tedy ověřena.

6.2 Volba systému pro prediktivní řízení a vytvoření modelového systému

6.2.1 Modelové systémy – geometrické abstrakce

Modelové systémy poslouží k simulaci a ke tvorbě trénovacích a testovacích množin. Z důvodu škálovatelnosti jsou všechny veličiny non-dimenzionalizovány. Nejprve nebudou systémy omezené, a otázka omezení a saturace bude diskutována později. Následuje příklad odvození na nejjednodušším modelovém systému, ostatní jsou z důvodu stručnosti uvedeny

v Tab. 2. Nejjednodušším modelem a **příkladem**, který bude uvažován, je **modelový systém válec** výšky H o průměru D , viz Obr. 6.4.



Obr. 6.4: Modelový systém: válec, vlastní zdroj.

Pro další odvození budeme uvažovat výšku $H = 1$ a velikost průměru jako násobek této výšky.

$$D = k_D H \Rightarrow D|_{H=1} = k_D \quad (6.3)$$

Aktuální objem kapaliny ve válci je závislý na aktuální výšce h . Všechny časově závislé veličiny budou označeny (t) pouze při prvním výskytu.

$$V_h(t) = \pi \left(\frac{D}{2}\right)^2 h(t) = \frac{\pi}{4} k_D^2 h \quad (6.4)$$

Změna objemu nádrže q_t je pak popsána následující diferenciální rovnicí.

$$q_t(t) = \frac{dV_h}{dt} = \frac{\pi}{4} k_D^2 \frac{dh}{dt} \Rightarrow \frac{dh}{dt} = \frac{q_t}{\frac{\pi}{4} k_D^2} \quad (6.5)$$

Pro nondimensionalizaci času se bude dále průtok uvažovat v jednotkách [objem nádrže/jednotka času]. Maximální průtok q_{max} je takový, kdy se celý objem ($h = H = 1$) nádrže napustí/vypustí za jednotku času ($\Delta t_{max} = 1$).

$$q_{max} \Big|_{h=H} = \frac{V_{max}}{\Delta t_{max}} = \frac{\pi}{4} k_D^2 \frac{H}{\Delta t_{max}} \quad (6.6)$$

$$q_{max} \Big|_{H=1 \wedge \Delta t_{max}=1} = \frac{\pi}{4} k_D^2$$

Pro pozdější potřeby simulace zavedeme do rovnice přepočet času jako

$$k_q = \frac{1}{\Delta t_{max}} \Rightarrow q_{max} = k_q \frac{\pi}{4} k_D^2 \quad (6.7)$$

Pro další účely odvození budou přítok q_{in} i odtok q_{out} uvažovány v intervalu $\langle 0,1 \rangle$ a tedy zavedením do rovnice (6.5) je získána finální podoba diferenciální rovnice modelového systému.

$$\frac{dh}{dt} = \frac{k_q(q_{in}(t) - q_{out}(t)) \frac{\pi}{4} k_D^2}{\frac{\pi}{4} k_D^2} = \frac{k_q(q_{in} - q_{out})}{1} \quad (6.8)$$

Z rovnice je patrné, že změna výšky hladiny (objemu) je nezávislá na průměru nádrže. Pro dosažení přenositelnosti je dále pomínut efekt hydrostatického tlaku, přičemž řízen bude samotný odtok z nádrže $q_{out} = u$

$$\frac{dh}{dt} = \frac{k_q(q_{in} - u)}{1} = \frac{k_q a(t)}{1} \quad (6.9)$$





kde a je akumulace kapaliny v nádrži.

Ostatní modelové systémy

Diferenciální rovnice modelového systému je tvaru (6.10), kde $d_{ms}(h)$ je jmenovatel rovnice, který byl odvozen pro každý konkrétní systém (Tab. 2).

$$\frac{dh}{dt} = k_q \frac{a(t)}{d_{ms}(h)} \quad (6.10)$$

Tab. 2: Jmenovatel diferenciální rovnice modelového systému, vlastní zdroj

Tvar	Název modelového systému	Jmenovatel modelového systému, $d_{ms}(h)$
	válec	1
	kužel	$3h^2$
	komolý kužel	$3 - 6h + 3h^2$
	dvojkůžel s úzkým hrdlem	$3 - 12h + 12h^2$



dvojkůžel

$$\text{pro } h \in \langle 0; 0,5 \rangle, \quad 12h^2$$

$$\text{pro } h \in \langle 0,5; 1 \rangle, \quad 12 - 24h + 12h^2$$

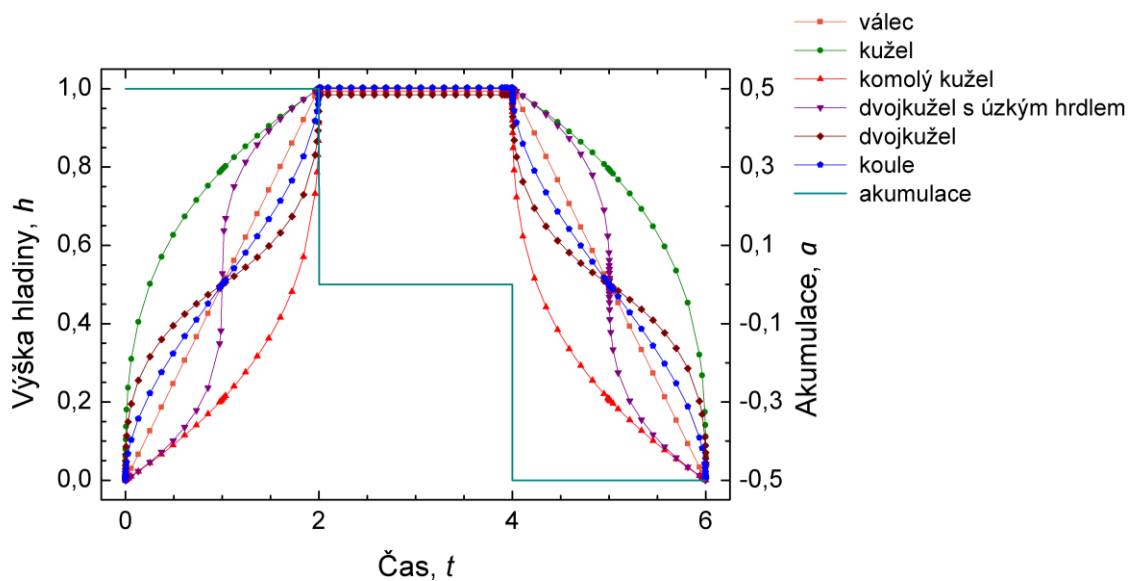


koule

$$6h - 6h^2$$

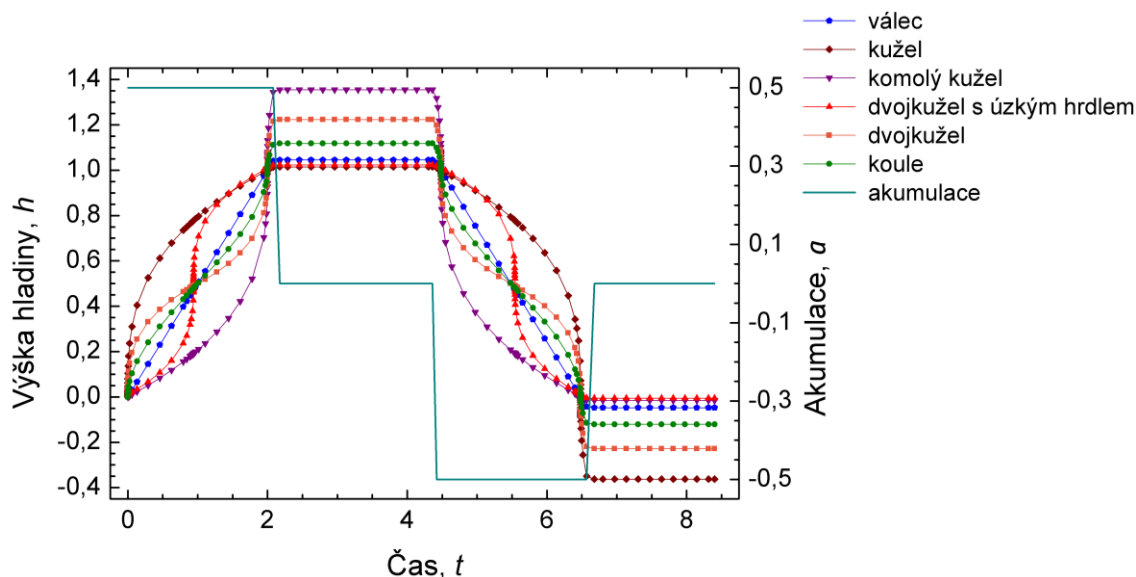
6.2.2 Modelové systémy: ověření správnosti

Po odvození modelových systémů je třeba ověřit, že získaný matematický popis odpovídá navrženým geometrickým vlastnostem a zároveň je správně provedena nondimensionalizace veličin. Pro testování byl zvolen konstantní přítok $q_{in} = 0,5$ a přepočtový faktor $k_q = 1$.



Obr. 6.5: Modelové systémy – ověření správnosti, vlastní zdroj.

