



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

**PARAMETRIZACE PID REGULÁTORU NA PLC
MITSUBISHI**

PARAMETRIZATION OF PID CONTROLLER ON MITSUBISHI PLC

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Michal Koláčný

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Blaha, Ph.D.

BRNO 2022

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Michal Koláčný

ID: 220991

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Parametrizace PID regulátoru na PLC Mitsubishi

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s PLC Mitsubishi řady iQ-F, s možnostmi implementace funkce PID regulátoru a s možnostmi programování operátorského panelu Mitsubishi GT-2104RTBD.
2. Zvolte vhodný fyzikální model pro ověření funkce PID regulátoru. Model sestavte a proveďte jeho propojení s PLC pomocí analogových vstupů a výstupů.
3. V prostředí MATLAB Simulink vytvořte simulační model realizovaného fyzikálního modelu.
4. Vytvořte software pro realizaci PID regulátoru v PLC a SW vybavení pro operátorský panel umožňující nastavení konstant PID regulátoru a ovládání demo aplikace.
5. Proveďte parametrizaci PID regulátoru několika vhodně zvolenými metodami a srovnajte s parametry získanými pomocí funkce Auto-Tuning.
6. Srovnajte výsledky řízení PID regulátoru reálného fyzikálního modelu se simulačními výsledky.

DOPORUČENÁ LITERATURA:

[1] Blaha, P., Vavřín, P.: Řízení a regulace I. Základy regulace lineárních systémů – spojité a diskrétní. Skriptum VUT.

další dle doporučení vedoucího práce

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: doc. Ing. Petr Blaha, Ph.D.

doc. Ing. Václav Jirsík, CSc.

předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Cílem této práce je vytvořit demo aplikaci, která slouží pro seznámení se s jednotlivými složkami PID regulátoru. PID regulátor je realizován funkčním blokem, který je implementován v PLC Mitsubishi řady iQ-F. Regulovanou veličinou je výška pingpongového míčku. Míček je udržován v požadované výšce změnou rychlosti proudění vzduchu. Změna rychlosti proudění je dosažena změnou otáček motoru. Na hřídeli motoru je upevněna vrtule, která vhání vzduch do trubice. V trubici je zabudovaný optický senzor vzdálenosti SHARP GP2Y0A41SK0F, který snímá aktuální výšku míčku. K této soustavě je vytvořen model, který je simulován v programu MATLAB. Aktuální informace o soustavě jsou zobrazeny na operátorském panelu Mitsubishi GT-2104RTBD, kde je dále možné měnit velikosti jednotlivých složek regulátoru.

Klíčová slova

PLC, PID, optický sensor, MATLAB, HMI, IQ works

Abstract

The aim of this thesis is to create a demo application for a brief introduction with the components of the PID regulator. The PID regulator is realized by a function block, which is implemented in a Mitsubishi PLC series iQ-F. The controlled variable is the height of the ping pong ball. The ball is kept at the desired height by changing the air flow rate. The change of the air flow is achieved by changing the motor speed. A propeller is mounted on the motor shaft which blows air into the tube. A SHARP GP2Y0A41SK0F optical distance sensor is built into the tube to sense the actual height of the ball. A model of this system is created and simulated in MATLAB program. The current information about the system is displayed on the Mitsubishi GT-2104RTBD operator panel, where it is also possible to change the sizes of individual parameters of the regulator.

Keywords

PLC, PID, optical sensor, MATLAB, HMI, IQ works

Bibliografická citace

KOLÁČNÝ, Michal. *Parametrizace PID regulátoru na PLC Mitsubishi* [online]. Brno, 2022 [cit.2022-05-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/142557>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Blaha.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Michal Kolářný</i>
VUT ID studenta:	<i>220991</i>
Typ práce:	<i>Bakalářská práce</i>
Akademický rok:	<i>2021/22</i>
Téma závěrečné práce:	<i>Parametrizace PID regulátoru na PLC Mitsubishi</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 22. května 2022

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Petru Blahovi, Ph.D. za odbornou pomoc a poskytnutí cenných rad a konzultací při tvorbě mé práce. Dále bych rád poděkoval konzultantovi Ing. Jiřímu Klimešovi za účinnou pomoc při zpracování práce.

V Brně dne: 22. května 2022

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. TEORETICKÝ NÁVRH ŘEŠENÍ	12
2. PLC MITSUBISHI IQ – F.....	13
2.1 POPIS PLC MITSUBISHI IQ – F	13
2.2 PŘÍDAVNÉ ANALOGOVÉ KARTY	14
2.2.1 MITSUBISHI MELSEC iQ-F FX5-4AD-ADP.....	14
2.2.2 MITSUBISHI MELSEC iQ-F FX5-4DA-ADP.....	15
2.3 FUNKČNÍ BLOK PID.....	15
2.3.1 Základní popis a parametry	15
2.3.2 Funkce Auto-Tuning.....	17
2.4 ZDROJOVÝ KÓD PLC	18
2.5 FUNKCE POUŽITÉ V SW.....	19
2.5.1 MOV.....	19
2.5.2 INT2FLT	19
2.5.3 SCL2.....	20
2.5.4 SET.....	21
2.5.5 RST.....	21
3. OPTICKÝ SENZOR SHARP GP2Y0A41SK0F	22
3.1 PRINCIP A ZÁKLADNÍ PARAMETRY	22
3.2 KALIBRACE	24
4. HMI – MITSUBISHI GT2705-VTBD	25
4.1 ZÁKLADNÍ POPIS A PARAMETRY	25
4.2 POPIS FUNKCE ZÁLOŽEK PANELU.....	26
5. VÝVOJOVÁ PROSTŘEDÍ.....	28
5.1 VÝVOJOVÉ PROSTŘEDÍ MATLAB.....	28
5.1.1 System identification toolbox	29
5.2 VÝVOJOVÉ PROSTŘEDÍ IQ WORKS	30
5.2.1 MELSOFT GX Works3.....	30
5.2.2 MELSOFT GT Designer3	31
6. PŘÍSLUŠENSTVÍ SOUSTAVY	34
6.1 BEZKARTÁČOVÝ MOTOR A2212/6T	34
6.1.1 Driver pro BLCD motor.....	34
6.1.2 Zdroj YS-100-12.....	35
6.2 RASPBERRY PI PICO	35
6.3 PRAKTICKÝ VÝROBEK	36
7. PID REGULÁTOR	39
7.1 POPIS JEDNOTLIVÝCH SLOŽEK PID REGULÁTORU.....	39

7.1.1	<i>P – složka</i>	39
7.1.2	<i>I – složka</i>	39
7.1.3	<i>D složka</i>	40
7.2	PID REGULÁTOR – POPIS	40
7.3	POUŽITÉ METODY PRO NÁVRH REGULÁTORU	41
7.3.1	<i>Metoda optimalizace ITAE kritéria</i>	41
7.3.2	<i>Metoda Ziegler-Nichols</i>	41
8.	SIMULACE NAVRŽENÉHO MODELU	43
8.1	NÁVRH PSD REGULÁTORU	43
8.2	NÁVRH SOUSTAVY	44
9.	ZÁVĚR	46
	LITERATURA	48
	SEZNAM SYMBOLŮ A ZKRATEK	50
	SEZNAM PŘÍLOH	52

SEZNAM OBRÁZKŮ

2.1	Použité PLC Mitsubishi IQ-F v demo aplikaci	14
2.2	Vnitřní zapojení funkčního bloku PID regulátoru [5]	16
2.3	Funkční blok PID regulátoru pro různé programovací jazyky [5].....	17
2.4	Řízení PID bloku.....	19
2.5	Příklad použití funkce SCL2 [5]	20
3.1	Optický senzor Sharp GP2Y0A41SK0F [6].....	22
3.2	Závislost výstupního napětí na vzdálenosti [7]	23
4.1	HMI GT-2705-VTBD [9].....	26
4.2	Vybrané záložky HMI panelu	27
5.1	Identifikace soustavy	30
5.2	Navigační software MELSOFT Navigator.....	32
5.3	Programovací prostředí MELSOFT GX Works3	32
5.4	Programovací prostředí MELSOFT GT Designer3.....	33
6.1	Bezkartáčový motor A2212/6T [16]	34
6.2	Zdroj YS-100-12	35
6.3	Model výrobku s vrtulí.....	37
6.4	Model výrobku s HMI.....	37
6.5	Schéma zapojení demo aplikace.....	38
8.1	Simulační schéma regulačního obvodu v MATLABU	44

SEZNAM TABULEK

2.1	Auto-Tuning – výpočty metody Limit Cycle [5].....	17
2.2	Auto-Tuning – výpočty metody Step Response [5]	18
7.1	Tabulka pro výpočet konstant regulátoru pomocí Z-N metody [20]	42
8.1	Tabulka srovnání identifikovaných přenosů soustav	45

ÚVOD

Bakalářská práce pojednává o návrhu optimálního systému nasazeného do reálné regulační smyčky. Snahou je dosáhnout takového systému, který automaticky dokáže reagovat na změnu, která je způsobena zásahem vnějších vlivů do procesu. Reakcí se rozumí, vrácení se do stavu, který přetrvával do doby před tímto zásahem. Řešením problému jsou například PID regulátory, které mají snahu vracet se do původního stavu před zásahem a udržovat si tak svoji pracovní polohu. PID regulátory jsou tedy zařízení, které ovlivňují systém pomocí akčních zásahů, čímž se snaží dosáhnout požadované veličiny a tu si co nejdéle udržet.

Zadání bakalářské práce spadá do procesů automatizace a bylo navrženo firmou B:TECH se sídlem v Havlíčkově Brodě, která se těmito problémy zabývá. V této bakalářské práci je podrobně popsán návrh a teoretická realizace PID regulátoru za pomoci PLC Mitsubishi řady iQ-F.

Úkolem této demo aplikace je vytvoření reálného PID regulátoru, na kterém lze porovnávat teoretické metody s funkcí Auto-Tuning. V aplikaci se PID regulátorem reguluje výška pingpongového míčku. Soustava je složena z několika hlavních částí a k ní jsou potřebné některé mechanické a elektrické součástky. Hlavním řídicím mechanismem je v demo aplikaci PLC, které se stará o celé řízení a regulaci soustavy.

Bakalářská práce se skládá z 9 hlavních kapitol, kde je konkrétně popsána celá problematika daného problému. V první kapitole je popsán teoretický návrh funkčnosti celé soustavy. Použité PLC a jeho výhody či nevýhody jsou uvedeny v druhé kapitole. Nedílnou součástí pro získávání, či měření dat jsou snímače uvedené ve třetí kapitole optický senzor SHARP GP2Y0A41SK0F. Čtvrtá kapitola se zabývá funkcemi a použitím HMI v soustavě. V páté kapitole jsou popsány vývojová prostředí MATLAB a IQ Works.

Doplňkové součástky použité v aplikaci jsou podrobněji popsány v šesté kapitole. V následující sedmé kapitole je tato práce zaměřena na teoretické předpoklady a popisy jednotlivých složek PID regulátoru. Osmá kapitola se zabývá konkrétním teoretickým návrhem PID regulátoru a soustavy. V poslední kapitole je zhodnocení závěrečné práce.

1. TEORETICKÝ NÁVRH ŘEŠENÍ

V následující kapitole je popsán teoretický návrh řešení pro demo aplikaci s PID regulátorem na řízení požadované výšky pingpongového míčku. Celá soustava je navržena a sestavena pro společnost B: tech se sídlem v Havlíčkově Brodě. Soustava je zde použita jako výukový materiál pro začínající techniky a programátory k představě reálného chování PID regulátoru.

