

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

Odbor mechaniky a mechatroniky



DIPLOMOVÁ PRÁCE

**Řízení a autotuning synchronního AC servomotoru Baldor
pomocí měniče Beckhoff AX5206**

Praha, 2019

Jaroslav Běťák

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Běťák** Jméno: **Jaroslav** Osobní číslo: **438958**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Strojní inženýrství**
Studijní obor: **Mechatronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení a autotuning synchronního AC servomotoru Baldor pomocí měniče Beckhoff AX5206.

Název diplomové práce anglicky:

Control and autotuning of the Baldor synchronous AC servo motor with Beckhoff AX5206 inverter.

Pokyny pro vypracování:

1. Proveďte rešerši na téma řízení synchronních elektrických pohonů se zaměřením na formy jejich autotuningu
2. Sestavte simulační model motoru včetně optimalizačního algoritmu autotuningingu regulační smyčky
3. Proveďte identifikaci parametrů motoru
4. Prakticky realizujte řízení daného motoru a aplikujte autotuning
5. Kriticky zhodnoťte dosažené výsledky

Seznam doporučené literatury:

- [1] R. Doncker, D.W.J. Pülle, A. Veltman: Advanced Electrical Drives, Springer Netherlands, 2011
- [2] G. Ellis: Control System Design Guide, Elsevier, 2012, 4th Edition
- [3] Valášek, M. a kol. Mechatronika, Praha : ČVUT 1995

Jméno a pracoviště vedoucí(ho) diplomové práce:

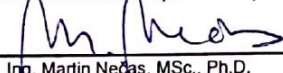
Ing. Martin Nečas, MSc., Ph.D., odbor mechaniky a mechatroniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **29.04.2019**

Termín odevzdání diplomové práce: **16.08.2019**

Platnost zadání diplomové práce:


Ing. Martin Nečas, MSc., Ph.D.
podpis vedoucí(ho) práce


prof. Ing. Milan Růžička, CSc.
podpis vedoucí(ho) ústavu/katedry


prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

7.5. 2019

Datum převzetí zadání



Podpis studenta

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pouze za odborného vedení vedoucího práce. Veškeré zdroje informací, ze kterých jsem čerpal, jsou uvedeny v seznamu použité literatury.

V Praze, dne

Podpis:

Poděkování

Tímto bych rád poděkoval vedoucímu práce, panu Ing. Martin Nečas, MSc., Ph.D. za odborné vedení práce, ochotu při poskytování rad i připomínek a vstřícnost během konzultací. Také bych chtěl poděkovat mé rodině za trpělivost a ochotu, s kterou mě podporovali během mého studia.

Anotační list

Jméno autora:	Jaroslav Běžák
Název diplomové práce:	Řízení a autotuning synchronního AC servomotoru Baldor pomocí měniče Beckhoff AX5206
Anglický název:	Control and autotuning of the Baldor synchronous AC servomotor with Beckhoff AX5206 inverter
Akademický rok:	2018/2019
Obor studia:	Mechatronika
Ústav/odbor:	Ústav mechaniky, biomechaniky a mechatroniky Odbor mechaniky a mechatroniky
Vedoucí diplomové práce:	Ing. Martin Nečas, MSc., Ph.D.
Bibliografické údaje:	počet stran: 71 počet obrázků: 57 počet příloh: 1 CD
Klíčová slova:	Synchronní AC servomotor, řízení, autotuning, optimalizace, Matlab, Simulink
Keywords:	Synchronous AC servomotor, control, autotuning, optimization, Matlab, Simulink
Anotace:	Cílem práce je detailnější seznámení se s možnostmi realizace řízení synchronních elektrických pohonů, aplikace některé z forem autotuningu a následné provedení experimentu s reálným motorem.
Abstract:	The aim of this work is to analyze possibilities of realization control of synchronous electric drives, application form of autotuning and realization of experiment with real motor.

Obsah

Prohlášení.....	3
Poděkování.....	4
Anotační list	5
Obsah	6
Úvod.....	8
Cíle práce	8
1 Řízení synchronních servomotorů.....	9
1.1 Synchronní servomotory	9
1.2 Kaskádní regulace	10
1.3 PID regulátor	11
1.3.1 P regulátor	11
1.3.2 PI regulátor.....	12
1.4 Naladění regulátoru	13
1.4.1 Metoda pokus-omyl	13
1.4.2 Metoda Ziegler-Nichols	14
1.4.3 Tříparametrový model.....	15
1.4.4 Ladění parametrů pomocí relé	17
1.4.5 Gradientní metoda.....	19
1.4.6 Metoda simulovaného žihání	21
2 Simulační model	23
2.1 Struktura řízení v měniči	23
2.2 Simulační model motoru	26
2.3 Simulační model řízení.....	28
2.4 Optimalizace pomocí gradientní metody	31
2.4.1 Algoritmus gradientní metody	31
2.4.2 Validace gradientního algoritmu.....	34

2.4.3	Optimalizace jednoho parametru	36
2.4.4	Optimalizace dvou parametrů	39
2.4.5	Optimalizace více parametrů.....	41
2.5	Optimalizace pomocí simulovaného žihání	43
2.5.1	Algoritmus simulovaného žihání	43
2.5.2	Optimalizace jednoho parametru	45
2.5.3	Optimalizace dvou parametrů	47
2.5.4	Optimalizace více parametrů.....	49
3	Identifikace parametrů motoru	51
3.1	Použitý motor	51
3.2	Odpor statorového vinutí.....	52
3.3	Indukčnost statorového vinutí	54
3.4	Identifikace mechanických veličin motoru	54
4	Příprava experimentu	55
4.1	Propojení motoru a frekvenčního měniče.....	55
4.2	Implementační verze modelu	56
4.3	Nahrání modelu do softwaru TwinCAT.....	57
4.4	První testovací experiment	58
5	Experimentální optimalizace dvou parametrů	60
5.1	Experiment s gradientní metodou	60
5.2	Experiment se simulovaným žiháním.....	62
5.2.1	První experiment (C. r. = 0.75)	62
5.2.2	Druhý experiment (C. r. = 0.9).....	63
5.3	Vyhodnocení experimentů	65
6	Závěr	66
7	Literatura.....	67
8	Seznam obrázků	70

Úvod

S rozvíjejícími se nároky na efektivitu průmyslových procesů a stoupající potřebou se dynamicky přizpůsobit neustále se měnícím podmínkám na trhu, roste snaha o maximální možnou autonomnost výrobních zařízení. Jednou z forem, jak dosáhnout zvýšení efektivity, je optimalizace na základě dat získaných přímo z výrobních procesů.

Pro tuto formu řízení už přestávají dostačovat běžně se používající regulační zásahy určené člověkem, často stylem pokus-omyl, a čím dál více se používají efektivnější formy jako např. autotuning.

Cíle práce

Předmětem této práce je detailnější seznámení se s možnostmi realizace řízení synchronních motorů s permanentními magnety, aplikace některé z forem autotuningu a následné provedení experimentu s reálným motorem. Práce postupuje podle následujících bodů:

1. Provést rešerši na téma řízení synchronních elektrických pohonů se zaměřením na formy jejich autotuningu.
2. Sestavit simulační model motoru včetně optimalizačního algoritmu autotuningu regulační smyčky.
3. Provést identifikaci parametrů motoru.
4. Prakticky realizovat řízení daného motoru a aplikovat autotuning.
5. Kriticky zhodnotit dosažené výsledky.

1 Řízení synchronních servomotorů

Servomotory jsou elektrické regulační pohony s možností vysoce přesného řízení polohy a rychlosti. Skládají se z hnacího elektromotoru s napájecím výkonovým měničem a řídicích regulačních obvodů. Zatímco například krokové motory jsou zapojené v takzvané otevřené regulační smyčce, tj. bez zpětné vazby, tak servomotory musí být vždy zapojené v uzavřené regulační smyčce, tedy se zpětnou vazbou.

Při výběru servomotoru pro určitou aplikaci má největší význam právě přesnost a rychlost regulační smyčky. Dle požadovaných regulačních a výkonových parametrů může být vybráno z různých konstrukčních provedení servomotoru. [1], [2]

1.1 Synchronní servomotory

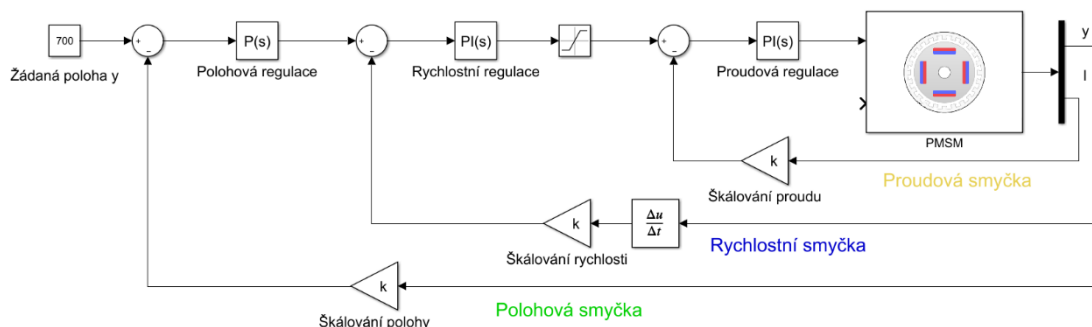
Synchronní motory jsou elektrické stroje, jejichž rotor se otáčí synchronně s unášivým magnetickým polem statoru. V současné době patří mezi nejpoužívanější typ pohonu pro polohové mechanismy a robotické struktury. Svou oblibu si získaly zejména díky dobrému poměru mezi výkonem a hmotností, vysoké účinnosti spojené s teplotní stabilitou a v neposlední řadě díky rovnoměrnosti momentového chodu.

Dle konstrukce jsou střídavé synchronní servomotory dvou či vícepólové. Jejich stator se skládá z plechů a v drážkách mezi plechy je uloženo statorové vinutí. Toto vinutí je, stejně jako u asynchronních motorů, třífázové. Vinutí statoru je napájeno proměnným napětím z tranzistorového měniče s pulzní šířkovou modulací. Výkonovou část napájí třífázové harmonické sinusové napětí. Rotor synchronních servopohonů je zpravidla buzen permanentními magnet. Ty mohou být buďto uloženy na povrchu nebo zapuštěné dovnitř rotoru.

Pro účely řízení tranzistorových měničů musí mít motor neustálou zpětnou vazbu na natočení rotoru. Nejpoužívanější snímače polohy jsou resolver a optický enkodér. Dalším prvkem motoru je například termistor na statoru motoru, jež slouží jako ochrana proti tepelnému přetížení vinutí. Dále se motory dovybavují také nouzovou brzdou pro případ selhání regulátoru nebo nenadálého výpadku proudu. [2], [3]

1.2 Kaskádní regulace

Jak bylo zmíněno v předchozích odstavcích, pro správné fungování servopohonů je klíčová uzavřená řídicí smyčka. V této smyčce je zajištěna zpětná vazba, která slouží k regulaci odchylek řízených veličin. Nejčastější formou zpětnovazebného řízení je tzv. kaskádní regulace, která využívá seřazení více regulátorů do kaskády, což vede ke zlepšení celkové regulovatelnosti systému.



Obrázek 1 - Schéma kaskádní regulace

Smyčky v jednotlivých vrstvách vrací dosaženou hodnotu veličin a následně ji odčítají od požadovaných hodnot, čímž je vyjádřena chybová hodnota veličiny, kterou je třeba minimalizovat. Zároveň lze do smyček zahrnout různé formy regulátorů, filtrů nebo případně získané hodnoty dle potřeby přeškálovat. Nejrychlejší smyčka bývá vnitřní a poté se postupuje až k vnější smyčce s nejpomalejší zpětnou vazbou.

V případě synchronního motoru je potřeba řídit požadovaný proud, jemuž bude následně odpovídat i moment motoru. Dále nás bude zajímat rychlost otáčení motoru a samozřejmě dosahování požadované polohy. Při plně polohovém řízení bude vstupem pro požadovaný proud výstup z regulátoru rychlosti a vstupem pro požadovanou rychlost bude výstup z regulátoru polohy. Když není potřeba řídit konečnou polohu lze vnější smyčky vynechat a řídit motor pouze na rychlostní, případně proudové úrovni. [1]

1.3 PID regulátor

PID regulátor je lineární, spojitý regulátor skládající se z proporcionální, integrační a derivační složky. Tento typ regulátoru je v současné době, i přes takřka stoleté stáří, stále nejpoužívanějším typem zpětnovazebného regulátoru. Oblibu mu zajistil poměrně jednoduchý princip fungování, který je ovšem schopen pokrýt požadavky na řízení většiny průmyslových procesů. [4] [7]

Rovnice PID regulátoru tuto podobu:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (1)$$

Kde: $u(t)$... akční zásah regulátoru

$e(t)$... velikost odchylky systému

K_p, T_i, T_d ... zesílení a časové konstanty regulátoru

Regulátor nemusí být vždy použit ve své plné formě, jednotlivé složky lze vypnout. Toho se v praxi často využívá a regulátor se ponechává ve formě PI, případně pouze P regulátoru. Vypnutá derivační složka regulátoru je sice na základě derivací aktuálních hodnot schopna predikovat hodnoty budoucí a používá se ke zrychlení regulačního pochodu, ale zároveň prudce reaguje na jakoukoliv odchylku a u signálů obsahujících šum může způsobovat problémy. [5]

1.3.1 P regulátor

Proporcionální regulátor je nejjednodušší formou regulátoru. Pracuje v podstatě jako zesilovač. Vezme velikost odchylky systému, vynásobí ji požadovaným zesílením a vloží na výstup.

Rovnice P regulátoru:

$$u(t) = K_p \cdot e(t) \quad (2)$$

Problémem tohoto typu regulátoru je zachovávání trvalé regulační odchylky. To je důsledkem toho, že při přibližování k cílové hodnotě se nám odchylka neustále zmenšuje, až je nakonec tak malá, že ani po vynásobení zesílením regulátoru nezpůsobí citelnou změnu. [5]

Zmíněné problémy a primitivnost regulátoru ovšem nezabraňují jeho hojnému využívání. Záleží vždy na posouzení, zda je zmiňovaná trvalá odchylka v toleranci či nikoliv. Při nastavování zesílení je potřeba mít na paměti, že větší hodnoty zesílení zvětšují citlivost regulátoru, ale zároveň můžou vést k případné nestabilitě řízení.



Obrázek 2 - Odezvy systému na různá zesílení ^[6]

1.3.2 PI regulátor

Nejčastější formou je proporcionalní regulátor rozšířený o integrační složku. Tato složka integruje regulační odchylku a vede k tomu, že i malá trvalá odchylka se nakonec projeví a bude regulována. Díky tomu je výstupem proces s nulovou trvalou regulační odchylkou.

Rovnice PI regulátoru:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right) \quad (3)$$

Přidáním integrátoru bude dosaženo nulové odchylky, zároveň ale vzroste nestabilita systému a dojde k fázovému zpoždění. Následné zvětšování integrační konstanty, respektive snižování T_i , může vést k oscilačnímu chování soustavy. Oscilace a fázové zpoždění lze eliminovat přidáním derivační složky, tím opět bude dosaženo plného tvaru PID regulátoru, viz rovnice (1). [5]

Vztah mezi tzv. integrační a integrační časovou konstantou regulátoru:

$$K_i = \frac{K_p}{T_i} \quad (4)$$

1.4 Naladění regulátoru

Tato kapitola se zabývá možnými postupy ladění parametrů regulátorů. Kvalitní naladění je otázkou správné identifikace systému, nastavení periody vzorkování a všeobecného povědomí o fungování jednotlivých regulátorů. Řada postupů vychází čistě z empirického poznání, další se snaží určit parametry na základě pozorování systému při jeho vybuzení a modernější algoritmy využívají iterační metody.

1.4.1 Metoda pokus-omyl

Metoda pokus-omyl je manuální postup založený na empirických poznatcích a subjektivním hodnocení operátora, jenž ladí regulátor. Při tomto postupu se testuje odezva uzavřené smyčky systému na skok vstupní veličiny a následně dochází k postupnému nastavování jednotlivých parametrů, dokud není docíleno požadovaných výstupů.

Časem byla vymyšlena řada postupů, která se snaží proces hledání parametrů zefektivnit. U všech platí, že se začíná s proporcionální složkou K_p , následně se přistupuje k nastavení časové integrační konstanty T_i a nakonec se nastaví časová derivační konstanta T_d . [4], [8]

Postup ladění metodou pokus omyl:

1. Eliminuje se vliv integrační a derivační složky. To znamená, že časová integrační konstanta je nastavena na maximum a derivační na minimum.
2. Poté se zvyšuje proporcionální složka, dokud nezačne systém trvale oscilovat. Po dosažení oscilace se sníží na 45% aktuální hodnoty.
3. Následně se začne snižovat hodnota integrační časové konstanty do okamžiku opětovného vzniku trvalých kmitů. Poté se nalezená hodnota zvýší trojnásobně.
4. Posledním krokem je zvětšování časové konstanty derivační složky. Ta se zvětšuje do okamžiku dosažení opětovného kmitání. Následně se hodnota časové derivační složky vydělí třemi.

1.4.2 Metoda Ziegler-Nichols

Tato heuristická metoda zpětnovazebné regulace patří k nejstarším, ale současně nejznámějším manuálním metodám nastavování regulačních parametrů. Metoda byla publikovaná v roce 1942 a díky své exaktnosti přispěla velkou mírou k oblíbenosti PID regulátorů.

Základem je regulovanou soustavu přivést na hranici stability a následně odečíst hodnotu kritické periody T_u a kritického zesílení K_u . Z těchto hodnot se poté, pomocí empirických vztahů, vypočtou parametry regulátoru. [8], [9]

Postup za pomoci metody Ziegler-Nichols:

1. Stejně jako u metody pokus-omyl se nejprve eliminuje vliv integrační a derivační složky.
2. Zvýší se proporcionální složka K_p , tím se přivede soustava na kmitavou mez stability.
3. Z naměřených záznamů regulované veličiny se odečte hodnota kritické periody T_u a odpovídající kritické zesílení regulátoru K_u . Z odečtených kritických hodnot se dopočtou parametry regulátoru.

	K_p	T_i	T_d
P	$0,5 \cdot K_u$		
PI	$0,45 \cdot K_u$	$T_u/1,2$	
PID	$0,6 \cdot K_u$	$T_u/2$	$T_u/8$

Tabulka 1 - Empirické vztahy pro výpočet parametrů regulátoru

1.4.3 Tříparametrový model

Klasické heuristické metody, jako například zmíněná Z-N, jsou v praxi hojně užívány, ale výsledkem je často silné přeregulování systému a kmitavá regulační odchylka. Proto byly časem vyvinuty metody robustnější, s vyšší mírou stability a rychlejším ustálením. Jednou z těchto metod je i aproximace procesu pomocí tříparametrového modelu. [4]

Tříparametrový model aproximuje zaznamenanou odezvu systému na jednotkový skok. Využívá k tomu odezvu přenosu získanou Laplaceovou transformací.

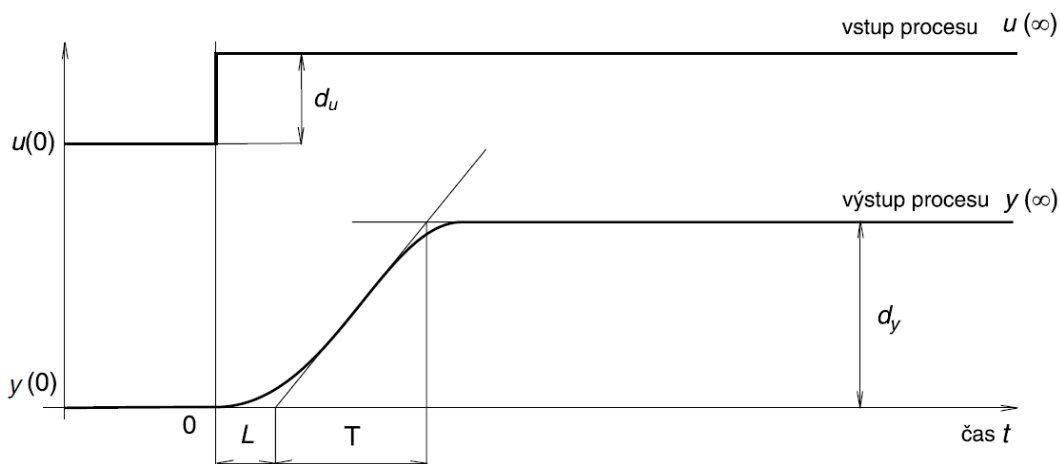
$$G(s) = \frac{K}{Ts + 1} e^{-sL} \quad (5)$$

Kde: K ... statické zesílení

T ... časová konstanta

L ... dopravní zpoždění

Parametry lze odečíst z vykreslené odezvy systému na jednotkový skok nebo je dopočíst z naměřených vzorkovaných vstupů a výstupů systému. Je nutné správně zvolit dobu vzorkování, tak aby naměřená data co nejvíce odpovídala spojitému ději.



Obrázek 3 - Odezva tříparametrového modelu na skok ^[4]

Při početním řešení se vychází z ustálených hodnot $u(\infty)$ a $y(\infty)$. Pro tyto hodnoty se určí průměrná doba ustálení T_{ar} , která se poté použije k dopočítání parametrů T a L .

$$K = \frac{d_y}{d_u} = \frac{y(\infty) - y(0)}{u(\infty) - u(0)} \quad (6)$$

$$T_{ar} = \int_0^{\infty} \frac{y(\infty) - y(t)}{y(\infty) - y(0)} dt \quad (7)$$

$$T = e \int_0^{T_{ar}} \frac{y(t) - y(0)}{y(\infty) - y(0)} dt \quad (8)$$

$$L = T_{ar} - T \quad (9)$$

Z těchto tří parametrů se následně, pomocí empirických postupů, stanoví parametry regulátoru. Zároveň lze dopočíst tzv. normalizované dopravní zpoždění, anglicky „normalized dead time“.

$$\tau = \frac{L}{L + T}, \quad \tau \in \langle 0, 1 \rangle \quad (10)$$

Na základě τ lze posoudit, jak obtížné bude proces regulovat. Pro malé hodnoty normalizovaného zpoždění je proces dobře regulovatelný i za pomoci méně robustních algoritmů. Pro vyšší hodnoty τ se výsledky pro klasické metody jako Ziegler-Nichols stávají hůře regulovatelnými a je potřeba využít některou z modernějších heuristických metod. Modernější algoritmy mají v sobě zohledněné vlivy normalizovaného zpoždění. Jednou z moderních metod je například tzv. vyvážené nastavení. [4]

$$K_P = \frac{1}{K} \left(1 - \frac{2\tau}{1 + \sqrt{1 + 2\tau^2}} \right) \quad (11)$$

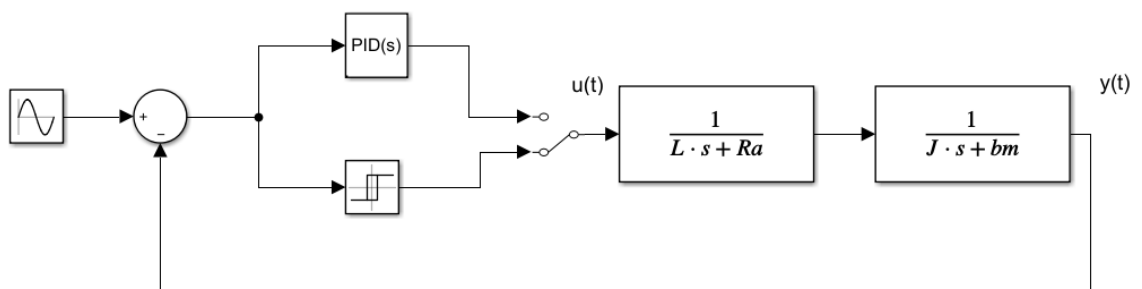
$$T_i = \left(\frac{1 + \sqrt{1 + 2\tau^2}}{2} - \tau \right) T_{ar} \quad (12)$$

$$T_D \leq \frac{T_i}{4} \quad (13)$$

1.4.4 Ladění parametrů pomocí relé

Metoda založena na umístění relé do uzavřeného zpětnovazebného obvodu představuje alternativu ke konvenčním heuristickým metodám. Zatímco identifikace parametrů kritického zesílení K_u a kritické oscilační periody T_u , dle Z-N metody, vyžaduje časově náročné nastavování proporcionální složky P regulátoru, tak pomocí relé se lze tomuto zdlouhavému procesu vyhnout a celý proces optimalizace urychlit a automatizovat.

