

Návrh self-adaptive regulátoru pro dávkování materiálu v rámci směšovací linky

Bc. Petr Botek

Diplomová práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Petr Botek**
Osobní číslo: **A22327**
Studijní program: **N0714A150006 Automatické řízení a informatika v průmyslu 4.0**
Forma studia: **Prezenční**
Téma práce: **Návrh self-adaptive regulátoru pro dávkování materiálu v rámci směšovací linky**
Téma práce anglicky: **Design of a Self-Adaptive Controller for Material Dosing Within a Mixing Line**

Zásady pro vypracování

1. Analyzujte současný stav řízení dávkování sypkých materiálů. Definujte nedostatky v kvalitě řízení dosažené použitým regulátorem s ohledem na požadavky na řízení.
2. Identifikujte matematický model řízeného systému vhodný pro návrh regulátoru.
3. Navrhněte vhodný způsob řízení, včetně stanovení struktury zvoleného regulátoru a výpočtu jeho parametrů.
4. Navržený regulátor simulačně ověřte v prostředí MATLAB/SIMULINK.
5. Implementujte regulátor do řídicího prostředí zařízení (PLC) pro řízení míchací linky.
6. Regulátor ověřte na reálném zařízení a porovnejte dosaženou kvalitu řízení s předchozím způsobem regulace.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BOBÁL, Vladimír. Adaptivní a prediktivní řízení, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2007, ISBN 978-80-7318-662-3.
2. MIKLEŠ, Ján a FIKAR, Miroslav. Process Modelling, Optimisation and Control, Berlin: Springer-Verlag, 2008, ISBN 978-3642091124.
3. MIKLEŠ, Ján a FIKAR, Miroslav. Modelovanie, Identifikácia a riadenie procesov II. STU Bratislava, 2004, ISBN 80-227-2134-4.
4. BOBÁL, Vladimír, BOHM Josef, FESSL Jaromír, MACHÁČEK Jiří. Digital Self-Tuning Controllers, Springer-Verlag, London, 2005, ISBN: 1-85233-980-2.
5. BOBÁL Vladimír, BOHM Josef, PROKOP Roman, Fessl Jaromír. Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace, VUT v Brně, 1999, ISBN 80-214-1299-2.
6. BOBÁL, Vladimír. Identifikace Systémů , Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2007, ISBN 978-80-7318-662-3.

Vedoucí diplomové práce: **doc. Ing. Marek Kubalčík, Ph.D.**
Ústav řízení procesů

Datum zadání diplomové práce: **8. prosince 2023**

Termín odevzdání diplomové práce: **27. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Petr Botek v.r.
.....
podpis studenta

ABSTRAKT

Cílem diplomové práce je návrh self-adaptive regulátoru pro řízení výšky hladiny materiálu při navážení na váhu míchací linky. Bylo aplikováno adaptivní prediktivní řízení s průběžnou identifikací parametrů modelu procesu. Navržené řízení bylo implementováno a testováno v reálném provozu linky a bylo provedeno srovnání s dosavadním způsobem řízení.

Klíčová slova: self-adaptive, míchací linka, ARX model, rekurzivní metoda nejmenších čtverců, prediktivní řízení

ABSTRACT

The aim of the master thesis is to design a self-adaptive controller for controlling the level of material when weighed on the weight of the mixing line. Adaptive predictive control was applied with continuous identification of process model parameters. The proposed control system was implemented and tested in the real operation of the line and a comparison with the existing control method was made.

Keywords: self-adaptive, mixing line, ARX model, recursive least square method, predictive control

Poděkování

Chtěl bych velmi poděkovat vedoucímu práce doc. Marku Kubalčíkovi za jeho skvělé vedení a dohled nad mojí diplomovou prací. Dále bych chtěl poděkovat Ing. Jakubu Korábovi za zprostředkování informací a kontaktů uvnitř Continentalu. Na závěr bych chtěl také poděkovat svým rodičům, kteří mi byli oporou po celou dobu mého studia.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 MÍCHACÍ LINKY V ZÁVODĚ CONTINENTAL	10
2 POUŽITÝ ŘÍDICÍ A MĚŘICÍ SYSTÉM	11
2.1 SIMATIC	11
2.2 SIMATIC PLC	11
2.3 PROCESS CONTROL SYSTEM 7 – PCS7	12
2.4 SIWAREX – VÁŽICÍ ELEKTRONIKA.....	15
3 SOFTWARE PRO MĚŘENÍ DAT	17
3.1 IBA AG	17
3.2 IBAPDA – PROCESS DATA ACQUISITION.....	17
4 ARX MODEL	19
4.1 PRINCIP METODY NEJMENŠÍCH ČTVERCŮ	21
4.2 MODIFIKACE PRO SYSTÉMY S DOPRAVNÍM ZPOŽDĚNÍM:	22
5 REKURZIVNÍ IDENTIFIKAČNÍ ALGORITMY	24
5.1 ALGORITMUS PRO REKURZIVNÍ MNČ.....	27
5.2 MODIFIKACE PRO SYSTÉMY S DOPRAVNÍM ZPOŽDĚNÍM.....	28
6 PREDIKTIVNÍ ŘÍZENÍ	29
6.1 VÝPOČET OPTIMÁLNÍHO ŘÍZENÍ - MINIMALIZACE KVADRATICKÉHO KRITÉRIA	31
6.2 VÝPOČET PREDIKTORU	32
6.3 MODIFIKACE PRO DOPRAVNÍ ZPOŽDĚNÍ.....	34
II PRAKTICKÁ ČÁST	36
7 ANALÝZA SOUČASNÉHO ŘÍZENÍ	37
7.1 VÁŽICÍ PROCES.....	37
7.2 POČÁTEČNÍ PODMÍNKY	38
7.3 ŘÍZENÍ DOPRAVNÍKU	38
8 IDENTIFIKACE NAMĚŘENÝCH DAT	43
8.1 MĚŘENÍ DAT.....	43
8.1.1 Průběh správného vážení.....	45
8.1.2 Průběh špatného vážení – mimo horní toleranci	46
8.2 APLIKACE ARX MODELU PRO ODHAD PARAMETRŮ.....	47
8.2.1 ARX model bez dopravního zpoždění	47
8.2.2 ARX model s dopravním zpožděním	49
9 PRŮBĚŽNÁ IDENTIFIKACE	51
9.1 RARX BEZ DOPRAVNÍHO ZPOŽDĚNÍ	51
9.2 RARX S DOPRAVNÍM ZPOŽDĚNÍM.....	52
10 NÁVRH ZPŮSOBU ŘÍZENÍ	54
10.1 CARIMA BEZ DOPRAVNÍHO ZPOŽDĚNÍ	54
10.2 CARIMA S DOPRAVNÍM ZPOŽDĚNÍM	55
11 SIMULAČNÍ OVĚŘENÍ V MATLAB SIMULINK	56

11.1	BEZ DOPRAVNÍHO ZPOŽDĚNÍ	56
11.2	S DOPRAVNÍM ZPOŽDĚNÍM	58
12	IMPLEMENTACE NAVRŽENÉHO ŘÍZENÍ DO PLC.....	60
13	OVĚŘENÍ NA REÁLNÉM ZAŘÍZENÍ	63
14	POROVNÁNÍ STÁVAJÍCÍHO A NAVRŽENÉHO ŘÍZENÍ.....	66
	ZÁVĚR	69
	SEZNAM POUŽITÉ LITERATURY.....	70
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	73
	SEZNAM OBRÁZKŮ	74
	SEZNAM PŘÍLOH.....	76

ÚVOD

Cílem diplomové práce je návrh řízení výšky hladiny materiálu při navážení na váhu míchací linky. Jedná se o míchací linky v závodě Continental Barum v Otrokovicích, který se zaměřuje na výrobu pneumatik. Jedná se tedy o řídicí aplikaci v gumárenském průmyslu.

Teoretická část je zaměřena na popis míchacích linek a popis řídicího problému, který je třeba vyřešit. Je zde popsáno jak hardwarové, tak softwarové vybavení použité pro řízení, měření a monitorování činnosti linky. Dále je v teoretické části popsán model a metody, které byly použity jak pro identifikaci počátečních odhadů parametrů řízeného procesu, tak pro průběžnou identifikaci parametrů. Je zde také popsána metoda řízení, která byla použita. Jednalo se o prediktivní řízení.

Praktická část je zaměřena na popis nového navrženého způsobu řízení míchací linky a jeho srovnání s dosavadním způsobem řízení. Bylo aplikováno prediktivní řízení v adaptivní verzi s průběžnou identifikací parametrů modelu řízeného procesu s využitím rekurzivní metody nejmenších čtverců a modelu ARX.

I. TEORETICKÁ ČÁST

1 MÍCHACÍ LINKY V ZÁVODĚ CONTINENTAL

V závodě Continental slouží míchací linky k míchaní různých druhů materiálů za účelem vytvoření směsi pro další zpracování ve výrobním procesu. Míchací linka je velké a komplexní zařízení, které je rozděleno na části. Diplomová práce se zabývá prvotní částí celého procesu, což je navažování materiálu na váhu. Materiál je skladován v silech, odkud je pomocí pásového dopravníku dopravován do váhy viz. obrázek:



Obrázek 1 Zjednodušený model míchací linky

Řízení rychlosti dopravníku má na starosti frekvenční měnič, který je spolu s celým řídicím systémem od firmy Siemens. Problém nastává při změně ročního období nebo změně druhu materiálu od dodavatele. Tímto problémem je myšleno špatné navážení materiálu. To je způsobeno změnou vlastností materiálu. Takovou změnu vlastnosti může především způsobit vlhkost.

2 POUŽITÝ ŘÍDICÍ A MĚŘICÍ SYSTÉM

Základem míchacích linek jsou systémy Siemens. Tyto systémy obstarávají běh programu, řízení pohonů, komunikaci s váhami, senzory, HMI apod. Siemens je jedna z největších a nejvýznamnějších technologických a inženýrských společností na světě. Specializuje se na různé oblasti, včetně průmyslové automatizace, energetiky, infrastruktury, digitálních technologií, dopravy, zdravotnictví a další. Mezi známé produkty a technologie společnosti Siemens patří například elektrické generátory a motory, řídicí systémy pro průmyslové procesy, lékařská zařízení jako magnetická rezonance a rentgenové přístroje, systémy pro energetické distribuce a mnoho dalšího. V současné době je Siemens vedoucím poskytovatelem řešení v oblasti digitalizace průmyslu, tzv. průmyslu 4.0, který se zaměřuje na propojení strojů a zařízení pomocí internetu věcí (IoT) a využívání dat a analýz pro optimalizaci výrobních procesů a výkonu. Požadavky na moderní stroje a výrobní systémy se stále zvětšují ve všech odvětvích průmyslu. Díky automatizačním systémům od Siemensu je možné pokrýt tyto požadavky spolu s maximální efektivitou, flexibilitou a snížením nákladů. [1]

2.1 SIMATIC

Průmyslové systémy SIMATIC mají vysokou konzistenci a efektivitu. Tím poskytují optimální řešení pro každou oblast aplikace. Pomocí integrovaného inženýrství pomocí TIA Portalu, je možné pomáhat podstatně snižovat finanční a časové náklady. Systémy pro průmyslovou automatizaci poskytují působivou komplexnost a rozšiřitelnost.

[2]

2.2 SIMATIC PLC

Jsou takzvaným srdcem strojů a výrobních linek (plants). Jejich výkonnost a potenciál využití v různých aplikacích je dělá úspěšnými v průmyslové automatizaci. Nespočet SIMATIC uživatelů a PLC programátorů prozkoumávají limity jejich kreativity a inovativní síly. SIMATIC PLC jsou navrženy pro optimalizaci jednoduchých, ale také i sofistikovaných projektů za pomoci TIA Portalu. Každý stroj má své požadavky z hlediska na výkon systému a komplexnosti aplikace. [3]



Obrázek 2 Siemens PLC – SIMATIC S7-1200 [4]

SIMATIC poskytuje širokou řadu variant řídicích jednotek. Liší se od sebe výkonem (otočka cyklu, počtem procesorů), počtem vstupů a výstupů, komunikačními schopnostmi, modulárností, rozšiřitelností, typem určení (řízení energetiky, motorů – 1osé, víceosé, PID, atd, měření, počítání). Obecně také nabízejí funkce jako je diagnostika, bezpečnost a jednoduchou implementaci do aplikací. Tyto funkce přispívají k větší spolehlivosti a efektivitě automatických systémů, což je klíčové pro průmyslové prostředí, kde spolehlivost a výkonnost hrají klíčovou roli. [3]

2.3 Process control system 7 – PCS7

Pro programování PLC SIMATIC a propojení Siemens hardwaru použitých u míchacích linek se používá PCS7. Jedná se o distribuovaný řídicí systém používaný v průmyslové automatizaci a aplikacích řízení procesu. Je navržen pro řízení a sledování různých proměnných procesu jako jsou například:

- Teplota
- Tlak
- Průtok
- Hladina

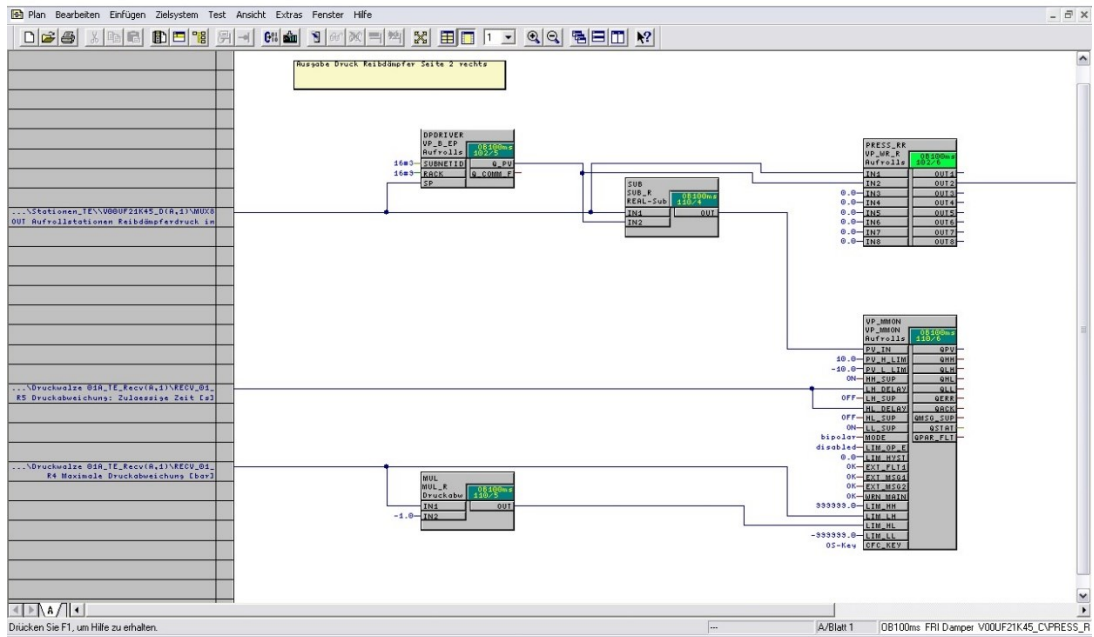
Je to modulární systém, který je snadno přizpůsobitelný různým aplikacím a lze ho integrovat do existujících Siemens systémů, jako jsou PLC a HMI, což zajišťuje kompatibilitu a harmonický provoz. PCS7 je navržen s důrazem na vysokou odolnost, spolehlivost, výkonnost a je vybaven integrovanými bezpečnostními funkcemi pro udržení nepřetržitého provozu i v případě poruchy. PCS7 je používán ve spoustě průmyslových sektorech jako:

- Ropa a plyn
- Energie
- Voda a ochrana před úniky
- Chemikálie
- Farmacie
- Potraviny
- A další

PCS7 poskytuje řadu vlastností:

- Řízení a sledování procesních proměnných v reálném čase
- Pokročilé možnosti řízení a optimalizační algoritmy
- Podpora komunikačních protokolů PROFIBUS, PROFINET and Ethernet/IP
- Uživatelské rozhraní pro operátory a vizualizaci
- Zaznamenávání události a alarmů
- Správa receptur a kontrola šarží
- Záznam a analýza historických dat
- Vzdálený přístup a vzdálená údržba

Hlavní prvky, které PCS7 nabízí jsou zejména rozšiřitelnost (ekonomické náklady, rekonfigurace), dostupnost (redundantní konfigurace – hlavně bezpečnostní aplikace), podpora většiny komunikačních protokolů (FOUNDATION Fieldbus, PROFIBUS, PROFINET), efektivní inženýrství (cause&effect matice, optimalizace), bezpečnost a správa jednotlivých komponent. [5][6]



Obrázek 3 CFC – Continuous function chart – ukázka z PCS7 [7]

Jazyk SCL se používá pro generování individuálně upravených bloků. Díky tomu je možné sestavit například složité výpočty v jednom bloku. Obsahuje i řídicí prvky pro motory, ventily atd. Podporuje spoustu rozhodovacích pravidel. Ochrana know-how ochrání individuálně naprogramovaný software. Dále poskytuje příslušné soubory, které popisují příslušný blok a jeho zapojení/propojení. [7]

```

113 -
114 LABEL
115   Sprung1;
116   Sprung2;
117   Sprung3;
118   Sprung4;
119 END_LABEL
120
121 BEGIN
122 // ***** Volumen Pulper / Zeitkonstante *****
123   Volumen_Pulper := (V / 100.0) * LIC_SOLL;
124   Volumen_in_l := (Volumen_Pulper * 1000.0);
125   Volumen_in_ml := (Volumen_in_l * 1000.0);
126   Zeitkonstante := T#0s;
127
128 // ***** Umrechnungen Grundmenge *****
129   Gewicht_Stoff_g := ((K / 100.0) * Volumen_in_ml);
130   Gewicht_Stoff_kg := (Gewicht_Stoff_g / 1000.0);
131   Ben_Ballen := (Gewicht_Stoff_kg / KG_B);
132   tf_real := (Ben_Ballen * SEC_M);
133   tf_dint := REAL_TO_DINT(tf_real);
134   tf_Faktor := (tf_dint * 1000);
135
136 // ***** Umrechnungen Aktuelle Menge *****
137   FIC_ml := (FIC * 1000.0);
138   Stofffaktuell_g := ((QIC / 100.0) * FIC_ml);
139   Stofffaktuell_kg := (Stofffaktuell_g / 1000.0);
140   Ballenaktuell := (Stofffaktuell_kg / KG_B);
141   tfaktuell_real := (Ballenaktuell * SEC_M);
142   tfaktuell_dint := REAL_TO_DINT(tfaktuell_real);
143   tfaktuell_Faktor := (tfaktuell_dint * 1000);
144
145 // ***** Programmstart *****
146 IF (LIC > V_MAX) OR (LIC <= 30.0) THEN //Niveauabfrage
147   Flag := FALSE;
148   Flag_M := FALSE;
149   Flag_M_Reg := FALSE;
150   tp := Zeitkonstante;
151   M := FALSE;
152   GOTO Sprung3;
153 END_IF;
154
155 LIC_Plus := LIC + LI_MAX;
156 IF (LI >= LIC_Plus) THEN //Stoffhöhe aus Wasser
157   Flag_Stoff := TRUE;
158 ELSE
159   Flag_Stoff := FALSE;
160 END_IF;

```

Obrázek 4 Ukázka strukturovaného textu – SCL [7]

2.4 Siwarex – vážicí elektronika

Siwarex Load Cells jsou vážicí moduly, které Siemens používá pro efektivní vážení, plnění, pytlování a další vážicí aplikace. Tyto váhové senzory jsou navrženy pro snadnou integraci do prostředí SIMATIC, což zaručuje jednoduché a bezproblémové integrace do průmyslových procesů. Využívají se v kombinaci s diagnostikou a bezpečností. To umožňuje úpravu prostředí dle potřeb a zároveň zajistí bezpečnost provozu. Pomocí systémové sběrnice lze přímou komunikací s CPU vytvořit vážicí systém o vysoké úrovni výkonnosti, spolehlivosti a přesnosti. Siwarex nabízí širokou škálu designů – platformové, ohýbací, nůžkové, S-typové a kompresní jednotky. Nominální váhy jsou v rozmezí 0,3 kg až 500 tun. Správná instalace a umístění senzorů vede k vysoké přesnosti. To je podpořeno širokým spektrem připevňovacích zařízení a příslušenství. Siwarex je integrován do prostředí Step7, PCS7 a TIA Portal.