Testovací simulace modelové systémy ověřila, viz Obr. 6.5. Dále byly provedeny simulace s prodlouženými časovými intervaly pro studium chování mimo definované meze. Jak je vidět na Obr. 6.6, výstupní veličina modelových systémů se pohybuje i mimo definovaný interval $\langle 0; 1 \rangle$, kde může vykazovat odlišné vlastnosti od vlastností očekávaných.



Obr. 6.6: Modelové systémy – nesaturované, vlastní zdroj.

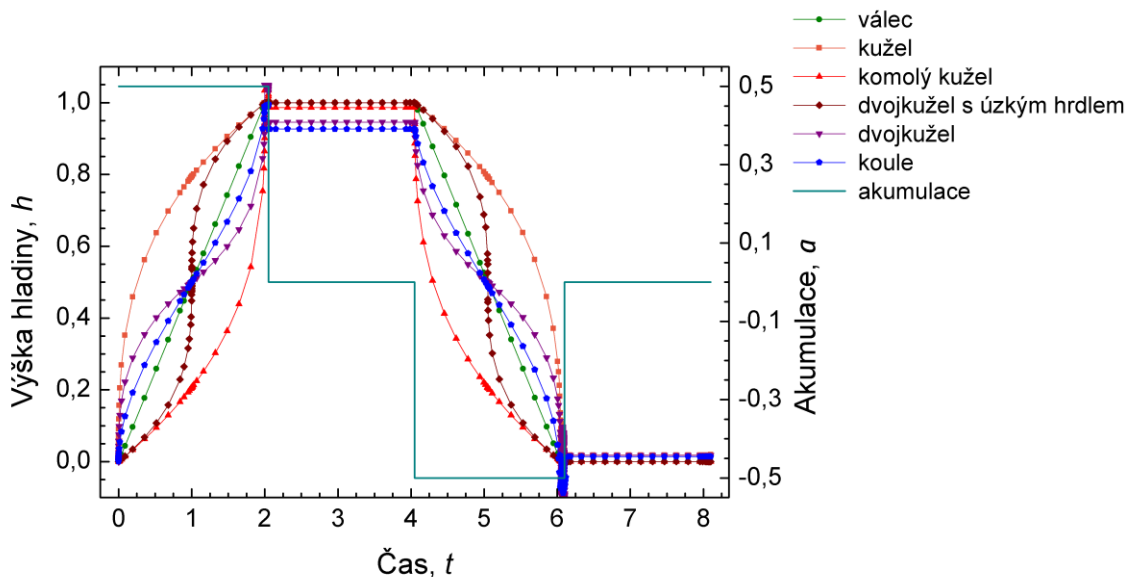
6.2.3 Modelové systémy: saturace, singularity

Získané modely nejsou omezeny v oboru hodnot výstupní veličiny. Pro účely simulace by ale mohlo být vhodné získat modely saturované. Přímé omezení výstupní veličiny je v případě S-Funkce (level 1) obtížné, a může vykazovat několik negativních efektů. Jedním z možných řešení je úprava jednotlivých modelových systémů zavedením funkce přepadu a dna, což vlastně odpovídá i fyzikální realitě.

Čítatel modelu lze tedy rozšířit o přepad q_{of} (overflow) a q_b dno (bottom).

$$\frac{dh}{dt} = \frac{k_q(+q_{in} - q_{of} + q_b - q_{out})}{3h^2} \quad (6.11)$$

Přepad i dno lze poté řídit tak, aby funkce byla omezená pro všechny modelové systémy.



Obr. 6.7: Modelové systémy - saturované, vlastní zdroj.

Jak je patrné z grafu v Obr. 6.7, omezení modelových systémů funkcí přepadů a dna jsou účinná. Nicméně, i tak signály mohou trpět jistými dalšími anomáliemi jako je kmitání v oblasti limitních hodnot výstupní funkce, či ustálení signálu na aktuální hodnotě překmitu (nebo podkmitu). Toto může být způsobeno jednak tím, jak funguje samotná S-Funkce, jednak geometrickým charakterem jednotlivých modelů. Byl zjištěn výskyt i dalších anomálií, způsobený chováním diskrétního výpočtu v blízkosti singularit, zejména v oblasti s neomezenou rychlostí průtoku (tvar typu špička a tvar kulového vrchlíku). Tomuto se dá předejít několika způsoby jako úpravou matematického popisu, úpravou dynamiky signálů (např. „zaoblením“ hran), testování/sledování stavů či omezení pracovní oblasti na „bezpečnou zónu“.

6.3 Volba a vytvoření modelu

Jak již bylo zmíněno v teoretické části, pro popis algebraicky těžko popsatelného dynamického systému může být výhodné použít neuronové sítě.

6.3.1 Model systému popsaný neuronovou sítí

Z matematického hlediska jsou neuronové sítě, stejně jako ostatní modely, popsány svými vstupy, výstupy (př. stavy) a svými parametry. Matematicky se dá neuronová síť zapsat jako

$$\mathbf{y} = f_{NN}(\mathbf{u}, \boldsymbol{\theta}_{NN}) \quad (6.12)$$

kde \mathbf{y} je výstupní vektor sítě, \mathbf{u} je vstupní vektor sítě a $\boldsymbol{\theta}_{NN}$ jsou parametry modelu. Struktura a přenosové funkce jsou popsány funkcí f_{NN} .

Dále budeme předpokládat, že i při aplikaci neuronových sítí jako modelu jsou požadavky na řízení procesu splněny.

6.3.2 RBF model systému

Při návrhu struktury sítě je nejprve nutné určit počet vstupů a výstupů. Jelikož se bude řídit vždy jedna nádrž, která má pouze jednu výstupní (stavovou) veličinu a to výšku hladiny, bude počet výstupů roven jedné. Pro popis dynamického systému je nutné zařadit do modelu paměť. Zařazením minulých hodnot výstupní veličiny na vstupy neuronové sítě se z dopředné sítě stane síť rekurentní, čímž síť získá paměť stavu. Z charakteru modelových systémů je patrné, že model systému by měl být alespoň 2. řádu. Počet vstupů neuronové sítě bude tedy roven čtyřem. Jelikož neuronová síť RBF obsahuje obvykle jednu skrytou vrstvu, je nutné určit počet neuronů skryté vrstvy. Určením parametrů modelu je předmětem další kapitoly.

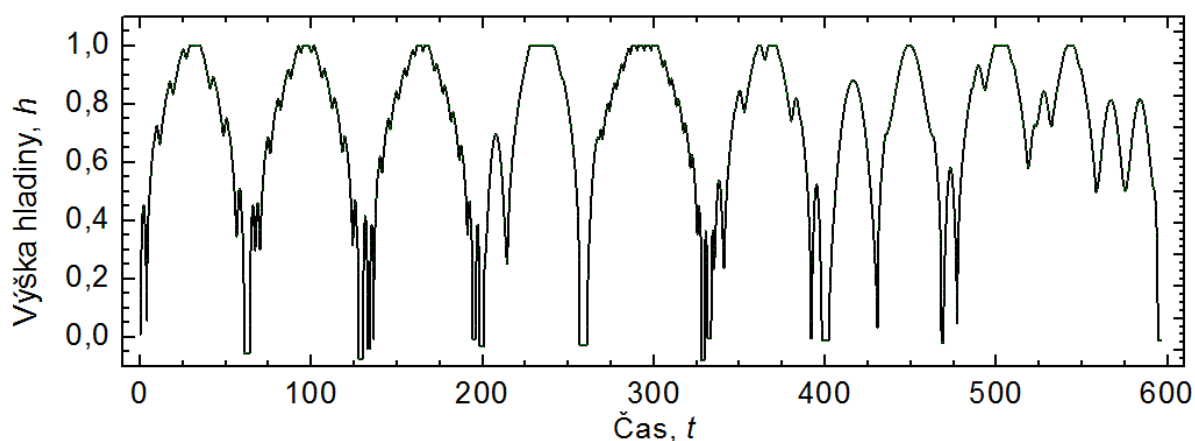
6.4 Identifikace soustavy a nalezení parametrů modelu

Pro identifikaci soustavy bude použit experimentální přístup, kdy experimentu bude podroben spojitý systém, reprezentovaný spojitou S-Funkcí v Simulinku. Pro účely snadnější simulace byl modelový systém časově přeškálován nastavením parametru $k_q = 0,075$. Dále je pro zachování škálovatelnosti předpokládána znalost okamžitého přítoku q_{in} , a proto je možné řídit systém přímo pomocí akumulace a . Po zavedení tohoto opatření je nutné změnit také model systému, neboť omezení systému zavedením funkcí přepadu a dna (6.2.3) je uzpůsobeno pro omezení přítoku a odtoku. Proto je kladná akumulace považována za přítok a záporná za odtok.

Jelikož model popisující spojitý systém má diskrétní charakter, je potřeba určit vzorkovací periodu. Perioda vzorkování by měla být taková, aby respektovala dynamiku systému. Zároveň by však měla být taková, aby byla časová velikost predikce dostatečná při zachování co nejmenší výpočetní náročnosti. Proto byla velikost periody nastavena na $\Delta t = 0,1$.

6.4.1 Trénovací množina

Tvar vstupního signálu byl zvolen tak, aby byla zjištěna reakce systému na změny v různých úrovních pracovní oblasti. Výsledná trénovací množina (část zobrazena na Obr. 6.8) pak tedy obsahuje množinu jednotlivých cyklů napuštění a vypuštění nádrže s různě rychlými změnami signálu.