Původní princip byl navrhnout dle Åströma a Hägglunda a uvažoval použití ideálního dvoustavového relé s obdélníkovým výstupem a nulovou hysterezí. K překlopení relé dochází pokaždé, když zpětnovazebný chybový signál změní znaménko. Relé je zapojeno v paralelní větvi k laděnému PID regulátoru a v případě potřeby ladění regulátoru se s jeho pomocí vyhledají nové konstanty do regulátoru.



Obrázek 4 - Autotuning pomocí relé

Obvod se dostane do stavu trvalé oscilace na pomezí stability do tzv. limitního cyklu. Z tohoto cyklu lze vyčíst informace o amplitudě a a periodě T_u přechodového bodu Nyquistovy křivky přes bod stability, tedy bod kde Nyquistova křivka poprvé překročí reálnou osu. Podmínku vzniku stabilního limitního cyklu lze vyjádřit za pomoci rovnice (14), kde $N(a)$ je přenos popisující relé a $G(j\omega_u)$ je přenos samotné regulované soustavy. [10]

$$1 + N(a)G(j\omega_u) = 0 \quad (14)$$

$$N(a) = -\frac{1}{G(j\omega_u)} \quad (15)$$

Záznam přenosů pomůže při identifikaci parametrů kritického zesílení K_u a kritické oscilační periody T_u , viz rovnice (16) a rovnice (17). Z vyčtených hodnot kritické periody lze dále dopočítat kritickou přechodovou frekvenci ω_u .

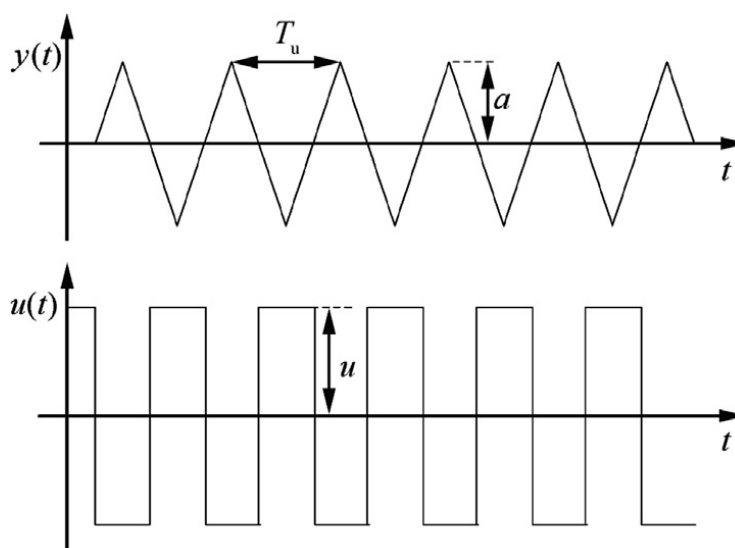
$$K_u = N(a) = \frac{4 \cdot u}{\pi \cdot a} \quad (16)$$

$$T_u = \frac{2 \cdot \pi}{\omega_u} \quad (17)$$

Pro získání zesílení lze použít už zmiňované relé bez hystereze. V případě, že je potřeba získat i časové konstanty, pak musí být do zpětnovazebné smyčky doplněn integrátor. Absence hystereze vede k snížené odolnosti vůči šumu, proto je výhodné určitou hysterezi doplnit. Překlápění stavů je poté opožděno právě o velikost této hystereze ϵ . [11]

$$K_u = N(a) = \frac{4 \cdot u}{\pi \cdot \sqrt{a^2 - \epsilon^2}} \quad (18)$$

Ve chvíli kdy jsou identifikované konstanty parametry regulované soustavy, tak se k výpočtu konstant regulátoru použijí empirické vztahy.



Obrázek 5 - Záznam výstupů relé a regulované soustavy [12]

1.4.5 Gradientní metoda

Kromě metod specializujících se právě na ladění regulátorů lze využít i obecných optimalizačních postupů. Jedním z nich je postup založený na iterativním výpočtu směru gradientu optimalizované cílové funkce. Gradientem se obecně nazývá první parciální derivace cílové funkce podle optimalizovaného parametru a určuje směr největšího růstu funkce. Při více rozměrné optimalizaci lze tyto parciální derivace parametrů seskupit do jednoho vektoru.

$$\text{grad } F(\vec{x}_j) = \nabla F(\vec{x}_j) = \vec{G}(\vec{x}_j) \quad (19)$$

$$\vec{G}(\vec{x}_j) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots, \frac{\partial F}{\partial x_n} \right]_j^T \quad (20)$$

V případech, kdy hodnotu gradientu nelze analyticky vypočítat nebo je výpočet příliš časově náročný, je potřeba využít numerických metod. Parciální derivaci lze nahradit diferenčním podílem, mezi nejpoužívanější patří jednosměrná, případně centrální diference.

Jednosměrná diference:

$$\frac{\partial F}{\partial x_1} = \frac{F(x_1 + \Delta x_1, x_2, \dots) - F(x_1, x_2, \dots)}{\Delta x_1} \quad (21)$$

Centrální diference:

$$\frac{\partial F}{\partial x_1} = \frac{F(x_1 + \Delta x_1, x_2, \dots) - F(x_1 - \Delta x_1, x_2, \dots)}{2 \cdot \Delta x_1} \quad (22)$$

Ve směru gradientu dochází k největšímu růstu, většina optimalizací je ale postavena na principu minimalizace cílové funkce. Nejinak je tomu i při využívání gradientní metody, proto se za cílovou funkci často volí velikost odchylky od ideálního řešení, kterou chceme minimalizovat a jako směr postupu gradientního algoritmu se volí směr největšího spádu, tedy směr záporný ke směru gradientu. [14]

Metoda postupu po spádnici zajistí pohyb z počátečního náhodně určeného bodu ve směru největšího poklesu cílové funkce. Při každé nové iteraci je velikost a směr gradientu, respektive směr největšího spádu, znovu aktualizován a postup má následně charakter „cik-cak“ pohybu.

Velikost výsledného vektoru směru postupu gradientního algoritmu se dále upravuje tzv. normováním, viz rovnice (23). Kde δ_j označujeme jako normovaný směr gradientního algoritmu a $\|\vec{G}_j\|$ je normou vektoru \vec{G}_j .

$$\delta_j = -\frac{\vec{G}_j}{\|\vec{G}_j\|} \quad (23)$$

$$\|\vec{G}_j\| = \sqrt{\sum_{i=1}^n \left(\frac{\partial F}{\partial x_i}\right)_j^2} \quad (24)$$

Výpočet nové hodnoty optimalizovaného parametru je poté dán rozdílem staré hodnoty parametru a kroku, o velikosti α , v záporně normovaném směru gradientu, viz rovnice (25).

$$\vec{x}_{j+1} = \vec{x}_j - \alpha \frac{\vec{G}_j}{\|\vec{G}_j\|} \quad (25)$$

Předešlá rovnice odpovídá tzv. jednoduché gradientní metodě. Největším problémem, který se musí u této metody vyřešit je správná volba kroku α . Vždy je otázkou či zvolit fixní velikost kroku nebo krok postupně upravovat dle fáze postupu algoritmu. Při volbě příliš velkého kroku hrozí, že se hledané minimum cílové funkce mine a metoda začne oscilovat. V takovém případě je potřeba přistoupit ke zmenšení kroku. Příliš malý krok naopak vede k časové náročnosti celé optimalizace. Pro zvýšení efektivity jsou doporučovány různé variace kroku jako např. občasné zařazování diagonálních kroků. [13]

Důležitá je volba podmínek, za kterých se má optimalizace ukončit. Jako podmínka se používá velikost zlepšení cílové funkce mezi jednotlivými iteracemi a pokles normy gradientu. Příliš přísné mezní podmínky celou optimalizaci zbytečně prodlužují a nevedou k výraznému zlepšení dosažených hodnot cílové funkce.

$$|F(\vec{x}_{j+1}) - F(\vec{x}_j)| \leq \varepsilon \quad (26)$$

$$\|\vec{G}_{j+1}\| \leq \varepsilon \quad (27)$$

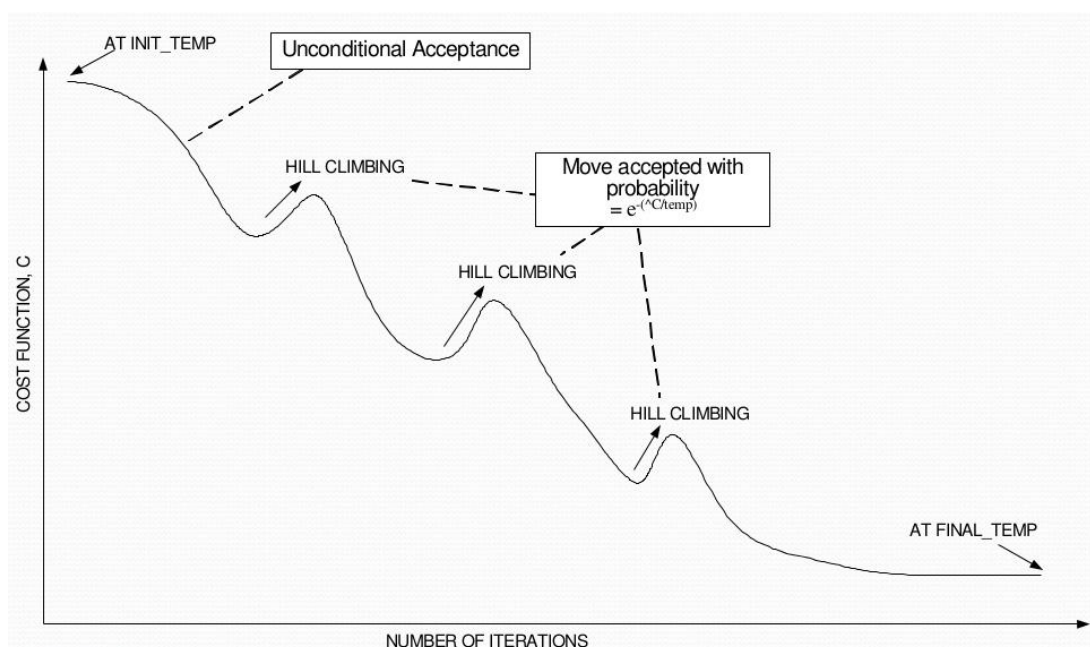
1.4.6 Metoda simulovaného žhání

Další obecnou optimalizační metodou, která připadá v úvahu pro ladění parametrů PID regulátorů je metoda simulovaného žhání. Jedná se o stochastickou optimalizační úlohu pro vyhledávání přibližného řešení ve velkých stavových prostorech, jenž by mohly být jen těžko řešeny analyticky.

Jak název napovídá, tak metoda je inspirována metalurgickým procesem při kterém, je zahřátý kov postupně ochlazován a dochází k minimalizaci jeho vnitřní energie. Ohřáté atomy během procesu žhání kmitají a náhodně přeskakují, během chladnutí se těleso jako celek snaží dostat do rovnovážného stavu a snížit svou vnitřní energii. [15]

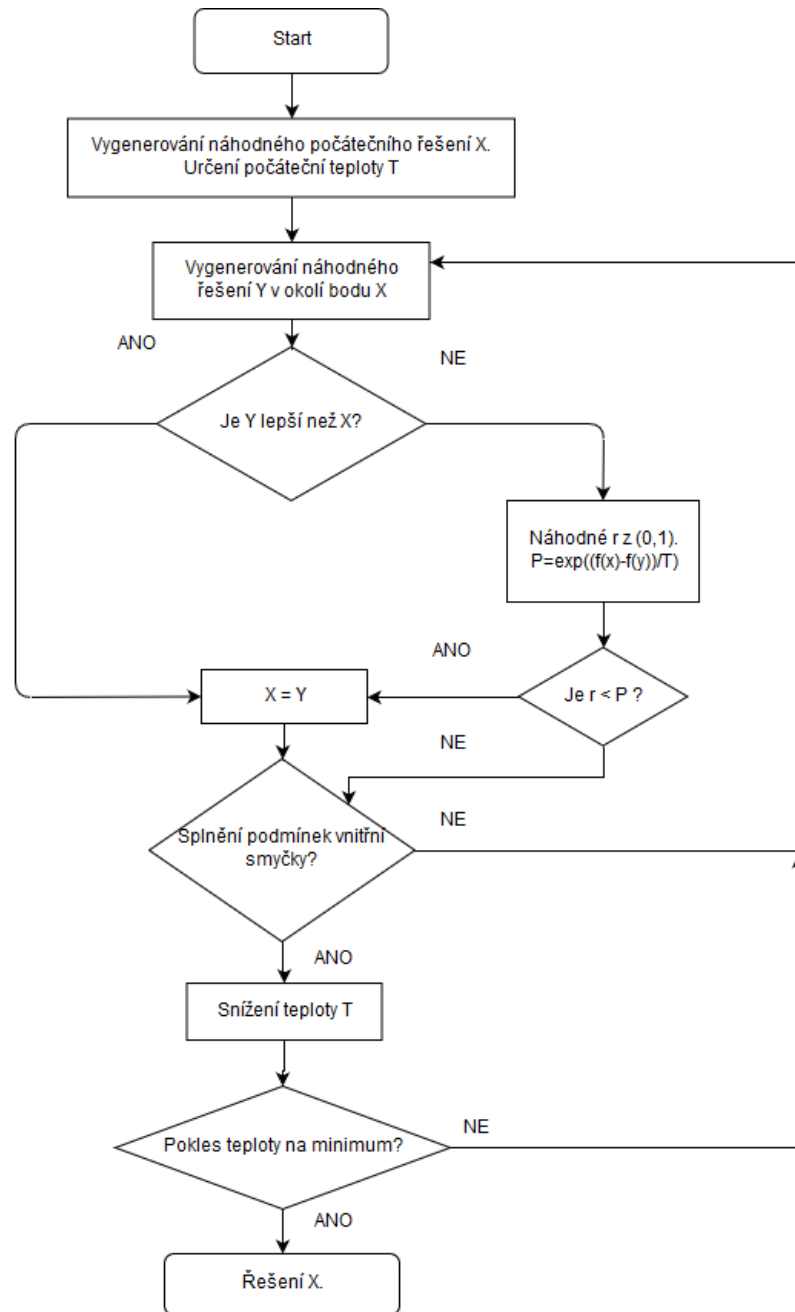
Podobný princip přeskakování a ustalování v bodech s minimem energie, byl převzat právě do algoritmu metody simulovaného žhání. Během procesu „chladnutí“ se snižuje pravděpodobnost přeskoků do jiných rovnovážných pozic a na konci algoritmu by mělo být nalezeno minimum cílové funkce. [16]

Možnost přeskoku do pozic s horší hodnotou cílové funkce řeší problém uváznutí v lokálním minimu, který se vyskytuje u řady jiných algoritmů, např. i u gradientní metody.



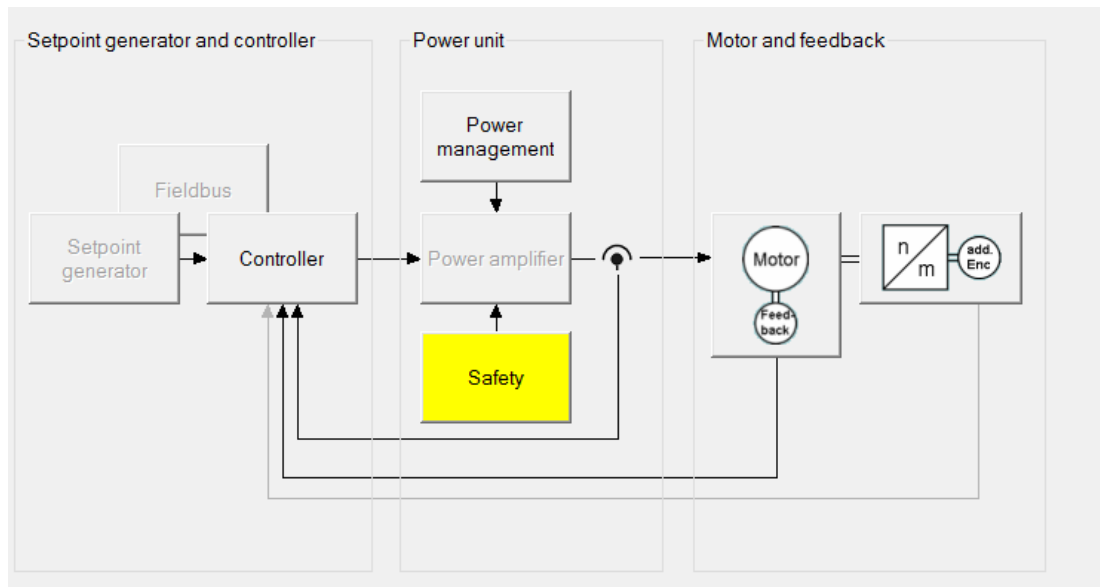
Obrázek 6 - Snižování hodnoty cílové funkce během chladnutí [17]

Algoritmus se skládá ze dvou, do sebe vnořených, smyček. Jedné vnější a jedné vnitřní. Ve vnější smyčce postupně dochází ke snižování teploty, čímž se snižuje i možnost přeskočení algoritmu do bodu s horší hodnotou cílové funkce. Ve vnitřní smyčce se zkoušejí náhodně vybrané body z okolí aktuálního bodu funkčního minima a testuje se pravděpodobnost přeskočení.



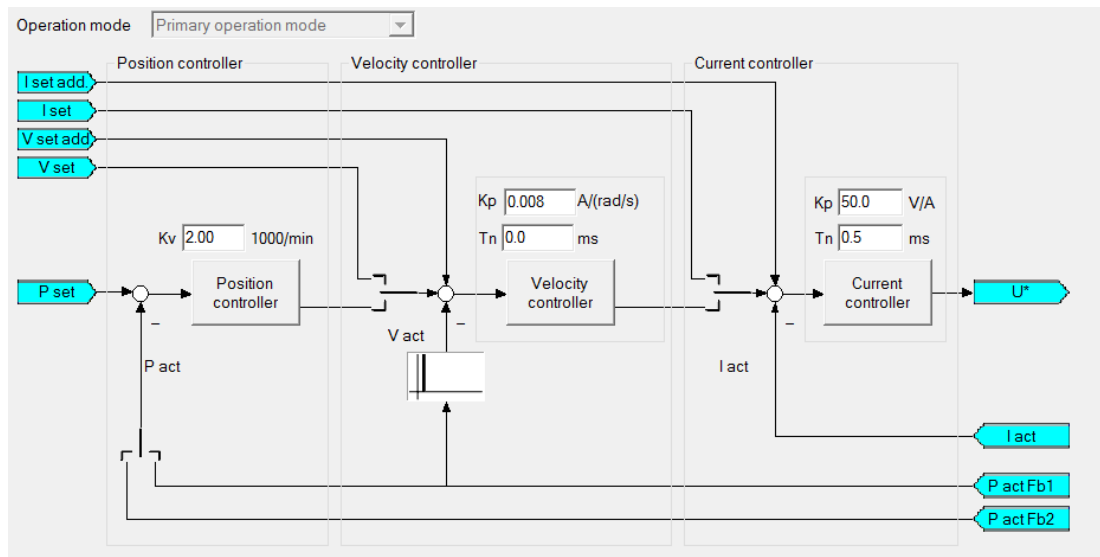
Obrázek 7 - Schématický popis algoritmu simulovaného žíhání

Z prostředí softwaru TwinCAT se lze připojit k měniči připojenému k danému počítači. Beckhoff AX5206 umožňuje v reálném čase monitorovat stav motoru a upravovat parametry motoru i samotného měniče.



Obrázek 9 - Výchozí řídicí schéma měniče

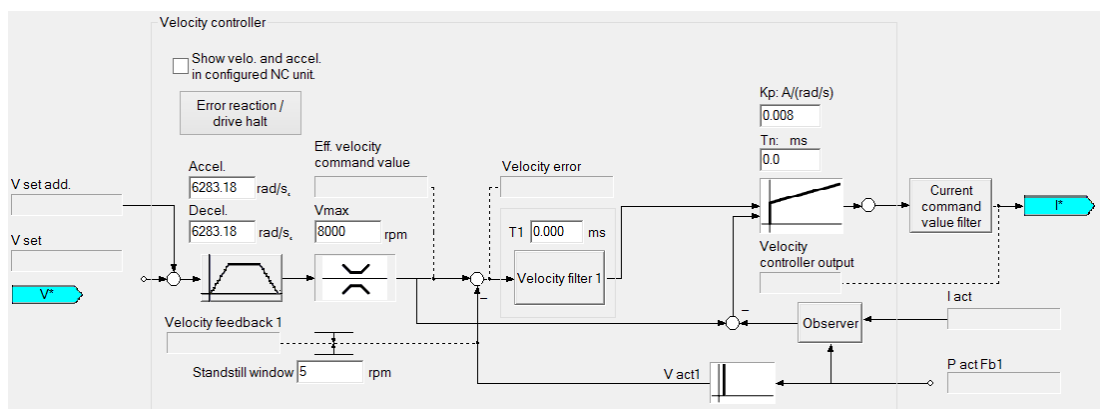
Po připojení k měniči je nejzajímavější záložka „Drive manager“ a v ní konkrétně informace o řídicí struktuře v bloku „Controller“. Odtud je patrné, že řízení je realizováno klasickou kaskádovou formou, jež již byla přiblížena v úvodní teoretické části práce, viz kapitola 1.2 Kaskádní regulace.



Obrázek 10 - Kaskádní regulace použita v AX5206

Ze schématu lze vyčíst, že regulace poziční smyčky je realizována pomocí proporcionálního regulátoru. V rychlostní a proudové smyčce jsou použity proporcionálně-integrační regulátory. To dává dohromady 5 základních parametrů, které se budou moci ladit v rámci optimalizačních algoritmů.