Díky integraci do těchto prostředí je zajištěna kompatibilita s řadou automatizačních systémů Siemens. [8][9]



Obrázek 5 Siwarex – ohýbací siloměr - WL230 BB-S SA [10]

Pro hlídání váhy u míchacích linek se používají ohýbací nosníkové siloměry. Tyto jsou používány do 500 kg a jsou použity převážně pro aplikace menšího rozsahu.[10]

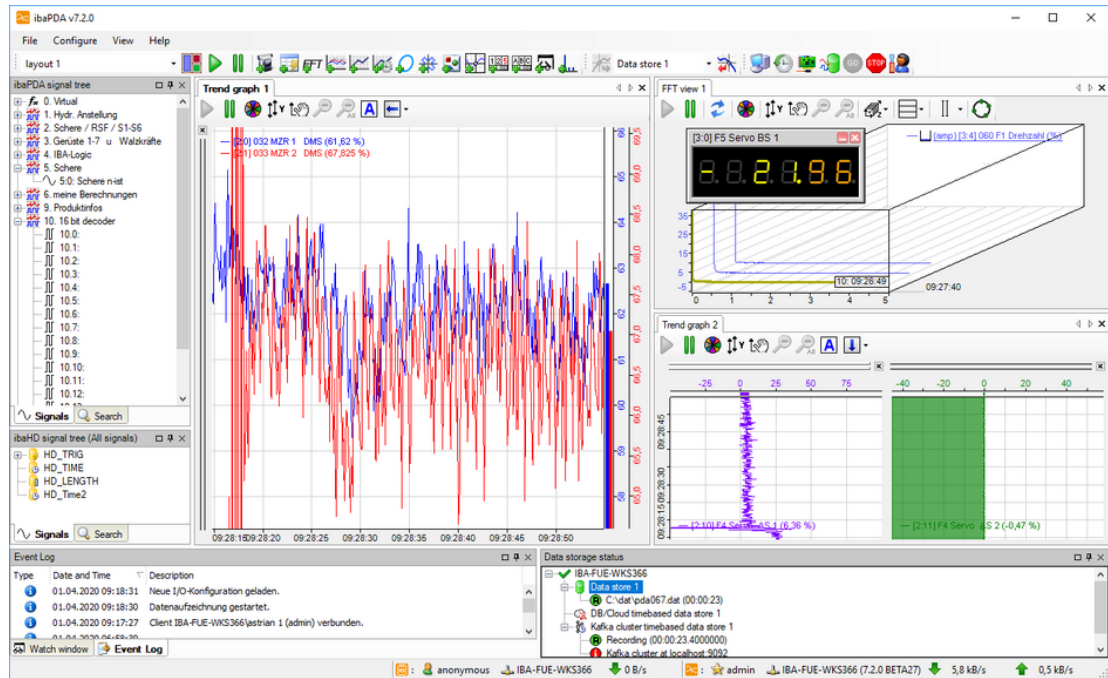
3 SOFTWARE PRO MĚŘENÍ DAT

3.1 iba AG

Pro měření potřebných dat z míchacích linek byl použit software od společnosti iba AG. Iba AG poskytuje hardwarové a softwarové nástroje pro sběr, záznam, analýzu a zpracování dat. Iba nabízí modulární konstrukci a jednoduchou konfiguraci. Díky tomu lze systém pohodlně přizpůsobit různým úkolům a je kdykoli škálovatelný. Základem každé analýzy procesních dat je časově synchronní získávání relevantních dat v charakteristických místech v průběhu automatického procesu na automatickém pracovišti. Iba je extrémně výkonný, PC-based software pro získávání a nahrávání různých procesních dat. Nahrává a ukládá jednotlivá naměřená data, která jsou přístupna dlouhou dobu – jako černá skříňka letadel. Data jsou měřena nepřetržitě 24/7 nebo je možné měření zahájit definovanou událostí. Je možné kombinovat různé zdroje signálů. Vzhledem k nesynchronnímu měření, běžné vazby mohou být nalezeny a pochopeny, dokonce i v komplexních a distribuovaných systémech. Tradiční systém měření získává signály z elektrických senzorů za pomoci A/D převodníků. V éře digitálních systémů, většina měřených hodnot jsou již přítomna v automatizovaném systému. Iba často poskytuje metody přímého přístupu k vnitřním hodnotám kontrolovaného systému. Přístup k vnitřním hodnotám PLC a jejich měření nevyžaduje vypnutí PLC nebo přizpůsobení systému k požadavku. Hlavním charakterem je široká konektivita (připojitelnost) – průmyslové sběrnice, Ethernet nebo systémové rozhraní. Může být připojeno k téměř každému automatizovanému systému nezávisle na výrobci a stáří stroje. [11][12][13]

3.2 ibaPDA – Process Data Acquisition

ibaPDA je centrální část systému. Je to nejuniverzálnější nástroj systému iba, který získává procesní data s vysokým rozlišením. Komplexní připojení k procesu s funkcí autodetekce pro snadnou konfiguraci, architektura klient-server a variabilní možnosti nahrávání. Pružné nastavení konfigurace a poskytování správně „ušitých“ řešení pro různé potřeby. Je určeno pro spojitě měření dat jednotlivé linky, ale také i celé továrny. Běžná časová základna je 1-1000 ms. Pro vyšší rychlosti v závislosti na změnách signálu je možné pomocí speciálních modulů realizovat zkrácení doby získávání hodnoty až na 10 μ s. Data je možné posílat na datové uložení umístěné na serveru, ale také je možné data uložit na jiných typech uložení. Pomocí licence je možné vysílat a vizualizovat data přímo v ibaPDA. Licence dále ovlivňuje maximální počet sledovaných signálů. [14]



Obrázek 6 Ukázka rozhraní ibaPDA pro získávání naměřených hodnot [14]

4 ARX MODEL

Pro popis řízeného procesu byl použit ARX model druhého řádu, který je vhodný a dostatečně přesný pro popis celé řady procesů. Následně použité prediktivní řízení vyžaduje velmi přesný model řízeného systému. Tento požadavek použitý ARX model splňuje. Pro odhad parametrů modelu ARX modelu existuje řada metod. V této práci byla použita metoda nejmenších čtverců. ARX (Autoregressive with eXogenous input) je autoregresní model s externím vstupem. ARX modely jsou silným nástrojem pro modelování a analýzu chování dynamického systému a jsou často použity v řídicích systémech. Obecně mají tyto modely široké využití.

Autoregresní model

Je to typ statistického modelu, který reprezentuje časové posloupnosti jako lineární kombinaci předchozích hodnot a stochastického procesu. To může být zapsáno rovnicí:

$$y(k) = c + a_1y(k-1) + a_2y(k-2) + \dots + a_{n_a}y(k-n_a) + e(k) \quad (4.1)$$

kde:

- $y(k)$ je hodnota časové posloupnosti v čase t
- c je konstantní parametr
- a_1, \dots, a_p jsou autoregresní koeficienty, kde n_a je stupeň polynomu A
- $y(k-1), \dots, y(k-p)$ jsou minulé hodnoty časové posloupnosti
- $e(k)$ je náhodná chyba

Exogenní vstup

Reprezentuje časovou posloupnost jako lineární kombinaci předchozích hodnot a set exogenních vstupních proměnných. To může být zapsáno rovnicí:

$$y(k) = c + b_1u(k-1) + b_2u(k-2) + \dots + b_{n_b}u(k-n_b) + e(k) \quad (4.2)$$

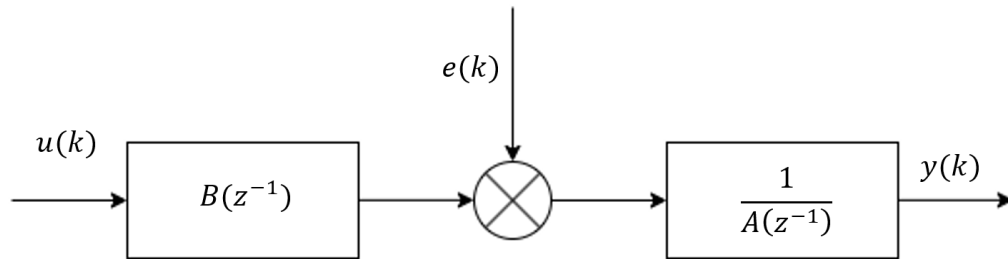
kde:

- b_1, \dots, b_q jsou koeficienty, které zachycují vztah mezi vstupními proměnnými a výstupem, kde n_b je stupeň polynomu B
- $u(k-1), \dots, u(k-n_b)$ jsou exogenní vstupní proměnné
- $e(k)$ je náhodná chyba

ARX model

ARX modely v časových posloupnostech jsou lineární reprezentací dynamických systémů v diskrétním čase. Je dán rovnicí

$$y(k) = c + a_1y(k-1) + \dots + a_{n_a}y(k-n_a) + b_1u(k) + \dots + b_{n_b}u(k-n_b) + e(k) \quad (4.3)$$



Obrázek 7 Blokové schéma ARX

Z blokového schématu je možné získat vztah:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{1}{A(z^{-1})}e(k) \quad (4.4)$$

kde je možné vidět diskrétní přenosovou funkci:

$$\frac{B(z^{-1})}{A(z^{-1})} = G(z^{-1}) \quad (4.5)$$

Jednotlivé polynomy mají následující tvar:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_a}z^{-n_a} \quad (4.6)$$

$$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_{n_b}z^{-n_b} \quad (4.7)$$

kde n_a a n_b jsou řády polynomu.

ARX model je často zapsán ve vektorové formě:

$$y(k) = \theta^T \phi(k) + e(k) \quad (4.8)$$

kde θ je vektor parametrů:

$$\theta^T = [a_1 \quad a_2 \quad \dots \quad a_{n_a} \quad b_1 \quad b_2 \quad \dots \quad b_{n_b}] \quad (4.9)$$

a $\phi(k)$ je vektor dat:

$$\phi(k) = [-y(k-1) \quad \dots \quad -y(k-n_a) \quad u(k-1) \quad \dots \quad u(k-n_b)] \quad (4.10)$$

Při uvažování ARX modelu 2 řádu dostaneme tedy následující přenosovou funkci:

$$G(z) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (4.11)$$

a vektorový zápis by vypadal následovně:

$$y(k) = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} [-y(k-1) \quad -y(k-2) \quad u(k-1) \quad u(k-2)] + e(k) \quad (4.12)$$

[15][16][17]

4.1 Princip metody nejmenších čtverců

Jedná se o metodu regresní analýzy vhodnou pro vyšetření chování statických i dynamických vlastností zkoumaného systému. Je založena na minimalizaci čtverců chyb mezi naměřenými a vypočtenými procesními výstupy. Uvažujme stochastický proces popsany pomocí ARX modelu a pro vektor parametrů a vektor dat předpokládejme $n_a = n_b = n$

Z obecného modelu ARX známe tvar vektorových zápisů:

$$\theta^T = [a_1 \quad \dots \quad a_n \quad b_1 \quad \dots \quad b_n] \quad (4.13)$$

$$\phi(k-1) = [-y(k-1) \quad \dots \quad -y(k-n) \quad u(k-1) \quad \dots \quad u(k-n)] \quad (4.14)$$

Poté generování výstupního signálu $y(k)$ v individuálním časovém momentu může být vyjádřeno pomocí maticové rovnice:

$$y = F\theta + e \quad (4.15)$$

kde matice F o rozměrech $(N-n, 2n)$ a vektorech y, e o rozměrech $(N-n, 1)$ mají tvar:

$$y^T = [y(n+1) \quad y(n+2) \quad \dots \quad y(N)] \quad (4.16)$$

$$e^T = [e(n+1) \quad e(n+2) \quad \dots \quad e(N)] \quad (4.17)$$

$$F = \begin{bmatrix} -y(n) & -y(n-1) & \dots & -y(1) & u(n) & \dots & u(1) \\ -y(n+1) & -y(n) & \dots & -y(2) & u(n+1) & \dots & u(2) \\ \vdots & & & & & & \vdots \\ -y(N-1) & -y(N-2) & \dots & -y(N-n) & u(N-1) & \dots & u(N-n) \end{bmatrix} \quad (4.18)$$

N je počet vzorků naměřených dat. Chyba může být stanovena z rovnice následovně:

$$e = y - F\theta \quad (4.19)$$

Hledáme takové odhady parametry modelu, které při minimalizaci sumy čtverců chyb mezi naměřenými daty a modelovanými výstupy:

$$J(\theta) = \sum_{i=1}^k (y(i) - F(i)\theta)^2 \quad (4.20)$$

Taková rovnice lze přepsat do vektorového zápisu:

$$J = e^T e = (y - F\theta)^T (y - F\theta) \quad (4.21)$$

Minimum může být získáno pomocí derivace kritéria podle θ . Následně se derivace položí nule:

$$\left. \frac{\partial J}{\partial \theta} \right|_{\theta=\hat{\theta}} = 0 \quad (4.22)$$

Dostaneme tedy:

$$(F^T F)\hat{\theta} = F^T y \quad (4.23)$$

Je-li matice $F^T F$ invertovatelná, získáme základní maticový vztah pro odhad parametrů pomocí MNČ:

$$\hat{\theta} = (F^T F)^{-1} F^T y \quad (4.24)$$

Tato rovnice slouží pro jednorázový výpočet odhadů parametrů modelu pomocí N vzorků naměřených dat. Tento výpočet požaduje větší nárok na paměť počítače, která musí být dost velká na uchování naměřených dat. [17][18]

4.2 Modifikace pro systémy s dopravním zpožděním:

Pro modely s dopravním zpožděním je možné modifikovat MNČ pomocí zakomponování dopravního zpoždění do matic a vektorů. Dopravní zpoždění označme jako d , kdy hodnota dopravního zpoždění je násobek periody vzorkování T_0 . Tedy upraví se vektor výstupních hodnot o rozměru $(N - n - d)$:

$$y^T = [y(n + d + 1) \quad y(n + d + 2) \quad \cdots \quad y(N)] \quad (4.25)$$

Dále se upraví matice dat o rozměru $(N - n - d, 2n)$:

$$F = \begin{bmatrix} -y(n+d) & \cdots & -y(d+1) & u(n) & \cdots & u(1) \\ -y(n+d+1) & \cdots & -y(d+2) & u(n+1) & \cdots & u(2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y(N-1) & \cdots & -y(N-n) & u(N-d-1) & \cdots & u(N-d-n) \end{bmatrix} \quad (4.26)$$

Po úpravě vektoru výstupních hodnot a matice dat se dále pokračuje stejně, jako u klasické verze MNČ. Tedy hledají se takové parametry, které by zajistili nejmenší chybu, kdy dostaneme výsledný vztah:

$$\hat{\theta} = (F^T F)^{-1} F^T y \quad (4.27)$$

[19]

5 REKURZIVNÍ IDENTIFIKAČNÍ ALGORITMY

Pro průběžnou aktualizaci parametrů modelu ARX byla použita rekurzivní metoda nejmenších čtverců, což je základem úspěšného prediktivního řízení. Rekurzivní identifikační metody se používají pro průběžnou identifikaci v reálném čase, kdy se odhady parametrů modelu $\hat{\theta}(k-1)$ v každém kroku opraví novými daty a získají se nové odhady parametrů $\hat{\theta}(k)$. Nová data se tedy použijí k aktualizaci odhadů parametrů z minulého kroku. Tento přístup snižuje náročnost na výpočetní techniku. Rekurzivní metoda nejmenších čtverců je nejvíce používaná metoda pro průběžnou identifikaci. [18][20]

Jednorázová verze MNČ je definována vztahem:

$$\hat{\theta} = (F^T F)^{-1} F^T y \quad (5.1)$$

Při použití vzorků z minulého pozorování lze tento vztah modifikovat:

$$\hat{\theta}(k-1) = (F_{k-1}^T F_{k-1})^{-1} F_{k-1}^T y(k-1) \quad (5.2)$$

kde:

$$y^T(k-1) = [y(1) \quad y(2) \quad \dots \quad y(k-1)] \quad (5.3)$$

je vektor výstupních hodnot do minulého pozorování a:

$$\hat{\theta}^T(k-1) = [\hat{\theta}_1(k-1) \quad \hat{\theta}_2(k-1) \quad \dots \quad \hat{\theta}_n(k-1)] \quad (5.4)$$

je vektor odhadů parametrů z minulého kroku.

Matrice:

$$F_{k-1} = \begin{bmatrix} f_1(1) & f_2(1) & \dots & f_r(1) \\ f_1(2) & f_2(2) & \dots & f_r(2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(k-1) & f_2(k-1) & \dots & f_r(k-1) \end{bmatrix} \quad (5.5)$$

je modifikována pro minulé pozorování.

Po novém pozorování získáme:

$$y(k) = \begin{bmatrix} y(k-1) \\ y(k) \end{bmatrix} \quad (5.6)$$

Dále lze přepsat matici F do tvaru:

$$F_k = \begin{bmatrix} f_1(1) & f_2(1) & \cdots & f_r(1) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(k-1) & f_2(k-1) & \cdots & f_r(k-1) \\ f_1(k) & f_2(k) & \cdots & f_r(k) \end{bmatrix} \begin{bmatrix} F_{k-1} \\ \phi^T(k) \end{bmatrix} \quad (5.7)$$

kde označíme poslední řádek matice F_k jako:

$$\phi^T(k) = [f_1(k) \quad f_2(k) \quad \cdots \quad f_r(k)] \quad (5.8)$$

Pro k -té pozorování hodnoty $y(k)$ lze zapsat:

$$y(k) = \theta_1 f_1(k) + \theta_2 f_2(k) + \cdots + \theta_r f_r(k) + e(k) \quad (5.9)$$

Tento zápis lze vyjádřit vektorově:

$$y(k) = \theta^T \phi(k) + e(k) \quad (5.10)$$

kde:

$$\theta^T = [\theta_1 \quad \theta_1 \quad \cdots \quad \theta_r] \quad (5.11)$$

Definujeme kovarianční matici $C(k-1)$:

$$C(k-1) = (F_{k-1}^T F_{k-1})^{-1} \quad (5.12)$$

Pro kovarianční matici $C(k)$ platí:

$$C(k) = (F_k^T F_k)^{-1} \quad (5.13)$$

a tedy:

$$C(k) = \left[[F_{k-1}^T \quad \phi(k)] \begin{bmatrix} F_{k-1} \\ \phi^T(k) \end{bmatrix} \right]^{-1} = [F_{k-1}^T F_{k-1} + \phi(k) \phi^T(k)]^{-1} \quad (5.14)$$

Dosazením (5.12) do (5.14) získáme:

$$C(k) = [C(k-1) + \phi(k) \phi^T(k)]^{-1} \quad (5.15)$$

Pro symetrické a pozitivně definitní matice A a C platí věta o inverzi matic:

$$[A + BCB^T]^{-1} = A^{-1} - A^{-1}B[C^{-1} + B^T A^{-1}B]^{-1}B^T A^{-1} \quad (5.16)$$

Provedeme přiřazení matic:

$$A \sim C^{-1}(k-1), \quad B \sim \phi(k), \quad C \sim 1$$

a po zavedení (5.16) do (5.15) získáme:

$$\begin{aligned} C(k) &= [C^{-1}(k-1) + \phi(k)\phi^T(k)]^{-1} = \\ &= C(k-1) - C(k-1)\phi(k)[1 + \phi^T(k)C(k-1)\phi(k)]^{-1}\phi^T(k)C(k-1) \end{aligned} \quad (5.17)$$

Zavedeme substituci:

$$\gamma(k-1) = [1 + \phi^T(k)C(k-1)\phi(k)]^{-1} \quad (5.18)$$

Obdržíme zjednodušený vztah:

$$C(k) = C(k-1) - \gamma(k-1)C(k-1)\phi(k)\phi^T(k)C(k-1) \quad (5.19)$$

Zavedená substituce $\gamma(k)$ je skalár. Tedy operace inverze matic má význam převrácené hodnoty. Pro optimální odhad parametrů bude platit:

$$\begin{aligned} \hat{\theta}(k) &= (F_k^T F_k)^{-1} F_k^T y(k) = C(k) \begin{bmatrix} F_{k-1}^T & \phi(k) \end{bmatrix} \begin{bmatrix} y(k-1) \\ y(k) \end{bmatrix} = \\ &= C(k) [F_{k-1}^T y(k-1) + \phi(k)y(k)] \end{aligned} \quad (5.20)$$

Po dosazení za $C(k)$:

$$\begin{aligned} \hat{\theta}(k) &= [C(k-1) - \gamma(k-1)C(k-1)\phi(k)\phi^T(k)C(k-1)] [F_{k-1}^T y(k-1) \\ &\quad + \phi(k)y(k)] \end{aligned} \quad (5.21)$$

První člen první závorky se vynásobí s druhou závorkou. Dále se zavede:

$$\gamma(k-1)\gamma^{-1}(k-1) = 1 \quad (5.22)$$

a dostaneme:

$$\begin{aligned} \hat{\theta}(k) &= C(k-1)F_{k-1}^T y(k-1) + C(k-1)\phi(k)\gamma(k-1)\gamma^{-1}(k-1)y(k) - \\ &\quad - \gamma(k-1)C(k-1)\phi(k)\phi^T(k)C(k-1)[F_{k-1}^T y(k-1) + \phi(k)y(k)] \end{aligned} \quad (5.23)$$

Jelikož platí $C(k-1) = (F_{k-1}^T F_{k-1})^{-1}$, potom plyne:

$$\hat{\theta}(k-1) = (F_{k-1}^T F_{k-1})^{-1} F_{k-1}^T y(k-1) = C(k-1)F_{k-1}^T y(k-1) \quad (5.24)$$

S ohledem, že platí (5.18) a následným dosazením (5.24) do rovnice (5.23), získáme tvar:

$$\begin{aligned} \hat{\theta}(k) &= \hat{\theta}(k-1) + \underline{C(k-1)\phi(k)\gamma(k-1)} [1 + \phi^T(k)C(k-1)\phi(k)]^{-1} y(k) - \\ &\quad - \underline{\gamma(k-1)C(k-1)\phi(k)\phi^T(k)C(k-1)F_{k-1}^T y(k-1)} - \\ &\quad - \underline{\gamma(k-1)C(k-1)\phi(k)\phi^T(k)C(k-1)\phi(k)y(k)} \end{aligned} \quad (5.25)$$

Podtržený výraz je možné vytknout a tím se získá zjednodušený vztah:

$$\begin{aligned}\hat{\theta}(k) &= \hat{\theta}(k-1) + \gamma(k-1)C(k-1)\phi(k) \\ & [y(k) + \phi^T(k)C(k-1)\phi(k)y(k) - \phi^T(k)C(k-1)F_{k-1}^T y(k-1) - \\ & - \phi^T(k)C(k-1)\phi(k)y(k)]\end{aligned}\quad (5.26)$$

Substitucí (5.24) do předešlého výrazu a po úpravě získáme konečný vztah pro odhad parametrů $\hat{\theta}(k)$ pomocí rekurzivní MNČ:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \gamma(k-1)C(k-1)\phi(k)[y(k) - \hat{\theta}^T(k-1)\phi(k)] \quad (5.27)$$

Obecný tvar rekurzivního algoritmu:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)[y(k) - \hat{\theta}^T(k-1)\phi(k)] \quad (5.28)$$

Korekční činitel je časově proměnlivý vektor zesílení. Ten je dán vztahem:

$$K(k) = \gamma(k-1)C(k-1)\phi(k) = \frac{C(k-1)\phi(k)}{1 + \phi^T(k)C(k-1)\phi(k)} \quad (5.29)$$

Pro čtvercovou kovarianční matici o rozměrech $(2n, 2n)$, kde n je řád systému, platí rekurzivní vztah:

$$\begin{aligned}C(k) &= C(k-1) - C(k-1)\phi(k)[1 + \phi^T(k)C(k-1)\phi(k)]^{-1}\phi^T(k)C(k-1) \\ &= C(k-1) \left[1 - \frac{\phi(k)\phi^T(k)C(k-1)}{1 + \phi^T(k)C(k-1)\phi(k)} \right]\end{aligned}\quad (5.30)$$

[20]

5.1 Algoritmus pro rekurzivní MNČ

Uvažujme 2 řád modelu:

1. Naplnění vektoru $\phi(k)$ novými daty

$$\phi^T(k) = [-y(k-1) \quad -y(k-2) \quad u(k-1) \quad u(k-2)]$$

2. Výpočet chyby

$$\hat{e}(k) = y(k) - \phi^T(k)\hat{\theta}(k-1)$$

3. Aktualizace kovarianční matice

$$C(k) = C(k-1) - \frac{C(k-1)\phi(k)\phi^T(k)C(k-1)}{1 + \phi^T(k)C(k-1)\phi(k)}$$

4. Aktualizace parametrů modelu

$$\hat{\theta}(k) = \hat{\theta}(k-1) + C(k)\phi(k)\hat{e}(k)$$

5. Opakování předchozích 4 bodů pro nová navzorkovaná data

[20]