Úkolem této aplikace je pomocí PID regulátoru regulovat a udržovat konstantní výšku pingpongového míčku nad trubicí. Pingpongový míček je nadnášen za pomoci mechanismu nasávající vzduch. Tento mechanismus je tvořen bezkartáčovým BLDC motorem, který pohání otočnou vrtuli a tato vrtule následně vhání nasátý vzduch do trubice. Tato trubice je navržena tak, aby se na svém konci zužovala a vytvářela tak větší rychlost proudění vzduchu než na začátku. Z důvodu větší stability míčku v horizontální poloze je dále na konci trubice připevněn ze stran otevřený válec. Trubice a držák motoru s vrtulí je vytisknut za pomoci 3D tiskárny. Aktuální výšku míčku snímá optoelektrický infračervený snímač od společnosti SHARP. Tato celá soustava je řízena za pomoci PLC od společnosti Mitsubishi.

Nejprve je vytvořen matematický model celé soustavy v závislosti na změřených datech z PLC. Tyto data jsou posléze použita v MATLABU k vytvoření simulace. Za použití několika vhodných metod, jako je například optimalizace ITAE kritéria, nebo Ziegler-Nicholsova metoda, jsou nalezeny hodnoty příslušných parametrů PID regulátoru. Dále se tyto parametry nastavují v simulaci. Po odsimulování soustavy jsou nalezené parametry nastaveny do funkčního bloku v PLC a odzkoušeny na reálné soustavě.

Všechna získaná a aktuálně nastavená data jsou uživateli přehledně zobrazena na operátorském panelu. Operátorský panel dále umožňuje nastavení jednotlivých parametrů PID regulátoru a tím je možné pozorovat změny a chování uzavřené smyčky.

2. PLC MITSUBISHI IQ – F

V následujícím textu je představeno PLC Mitsubishi IQ-F, které je použito k řízení soustavy. Pro implementaci PID regulátoru slouží knihovní funkčního bloku PID. V kapitole 2.2 jsou popsány přídatné analogové karty sloužící pro získávání dat z optického snímače, který je přiveden na analogové vstupy přídatné karty.

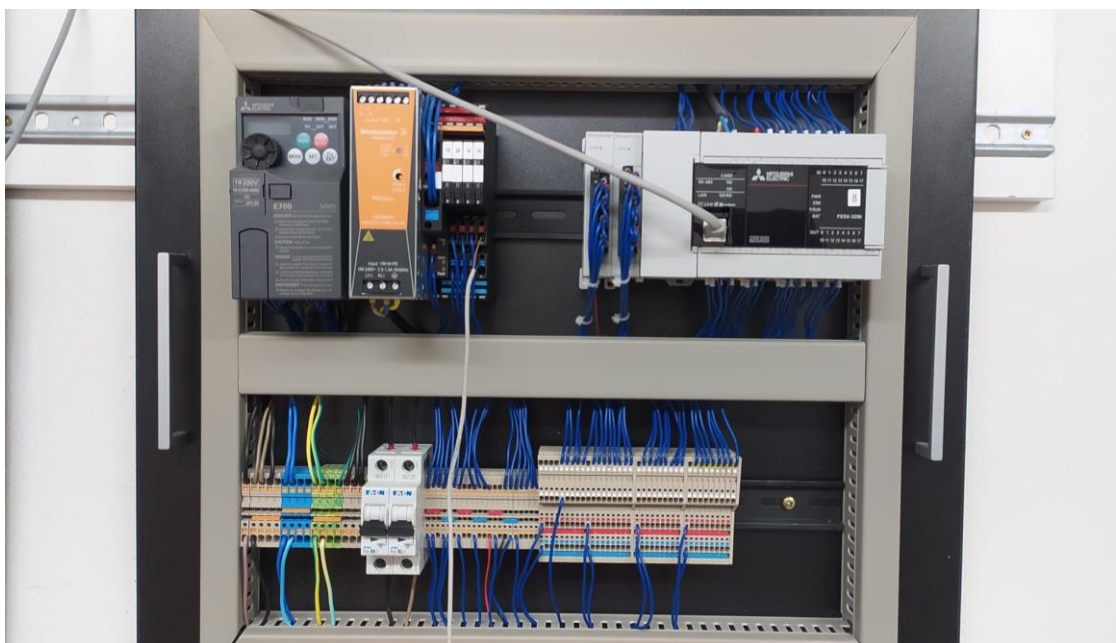
2.1 Popis PLC Mitsubishi IQ – F

Pro implementaci funkčního bloku PID regulátoru je použito PLC společnosti Mitsubishi Electric, a to konkrétně řady Mitsubishi IQ – F. Tato řada je určena pro jednodušší aplikace. Řada MELSEC F prošla v posledních letech několika změnami a následně uvolnila místo pro novou řadu MELSEC iQ – F.

PLC této řady má vylepšenou sběrnici s vysokorychlostním přenosem dat. Sběrnice zvyšuje celkový výkon systému. PLC IQ-F je rozšířeno několika novými funkcemi jako jsou například funkce protokolování dat, řízení polohy nebo zabezpečení, které jsou standardně integrovány v systému [1]. PLC řady iQ-F je vyráběno v kompaktní sestavě. Dále je vybaveno komunikačními rozhraními RS-422 (lze zapojit jako RS-485) a Ethernetem. Obě komunikační rozhraní umožňují komunikaci s externími zařízeními jako například operátorské panely, frekvenční měniče, moduly vzdálených vstupů a výstupů, SCADA systémy. Toto PLC je dále vybaveno dvěma 12-ti bitovými analogovými vstupními porty a jedním výstupním portem s alarmovým hlášením. Tyto porty jsou použity pro jiné zařízení, tudíž v demo aplikaci jsou použity přídatné analogové a digitální karty. Pomocí rozhraní USB (Mini – B) je možné připojení do programovatelného prostředí IQ Works3. Mezi základní sloty programovatelné logické jednotky se řadí slot na SD karty vhodný pro zálohování nebo aktualizaci programu. Hlavními bezpečnostními opatřeními proti krádeži nebo změně dat jsou funkce zaheslování souboru či použití zabezpečovacího klíče. Všechny aplikace a programy PLC jsou uchovávány v ROM (read only memory). ROM je nezávislá na napájení a v případě náhlého výpadku zdroje napájení nedojde ke ztrátě dat.

V této aplikaci je použito PLC s CPU FX5U-32MT/ESS. PLC s tímto CPU plně podporuje všechny funkce uvedené v přechozím odstavci. Dále jsou na přední straně automatu umístěny LED indikátory zobrazující aktuální stavy vstupních a výstupních portů. Další dva LED indikátory slouží pro kontrolu PLC, zda je v režimu RUN, nebo ERROR. PLC je vybaveno trojicí tlačítek RUN, STOP a RESET, pomocí nichž je možné celé PLC zapnout, vypnout a restartovat bez nutnosti odpojení hlavního napájecího zdroje. Tento logický automat je napájen klasickým síťovým napětím 230 V/50 Hz. PLC FX5U-32MT/ESS je vybaveno 16 binárními vstupy 24VDC. Zapojením lze zvolit, zda jsou vstupy v PNP (source) nebo NPN (sink) logice – pomocí svorky S/S. Dále je PLC vybaveno 16 tranzistorovými výstupy v PNP (source logice). Vzhledem k tomu, že jsou

vstupy i výstupy připojeny přímo k procesoru (PLC je kompaktní), lze prvních 8 vstupů a výstupů použít pro rychlé funkce (připojení enkodéru, ovládání krokových motorů nebo servomotorů), případně použít vstupy pro generování přerušení. PLC dále obsahuje 2 analogové vstupy a 1 analogový výstup. Integrované analogové IO umožňují připojení pouze napěťových signálů 0 až 10 V. Na testovací desce jsou tyto analogové vstupy připojeny k frekvenčnímu měniči, který v dané úloze není použit. Kapacita programu, který lze do paměti PLC nahrát, je okolo 64 tisíc kroků, nebo 128 tisíc kroků s určitými provozními omezeními, což je pro účely této úlohy dostačující [2]. PLC je zkonstruováno pro umístění na DIN lištu společně s krytím IP20, které říká, že musí být PLC umístěno v rozvaděči. Lze ho dále rozšířit některými přídatnými kartami jako v tomto případě.



Obrázek 2.1 Použité PLC Mitsubishi IQ-F v demo aplikaci

2.2 Přídavné analogové karty

PLC je rozšířeno o analogové adaptéry (karty) FX5U-4AD-ADP (analogový vstup) a FX5U-4DA-ADP (analogový výstup). Zapojením a konfigurací těchto adaptérů lze měřit a generovat jak napěťové, tak proudové signály. V podkapitolách 2.2.1 a 2.2.2 jsou tyto karty podrobněji popsány.

2.2.1 MITSUBISHI MELSEC iQ-F FX5-4AD-ADP

Tato analogová karta je rozšířením PLC o 4 přídavné analogové vstupy. Karta je plně kompatibilní a je vyráběna primárně k připojení PLC Mitsubishi IQ – F, ale lze ji připojit i k jiným PLC. Analogové vstupy jsou v tomto případě použity k připojení optického senzoru GP2Y0A21YK0F ke snímání výšky pingpongového míčku. Karta je vybavena pouze LED indikátorem napájení. Vstupy karty jsou v rozsahu napětí

- 10 až +10 V s odporem 1 M Ω a rozsahem proudu -20 až +20 mA DC při odporu 250 Ω a lze se mezi těmito rozsahy přepínat. Karta je připojena k PLC přes datovou sběrnici backplane, která slouží pro vzájemnou komunikaci [3].

2.2.2 MITSUBISHI MELSEC iQ-F FX5-4DA-ADP

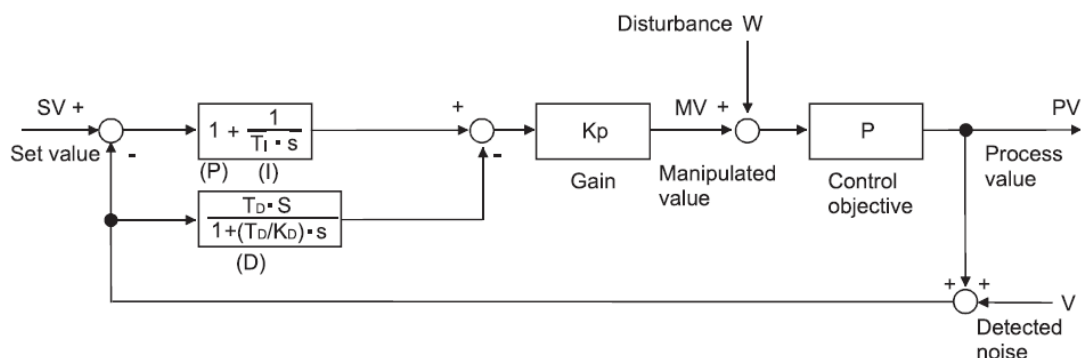
Analogová karta MELSEC iQ-F FX5-4DA-ADP slouží k ovládání pomocného Raspberry Pi Pico, kterým se následně pomocí PWM signálu ovládají a mění otáčky motoru. Je primárně určena k připojení PLC od společnosti Mitsubishi, ale je schopna komunikovat i s automaty od jiných výrobců. Tato analogová karta má pouze 4 výstupní analogové porty v rozsahu napětí 0 V až 10 V pro 16000 digitů, nebo -10 V až 10 V pro -8000 až 8000 digitů. Rozsahy proudů jsou 0 až 20 mA nebo více používanější 4 až 20 mA pro rozlišení 16000 digitů. Karta je připojena stejně jako MELSEC iQ-F FX5 - 4AD-ADP a to přes sběrnici backplane [4].