Při bližším pohledu do struktur jednotlivých smyček, namátkou je vybrána regulační část rychlostní smyčky. Tato část obsahuje řadu dalších parametrů, kterými lze chod motoru ovlivnit. Lze nastavit maximální míru strmosti náběhů a poklesů jednotlivých regulovaných veličin, dále omezovat velikost regulačních zásahů a aplikovat filtry.



Obrázek 11 - Schéma rychlostní regulační smyčky v měniči AX5206

Zároveň lze z nastavení jednotlivých zpětnovazebných smyček vyčíst jejich takt. Dalším zakomponovaným prvkem je feed-forward řízení, které má schopnost částečně předvídat výchytky regulovaných veličin a tím pozitivně přispět ke kvalitě regulačních zásahů.

Jednotlivé prvky v tomto kontrolním schématu se dají dle uvážení měnit, případně úplně vypustit. Daná struktura řízení dává velkou svobodu v rámci možných nastavení a významně ulehčí prvotní rozběh motoru. Jakmile bude motor v provozu, bude možné předdefinovanou strukturu regulačních smyček obejít a implementovat regulaci vycházející ze Simulinkového modelu.

2.2 Simulační model motoru

Poté, co jsme se blíže seznámili se strukturou řízení v měniči, tak se může začít se sestavováním simulačního modelu ve vývojovém prostředí Simulink. Prvním úkolem je sestavení jednoduchého modelu motoru, ze kterého bude možno získávat zpětnovazebnou odezvu pro regulační smyčky. V těchto smyčkách budou následně namodelovány regulátory, na nichž bude možno testovat optimalizační algoritmy.

Snažit se zachytit všechny prvky použité v jednotlivých regulačních smyčkách bloku „Controller“ by bylo zbytečně složité. Pro účely této práce postačí jednoduchý elektrický model simulovaného motoru rozšířen o část zohledňující jeho mechanické vlastnosti, konkrétně setrvačnost rotoru.

Výpočty elektrické části modelu budou, pro zjednodušení, prováděny v transformovaných souřadnicích d-q, které jsou pevně spjaté s rotorem. Magnetické toky v transformovaných osách lze vyjádřit dle rovnic (28) a (29).

$$\Psi_d = L_d i_d + \Psi_f \quad (28)$$

$$\Psi_q = L_q i_q \quad (29)$$

Kde L_q , L_d , i_q a i_d jsou indukčnosti a proudy v jednotlivých osách, Ψ_f je tzv. spřažený magnetický tok magnetu. Za pomoci znalosti magnetických toků se můžou vyjádřit napěťové rovnice v referenčních osách.

$$u_d = R_a i_d + \frac{d\Psi_d}{dt} - \omega \Psi_q = R_a i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \quad (30)$$

$$u_q = R_a i_q + \frac{d\Psi_q}{dt} - \omega \Psi_d = R_a i_q + L_q \frac{di_q}{dt} - (\omega L_d i_d + \Psi_f) \quad (31)$$

Z hlediska vytvářeného modelu je důležitější rovnice pro napětí u_q , která obsahuje proud i_q . Tento proud je tzv. momentotvornou složkou. V této rovnici je obsaženo i zpětná elektromagnetická síla reprezentovaná členem $(\omega L_d i_d + \Psi_f)$. Tuto sílu z důvodu celkového zjednodušení modelu nebudu uvažovat.

Po této úvaze se dostáváme k výsledné rovnici, jež bude považována za dostačující model elektrické části motoru. Jediné neznáme konstanty, které bude nutné identifikovat, jsou hodnota indukčnosti L_q a hodnota odporu statorového vinutí R_a

$$u_q = R_a i_q + L_q \frac{di_q}{dt} \quad (32)$$

Nyní se může přistoupit k řešení mechanických veličin daných dynamikou motoru. Momentotvorná složka proudu po vynásobení momentovou konstantou motoru k_M poskytne představu o získaném točivém momentu.

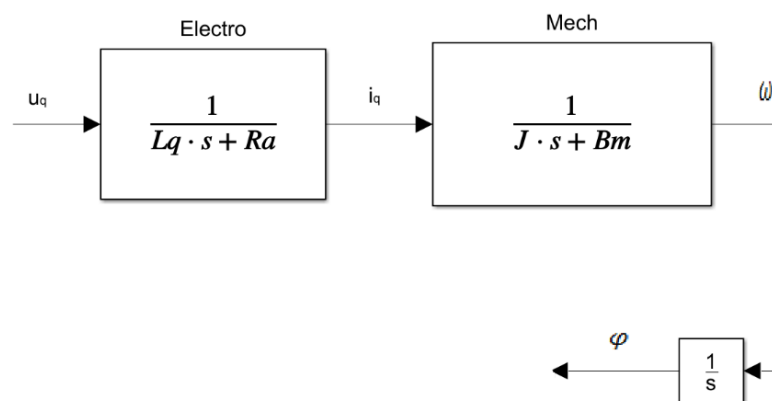
$$M = k_M \cdot i_q \quad (33)$$

Ten po dosazení do pohybové rovnice zohledňující setrvačnost rotoru J a vlivu tření B_m dovolí vyjádření úhlové rychlosti ω . Následnou integrací lze získat i natočení rotoru φ .

$$J \cdot \frac{d\omega}{dt} + B_m \cdot \omega = M - M_z \quad (34)$$

Je uvedena pohybová rovnice i s možnou připojenou zátěží, která by generovala zátěžný moment M_z . Při simulaci bude počítáno jen s hmotou samotného motoru a žádná zátěž připojena nebude.

Po získání stavových rovnic pro elektrické i mechanické veličiny, budou rovnice podrobeny Laplaceově transformaci. Po transformaci je bude snadné přidat do Simulinku.



Obrázek 12 - Model motoru

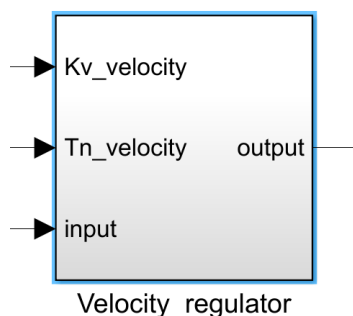
2.3 Simulační model řízení

Ze struktury řízení, vyčtené z „Drive manager“ v softwaru TwinCAT, byla zjištěna podoba regulátorů. Je potřeba namodelovat proporcionální regulátor pro polohovou smyčku a proporcionálně-integrační regulátory pro rychlostní a proudovou smyčku.

Proporcionální regulátor, jehož problematika byla poodhalena v kapitole 1.3.1, je jednoduchý a bude nasimulován za pomoci vynásobení vstupního signálu odchylky požadované polohy $e(t)$ regulačním parametrem $K_v position$.

$$u(t) = K_v position \cdot e(t) \quad (35)$$

Proporcionálně-integrační regulátory, použité ve zbývajících dvou smyčkách, budou vzájemně identické. Z důvodů toho, že se budou parametry regulátorů během optimalizace v předem daných simulačních cyklech měnit, je výhodné regulátory vytvořit jako samostatné bloky se vstupy a výstupy. Vstupy bude vždy dvojice optimalizovaných parametrů zesílení $K_v velocity$ a časová konstanta $Tn velocity$, respektive $K_v current$ a $Tn current$. Dalším vstupem bude odchylka regulované veličiny. Výstupem bude zadávaná veličina pro další regulátor, potažmo přímo motor.



Obrázek 13 - Blok PI regulátoru

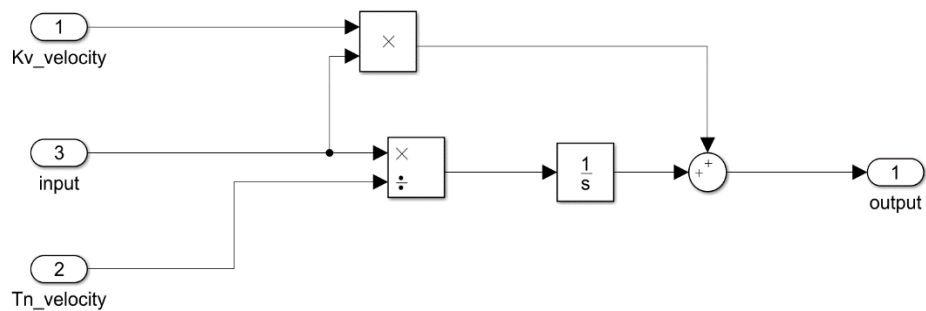
Při tvorbě vnitřní podoby těchto bloků máme na výběr mezi dvěma možnými provedeními. První z možných provedení je PI regulátor se sériovou strukturou, také nazýván jako tzv. ideální PI regulátor, viz rovnice (36). U tohoto typu hodnota proporcionální složky ovlivňuje i složku integrační.

$$u(t) = K_v velocity \cdot \left(e(t) + \frac{1}{Tn velocity} \cdot \int_0^t e(t) dt \right) \quad (36)$$

Druhou možností je PI regulátor v paralelním provedení. Hodnoty jednotlivých složek se zde neovlivňují a na výstupu z regulátoru se obě složky pouze sečtou.

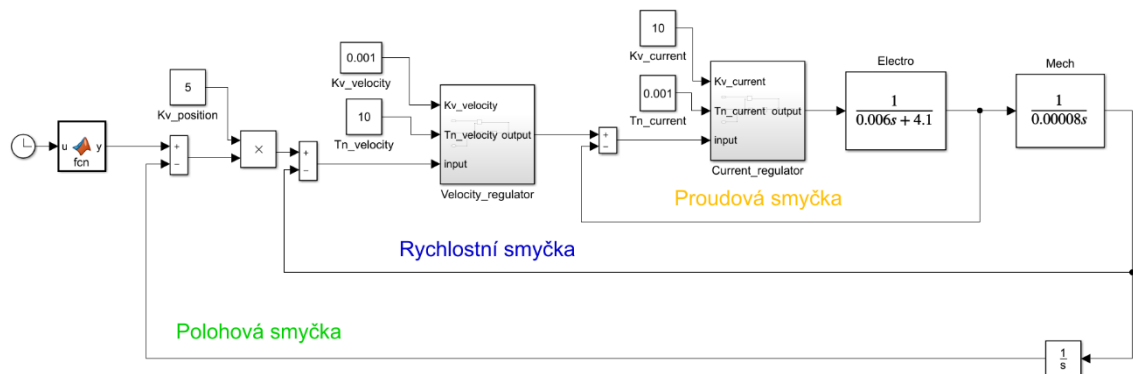
$$u(t) = K_v \text{ velocity} \cdot e(t) + \frac{1}{T_n \text{ velocity}} \cdot e(t) \quad (37)$$

Tento typ je v praxi běžnější. Protože ze schématu řízení získaného z „Drive manageru“ není zřejmé, které ze dvou provedení je použito, tak máme možnost volby. Pro naše potřeby se jeví jako lepší paralelní regulátor a to z důvodu, že při optimalizaci jednoho parametru nemusíme mít obavy ze změn vyvolaných vlivem zapojení do série.



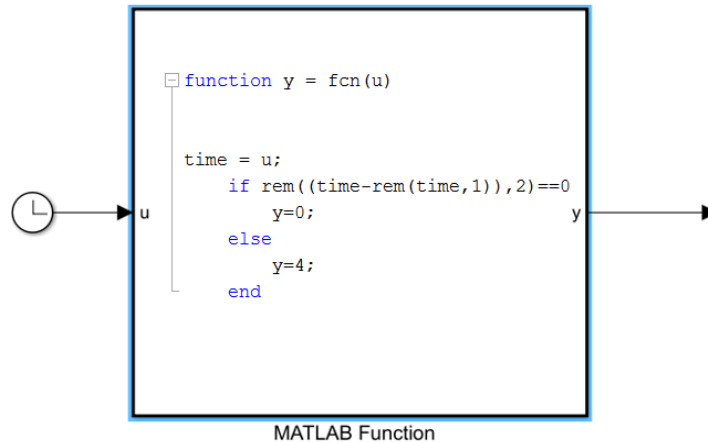
Obrázek 14 - Vnitřní struktura PI regulátoru

Poté co byl dokončen model motoru a řídicí struktura s regulátory, tak se vše může propojit dohromady. Do modelu motoru lze doplnit skutečně identifikované hodnoty jednotlivých parametrů. Procesem identifikace se bude zabývat kapitola 3.



Obrázek 15 - Model motoru s řízením

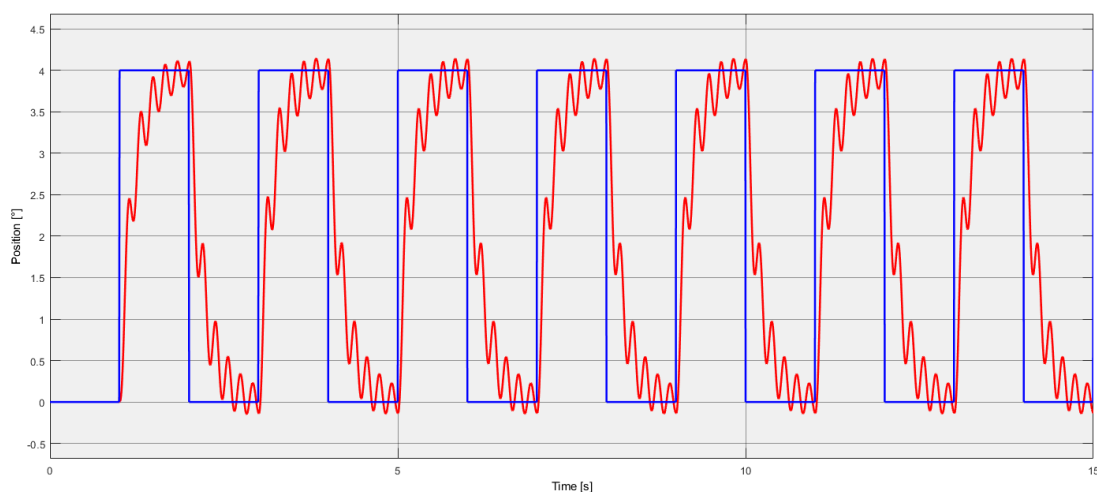
Poslední věcí, kterou zbývá vyřešit, je určit podobu testovacího signálu. Byla zvolena podoba opakujícího se step signálu. Vždy po vteřině dojde k vzestupnému či sestupnému stepu, jenž je v základní verzi modelu bez optimalizace určen sudou či lichou hodnotou simulačních času.



Obrázek 16 - Zadávání testovacího signálu

Daný step je prováděn jen o malou hodnotu, v experimentu to budou 4 stupně. Hlavním důvodem je to, aby při následném překlopení modelu do hardwarově implementační verze neměl reálný motor problémy se sérii rychle po sobě jdoucích velkých kroků.

Nakonec je potřeba uskutečnit test modelu a řízení. Provede se první nástřel parametrů regulátorů a otestuje, zda je model schopen aspoň částečně sledovat zadávanou trajektorii natočení.



Obrázek 17 - Test modelu

2.4 Optimalizace pomocí gradientní metody

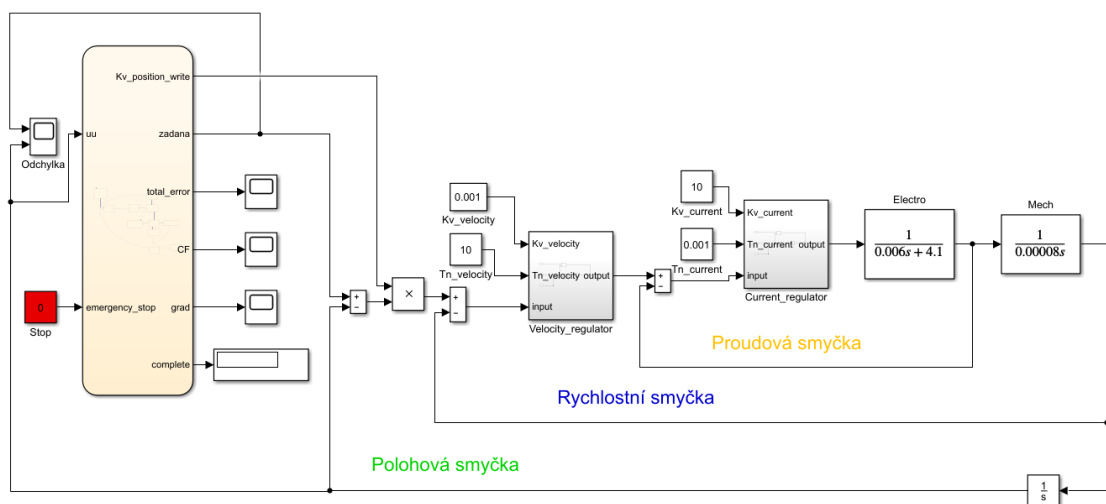
V předchozích kapitolách byl sestaven a otestován model motoru s řízením. Test řízení proběhl úspěšně, ale z jeho výsledků je patrné, že regulátory je potřeba naladit.

První metodou, která bude k optimalizaci použita, je gradientní metoda. Jak již bylo přiblíženo v úvodu práce, tato metoda vypočítává směr největšího růstu, tzv. gradient, optimalizované funkce a následně pokračuje proti směru gradientu a minimalizuje zvolenou funkci.

2.4.1 Algoritmus gradientní metody

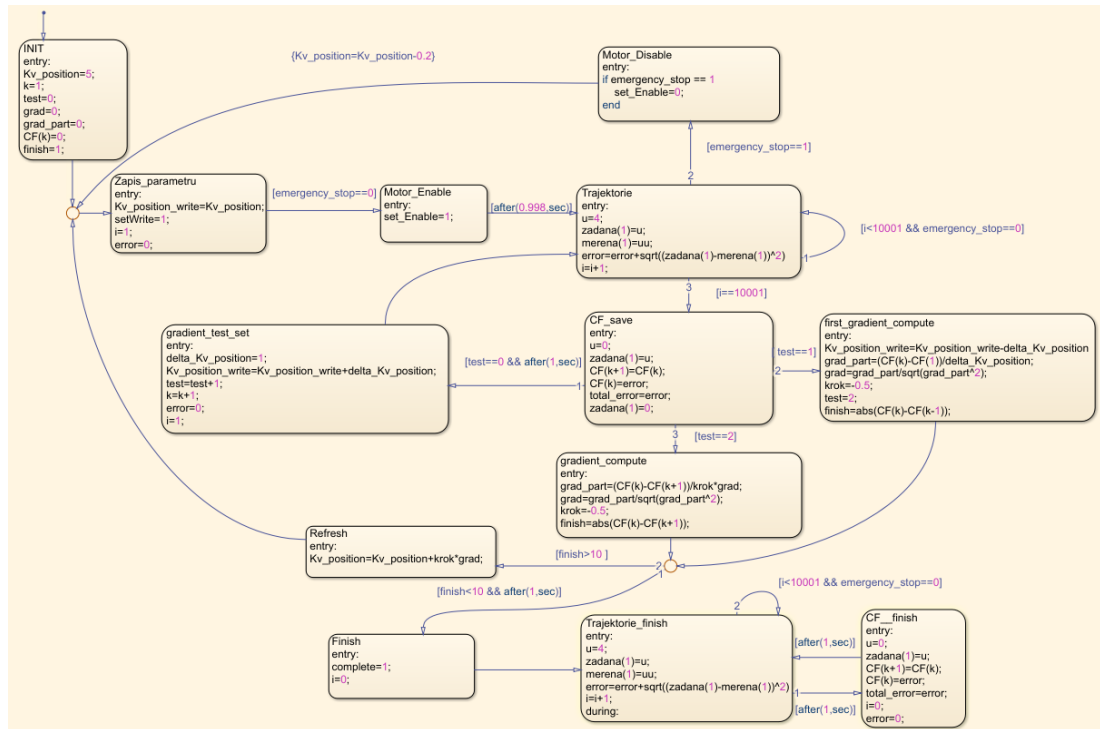
Dohromady máme 5 parametrů regulátorů, jenž je možno za pomoci gradientního algoritmu optimalizovat. Začít optimalizovat všechny parametry najednou by bylo složité, a proto se začne pouze s jedním parametrem. Pro začátek bylo zvoleno zesílení K_v *position* v proporcionálním regulátoru, jenž reguluje polohovou smyčku. Hodnoty ostatních parametrů budou ponechány konstantní.

Algoritmus metody bude uskutečněn v bloku stavového automatu, který bude připojen k simulačnímu modelu. Tento automat najdeme v Simulinku pod názvem „Stateflow (Chart)“. Jeho hlavní výhodou je to, že umožňuje intuitivní programování algoritmu a tím zvyšuje efektivitu práce.



Obrázek 18 - Gradientní optimalizace pro jeden parametr

Stavový automat byl zakomponován do polohové smyčky. Jako jeho vstup poslouží informace o aktuálním natočení získaná z modelu motoru. Všechny prováděné optimalizace budou zaměřené na hodnoty polohy a jejich odchylku od zadávané trajektorie.



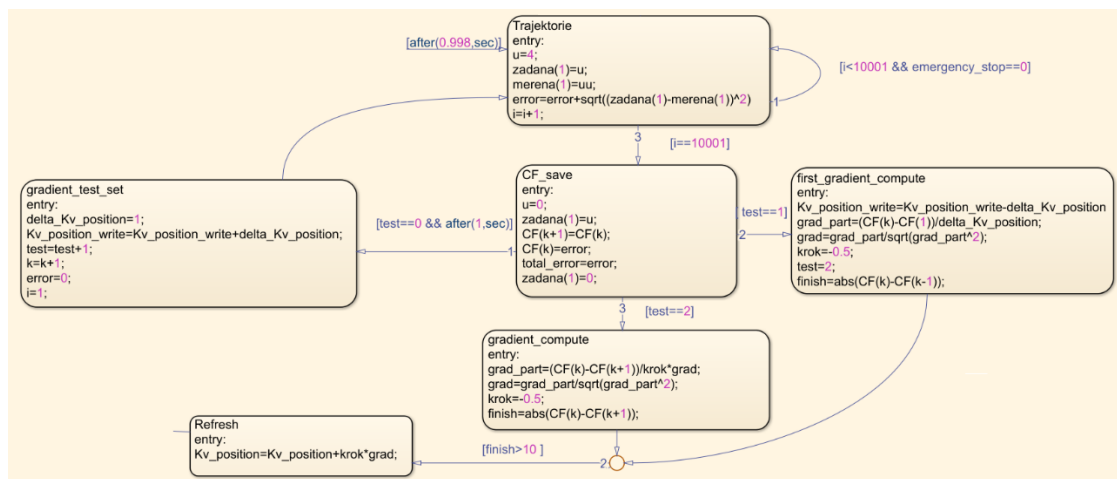
Obrázek 19 - Stavový diagram gradientního algoritmu

Každá proměnná uvnitř diagram musí být řádně deklarovaná a musí být zvoleno, zda se jedná o lokální proměnnou diagramu, vstup nebo naopak výstup do Simulinku. Zaoblené bloky, z kterých se digram skládá, představují jednotlivé stavy. Uvnitř každého stavu je seznam příkazů, který se má během daného stavu vykonat. Vykonání jednoho bloku trvá jeden simulační časový krok. Ten je převzat ze základního simulačního časového kroku Simulinku, v našem případě je tento krok nastaven pevně na 0.1ms.