5.2 Modifikace pro systémy s dopravním zpožděním

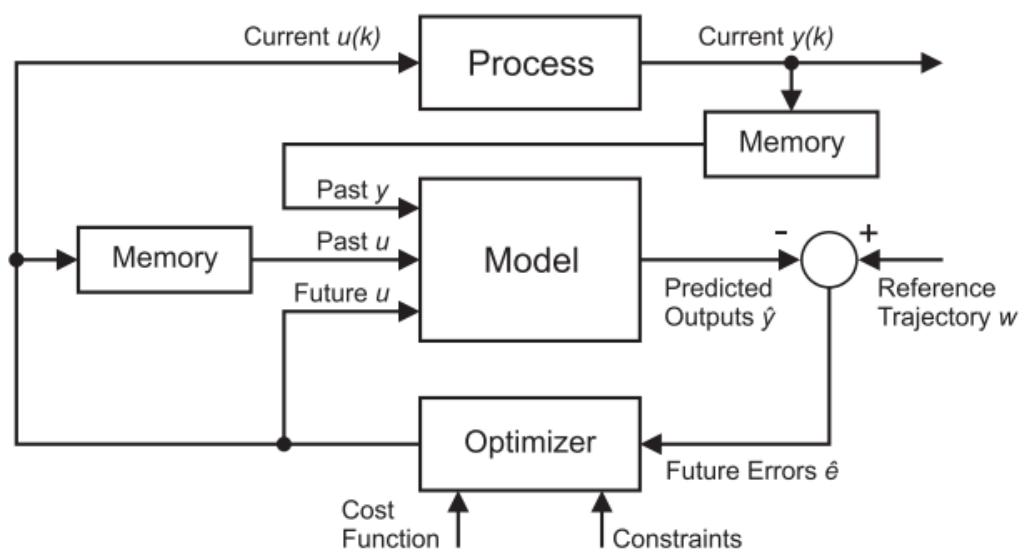
Pro systémy s dopravním zpožděním je potřeba algoritmus modifikovat. Modifikace se provádí v prvním kroku algoritmu rekurzivní MNČ, kdy se posunou naměřená data o d , které značí hodnotu dopravního zpoždění, tedy násobku periody vzorkování T_0 . Tedy:

$$\phi^T(k) = [-y(k-1) \quad -y(k-2) \quad u(k-d-1) \quad u(k-d-2)] \quad (5.30)$$

Algoritmus dále pokračuje beze změny. [19]

6 PREDIKTIVNÍ ŘÍZENÍ

Pro návrh regulace bylo použito prediktivního řízení. Přesněji byl použit model CARIMA, který vyžaduje přesné znalosti parametrů modelu, které jsou získány rekurzivní MNČ. Některé technologické procesy v průmyslu jsou charakterizovány dynamikou vysokého řádu, velkými časovými konstantami či dopravním zpožděním. Takové procesy mohou taky často nabývat nelinearit a nejisto, což zvyšuje složitost jejich řízení. Nicméně aproximací procesu vyššího řádu pomocí nízko-řádového modelu poskytuje zjednodušení řídicích algoritmů. Když je požadován vysoký výkon řízeného procesu nebo je dopravní zpoždění velmi velké, strategie prediktivního řízení je jeden z možných přístupů. Strategie prediktivního řízení zahrnuje model procesu ve struktuře regulátoru. Nejmodernější regulátory jsou implementovány na digitální platformě. Digitální kompenzátory dopravního zpoždění jsou více citlivé na proměnlivé parametry procesu a proto je zapotřebí adaptivního nebo samočinně se nastavující přístup v mnoha aplikacích. Jeden z možných přístupů je model prediktivního řízení (MPC). Ten popisuje sadu pokročilých řídicích metod, kde se pro predikci budoucího chování řízeného systému používá model. Řešením optimalizačního stanovuje MPC řídicí zákon. MPC může také řídit systémy, které nemohou být řízeny pomocí konvenčních zpětnovazebních regulátorů.



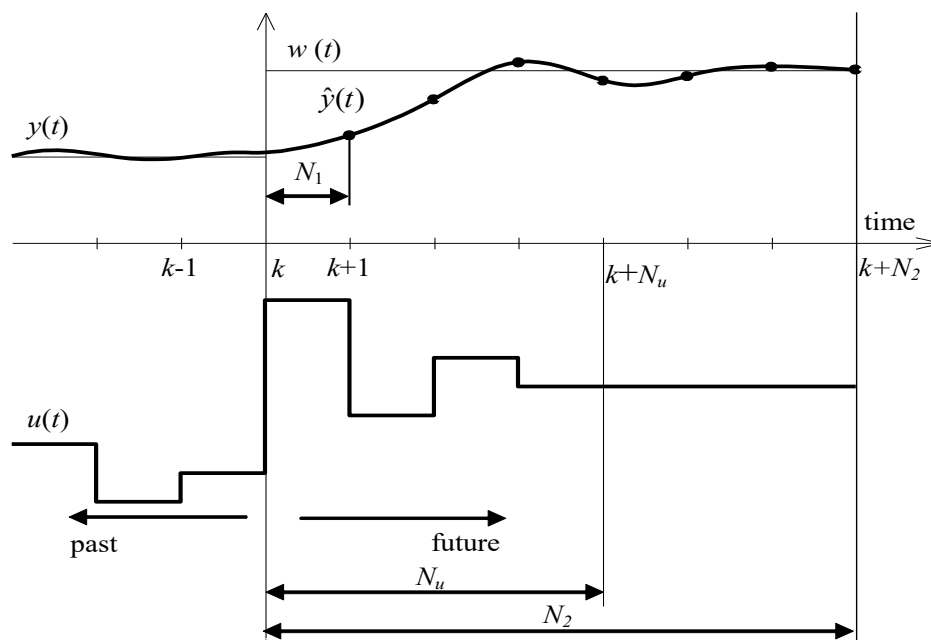
Obrázek 8 Schématicky znázorněný princip prediktivního řízení

Základní myšlenky prediktivního řízení jsou:

- Využití modelu procesu pro předpověď výstupu procesu pro pevný počet kroků do budoucnosti
- Budoucí trajektorie žádané veličiny je známa
- Výpočet řídicí sekvence minimalizuje danou funkci
- Strategie ústupu – pro každou vzorkovací periodu se pouze první řídicí signál vypočítané frekvence aplikuje na proces

Zájem pro aplikaci do průmyslu vyvolala vlastnost, kdy je možné omezení procesu snadno začlenit do návrhu řízení. Prediktivní řízení se vyznačuje všestranností a robustností v aplikacích řízení procesu. Jsou aplikovány do procesů s neminimální fází, s nestabilní otevřenou smyčkou, s dopravním zpožděním nebo do procesů s více proměnnými.

Model procesu je určen k předpovědi výstupů procesu v nějakém časovém období. Tyto predikce jsou vypočítány na základě aktuálně známých informací a budoucích řídicích zásahů, které se vypočítávají. Tyto řídicí zásahy se vypočítávají jako řešení optimalizační úlohy zahrnující účelovou funkci a případně omezení. Účelová funkce zahrnuje predikce výstupu, trajektorii žádané veličiny a akční zásahy.



Obrázek 9 Princip prediktivního řízení

Horizonty N_1 , N_2 a N_u se nazývají minimální, maximální a řídicí horizont. Horizonty N_1 a N_2 označují časový interval, kde je žádoucí sledovat trajektorii. V případě dopravního

zpoždění by měla být volba N_1 alespoň rovna $T_d + 1$, kde T_d je hodnota dopravního zpoždění. N_1 může také kompenzovat chování neminimální fáze. Hodnota N_2 by měla být pokrývat významnou část přechodového děje křivky. N_u ovlivňuje výpočetní složitost metody. [18][19][21][22]

6.1 Výpočet optimálního řízení - minimalizace kvadratického kritéria

Řídicí algoritmus je založen na zobecněném prediktivním řízení (GPC). Základní účelová funkce použita v GPC metodě obsahuje kvadratický člen chyby řízení a řídicí inkrementy na konečném horizontu v budoucnosti. Má tvar:

$$J = \sum_{i=N_1}^{N_2} [\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_u} [\lambda(i)\Delta u(k+i-1)]^2 \quad (6.1)$$

Kde $\hat{y}(k+i)$ jsou předvídané procesní výstupy i kroků v budoucnu na základě informací dostupných v čase k , $w(k+i)$ je sekvence referenčního signálu a $\Delta u(k+i-1)$ je sekvence budoucích řídicích inkrementů, které musí být vypočteny. Parametr $\lambda(i)$ je sekvence, která ovlivňuje budoucí chování řízeného procesu. Obecně je volen ve formě konstant nebo exponenciálních vah. Výstup z modelu (prediktoru) je vypočítán jako suma nuceného výstupu y_n a volné odezvy y_0 :

$$\hat{y} = y_n + y_0 \quad (6.2)$$

Volná odezva je ta část predikce, která je stanovena minulými hodnotami manipulované proměnné a minulými hodnotami výstupu systému. Nucená odezva je dána budoucími inkrementy manipulované proměnné a je vypočtena jako násobení matice G (Jacobiánova matice systému) a vektoru budoucích inkrementů Δu , který je obecně neznámá. Tedy:

$$y_n = G\Delta u \quad (6.3)$$

kde:

$$G = \begin{bmatrix} g_1 & 0 & 0 & \dots & 0 \\ g_2 & g_1 & 0 & \dots & 0 \\ g_3 & g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & g_{N_2-1} & \dots & g_{N_2-N_u+1} \end{bmatrix} \quad (6.4)$$

je matice, která obsahuje hodnoty posloupnosti kroků.

Prediktor ve vektorové formě je tedy dán rovnicemi (6.2) a (6.3):

$$(6.5)$$

$$\hat{y} = G\Delta u + y_0$$

Účelová funkce (6.1) může být modifikována na tvar:

$$\begin{aligned} J &= (\hat{y} - w)^T (\hat{y} - w) + \lambda \Delta u^T \Delta u \\ &= (G\Delta u + y_0 - w)^T (G\Delta u + y_0 - w) + \lambda \Delta u^T \Delta u \end{aligned} \quad (6.6)$$

Kde w je vektor trajektorie reference v budoucnu.

Minimalizace této účelové funkce se nyní stává přímým problémem lineární algebry. Řešení v neomezeném případě může být nalezen pomocí položení parciální derivace J vzhledem k Δu nule:

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (y_0 - w) \quad (6.7)$$

Kde gradient g a Hessián H jsou definovány jako:

$$g^T = G^T (y_0 - w) \quad (6.8)$$

$$H = G^T G + \lambda I \quad (6.9)$$

Rovnice udává celou trajektorii budoucích řídicích inkrementů a to je strategie otevřené smyčky. Pro uzavření smyčky se pouze první element aplikuje na systém a celý algoritmus se přepočítá pro čas $k + 1$. Pokud zapíšeme první řádek matice $(G^T G + \lambda I)^{-1} G^T$ jako K , potom aktuální řídicí inkrement může být vypočítán jako:

$$\Delta u = K(w - y_0) \quad (6.10)$$

[18][19]

6.2 Výpočet prediktoru

Důležitou úlohou je vypočítání predikce pro libovolné predikce a řídicí horizonty. Dynamika většiny procesů požaduje horizonty o délce, kde není možné jednoduše vypočítat predikce. Rekurzivní výrazy pro výpočet volné odezvy a matice G v každém bodě vzorkovací periody musí být vyjádřeny. Vyjádřila se metoda pro rekurzivní vypočítání volné odezvy i matice dynamiky. Uvažujme systém druhého řádu:

$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{B(z^{-1})}{A(z^{-1})} \quad (6.11)$$

Široce používaný model v obecném modelovém prediktivním řízení je CARIMA model, který je možné získat z nominálního modelu přidáním modelu rušení:

$$A(z^{-1})y(k) = B(z^{-1})u(k) + \frac{C(z^{-1})}{\Delta} e(k) \quad (6.12)$$

Kde $e(k)$ je neměřitelné náhodné rušení, kterému je přisuzována nulová střední hodnota s konstantní kovariancí a operátorem delta $1 - z^{-1}$. Invertovaná delta je poté integrátor. Polynom $C(z^{-1})$ bude dále zvažováno jako $C(z^{-1}) = 1$. Poté má CARIMA popis systému následující:

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k) + e(k) \quad (6.13)$$

Diferenční rovnice CARIMA modelu bez neznámého prvku $e(k)$ může být rozepsán jako:

$$\begin{aligned} y(k) = & (1 - a_1)y(k - 1) + (a_1 - a_2)y(k - 2) + a_2y(k - 3) + \\ & + b_1\Delta u(k - 1) + b_2\Delta u(k - 2) \end{aligned} \quad (6.14)$$

Bylo nutné vypočítat předpovědi o tři kroky dopředu přímým způsobem stanovením nižších predikcí na vyšší predikce. Řád systému definuje, že na výpočet predikce o krok vpřed jsou zapotřebí 3 minulé hodnoty výstupu systému. Predikce 3 kroky dopředu vypadají následovně:

$$\begin{aligned} \hat{y}(k + 1) = & (1 - a_1)y(k) + (a_1 - a_2)y(k - 1) + a_2y(k - 2) + b_1\Delta u(k) + \\ & + b_2\Delta u(k - 1) \\ \hat{y}(k + 2) = & (1 - a_1)y(k + 1) + (a_1 - a_2)y(k) + a_2y(k - 1) + b_1\Delta u(k + 1) + \\ & + b_2\Delta u(k) \\ \hat{y}(k + 3) = & (1 - a_1)y(k + 2) + (a_1 - a_2)y(k + 1) + a_2y(k) + b_1\Delta u(k + 2) + \\ & + b_2\Delta u(k + 1) \end{aligned} \quad (6.15)$$

Predikce je možné po modifikaci zapsat v maticové formě:

$$\begin{aligned} \begin{bmatrix} \hat{y}(k + 1) \\ \hat{y}(k + 2) \\ \hat{y}(k + 3) \end{bmatrix} = & \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k - 1) \\ y(k - 2) \\ \Delta u(k - 1) \end{bmatrix} = \\ = & \begin{bmatrix} & b_1 & & 0 \\ & b_1(1 - a_1) + b_2 & & b_1 \\ (a_1 - a_2)b_1 + (1 - a_1)^2b_1 + (1 - a_1)b_2 & & b_1(1 - a_1) + b_2 & \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \end{bmatrix} + \end{aligned}$$

$$\begin{aligned}
& + \begin{bmatrix} (1-a_1) & (a_1-a_2) \\ (1-a_1)^2 + (a_1-a_2) & (1-a_1)(a_1-a_2) + a_2 \\ (1-a_1)^3 + 2(1-a_1)(a_1-a_2) + a_2 & (1-a_1)^2(a_1-a_2) + a_2(1-a_1) + (a_1-a_2)^2 \end{bmatrix} \\
& \begin{bmatrix} a_2 & b_2 \\ a_2(1-a_1) & b_2(1-a_1) \\ a_2(1-a_2)^2 + (a_1-a_2)a_2 & b_2(1-a_2)^2 + (a_1-a_2)b_2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \end{bmatrix} \quad (6.16)
\end{aligned}$$

Je možné rozdělit výpočet predikcí na rekurzi volné odezvy a rekurzi výpočtu matice dynamiky. Na základě 3 minulých predikcí se vypočítá další řádek volné odezvy následovně:

$$\begin{aligned}
p_{41} &= (1-a_1)p_{31} + (a_1-a_2)p_{21} + a_2p_{11} \\
p_{42} &= (1-a_1)p_{32} + (a_1-a_2)p_{22} + a_2p_{12} \\
p_{43} &= (1-a_1)p_{33} + (a_1-a_2)p_{23} + a_2p_{13} \\
p_{44} &= (1-a_1)p_{34} + (a_1-a_2)p_{24} + a_2p_{14} \quad (6.17)
\end{aligned}$$

První řádek matice je v dalším kroku zapomenut a následující predikce je vypočítána na základě posledních 3 řádků. To zahrnuje i ten poslední vypočítaný v minulém kroku. Procedura je cyklicky opakována. Je možné vypočítat libovolný počet řádků matice. Rekurze matice dynamiky je podobná. Další prvek prvního sloupce je opakovaně vypočítán podobně jako v minulém případě a zbývající sloupce jsou posunuty, aby tvořili triangulární matici. Tato procedura se vykonává opakovaně, dokud není dosažen horizont predikce. Pokud je řídicí horizont nižší než predikční horizont, počet sloupců matice je redukován Výpočet nového elementu je následující:

$$g_4 = (1-a_1)g_3 + (a_1-a_2)g_2 + a_2g_1 \quad (6.18)$$

[22]

6.3 Modifikace pro dopravní zpoždění

Uvažujme systém s dopravní zpožděním ve tvaru:

$$G(z^{-1}) = \frac{b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} z^{-d} = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} \quad (6.19)$$

Kde d je počet kroků dopravního zpoždění

CARIMA model pro systémy s dopravním systémem má tvar:

$$\Delta A(z^{-1})y(k) = z^{-d}B(z^{-1})\Delta u(k) + e(k) \quad (6.20)$$

Aby bylo možné vypočítat řídicí zásahy, je potřeba stanovit predikce od $d + 1$ do $d + N_2$.

Prediktor je poté modifikován pro libovolné počet kroků dopravního zpoždění:

$$\begin{aligned} \begin{bmatrix} \hat{y}(k + d + 1) \\ \hat{y}(k + d + 2) \\ \hat{y}(k + d + 3) \end{bmatrix} &= \begin{bmatrix} p_{(1+d)1} & p_{(1+d)2} & p_{(1+d)3} \\ p_{(2+d)1} & p_{(2+d)2} & p_{(2+d)3} \\ p_{(3+d)1} & p_{(3+d)2} & p_{(3+d)3} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k - 1) \\ y(k - 2) \end{bmatrix} + \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \end{bmatrix} \\ &+ \begin{bmatrix} g_{1+d-1} & g_{2+d-1} & p_{(1+d)4} \\ g_{2+d-1} & g_{3+d-1} & p_{(2+d)4} \\ g_{3+d-1} & g_{4+d-1} & p_{(3+d)4} \end{bmatrix} \begin{bmatrix} \Delta u(k - 1) \\ \Delta u(k - 2) \\ \Delta u(k - 3) \end{bmatrix} \end{aligned} \quad (6.21)$$

Rekurzivní výpočet matic je analogií k rekurzivnímu počítání, které je popsáno pro systém druhého řádu bez dopravního zpoždění. [19]

II. PRAKTICKÁ ČÁST

7 ANALÝZA SOUČASNÉHO ŘÍZENÍ

Část míchací linky pro navažování materiálu se skládá ze 4 důležitých částí:

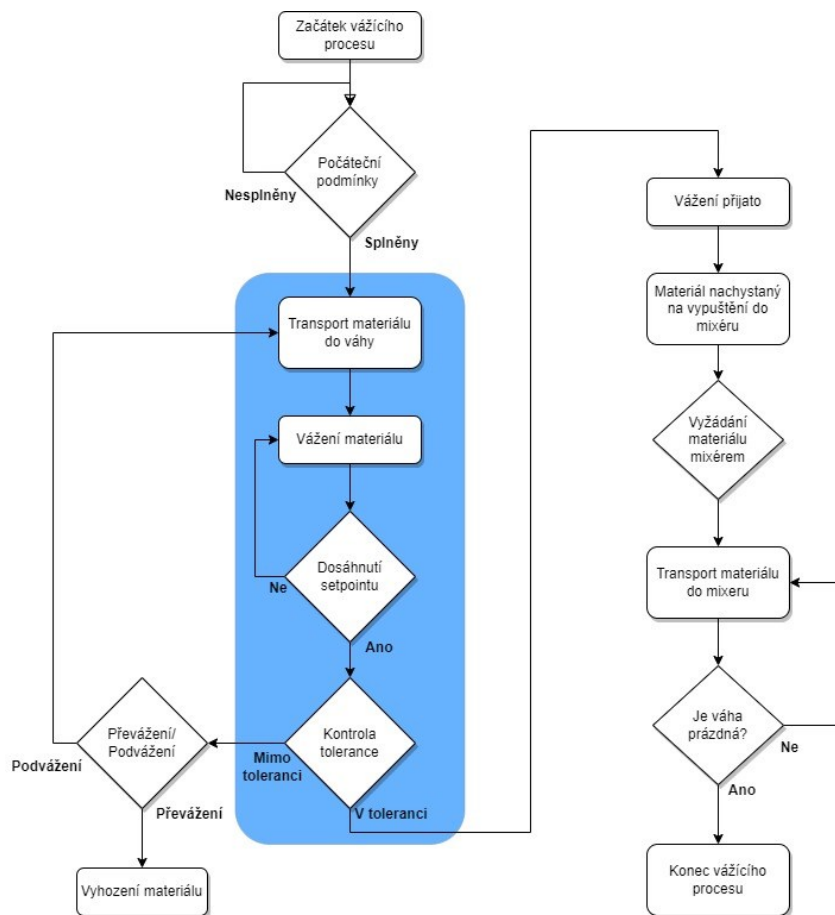
- Load cells
- Dopravník
- Silo
- Váha

Chemikálie často používané v míchacím procesu jsou uchovávány v silu. Některé chemikálie ve formě prášku mají nízký bod tavení a mohly by ucpat šnekový dopravník. Z tohoto důvodu se pro přepravu chemikálií do váhy používá pásový dopravník. Ten je poháněn pomocí motoru, který je řízený frekvenčním měničem. Měřicí váha kuželovitého tvaru je zavěšena na 4 měřicích zařízení Siwarex load cells.

Stávající problém navažování je překmit mimo toleranci. Takový překmit zapříčiní, že je potřeba operátora, aby ručně odstranil materiál z váhy a dochází k časové prodlevě míchacího procesu a k finančním a materiálním ztrátám. Cílem navrženého řízení je odstranění překmitu, aby nebyl nutný ruční zásah operátora.

7.1 Vážicí proces

Vážicí proces začíná po stažení dat z receptu a po splnění všech počátečních podmínek. Následuje otevření klapky sila a transport sypkého materiálu do váhy pomocí pásového dopravníku. Mezi vážením jednotlivých chemikálií se musí váha vynulovat. Vážení je prováděno průměrováním výsledků z měřicích zařízeních. Pro přesnost měření je potřeba mít správně zkalibrovanou váhu. Kalibrace je během dne prováděna 3x pomocí závaží o předem definované hmotnosti. Vážení pokračuje do doby, než se dosáhne požadované hodnoty, která se pohybuje v toleranci. Pokud je však váha mimo toleranci, musí operátor odebrat materiál z váhy. To se provádí přepnutím výstupní klapky na pytlování, kdy se materiál vysype do odpadového pytle.