2.3 Funkční blok PID

V samotném programu MELSOFT GX Works 3 je možnost programování PID regulátoru pomocí funkčního bloku, který je podrobně popsán v e – Manuálu. Manuál slouží jako nápověda k jednotlivým funkčním blokům, i k celému softwaru od společnosti Mitsubishi Electric. Jelikož PLC není schopno pracovat ve spojitém čase, ale běží v diskrétním čase, v podstatě se jedná o PSD regulátor s vzorkovací periodou 0,01s.

2.3.1 Základní popis a parametry

Funkční blok PID regulátoru v programu představuje reálný PSD regulátor, který reguluje soustavu v dané demo aplikaci. Vnitřní zapojení funkčního bloku je na Obrázek 2.2. Instrukce pro výpočet výstupní hodnoty (d) potřebuje, aby na (s2) byla přivedena měřená data, která jsou měřena optickým snímačem. Pro výpočet výstupní hodnoty je nastavena vhodná kombinace proporcionální, integrační a derivační složky. Jakmile je nastavena cílová hodnota, měřená hodnota a parametry PID a program je spuštěn, výsledek operace se uloží do výstupní hodnoty při každé periodě vzorkování. Perioda vzorkování je nastavena tak, aby byla delší, než je provozní cyklus PLC. Výstupní hodnotu PID regulátoru lze omezit horní a dolní hodnotou, z důvodu omezení růstu integrační složky. Aby bylo možné použít více funkčních bloků PID regulátoru zároveň, je zapotřebí odlišných operandů (název vstupních proměnných, hodnot) [5]. Možností je použít funkci Auto-Tuning, která je popsána v podkapitole 2.3.2. Funkční blok je zobrazen na obrázku 2.3.

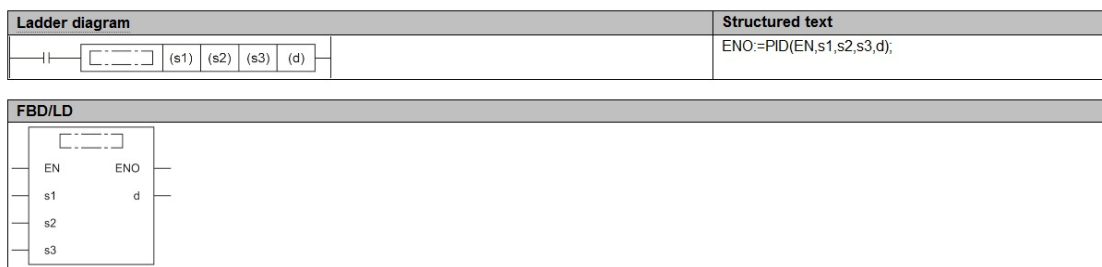


Obrázek 2.2 Vnitřní zapojení funkčního bloku PID regulátoru [5]

- **Cílová hodnota (s1)** – Do cílové hodnoty se nastavuje požadovaná hodnota, kterou chceme za pomoci regulace dosáhnout. V demo aplikaci je cílová hodnota požadovaná výška míčku a lze ji volit z rozmezí 60 až 220 mm.
- **Měřená hodnota (s2)** – Vstupní hodnota pro blok PID, která je získávána měřením pomocí optického snímače. Je to aktuální výška míčku v mm.
- **Parametry PID (s3)** – Operand (s3) se skládá z několika parametrů, které jsou společně s rozsahy popsány níže. Parametry musí být při zápisu do globálních proměnných seřazeny za sebou (D100, D101, ...) a při tvorbě instrukce se zapisuje vždy pouze první parametr a funkce si další parametry čte sama.
 - **Sampling time (TS)** – 1 to 32767 [ms]
 - **Input filter constant (α)** – 0 to 99 [%]
 - **Proportional gain (KP)** – 1 to 32767 [%]
 - **Differential gain (KD)** – 0 to 100 [%]
 - **Differential time (TD)** – 0 to 32767 [$\times 10$ ms]
 - **Integral time (TI)** – 0 to 32767 [$\times 100$ ms]
- **PID řízení** – blok PID obsahuje funkci Auto-Tuning, která využívá dvě metody nastavení parametrů regulátoru. Blíže je funkce popsána v podkapitole 2.3.2.
 - **Auto-Tuning – Metoda Limit Cycle**
 - **Auto-Tuning – Metoda Step Response**
- **Funkce alarmu** – Funkci lze rozdělit na dva případy:
 - **Kontrola odchylky na vstupu** – pro měřenou hodnotu
 - **Kontrola odchylky na výstupu** – pro požadovanou hodnotu

Pokud dojde k překročení nastaveného alarmu u hodnot (měřená, požadovaná veličina), nastaví se příslušné bity na ON/OFF. V této aplikaci není tato funkce použita, jelikož je pro dané účely nepotřebná.

- **Nastavení mezních hodnot** – Pro výstupní hodnotu lze nastavit horní a dolní mez. Horní a dolní mez výstupní hodnoty zmírňuje nárůst integrační složky v PID regulaci [5].



Obrázek 2.3 Funkční blok PID regulátoru pro různé programovací jazyky [5]

2.3.2 Funkce Auto-Tuning

Funkce Auto-Tuning automaticky nalezne optimální parametry (proporcionální zesílení, časové konstanty, ...). Parametry PID regulátoru jsou nalezeny tak, aby se dosáhlo co nejoptimálnější regulace soustavy. Existují dvě metody pro nalezení parametrů. První metodou je Limit Cycle a druhou metodou je Step Response.

- **Metoda Limit Cycle** – Této metodě je zapotřebí nejprve předem nastavit parametry regulace, ze kterých je nalezena a (amplituda) a τ_{on} (cyklus kmitání). Z těchto parametrů jsou následně pomocí Tabulka 2.1. vypočteny optimální hodnoty pro PID regulaci. Pro výpočet je nutné nastavit dolní a horní limit výstupu regulátoru. Funkce generuje v době výpočtu PWM signál, ze kterého se po ukončení výpočtu stává spojitý signál ustalující se na žádané hodnotě.

Tabulka 2.1 Auto-Tuning – výpočty metody Limit Cycle [5]

Control type	Proportional gain (Kp) [%]	Integral time (TI) [x 100 ms]	Differential time (TD) [x 10 ms]
PID control (PID operation)	$\frac{1,2}{a} (ULV - LLV) * 100$	$20 * \tau_{on} (1 - \frac{\tau_{on}}{\tau})$	$20 * \tau_{on} (1 - \frac{\tau_{on}}{\tau})$

- **Metoda Step Response** – Druhou metodou pro nalezení optimálních parametrů PID regulátoru je metoda založená na změření přechodové charakteristiky. Metoda spočívá v tom, že získává parametry v době, kdy je na výstupu regulátoru jednotkový skok (ideálně 0–100 % možná hodnota výstupu regulátoru). V této době metoda z převodní charakteristiky soustavy nalezne parametry R (maximální náběh) a L (mrtvou dobu). Následně jsou z těchto parametrů pomocí Tabulka 2.2. vypočteny optimální hodnoty pro PID regulaci.

Tabulka 2.2 Auto-Tuning – výpočty metody Step Response [5]

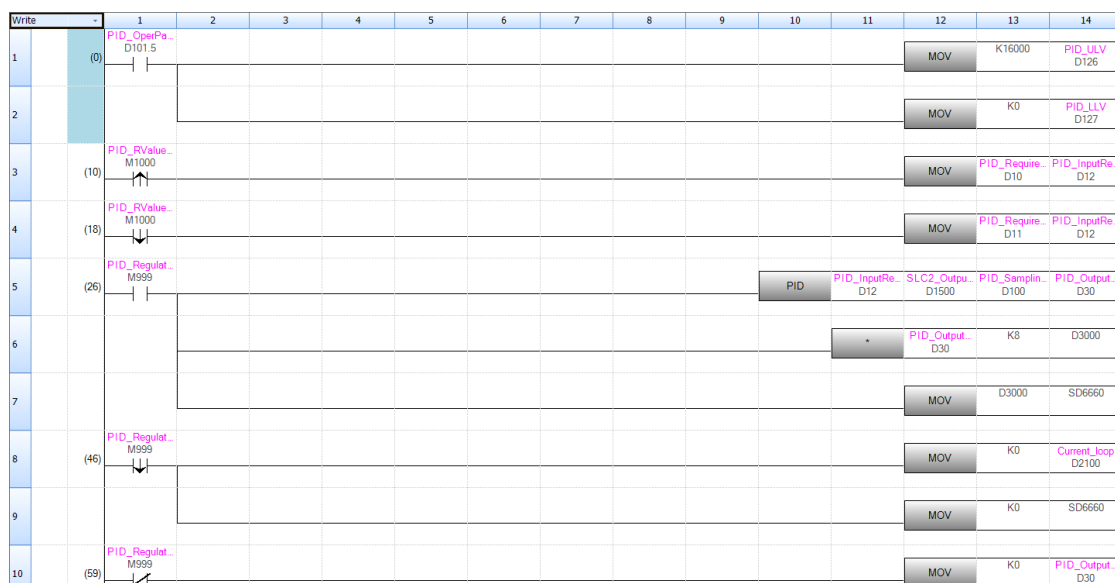
Control type	Proportional gain (Kp) [%]	Integral time (TI) [x 100 ms]	Differential time (TD) [x 10 ms]
PID control (PID operation)	$\frac{1,2}{RL} * Output\ value * 100$	20L	50L

2.4 Zdrojový kód PLC

Celý SW vytvořené demo aplikace se skládá z několika hlavních programů, kde každý plní určitou funkci a jsou mezi sebou provázány. Software je v programovacím jazyce Ladder vytvořený v programovacím prostředí GX Works3, který je podrobně popsán v podkapitole 5.2.1. Mezi hlavní části programu patří:

- **ProgPou** – Hlavní program, ve kterém je naimplementovaný funkční blok PID, popsáný v kapitole 2.3. Dále se zde používají funkce pro přehlednější zobrazení dat v uživatelském rozhraní (HMI).
- **ADConvScale** – Program, ve kterém dochází k výpočtu a škálování dat ze snímače pomocí funkce SCL2, která je popsána níže. Tímto blokem se vypočítává aktuální vzdálenost míčku od senzoru.
- **Init** – Část kódu, ve které dochází k nastavení proměnných na hodnotu, která se během chodu programu již nemění. Konstanty a bity nastaveny na logickou 1 nebo logickou 0.
- **ITAE** – Program, ve kterém se při stisku tlačítka na HMI do funkčního bloku PID regulátoru přepíše nastavené hodnoty vypočtené simulací z metody minimalizace ITAE kriteria.
- **Z_NMethod** – Kód, ve kterém se nahrávají parametry do PID funkčního bloku a následně se při stisku tlačítka do HMI nastaví předložené hodnoty.
- **Optimal_param** – Program, ve kterém se při stisku tlačítka na HMI do funkčního bloku PID regulátoru přepíše optimalizované parametry (úprava již vypočtených parametrů z předchozích metod). Je zde dále zajištěno zobrazování hodnoty těchto parametrů na HMI.
- **Auto-Tuning** – Program, ve kterém se nastavuje a pracuje s interní funkcí PID bloku, kde je nutností nastavit parametry popsané v kapitole 2.3.2. Volají se obě dvě metody nastavení funkce Auto-Tuning.