Stavy se vzájemně propojují orientovanými přechody. Každý přechod je uskutečněn po vykonání instrukcí v bloku. Přechodům lze nastavovat různé stupně priority a jejich uskutečnění podmiňovat podmínkami. Zároveň lze nastavit časovou prodlevu mezi přechodem z jednoho stavu do druhého

Stavový diagram začíná ve stavu „INIT“. Tento výchozí stav musí být označen šipkou tzv. defaultního přechodu. V tomto stavu jsou taky nadefinované výchozí hodnoty proměnných. Následuje přechod do dvou, po sobě jdoucích, stavů souvisejících se zamýšlenou hardwarovou implementací. Tyto dva stavy, „Zapis parametru“ a „Motor enable“, obsahují předpřipravené proměnné, jenž poslouží při propojování implementačního modelu a měniče.

Pomyslné srdce optimalizačního algoritmu tvoří stavy zobrazené ve výřezu na obrázku 20. V bloku „Trajektorie“ dochází k zadávání, již dříve popisované, obdélníkové trajektorie. Dále se v tomto bloku počítá odchylka mezi zadanou a měřenou polohou. Odchylka představuje cílovou funkci, kterou budeme chtít v průběhu optimalizace minimalizovat.



Obrázek 20 - Logika gradientního algoritmu

Jakmile je známa hodnota cílové funkce pro inicializační nastavení regulátoru, tak může být přistoupeno k bloku „gradient test set“. V něm se odehraje posun optimalizovaného parametru o malou hodnotu $\delta Kv_position$. Následně se naměří nová hodnota cílové funkce pro takto změněný parametr regulátoru. Z rozdílu hodnot dvou, po sobě jdoucích, vzorků cílové funkce se určí gradient.

Gradient se normalizuje do jednotkového tvaru a poté už se může za pomoci záporného kroku vypočítat nová hodnota $Kv_position$ vedoucí k minimalizaci cílové funkce. V podstatě stavy „gradient test set“ a „first gradient compute“ nedělají nic jiného, než že počítají jednosměrnou diferenci.

Následujícím stavem je tzv. „Refresh“ v němž se aktualizuje hodnota K_v , *position* a algoritmus přechází do další iterace. Po prvním průchodu smyčky už není potřeba realizovat stavy „gradient test set“ a „first gradient compute“, bude se procházet pouze přes stav „gradient compute“.

Velikost kroku, se kterým se mění optimalizovaná veličena, zůstává neměnná. Změnou velikosti kroku by mohlo být dosaženo vyšší efektivity při hledání minima cílové funkce. Zároveň by špatně zvolený postup volby kroku, mezi jednotlivými iteracemi, mohl celý proces optimalizace zkomplikovat.

Algoritmus se zastaví, když zlepšení cílové funkce mezi iteracemi poklesne pod uživatelem zadanou mez. Zároveň lze motor okamžitě zastavit pomocí proměnné *emergency stop*, která slouží pro zastavení motoru při nenadálých komplikacích.

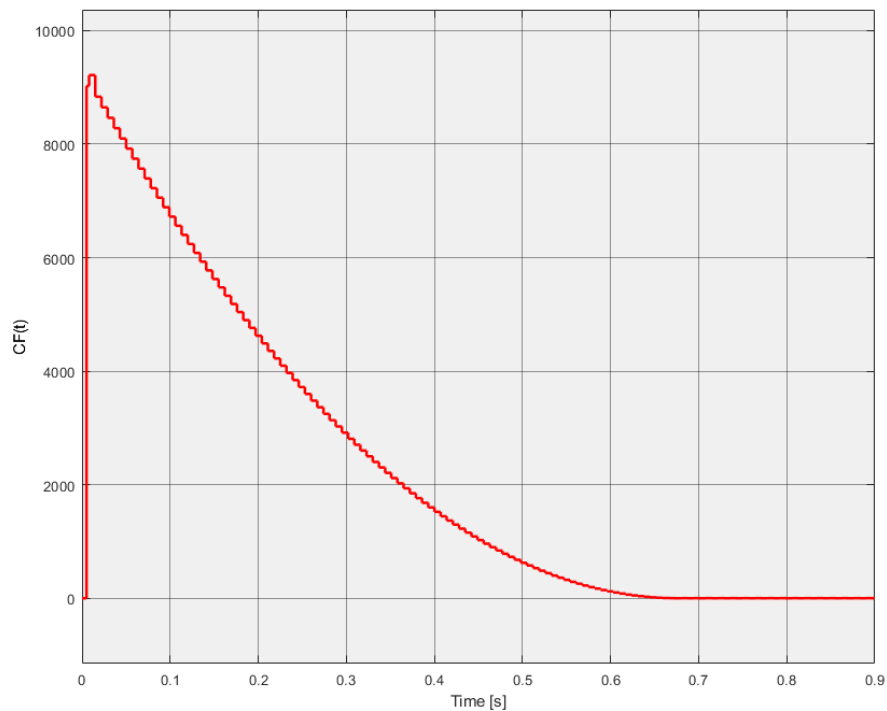
2.4.2 Validace gradientního algoritmu

Ve chvíli, kdy je optimalizační algoritmus připraven k aplikačnímu nasazení, je vhodné provést otestování postupu na některém analyticky řešitelném problému. Tím bude dosaženo validace celého postupu a budeme mít větší jistotu při jeho použití na modelu motoru, případně později při implementaci na reálný motor.

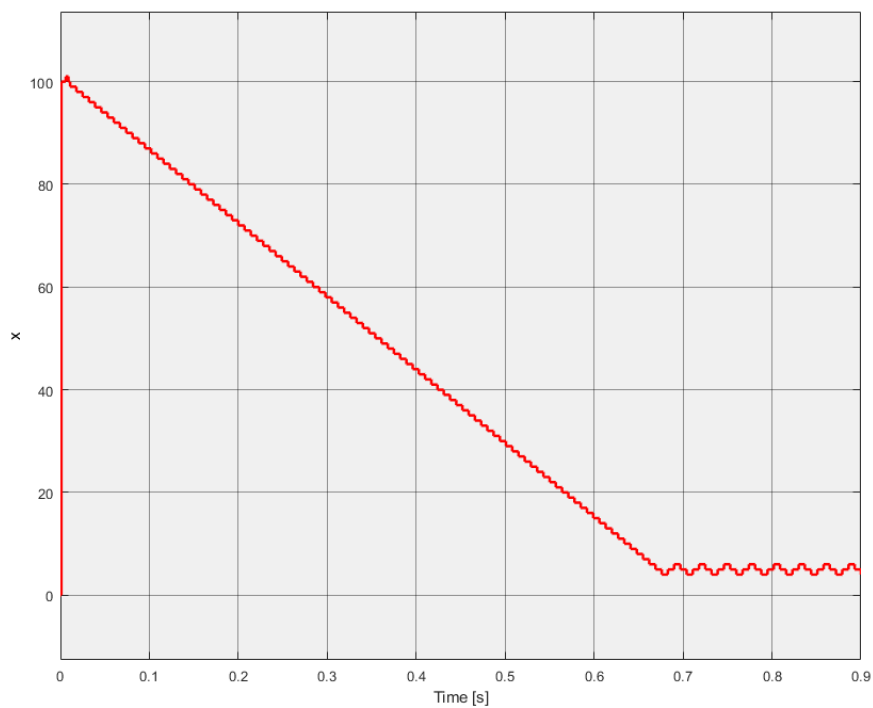
Validace proběhne bez připojeného modelu motoru a řídicích smyček s regulací. Testuje se jenom algoritmus vyhledávání minima. Testovací funkce bude mít předpis dle rovnice (38).

$$CF(t) = (x - 5)^2 \tag{38}$$

Hledané minimum cílové funkce by se mělo nacházet v bodě $CF(t) = 0$, pro množinu reálných čísel. Z rovnice (38) lze analyticky dopočítat, že by minimum mělo být nalezeno pro $x = 5$, pro $x \in R$.



Obrázek 21 - Minimalizace testovací funkce



Obrázek 22 - Nalezení hodnoty parametru x

Optimalizovaný parametr x skutečně klesl, z inicializační hodnoty $x = 100$, na analyticky vypočtenou hodnotu. Ve chvíli kdy ji překmitl, došlo ke změně směru gradientu a návratu do minima vedoucím k oscilaci kolem hledané hladiny.

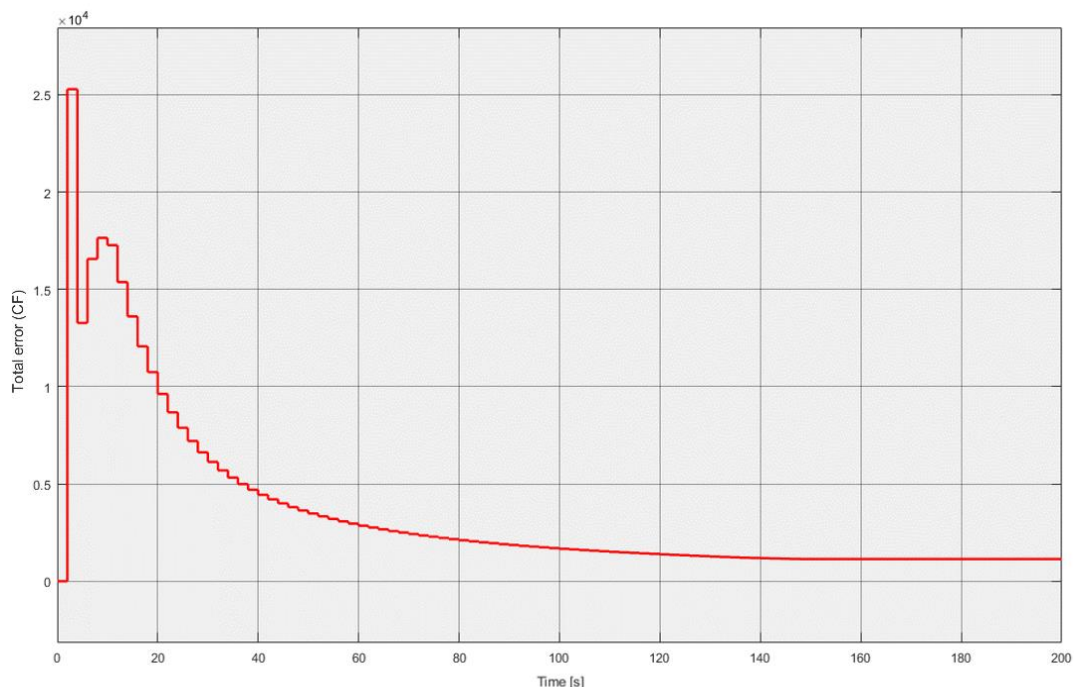
2.4.3 Optimalizace jednoho parametru

Po ověření správné funkce algoritmu gradientní metody, lze ke stavovému automatu připojit zpátky regulační smyčky s modelem motoru, dle obrázku 18. Jak bylo avizováno, prvním optimalizovaným parametrem bude $K_v position$. Ostatní parametry budou ponechány konstantní. Hodnoty konstantních parametrů regulátorů a výchozí hodnota $K_v position$ jsou v tabulce číslo 2.

Parametry regulátorů	Hodnoty parametrů	Velikost kroku
$K_v position$	1	0.5
$K_v velocity$	0.008	0
$K_v current$	10	0
$T_n velocity$	10	0
$T_n current$	0.001	0

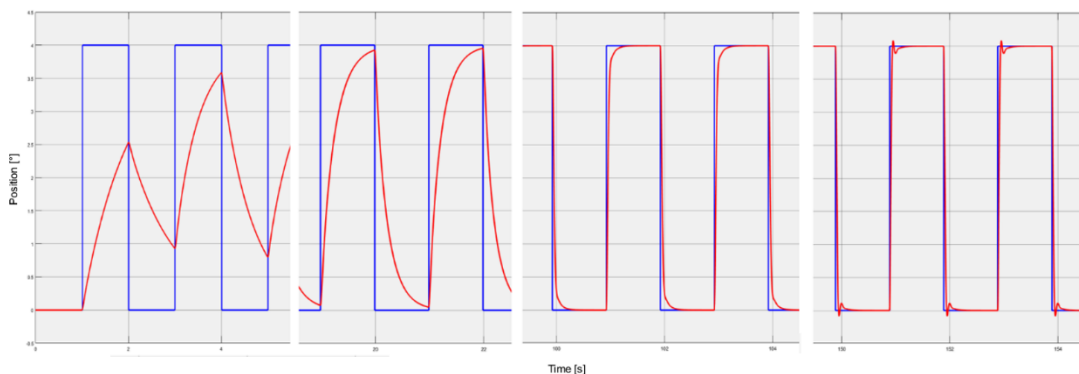
Tabulka 2 - Výchozí parametry regulátorů

Simulace je spuštěna na 200 sekund a kritérium pro ukončení optimalizace je pokles zlepšení cílové funkce, mezi po sobě jdoucími iteracemi, pod 10. Bez tohoto kritéria by simulace běžela dlouhou dobu a zlepšení tím získané by bylo mizivé.



Obrázek 23 - Pokles cílové funkce během simulační optimalizace

Během simulace byla splněna podmínka pro úspěšné ukončení gradientního algoritmu a bylo nalezeno řešení $K_v position = 36$. Z obrázku 23 je patrné, že pokles chyby, respektive minimalizace cílové funkce, byl opravdu výrazný.



Obrázek 24 - Dosažené naladění polohové smyčky při simulační optimalizaci

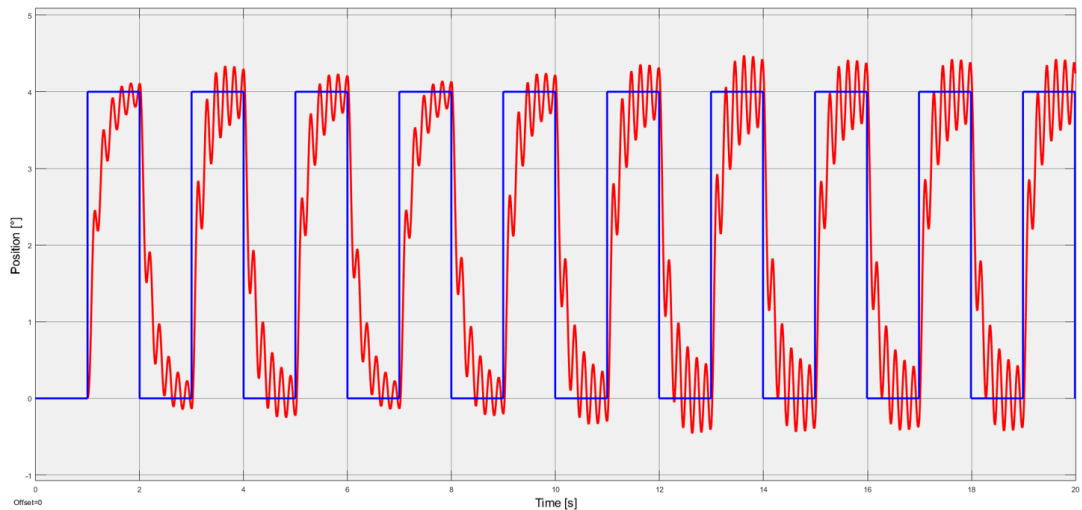
Inicializační hodnota parametru $K_v position$, dle naměřených poloh, nebyla dostačující, motor během jednotlivých kroků nestíhal následovat zadávanou polohu. Je patrné, že s přibývajícimi iteracemi se regulátory dostaly blíže k optimálnímu naladění.

Zajímavé je, že těsně před ukončením procesu ladění se začala projevovat snaha motoru překmitnout přes maximální zadanou hodnotu polohy. Při těchto překmitech se i nadále snižovala velikost cílové funkce. Tento jev lze vysvětlit tím, že výraznější odchylka v kratším časovém úseku vede v součtu jedné iterace k menší celkové chybě než menší odchylka přetrvávající na delším časovém úseku.

Další simulace se blíže zaměří právě na jev překmitů. K tomu musí být upraveny výchozí parametry v ostatních regulátorech. Vyzkoušíme nastavení, jež bylo použito při úplně prvním testu řízení a vykazovalo větší míru kmitavosti, viz tabulka 3.

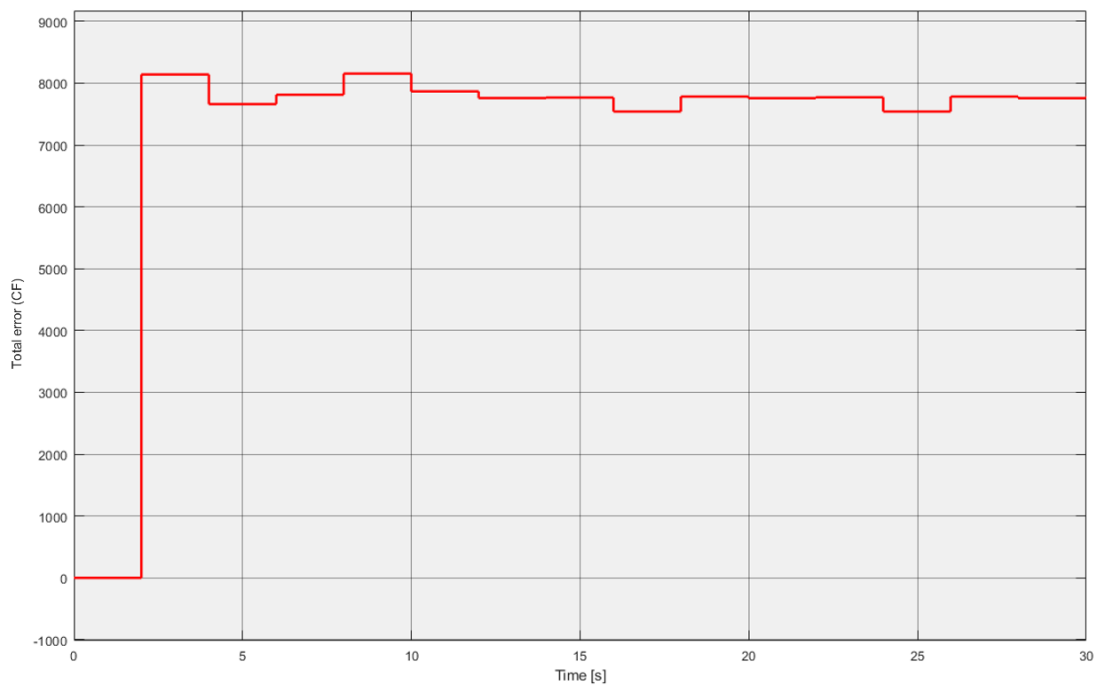
Parametry regulátorů	Hodnoty parametrů	Velikost kroku
$K_v position$	5	0.5
$K_v velocity$	0.001	0
$K_v current$	10	0
$T_n velocity$	10	0
$T_n current$	0.001	0

Tabulka 3 - Výchozí nastavení 2. simulace



Obrázek 25 - Dosažené naladění polohové smyčky během 2. simulace

Výsledky získané během druhého měření jasně dokazují, jak velký vliv mají parametry v ostatních regulátorech. Zároveň se opětovně projevila snaha algoritmu snížit časový úsek s chybou za cenu překmitů.



Obrázek 26 - Vývoj cílové funkce během 2. simulace

Cílovou funkci se za pomoci algoritmu podařilo drobně snížit, ale pak se optimalizace dostala do bodu, kdy gradient začal překmitávat z jednoho směru do druhého a řešení se už dále nezlepšovalo. Pro dosažení lepších výsledků bude muset být optimalizační algoritmus doplněn o parametry dalších regulátorů.

2.4.4 Optimalizace dvou parametrů

Jak se ukázalo během druhého simulačního pokusu, tak velký vliv na výsledek optimalizace má parametr K_v *velocity*. Proto byl zvolen jako další v pořadí, který se bude optimalizovat.

V simulačním model z obrázku 18 je poměrně univerzální a jeho rozšíření o další parametr by mělo být snadné. Nejprve bude do stavového automatu přidána proměnná K_v *velocity*. Pro tu bude určen stejný postup výpočtu gradientu jako pro K_v *position*. Do automatu přibudou pouze dva nové stavy „gradient velocity test set“ a „first gradient velocity compute“. Tyto dva stavy se zařadí hned za stávající stavy pro výpočet gradientu polohy a představují jednosměrnou diferenci dle K_v *velocity*. Nově vypočtená veličinu K_v *velocity* bude vyvedena ven ze stavového automatu pomocí proměnné K_v *velocity write*. Ta bude připojena ke vstupu v regulátoru rychlosti.

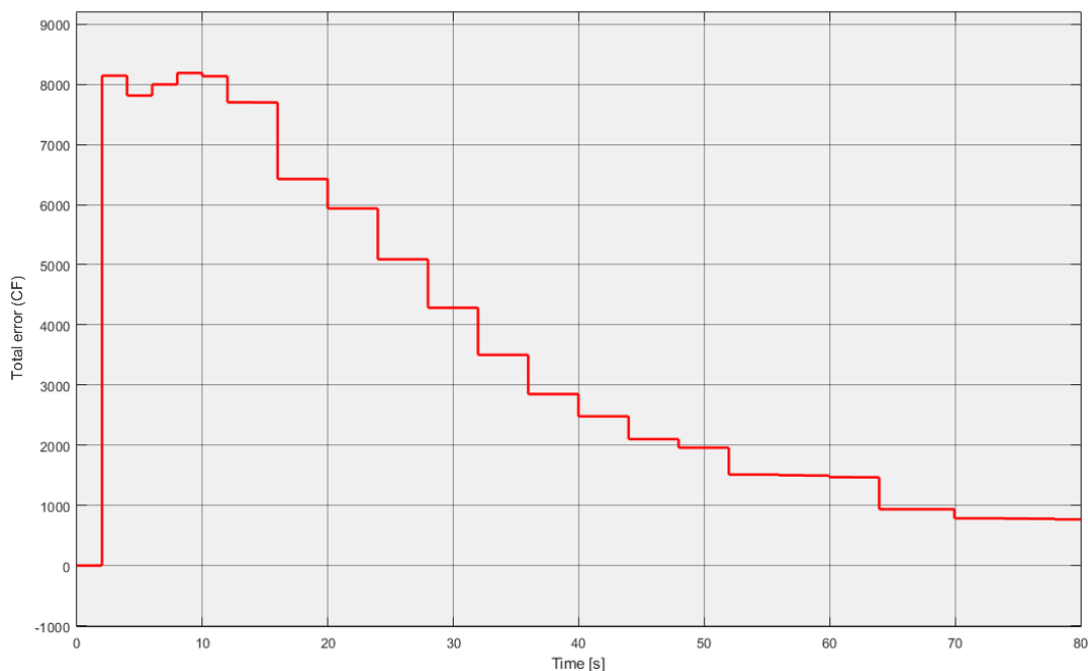
Výchozí nastavení parametrů bude stejné jako u poslední realizované optimalizace. Rozdíl bude v tom, že nyní se budou současně optimalizovat parametry polohového a rychlostního regulátoru.