Obrázek 10 Vývojový diagram vážicího cyklu

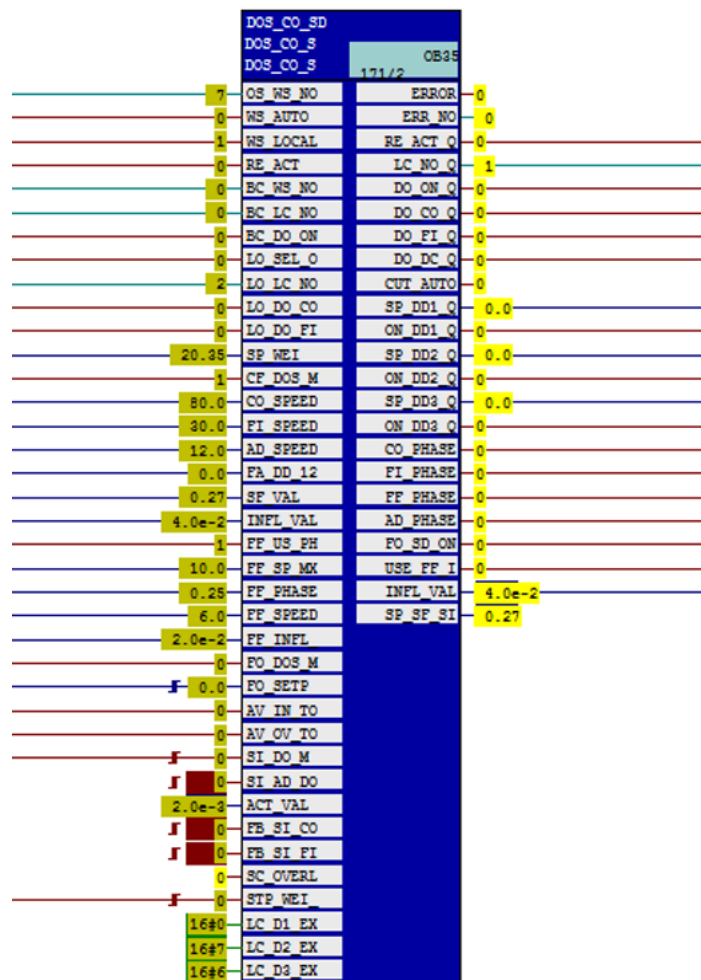
7.2 Počáteční podmínky

Mezi počáteční podmínky patří provedená kalibrace. Dále se kontroluje, zda je váha prázdná a připravena na další vážení. Také se kontroluje, v jakém operačním módu je váha nastavena. Dále se kontrolují chyby, aktivní recept a mnoho dalšího.

7.3 Řízení dopravníku

Dávkování materiálu se dělí na 4 fáze:

- Coarse - Hrubé dávkování
- Fine – jemné dávkování
- Finest – jemnější dávkování
- After dosing – dovažování



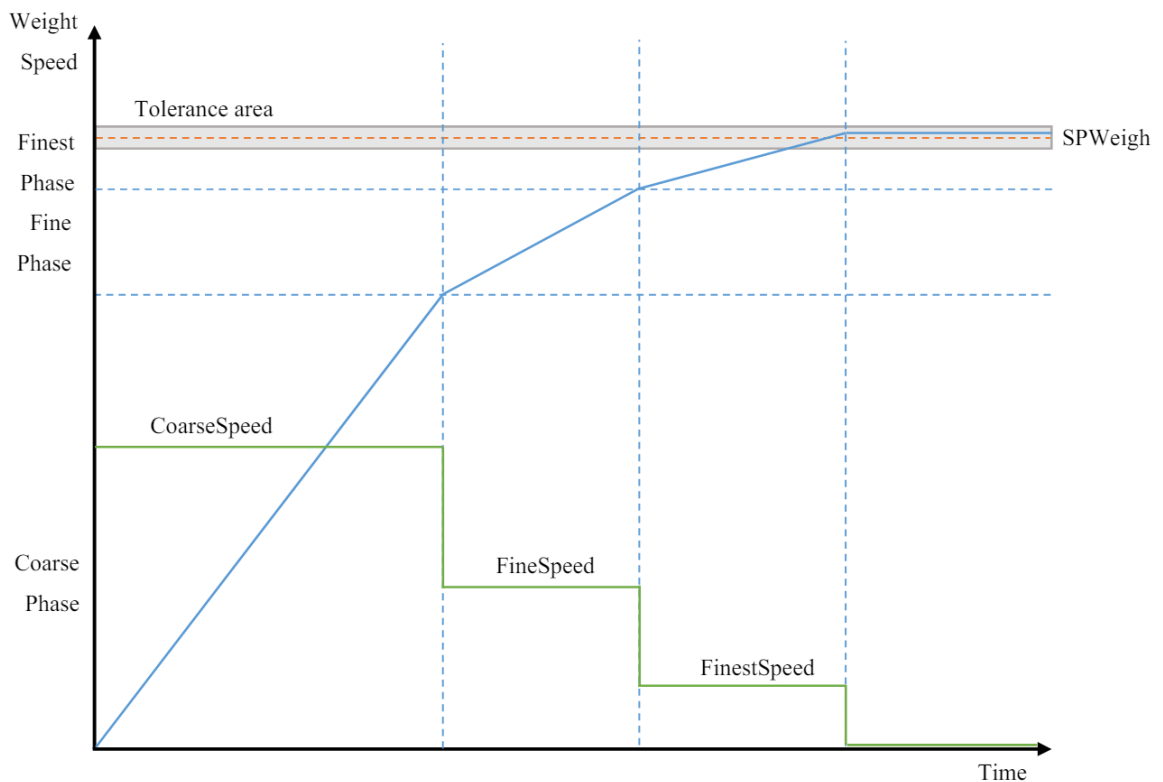
Obrázek 11 Funkční blok pro určení fáze vážení

Tento funkční blok má za úkol určit fázi vážicího procesu. Pro jednotlivé fáze vážicího procesu se používají různé rychlosti a proto je potřeba fáze rozlišit. To se dělá pomocí hladin, které se získávají porovnáním aktuální hodnoty s požadovanou hodnotou. Od SETPOINTu se odečítají hodnoty (konstanty), které následně určují hladiny. V podmínce pro jednotlivé fáze se tedy kontroluje operační mód, zda je vážící proces aktivní a následně porovnání aktuální a požadované váhy. Výstupem podmínek jsou tedy stavové bity jednotlivých fází vážení. Stavové bity jednotlivých fází jdou do dalšího bloku, který podle těchto stavových bitů zvolí rychlost danou z receptu.


```

// Auswertung Dosierphase
IF STP_WEI_RUN AND ST_AD_DO THEN // #2# e.) Dosing in Phase 'After Dosing'
CO_PHASE_Q := FALSE;
FI_PHASE_Q := FALSE;
FF_PHASE_Q := FALSE;
AD_PHASE_Q := TRUE;
FF_INACT := FALSE;
ELSIF STP_WEI_RUN AND FF_US_PH_Q AND (SP_WEI < FF_SP_MX) AND (ACT_VAL >= (SP_WEI - FF_PHASE)) THEN // #2# d.) Dosing in Phase 'Finest' [activ]
CO_PHASE_Q := FALSE;
FI_PHASE_Q := FALSE;
FF_PHASE_Q := TRUE;
AD_PHASE_Q := FALSE;
FF_INACT := FALSE;
ELSIF STP_WEI_RUN AND FF_US_PH_Q AND (SP_WEI >= FF_SP_MX) AND (ACT_VAL >= (SP_WEI - FF_PHASE)) THEN // #2# c.) Dosing in Phase 'Finest' [inactiv]
CO_PHASE_Q := FALSE;
FI_PHASE_Q := TRUE;
FF_PHASE_Q := FALSE;
AD_PHASE_Q := FALSE;
FF_INACT := TRUE;
ELSIF STP_WEI_RUN AND ((WS_AUTO AND FB_SI_CO) OR (WS_LOCAL AND (ACT_VAL < (SP_WEI - SF_VAL - INFL_VAL)))) THEN // #2# a.) Dosing in Phase 'Coarse'
CO_PHASE_Q := TRUE;
FI_PHASE_Q := FALSE;
FF_PHASE_Q := FALSE;
AD_PHASE_Q := FALSE;
FF_INACT := FALSE;
ELSIF STP_WEI_RUN AND ((WS_AUTO AND FB_SI_FI) OR (WS_LOCAL AND (ACT_VAL >= (SP_WEI - SF_VAL - INFL_VAL)))) THEN // #2# b.) Dosing in Phase 'Fine'
CO_PHASE_Q := FALSE;
FI_PHASE_Q := TRUE;
FF_PHASE_Q := FALSE;
AD_PHASE_Q := FALSE;
FF_INACT := FALSE;
ELSE // #2#
CO_PHASE_Q := FALSE;
FI_PHASE_Q := FALSE;
FF_PHASE_Q := FALSE;
AD_PHASE_Q := FALSE;
FF_INACT := FALSE;
END_IF; // #2#
    
```

Obrázek 12 Část kódu pro určení fáze vážení



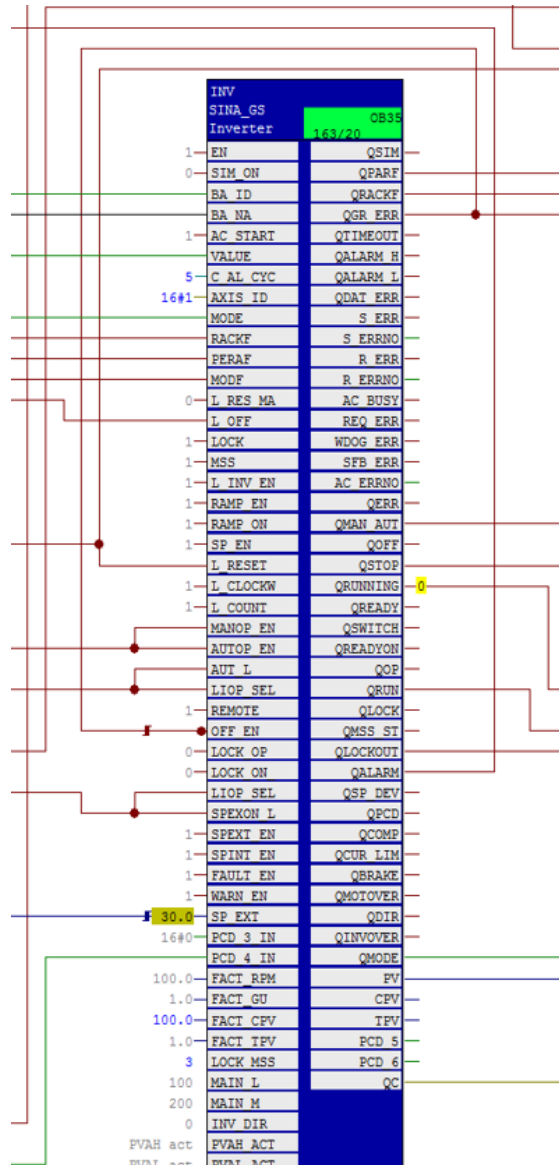
Obrázek 13 Průběh vážicího procesu – rychlost a váha

Na obrázku lze vidět průběh vážení v závislosti na rychlosti. K vidění jsou 4 různé fáze. Ty udávají rychlosti, které se získávají z receptu a na obrázku č.11 je můžeme najít pod proměnnými:

- CO_SPEED – 80%
- FI_SPEED – 30%
- AD_SPEED – 12%
- FF_SPEED – 6%

Dalšími důležitými proměnnými bloku jsou:

- SP_WEI – udává požadovanou váhu
- ACT_VAL – aktuální váha – z měřicích jednotek Siwax
- SF_VAL – hodnota pro pomalé dávkování v kg
- INFL_VAL – hodnota volného pádu v kg
- FF_PHASE – úroveň, kdy začíná FF fáze v kg
- DO_ON_Q – stav vážení ON/OFF



Obrázek 14 Blok měniče

Měníč přemění vstupní hodnotu SP_EXT na výstupní hodnotu PV. Respektive SP_EXT je požadavek na rychlost, která je zadaná v % z a PV je rychlost, kterou měnič poskytuje.

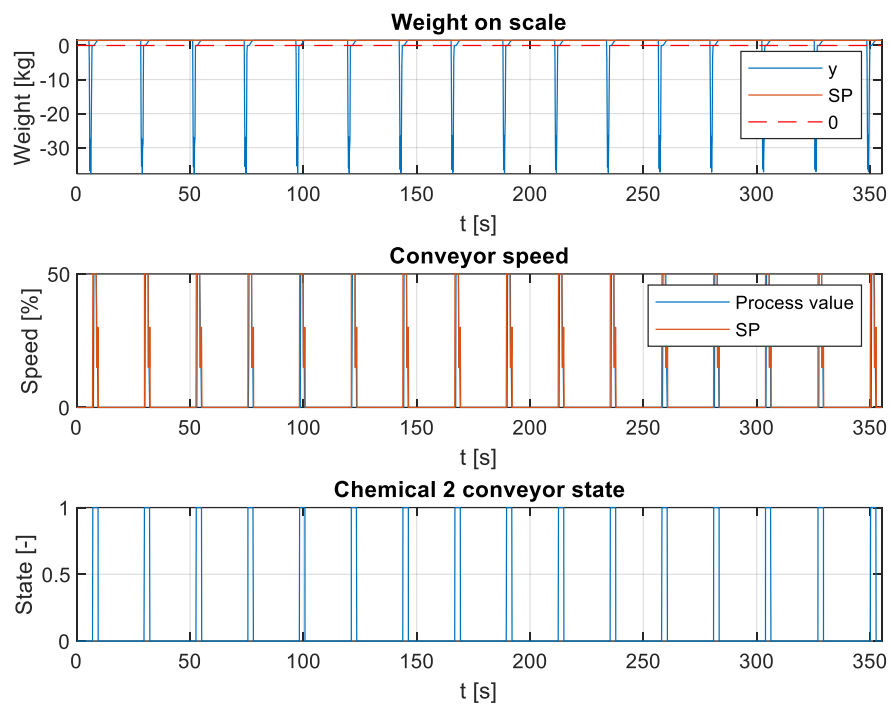
8 IDENTIFIKACE NAMĚŘENÝCH DAT

8.1 Měření dat

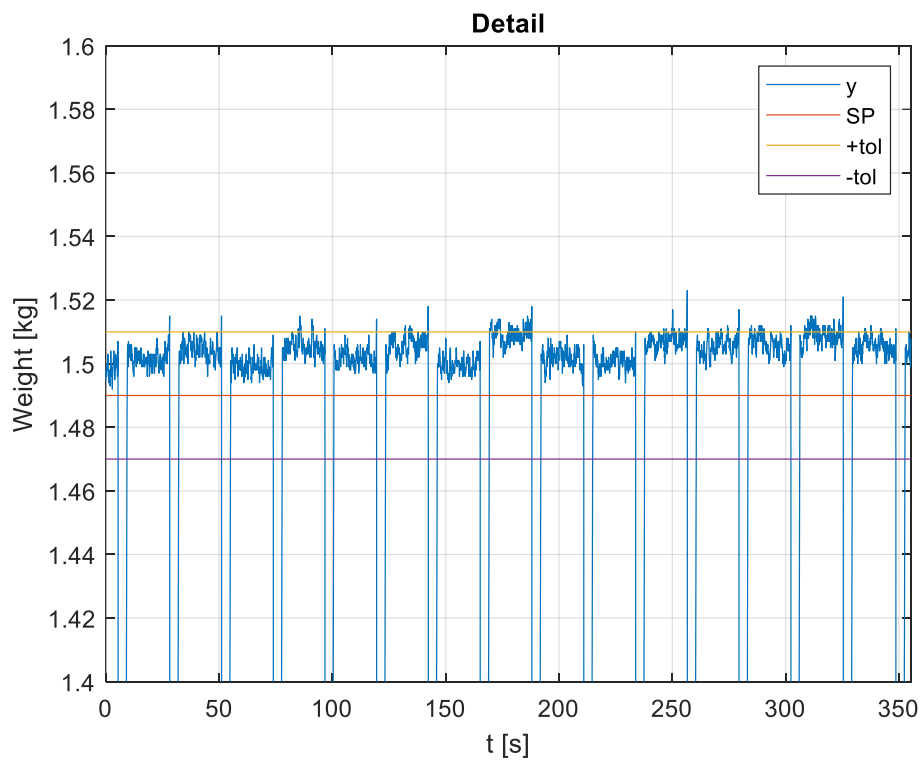
Data z míchací linky byla získána pomocí systému iba AG, kdy se z projektu vytvořila adresová mapa a následně se pomocí symbolických proměnných zvolily ty proměnné, které byly potřeba naměřit. Bylo naměřeno celkem 16 průběhů vážení. Konkrétně se měřila hodnota váhy, rychlost dopravníku a stavová proměnná, která udává, zda je dopravník v chodu či ne. Naměřená data z ibaPDA se exportovala do textového souboru. Tento textový soubor se načel do Excelu, kde už s daty bylo dále možné pracovat. Konkrétně byla data načtena do Matlabu, kde se provedla konverze datového typu pro snazší práci s daty. Data se rozdělila a průběh měření se vykreslil.

V průběhu těchto měření se došlo i k zaznamenání špatného měření. Známé věci z receptu:

- SETPOINT - 1,490 *kg*
- Tolerance - $\pm 0,02$ *kg*
- CO_SPEED - 50%
- FI_SPEED - 15%



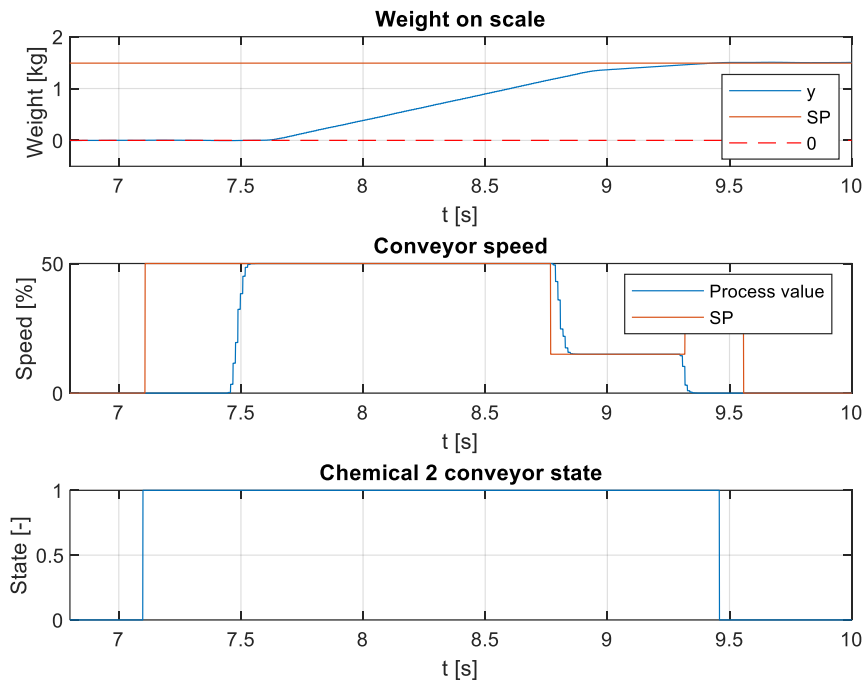
Obrázek 15 Naměřená data



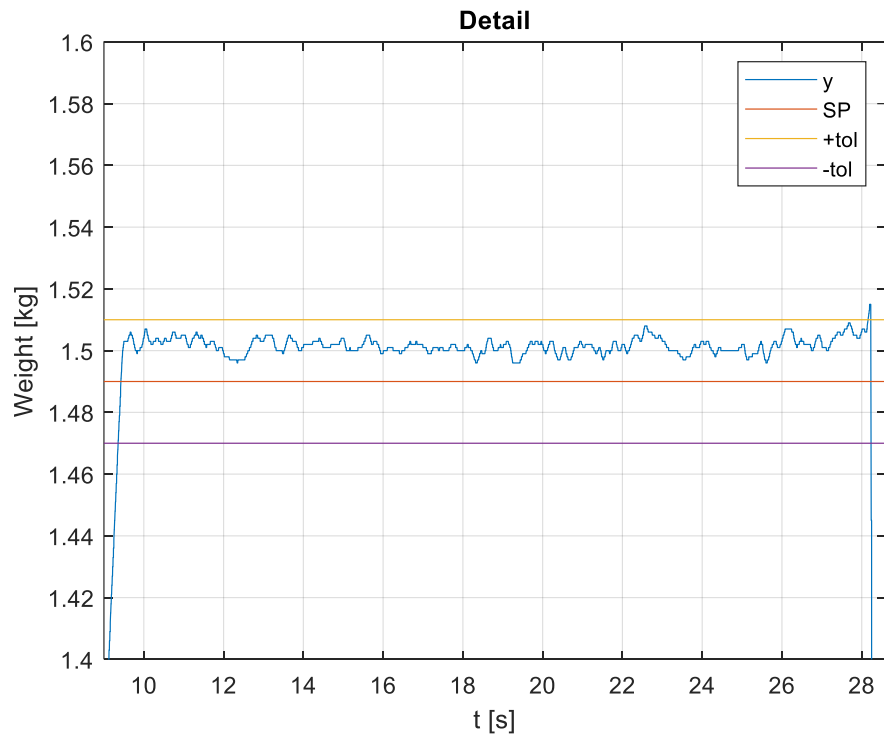
Obrázek 16 Detail – okolí setpointu

8.1.1 Průběh správného vážení

Na následujících obrázcích je zobrazen průběh správně naváženého materiálu:



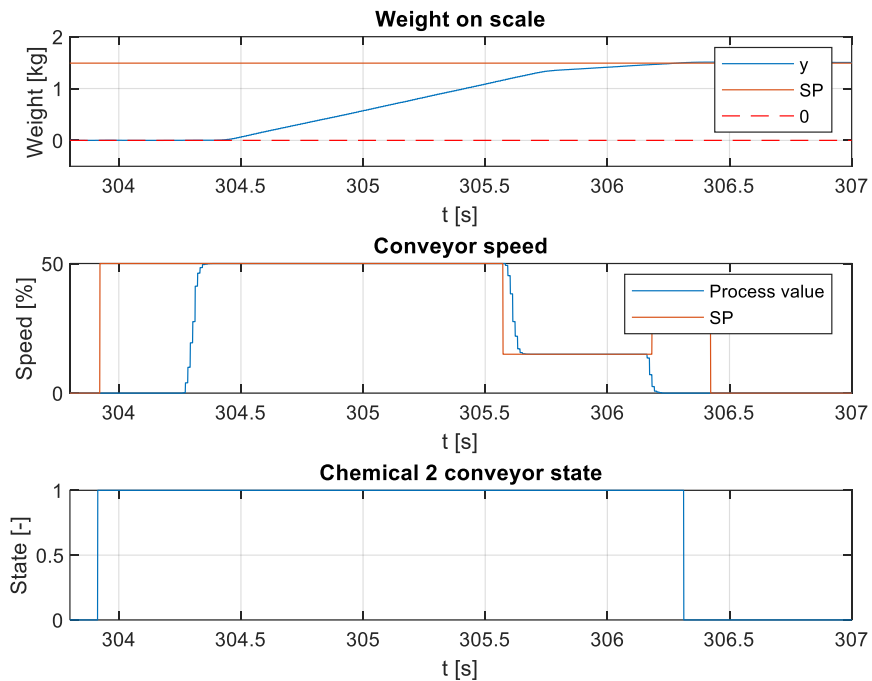
Obrázek 17 Průběh přijatého vážení



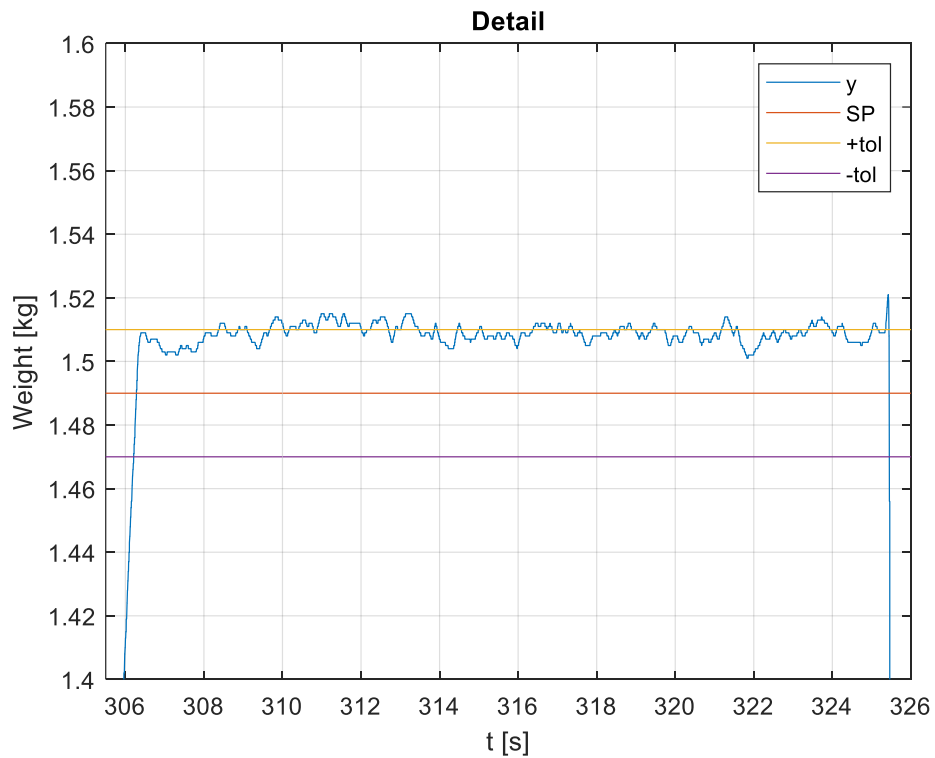
Obrázek 18 Detail přijatého vážení – hmotnost na váze

8.1.2 Průběh špatného vážení – mimo horní toleranci

Na následujících obrázcích je zobrazen průběh naváženého materiálu mimo toleranci:



Obrázek 19 Průběh špatného měření



Obrázek 20 Detail špatného vážení – hmotnost na váze

8.2 Aplikace ARX modelu pro odhad parametrů

8.2.1 ARX model bez dopravního zpoždění

Pro odhad parametrů pomocí ARX modelu je uvažován systém 2. řádu, tedy:

$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Vektor neznámých parametrů modelu je tedy následující:

$$\hat{\theta} = [a_1 \quad a_2 \quad b_1 \quad b_2]^T$$

Vytvoří se tedy regresní vektor:

$$F = \begin{bmatrix} -y(2) & -y(1) & u(2) & u(1) \\ -y(3) & -y(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & u(N-1) & u(N-2) \end{bmatrix}$$

Dále se vytvoří vektor hodnot výstupu systému:

$$y^T = [y(3) \quad y(4) \quad \dots \quad y(N)]$$

kde N je počet vzorků naměřených dat vstupu a výstupu.

Řešením následujícího vztahu dostaneme odhady parametrů:

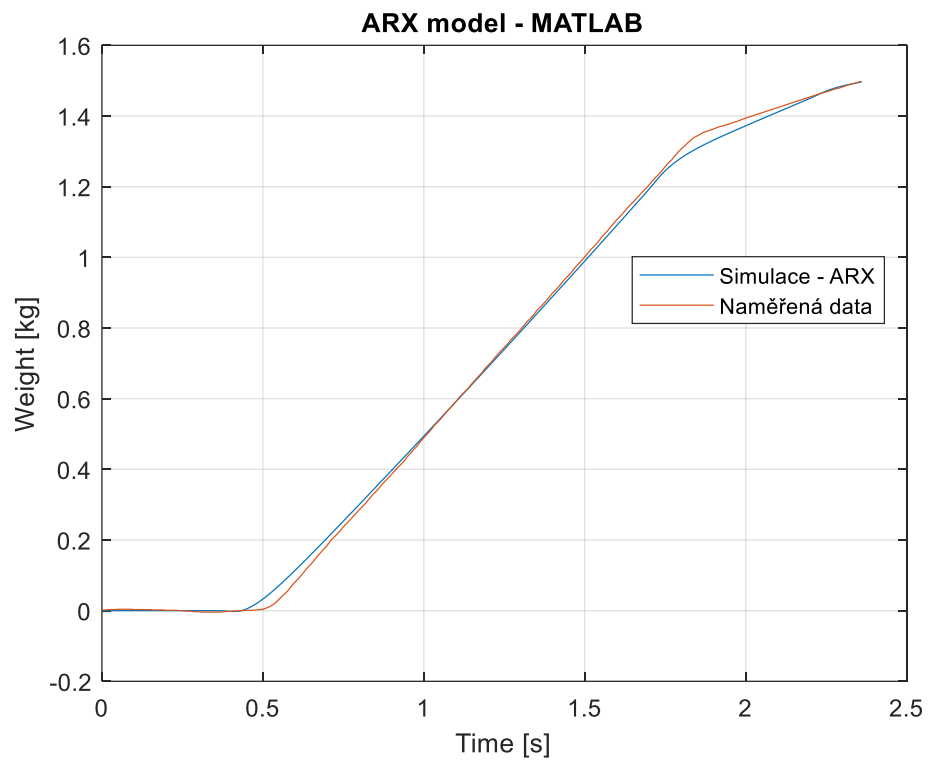
$$\hat{\theta} = (F^T F)^{-1} F^T y$$

Pro porovnání je využita funkce ARX v Matlabu. Výsledky 1. průběhu vypadají následovně:

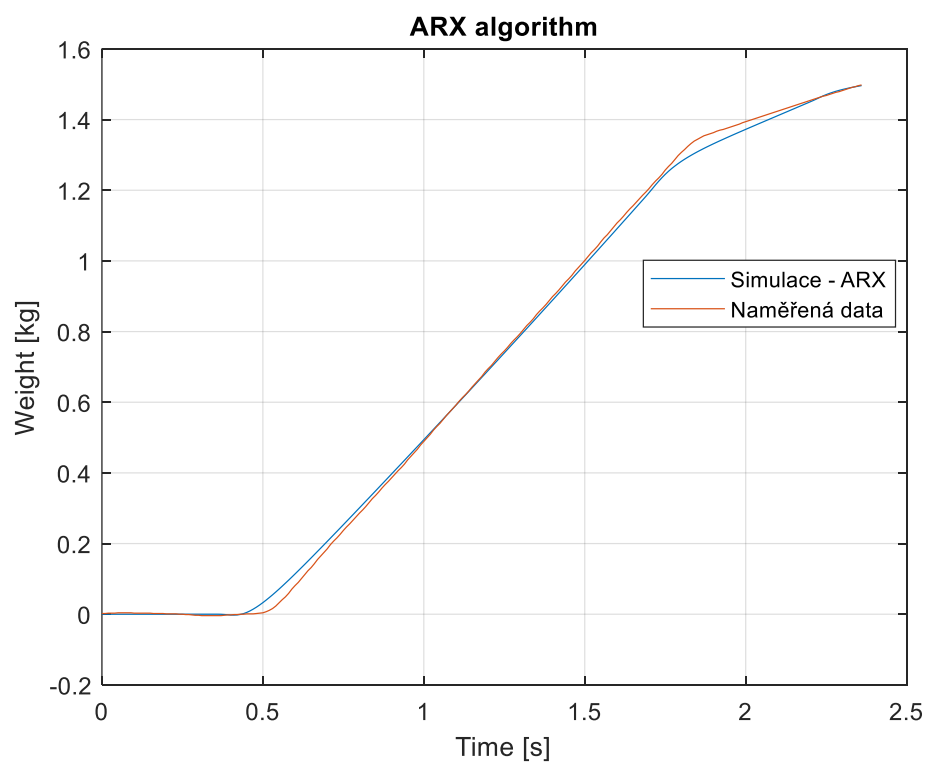
$$\hat{\theta}_{MAT} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1,857 \\ 0,8568 \\ -6,12 \cdot 10^{-5} \\ 8,813 \cdot 10^{-5} \end{bmatrix} \rightarrow G_{MAT}(z^{-1}) = \frac{-6,12 \cdot 10^{-5} z^{-1} + 8,813 \cdot 10^{-5} z^{-2}}{1 - 1,857 z^{-1} + 0,8568 z^{-2}}$$

Odhady parametrů získané z algoritmu odpovídajícímu ARX modelu:

$$\hat{\theta}_{ALG} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1,857 \\ 0,8568 \\ -6,119 \cdot 10^{-5} \\ 8,813 \cdot 10^{-5} \end{bmatrix} \rightarrow G_{ALG}(z^{-1}) = \frac{-6,119 \cdot 10^{-5} z^{-1} + 8,813 \cdot 10^{-5} z^{-2}}{1 - 1,857 z^{-1} + 0,8568 z^{-2}}$$



Obrázek 21 Porovnání naměřených dat se simulovaným průběhem získaným pomocí funkce ARX v Matlabu



Obrázek 22 Porovnání naměřených dat se simulovaným průběhem získaným pomocí vlastního algoritmu pro ARX

8.2.2 ARX model s dopravním zpožděním

Parametry získané pomocí algoritmu jsou shodné jako parametry získané pomocí ARX funkce v Matlabu. I přes tuto shodu je v porovnáváních průběžích viditelný rozdíl. Ze simulace systému s parametry získanými z ARX algoritmu lze pozorovat, že simulovaný průběh začíná dříve o 0,1 s. Je tedy patrné, že se jedná o systém s dopravním zpožděním.

Modifikuje se tedy matice F o dopravní zpoždění:

$$F = \begin{bmatrix} -y(2+d) & -y(1+d) & u(2) & u(1) \\ -y(3+d) & -y(2+d) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & u(N-d-1) & u(N-d-2) \end{bmatrix}$$

A tedy tvar přenosu ARX modelu je následující:

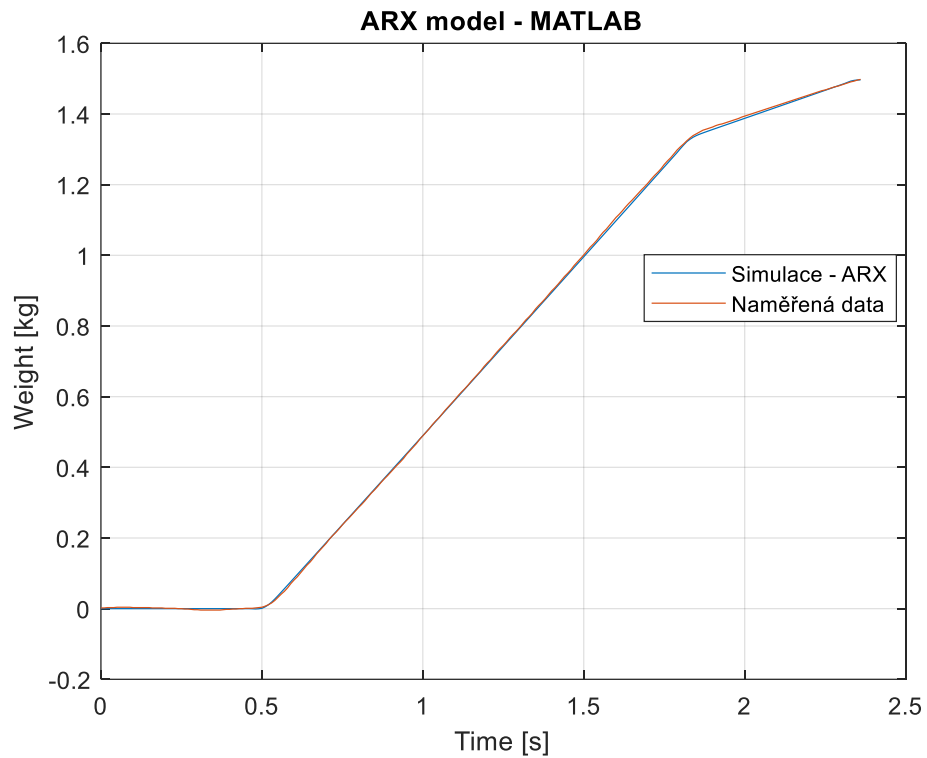
$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-d} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-10}$$

Odhady parametrů získané z ARX funkce z Matlabu:

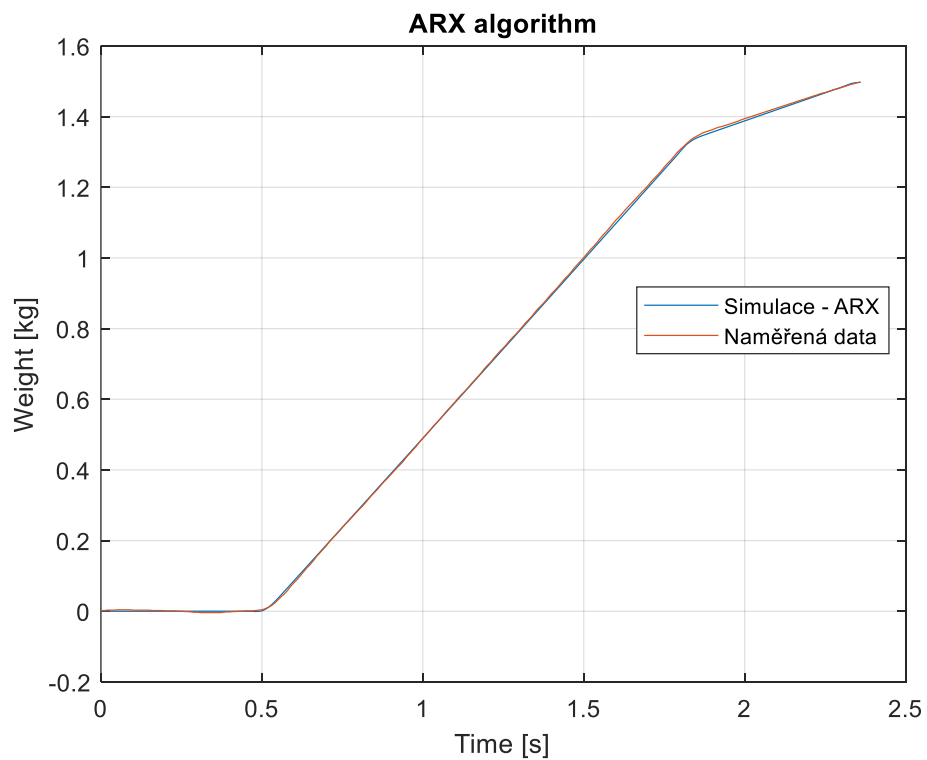
$$\hat{\theta}_{MAT} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1,355 \\ 0,3546 \\ -1,048 \cdot 10^{-4} \\ 2,345 \cdot 10^{-4} \end{bmatrix} \rightarrow G_{MAT}(z^{-1}) = \frac{-1,048 \cdot 10^{-4} z^{-1} + 2,345 \cdot 10^{-4} z^{-2}}{1 - 1,355 z^{-1} + 0,3546 z^{-2}}$$

Odhady parametrů získané z algoritmu odpovídajícímu ARX modelu:

$$\hat{\theta}_{ALG} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1,359 \\ 0,359 \\ -1,036 \cdot 10^{-4} \\ 2,324 \cdot 10^{-4} \end{bmatrix} \rightarrow G_{ALG}(z^{-1}) = \frac{-1,036 \cdot 10^{-4} z^{-1} + 2,324 \cdot 10^{-4} z^{-2}}{1 - 1,359 z^{-1} + 0,359 z^{-2}}$$



Obrázek 23 Porovnání naměřených dat se simulovaným průběhem získaným ARX funkcí – s dopravním zpožděním



Obrázek 24 Porovnání naměřených dat se simulovaným průběhem získaným pomocí vlastního algoritmu pro ARX- s dopravním zpožděním

9 PRŮBĚŽNÁ IDENTIFIKACE

9.1 RARX bez dopravního zpoždění

Pro průběžnou identifikaci modelu ARX je vhodné použít jeho rekurzivní metodu, která se používá pro online odhad parametrů. Je tedy potřeba vytvořit regresní vektor:

$$\phi^T(k) = [-y(k-1) \quad -y(k-2) \quad u(k-1) \quad u(k-2)]$$

Vypočítá se chyba predikce:

$$\hat{e}(k) = y(k) - \phi^T(k)\hat{\theta}(k-1)$$

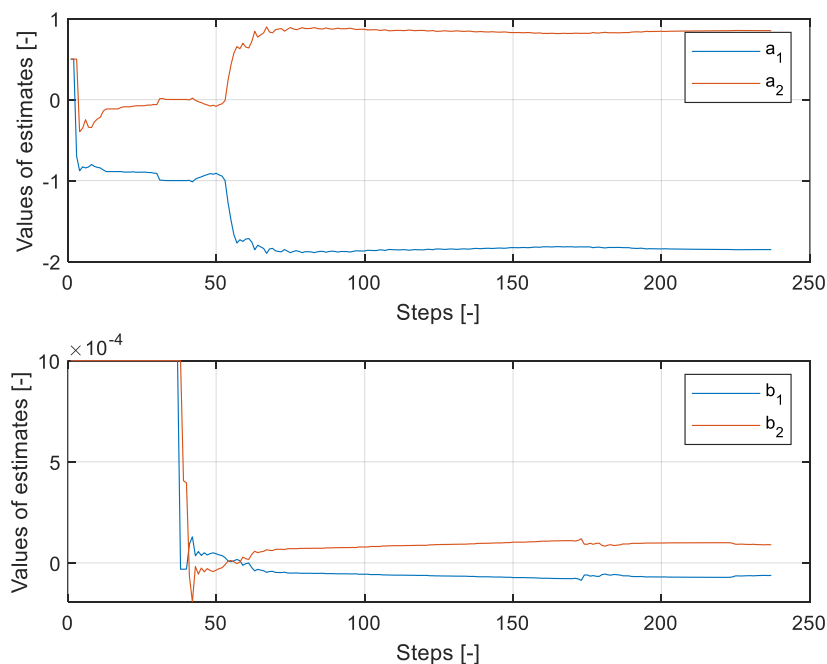
Dále se aktualizuje kovarianční matice:

$$C(k) = C(k-1) - \frac{C(k-1)\phi(k)\phi^T(k)C(k-1)}{1 + \phi^T(k)C(k-1)\phi(k)}$$

A jako poslední se aktualizuje vektor odhadů parametrů:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + C(k)\phi(k)\hat{e}(k)$$

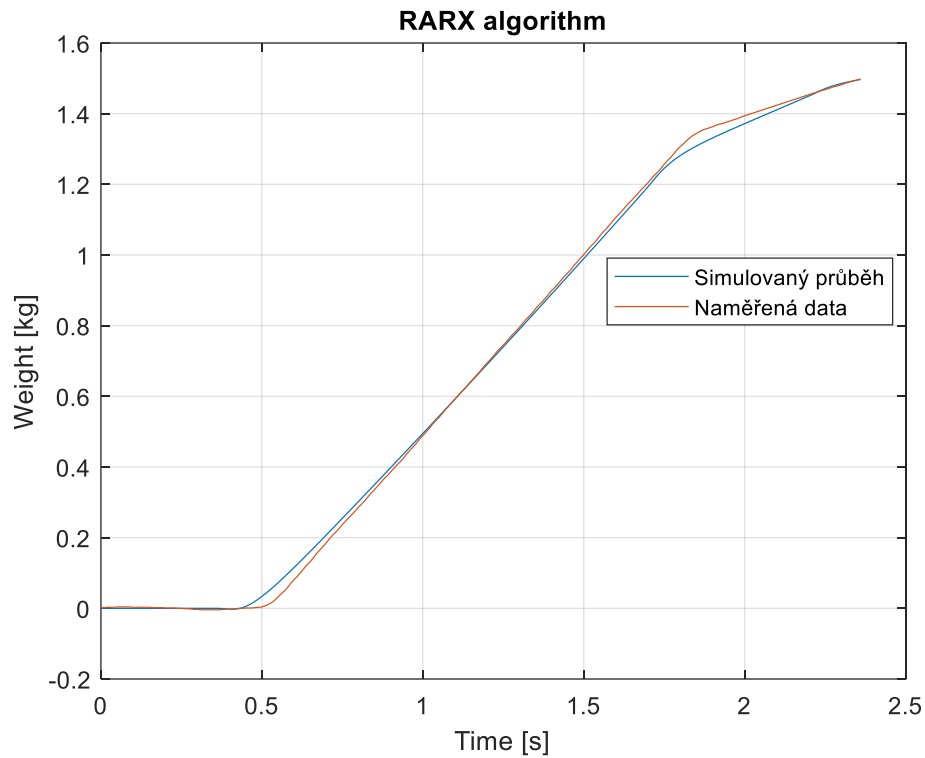
Pro počáteční hodnoty matice C na hlavní diagonále byla zvolena hodnota 10^6 . Jako počáteční odhady byly zvoleny: $a_1 = 0,5$; $a_2 = 0,5$; $b_1 = 0,001$; $b_2 = 0,001$. Pro otestování byl použit opět první naměřený průběh.