Kompletní zdrojový kód a záloha kódu je vložena do Příloha B – CD.



Obrázek 2.4 Řízení PID bloku

2.5 Funkce použité v SW

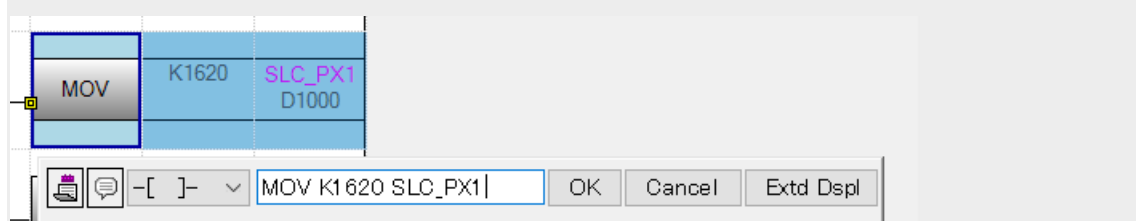
V této podkapitole jsou popsány jednotlivé funkce a příkazy (Ladder), které jsou používány ve zdrojovém kódu aplikace.

2.5.1 MOV

Základní instrukce, která slouží k přesunu a uložení hodnoty proměnné případně konstanty do jiné proměnné. Má pouze dva parametry, vstup a výstup. Vstupem je tedy hodnota, kterou chceme zapsat na výstup (jiná proměnná). Proměnné mohou být různého datového typu (např.: INT, DINT, WORD, DWORD, REAL). Výstupní proměnná je stejného datového typu jako zapisovaná vstupní proměnná. Zapisovaná hodnota začíná písmenem K jako konstanta [5].

Příklad zápisu:

MOV K1620 SCL_PX1 (zapisovaná hodnota do proměnné SCL_PX1 je 1620)

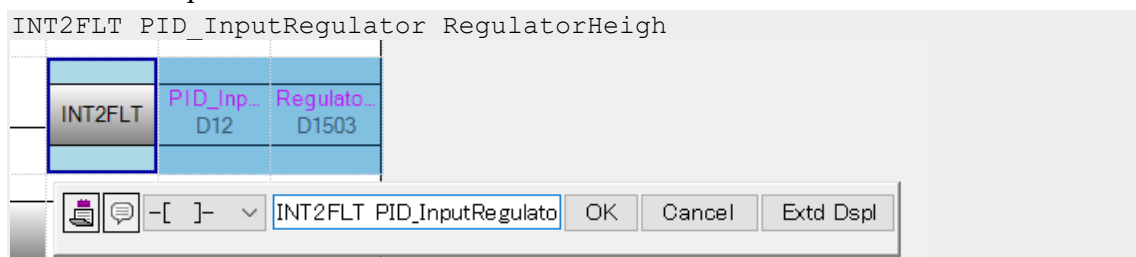


2.5.2 INT2FLT

Tato instrukce slouží pro přetypování datového typu. Konkrétně dochází k přetypování 16-ti bitových binárních znaménkových dat (INT) na reálné číslo datového typu FLOAT.

Instrukce má dva parametry (s) a (d), kdy (s) je vstup – data před přetypováním a (d) je výstup – data po převodu [5].

Příklad zápisu:

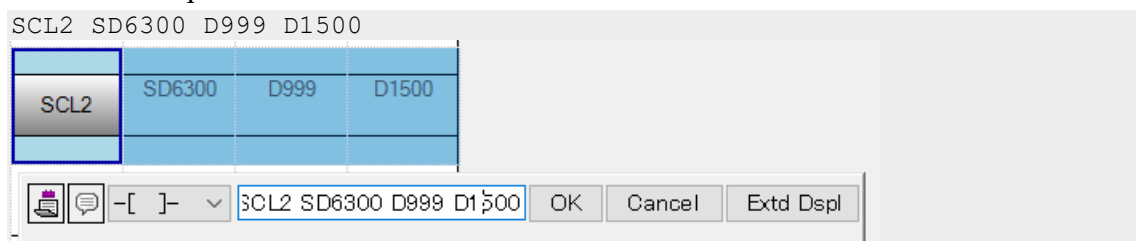


2.5.3 SCL2

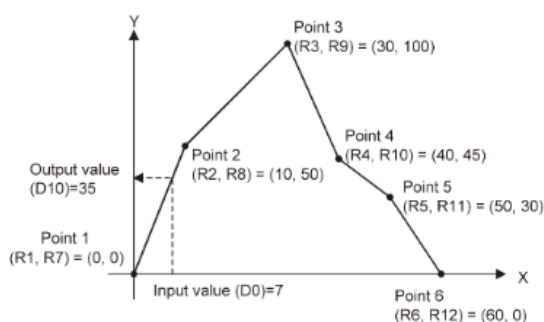
Funkce SCL2 slouží ke zpracování dat za pomoci škálování předložených hodnot. Tyto zpracovávané hodnoty musí být 16-ti bitové. Tato funkce má 3 parametry:

- **s1** – Vstupní hodnota použitá pro škálování.
- **s2** – Udává počet jednotlivých škálovaných bodů, které budou zpracovány a hodnoty jejich souřadnic (souřadnice X a Y jdoucích za sebou v pořadí, nejprve všechny body souřadnice X a poté všechny souřadnice bodů Y).
- **d** – Výstupní hodnota, do které se ukládají data po škálování [5].

Příklad zápisu:



• Operation



• Conversion table for scaling

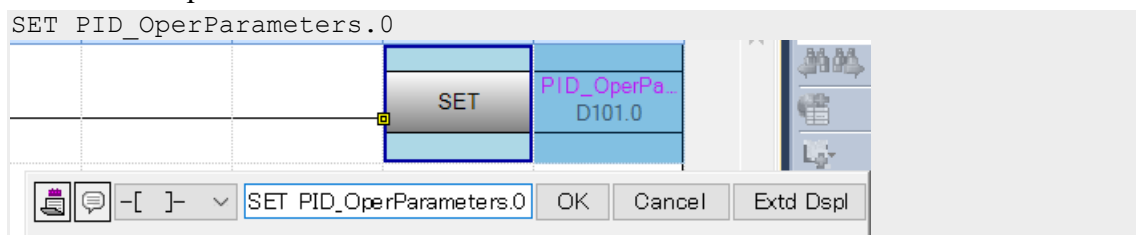
Set item	Device	Setting contents
Number of coordinate points	R0	K6
X coordinate	Point 1	R1
	Point 2	R2
	Point 3	R3
	Point 4	R4
	Point 5	R5
	Point 6	R6
Y coordinate	Point 1	R7
	Point 2	R8
	Point 3	R9
	Point 4	R10
	Point 5	R11
	Point 6	R12

Obrázek 2.5 Příklad použití funkce SCL2 [5]

2.5.4 SET

Funkce set nastaví požadované zařízení na hodnotu 1 (zapnuto) a zůstane tak do doby, dokud nedojde k jeho vypnutí (resetu). Tato funkce má jeden parametr a to zařízení, které chceme nastavit. Pokud se při scanování v programu vyskytne více funkcí SET na jedno zařízení, zůstane toto zařízení stále zapnuto i když je aktuálně vykonána jen jedna funkce SET z těchto funkcí [5].

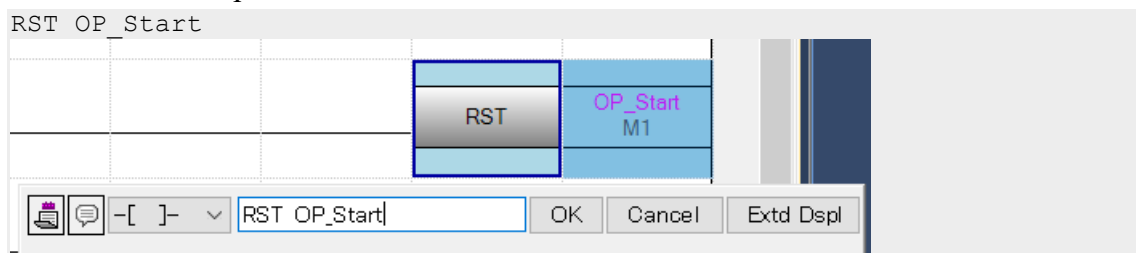
Příklad zápisu:



2.5.5 RST

Opačnou funkcí k funkci SET je funkce RST (reset). Význam této funkce je takový, že pokud bude tato funkce aktivní, dojde k vypnutí (resetu) zařízení, do doby dokud nedojde k opětovnému zapnutí (set). Při použití více funkcí RST na jedno zařízení během jednoho scanovacího cyklu, chová se tato funkce obdobně jako funkce SET.

Příklad zápisu:



Jestliže nastane situace, kdy během jednoho scanovacího cyklu jsou na jednom zařízení použity obě funkce (SET i RST), zařízení zůstane v původním stavu, a jeho hodnota se nikterak nezmění [5].

Celý zdrojový kód je k nahlédnutí v příloze bakalářské práce.

3. OPTICKÝ SENZOR SHARP GP2Y0A41SK0F

V této kapitole je popsán optický senzor Sharp GP2Y0A41SK0F, který slouží jako snímač polohy pingpongového míčku vznášejícího se nad trubicí. V následujícím textu je uveden základní princip, parametry a kalibrace tohoto snímače.

3.1 Princip a základní parametry

Optický senzor Sharp GP2Y0A41SK0F je snímač vzdálenosti pracující na principu infračerveného záření. Tento snímač je konstruován a vyroben společností SHARP a je zcela kompatibilní se všemi PLC, které disponují analogovými vstupy. Senzor má přímo ve svém těle umístěn vysílací a přijímací zdroj infračerveného záření. Toto záření se odráží od měřeného předmětu a senzor vyhodnocuje vzdálenost. Snímačem lze měřit jakékoliv předměty, kromě předmětů vyzařujících jakékoli zdroje světelného záření. Tyto jiné zdroje záření by se mohly následně ovlivňovat a detekce vzdálenosti by byla chybná, nebo nepřesná. Snímač je připojen pouze pomocí tří vodičů, kde jeden je napájecí, druhý uzemňovací a třetí datový s analogovým výstupem v podobě napětí v rozsahu 0 až + 3 V.