Ve výpočtu kroku byla provedena také drobná změna. Bude ustavován podle toho, jakou má gradient dané proměnné podíl nenormovaného gradientu k sumě gradientů. Takto vypočtený podíl by měl pomoci určit, která z daných proměnných vyžaduje větší změnu kroku.

Parametry regulátorů	Hodnoty parametrů	Velikost kroku
K_v <i>position</i>	5	0.5
K_v <i>velocity</i>	0.001	0.001
K_v <i>current</i>	10	0
T_n <i>velocity</i>	10	0
T_n <i>current</i>	0.001	0

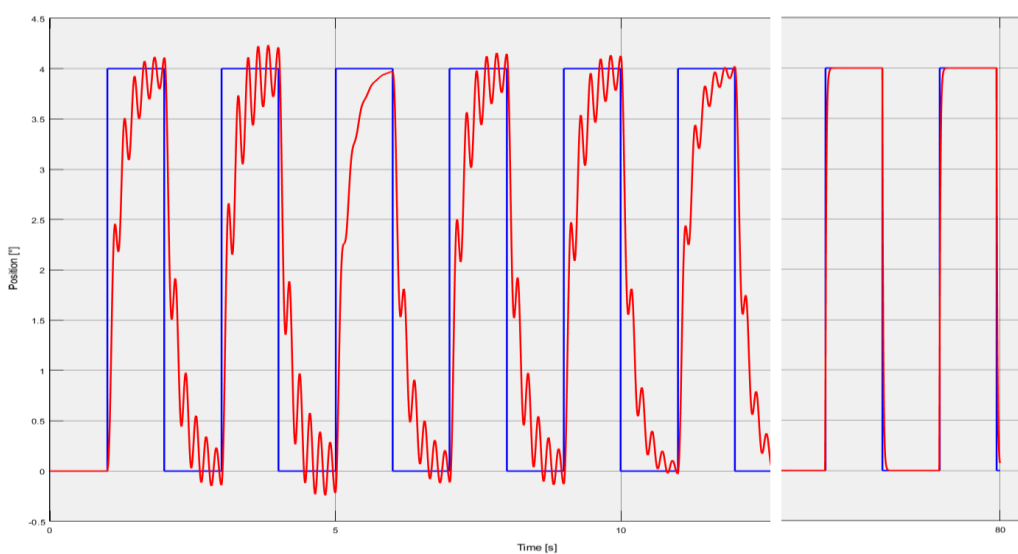
Tabulka 4 - Výchozí nastavení pro simulační optimalizaci dvou parametrů

Za pomoci těchto úprav a především přidáním druhého parametru bylo dosaženo uspokojivé minimalizace chyby.



Obrázek 27 - Pokles cílové funkce při simulační optimalizaci dvou parametrů

Řízení polohy se díky naladění regulátorů výrazně zlepšilo. Při prvních iteracích je jasně patrný sklon ke vzniku oscilací a překmitů. S přibývajícimi optimalizačními cykly algoritmu tyto oscilace vymizely a ke konci simulace bylo sledování trajektorie takřka dokonalé. Výsledné optimalizované parametry byly $K_v position = 52$ a $K_v velocity = 0.095$.



Obrázek 28 - Naladění polohové smyčky při sim. optimalizaci dvou parametrů

2.4.5 Optimalizace více parametrů

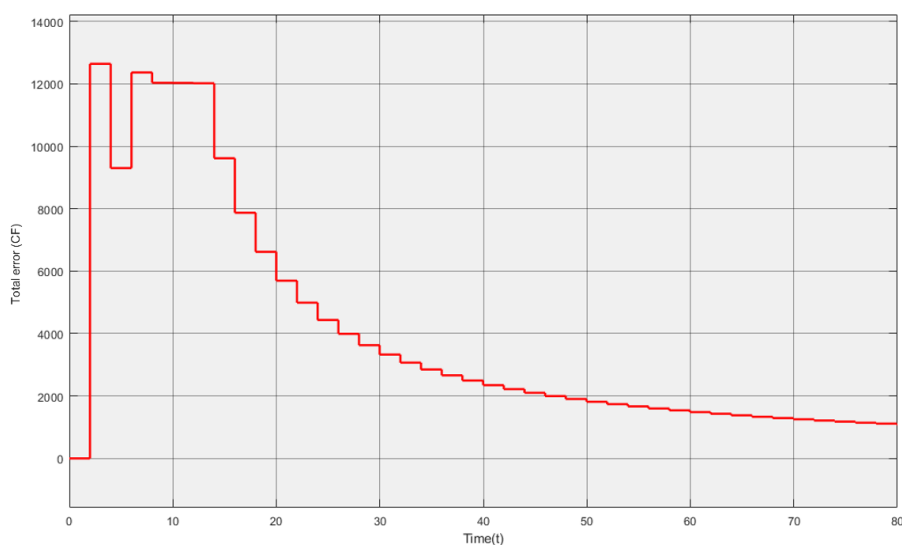
Poté co bylo dosaženo úspěšného naladění dvou parametrů regulátorů, je na čase vyzkoušet optimalizační metodu pro všech 5 parametrů současně. Do algoritmu budou přidány parametry K_v current, T_n velocity a T_n current.

Stavový automat, obsahující gradientní algoritmus, bude rozšířen stejným způsobem jako při přidávání druhého parametru. Vstupy do regulátorů rychlosti a proudu budou připojeny ke stavovému automatu. Výchozí nastavení zesílení a časových konstant je zapsáno v tabulce 5. Bylo vybráno jiné nastavení výchozích parametrů, aby se odzkoušela optimalizace z většího počtu výchozích bodů.

Parametry regulátorů	Hodnoty parametrů	Velikost kroku
K_v position	3	1
K_v velocity	0.001	0.001
K_v current	14	1
T_n velocity	20	1
T_n current	0.003	0.0005

Tabulka 5 - Výchozí nastavení optimalizace 5 parametrů

Pokles cílové funkce, vyjádřen velikostí odchylky od požadované polohy, byl poměrně plynulý. Je vidět, že v prvních cyklech klesá cílová funkce strměji a ke konci optimalizace už je pokles jen nepatrný.



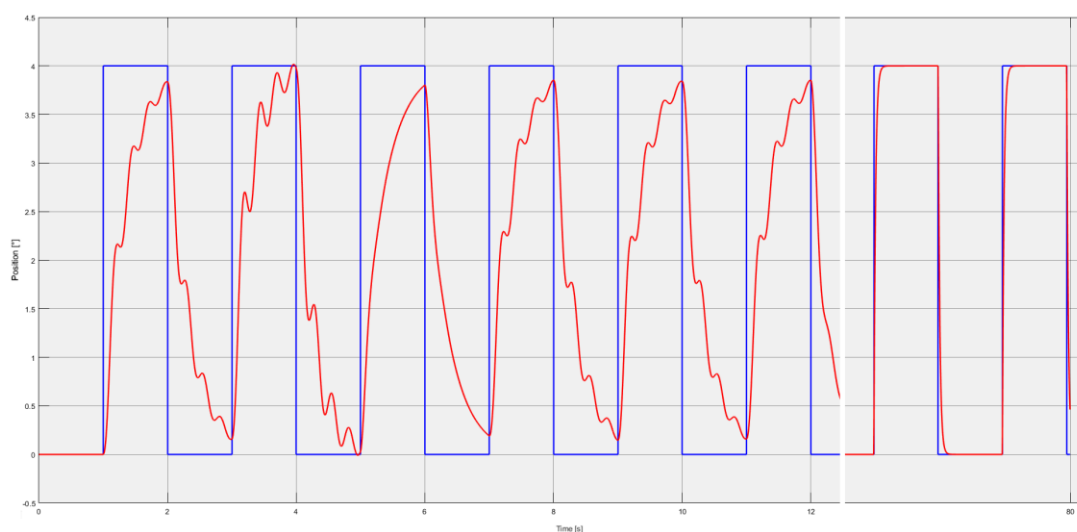
Obrázek 29 - Pokles cílové funkce při simulační optimalizaci pěti parametrů

Vykreslované odchylky naměřené polohy od té zadávané se během simulace významně snížily. Získané naladěné parametry byly zapsány do tabulky 6.

Parametry regulátorů	Hodnoty parametrů	Velikost kroku
K_v position	38	1
K_v velocity	0.035	0.001
K_v current	49	1
T_n velocity	14	1
T_n current	0.02	0.0005

Tabulka 6 - Naladěné parametry

Tyto hodnoty se liší od získaných hodnot v předchozích měřeních. Při pohledu na kvalitu řízení je patrné, že tyto rozdíly nemají příliš vliv. Nejspíše je to způsobeno tím, že optimalizace měla jiný výchozí bod a parametry regulátorů se navzájem během optimalizace ovlivňují.



Obrázek 30 - Naladění polohové smyčky při sim. optimalizaci pěti parametrů

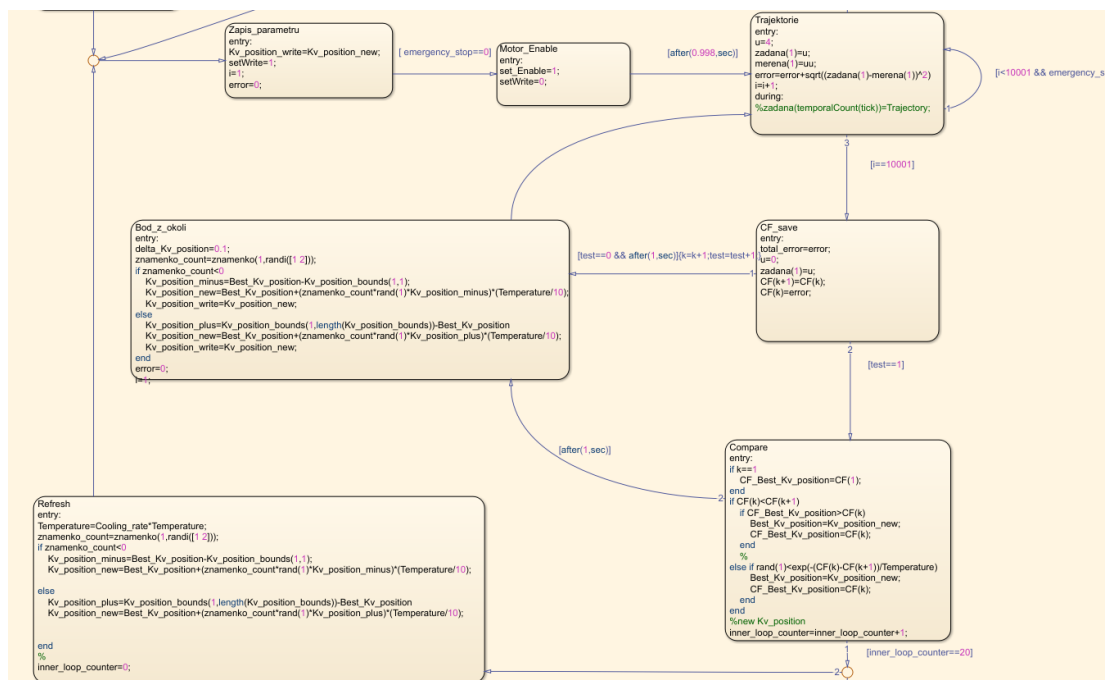
2.5 Optimalizace pomocí simulovaného žihání

Druhou testovanou optimalizační metodou bude metoda simulovaného žihání. Blíže jsme se s ní seznámili v kapitole 1.4.6. Tato metoda je založena na prohledávání prostoru možných řešení za pomoci pravděpodobnosti a konvergence k řešení díky „ochlazování“. Výhodou této metody je to, že by neměla uváznout v lokálním minimu.

2.5.1 Algoritmus simulovaného žihání

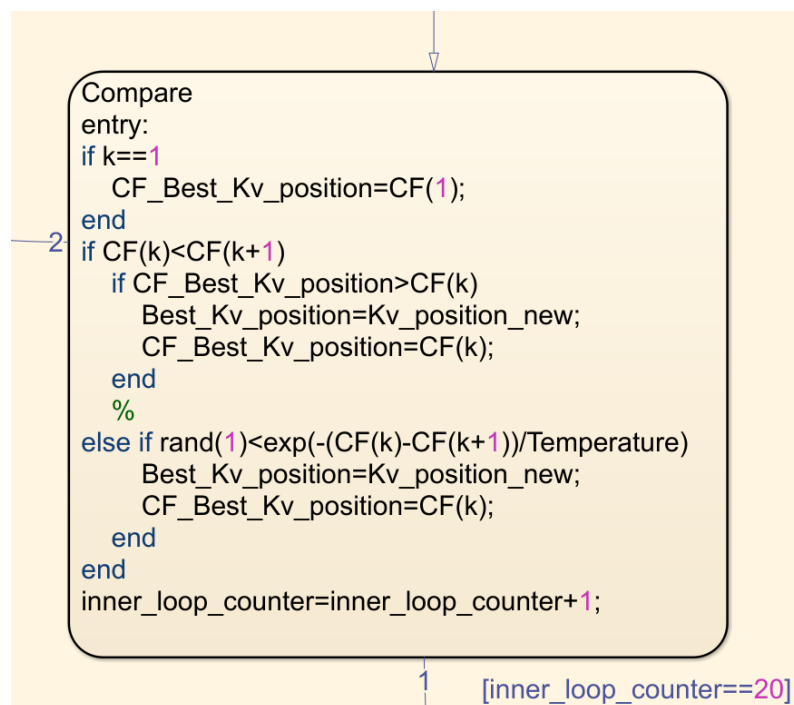
Podobně jako u gradientní metody bude i algoritmus simulovaného žihání realizován pomocí stavového automatu. Bude přepracována již používaná struktura automatu pro gradientní metodu.

Stavový diagram začíná ve stavu „INIT“. V tomto stavu jsou nadefinované výchozí hodnoty proměnných a nově i množina, ve které se má hledaný parametr nacházet. Tato množina musí být dopředu určena a vyžaduje, aby uživatel měl aspoň hrubou představu kde řešení hledat. Následuje přechod přes stavy „Zapis parametru“ a „Motor enable“. Následuje stav s názvem „Trajektorie“. Jedná se o totožný stav jako pro gradientní metodu a i zadávaná trajektorie je totožná. Cílovou funkci představuje velikost odchylky mezi zadávanou a měřenou trajektorií, tudíž ji budeme chtít minimalizovat.



Obrázek 31 - Iterační smyčky algoritmu simulovaného žihání

Simulované žihání je dvousmyčkový algoritmus. Ve vnitřní smyčce se počítají náhodně zvolené body z okolí počátečního bodu, v našem případě výchozí hodnoty regulátorů. Tyto body mohou být přijaté jako nové nejlepší řešení a nahradit počáteční bod. K tomu dojde, jestliže mají nižší cílovou funkci, případně když splní podmínku akceptace danou pravděpodobností ve stavu „Compare“.



Obrázek 32 - Blok stavu "Compare"

Přijetí bodu s horší hodnotou cílové funkce má zajistit, že algoritmus neuvázne v lokálním minimu. Pravděpodobnost v akceptační podmínce je dána náhodně vygenerovaným číslem od nuly do jedné. Jestliže je náhodné číslo menší než záporně vzatá chyba dělená teplotou, pak je řešení přijato.

Ve vnější smyčce se počítá pokles teploty. Na začátku optimalizace je dána výchozí teplota a tzv. *Cooling rate*. Ten při každém průchodu vnější smyčky sníží teplotu. S klesající teplotou klesá i pravděpodobnost splnění akceptační podmínky na přijetí horšího řešení. Zároveň se snižuje prohledávané okolí nejlepšího bodu.

Za pomoci nastavení výchozí teploty, parametru *Cooling ratu* nebo počtu vnitřních cyklů lze zvýšit pravděpodobnost nalezení optimálního řešení. Zároveň se tím ale může neúnosně prodloužit doba optimalizace. Algoritmus je ukončen ve chvíli, kdy teplota klesne pod uživatelem zvolenou mez.

2.5.2 Optimalizace jednoho parametru

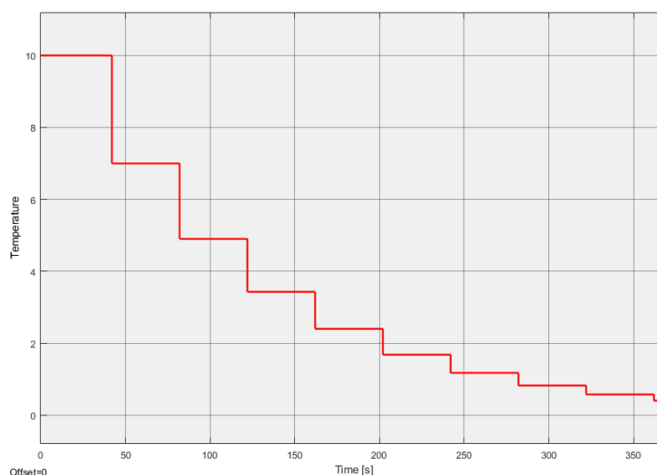
Stejně jako u gradientní metody začneme s optimalizací pouze jednoho parametru a to zesílením polohové smyčky $K_v position$. Díky tomu, že už máme představu o chování modelu, můžeme nastavit, v jakých mezích se má optimalizovaný parametr nacházet. Přesnější nastavení tohoto intervalu by mělo urychlit optimalizaci. Budeme hledat řešení v intervalu $K_v position \in < 0, 40 >$.

Nastavíme počáteční teplotu, rychlost ochlazování a další výchozí parametry. Nastavení je v tabulce parametrů číslo 7. Výchozí nastavení regulátorů bude stejné jako u první simulační optimalizace gradientní metodou.

Výchozí parametry	Hodnoty parametrů
<i>Teplota</i>	10
<i>Cooling rate</i>	0.7
<i>$K_v position$</i>	1
<i>$K_v velocity$</i>	0.008
<i>$K_v current$</i>	10
<i>$T_n velocity$</i>	10
<i>$T_n current$</i>	0.001

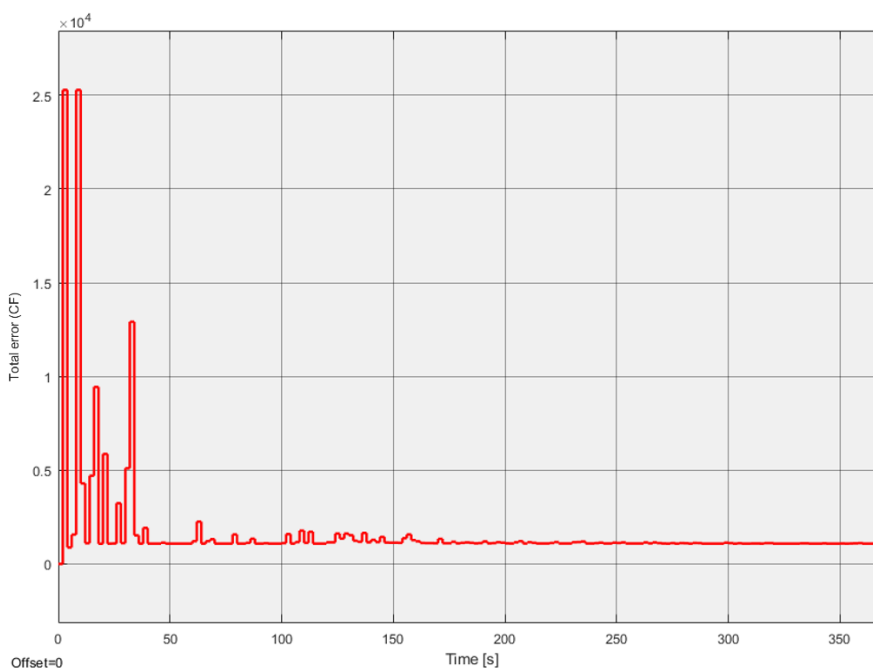
Tabulka 7- Výchozí parametry simulace

Během simulace postupně klesá cílová teplota a spolu s ní se i ustaluje hodnota cílové funkce v nalezeném minimu. Algoritmus se zastaví při poklesu teploty pod hodnotu 0,5.



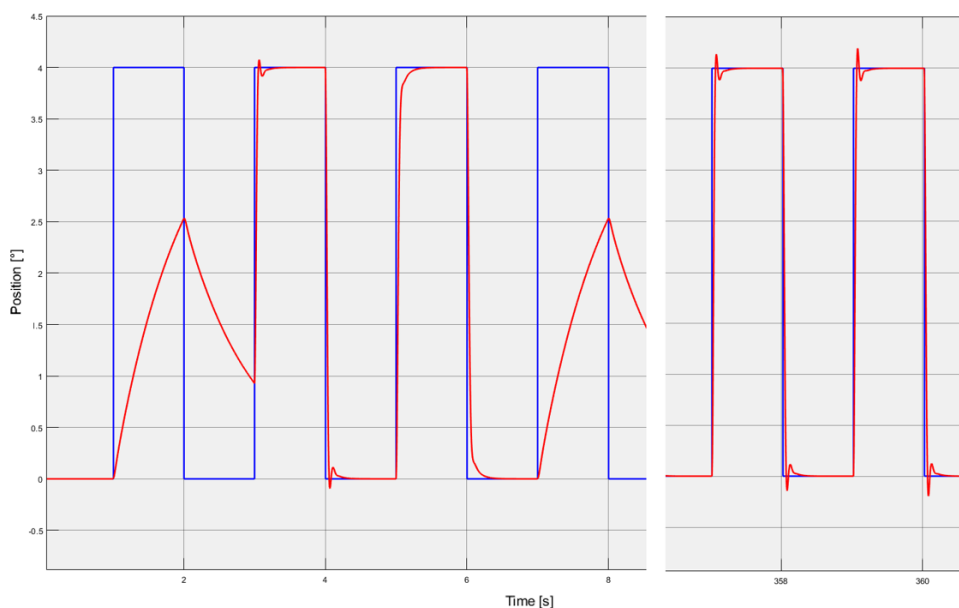
Obrázek 33 - Pokles teploty v průběhu simulační optimalizace

Přestože je pokles hodně strmý, tak je dosaženo pěkného ustálení cílové na hodnotě $Total\ error(CF) = 1100$. To odpovídá hodnotě dosažené gradientním algoritmem.



Obrázek 34 - Pokles cílové funkce během simulační optimalizace

Regulátor byl algoritmem naladěn na hodnotu $K_v\ position = 39.9$. Gradientní metodou byla získána hodnota $K_v\ position = 36$. Znovu se projevuje vliv překmitů, které by bylo vhodné penalizovat.



Obrázek 35 - Dosažené naladění polohové smyčky při simulační optimalizaci

2.5.3 Optimalizace dvou parametrů

Dalším krokem je rozšíření algoritmu o další parametr. Stejně jako u gradientní metody bude zvoleno zesílení rychlostního regulátoru K_v *velocity*. Uvnitř stavového automatu není nutné přidávat žádné nové stavy. Do těch stávajících stačí jenom dopsat rovnice pro výpočet K_v *velocity*, které budou podobné rovnicím pro K_v *position*.

Meze K_v *position* budou rozšířeny na interval K_v *position* $\in < 0.1, 100 >$. Pro K_v *velocity* bude řešení hledáno na intervalu K_v *velocity* $\in < 0.0005, 0.5 >$. Bylo by možné nastavit i širší meze, ale poté by prohledávání prostoru řešení vyžadovalo příliš dlouhou dobu a muselo by se uskutečňovat s mnohem větším počtem iterací ve vnitřní i vnější smyčce. Zbytek nastavení je převzat z gradientní optimalizace pro 2 parametry. Díky tomu bude možné porovnat výsledky obou metod.