Průběh odhadů parametrů modelu – RARX bez dopravního zpoždění

Poslední hodnoty odhadů parametrů získané pomocí RARX:

$$\hat{\theta}^T = [-1,8520 \quad 0,8519 \quad -6,2518 \cdot 10^{-5} \quad 9,0271 \cdot 10^{-5}]$$



Obrázek 25 Porovnání naměřených dat a simulovaného průběhu

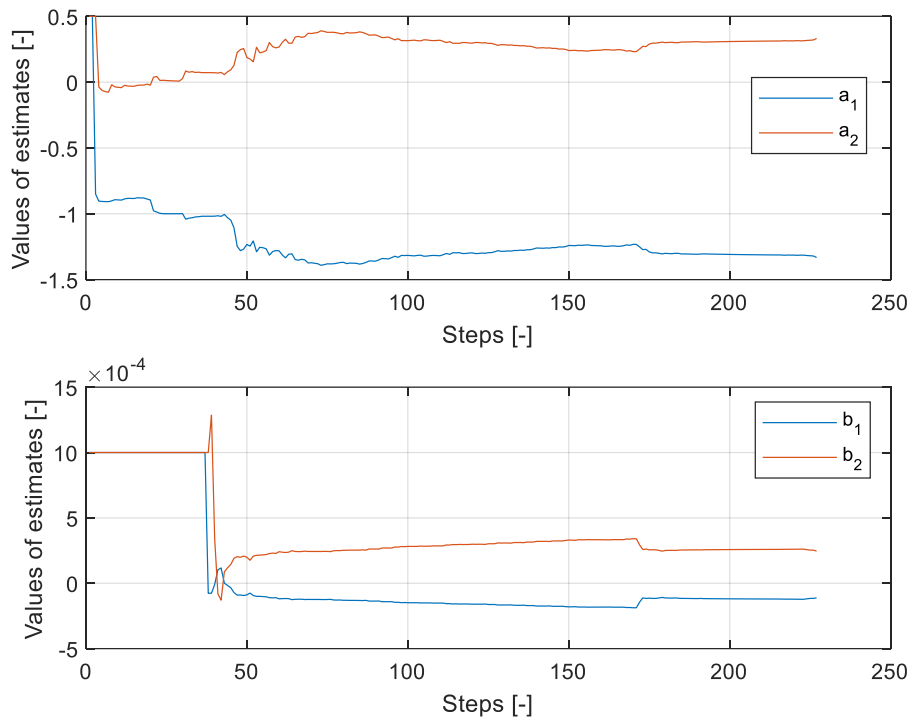
9.2 RARX s dopravním zpožděním

Jak již bylo zmíněno výše, posunutí dat nasvědčuje, že se jedná o systém s dopravním zpožděním. Tedy je potřeba modifikovat RARX algoritmus. Jediná úprava se provede v regresním vektoru:

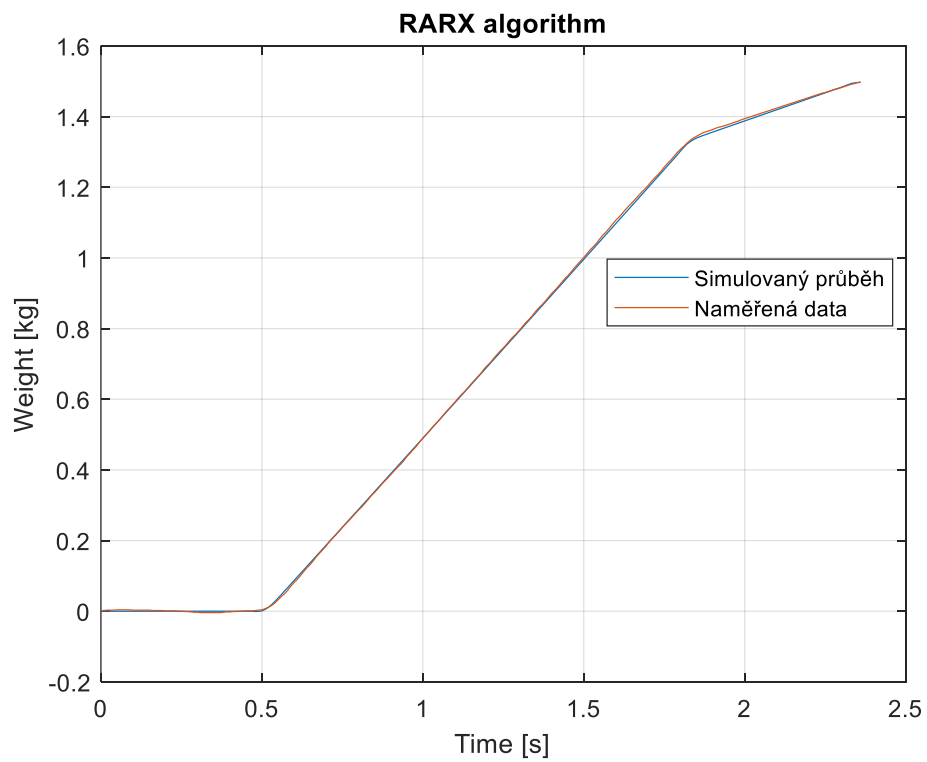
$$\phi^T(k) = [-y(k-1) \quad -y(k-2) \quad u(k-d-1) \quad u(k-d-2)]$$

Pro porovnání se opět použije první průběh. Získané odhady parametrů pomocí RARX s dopravním zpožděním o hodnotě 10, tj. 0,1 s:

$$\hat{\theta}^T = [-1,3396 \quad 0,3396 \quad -1,088 \cdot 10^{-4} \quad 2,4144 \cdot 10^{-4}]$$



Obrázek 26 Průběh odhadů parametrů modelu – RARX s dopravním zpožděním



Obrázek 27 Porovnání naměřených dat a získaného průběhu s dopravním zpožděním o hodnotě 0,1 s

10 NÁVRH ZPŮSOBU ŘÍZENÍ

Z důvodu dopravního zpoždění mezi vstupem a výstupem je vhodné použít prediktivní řízení. Jelikož model systému je znám a jeho parametry se budou průběžně aktualizovat pomocí RARX, lze použít model prediktivního řízení (MPC). Důvodem zvolení self-adaptive regulátoru jsou změny parametrů procesu, které se mění vlivem změny počasí a vlhkosti. Z tohoto důvodu byl zvolen model prediktivního řízení CARIMA (Controlled AutoRegressive Integrated Moving Average).

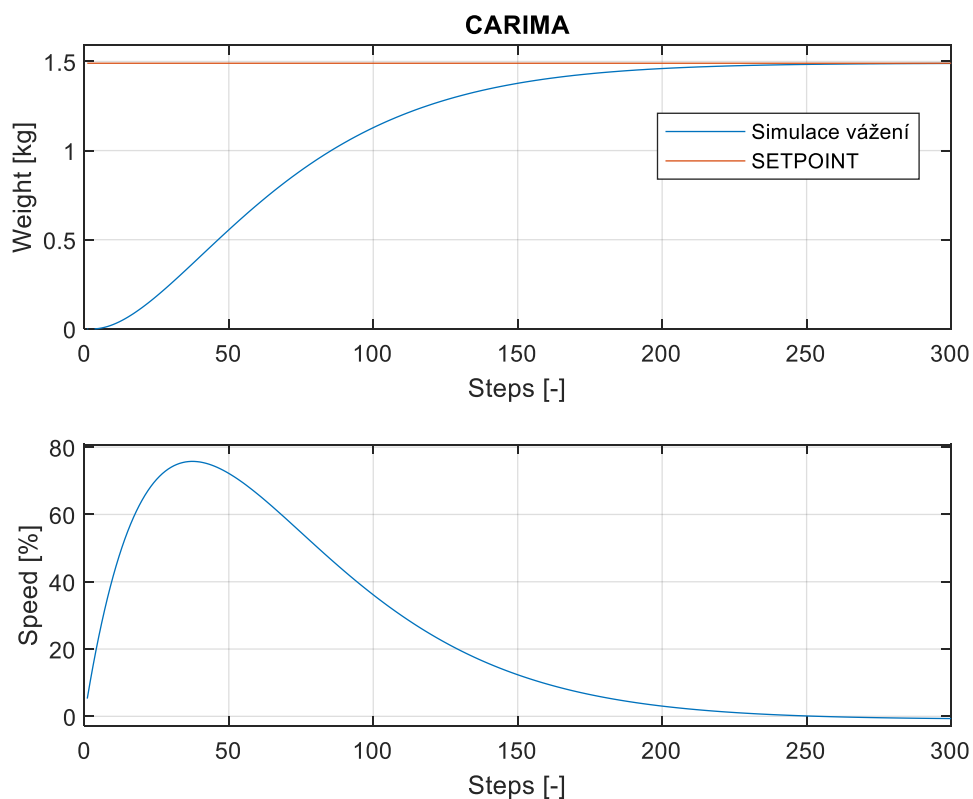
10.1 CARIMA bez dopravního zpoždění

Pro simulaci v Matlabu se použijí odhady parametrů, které byly identifikovány pomocí RARX bez dopravního zpoždění:

$$\hat{\theta}^T = [-1,8520 \quad 0,8519 \quad -6,2518 \cdot 10^{-5} \quad 9,0271 \cdot 10^{-5}]$$

Simulací byl odzkoušen simulační pochod a zároveň se zkoušely různé hodnoty řídicích parametrů. Nejlepší průběh byl dosažen při následujících parametrech:

$$N_1 = 1 \quad N_2 = 120 \quad N_u = 20 \quad \lambda = 0.1$$



Obrázek 28 Průběh predikce vážení a akčního zásahu - CARIMA

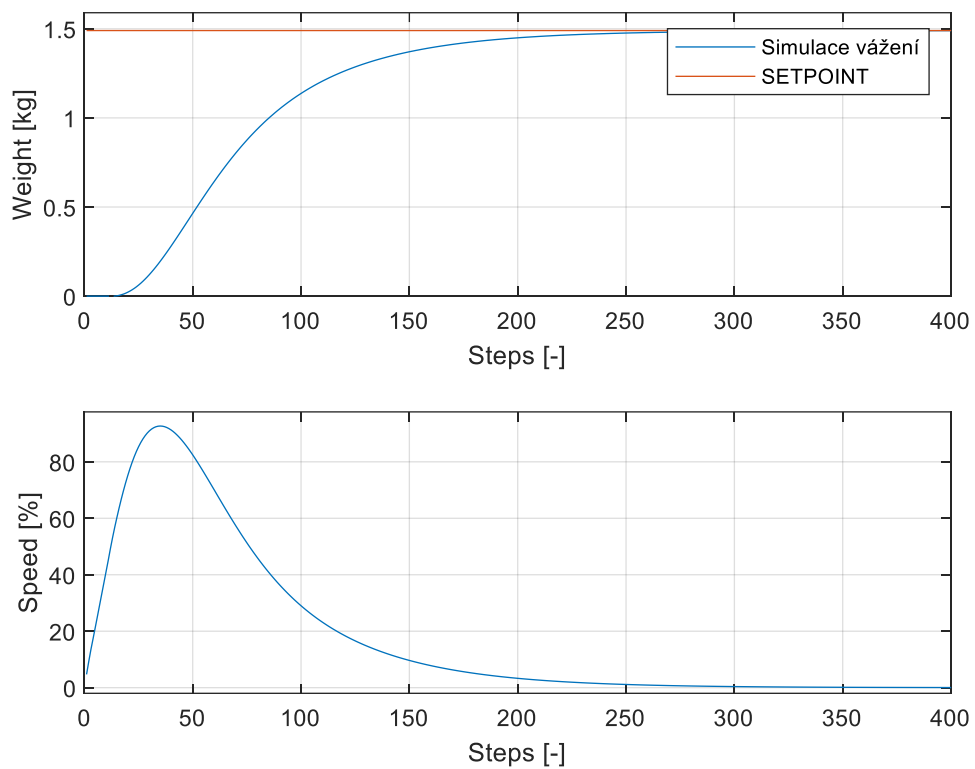
10.2 CARIMA s dopravním zpožděním

Pro simulaci v Matlabu se použijí odhady parametrů, které byly identifikovány pomocí RARX s dopravním zpožděním:

$$\hat{\theta}^T = [-1,3396 \quad 0,3396 \quad -1,088 \cdot 10^{-4} \quad 2,4143 \cdot 10^{-4}]$$

Simulací byl odzkoušen simulační pochod a zároveň se zkoušely různé hodnoty řídicích parametrů. Nejlepší průběh byl dosažen při následujících parametrech:

$$N_1 = 1 \quad N_2 = 120 \quad N_u = 30 \quad \lambda = 0.1$$



Obrázek 29 Průběh predikce vážení a akčního zásahu – CARIMA s dopravním zpožděním

11 SIMULAČNÍ OVĚŘENÍ V MATLAB SIMULINK

Pro simulační ověření byl zvolen Matlab Simulink. Tedy do Simulinku bylo potřeba převést simulaci systému, RARX a CARIMA. Pro systém modelu byl použit diskretní filtr, kde se dají zadat parametry modelu. Algoritmy RARX a CARIMA byly vloženy do bloku Matlab Function a dále upraveny tak, aby bylo možné algoritmy spustit v Simulinku. Po jejich úpravě a zajištění správného chodů algoritmů byly vytvořeny RARX a CARIMA subsystémy.

11.1 Bez dopravního zpoždění

Řídicí parametry byly nastaveny podle simulačního ověření z Matlab scriptu.

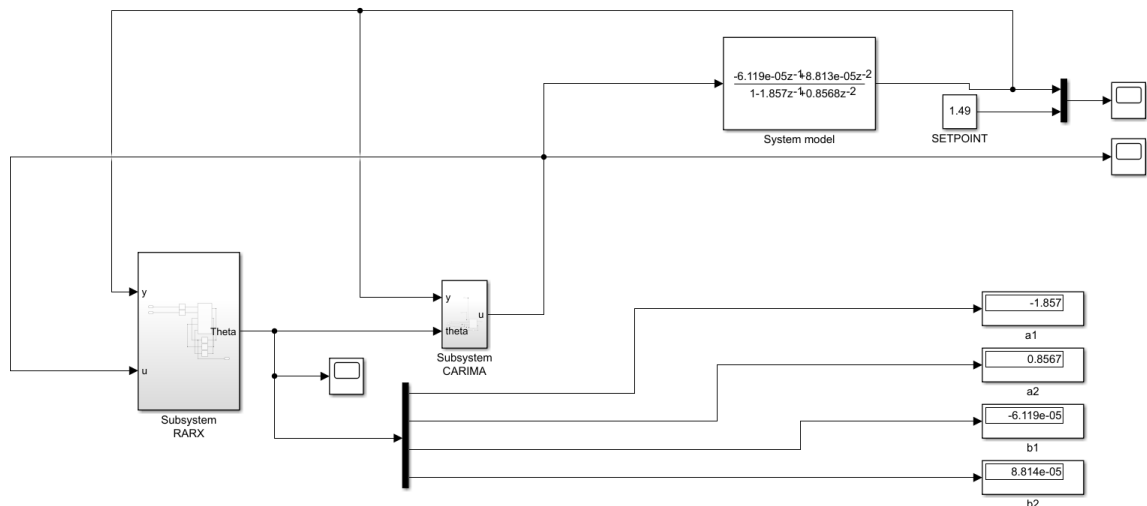
Počáteční odhady parametrů:

$$\hat{\theta}^T = [-1,8 \quad 0,8 \quad -6 \cdot 10^{-5} \quad 8 \cdot 10^{-5}]$$

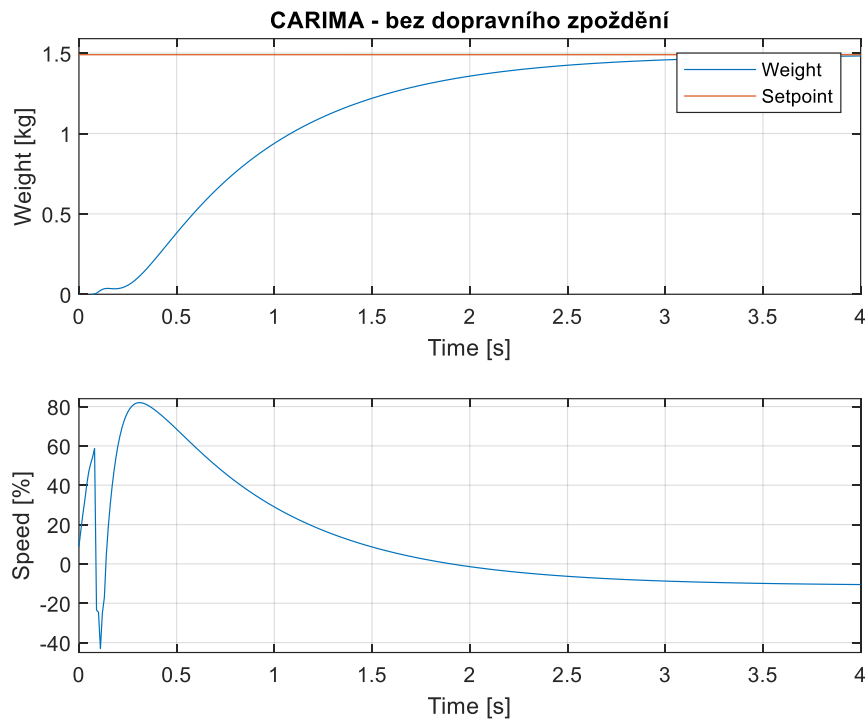
Přenos systému je převzat z jednorázové identifikace:

$$G(z^{-1}) = \frac{-6,119 \cdot 10^{-5}z^{-1} + 8,813 \cdot 10^{-5}z^{-2}}{1 - 1,857z^{-1} + 0,8568z^{-2}}$$

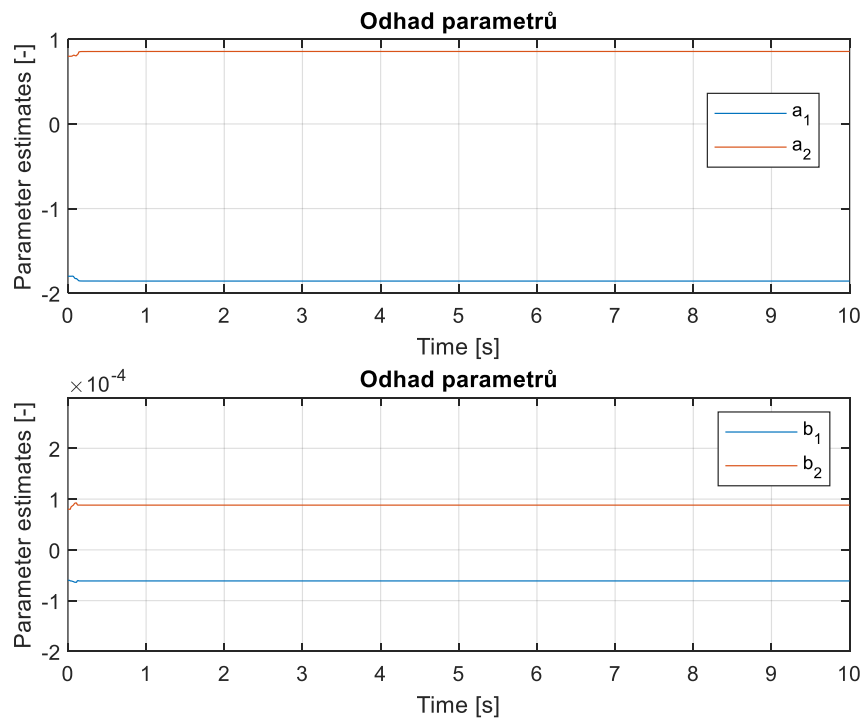
Prvky na hlavní diagonále kovarianční matice mají hodnotu 10^6



Obrázek 30 Schéma zapojení pro simulaci regulace



Obrázek 31 Průběh výstupu systému a akčního zásahu bez dopravního zpoždění



Obrázek 32 Průběh odhadů parametrů

11.2 S dopravním zpožděním

Řídicí parametry byly nastaveny podle simulačního ověření z Matlab scriptu.

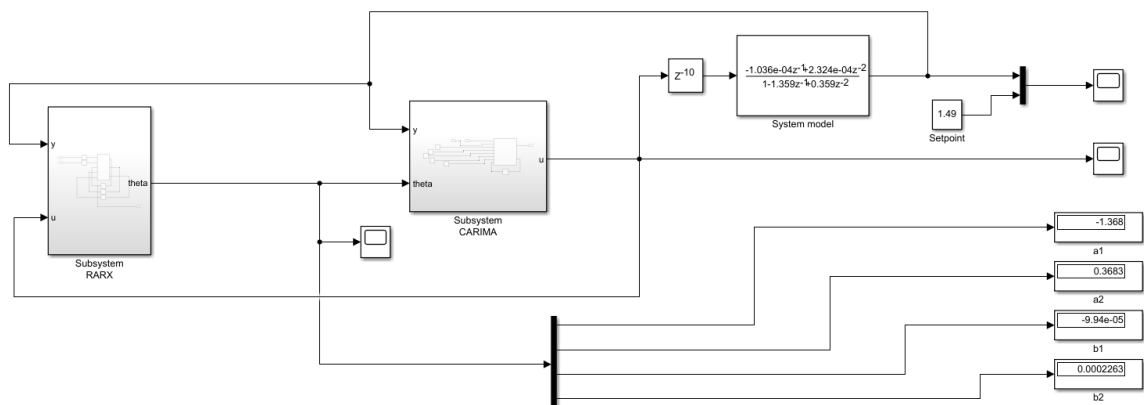
Počáteční odhady parametrů:

$$\hat{\theta}^T = [-1,4 \quad 0,4 \quad -1 \cdot 10^{-4} \quad 2 \cdot 10^{-4}]$$

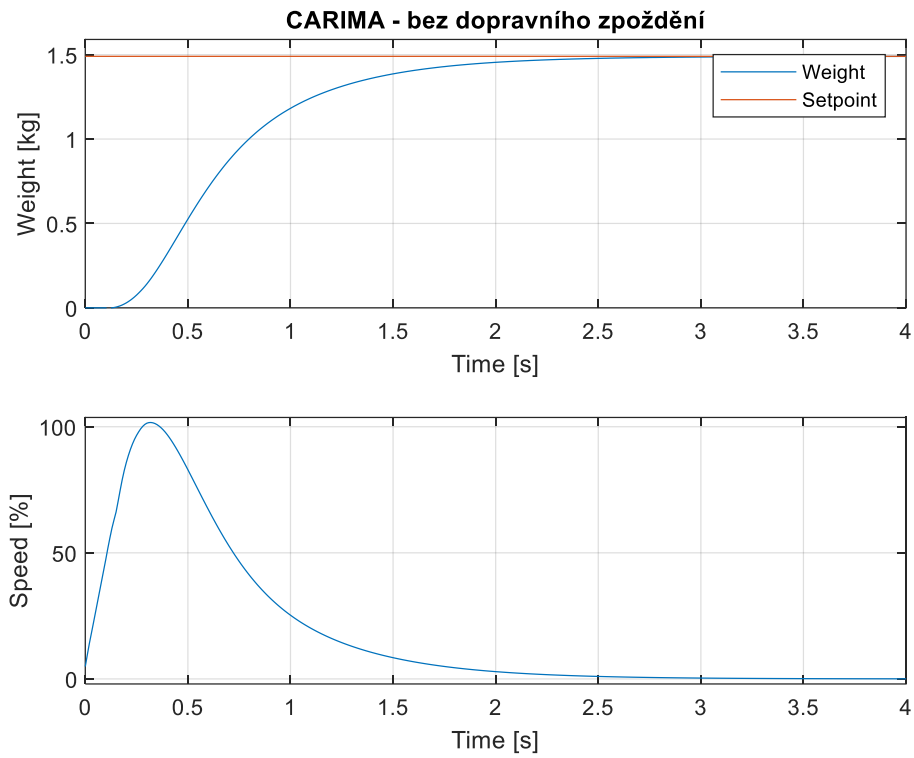
Přenos systému je převzat z jednorázové identifikace:

$$G(z^{-1}) = \frac{-1,036 \cdot 10^{-4} z^{-1} + 2,324 \cdot 10^{-4} z^{-2}}{1 - 1,359 z^{-1} + 0,359 z^{-2}}$$