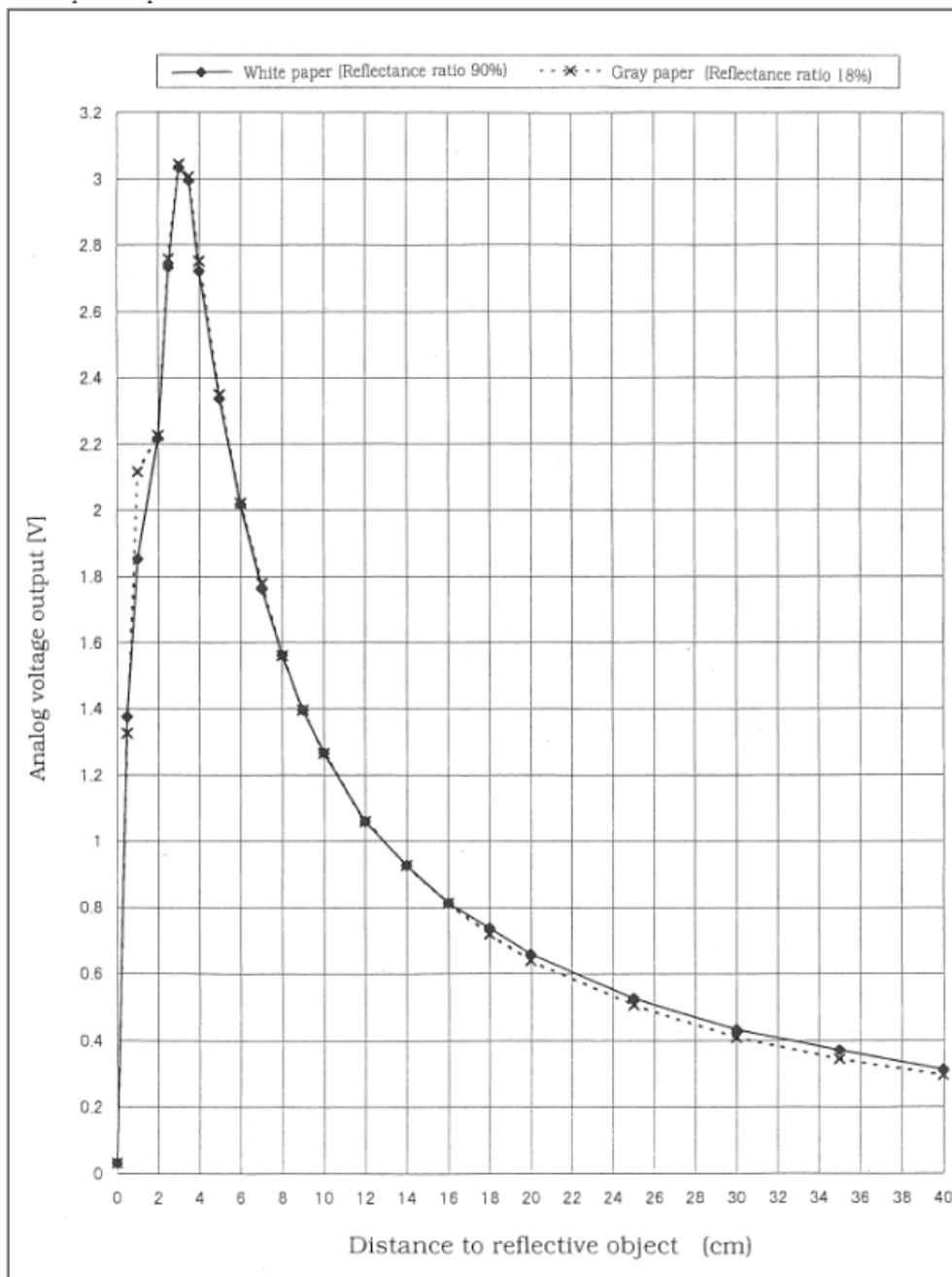
Snímač je schopen detekovat nacházející se předměty ve vzdálenosti 4–30 cm, nicméně jeho rozptyl hodnot na tomto rozsahu není lineární. Tato závislost je znázorněna na Obrázku 3.2. Provozní teplota snímače se nachází na rozsahu – 10 až + 60 °C, tudíž není určený pro použití v nepříznivých teplotních podmínkách. Optický senzor je napájen stejnosměrným napětím 5 V při proudovém odběru maximálně 12 mA. Tento snímač je nejpoužívanější pro modelářskou techniku, ale může se uplatnit i v automatizaci. Mezi nejběžnější nasazení jsou například robotická autíčka, nebo bezpečnostní hlídání polohy objektu, jakožto světelná závora [6].



Obrázek 3.1 Optický senzor Sharp GP2Y0A41SK0F [6]

■ Supplements

- Example of output distance characteristics



Sheet No.: OP13008EN

Obrázek 3.2 Závislost výstupního napětí na vzdálenosti [7]

3.2 Kalibrace

Kalibrace optického snímače se provádí z důvodu převodu napětí na analogový rozsah vstupu PLC. Výstupní hodnota snímače je v rozsahu napětí 0 až + 3 V, ale jelikož PLC je schopné rozlišovat jen proudovou smyčku, je zapotřebí toto napětí převést a snímač zkalibrovat. Jelikož je analogový vstup karty nastaven na proudovou smyčku, je zapotřebí převést napětí ze snímače na proud 4-20 mA. To zajišťuje převodník napětí na proud.

Pro kalibraci byl vybrán rozsah 5–40 cm. Počátek je zvolen z důvodu konstrukce soustavy, kde nulová poloha míčku je ve vzdálenosti 5 cm od snímače. Na Obrázku 3.2 je vynesena závislost napětí na vzdálenosti snímače, kde nalezneme konkrétní zvolené body použité v kalibraci. Bylo nutné pro kalibraci použít všechny body z tohoto rozsahu, jelikož tato závislost je nelineární. Nejprve je pracováno s hodnotami na ose Y, kde je příslušný rozsah napětí snímače převeden rovnoměrně na rozsah analogového vstupu karty (0–16000). Tento převod byl proveden pomocí nulového bodu, kde při napětí 2,37 (viz Obrázek 3.2) byla na vstupu do PLC hodnota 12800. Zbývající hodnoty byly dopočteny trojčlenkou. K těmto přepočteným analogovým hodnotám odpovídají na ose X příslušné vzdálenosti (200–4000 desetin mm pro větší přesnost).

Pro sestavení závislosti v PLC z těchto bodů a následné rovnice je použita funkce SCL2, uvedená v podkapitole 2.5.3. Tato funkce má tři parametry. Prvním parametrem je aktuální převedená výstupní hodnota ze snímače (0–16000). Druhým parametrem jsou data (1. bit – počet bodů, 2 až n bit jsou Y souřadnice a n+1 až 2n bit jsou X souřadnice). Třetím parametrem je aktuální výška míčku odpovídající prvnímu parametru. Dále bylo zapotřebí tuto hodnotu převést na milimetry a offsetem vynulovat. Kalibrace senzoru je realizována v programu MELSOFT GX Works3. Porovnání převedených závislostí napětí na vzdálenosti a digitální hodnotě na vzdálenosti je zobrazeno v Grafu 1 v Příloha A2.

4. HMI – MITSUBISHI GT2705-VTBD

V této kapitole je podrobně popsán operátorský panel Mitsubishi GT-2705-VTBD sloužící pro přehledné zobrazování aktuálních hodnot PID regulátoru. Dále je možné na operátorském panelu ručně nastavovat a měnit hodnoty konstant PID regulátoru.

4.1 Základní popis a parametry

Operátorský panel Mitsubishi GT-2705-VTBD je panel, pracující jako zobrazovač nebo ovladač k připojenému programu. Má 5,7palcový displej s rozlišením 640 x 480 pixelů. Jeho základem je TFT LCD panel, který následně umožňuje zobrazovat všechny kombinace RGB barev. V konečném stavu tedy dokáže zobrazovat až 65 tisíc barevných kombinací, což je přívětivé vizuální provedení pro uživatele. Tento LCD panel je nevyměnitelný, ale jeho provozní doba se pohybuje v řádech 50 tisíc hodin.

Nespornou výhodou panelu je jeho dotyková obrazovka umožňující flexibilní a jednoduché používání. Je primárně určen na montáž na panel, ale lze ho používat i jiným způsobem, jako například samostatně bez umístění na panel jako v této demo aplikaci. Pokud je správně umístěn, zaručuje krytí IP54, které má částečnou ochranu před prachem a ochranu proti stříkající vodě z každého směru [8]. Jeho hlavní jednotkou CPU je GOT2000. Operátorský panel lze používat v rozsahu teplot 0–50 °C, tudíž není vhodný pro montáže do náročnějších prostředí, kde se okolní teplota nepohybuje v okolí pokojové teploty. Na panelu lze nastavit až 32 úrovní jasového podsvícení displeje. Operátorský panel Mitsubishi GT-2705-VTBD je vybaven slotem na SD kartu, do které lze ukládat program, nebo jej nahrávat do operátorského panelu. Primárně má však základní interní 32MB ROM paměť, umožňující až 100 000x přepsání této paměti.

K programování operátorského panelu slouží slot na USB, pomocí něhož se připojuje panel k PC. Pro vytváření jak vizuální, tak i funkční stránky panelu slouží jeden ze softwarů, a to konkrétně MELSOFT Designer3, který je podrobně popsán v podkapitole 5.2.2 MELSOFT Designer3. Panel je napájen stejnosměrným napětím 24 V. Pro komunikaci mezi PLC a operátorským panelem slouží sběrnice RS – 485 nebo RS – 232. Pro připojení do sítě má v sobě panel předpřipraven slot na Ethernetový vodič s konektorem JR – 45 [9].



Obrázek 4.1 HMI GT-2705-VTBD [9]

4.2 Popis funkce záložek panelu

Jednotlivé záložky HMI jsou vytvářeny v programovacím softwaru GX Designer3, který je popsán v kapitole 5.2.2. Na obrázku 4.2 je několik vybraných záložek použitých v aplikaci.

Na hlavní záložce jsou tlačítka odkazující na další vytvořené záložky, které jsou popsány níže:

- **Actual system info** – Slouží k zobrazení aktuálních informací o soustavě, jako jsou velikosti jednotlivých složek PID regulátoru společně se vzorkovacím časem a aktuální výškou.
- **Setting parameters** – Záložka pro volbu vlastních konstant zadávaných pomocí klávesnice.
- **Graph** – Zobrazuje základní informace o soustavě (aktuální výška míčku, velikost žádané hodnoty, průběh PID regulátoru) a možnost zapnutí/vypnutí PID regulátoru.
- **Auto-Tuning** – Nastavuje a zobrazuje automaticky vypočtené konstanty PID regulátoru pro danou soustavu pomocí funkce Auto-Tuning. Možnost spuštění jednotlivých metod Auto-Tuning.
- **Methods** – slouží k volbě a přístupu k jednotlivým vypočteným metodám (ITAE, Ziegler-Nichols, ...)
- **PID Control** – Má funkci řízení a ovládání celé soustavy. Lze zde přepnout na manuální řízení.

Na těchto záložkách se v pravém horním rohu nachází tlačítka Menu, Back, Graph. Menu odkazuje na hlavní menu, přes které lze vstupovat do jednotlivých záložek. Šipka Back má funkci návratu na předchozí navštívenou záložku. Přes tlačítko Graph se lze

dostat na záložku s grafem. Všechna tlačítka jsou tvořena jako přepínač (switch), kterým je přiřazena nějaká funkce. Tato funkce může být například, při stisku tlačítka Required value (záložka s grafem) vykreslí nastavenou požadovanou hodnotu do grafu. Aby se mohl vytvořit graf, nejdříve se musely přidat proměnné přes Logging a až poté je možné tyto proměnné zobrazovat v grafu.

Na každé záložce, která se týká volby metody se nachází tlačítko START Simulation, které slouží pro nahrání konstant PID regulátoru do soustavy. Tyto konstanty jsou vypočteny pomocí metod ze simulovaného modelu návrhu soustavy. Hodnoty těchto konstant se zobrazují pomocí bloku Numerical display a nelze je přepisovat. Lze si zvolit i vlastní konstanty, které jsou přímo ručně zadávány pomocí klávesnice. Jednotlivé číslovky na klávesnici se zadávají do Numerical input bloků. Klávesnicové bloky jsou tvořeny Key code switchemi a jsou pevně svázány s odpovídajícím číslem či akcí (delete, desetinná čárka, ...). Pomocí potvrzujícího tlačítka Apply, se hodnoty pevně zapíší do globálních proměnných spojených s Numerical input blokem. Aktuální výška je zobrazena nejen číselně, ale i přes vizuální Bargraf s omezením.