Obr. 6.8: Část trénovací množiny pro RBF model systému (zobrazeno bez vstupu), vlastní zdroj.

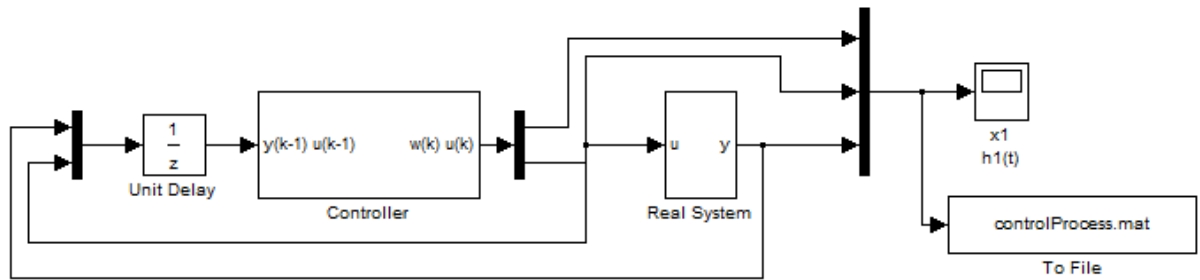
Získaná trénovací množina je využita pro trénování, které je pak provedeno ve dvou fázích. Nejprve je určen efektivní počet neuronů, a poté je provedeno finální trénování.

6.5 Vytvoření prediktoru

K tvorbě prediktoru lze využít nejméně dvou přístupů. Prvním je rozšíření samotného modelu[96] systému využitím trénovacího procesu pro přímé vytvoření prediktoru. Výhodou je zmenšení výpočetní náročnosti predikce, a tedy i optimalizace. Nevýhodou je složitější proces trénování a dále nutnost změny struktury a nové přetrénování prediktoru při změně horizontů. Druhým přístupem je rekurzivní přístup, kdy je model systému považován za funkci, kterou lze opakovaně volat jako proceduru. Výhodou je snadná změna prediktivního horizontu. Nevýhodou je zvýšená výpočetní náročnost predikce (a optimalizace), ale u dostatečně pomalých procesů může být perioda optimalizace taková, aby bylo možno tento přístup použít. Proto byl prediktor realizován rekurzivně jako procedura v rámci S-Funkce regulátoru.

6.6 Optimalizace řízení

Velikost horizontu byla zvolena následovně $\{N_l, N_2, N_u\} = \{1, 3, 2\}$, pro dosažení nízké výpočetní náročnosti. Proces regulace byl simulován v prostředí Simulink, viz Obr. 6.9.

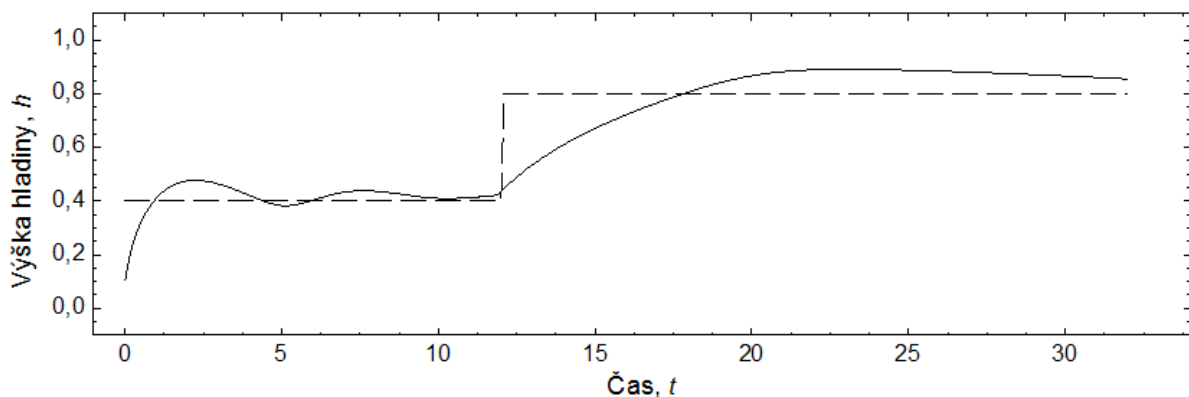


Obr. 6.9: Simulace prediktivního řízení modelových systémů, vlastní zdroj.

Optimalizátor v regulátoru využívá vestavěnou funkci prostředí Matlab `fmincon()`, kterou je možné použít pro nelineární optimalizaci. Jako optimalizační algoritmus se osvědčil algoritmus *trust-region-reflective*.

6.6.1 Řízení modelových systémů

Pilotní experimenty byly z důvodu vyšší výpočetní náročnosti provedeny pro kratší časové úseky než simulace pozdější. Bylo provedeno několik úprav, jak v oblasti trénovací množiny, tak v oblasti regulátoru. Ačkoliv se zprvu zdálo, že řízení není úspěšné, bylo později zjištěno, že průběh regulace odpovídá charakteru modelového systému a jeho nelineární dynamiky v různých částech pracovní oblasti (Obr. 6.10).

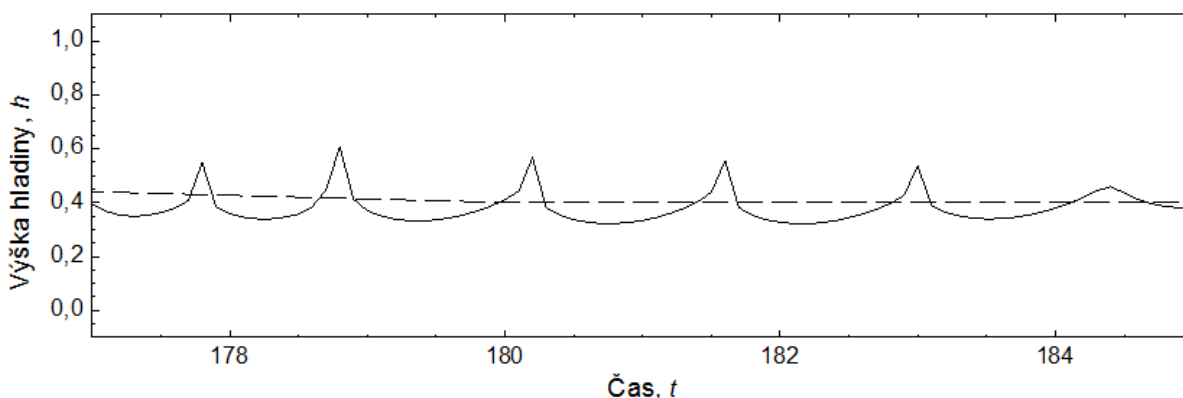


Obr. 6.10: Pilotní experiment – dynamika systému v různých úrovních, vlastní zdroj.

Po odladění optimalizačního procesu byla provedena simulace pro pět modelových systémů. Původní výsledky časových průběhů řízení vykazovaly drobné regulační odchylky od ustáleného stavu. To může být způsobeno tím, že se optimalizace řízení provádí pomocí modelu systému, který však nemusí popisovat reálný systém úplně přesně. Dalšími důvody mohou být nastavení regulátoru či nemožnosti přímého zapojení aktuální výstupu systému do regulátoru (algebraická smyčka). Překonat problém trvalé regulační odchylky je možné např. dalším zpřesněním modelu, úpravou trénovací množiny, úpravou regulátoru, korekcí vstupní, výstupní či žádané veličiny, či zavedením