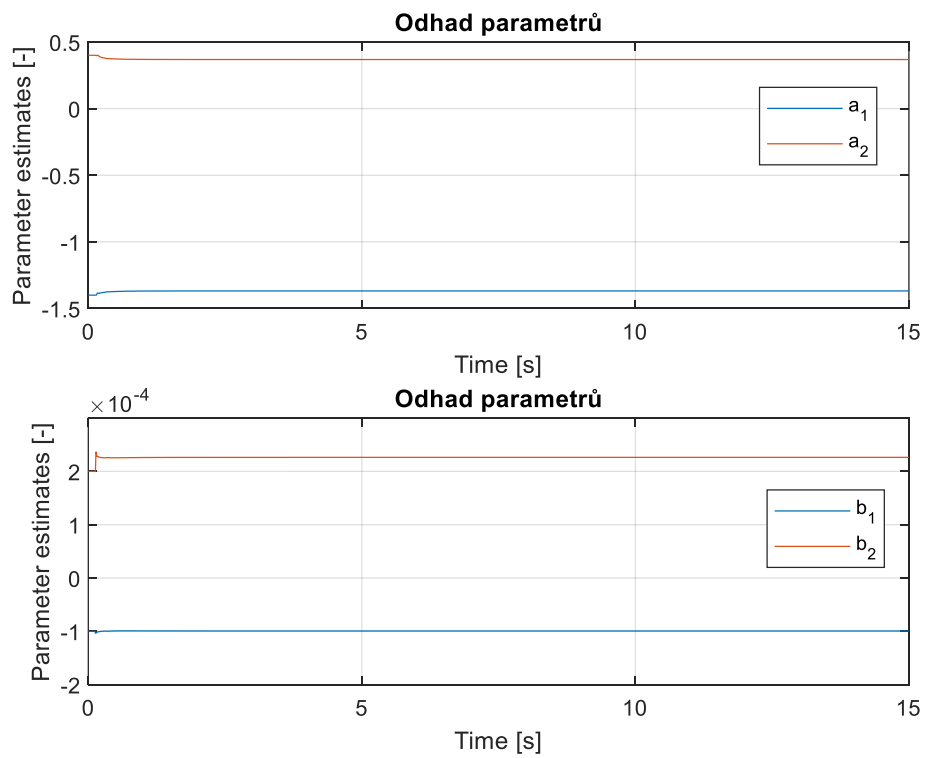
Prvky na hlavní diagonále kovarianční matice mají hodnotu 10^6



Obrázek 33 Schéma zapojení pro ověření regulace – systém s dopravním zpožděním



Obrázek 34 Průběh výstupu systému a akčního zásahu- systém s dopravním zpožděním



Obrázek 35 Průběh odhadů parametrů

12 IMPLEMENTACE NAVRŽENÉHO ŘÍZENÍ DO PLC

Pro implementaci navrženého řízení do PLC bylo potřeba přepsat skripty z Matlabu do formy vhodné pro PLC. Jelikož PLC neobsahuje maticové operace, bylo potřeba tyto operace provádět pomocí smyček a podmínek. Siemens poskytuje jazyk SCL, což je textově orientovaný jazyk podobný Pascalu nebo C. Pomocí tohoto programovacího jazyka byly maticové operace zapsány do cyklů **for**, které tvoří většinu kódu.

Z důvodu velmi omezeného přístupu do řídicí místnosti míchací linky, kde je nainstalována PCS7 s licenci, bylo využito podobnosti s programovacím jazykem Structured Text (ST), kterým se programují PLC Tecomat.

U funkčního bloku obsahující kód pro realizaci RARX byla největší matice 4x4. Jednotlivé maticové výpočty se musely rozdělit na části. Například aktualizace kovarianční matice:

$$C(k) = C(k-1) - \frac{C(k-1)\phi(k)\phi^T(k)C(k-1)}{1 + \phi^T(k)C(k-1)\phi(k)}$$

Následující obrázek ukazuje výpočet čitatele zlomku ze zmíněného vztahu:

```
//C*fi - 4x4 * 4x1 -> 4x1
FOR i := 1 TO 4 DO //radek
  C_FI[i] := 0.0;
  FOR j := 1 TO 4 DO //slopec
    C_FI[i] := C_FI[i] + C[i, j] * fi[j];
  END_FOR;
END_FOR;

//ans*fi' 4x1 * 1x4 ->4x4
FOR i := 1 TO 4 DO
  FOR j := 1 TO 4 DO
    C_FI_FI[i, j] := C_FI[i] * fi[j];
  END_FOR;
END_FOR;

//ans*C 4x4 * 4x4 =4x4
FOR i := 1 TO 4 DO
  FOR j := 1 TO 4 DO
    VRCH[i, j] := 0.0;
    FOR k := 1 TO 4 DO
      VRCH[i, j] := VRCH[i, j] + C_FI_FI[i, k] * C[k, j];
    END_FOR;
  END_FOR;
END_FOR;
```

Obrázek 36 Ukázka části kódu pro RARX

U implementace CARIMA modelu je výpočetní náročnost složitější, jelikož se zde nacházejí větší matice, např.: $N_2 \cdot N_u, N_u \cdot N_u$, kde $N_2 = 120$ a $N_u = 30$. Dále obsahuje CARIMA složitější maticové operace, jako je například inverze matice.

```

FOR i:=1 TO Nu DO
  for j:=1 TO Nu DO
    Inver[i,j]:=0;
  end_for;
  Inver[i,i]:=1;
end_for;
FOR i := 1 TO Nu DO
  maxRow := i;
  FOR k := (i + 1) TO Nu DO
    IF ABS(PreInv[k, i]) > ABS(PreInv[maxRow, i]) THEN
      maxRow := k;
    END_IF;
  END_FOR;

  FOR k := 1 TO Nu DO
    temp := PreInv[i, k];
    PreInv[i, k] := PreInv[maxRow, k];
    PreInv[maxRow, k] := temp;

    temp := Inver[i, k];
    Inver[i, k] := Inver[maxRow, k];
    Inver[maxRow, k] := temp;
  END_FOR;

  FOR j := (i + 1) TO Nu DO
    temp := PreInv[j, i] / PreInv[i, i];
    FOR k := 1 TO Nu DO
      PreInv[j, k] := PreInv[j, k] - PreInv[i, k] * temp;
      Inver[j, k] := Inver[j, k] - Inver[i, k] * temp;
    END_FOR;
  END_FOR;
END_FOR;
FOR i := Nu TO 1 BY -1 DO
  FOR j := (i - 1) To 1 BY -1 DO
    temp := PreInv[j, i] / PreInv[i, i];
    FOR k := 1 TO Nu DO
      PreInv[j, k] := PreInv[j, k] - PreInv[i, k] * temp;
      Inver[j, k] := Inver[j, k] - Inver[i, k] * temp;
    END_FOR;
  END_FOR;
  temp := PreInv[i, i];
  FOR j := 1 TO Nu DO
    Inver[i, j] := Inver[i, j] / temp;
  END_FOR;
END_FOR;

```

Obrázek 37 CARIMA – výpočet inverzní matice

Při implementaci CARIMA modelu došlo k problému. Jednalo se o to, že deklarované proměnné a dvourozměrná pole zabírala více místa, než bylo povolených 64 kB na funkční blok, aby bylo zajištěna správnost otočky cyklu. Z tohoto důvodu bylo potřeba upravit řídicí parametry, kdy se změnila hodnota N_u z 30 na 20.

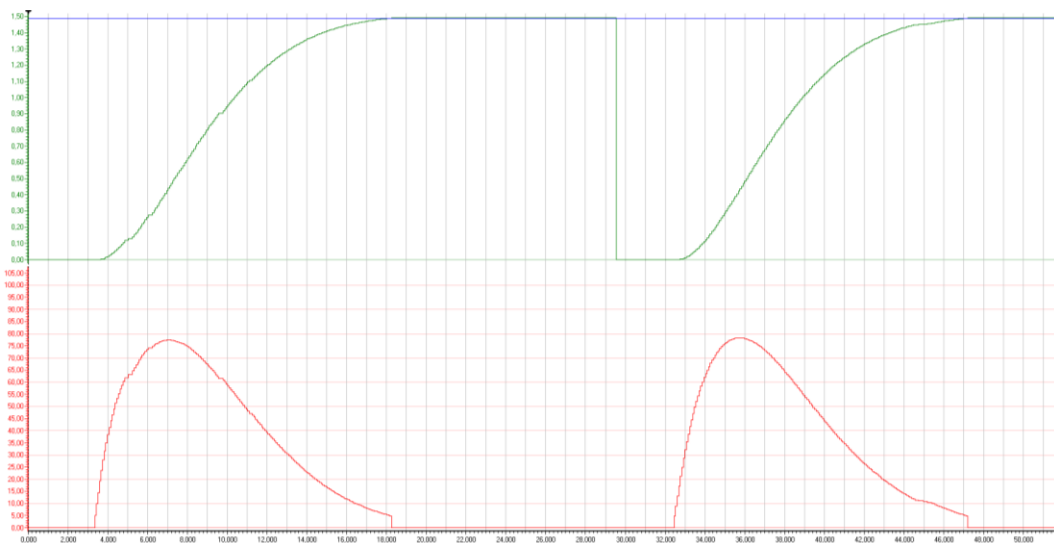
Simulace průběhu byla ověřena podobným způsobem, jako v Matlab Simulink. Tedy z funkčních bloků byl vytvořen regulační obvod.

Zelený průběh – ACT_VAL

Modrý průběh – SETPOINT

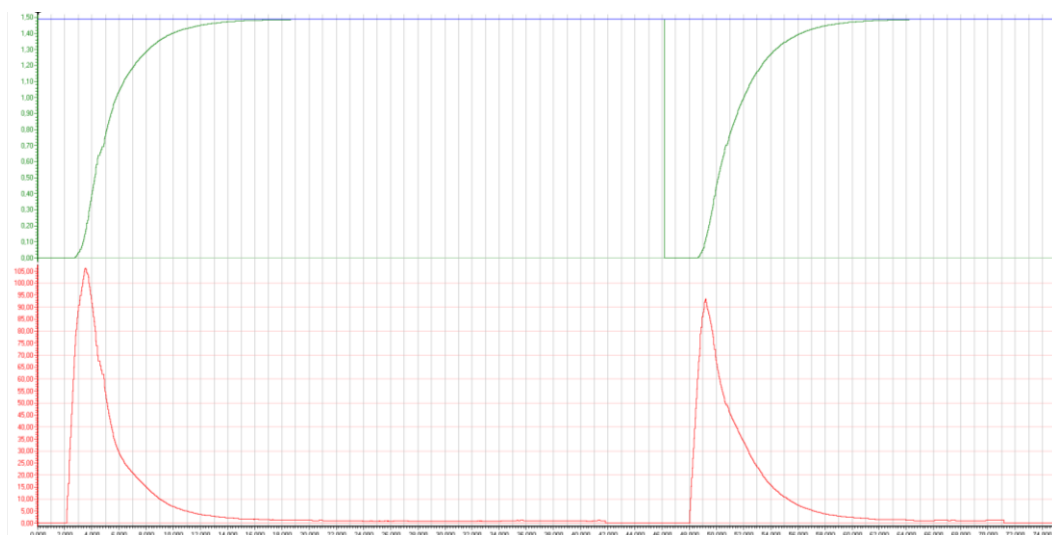
Červený průběh – SP_EXT

Simulace systému bez dopravního zpoždění:



Obrázek 38 Průběh simulace regulace systému bez dopravního zpožděním

Simulace systému s dopravním zpožděním:



Obrázek 39 Průběh simulace regulace systému s dopravním zpožděním

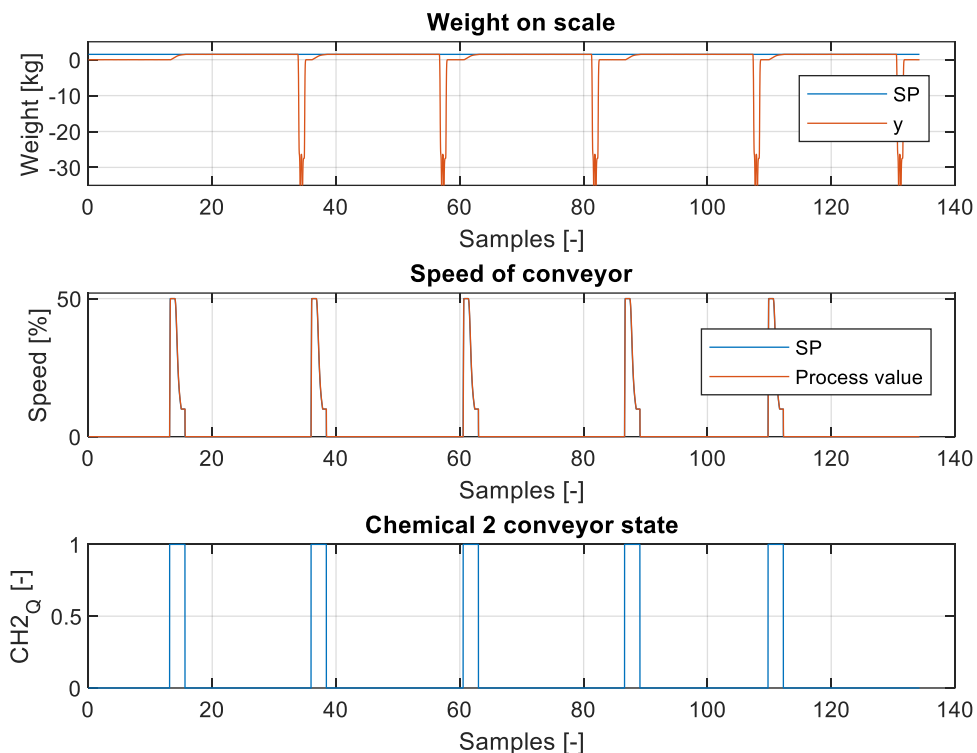
13 OVĚŘENÍ NA REÁLNÉM ZAŘÍZENÍ

Po implementaci kódů do PCS7 následovalo přidání zapínací a vypínací logiky. Následovalo propojení funkční bloků, aby tvořily regulační obvod.

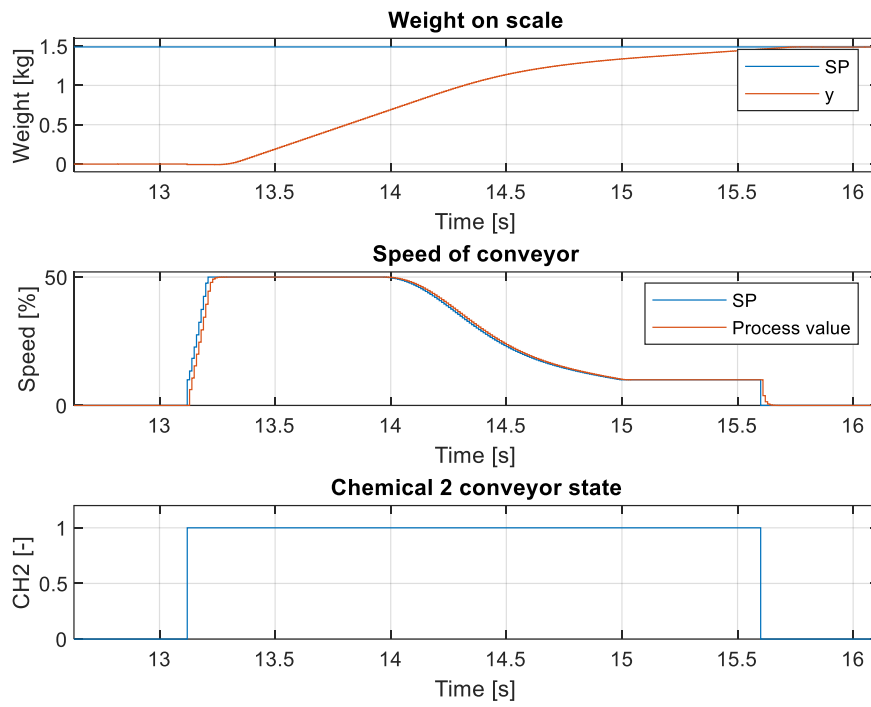
Vstupem do bloku rekurzivní identifikace byly proměnné *ENABLE* (slouží pro spuštění bloku), *PV* (aktuální rychlost pásového dopravníku) a *ACT_VAL* (aktuální hmotnost na váze). Výstupem byly parametry modelu a_1 , a_2 , b_1 , b_2 .

Vstupem do bloku prediktoru byly proměnné *ENABLE* (slouží pro spuštění bloku), *ACT_VAL* (aktuální hmotnost na váze) a hodnoty parametrů modelu a_1 , a_2 , b_1 , b_2 . Výstupní proměnnou byla *SP_EXT*, která udává požadovanou rychlost. Právě tato proměnná zasáhla do stávajícího řízení, kdy nahradila žádanou hodnotu rychlosti, která přicházela do měniče.

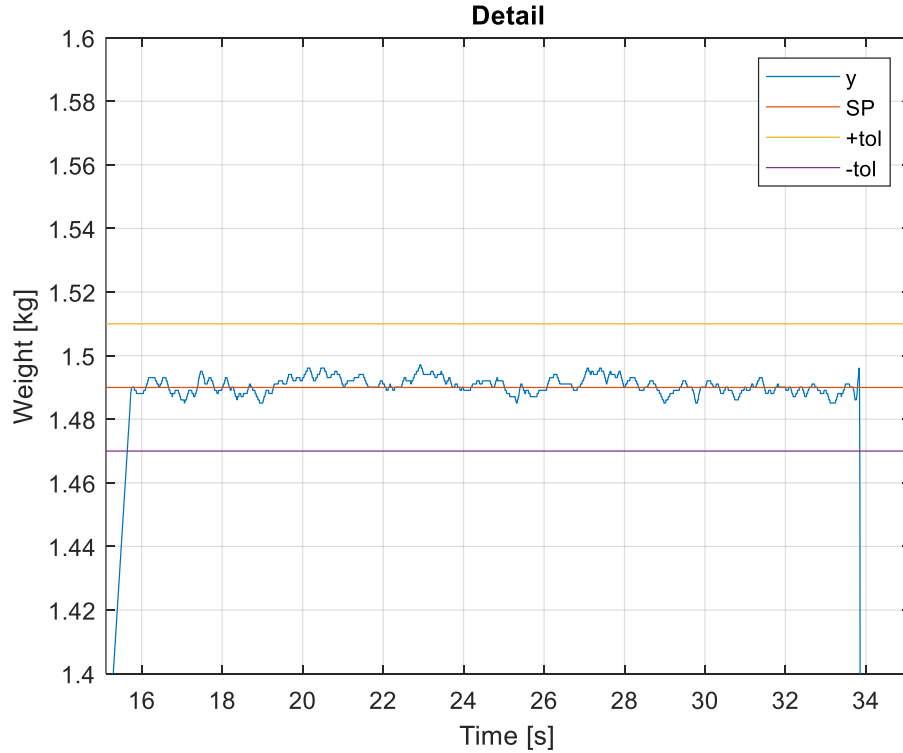
Po několika neúspěšných pokusech a následných korekcích v kódu se podařilo naměřit pár průběhů vážení s nově implementovaným řízením.



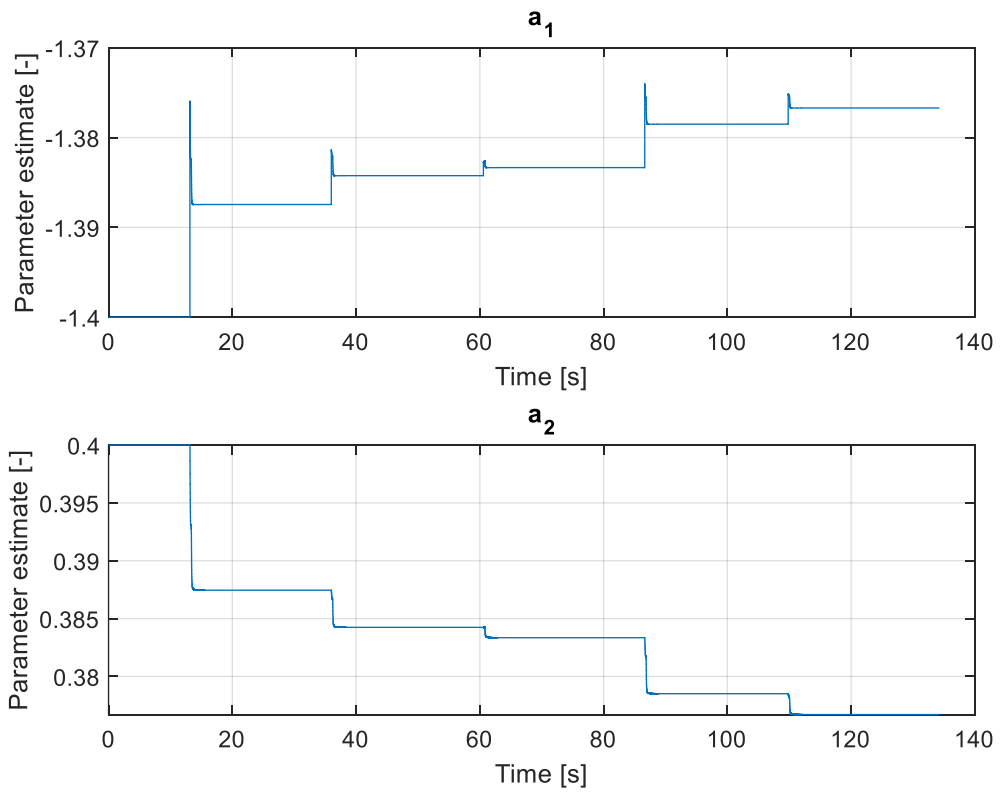
Obrázek 40 Naměřená data po implementaci nového řízení