Všechny switche jsou nastaveny jako průhledné a jednotlivé ikony těchto switchu jsou staženy z volně přístupných webových stránek material.design.icon.



Obrázek 4.2 Vybrané záložky HMI panelu

5. VÝVOJOVÁ PROSTŘEDÍ

V této kapitole jsou popsány vývojové prostředí, ve kterých se vytváří software k demo aplikaci. V podkapitole 5.1 je podrobně popsán MATLAB společně s identifikačním toolboxem pro nalezení přenosu soustavy. Podkapitola 5.2 se zabývá vývojovými prostředími, od společnosti Mitsubishi, IQ Works.

5.1 Vývojové prostředí MATLAB

MATLAB je prostředí vyvinuto společností MathWorks, ve kterém probíhají technické výpočty, vizualizace, modelování a implementace algoritmů a mnoho dalších. MATLAB dále pracuje na rozhraní API, což je knihovna, která obsahuje příkazy z jazyka C a Fortranu. Dříve sloužil MATLAB pouze jako maticová kalkulačka, ze které vyplývá i název tohoto softwaru z překladu Maticová laboratoř. Dnes již existuje IV generace tohoto softwaru, poslední verzí je R2022a. Součástí softwaru jsou tzv. toolboxy, které prostředí rozšiřují o nejrůznější funkce v odvětví zpracování signálů, neuronové sítě, simulace, řídicí systémy a mnoho dalších. Knihovna MATLABU obsahuje nejen elementární operace (součet, rozdíl), goniometrické funkce (sinus, cosinus) a komplexní čísla, ale i náročnější operace jako je práce s maticemi a nejrůznější transformace [10].

Důležitým doplňkovým produktem MATLABU je Simulink. Simulink poskytuje grafické prostředí pro modelování, simulaci a analýzu dynamických systémů. Grafické modelování a konstrukce systémů se vytváří pomocí blokových schémat v rozhraní (GUI – grafické uživatelské rozhraní). Jednotlivé předdefinované bloky se nacházejí v komplexní knihovně. Vybraný blok se z knihovny vkládá do programovací plochy Simulinku pouhým přetažením. Následné propojení těchto bloků se provádí pomocí čar, které symbolizují přenos signálu z jednoho bloku do druhého. Spojovací čára vede z výstupu jednoho bloku na vstup druhého, dále lze tuto čáru vést z uzlu a připojit ji na vstup nějakého bloku. V Simulinku lze pracovat s lineárními i nelineárními systémy modelované ve spojitém i diskretním čase. Existuje několik základních sekcí, do kterých se tyto bloky rozdělují:

- **Sources** – Tyto bloky se používají pro generování signálů. Vyznačují se tím, že mají pouze výstupy a neobsahují žádné vstupy. Mezi základní typy patří například Sine Wave, Ramp, Step a mnoho dalších.
- **Sinks** – Tento druh bloků se používá k zobrazení výstupních signálů. Například u bloku Scope, který se nachází v této kategorii, se při spuštění simulace otevře nové grafické okno s mřížkou a průběhem dané veličiny. Dalšími bloky jsou například Display, XY Graph a další. Vyznačují se tím, že mají pouze vstupy a neobsahují žádné výstupy.

- **Discrete** – Tato sekce bloků obsahuje většinou předdefinované předpisy diskrétních funkcí, které lze poté upravovat. Mezi některé patří například Discrete Transfer Fcn, Zero-Order Hold, Unit Delay a další.
- **Continuous** – Bloky v této konkrétní sekci jsou převážně složitější matematické funkce či funkční bloky. Mezi základní lze považovat PID Controller, Derivative, nebo Integrator.
- **Signal routing** – V této sekci lze najít všechny bloky umožňující směrování signálu. Mezi hlavní typy se řadí Mux, Demux a Switch.
- **Math** – V poslední základní sekci Math se nachází všechny matematické funkce. Mezi ty nejdůležitější lze zařadit například Abs, Suma, Gain a Sing [11].

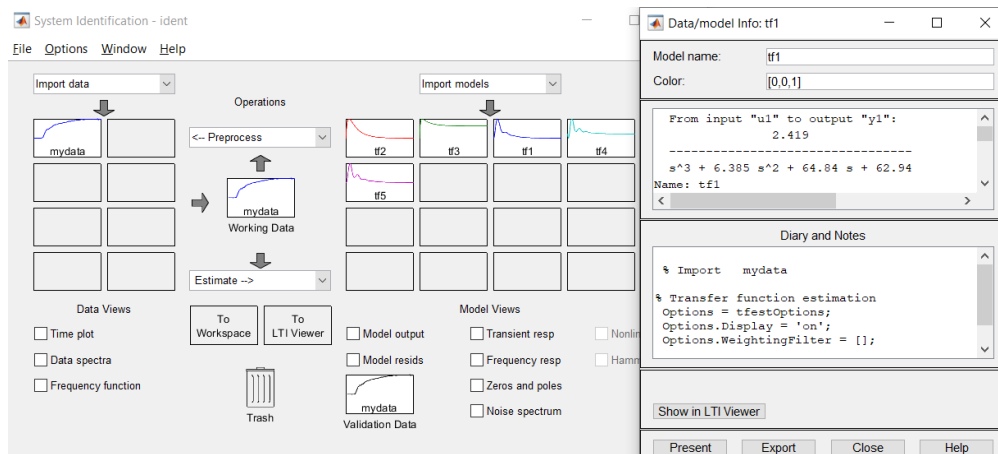
5.1.1 System identification toolbox

Toolbox sloužící k identifikaci dynamických systémů z I/O dat. Lze aplikovat na spojité i diskrétní modely, u kterých nelze na první pohled zjistit matematický přenos. Identifikace systému spočívá na principu odhadu parametrů po předložení naměřených dat z reálného modelu a následném zvolení určitého nastavení. Nastavením si lze zvolit například dobu zpoždění identifikace, nebo periodu vzorkování.

Vstupem do tohoto toolboxu jsou naměřená data, která se musejí nejprve do Matlabu importovat. Data se dají importovat v časové, frekvenční doméně nebo jako data object. Z importovaných dat je proveden odhad, který lze provést několika způsoby:

- Process Models
- Input-Output Polynomial Models
- State-Space Models
- Transfer Function Models
- Frequency-Response Models
- Correlation Models
- Nonlinear Model Identification
- Nonlinear ARX Models
- Hammerstein-Wiener Models

V demo aplikaci jsou data importována v časové doméně s použitým odhadem Transfer Function Models. Pro porovnání přesnosti identifikace bylo navrženo více modelů [12].



Obrázek 5.1 Identifikace soustavy

5.2 Vývojové prostředí IQ works

Vývojové prostředí IQ works je vyvinuto ve spolupráci s firmou Mitsubishi a slouží k programování PLC.

Tento software je rozdělen do podružných prostředí:

- **GX Works3**
- **MT Works2**
- **GT Works3**
- **RT ToolBox3**
- **FR Configurator2**

Každé podružné prostředí slouží k programování jiných částí sestavy (robot, PLC, HMI, ...). Pomocí hlavního softwaru MELSOFT Navigator vstupujeme do těchto jednotlivých prostředí. V tomto případě je použito GX Works3 pro vytvoření softwaru PLC a MELSOFT Designer 3 pro vytvoření designu na operátorském panelu.

5.2.1 MELSOFT GX Works3

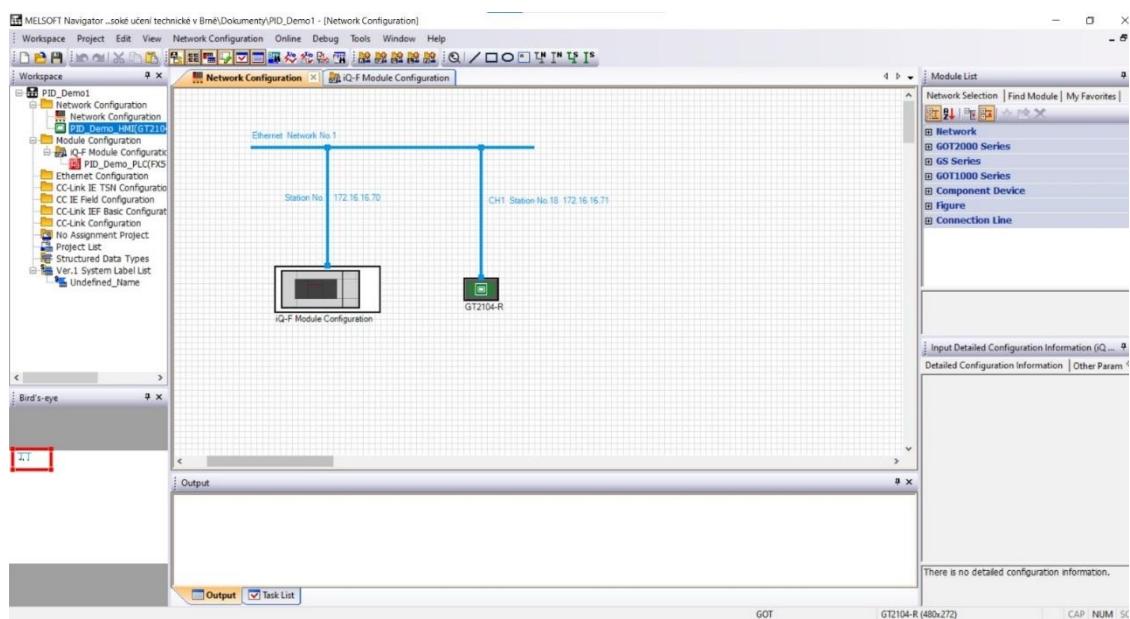
Prostředí sloužící k programování a údržbě softwaru pro PLC, které je vytvořené přímo společností Mitsubishi Electric. V tomto prostředí se nastavuje reálná sestava PLC a jeho komponenty (přídavné karty, ...). Moduly jsou umístěny do diagramu a propojeny pomocí virtuálních sběrnic. V těchto modulech lze dále nastavit parametry dle uživatelských potřeb (IP adresa, ...). V GX Works3 vzniká konkrétní kód úlohy, kterou PLC vykonává. Obsahuje dále možnost volby programovacího jazyka dle normy IEC (ST, FBD, LD). Umožňuje uživateli propojovat jazyky v jednotlivých programech pomocí funkčních bloků. Dále se v tomto prostředí inicializují proměnné a názvy vstupních a výstupních portů PLC. Melsoft GX Works 3 je také vybaven nástrojem pro nastavení pohybu, který umožňuje přehlednější nastavení parametrů servomotorů [13].