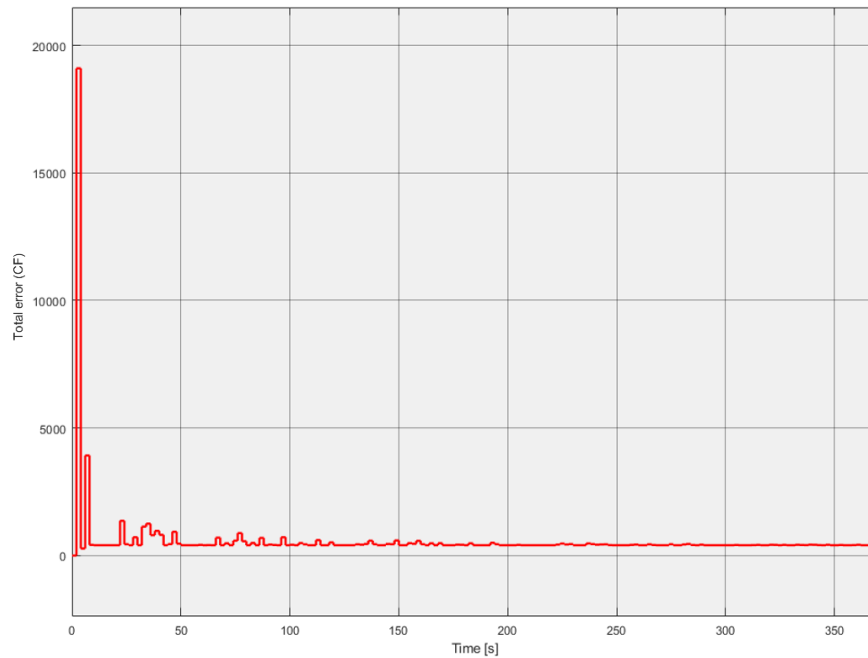
Výchozí parametry	Výchozí hodnoty parametrů
<i>Teplota</i>	10
<i>Cooling rate</i>	0.7
K_v <i>position</i>	5
K_v <i>velocity</i>	0.001
K_v <i>current</i>	10
T_n <i>velocity</i>	10
T_n <i>current</i>	0.001

Tabulka 8 - Výchozí nastavení simulace

Po dokončení simulace byly získány dvojice optimalizovaných parametrů K_v *position* = 99.5 a K_v *velocity* = 0.49, je pravděpodobné, že při širších mezích by rostly ještě dál. Hodnoty získané gradientní metodou byly K_v *position* = 52 a K_v *velocity* = 0.095. Rozdíl těchto hodnot je značný a mohl by ukazovat na špatné počáteční nastavení optimalizací.

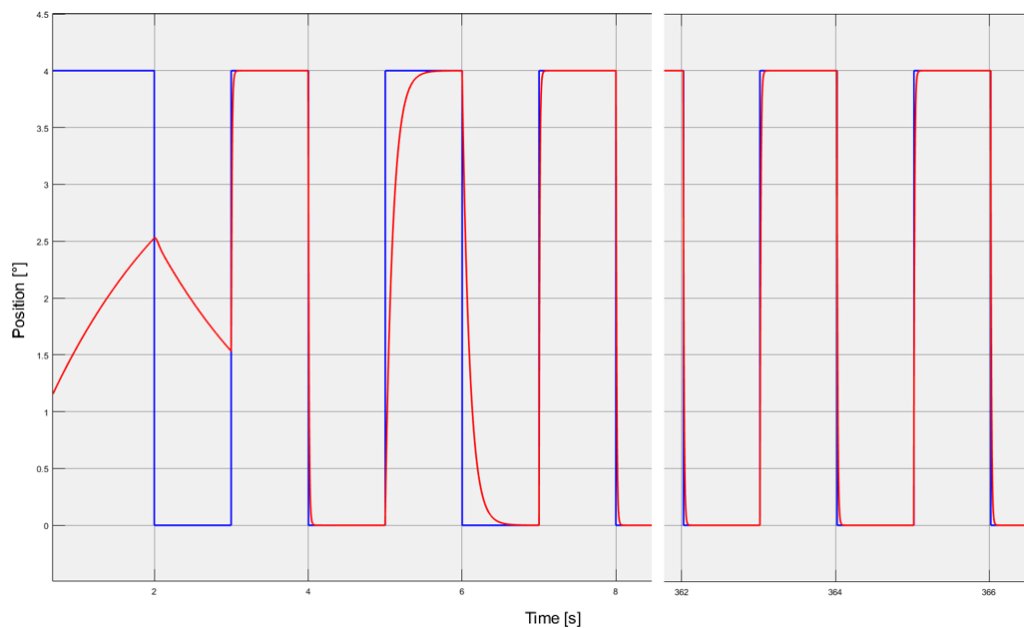
Vysvětlení se nejspíše ukrývá v podmínkách, za jakých jsou optimalizace ukončeny. Gradientní metoda končí, když je pokles cílové funkce už příliš nízký. Simulované žíhání končí až při poklesu teploty pod zadanou mez. Větší vypovídající hodnotu bude mít pohled na vývoje cílových funkcí.

Gradientní metoda se zastavila na hodnotě $Total\ error(CF) = 800$, zatímco metoda simulovaného žihání na hodnotě $Total\ error(CF) = 400$. Tento rozdíl se může zdát značný, ale při uvážení výchozích hodnot cílové funkce to velký rozdíl není.



Obrázek 36 - Pokles cílové funkce při simulační optimalizaci dvou parametrů

U simulovaného žihání optimalizované parametry narostly na výrazně vyšší hodnoty, ale vliv na kvalitu řízení to příliš nemá. Otázkou je, zda by takto vysoké hodnoty mohly být aplikovatelné při řízení reálného motoru.



Obrázek 37 - Naladění polohové smyčky při sim. optimalizaci dvou parametrů

2.5.4 Optimalizace více parametrů

Posledním krokem je rozšíření optimalizace na všechny tři regulátory. Opět není nutné dělat velké úpravy modelu. Stačí zaměnit dosud konstantní vstupy do regulátorů za hodnoty vypočtené ve stavovém automatu. Do toho je třeba dopsat část kódu pro výpočet K_v *current*, T_n *velocity* a T_n *current*.

Meze pro parametry K_v *position* a K_v *velocity* zůstanou stejné jako při simulaci optimalizace dvou parametrů. Ostatní meze jsou nastavené následovně.

$$K_v \text{ current} \in \langle 0.1, 100 \rangle$$

$$T_n \text{ velocity} \in \langle 0.1, 100 \rangle$$

$$T_n \text{ current} \in \langle 0.0005, 0.5 \rangle$$

Výchozí parametry jsou stejné jako pro simulaci optimalizace 5 parametrů u gradientní metody. U metody simulovaného žíhání ovšem výchozí parametry nemají takový vliv.

Výchozí parametry	Výchozí hodnoty parametrů
<i>Teplota</i>	10
<i>Cooling rate</i>	0.7
K_v <i>position</i>	3
K_v <i>velocity</i>	0.001
K_v <i>current</i>	14
T_n <i>velocity</i>	20
T_n <i>current</i>	0.003

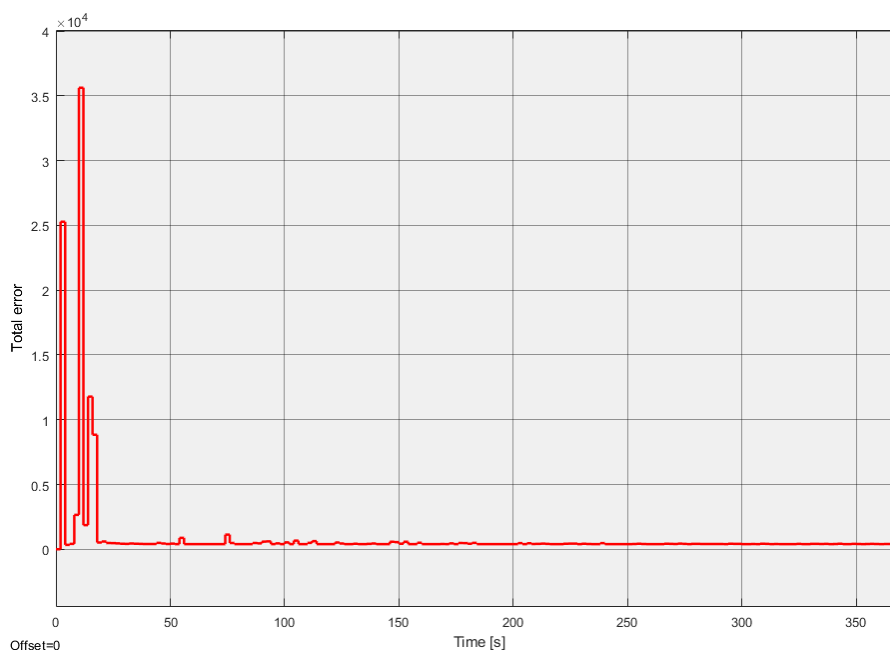
Tabulka 9 - Výchozí nastavení optimalizace 5 parametrů

Získané hodnoty byly opět značně odlišné než u gradientní optimalizace. Větší vypovídající hodnotu bude mít pohled na vývoj cílové funkce.

Parametry regulátorů	Gradientní metoda	Simulované žíhání
K_v <i>position</i>	38	100
K_v <i>velocity</i>	0.035	0.5
K_v <i>current</i>	49	100
T_n <i>velocity</i>	14	100
T_n <i>current</i>	0.02	0.5

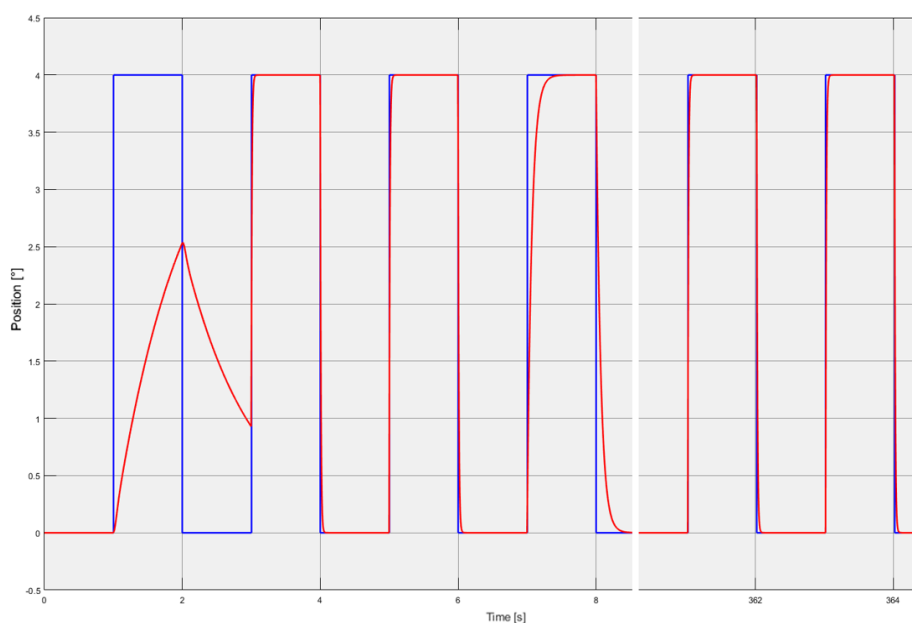
Tabulka 10 - Optimalizované parametry regulátorů

Cílová funkce gradientní metody se ustálila na $Total\ error(CF) = 1200$.
 S algoritmem simulovaného žíhání jsme se dostali na $Total\ error(CF) = 420$.



Obrázek 38 - Pokles cílové funkce při simulační optimalizaci pěti parametrů

Výsledky jsou podobné jako u optimalizace dvou parametrů. Rozdíl cílových funkcí vzhledem k jejich počátečním hodnotám je nepatrný. Stejně tak rozdíl v kvalitě řízení není příliš velký, oproti řízení získaného pomocí gradientní optimalizace, viz obrázek 30 a obrázek 39.



Obrázek 39 - Naladění polohové smyčky při sim. optimalizaci pěti parametrů

3 Identifikace parametrů motoru

Tato kapitola se věnuje procesu identifikace jednotlivých parametrů motoru použitého při experimentálním ověření funkčnosti optimalizačních algoritmů. Tyto parametry se použijí při nastavování měniče a jako konstanty v simulačním modelu.

3.1 Použitý motor

Nejprve je potřeba se podívat na typ použitého motoru. Jedná se o AC synchronní servomotor s permanentními magnety. Výrobce motoru je americká firma Baldor, jenž v současné době spadá do skupiny firem ABB.

Motor nese konkrétně označení BALDOR BSM1R-702/12112302. Tento typ motoru se dle dostupných informací používá v robotech a je pravděpodobné, že i námi použitý motor byl z jednoho takového robota vyjmut. Tento původ může působit problémy při získávání katalogů a dokumentace k motoru, protože se zřejmě nejedná o volně prodávaný model.

Motor se připojuje pomocí jednoho 12-pinového konektoru. Ten obsahuje piny pro napájení a piny výstupů z resolveru. Co se týká napájení tak se jedná o třífázový motor s dvěma pól páry.



Obrázek 40 - Fotka použitého motoru Baldor

3.2 Odpor statorového vinutí

Pro větší věrohodnost simulačního modelu, a také pro správné nastavení frekvenčního měniče je potřeba naměřit hodnotu odporu statorového vinutí. K měření odporu bude použit multimetr Faithful FT-3900. Předpokládáme hodnotu odporu v řádu jednotek Ω , podle toho bude nastaven měřicí rozsah na multimetru. Měřit se bude na napájecích pinech konektoru umístěného v zadní části motoru.



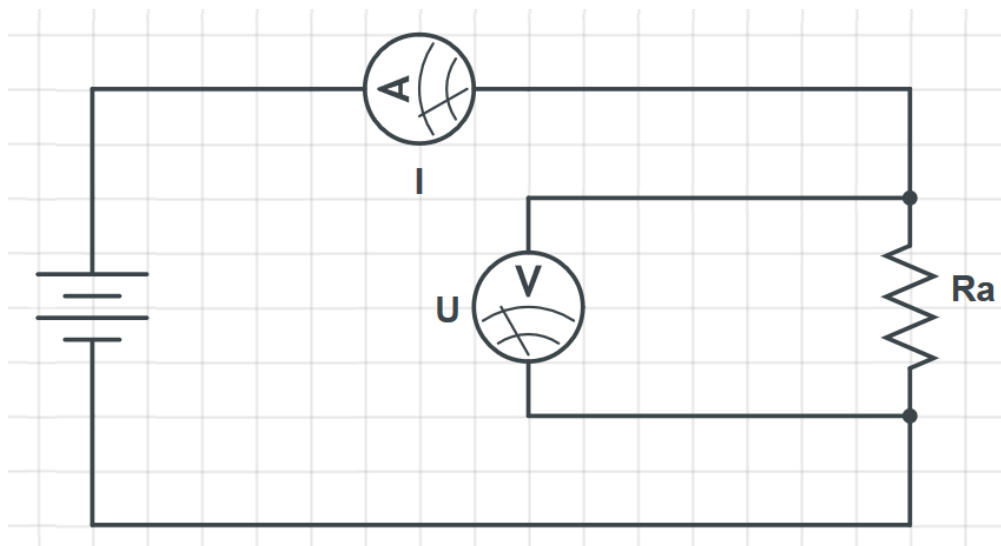
Obrázek 41 - Napájecí piny motoru

Námi naměřený odpor statorového vinutí je $R_a = 12,3[\Omega]$. Problematický je nejmenší možný rozsah multimetru (0 – 200)[Ω]. Chyba měření, vzhledem k rozdílu mezi velikostí rozsahu a naměřenou hodnotou odporu, může být signifikantní.

Další možná chyba může vyplývat přímo z principu odporového měření na multimetru. Číslkové multimetry měří úbytek proudu na rezistoru při průchodu známého proudu. Následně dopočítávají měřenou hodnotu proudu z Ohmova zákona, viz rovnice (39). Problémem je, že máme pouze dva vodiče. Těmi při měření odporu prochází testovací proud, který v kombinaci s vnitřními přechodovými odpory vodičů sám o sobě vyvolává úbytek napětí. Ten spolu s chybou danou rozsahem měření může, u takto malého měřeného odporu, znamenat poměrně velký odklon mezi reálnou a naměřenou hodnotou. [19]

$$R_a = \frac{\Delta U}{I} \quad (39)$$

Z vyjmenovaných důvodů bude jistější měření znovu opakovat za použití multimetru DMM HP 34401A. Výhodou je, že tento multimetr má oddělené proudové a napět'ové svorky. To nám umožní použít tzv. čtyřvodičovou Kelvinovu měřicí metodu.



Obrázek 42 - Schéma Kelvinovy metody měření odporu

Tato metoda se používá pro měření velmi malých odporů, kdy se snažíme zamezit vlivu odporu přípojných drátů a svorek. Díky rozdělení proudové a napět'ové smyčky se nám vnitřní odpor vodičů v proudové smyčce nepromítne do měření. V napět'ové smyčce je, díky velkému odporu voltmetru, minimální proud a dochází k minimálnímu úbytku napětí na vodičích. [20]

Multimetrem DMM HP 34401A jsme, za pomoci čtyřvodičové Kelvinovy měřicí metody, naměřili odpor statorového vinutí $R_a = 4.1 \text{ } [\Omega]$. Vidíme, že je poloviční oproti původní naměřené hodnotě. Z vyjmenovaných důvodů budeme měření Kelvinovou metodou považovat za spolehlivější.

3.3 Indukčnost statorového vinutí

Druhým parametrem elektrické části modelu motoru je indukčnost statorového vinutí. Indukčnost je zjednodušeně schopnost vinutí generovat magnetické pole a v obvodech se střídavými proudy významně promlouvá do fázového zpoždění mezi napětím a proudem.

Měření provedeme ručním LCR metrem ELC-132A/ELC-133A. Tyto přístroje mohou fungovat na dvou principech. První je, když se za pomoci průchodu střídavého napětí měřenou součástkou analyzují její vlastnosti. Přístroj snímá napětí a proud, z těch následně dopočte impedanci. Z impedance poté může dopočíst indukčnost, odpor nebo kapacitu komponentu. Druhým používaným způsobem je měření pomocí vyvažování můstku. Můstek se balancuje do chvíle, kdy měřičem nepřestane protékat proud, poté je také odečtena impedance. [21]

Indukčnost se bude měřit na stejných pinech konektoru, jako byl měřen odpor statorového vinutí. Naměřená indukčnost vinutí byla $L_q = 0.006 [H]$.

3.4 Identifikace mechanických veličin motoru

V mechanické části modelu figurují dva parametry, setrvačnost a tření působící na rotor. Parametr setrvačnosti motoru se zadává i do softwaru TwinCAT a slouží k dosažení kvalitnějšího řízení. Protože se k motoru nepodařilo dohledat technické parametry určené výrobcem, je potřeba si poradit jinak.

V katalozích výrobců se často ani hodnoty třecích koeficientů jednotlivých motorů nevyskytují. Je tomu tak proto, že setrvačnost má na chování motoru výrazně vyšší vliv než tření. Z těchto důvodů ho budeme zanedbávat, budeme ho tedy považovat za nulový.

Setrvačnost rotoru motoru se v katalozích běžně vyskytuje. V našem případě, kdy nemáme technické parametry k dispozici, by šla ověřit experimentálně pomocí zkoumání dobehových charakteristik. My si vystačíme s odhadem stanoveným na základě podobnosti našeho motoru s motorem, k němuž je dostupná dokumentace. Důležitá je podobnost v rozměrech rotoru a stavbě motoru. Odhad na základě podobnosti činí $J = 0,00008 [kg \cdot m^2]$.

4 Příprava experimentu

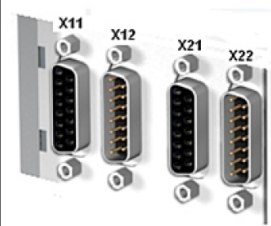
Tato kapitola se bude zabývat hardwarovou implementací simulačně odzkoušených optimalizačních algoritmů. Bude popsán způsob propojení motoru a měniče. Dále následuje popis propojení vývojového prostředí Simulink s prostředím TwinCAT. Nakonec bude proveden testovací experiment, který ověří funkčnost propojených softwarů.

4.1 Propojení motoru a frekvenčního měniče

Nejprve bude zprovozněno propojení mezi počítačem a frekvenčním měničem Beckhoff AX5206. Na počítači za tímto účelem musí být nainstalován Microsoft Visual s integrovaným modulem softwaru TwinCAT. Měnič komunikuje s počítačem prostřednictvím komunikačního rozhraní EtherCat. Po fyzickém propojení počítače s měničem, se nechají v TwinCATu vyhledat nová zařízení a měnič by měl být připojen.

Motor bude ovládán z měniče prostřednictvím rozhraní SERCOS. Jedná se o sériový real-time komunikační systém založený na komunikaci přes průmyslový Ethernet. Tento systém je jeden z nejpoužívanějšího k řízení průmyslových aplikací a digitálních pohonů. [22]

K měniči bude motor připojen přes konektor v zadní části, viz obrázek 41. prostřední silnější piny konektoru jsou napájecí a krajní piny umožňují připojit resolver. Resolver je zařízení sloužící k získávání informace o aktuální poloze motoru. K připojení signálu s resolveru slouží konektor X12.



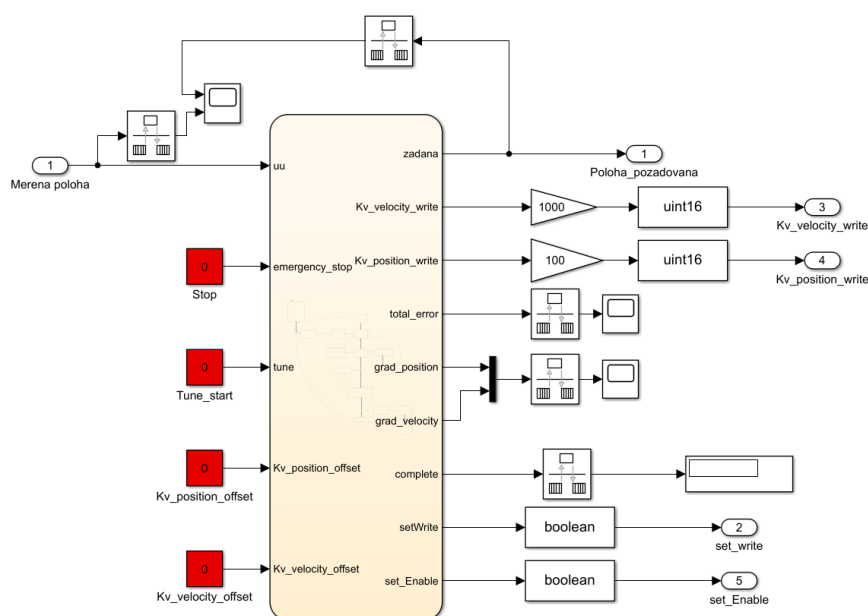
Pin	Resolver
1	Temperature (only PTC, Klixon or bimetal!). Switching point: 1300 Ω ± 3%
2	AGND
3	COS - (S3)
4	SIN - (S4)
5	REF - (R2)
6	n.c.
7	n.c.
8	n.c.
9	Temp_GND
10	COS + (S1)
11	SIN + (S2)
12	REF + (R1)
13	n.c.
14	n.c.
15	n.c.