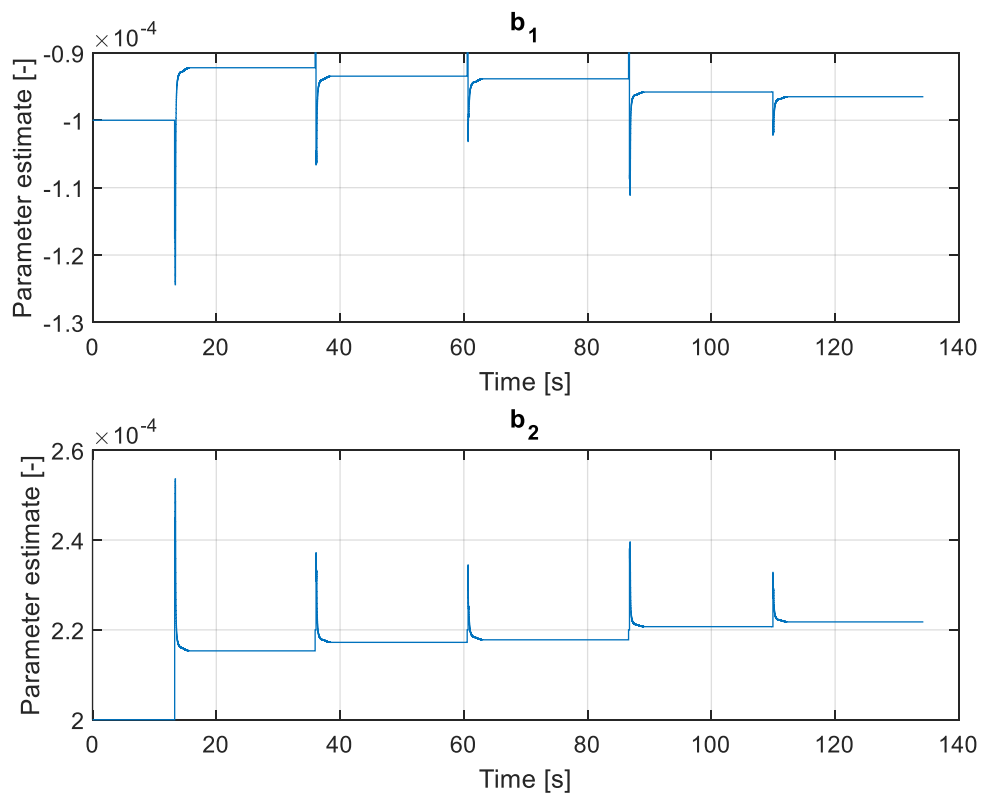
Obrázek 41 Průběh prvního vážení po implementaci nového řízení



Obrázek 42 Detail vážení – hmotnost na váze po implementaci nového řízení

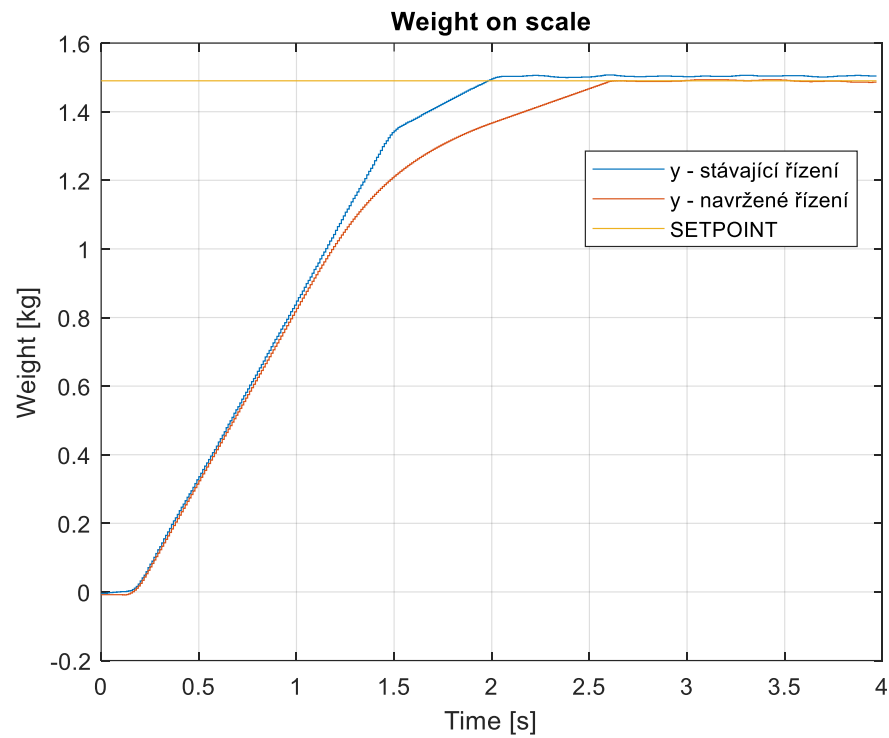


Obrázek 43 Odhady parametrů a_1 a a_2

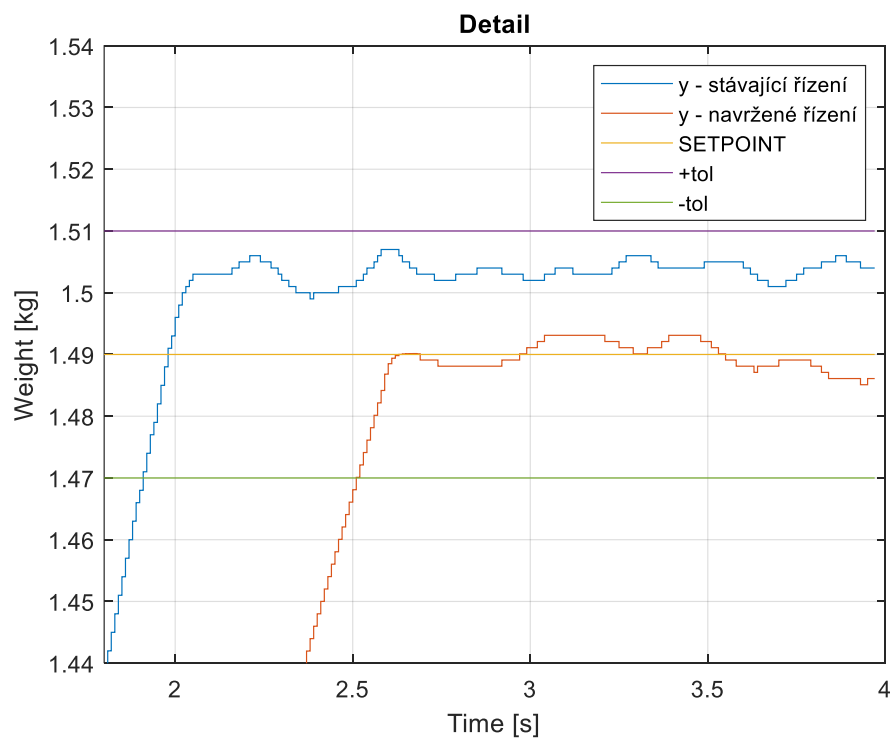


Obrázek 44 Odhady parametrů b_1 a b_2

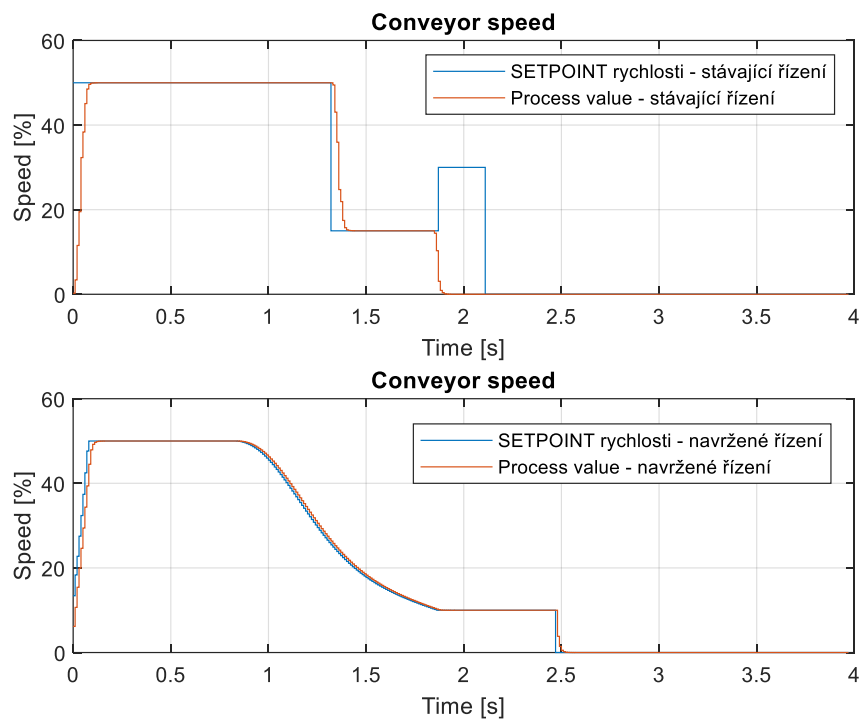
14 POROVNÁNÍ STÁVAJÍCÍHO A NAVRŽENÉHO ŘÍZENÍ



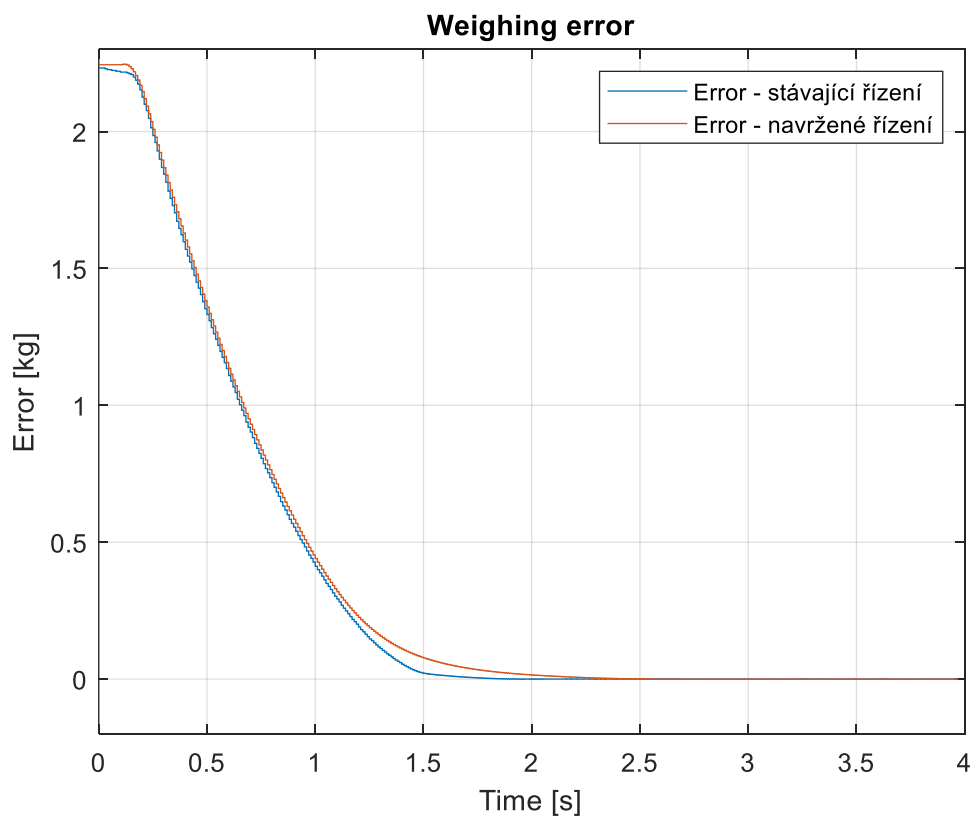
Obrázek 45 Porovnání stávajícího a navrženého způsobu řízení



Obrázek 46 Detail navážené váhy



Obrázek 47 Porovnání rychlostí způsobů řízení



Obrázek 48 Výpočetní chyba způsobů řízení

Stávající řízení dosáhne žádané hodnoty rychleji, ale vykazuje překmit a blíží se k hranici horní tolerance. Cílem návrhu nového způsobu řízení bylo odstranění právě tohoto překmitu a tím zvýšit celkovou přesnost vážení. Odstraněním překmitu se sníží nutnost, aby operátor míchací linky musel zasahovat do procesu navažování. Další výhodou řízení bez překmitu se také sníží nutnost vyhazování materiálu mimo toleranci z váhy, kdy je tento zásah z časového a finančního hlediska nežádoucí. Navržený způsob řízení je pomalejší, což znamená, že dosáhne žádané hodnoty později než stávající řízení. Tento rozdíl je viditelný na obrázku č. 45, kde je zobrazen celý průběh vážení obou způsobů řízení. Nicméně hlavní výhodou navrženého způsobu řízení je vyšší přesnost, která minimalizuje překmit a tím se oscilace způsobené průměrováním vah pohybují v těsném okolí žádané hodnoty. Detailní pohled na rozdíly mezi způsoby řízení poskytuje obrázek č.46, který znázorňuje, že stávající způsob řízení má větší odchylky od žádané hodnoty, než navržený způsob řízení.

Požadavek na nový způsob řízení, kterým bylo zvýšení přesnosti navažování materiál, se podařilo splnit za cenu zvýšení potřebného času pro navažovací proces. Zvýšení potřebného času pro navažování není z provozního hlediska zásadním problémem.

ZÁVĚR

Cílem diplomové práce byl návrh self-adaptive regulátoru pro řízení výšky hladiny materiálu při navažování na míchací linku v závodě Continental Barum v Otrokovicích. Bylo aplikováno prediktivní řízení s průběžnou identifikací parametrů modelu procesu pomocí rekursivní metody nejmenších čtverců. Navržené řízení bylo implementováno a testováno v provozu míchací linky a bylo provedeno srovnání se stávajícím způsobem řízení.

Teoretická část práce byla zaměřena na popis míchacích linek a řídicího problému, který bylo potřeba vyřešit. Bylo popsáno jak hardwarové, tak softwarové vybavení použité pro řízení, měření a monitorování činnosti linky. Dále byly v teoretické části popsány ARX model a metody použité pro identifikaci počátečních odhadů parametrů řízeného procesu i pro průběžnou identifikaci parametrů. Pro řízení bylo aplikováno prediktivní řízení.

Praktická část se soustředila na popis nového navrženého způsobu řízení míchací linky a jeho srovnání s dosavadním způsobem řízení. Na základě provedeného měření navrženého způsobu řízení a srovnání se stávajícím způsobem řešení lze konstatovat, že navržený self-adaptive regulátor představuje zlepšení z hlediska k přesnosti navažování oproti stávajícímu způsobu řízení. Navržený způsob řízení minimalizuje překmit za cenu prodloužení navažovacího procesu, což pro provoz míchací linky není zásadní problém.

SEZNAM POUŽITÉ LITERATURY

- [1] SIEMENS. *Automation systems*. Online. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/systems.html>. [cit. 2024-04-05].
- [2] SIEMENS. *Industrial automation*. Online. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/systems.html>. [cit. 2024-04-05].
- [3] SIEMENS. *SIMATIC controller*. Online. SIEMENS. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/systems/industrial/plc.html>. [cit. 2024-04-05].
- [4] SIEMENS. *SIMATIC S7-1200*. Online, PNG. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/systems.html>. [cit. 2024-04-05].
- [5] *What is PCS7?* Online. Automation Community. 2023. Dostupné z: <https://automationcommunity.com/what-is-pcs-7>. [cit. 2024-04-05].
- [6] SIMATIC PCS 7 Process Control System Online. 2023. Dostupné z: https://cache.industry.siemens.com/dl/files/406/109766406/att_981146/v1/br_simatic_pcs7_en_2017_Web.pdf
- [7] *Programmierung mit Siemens*. Online. Kuntec Automation. C2023. Dostupné z: <https://www.kuntec-automation.de/leistung/programmierung/siemens/>. [cit. 2024-04-12].
- [8] *SIWAREX weighing electronics*. Online. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/process-instrumentation/weight-measurement/weighing-electronics-simatic.html>. [cit. 2024-04-05].
- [9] *SIWAREX load cells*. Online. Siemens.com. C1996-2024. Dostupné z: <https://www.siemens.com/global/en/products/automation/process-instrumentation/weight-measurement/load-cells.html>. [cit. 2024-04-05].
- [10] *SIWAREX WL230 BB-S SA*. Online. Siemens.com. C2022. Dostupné z: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10044458?tree=CatalogTree>. [cit. 2024-04-23].

- [11] *The iba system*. Online. Iba-ag.com. Dostupné z: <https://www.iba-ag.com/en/>. [cit. 2024-04-23].
- [12] *Process Connectivity*. Online. Iba-ag.com. Dostupné z: <https://www.iba-ag.com/en/process-connectivity>. [cit. 2024-04-23].
- [13] *Acquire & Record*. Online. Iba-ag.com. Dostupné z: <https://www.iba-ag.com/en/acquire-record>. [cit. 2024-04-23].
- [14] *IbaPDA*. Online. Iba-ag.com. Dostupné z: <https://www.iba-ag.com/en/ibapda>. [cit. 2024-04-23].
- [15] *ARX Time-Series Model*. Online. Apmonitor: Data-Driven Engineering. Aktualizováno 17.10.2023. Dostupné z: <https://apmonitor.com/dde/index.php/Main/AutoRegressive>. [cit. 2024-04-24].
- [16] ZHU, Quanmin; TU, Quan; RONG, Yingjiao a CHEN, Jing. Parameter Identification of ARX Models Based on Modified Momentum Gradient Descent Algorithm. Online. *Complexity*. 2020, roč. 2020, s. 1-2. Dostupné z: <https://doi.org/10.1155/2020/9537075>. [cit. 2024-04-24].
- [17] BOBÁL, Vladimír. *Digital self-tuning controllers: algorithms, implementation and applications*. Advanced textbooks in control and signal processing. London: Springer, c2005. ISBN 1852339802.
- [18] MIKLEŠ, Ján a FIKAR, Miroslav. *Process modelling, identification, and control*. Berlin: Springer, c2007. ISBN 9783540719694.
- [19] BOBÁL, Vladimír; KUBALČÍK, Marek; DOSTÁL, Petr a MATĚJČEK, Jakub. Adaptive predictive control of time-delay systems. *Computers & Mathematics with applications*. 2013, roč. 66, s. 165-176. ISSN 0898-1221.
- [20] BOBÁL, Vladimír. *Identifikace systémů*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009. ISBN 978-80-7318-888-7.
- [21] SCHWENZER, Max; AY, Muzaffer; ABEL, Dirk a BERGS, Thomas. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*. 2021, roč. 117, č. 5-6, s. 1327-1349. ISSN 0268-3768, 1433-3015.

- [22] KUBALČÍK, Marek a BOBÁL, Vladimír. Predictive Control Of Time-Delay Processes. In: *ECMS 2012 Proceedings*. Koblenz, Německo, 29.5-1.6.2012. ECMS 2012: European Council for Modelling and Simulation, 2012, s. 455-460. ISBN 978-0-9564944-4-3. Dostupné z: <https://doi.org/doi:10.7148/2012>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MPC	Model Predictive Control
GPC	Generalized Predictive Control
ARX	Autoregressive with eXogenous input
MNČ	Metoda nejmenších čtverců.
RARX	Rekurzivní ARX
CARIMA	Controlled AutoRegressive Integrated Moving Average

SEZNAM OBRÁZKŮ

Obrázek 1 Zjednodušený model míchací linky	10
Obrázek 2 Siemens PLC – SIMATIC S7-1200 [4]	12
Obrázek 3 CFC – Continuous function chart – ukázka z PCS7 [7].....	14
Obrázek 4 Ukázka strukturovaného textu – SCL [7].....	15
Obrázek 5 Siwarex – ohýbací siloměr - WL230 BB-S SA [10].....	16
Obrázek 6 Ukázka rozhraní ibaPDA pro získávání naměřených hodnot [14].....	18
Obrázek 7 Blokové schéma ARX.....	20
Obrázek 8 Schématicky znázorněný princip prediktivního řízení.....	29
Obrázek 9 Princip prediktivního řízení.....	30
Obrázek 10 Vývojový diagram vážicího cyklu	38
Obrázek 11 Funkční blok pro určení fáze vážení	39
Obrázek 12 Část kódu pro určení fáze vážení	40
Obrázek 13 Průběh vážicího procesu – rychlost a váha	40
Obrázek 14 Blok měniče.....	42
Obrázek 15 Naměřená data.....	44
Obrázek 16 Detail – okolí setpointu	44
Obrázek 17 Průběh přijatého vážení.....	45
Obrázek 18 Detail přijatého vážení – hmotnost na váze	45
Obrázek 19 Průběh špatného měření	46
Obrázek 20 Detail špatného vážení – hmotnost na váze	46
Obrázek 21 Porovnání naměřených dat se simulovaným průběhem získaným pomocí funkce ARX v Matlabu	48
Obrázek 22 Porovnání naměřených dat se simulovaným průběhem získaným pomocí vlastního algoritmu pro ARX.....	48
Obrázek 23 Porovnání naměřených dat se simulovaným průběhem získaným ARX funkce – s dopravním zpožděním.....	50
Obrázek 24 Porovnání naměřených dat se simulovaným průběhem získaným pomocí vlastního algoritmu pro ARX- s dopravním zpožděním	50
Obrázek 25 Porovnání naměřených dat a simulovaného průběhu.....	52
Obrázek 26 Průběh odhadů parametrů modelu – RARX s dopravním zpožděním...53	
Obrázek 27 Porovnání naměřených dat a získaného průběhu s dopravním zpožděním o hodnotě 0,1 s.....	53

Obrázek 28 Průběh predikce vážení a akčního zásahu - CARIMA.....	54
Obrázek 29 Průběh predikce vážení a akčního zásahu – CARIMA s dopravním zpožděním	55
Obrázek 30 Schéma zapojení pro simulaci regulace	56
Obrázek 31 Průběh výstupu systému a akčního zásahu bez dopravního zpoždění	57
Obrázek 32 Průběh odhadů parametrů.....	57
Obrázek 33 Schéma zapojení pro ověření regulace – systém s dopravním zpožděním	58
Obrázek 34 Průběh výstupu systému a akčního zásahu- systém s dopravním zpožděním	59
Obrázek 35 Průběh odhadů parametrů.....	59
Obrázek 36 Ukázka části kódu pro RARX.....	60
Obrázek 37 CARIMA – výpočet inverzní matice	61
Obrázek 38 Průběh simulace regulace systému bez dopravního zpožděním	62
Obrázek 39 Průběh simulace regulace systému s dopravním zpožděním	62
Obrázek 40 Naměřená data po implementaci nového řízení	63
Obrázek 41 Průběh prvního vážení po implementaci nového řízení	64
Obrázek 42 Detail vážení – hmotnost na váze po implementaci nového řízení	64
Obrázek 43 Odhady parametrů a_1 a a_2	65
Obrázek 44 Odhady parametrů b_1 a b_2	65
Obrázek 45 Porovnání stávajícího a navrženého způsobu řízení	66
Obrázek 46 Detail navážené váhy	66
Obrázek 47 Porovnání rychlostí způsobů řízení	67
Obrázek 48 Výpočetní chyba způsobů řízení	67

SEZNAM PŘÍLOH

P I Obsah CD

PŘÍLOHA P I: OBSAH CD

Struktura přiloženého CD:

- Soubor **fulltext.pdf** – elektronická verze práce
- Adresář **Programy** – obsahuje funkční bloky s RARX a CARIMA
- Soubor **StavajiciRizeni** – obsahuje zobrazení všech průběhů měření stávajícího řízení
- Soubor **NavrzeneRizeni** – obsahuje zobrazení všech průběhů měření navrženého řízení