- **Programovací jazyk strukturovaného textu ST** – Tento programovací jazyk má své předchůdce v Pascalu a v jazyce C. Je poměrně jednoduchý, avšak pro většinu lidí může být i nepřehledný. Má velkou výhodu v programování například funkčních bloků. Lze ho propojovat s jinými jazyky. Každý řádek musí být na svém konci oddělen středníkem. Na jeden řádek je možno psát i více příkazů.
- **Programovací jazyk funkčních bloků FBD** – Je založen na grafickém rozložení jednotlivých bloků, které mají funkci příkazů. Tyto funkční bloky v podstatě představují jen bloky logických funkcí OR, AND a NOT. Jsou uspořádány sekvenčně za sebou a jejich vykonávání probíhá také sekvenčně.
- **Programovací diagramu LD** – LD je stejně jako FB grafický programovací jazyk mající však jinou strukturu. V Ladder diagramu jsou vytvořeny takzvané příčky (linky), na které se postupně umísťují prvky releové logiky. Na levou stranu spínací/rozpínací relé, představující podmínky vykonání jednotlivých větví a na pravou stranu funkce, které se mají vykonat. Tento programovací jazyk je velmi přehledný a je použit i v této demo aplikaci pro programování PLC [14].

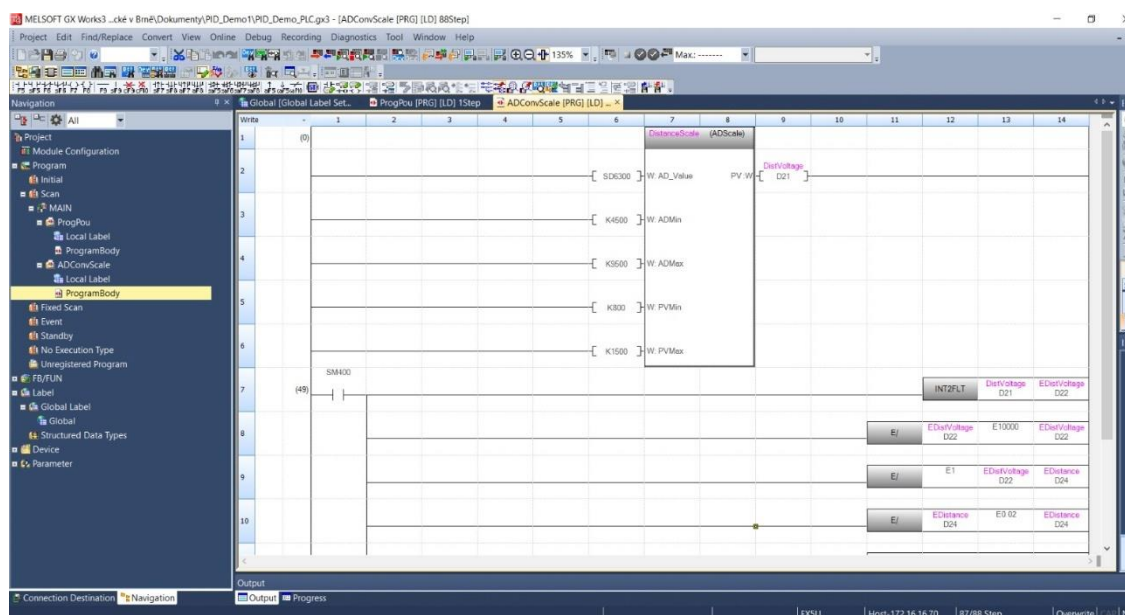
5.2.2 MELSOFT GT Designer3

GT Designer3 je software sloužící pro vytváření designu operátorského panelu. V tomto softwaru se vytvářejí jednotlivé plochy (pozadí), mezi kterými se lze na operátorském panelu přepínat. Lze vytvářet a přidávat libovolný počet obrazovek. Designer po spuštění nastavuje podbarvení obrazovky na černo, je však možností zvolit si design z předem předpřipravených barev pozadí.

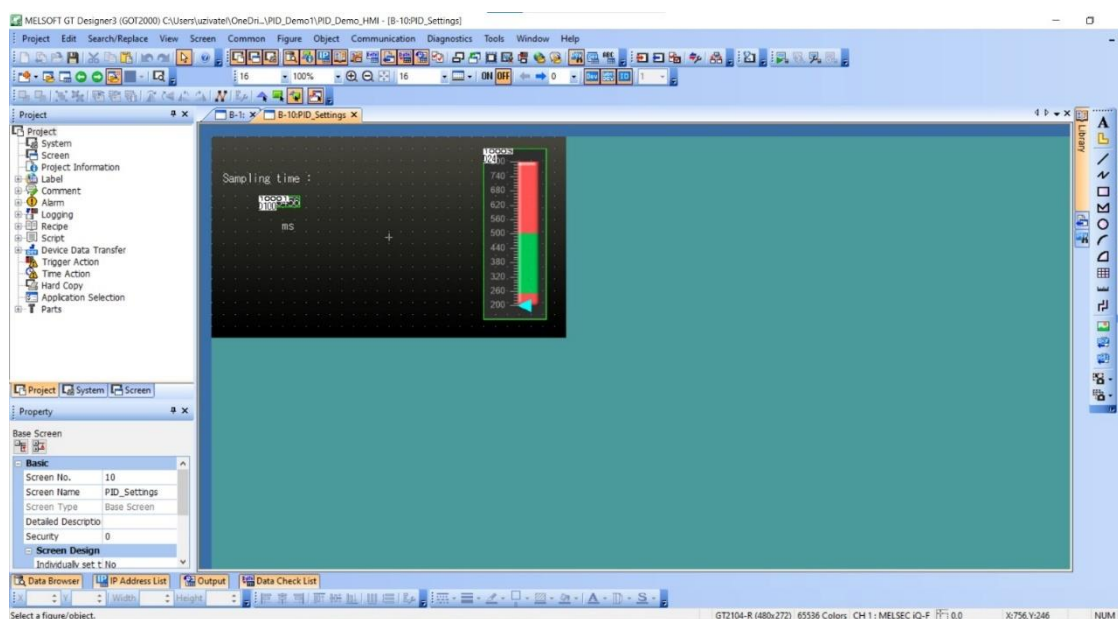
Dále se dají vkládat bloky (Numerical display) zobrazující aktuální hodnoty z POU, které lze s tímto zobrazovačem manuálně propojit. Tyto bloky se dají dále upravovat, měnit velikost, barvu, pozici, a další. Možností softwaru je také vkládání stažených obrázků, které však musí být umístěny ve stejné složce, jako je celý program. Dále je zde možnost vytvoření klávesnice, která se umísťuje na displej a plní úkol nastavování nebo změny jednotlivých parametrů soustav. V tomto programu se dají zobrazit jednotlivé průběhy aktuálních hodnot do grafů. Další podrobné informace o tomto softwaru jsou uvedeny v e – Manuálu [15].



Obrázek 5.2 Navigační software MELSOFT Navigator



Obrázek 5.3 Programovací prostředí MELSOFT GX Works3



Obrázek 5.4 Programovací prostředí MELSOFT GT Designer3

6. PŘÍSLUŠENSTVÍ SOUSTAVY

V této kapitole jsou podrobně popsány jednotlivé fyzické komponenty potřebné pro funkčnost soustavy. Jako první je v podkapitole 6.1 uveden motor pohánějící otáčející se vrtuli se zdrojem 24 V. Následně je v podkapitole 6.2 popsáno Raspberry Pi Pico sloužící v soustavě k pohánění bezkartáčového motoru přes driver uvedený níže. Nakonec je v podkapitole 6.3 uvedeno, jakým způsobem a na jakém konkrétním zařízení je zhotovena trubice, nad kterou se vznáší míček. Zapojení jednotlivých komponentů do sestavy je na Obrázek 6.5.

6.1 Bezkartáčový motor A2212/6T

V soustavě je umístěn BLDC motor outrunnerový 2200 kV A2212/6T. Outrunner znamená, že se točí jen obal motoru, prostřední část je statická. Tento motor pohání vrtuli, která následně vhání vzduch do trubice a udržuje konstantní výšku míčku. Motor je umístěn a připevněn tak, aby driver tohoto motoru byl neustále chlazen vzduchem, který vrtule vhání do soustavy. Tento motor je vhodný pro menší aplikace a je nejpoužívanější v modelářské technice. Rozměry motoru jsou 26,7 x 27,7 mm a hodí se do automatizačních aplikací. Má účinnost pohybující se okolo 80 %, což je pro účely této úlohy dostačující. Je napájen stejnosměrným napětím 24 V ze zdroje (podkapitola 6.1.1). Napájení je však možné pomocí stejnosměrného napětí 7 až 12 V. Lze dosáhnout otáček 2200 ot/V při příkonu 220 W. Celkový tah motoru je v rozsahu 300-1000 g [16].



Obrázek 6.1 Bezkartáčový motor A2212/6T [16]

6.1.1 Driver pro BLCD motor

Pro ovládání a regulaci otáček vrtule je k motoru připojen střídavý modelářský radič. Je určen pro BLCD motory s rozsahem napětí 3,7V-14,8V s proudovým odběrem

maximálně 30 A. Výstupem regulátoru je PWM signál s frekvencí do 8 kHz. Je vybaven detekcí připojené baterie. Dále je z bezpečnostních důvodů vybaven funkcí zastavení motoru při poruše. Pokud nejsou nastaveny minimální otáčky po připojení k napájení nedojde z bezpečnostních důvodů k roztočení motoru do doby, dokud se hodnota na vstupu nezmění. Je vybaven teplotní stabilizací pomocí vnějšího chladiče. Používá se v modelářské technice a je vhodný například pro řízení motorů u dronů [17].

6.1.2 Zdroj YS-100-12

Pro napájení bezkartáčového motoru je zapotřebí 12 V zdroj. Pro tuto demo aplikaci byl vybrán stejnosměrný napájecí zdroj YS-100-12. Spínaný zdroj od společnosti SANPU je chlazený pouze vzduchem a je schopen při konstantním napětí 12 V dodávat proud o velikosti až 8,3 A a výkon 100 W. Na jeho vrchní straně je jedna LED indikující stav zapnutí. Jeho vstupní (napájecí) napětí se nachází v rozsahu 170–240 V pro 50/60 Hz. Je schopen pracovat v rozsahu teplot od -20 až do 60 °C, což plně dostačuje pro demo aplikaci. Jeho účinnost je poměrně vysoká okolo 84 %. Z důvodu jeho konstrukce neobsahující žádné ochranné prvky, musí být zařízení uzemněno [18].