Obrázek 43 - Návod k zapojení resolveru [23]

4.2 Implementační verze modelu

Motor bude řízen upravenou simulační verzí modelu vytvořeného v prostředí Simulink. Propojení Simulinku a TwinCATu zajišťuje zásuvný modul „TE1400 | TC3 Target for MATLAB®/Simulink®“. Pomocí modulu se může model zkompileovat v Simulinku do jazyka C++ a přenést do softwaru TwinCAT.

V simulačních modelech, které budeme chtít experimentálně otestovat je potřeba provést řadu změn. Z modelu budou odstraněny bloky simulující model motoru a bloky regulátorů. Zůstane pouze stavový automat s optimalizačním algoritmem. Ze stavového automatu budou vyvedeny výstupy jednotlivých optimalizovaných parametrů, které poté budou propojeny s vytvořeným PLC programem v TwinCATu.



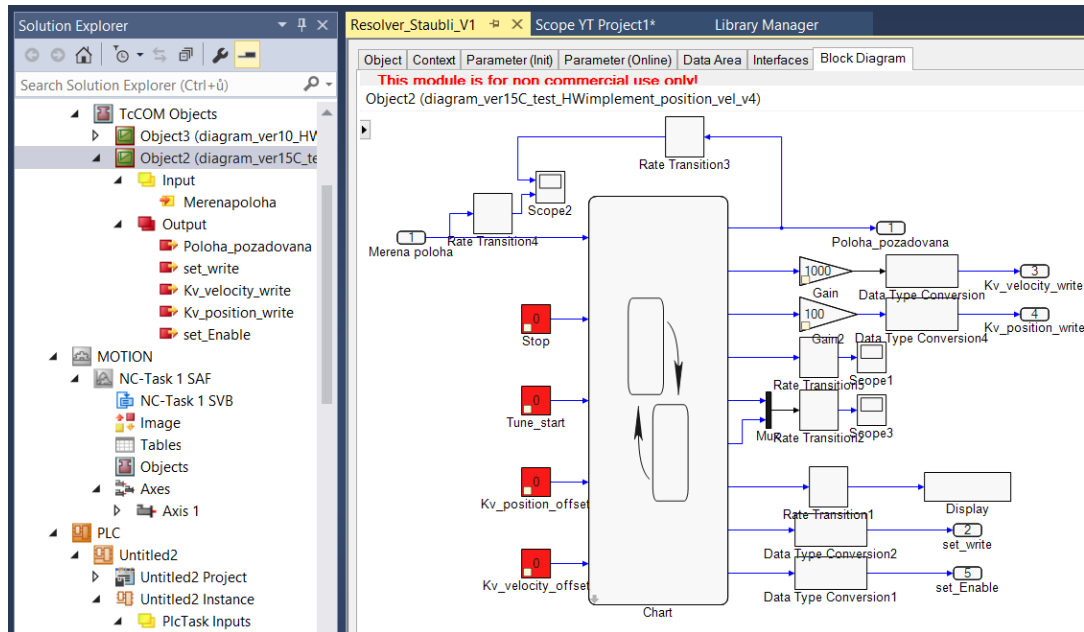
Obrázek 44 - Schéma implementačního modelu gradientní metody

Nově budou muset být vyvedeny i proměnné *set Write* a *set Enable*. Slouží k propsání parametrů do měniče a k inicializaci motoru. Přibyly také bloky upravující datový typy proměnných na typ akceptován řídicím softwarem měniče. Parametry $K_v position$ a $K_v velocity$ musely být ze stejných důvodů přeškálovány.

Mezi vstupy přibyly *Merena poloha*, ta je stanovována na základě dat z resolveru. Vstupy $K_v position offset$ a $K_v velocity offset$ lze měnit výchozí body optimalizace a slouží taky k počátečnímu otestování přepisu parametrů v měniči. Nastavením konstanty *Tune start* na hodnotu 1 se spustí celý proces optimalizace.

4.3 Nahrání modelu do softwaru TwinCAT

Po úspěšné kompilaci může být implementační model nahrán do prostředí TwinCAT. Tam se přidá jako tzv. „TcCOM object“. Strukturu objektu lze zobrazit v samostatném okně. V tomto okně poté mohou být monitorovány hodnoty jednotlivých proměnných, přepisovány konstantní vstupy a nahlíženo do grafů.



Obrázek 45 - TcCOM Object

U každého objektu jsou nahrané vstupy a výstupy, které se následně propojí s vytvořeným PLC programem. Tento program umožní číst a přepisovat proměnné v softwaru měniče. Měníč pak bude nastavován přímo z řídicího modelu v TwinCATu nebo v externím módu ze Simulinku.

```
70 WriteValue := Writen_value;
71 ReadValue := Read_value;
72
73 IF Write THEN
74   Idn_write := S_0_IDN + 104;
75   fbWrite(
76     Axis      := Axis1,
77     Idn       := Idn_write,
78     Element   := 16#40,
79     pSrcBuf   := ADR(WriteValue),
80     BufLen    := SIZEOF(WriteValue),
81     Password  := Password,
82     Execute   := TRUE,
83     Timeout   := DEFAULTT_ADS_TIMEOUT,
84   );
85   IF NOT fbWrite.Busy THEN
86     fbWrite(Axis := Axis1, Execute := FALSE);
87     Write := FALSE;
88   END_IF
89 END_IF
```

Obrázek 46 - Ukázka části PLC kódu pro přepis parametru Kv position

4.4 První testovací experiment

Stejně jako v simulační části práce se bude začínat pouze s optimalizací parametru $K_v position$. K prvnímu experimentálnímu pokusu byl vybrán implementační model s gradientní metodou. Velikost jednoho časového kroku modelu je 0.001[s], tato hodnota byla zvolena podle taktu PLC.

První spouštěná optimalizace má především za úkol otestovat řídicí mechanismy modelu. Bude sledováno, zda dojde ke správné inicializaci motoru a zda bude správně zadávána požadovaná trajektorie. Další pozornost bude upřena na schopnost přepisování parametru $K_v position$ v měniči. Posledním bodem je test optimalizačního algoritmu a posouzení jeho funkčnosti.

Výchozí nastavení testovacího experimentu je uvedeno v tabulce 11. Toto nastavení bylo získáno pomocí experimentálního nastavování „drive manageru“ a je to nastavení, které vyžaduje autotuning.

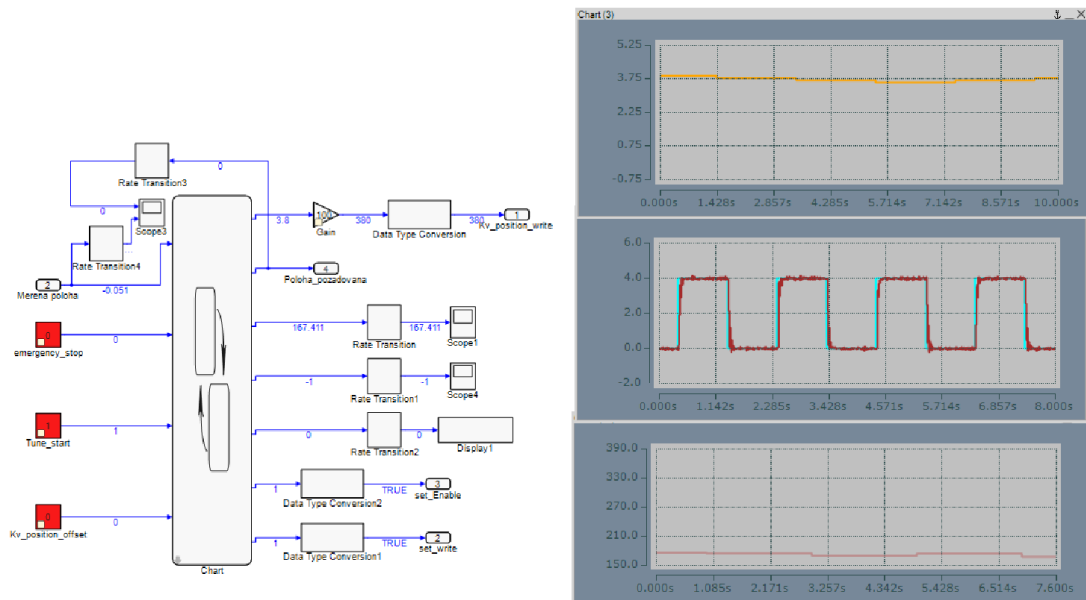
Parametry regulátorů	Hodnoty parametrů	Velikost kroku
$K_v position$	1	0.5
$K_v velocity$	0.008	0
$K_v current$	50	0
$T_n velocity$	2	0
$T_n current$	0.5	0

Tabulka 11 - Výchozí nastavení pokusného experimentu

V případě funkčnosti všech mechanismů bude rovnou přistoupeno k experimentální optimalizaci dvou parametrů, konkrétně $K_v position$ a $K_v velocity$. Z licenčních důvodů bohužel není možné provést optimalizace všech pěti parametrů, proto bude hlavní pozornost obráceno právě na tyto dva parametry které, jak ukázala simulační část práce, mají na výsledné naladění největší vliv.

Každý experiment začíná spuštěním PLC. Funkčnost PLC se pozná podle toho, že u vazeb blokového schématu se objeví aktuální posílané hodnoty. Následuje přepnutí do okna Simulinku, kde musí být u implementačního modelu zahájena synchronizace tlačítkem „Connect to Target“ a následně spuštěna simulace.

V okamžiku spuštění se inicializuje motor a začne následovat trajektorii. Vše lze sledovat na grafech u řídicího modelu. Po přepsání konstanty *Tune start* na hodnotu 1 se začnou měnit hodnoty u proměnné $K_v position write$. Zároveň se musí začít snižovat hodnota cílové funkce a měřená poloha by se měla přibližovat k zadávané.



Obrázek 47 - Výsledek prvního testovacího experimentu (gradientní metoda)

Z výsledku prvního experimentu je patrné, že navržené mechanismy fungují a může být přistoupeno k hlavní části experimentální fáze. Hlavní částí je současná optimalizace $K_v position$ a $K_v velocity$ za pomoci gradientní metody a metody simulovaného žihání.

Za povšimnutí stojí, že bylo dosaženo dobrého naladění už při hodnotě $K_v position = 3.75$. Což je podstatně nižší hodnota než u simulačního modelu, kde byla optimalizace zastavena až na hodnotě $K_v position = 36$. Z toho se dá usuzovat, že chování simulačního modelu není příliš věrné tomu reálnému. Proto nebudou srovnávány výsledky simulací a experimentů. Bude pouze posuzováno, zda je algoritmus schopen minimalizovat cílovou funkci a dosáhnout naladění, při kterém se bude měřená trajektorie shodovat se zadávanou.

5 Experimentální optimalizace dvou parametrů

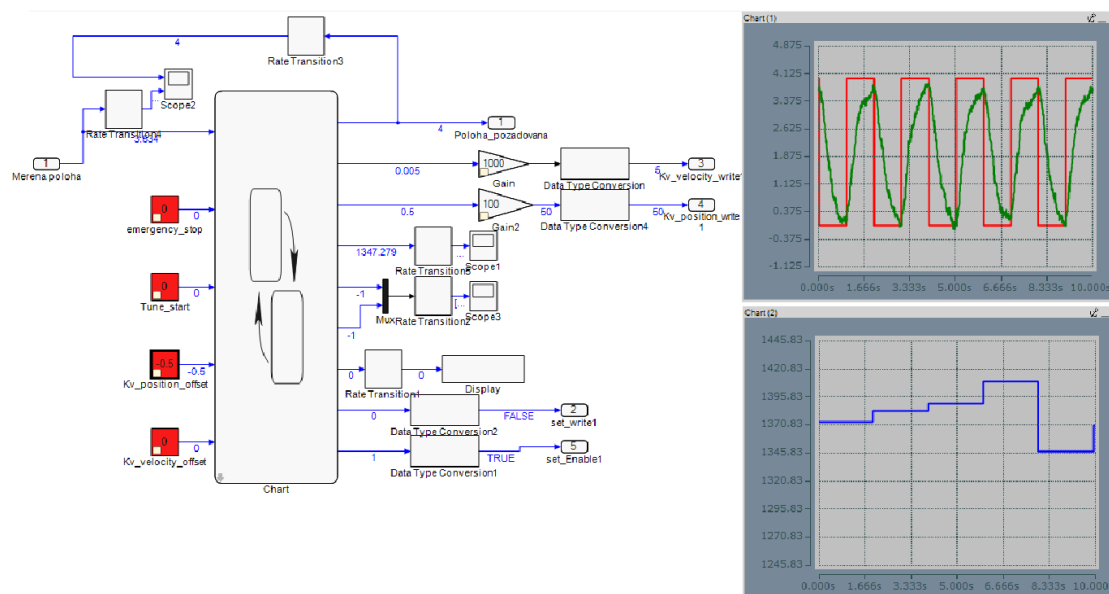
V této kapitole bude posouzeno, zda optimalizační algoritmy dvou proměnných vedou k minimalizaci cílové funkce a zkvalitnění řízení polohy. Obě metody budou mít stejné počáteční parametry. Výchozí nastavení parametrů je uvedeno v tabulce 12.

Parametry regulátorů	Výchozí hodnoty
K_v position	0.5
K_v velocity	0.005
K_v current	50
T_n velocity	2
T_n current	0.5

Tabulka 12 - Výchozí parametry pro experiment dvou proměnných

5.1 Experiment s gradientní metodou

Konstantní výchozí parametry se budou načítat z „Drive manageru“. Parametry k optimalizaci jsou zapsané ve stavovém automatu. Velikosti kroků jednotlivých parametrů jsou $krok\ position = 0.1$ a $krok\ velocity = 0.005$. Velikosti kroků jsou po celou dobu konstantní a s ohledem na výsledek testovacího experimentu jsou zvoleny menší velikosti kroků než v simulační části.



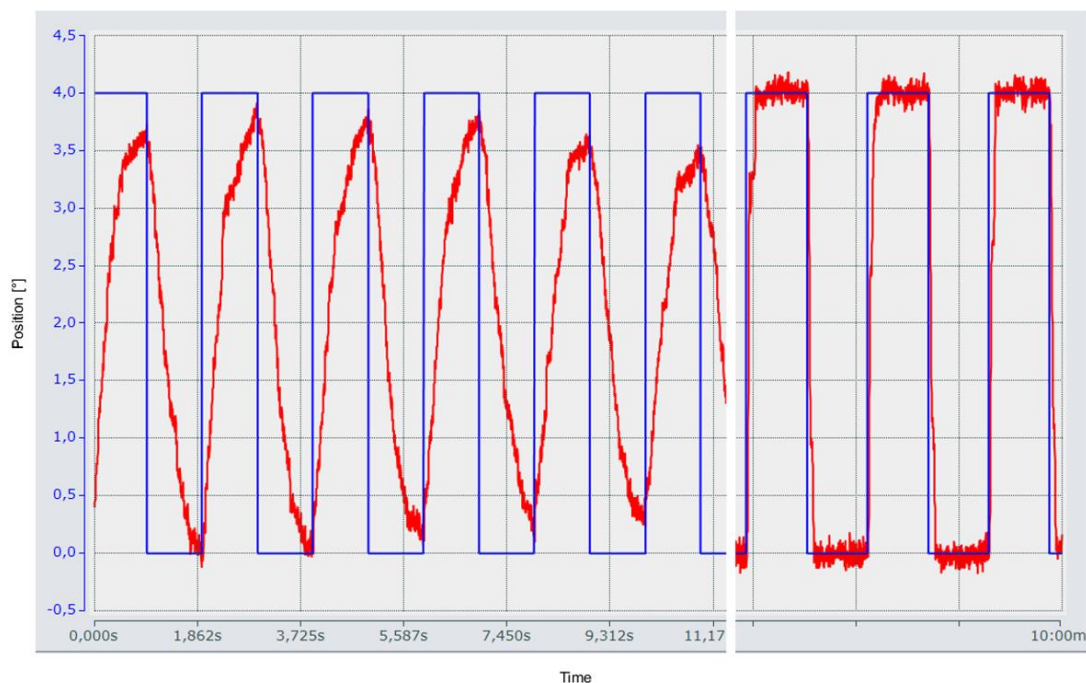
Obrázek 48 - Výchozí stav experimentu před spuštěním autotuningu

Z důvodů přehlednosti nebudou k prezentaci výsledků použity grafy v TcCOM objektu. Naměřená data budou lépe čitelná z grafů TwinCAT Measurement projektu.



Obrázek 49 - Pokles cílové funkce během experimentu

Cílová funkce během experimentu poklesla na $Total\ error(CF) = 275$. Byly nalezeny optimalizované parametry $K_v\ position = 5.89$ a $K_v\ velocity = 0.275$.



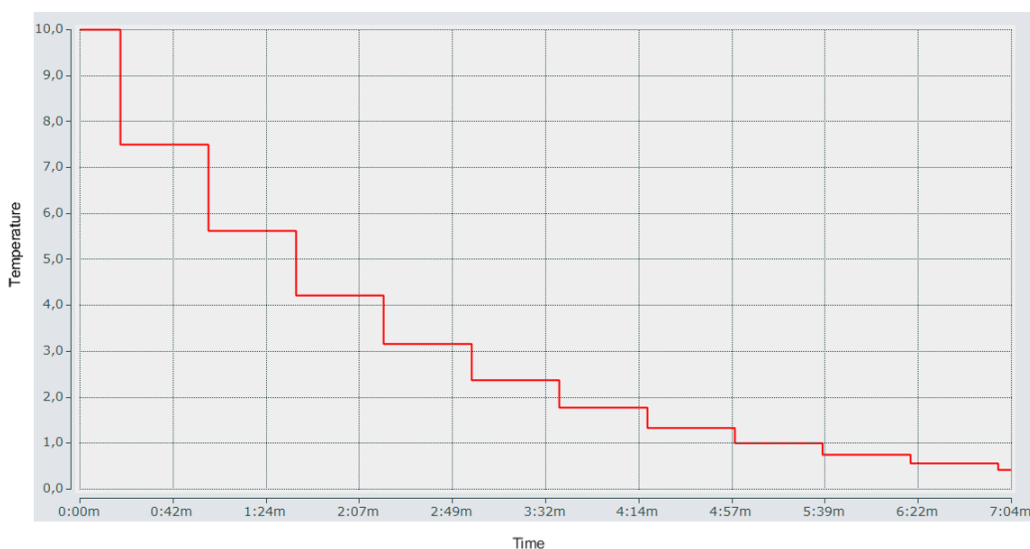
Obrázek 50 - Dosažené naladění polohové smyčky během experimentu

5.2 Experiment se simulovaným žiháním

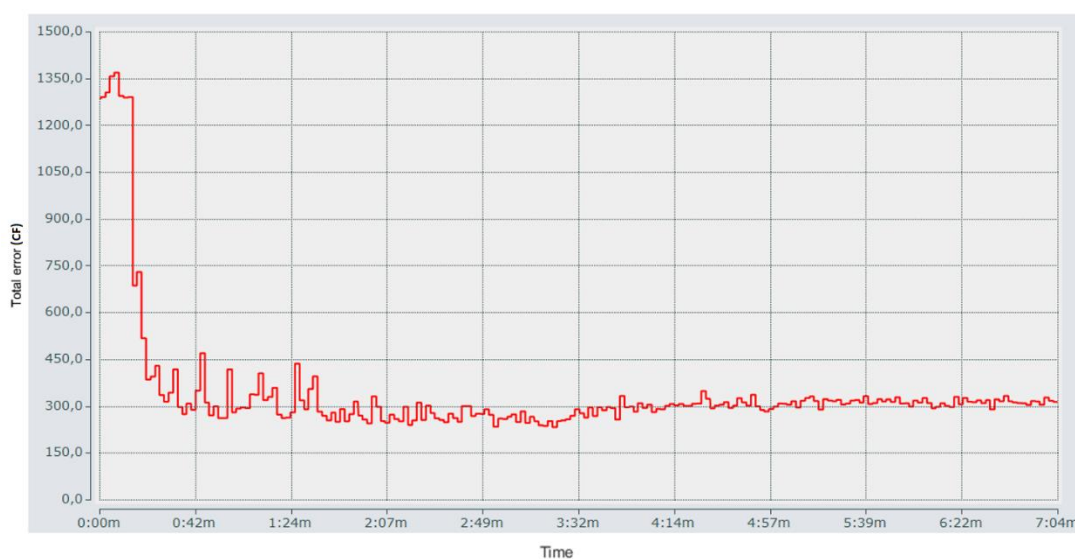
S algoritmem simulovaného žihání budou provedeny dvě experimentální měření. U obou bude výchozí teplota nastavena jako $Teplota = 10$. Lišit se budou v rychlosti ochlazování. U prvního bude rychlost chladnutí $Cooling\ rate = 0.75$. Druhý experiment bude uskutečněn s $Cooling\ rate = 0.9$, bude tedy chladnou pomaleji.

5.2.1 První experiment (C. r. = 0.75)

Výchozí nastavení parametrů regulátorů je stejné jako u experimentu s gradientním algoritmem, viz tabulka 12. Rychlost klesání teploty a dosažená minimalizace cílové funkce je patrná z obrázků 51 a 52.

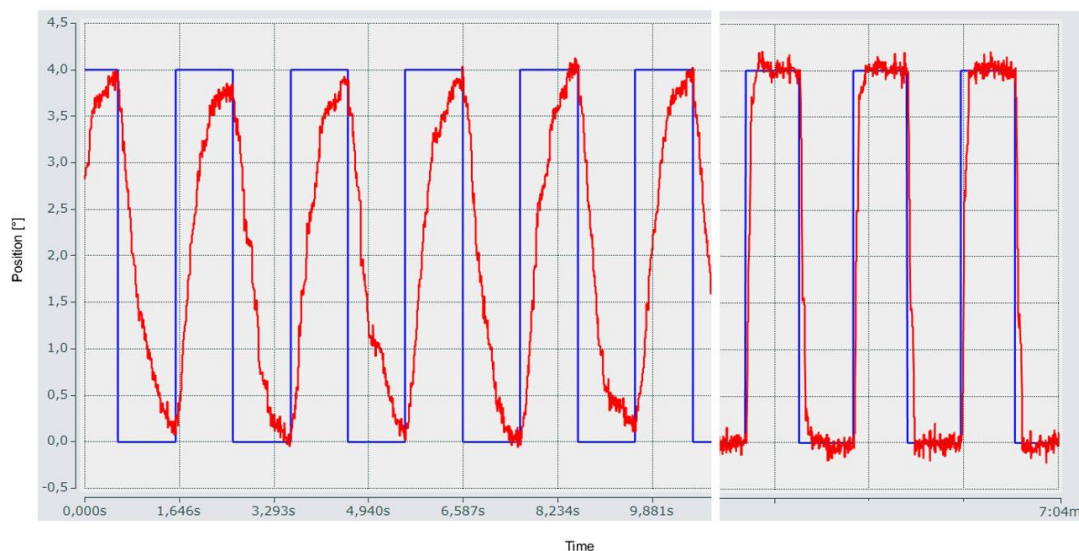


Obrázek 51 - Pokles teploty (C. r. = 0.75)



Obrázek 52 - Minimalizace cílové funkce během experimentu (C. r. = 0.75)

Po poklesu teploty na minimální hodnotu 0.5 se optimalizace zastavila. Během experimentu klesla cílová funkce na hodnotu $Total\ error(CF) = 303$. Byly nalezené optimalizované parametry $K_v\ position = 3.65$ a $K_v\ velocity = 0.105$.