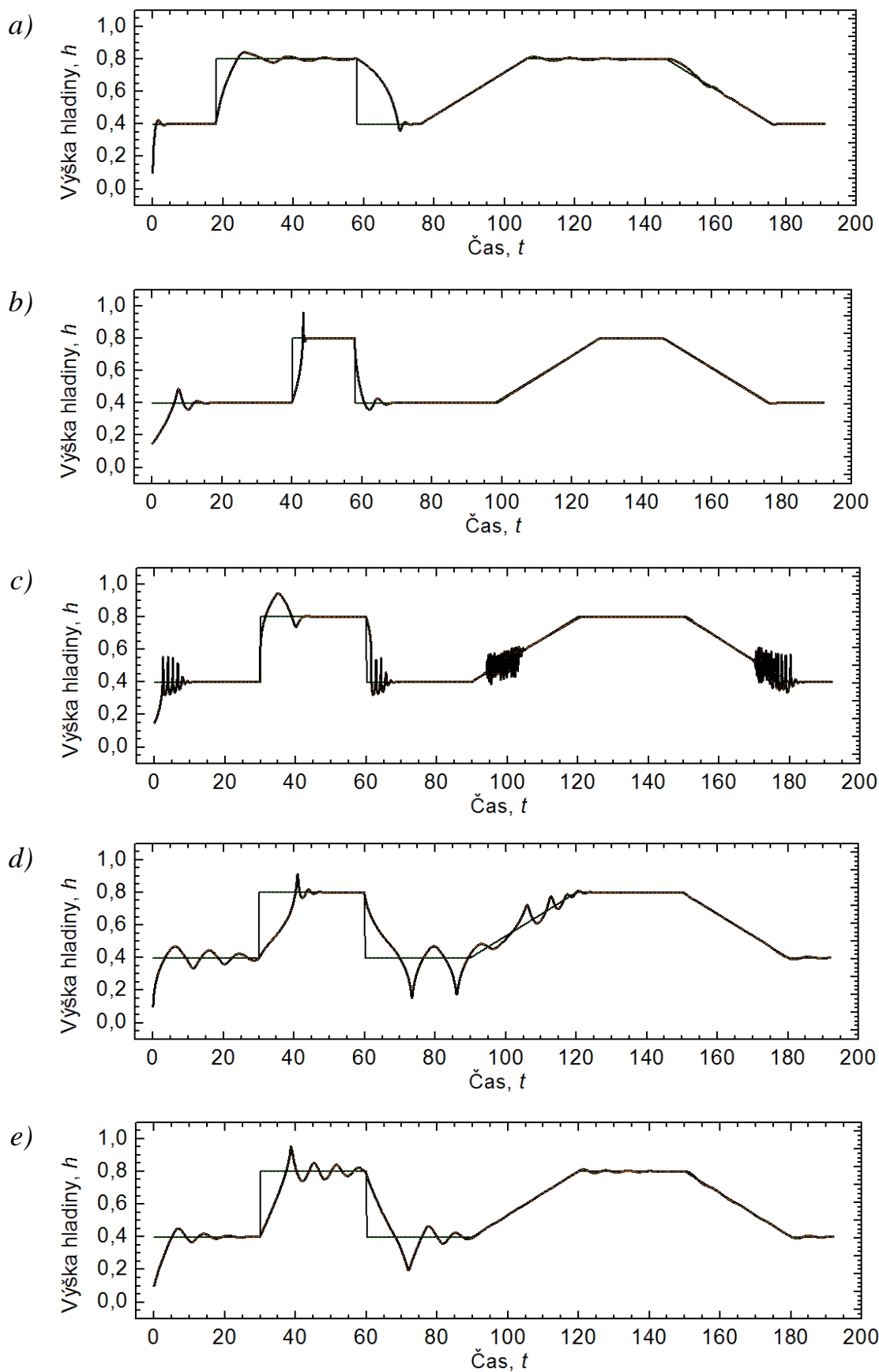
adaptivního přístupu. Pro účely simulace byl zvolen přístup korekce žádané hodnoty. Po změření ustálených hodnot byly vytvořeny korekční křivky pro všechny modelové systémy. Výhodou použití korekčních křivek je fakt, že je možné tyto křivky postupně upravovat zahrnutím nových měření.

Výsledky časových průběhů s korekcemi jsou zobrazeny na Obr. 6.12. Modelový systém válec byl záměrně vynechán, kvůli své lineární povaze. Na časových průbězích je velmi patrná změna dynamiky systému v různých úrovních pracovní oblasti. Zajímavý je také průběh (c) pro řízení modelového systému s úzkým hrdlem, kde je velmi patrná střední oblast obsahující singularitu. Přestože se může zdát, že u střední části není řízení moc dobré, je nutné si uvědomit, že jde o matematický modelový systém, kde rychlost proudění tekutiny ve střední části systému může být neomezená. Vzhledem k tomuto faktu je řízení ve střední části relativně dobré, jak je zřejmé z detailního zobrazení na Obr. 6.11.



Obr. 6.11: Řízení modelového systému s úzkým hrdlem v oblasti singularity, vlastní zdroj.

Průběhy *d)* a *e)* jsou podle předpokladů podobné, neboť se podobají i tvarově. Je možné si povšimnout kmitavého chování v okolí rychlých změn žádané hodnoty. Pokud se navíc stav systému nachází blízko hranic/singularit, může docházet k poměrně velkým výkyvům. Tomuto jevu lze předejít poměrně snadno úpravou žádané hodnoty tak, aby její změna byla pozvolnější. Poté je možné dosáhnout relativně dobrého řízení.



Obr. 6.12: Časový průběh řízení výšky hladiny pro modelové systémy s upravenou žádanou hodnotou, vlastní zdroj.

6.7 Návrh přenosu získaných výsledků na reálný systém

Získané modely jsou aplikovatelné na reálné systémy pouze za určitých podmínek. Odvozené modely jsou zcela platné pouze při splnění uvedených předpokladů. Diferenciální rovnice popisující systémy obsahují veličiny přítoku a odtoku, kdy přítok minus odtok je považován za akumulaci, přičemž je zanedbán hydrostatický tlak, a fyzikální limity proudění kapalin vůbec. Předpokládá se tedy, že tyto veličiny lze přímo řídit, či alespoň měřit, což lze v praxi provést např. pomocí regulátorů, čerpadel či měřičů průtoku. Dále je třeba si uvědomit, že odvozené modelové systémy byly non-dimenzionalizovány. Proto je nutné reálné rozsahy přemapovat tak, aby byly v rozsazích, platných pro modelové systémy.

Pro použití naučených neuronových sítí je nejdříve nutné změřit dobu naplnění/vyprázdnění nádrže. Neuronové sítě byly trénovány na časově přeškálovaných modelových systémech s koeficientem $k_q = 0,075$. Čas naplnění/vyprázdnění modelového systému při maximálním přítoku/odtoku je tedy převrácená hodnota této konstanty.

$$k_q = \frac{1}{\Delta t_{max}} \Rightarrow \Delta t_{max} = \frac{1}{0,075} = 13, \bar{3} \quad (6.13)$$

Vzorkovací perioda v simulacích byla zvolena $\Delta t = 0,1$, takže reálná vzorkovací perioda odpovídá poměru 1:133. V každém jiném případě je nutné převzorkovat na frekvenci odpovídající tomuto poměru. Alternativně je možné provést přetrénování modelu pro novou periodu, což je ovšem mnohem náročnější postup, a jeho nasazení by bylo praktické, pouze pokud je reálná vzorkovací frekvence nižší, než je třeba.

Při aplikaci modelu na reálné systémy bude potlačen výskyt scénářů selhání spojených s výskytem singularit. Na druhou stranu, situace spojené se saturací jsou v praxi řešeny reálně přepadem, nebo uzavřením výpusti.

Uvedené modely lze použít pro systémy, které jsou tvarově příbuzné modelovým systémům, jejichž výběr ovšem pokrývá významné množství případů. Nejlépe by bylo vždy vhodné modely natrénovat přímo z dat reálného systému. K tomu je však zapotřebí velkého množství experimentů, a to v celém rozsahu pracovní oblasti, popřípadě disponovat dlouhou historickou datovou řadou, z níž by bylo možné trénovací množinu vybrat či doplnit. Vzhledem k nedostupnosti reálných dat je aplikace těchto modelů na reálné systémy nad rámec této disertační práce. Přesto lze oprávněně formulovat následující pravidla odhadu:

1. Doba naplnění/vypuštění nádrže je pokryta škálováním času kompletně.
2. Objem nádrže je pokryt škálováním kompletně.
3. Korekční křivka žádané hodnoty je pokryta škálováním kompletně.
4. Model platný pro jeden typ nádrže platí i pro všechny podobné nádrže.
5. Efekt škálování je ovšem širší, než předchozí bod, neboť z rovnic vypadl přepočtový faktor nádrže (tj. poměr průměru a výšky), a tedy na tomto poměru při daném tvaru rovnice rychlosti změny výšky hladiny nezáleží, což platí tehdy, pokud je objem nádrže lineárně závislý na tomto poměru.
6. Pokud je tvar zcela nezávislý ve všech třech rozměrech, uplatní se fakt, že ze všech poměrů mezi výškou a průměrem byl u použitých modelových systémů volen ten, který uzavírá relativně největší objem; je to poměr 1:1, který představuje nejmenší extrém z obou stran, a dá se se tudíž považovat za dobrý střední odhad, což platí i pro tvary složitější než pláště válce, kuželů, koule a ostatních centrosymetrických a rotačně symetrických těles, pokud si vybereme správný typ modelu.
7. V praxi bude také zapotřebí znát a použít přepočtovou funkci mezi reálnou akční veličinou (např. výškou vytažení stavidla, úhlem otočené kohoutem, atd.) a akční veličinou modelovou, tj. musí být známa inverzní charakteristika daného zařízení (výpustě).
8. Efekt hydrostatického tlaku, je stejný v nádrži jakéhokoliv tvaru a závisí jen a pouze na výšce, a je zpětně implementovatelný do regulátoru, až by byly známy konkrétní rozměry nádrže, pokud již nebude implicitně zahrnut v přepočtové funkci regulačního zařízení.

Výpočetní náročnost je kritická teprve až vzhledem k reálné době vzorkovací periody, tedy $1/133$ doby, za kterou se nádrž napustí/vypustí při maximální/minimální hodnotě akumulace. V případě nutnosti by jednou z dalších cest optimalizace regulačního procesu mohla být paralelizace nezávislých prvků umělé inteligence, což by snížilo výpočetní náročnost a umožnilo nasazení i na procesy s rychlejší dynamikou. Při dostatečném počtu jader/vláken by pak v ideálním případě nemuselo záležet na velikosti množiny (počtu neuronů či jedinců) a výpočetní náročnost modelu by se pak zredukovala pouze na počet průchodů (vrstev neuronové sítě, iterací evolučního algoritmu). Toho může být dosaženo použitím paralelního výpočtu při simulaci modelu, případně využitím jader grafického procesoru.