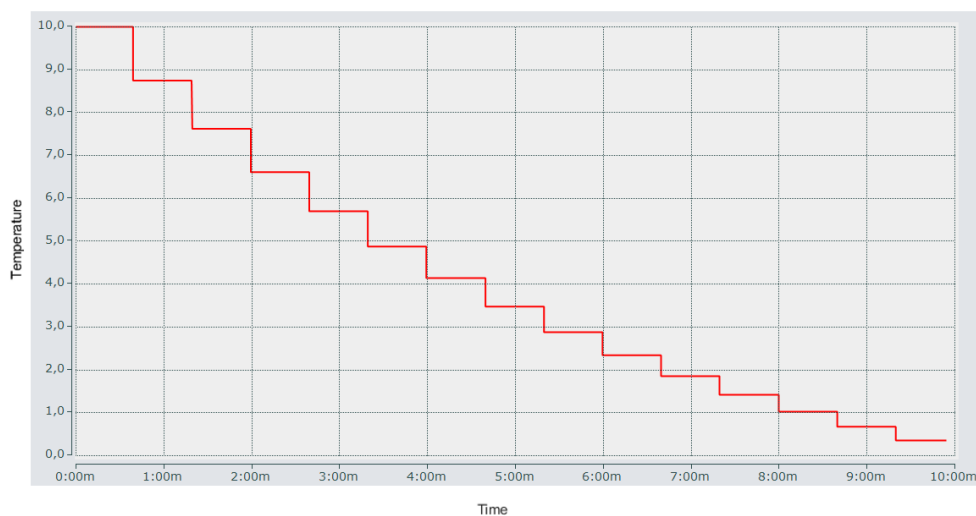


Obrázek 53 - Dosažené naladění během experimentu (C. r. = 0.75)

Z dosaženého naladění je patrné značné zlepšení. Naladění není tak dokonalé jako u gradientní optimalizace, čemuž odpovídají i dosažené hodnoty cílové funkce. Na kvalitě řízení se nejspíše hodně podepsal šum, který je z grafů patrný.

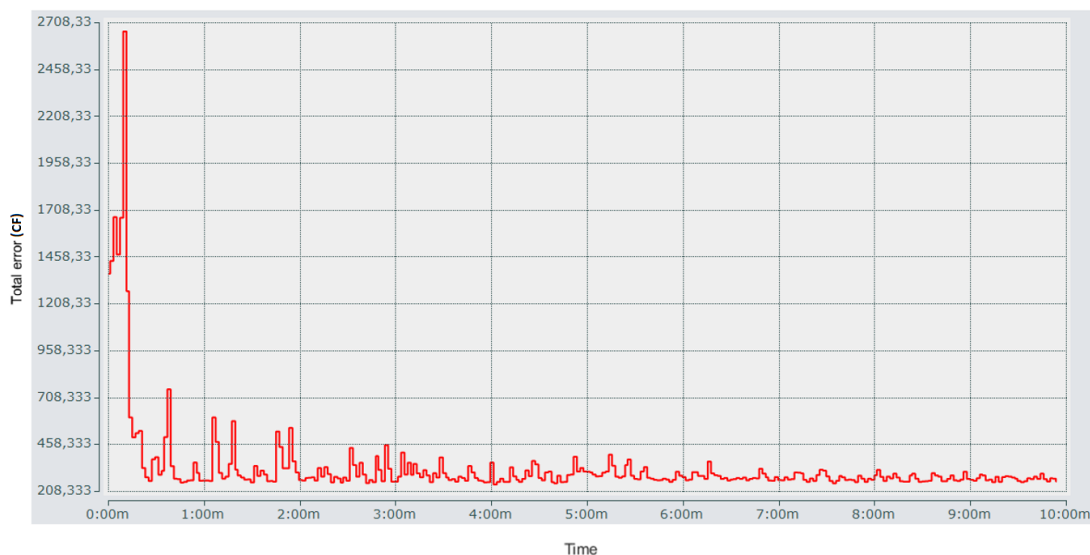
5.2.2 Druhý experiment (C. r. = 0.9)

Při druhém experimentu se zvýší rychlost chladnutí. Optimalizace bude trvat delší dobu, ale zvýší se pravděpodobnost nalezení globálního minima.



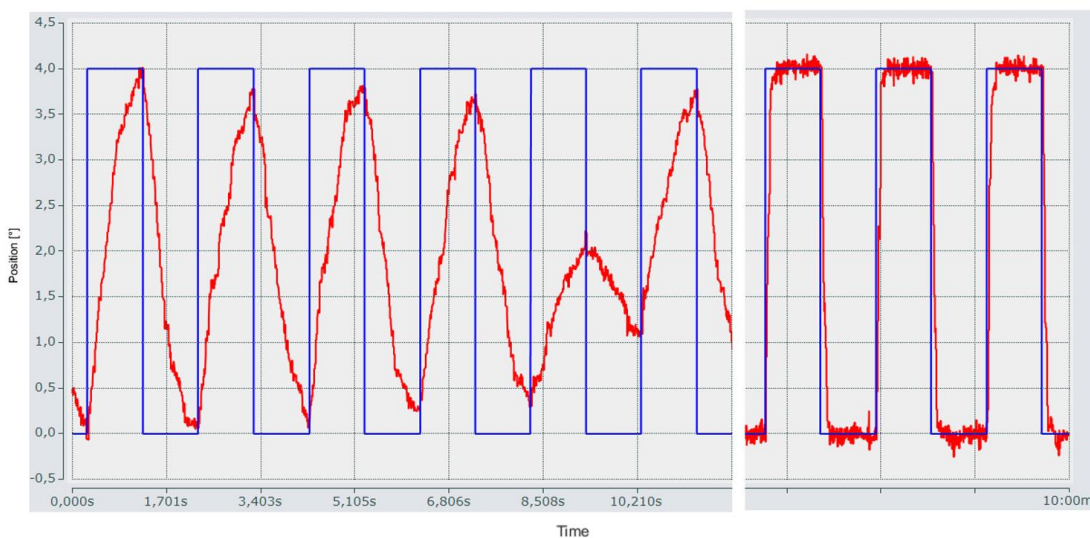
Obrázek 54 - Pokles teploty během experimentu (C. r. = 0.9)

Po dokončení optimalizace byla hodnota cílová funkce $Total\ error(CF) = 262$. Vlivem pravděpodobnosti, na které je metoda simulovaného žíhání založena, bylo optimální řešení nalezeno hned zpočátku a následně se všechny další testované body se pohybovaly pouze v jeho okolí. Byly nalezené optimalizované parametry regulátorů $K_v\ position = 3.87$ a $K_v\ velocity = 0.119$.



Obrázek 55 - Minimalizace cílové funkce během experimentu (C. r. = 0.9)

V naměřených datech polohy je opět patrné značné zlepšení. Znovu se ovšem potýkáme se šumem, jehož vliv na experiment stojí za zvážení.



Obrázek 56 - Dosažené naladění během experimentu (C. r. = 0.9)

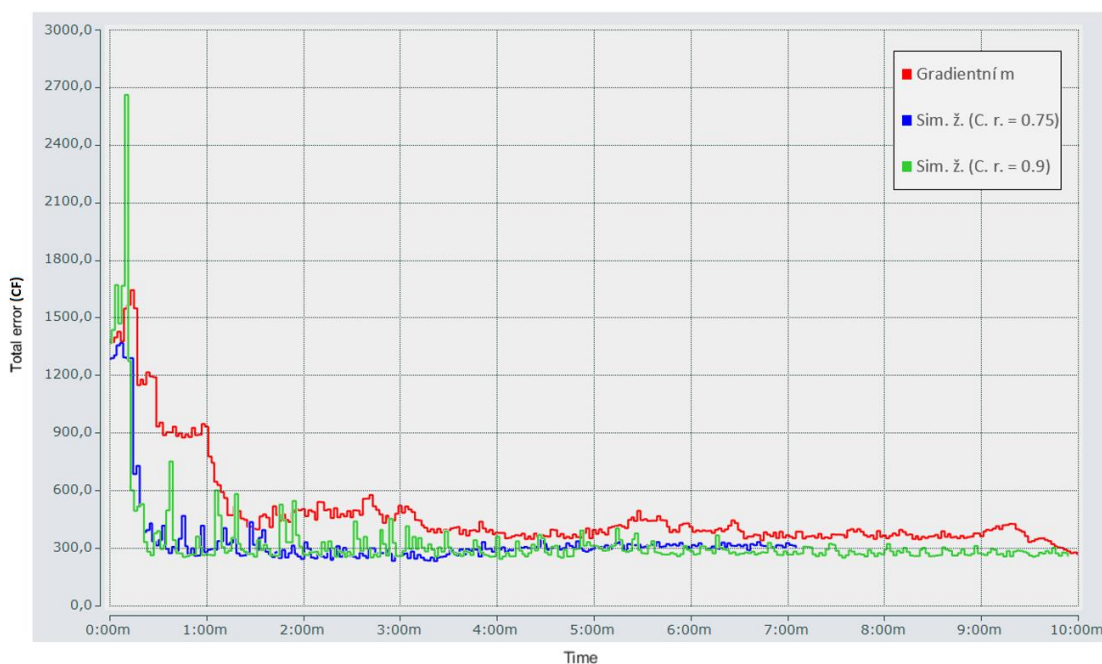
5.3 Vyhodnocení experimentů

Všechny experimenty začínaly ze stejného výchozího nastavení regulátorů a proběhly úspěšně. Výchozí hodnota cílové funkce byla $Total\ error(CF) = 1370$. Pokles cílové funkce u všech metod byl více než čtyřnásobný. Zvýšená přesnost nastavené polohové smyčky poté byla na první pohled patrná. Dosažené minimalizované hodnoty cílové funkce a parametry jsou uvedeny v tabulce 13.

	Gradientní m.	Sim. ž. (C. r. = 0.75)	Sim. ž. (C. r. = 0.9)
$Total\ error(CF)$	275	303	262
$K_v\ position$	5.89	3.65	3.87
$K_v\ velocity$	0.275	0.105	0.119

Tabulka 13 – Nalezené optimalizované veličiny

Metody simulovaného žihání našly podobné nastavení parametrů regulátorů. Gradientní metoda se s parametry regulátorů dostala do vyšších hodnot, ale cílovou funkci už tím více nesnížila. Tyto rozdíly nejspíše budou způsobeny vlivem šumu a také možným pokusům o snížení cílové funkce za cenu překmitů. Podobné chování bylo pozorované i v simulační části práce. U metody simulovaného žihání se potvrdilo, že rychlost chladnutí má vliv na nalezení minima cílové funkce. S pomalejším chladnutím proběhlo více optimalizačních cyklů a byla větší pravděpodobnost na získání lepších výsledků.



Obrázek 57 - Srovnání vývoje cílových funkcí u jednotlivých experimentů

6 Závěr

V práci je popsán vývoj dvou algoritmů autotuningu pro řízení servomotoru Baldor za pomoci měniče Beckhoff AX5206. Algoritmy jsou založené na gradientní metodě a metodě simulovaného žihání. Oba postupy byly nejprve simulačně otestovány na vytvořeném modelu motoru a následně použity k ladění parametrů regulátorů u skutečného motoru.

V simulační i experimentální části bylo dosaženo významného zlepšení funkce regulačních smyček řízení motoru. Oba optimalizační algoritmy dosahovaly srovnatelných výsledků v minimalizaci chyby řízení polohy.

Nalezené optimalizované parametry regulátorů se u jednotlivých algoritmů lišily. Pravděpodobnou příčinou jsou rozdílné podmínky, za jakých byly optimalizační algoritmy ukončovány. Zvláště u gradientní metody docházelo k tomu, že algoritmus se snažil měnit optimalizované parametry i ve chvíli, kdy se již cílová funkce takřka nezlepšovala. Odstranění tohoto jevu a synchronizace výstupních podmínek obou algoritmů představují prostor pro další možné zlepšení dosažených výsledků.

U obou algoritmů se projevovala snaha minimalizovat cílovou funkci za cenu překmitů. Tento jev je nežádoucí a v případných navazujících pracích by měl být eliminován penalizací. Otázkou je také vliv šumu, jenž se objevoval v experimentální části. Tento šum nepříznivě ovlivňoval rychlost minimalizace cílové funkce a jeho odstraněním by mohlo být dosaženo dalšího výrazného zlepšení výsledků.

7 Literatura

- [1] Servo motor. In: *PohonnaTechnika.cz* [online]. [cit. 2019-06-10]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>
- [2] Elektrické servopohony. In: *Elektrické servopohony* [online]. [cit. 2019-06-10]. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=33400
- [3] UHLÍŘ, Ivan. *Elektrické stroje a pohony*. Vyd. 2. přeprac. Praha: ČVUT, 2007. ISBN 978-80-01-03730-0.
- [4] KLÁN, Petr. Moderní metody nastavení pid regulátorů. *Automata* [online]. 2009, 2009(9), 4 [cit. 2019-08-05]. Dostupné z: https://valter.byl.cz/sites/default/files/soubory/moderni_nastav_pid.pdf.
- [5] HLINOVSKÝ, Martin. Regulátory. In: *Regulátory - Regulatory2.pdf* [online]. Praha, 2013 [cit. 2019-08-05]. Dostupné z: <http://rbs.felk.cvut.cz/files/Regulatory2.pdf>
- [6] Odezvy systému. In: *Regulátory - Regulatory2.pdf* [online]. [cit. 2019-06-19]. Dostupné z: <http://rbs.felk.cvut.cz/files/Regulatory2.pdf>
- [7] Číslicová regulace - katedra 352, VŠB - Technická univerzita Ostrava. *Číslicová regulace - katedra 352, VŠB - Technická univerzita Ostrava* [online]. Ostrava, 2008 [cit. 2019-08-05]. Dostupné z: <http://books.fs.vsb.cz/cislicovaregulace/data/kapitola4.html>
- [8] PRŮMYSLOVÉ PID REGULÁTORY: TEORIE PRO PRAXI. *Pragoregula* [online]. 2001, <http://zcu.arcao.com/kky/zky/Prago1.pdf> [cit. 2019-08-06].
- [9] Ziegler-Nicholsova metoda návrhu regulátorů. *Ziegler-Nicholsova metoda návrhu regulátor* [online]. Brno, 2009 [cit. 2019-08-06]. Dostupné z: http://www.uamt.feec.vutbr.cz/~richter/vyuka/0809_BRR1/labs/zn_reg.html.
- cs

- [10] PI controller relay auto-tuning using delay and phase margin in PMSM drives. *Chinese Journal of Aeronautics* [online]. 2014, 2014 [cit. 2019-08-06]. http://www.uamt.feec.vutbr.cz/~richter/vyuka/0809_BRR1/labs/zn_reg.html.
cs
- [11] Experimental Plant Identification by Relay Method. *ASR* [online]. 2005, 2005 [cit.2019-08-06]. Dostupné z: <http://akce.fs.vsb.cz/2005/ASR2005/Proceedings/papers/479.pdf>
- [12] Oscillation curves of relay output and plant output. In: *PI controller relay auto-tuning using delay and phase margin in PMSM drives - ScienceDirect* [online]. *Chinese Journal of Aeronautics*, 2014 [cit. 2019-08-02]. Dostupné z: <https://ars.els-cdn.com/content/image/1-s2.0-S1000936114001733-gr4.jpg>
- [13] PID controller tuning optimization using gradient descent technique for an electro-hydraulic servo system. *Jurnal Teknologi*. 2015, 2015(33), 7.
- [14] LEDERER, Pavel. *Teorie a optimalizace mechanických systémů 2*. Praha: ČVUT, 1988.
- [15] *Simulované žihání jako nástroj k hledání optimálního řešení* [online]. 2011 [cit. 2019-08-06]. Dostupné z: <http://tydenvedy.fjfi.cvut.cz/2011/cd/prispevky/sbpdf/zihani.pdf>
- [16] Výzkumná skupina aplikovaných evolučních algoritmu. *Výzkumná skupina aplikovaných evolučních algoritmu* [online]. 2006 [cit. 2019-08-06]. Dostupné z: <http://www.fit.vutbr.cz/~jarosjir/groups/eva/sa.html.cz>
- [17] Convergence of simulated annealing. In: *Simulated Annealing* [online]. [cit. 2019-08-06]. Dostupné z: <https://www.slideshare.net/idforjoydutta/simulated-annealing-24528483>
- [18] *TwinCAT 3 – Getting started*. Eiserstraße 5 33415 Verl Germany: Beckhoff Automation, 2012.

- [19] AM - laboratorní cvičení č. 3. *FD hlavní stránka | ČVUT v Praze Fakulta dopravní* [online]. 2017 [cit. 2019-08-07]. Dostupné z: <https://www.fd.cvut.cz/personal/xfabera/AM/labcv/labcv3/labcv3.html>
- [20] Kelvin (4-vodičové) měření odporu | dc měřících obvodů - učebnice - 2019. *Elektrikář V Domě A Na Pracovišti* [online]. 2019 [cit. 2019-08-07]. Dostupné z: <https://cs.electronics-council.com/kelvin-resistance-measurement-51851>
- [21] LCR Meter | LCR Bridge | Electronics Notes. *Electronics Notes: reference site for electronics, radio & wireless* [online]. [cit. 2019-08-09]. Dostupné z: <https://www.electronics-notes.com/articles/test-methods/lcr-meter-bridge/primer-basics.php>
- [22] SERCOS: SSI Encoder Gateways and Blended EtherNet/IP Infrastructures | Automation World. *Automation World | Covering the field of industrial automation* [online]. [cit. 2019-08-09]. Dostupné z: <https://www.automationworld.com/article/technologies/networking-connectivity/sercos/sercos-ssi-encoder-gateways-and-blended-ethernet>
- [23] *AX5000 - Startup - ax5000_startup_en.pdf* [online]. 5.9. Germany, 2018 [cit. 2019-08-09]. Dostupné z: https://download.beckhoff.com/download/document/motion/ax5000_startup_en.pdf

8 Seznam obrázků

Obrázek 1 - Schéma kaskádní regulace	10
Obrázek 2 - Odezvy systému na různá zesílení ^[6]	12
Obrázek 3 - Odezva tříparametrového modelu na skok ^[4]	15
Obrázek 4 - Autotuning pomocí relé.....	17
Obrázek 5 - Záznam výstupů relé a regulované soustavy ^[12]	18
Obrázek 6 - Snižování hodnoty cílové funkce během chladnutí ^[17]	21
Obrázek 7 - Schématický popis algoritmu simulovaného žihání.....	22
Obrázek 8 - Schéma propojení použitých vývojových prostředí ^[18]	23
Obrázek 9 - Výchozí řídicí schéma měniče	24
Obrázek 10 - Kaskádní regulace použitá v AX5206.....	24
Obrázek 11 - Schéma rychlostní regulační smyčky v měniči AX5206	25
Obrázek 12 - Model motoru	27
Obrázek 13 - Blok PI regulátoru	28
Obrázek 14 - Vnitřní struktura PI regulátoru	29
Obrázek 15 - Model motoru s řízením	29
Obrázek 16 - Zadávání testovacího signálu	30
Obrázek 17 - Test modelu	30
Obrázek 18 - Gradientní optimalizace pro jeden parametr	31
Obrázek 19 - Stavový diagram gradientního algoritmu.....	32
Obrázek 20 - Logika gradientního algoritmu.....	33
Obrázek 21 - Minimalizace testovací funkce.....	35
Obrázek 22 - Nalezení hodnoty parametru x	35
Obrázek 23 - Pokles cílové funkce během simulační optimalizace.....	36
Obrázek 24 - Dosažené naladění polohové smyčky při simulační optimalizaci.....	37
Obrázek 25 - Dosažené naladění polohové smyčky během 2. simulace.....	38
Obrázek 26 - Vývoj cílové funkce během 2. simulace	38
Obrázek 27 - Pokles cílové funkce při simulační optimalizaci dvou parametrů.....	40
Obrázek 28 - Naladění polohové smyčky při sim. optimalizaci dvou parametrů.....	40
Obrázek 29 - Pokles cílové funkce při simulační optimalizaci pěti parametrů.....	41
Obrázek 30 - Naladění polohové smyčky při sim. optimalizaci pěti parametrů.....	42
Obrázek 31 - Iterační smyčky algoritmu simulovaného žihání	43
Obrázek 32 - Blok stavu "Compare".....	44

Obrázek 33 - Pokles teploty v průběhu simulační optimalizace	45
Obrázek 34 - Pokles cílové funkce během simulační optimalizace.....	46
Obrázek 35 - Dosažené naladění polohové smyčky při simulační optimalizaci.....	46
Obrázek 36 - Pokles cílové funkce při simulační optimalizaci dvou parametrů.....	48
Obrázek 37 - Naladění polohové smyčky při sim. optimalizaci dvou parametrů.....	48
Obrázek 38 - Pokles cílové funkce při simulační optimalizaci pěti parametrů.....	50
Obrázek 39 - Naladění polohové smyčky při sim. optimalizaci pěti parametrů.....	50
Obrázek 40 - Fotka použitého motoru Baldor.....	51
Obrázek 41 - Napájecí piny motoru	52
Obrázek 42 - Schéma Kelvinovy metody měření odporu.....	53
Obrázek 43 - Návod k zapojení resolveru ^[23]	55
Obrázek 44 - Schéma implementačního modelu gradientní metody	56
Obrázek 45 - TcCOM Object.....	57
Obrázek 46 - Ukázka části PLC kódu pro přepis parametru Kv position.....	57
Obrázek 47 - Výsledek prvního testovacího experimentu (gradientní metoda)	59
Obrázek 48 - Výchozí stav experimentu před spuštěním autotuningu	60
Obrázek 49 - Pokles cílové funkce během experimentu.....	61
Obrázek 50 - Dosažené naladění polohové smyčky během experimentu.....	61
Obrázek 51 - Pokles teploty (C. r. = 0.75)	62
Obrázek 52 - Minimalizace cílové funkce během experimentu (C. r. = 0.75).....	62
Obrázek 53 - Dosažené naladění během experimentu (C. r. = 0.75)	63
Obrázek 54 - Pokles teploty během experimentu (C. r. = 0.9)	63
Obrázek 55 - Minimalizace cílové funkce během experimentu (C. r. = 0.9).....	64
Obrázek 56 - Dosažené naladění během experimentu (C. r. = 0.9)	64
Obrázek 57 - Srovnání vývoje cílových funkcí u jednotlivých experimentů.....	65