7. ZÁVĚR A PŘÍNOS PRÁCE PRO VĚDU A PRAXI

V experimentální části práce byl vytvořen algoritmus využívající principů klasických i evolučních metod a porovnán s klasickou metodou kvadratického programování. Bylo ukázáno, že může být dosaženo podobné výpočetní náročnosti (v závislosti na nastavení) jako u metody QuadProg(). Později bylo dokonce zjištěno, že výpočetní náročnost by mohla být dále snížena pomocí paralelizace, nicméně řešení této otázky je již nad rámec této disertační práce. Algoritmus přesto obstál i v neparalelizované verzi.

Dále byly v rámci disertační práce odvozeny matematické tvary modelových systémů (nádrží), jejichž tvar je podobný tvarům systémů vyskytujících se v praxi a v přírodě. Uvedené modelové systémy byly navíc non-dimenzionalizovány, což umožňuje jejich škálovatelnost a nasazení na různé typy systémů s různými rozsahy při splnění podmínek pro jejich aplikovatelnost. Dále byly vytvořeny modely systémů založené na neuronové síti RBF. Byly vytvořeny trénovací množiny a parametry těchto modelů byly určeny pomocí trénování na modelových systémech. Dále byly určeny jejich kalibrační křivky pro ustálené hodnoty výstupu. Jelikož původní modelové systémy byly non-dimenzionalizovány a škálovatelné, mají také vytvořené modely i kalibrační křivky tyto vlastnosti. Při splnění podmínek aplikovatelnosti je tedy možné tyto modely nasadit na řízení systémů reálných. Vzhledem k nedostupnosti dat a nemožnosti experimentu s reálným systémem např. vodní nádrže, jsou v poslední kapitole disertační práce analyzovány možnosti přenosu získaných modelů do praxe a formulovány zásady, jak postupovat, čímž byl splněn poslední z cílů práce.

8. ZÁVĚREČNÉ SHRNUTÍ

Tato práce se zabývá možnostmi nasazení některých prvků umělé inteligence ve specifických částech prediktivního řízení. Jedná se především o využití neuronových sítí jako modelu systému a evolučních algoritmů v rámci optimalizace řízení.

Práce je rozdělena do sedmi hlavních kapitol, kdy v první kapitole jsou nastíněné základní principy prediktivního řízení a ve druhé kapitole je provedena rešerše současného stavu problematiky nelineárních systémů ve vztahu k výpočetní komplexitě, prediktivnímu řízení a aplikaci prvků umělé inteligence, přičemž hlavní část je věnována oblasti řízení hladiny kapaliny v nádržích, která byla vybrána jako vhodná pro řešení daného zadání. Ze souhrnu rešerše vyplynula obecná formulace úkolů práce.

Následuje třetí kapitola, která poskytuje teoretický rámec, kde v rámci prediktivního řízení jsou popsány jednotlivé jeho prvky, jako jsou obvykle používané modely, tvorba prediktoru a optimalizátoru a základní metody optimalizace. V druhé části teoretického rámce jsou popsány prvky umělé inteligence zejména struktura a vytvoření neuronové sítě Perceptron, Multi-layer Perceptron a síť založená na funkcích typu radiální báze RBF. Dále je pak stručně popsán princip evolučních algoritmů a jejich stručný přehled.

Čtvrtá kapitola popisuje hlavní cíl práce, což je aplikace některých prvků umělé inteligence ve vhodných oblastech prediktivního řízení. Dílčí cíle pak popisují jednotlivé kroky, které jsou nutné pro splnění cíle hlavního. Těmito dílčími cíli jsou předběžná analýza a ověření použití prvků umělé inteligence, volba vhodných modelových systémů, volba a vytvoření modelu, identifikace soustavy a nalezení parametrů modelu, vytvoření prediktoru a optimalizátoru a návrh přenosu poznatků na řízení systémů reálných.

Pátá kapitola se zabývá experimentální a metodickou částí, kde jsou popsány použité SW nástroje, algoritmy a funkce. Dále je popsán metodický postup řešení jednotlivých dílčích cílů.

Šestá kapitola se zabývá splněním jednotlivých cílů práce. Nejprve je provedena předběžná analýza a ověření použití prvků UI. Byla provedena analýza účelové funkce a vlivu omezení na dosažitelnou oblast, ověření možnosti nasazení evolučního algoritmu Hill Climbing a algoritmu evolučně-gradientního v rámci optimalizátoru a ověřením možnosti nasazení neuronové sítě jako modelu a její verifikace v rámci zvolené pracovní oblasti. Druhý dílčí cíl se zabývá odvozením tvarů diferenciálních rovnic modelových systémů a jejich úpravou pro účely simulace (saturace, singularity, aplikovatelnost). Dále je popsáno vytvoření RBF modelu systému. Čtvrtá část se zabývá určením parametrů modelu, tj. volbou periody, tvorbou trénovací množiny a trénováním neuronové sítě. Další část popisuje výběr a tvorbu prediktoru a možnosti nasazení. Šestá část popisuje tvorbu regulátoru a prezentuje výsledky experimentů řízení pro všechny modelové systémy a úpravu řízení pomocí korekčních křivek. Poslední část se zabývá návrhem přenosu získaných poznatků pro řízení systémů reálných a podmínkami, které je nutné splnit pro možnost nasazení těchto modelů a přístupů.

Sedmou kapitolu představuje věcný závěr provedené studie a souhrn přínosů práce pro vědu a praxi.

Hlavním cílem disertační práce byla aplikace některých prvků umělé inteligence ve vhodných oblastech prediktivního řízení. Cíle práce bylo dosaženo.

SEZNAM POUŽITÉ LITERATURY

- [1] RAWLINGS, James B. a David Q. MAYNE. *Model predictive control: theory and design*. Madison, Wis.: Nob Hill Pub., 2009. 533 s. ISBN 9780975937709.
- [2] KWON, W. H. a S. HAN. *Receding horizon control: model predictive control for state models*. Berlin ; London: Springer, 2005. 380 s. ISBN 9781846280245.
- [3] CAMACHO, E. F. a C. BORDONS. *Model predictive control*. London ; New York: Springer, 2003. 405 s. ISBN 1852336943.
- [4] WANG, Liuping. *Model predictive control system design and implementation using MATLAB*. London: Springer, 2009. 375 s. ISBN 9781848823303.
- [5] KHALED, Nassim a Bibin PATTEL. *Practical design and application of model predictive control: MPC for MATLAB® and Simulink® users*. Kidlington, Oxford: Butterworth-Heinemann, an imprint of Elsevier, 2018. ISBN 9780128139196.
- [6] KLUEVER, Craig A. *Dynamic systems: modeling, simulation, and control*. Hoboken, NJ: John Wiley and Sons, Inc., 2015. ISBN 9781118289457.
- [7] VÍTEČEK, Antonín a Miluše VÍTEČKOVÁ. *Optimální systémy řízení*. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, Strojní fakulta, 1999. 158 s. ISBN 8070787368.
- [8] ROSSITER, John. *Model-based Predictive Control: A Practical Approach*. CRC Press, 2003.
- [9] MIKLEŠ JÁN, Fikar M. *Modelovanie, identifikácia a riadenie procesov 2. Identifikácia a optimálne riadenie*. STU Press, Bratislava, 2004. 260 s. ISBN 80-227-2134-4.
- [10] BOBÁL, Vladimír a Univerzita Tomáše Bati ve Zlíně Fakulta aplikované informatiky. *Adaptivní a prediktivní řízení*. Vyd 1. Zlín: Univerzita Tomáše Bati ve Zlíně, 2008. 134 s. ISBN 978-80-7318-662-3.
- [11] BOBÁL, Vladimír a Univerzita Tomáše Bati ve Zlíně Fakulta aplikované informatiky. *Identifikace systémů*. Vyd 1. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009. 128 s. ISBN 978-80-7318-888-7.
- [12] DU, K. -. a M. N. S. SWAMY. *Neural networks in a softcomputing framework*. London: Springer, 2006. 566 s. ISBN 1846283027.

- [13] Kröse Ben, van der Smagt Patrick. *An introduction to Neural Networks*. Eight Edition. The University of Amsterdam, 1996.
- [14] Lee Gue Myung, Tam N.N., Yen Nguyen Dong. *Quadratic Programming and Affine Variational Inequalities: a Qualitative Study*. Springer US, 2005. ISBN 978-0-387-24278-1.
- [15] Dostál Zdeněk. *Optimal Quadratic Programming Algorithms: with Applications to Variational Inequalities*. Springer US, 2009. 284 s. ISBN 978-0-387-84806-8.
- [16] Luenberger David G., Ye Yinyu. *Linear and Nonlinear Programming*. Springer US, 2008. 546 s.
- [17] PLOSKAS, Nikolaos a Nikolaos SAMARAS. *Linear programming using MATLAB®*. Cham: Springer, 2017. ISBN 9783319659190.
- [18] HU, T. C. a Andrew B. KAHNG. *Linear and integer programming made easy*. Switzerland: Springer, 2016. ISBN 9783319240015.
- [19] Back Thomas, B. FOGEL DAVID a Michalewicz Zbigniew. *Handbook of Evolutionary Computation*. Oxford University Press, 1997. 988 s. ISBN 0750303927.
- [20] MIRJALILI, Seyedali. *Evolutionary algorithms and neural networks : theory and applications*. . ISBN 9783319930251; 1860-949X.
- [21] SIMON, Dan. *Evolutionary optimization algorithms: biologically-Inspired and population-based approaches to computer intelligence*. Hoboken, New Jersey: John Wiley & Sons Inc., 2013. ISBN 9780470937419.
- [22] KOCHENDERFER, Mykel J. *Algorithms for optimization*. . Tim A. WHEELER. . ISBN 9780262039420.
- [23] RAU, M. a D. SCHRODER. *Model predictive control with nonlinear state space models*. 2002. 136-141 s. ISBN 0-7803-7479-7.
- [24] STREJC, Vladimír. *Stavová teorie lineárního diskrétního řízení*. 1 vyd. Praha: Academia, 1978. 374 s.
- [25] CAMACHO, Eduardo a Carlos BORDONS. *Model Predictive Control*. Springer-Verlag, London, 2004. ISBN 978-0-85729-398-5.
- [26] TANGIRALA, Arun K. *Principles of system identification : theory and practice*. 1st. . ISBN 9781439896020.
- [27] LJUNG, Lennart. *System identification: theory for the user*. Upper Saddle River, N.J.: Prentice Hall, 1999. 609 s. ISBN 0136566952.

- [28] Anonymous. *Selecting a Model Structure in the System Identification Process*. 2018 Dostupné z: <http://www.ni.com/product-documentation/4028/en/>.
- [29] ZHENG, A. Reducing on-line computational demands in model predictive control by approximating QP constraints. *Journal of Process Control*. 1999, vol. 9, no. 4, s. 279-290. ISSN 0959-1524.
- [30] WAN, Z. Y. a M. V. KOTHARE. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*. 2003, vol. 39, no. 5, s. 837-846. ISSN 0005-1098.
- [31] SALTIK, M. Bahadir et al. An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*. 2018, vol. 61, s. 77-102. ISSN 0959-1524.
- [32] RAMIREZ, D. R., T. ALAMO a E. F. CAMACHO. Computational burden reduction in min-max MPC. *Journal of the Franklin Institute-Engineering and Applied Mathematics*. 2011, vol. 348, no. 9, s. 2430-2447. ISSN 0016-0032.
- [33] ORAVEC, J. a M. BAKOSOVA. Robust MPC Based on Nominal System Optimization and Weighted Control Input Saturation. *2015 54th Ieee Conference on Decision and Control (Cdc)*. 2015, s. 6239-6244. ISSN 0743-1546.
- [34] MUEHLENBEIN, Heinz. Evolutionary Computation: Centralized, Parallel or Collaborative. *Computational Intelligence: Collaboration, Fusion and Emergence*. 2009, vol. 1, s. 561-595. ISSN 1868-4394.
- [35] PARRILLA, M., J. ARANDA a S. DORMIDO-CANTO. Parallel evolutionary computation: Application of an EA to controller design. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Pt 2, Proceedings*. 2005, vol. 3562, s. 153-162. ISSN 0302-9743.
- [36] MURTHY, V. K. a E. V. KRISHNAMURTHY. *Parallel programming paradigm: Application to evolutionary computations*. , 1995. 299 s. ISBN 0-7803-2759-4.
- [37] TSUTSUI, Shigeyoshi a Noriyuki FUJIMOTO. *An Analytical Study of GPU Computation for Solving QAPs by Parallel Evolutionary Computation with Independent Run*. , 2010. ISBN 978-1-4244-8126-2.
- [38] SADRIEH, Arash a Parisa A. BAHRI. Application of Graphic Processing Unit in Model Predictive Control. *21st European Symposium on Computer Aided Process Engineering*. 2011, vol. 29, s. 492-496. ISSN 1570-7946.

- [39] PHUNG, Duc-Kien et al. Model Predictive Control for Autonomous Navigation Using Embedded Graphics Processing Unit. *Ifac Papersonline*. 2017, vol. 50, no. 1, s. 11883-11888. ISSN 2405-8963.
- [40] HYATT, Phillip a Marc D. KILLPACK. *Real-Time Evolutionary Model Predictive Control Using a Graphics Processing Unit.* , 2017. 576 s. ISBN 978-1-5386-4678-6.
- [41] MAZINAN, A. H. a A. R. KHALAJI. A comparative study on applications of artificial intelligence-based multiple models predictive control schemes to a class of industrial complicated systems. *Energy Systems-Optimization Modeling Simulation and Economic Aspects*. 2016, vol. 7, no. 2, s. 237-269. ISSN 1868-3967.
- [42] RUSSELL, Stuart J. a Peter NORVIG. *Artificial intelligence: a modern approach*. Upper Saddle River, N.J.: Prentice Hall, 2010. 1132 s. ISBN 9780132071482.
- [43] GRÜNE, Lars a Jürgen PANNEK. *Nonlinear model predictive control: theory and algorithms*. Second. Cham: Springer, 2016. ISBN 9783319460246.
- [44] LEE, Jay H. Model Predictive Control: Review of the Three Decades of Development. *International Journal of Control Automation and Systems*. 2011, vol. 9, no. 3, s. 415-424. ISSN 1598-6446.
- [45] ZIMMER, Andrea et al. Evolutionary algorithm enhancement for model predictive control and real-time decision support. *Environmental Modelling & Software*. 2015, vol. 69, s. 330-341. ISSN 1364-8152.
- [46] YAN, Jingyu, Qing LING a Wei CHEN. *Nonlinear Model Predictive Control Based on Evolutionary Algorithms: Framework, Theory, and Application.* . AJ Costa GIORDANO GE. , 2009. 205 s. ISBN 978-1-60456-883-7.
- [47] CANNON, M. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*. 2004, vol. 28, no. 2, s. 229-237. ISSN 1367-5788.
- [48] NANDOLA, Naresh N. a Sharad BHARTIYA. A multiple model approach for predictive control of nonlinear hybrid systems. *Journal of Process Control*. 2008, vol. 18, no. 2, s. 131-148. ISSN 0959-1524.
- [49] BINDLISH, R. a J. B. RAWLINGS. Target linearization and model predictive control of polymerization processes. *AIChE Journal*. 2003, vol. 49, no. 11, s. 2885-2899. ISSN 0001-1541.

- [50] LEE, J. H. a N. L. RICKER. Extended Kalman Filter Based Nonlinear Model-Predictive Control. *Industrial & Engineering Chemistry Research*. 1994, vol. 33, no. 6, s. 1530-1541. ISSN 0888-5885.
- [51] WANG, Xin, Longxiang GUO a Yunyi JIA. Road Condition Based Adaptive Model Predictive Control for Autonomous Vehicles. *Proceedings of the Asme 11th Annual Dynamic Systems and Control Conference, 2018, Vol 3*. 2018, s. V003T37A005. ISSN 2151-1853.
- [52] TANASKOVIC, Marko, Lorenzo FAGIANO a Vojislav GLIGOROVSKI. *Adaptive Model Predictive Control for Constrained Time Varying Systems*. , 2018. 1703 s. ISBN 978-3-9524-2698-2.
- [53] TESHAY, Mehari et al. Adaptive-model predictive control of electronic expansion valves with adjustable setpoint for evaporator superheat minimization. *Building and Environment*. 2018, vol. 133, s. 151-160. ISSN 0360-1323.
- [54] GRIFFITH, Devin W., Lorenz T. BIEGLER a Sachin C. PATWARDHAN. Robustly stable adaptive horizon nonlinear model predictive control. *Journal of Process Control*. 2018, vol. 70, s. 109-122. ISSN 0959-1524.
- [55] SUÁREZ, G. I. et al. *Adaptive neural model predictive control for the grape juice concentration process*. 2010 [cit. 12 September 2018]. 57-62 s.
- [56] KUMAR, Steven Spielberg Pon et al. A Deep Learning Architecture for Predictive Control. *Ifac Papersonline*. 2018, vol. 51, no. 18, s. 512-517. ISSN 2405-8963.
- [57] PICHE, S. et al. Neural network based Model Predictive Control. *Advances in Neural Information Processing Systems 12*. 2000, vol. 12, s. 1029-1035. ISSN 1049-5258.
- [58] WONG, Wee Chin et al. Recurrent Neural Network-Based Model Predictive Control for Continuous Pharmaceutical Manufacturing. *Mathematics*. 2018, vol. 6, no. 11, s. 242. ISSN 2227-7390.
- [59] DEEPA, S. N. a I. BARANILINGESAN. Optimized deep learning neural network predictive controller for continuous stirred tank reactor. *Computers & Electrical Engineering*. 2018, vol. 71, s. 782-797. ISSN 0045-7906.
- [60] HOU, Guolian et al. *Multi-model Predictive Control based on Neural Network and Its Application in Power Plant*. NEW YORK; 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2014. 4383 s. ISBN 978-1-4799-5825-2.

- [61] HAN, H. -G, X. -L WU a J. -F QIAO. Real-time model predictive control using a self-organizing neural network. *IEEE Transactions on Neural Networks and Learning Systems*. 2013, vol. 24, no. 9, s. 1425-1436 [cit. 12 September 2018].
- [62] KUBALCÍK, Marek a Vladimír BOBÁL. *Predictive control of three-tank-system utilizing both state-space and input-output models*. Regensburg, Germany: European Council for Modelling and Simulation (ECMS), 2016. 348-353 s. ISBN 9780-993244025.
- [63] SATHISHKUMAR, K., V. KIRUBAKARAN a T. K. RADHAKRISHNAN. Real Time Modeling and Control of Three Tank Hybrid System. *Chemical Product and Process Modeling*. 2018, vol. 13, no. 1, s. 20170016. ISSN 1934-2659.
- [64] MAXIM, Anca, Clara IONESCU a Robin DE KEYSER. Modelling and identification of a coupled sextuple water tank system. *Proceeding of 2016 Ieee International Conference on Automation, Quality and Testing, Robotics (Aqtr)*. 2016, s. 393-398. ISSN 1844-7872.
- [65] COPOT, Dana et al. Multivariable control of sextuple tank system with non-minimum phase dynamics. *Proceeding of 2016 Ieee International Conference on Automation, Quality and Testing, Robotics (Aqtr)*. 2016, s. 399-404. ISSN 1844-7872.
- [66] ZHOU, Feng et al. RBF-ARX model-based MPC strategies with application to a water tank system. *Journal of Process Control*. 2015, vol. 34, s. 97-116. ISSN 0959-1524.
- [67] XU, Jing et al. Recurrent neural network for solving model predictive control problem in application of four-tank benchmark. *Neurocomputing*. 2016, vol. 190, s. 172-178. ISSN 0925-2312.
- [68] MANOHAR, Gunaselvi et al. *Neural Network Based Level Control in Two Tank Conical Interacting System*. . Manoharan, S Arunkumar, E Ananthi, K Ganesh,RM. , 2013. 196 s. ISBN 978-1-4673-4601-6.
- [69] KALA, H., P. ARAVIND a M. VALLUVAN. *Comparative Analysis of Different Controller for a Nonlinear Level Control Process*. , 2013. 729 s. ISBN 978-1-4673-5758-6.
- [70] VENKATESAN, N. a N. ANANTHARAMAN. *Controller design based on model predictive control for a nonlinear process*. , 2012 [cit. 6 June 2019]. Cited By :2.
- [71] NITHYA, S. et al. Model based controller design for a spherical tank process in real time. *International Journal of Simulation: Systems, Science and Technology*. 2008, vol. 9, no. 4, s. 25-31 [cit. 6 June 2019].

- [72] MERCY, D. a S. M. GIRIRAJKUMAR. Modeling and analysis of a real time spherical tank process for sewage treatment plant. *Applied Mathematics and Information Sciences*. 2017, vol. 11, no. 5, s. 1491-1498 [cit. 10 June 2019].
- [73] KLAUCO, Martin, L'ubos CIRKA a Juraj KUKLA. Non-linear model predictive control of conically shaped liquid storage tanks. *Acta Chimica Slovaca*. 2018, vol. 11, no. 2, s. 141-146. ISSN 1337-978X.
- [74] MÍKA, Vladimír. *Základy chemického inženýrství*. 2 nezm vyd. Praha: Státní nakladatelství technické literatury, 1981. 870 s.
- [75] KUNEŠ, Josef a Josef KUNEŠ. *Dimensionless physical quantities in science and engineering*. London ; Waltham, MA: Elsevier, 2012. 441 s. ISBN 9780124160132; 0124160131.
- [76] JERRARD, H. G. a D. B. MCNEILL. *A dictionary of scientific units, including dimensionless numbers and scales*. 2d. London: Chapman & Hall, 1964. 204 s.
- [77] HAHN, Brian D. a Daniel T. VALENTINE. *Essential MATLAB for engineers and scientists*. 5th. Waltham, Mass. ; Oxford: Academic Press, 2013. 408 s. ISBN 9780123943989; 0123943981.
- [78] LINDFIELD, G. R. a J. E. T. PENNY. *Numerical methods*. Academic Press, 2012. [electronic resource]: using MATLAB / George Lindfield, John Penny.; 1 online resource; MATLAB. ISBN 9780-123869425.
- [79] DUUN-HENRIKSEN, A. K. et al. Model identification using stochastic differential equation grey-box models in diabetes. *Journal of Diabetes Science and Technology*. 2013, vol. 7, no. 2, s. 431-440 [cit. 6 June 2019].
- [80] CORRIOU, Jean-Pierre. *Process Control: Theory and Applications*. 2018. Cham: Springer International Publishing : Imprint : Springer, 2018. ISBN 9783319611433; 3319611437; 9783319611426; 3319611429.
- [81] ŠULC BOHUMIL, Vítečková M. *Teorie a praxe návrhu regulačních obvodů*. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-03007-5.
- [82] Balátě Jaroslav. *Automatické řízení*. Praha: BEN, 2003. ISBN 80-7300-020-2.
- [83] GENG, G. a G. M. GEARY. Experimental comparisons between generalised predictive control algorithms using CARMA and CARIMA models. *Proceedings of the 1996 Ieee Iecon - 22nd International Conference on Industrial Electronics, Control, and Instrumentation, Vols 1-3*. 1996, s. 108-113. ISSN 1553-572X.
- [84] HUANG, Sunan, Kok K. TAN a Tong H. LEE. *Applied predictive control*. London ; New York: Springer, 2002. 264 s. ISBN 1852333383.

- [85] CLARKE, D. W., C. MOHTADI a P. S. TUFFS. Generalized predictive control- Part I. The basic algorithm. *Automatica*. 1987, vol. 23, no. 2, s. 137-148 [cit. 11 September 2018].
- [86] CLARKE, D. W., C. MOHTADI a P. S. TUFFS. Generalized Predictive Control- Part II Extensions and interpretations. *Automatica*. 1987, vol. 23, no. 2, s. 149-160 [cit. 11 September 2018].
- [87] BERTSEKAS, Dimitri P. *Nonlinear programming*. Third. Belmont, Mass.: Athena Scientific, 2016. 861 s. ISBN 9781886529052.
- [88] Nelles Oliver. *Nonlinear System Identification: from Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verla Berlin Heidelberg, 2001. 786 s.
- [89] Hudson Beale, MarkHagan, Martin. T.Demuth, Howard B. *Neural network toolbox™ getting started guide*. MathWorks, 2016.
- [90] *Neural networks for modelling and control of dynamic systems : a practitioner's handbook*. . Magnus NØRGAARD. London: Springer, 2000. ISBN 1852332271
- [91] ZELINKA, Ivan a SpringerLink. *Evolutionary algorithms and chaotic systems*. Berlin ; Heidelberg: Springer-Verlag, 2010. 521 s. [electronic resource] / Ivan Zelinka ... [et al.] (Eds.); 1 online resource; Includes bibliographical references.; Electronic reproduction. New York : Springer, 2010. v. 267. ISBN 9783-642107078.
- [92] HOLLAND, John H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press, 1975. 183 s. ISBN 0472084607.
- [93] RIBEIRO, J. L., P. C. TRELEAVEN a C. ALIPPI. Genetic-Algorithm Programming Environments. *Computer*. 1994, vol. 27, no. 6, s. 28-29. ISSN 0018-9162.
- [94] ANTOŠ JAN, Kubalčík Marek. Analysis of some aspects of optimization problem in predictive control. *International Journal of Mathematical Models and Methods in Applied Sciences*. 2014, vol. 8, s. 8. ISSN 1998-0140.
- [95] ANTOS, J. a M. KUBALCIK. Combination of evolutionary and gradient optimization techniques in model predictive control. *International Journal of Mathematical Models and Methods in Applied Sciences*. 2016, vol. 10, s. 34 [cit. 25 September 2018].
- [96] TAN, G., H. HAO a Y. WANG. *Real time turning flow estimation based on model predictive control*. , 2011 [cit. 12 September 2018]. 356-360 s.

SEZNAM OBRÁZKŮ A TABULEK

Obr. 3.1: Obecný lineární model, podle ref. [26, 28]	10
Obr. 3.2: Schéma neuronu, volně podle ref.[13, 88]	12
Obr. 3.3: Evoluční vývoj, nakresleno volně podle ref.[91-93]	14
Obr. 6.1: Účelová funkce (vlevo), b) dosažitelná oblast (vpravo), vlastní zdroj[94]. ..	19
Obr. 6.2: Algoritmus hill climbing (vlevo), algoritmus evolučně-gradientní (vpravo), vlastní zdroj[95].....	19
Obr. 6.3: Trénování (vlevo) a testování (vpravo) RBF sítě, vlastní zdroj.	21
Obr. 6.4: Modelový systém: válec, vlastní zdroj.....	22
Obr. 6.5: Modelové systémy – ověření správnosti, vlastní zdroj.	24
Obr. 6.6: Modelové systémy – nesaturované, vlastní zdroj.	25
Obr. 6.7: Modelové systémy - saturované, vlastní zdroj.....	26
Obr. 6.8: Část trénovací množiny pro RBF model systému (zobrazeno bez vstupu), vlastní zdroj.	28
Obr. 6.9: Simulace prediktivního řízení modelových systémů, vlastní zdroj.	29
Obr. 6.10: Pilotní experiment – dynamika systému v různých úrovních, vlastní zdroj.	29
Obr. 6.11: Řízení modelového systému s úzkým hrdlem v oblasti singularity, vlastní zdroj.	30
Obr. 6.12: Časový průběh řízení výšky hladiny pro modelové systémy s upravenou žádanou hodnotou, vlastní zdroj.	31
Tab. 1: Porovnání algoritmů optimalizátoru pro řízení s CARIMA modelem, vlastní zdroj[95]	20
Tab. 2: Jmenovatel diferenciální rovnice modelového systému, vlastní zdroj	23

SEZNAM POUŽITÝCH SYMBOLŮ

$G(z^{-1})$	přenos (z-oblast)
$U(z^{-1}), Y(z^{-1})$	obrazy funkcí vstupu a výstupu
A, B, C, D, F	polynomy lineárních modelů
Δ	přírůstek
n	bílý šum
u	vstup/akční zásah
y	výstup
θ	parametry modelu
f	matematická funkce
\bar{u}, \bar{y}	minulé hodnoty vstupu a výstupu
$\vec{u}, \vec{y}, \vec{w}$	budoucí hodnoty vstupu, výstupu a žádané hodnoty
\tilde{u}, \hat{y}	vstup a výstup prediktoru
$\langle N_1, N_2 \rangle$	spodní a horní mez výstupního horizontu
N_u	řídící horizont
J, λ	účelová funkce, váhový parametr
x, w, a, b	vstup, váha a aktivační funkce a práh neuronové sítě
x_i, c_i, Σ_i	střední vzdálenost středů od vstupů, střed a norma neuronu
Q, h, S, V_1, V_2	objemový průtok, výška hladiny, průřez nádrže, výstupní ventily
D, H, k_D, t_{max}	průměr, výška, přepočtový faktor a doba napuštění/vypuštění nádrže (konstanty)
V_h, q_t, h, d, k_d	objem, změna objemu, výška nádrže, průměr nádrže a přepočtový faktor (proměnné)
q_{max}, V_{max}, k_q, t	max. průtok, max. objem, přepočtový faktor (časový), čas

$q_{in}, q_{out}, a, d_{ms}$ přítok, odtok, akumulace, jmenovatel modelového systému

q_b, q_{of} funkce dna (bottom), přepadu (overflow)

SEZNAM ZKRATEK

ARMAX	AutoRegressive-Moving-Average model with eXogenous input
CARIMA	Controlled AutoRegressive Integrated Moving Average
EA	Evoluční Algoritmus
EG	Evolučně-Gradientní algoritmus
HC	Hill Climbing algoritmus
HW	HardWare
MLP	Multi-Layer Perceptron
MPC	Model Predictive Control
NN	Neural Network
PID	Proporcionálně-Integračně-Derivační regulátor
QP	QuadProg
RBF	Radial Basis Function
SOMA	Self-Organizing Migrating Algorithm
SSE	Sum of Squared Errors
SW	SoftWare
UI	Umělá Inteligence

PUBLIKAČNÍ AKTIVITY AUTORA

D₂ – Článek ve sborníku mezinárodní konference neevidované v databázi ISI Proceedings společnosti Thomson Reuters – světový jazyk

ANTOŠ JAN, Kubalčík M. *Optimization in Predictive Control Algorithm. in: Latest Trends in Circuits, Systems, Signal Processing and Automatic Control.* Salerno, Italy: WSEAS Press, 2014, 6 s. ISSN 1790-5117. ISBN 978-960-474-374-2. Dostupné z: <http://www.wseas.us/e-library/conferences/2014/Salerno/CISSPA/CISSPA-19.pdf>

D_{nehodn1} – Články ve sborníku mezinárodní konference neevidované v databázi WoS nebo SCOPUS

ANTOŠ JAN; Kubalčík M. *Optimization in Model Predictive Control Using Evolutionary-Gradient Algorithm. in: The 3rd International Conference on Applied, Numerical and Computational Mathematics.* Sliema, Malta: WSEAS, 2015, 35-41 s. ISSN 2227-4588. ISBN 978-1-61804-328-3. BC - Teorie a systémy řízení

D_{sjr} – Články ve sborníku konference evidované v databázi SCOPUS

ANTOS, J. a M. KUBALCIK. *Alternative approach to optimization in model predictive control using hill climbing algorithm* [online]. , 2015 [cit. 25 September 2018]. 856-864 s.

J_{sc} - Články v recenzovaných časopisech indexovaných v databázi SCOPUS

ANTOŠ JAN, Kubalčík Marek. Analysis of some aspects of optimization problem in predictive control. *International Journal of Mathematical Models and Methods in Applied Sciences* [online]. 2014, vol. 8, s. 8. ISSN 1998-0140. Dostupné z: <http://www.naun.org/main/NAUN/ijmmas/2014/a422001-049.pdf>.

ANTOS, J. a M. KUBALCIK. Combination of evolutionary and gradient optimization techniques in model predictive control. *International Journal of Mathematical Models and Methods in Applied Sciences* [online]. 2016, vol. 10, s. 34 [cit. 25 September 2018]

Ostatní publikační aktivity

SKODA, David; URBANEK, Pavel; SEVCIK, Jakub; MUNSTER, Lukas; NADAZDY, Vojtech; CULLEN, David A; BAZANT, Pavel; ANTOS, Jan; KURITKA, Ivo. Colloidal Cobalt-doped ZnO Nanoparticles by Microwave-assisted Synthesis and Their Utilization in Thin Composite Layers with MEH-PPV as an Electroluminescent Material for Polymer Light Emitting Diodes. *Organic Electronics*. 2018, vol. 59, no. C s. 337-348. ISSN:1566-1199.

SKODA, David, Pavel URBANEK, Jakub SEVCIK, et al. Colloidal cobalt-doped ZnO nanoparticles by microwave-assisted synthesis and their utilization in thin composite layers with MEH-PPV as an electroluminescent material for polymer light emitting diodes. *Organic Electronics*. 2018, 59, 337-348. DOI: 10.1016/j.orgel.2018.05.037. ISSN 15661199.

DATTA, Sanjoy; ANTOS, Jan; STOCEK, Radek. Characterisation of Ground Tyre Rubber by Using Combination of FT-IR Numerical Parameter and DTG Analysis to Determine the Composition of Ternary Rubber Blend. *Polymer Testing*. 2017, vol. 59 s. 308-315. ISSN:0142-9418.

DATTA, Sanjoy; ANTOŠ, Jan; STOČEK, Radek. Smart Numerical Method for Calculation of Simple General Infrared Parameter Identifying Binary Rubber Blends. *Polymer Testing*. 2017, vol. 57 s. 192-202. ISSN:0142-9418.

DATTA, Sanjoy, Jan ANTOŠ a Radek STOČEK. A novel algorithm. LION, Alexander a Michael JOHLITZ, ed. *Constitutive Models for Rubber X*. CRC Press, 2017, 2017-8-15, s. 213-217. DOI: 10.1201/9781315223278-31. ISBN 9781315223278.

MASAŘ, Milan; MACHOVSKÝ, Michal; URBÁNEK, Michal; URBÁNEK, Pavel; ANTOŠ, Jan; KUŘITKA, Ivo. On-line Measurement of Photocatalytic Activity of Powdered Samples. 8th International Conference on Nanomaterials - Research & Application (NANOCON 2016). 2016, 275-279 s.

ŠEVČÍK, Jakub; URBÁNEK, Pavel; ŠULY, Pavol; URBÁNEK, Michal; MAŠLÍK, Jan; ANTOŠ, Jan; KUŘITKA, Ivo. Preparation and Characterization of Nanostructured Thin Films Applicable in Polymer Light Emitting Devices. 8th International Conference on Nanomaterials - Research & Application (NANOCON 2016). 2016, 817-821 s.

ODBORNÝ ŽIVOTOPIS

Osobní informace

Jméno, příjmení **Jan Antoš**
Adresa Rymice 112, 769 01 Holešov
Telefon +420 603 915 687
E-mail antos@utb.cz
Státní občanství Česká republika
Datum narození 05. 12. 1988
Pohlaví Muž

Vzdělání, odborná příprava

Datum 2013 až současnost
Aktuální stav doktorský student
Organizace Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Nad Stráněmi 4511, 760 05 Zlín

Datum 2011 – 2013
Dosažený stupeň Inženýr / Ing.
Organizace Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Nad Stráněmi 4511, 760 05 Zlín

Datum 2008 – 2011
Dosažený stupeň Bakalář / Bc.
Organizace Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Nad Stráněmi 4511, 760 05 Zlín

Datum 2004 – 2008
Organizace Střední škola informatiky, elektrotechniky a řemesel, Školní 1610, 756 61 Rožnov pod Radhoštěm

Osobní dovednosti

Mateřský jazyk Čeština
Další jazyky Angličtina – C1
Počítačové dovednosti Matlab, Mathematica, Labview, C#, MS Office

Ve Zlíně dne 02. 06. 2019

Ing. Jan Antoš

Ing. Jan Antoš, Ph.D.

Prediktivní řízení procesů s využitím prvků umělé inteligence

Predictive Control of Processes with Utilization of Artificial Intelligence
Elements

Teze disertační práce

Vydala Univerzita Tomáše Bati ve Zlíně
nám. T. G. Masaryka 5555, 760 01 Zlín.

Náklad: vyšlo elektronicky

Sazba: Jan Antoš

Publikace neprošla jazykovou ani redakční úpravou.

Rok vydání 2019

ISBN 978-80-7454-860-4