Obrázek 6.2 Zdroj YS-100-12

6.2 Raspberry Pi Pico

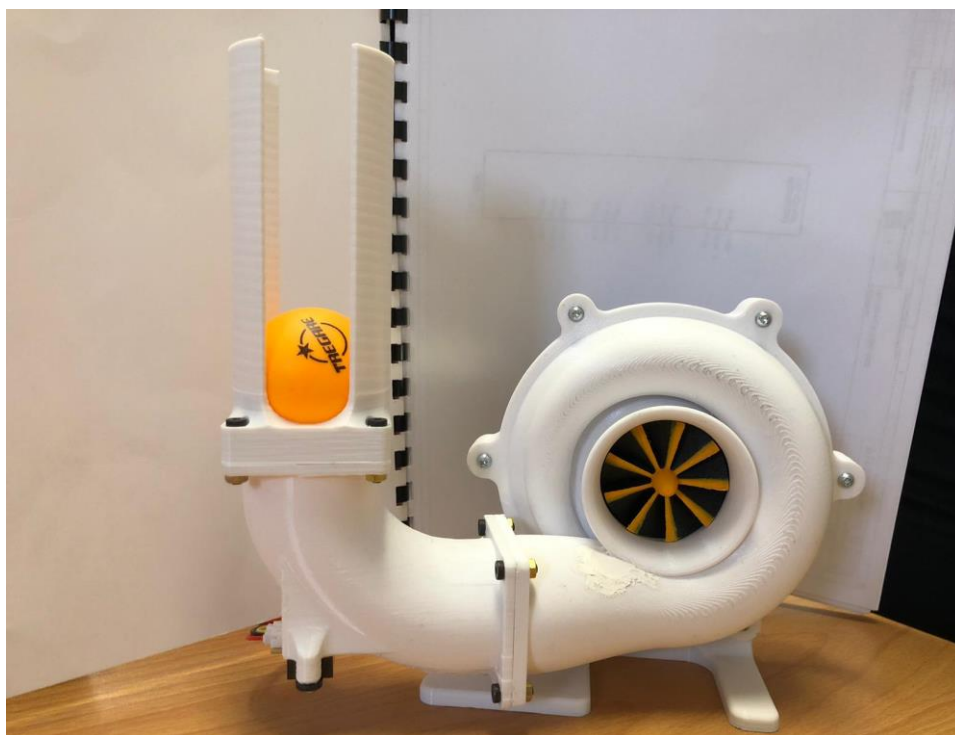
Pro pohon ventilátoru je použit bezkartáčový střídavý modelářský motor s driverem (regulátorem). Vstupní signál regulátoru je PWM s frekvencí 50 Hz (perioda 20ms). Šířka impulsu musí být v rozsahu 1 až 2 ms (1 ms motor zastaven, 2 ms motor v plných otáčkách). Takový signál nelze generovat pomocí PLC standartně, protože stav výstupů je aktualizován vždy na konci programového scanu, který v závislosti na složitosti

programu trvá jednotky až nižší desítky ms. Pro generování tohoto signálu pomocí PLC by tedy bylo nutné využít rychlé výstupy a časové přerušení, což je pokročilá programovací technika, vyžadující hlubší znalost konkrétního PLC. Záměrem aplikace je seznámení s vlastnostmi PID regulace, a nikoliv přerušovacího systému PLC. Toto řešení navíc umožňuje aplikaci použít s modulárními PLC (např. řady iQ-R nebo Q), které standartně rychlými výstupy nedisponují. Z těchto důvodů je použito Raspberry Pico s jednoduchým programem v jazyce Python. Je změřen proud smyčky (jako napětí na snímacím rezistoru 250 ohm), který je převeden na PWM pro řízení regulátoru. Samotný zdrojový kód pro tento převod byl vytvořen společností B:TECH a v této soustavě je použit pouze jako black box.

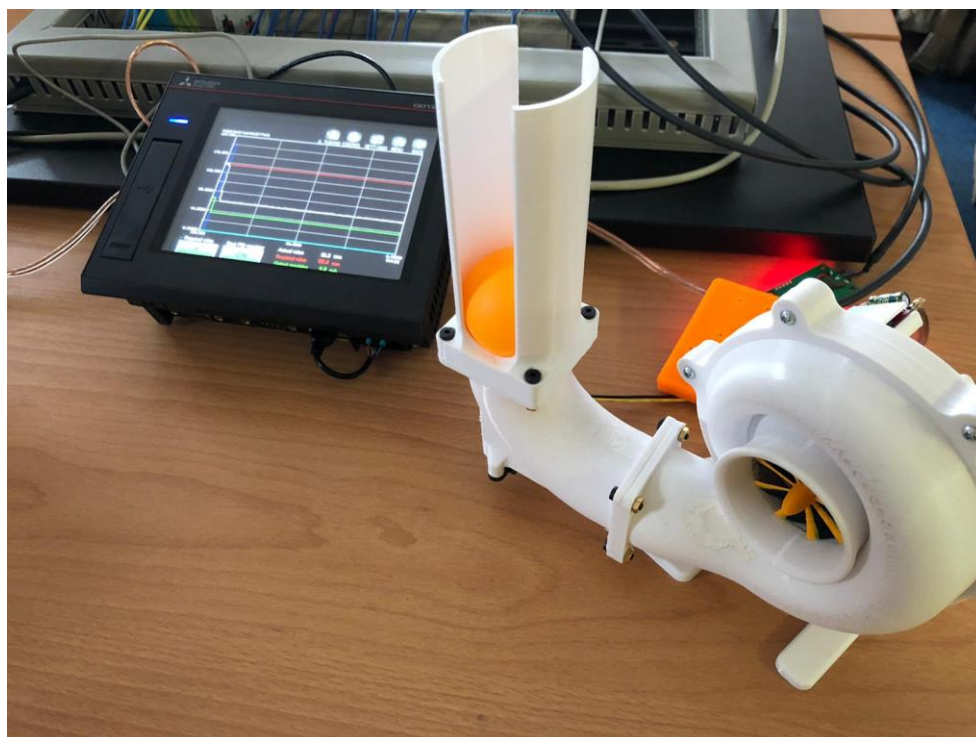
Raspberry Pi Pico je jednočipový počítač. Na destičce se nachází Flash paměť, oscilátorový krystal, napájecí zdroj, vstupní/výstupní piny a Micro-USB konektor pro připojení k PC. Flash paměť má velikost pouze 2 MB. Mikrokontroler je vybaven jen jedním čipem RP2040 s frekvencí 0 - 133MHz. Dále je na destičce 12-ti bitový AD převodník s rozlišením až 500ksps. Nachází se na něm celkem 30 vstupních nebo výstupních periférií, ale jen některé jsou schopné po nastavení v programu pracovat jako analogově digitální. Dále má za pomoci krystalu možnost do programu použít i časovač časující v reálném čase až se 4 alarmy. Raspberry Pi Pico může být napájeno několika způsoby, buď jednoduše po připojení k PC přes Micro – USB, nebo pomocí jednoho lithium-iontového článku anebo třemi AA články zapojenými v sérii. Celá destička je navržena tak, aby se pomocí jednotlivých pinů dala usadit do nepájivého pole, ale zároveň aby mohla fungovat i samostatně [19].

6.3 Praktický výrobek

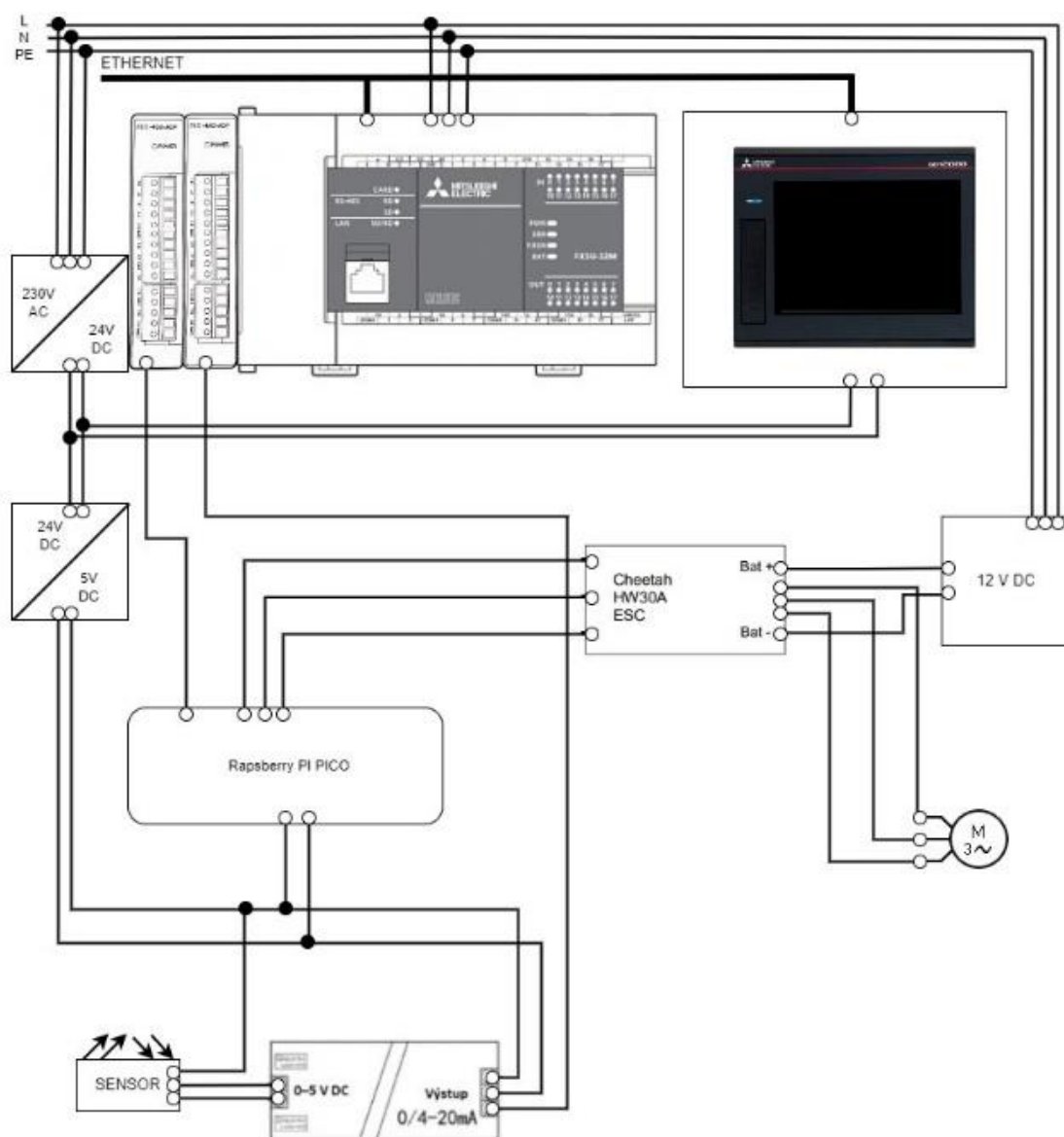
Hlavní částí celé demo aplikace pro regulaci soustavy pomocí PID regulátoru je trubice a držák pro vrtuli. Návrh trubice a držáku pro vrtuli byl podrobně zkonzultován s konzultantem, ale na samotné výrobě jsem se nikterak závažně nepodílel. Tato součást byla vytvořena společností B:TECH na 3D tiskárně. Konkrétní výrobek je vyobrazen na Obrázku 6.3 a 6.4.



Obrázek 6.3 Model výrobku s vrtulí



Obrázek 6.4 Model výrobku s HMI



Obrázek 6.5 Schéma zapojení demo aplikace

7. PID REGULÁTOR

V následující kapitole jsou podrobně popsány jednotlivé složky PID regulátoru, jejich výhody a nevýhody, funkce a použití. Následně je popsán reálný PID regulátor nasazený do regulačního obvodu.

7.1 Popis jednotlivých složek PID regulátoru

Regulátory jsou zařízení, které automaticky mění akční veličinu $x(t)$ čímž se snaží dosáhnout toho, aby hodnota regulované veličiny $y(t)$ byla shodná s žádanou veličinou $w(t)$. Existují různé typy regulátorů a jejich spojení jako jsou například P, I, PI, PD nebo PID regulátor. Tyto jednotlivé složky se při jejich spojování sčítají. Samostatné složky se moc nepoužívají, spíše dochází k jejich spojování. V tomto případě se pracuje s nejpoužívanějším PID regulátorem, který se skládá ze tří složek, P (proporcionální) složky, I (integrační) složky a D (derivační) složky. Každá z těchto složek má určitý vliv na celou regulovanou soustavu a jejich hodnota zásadně mění velikost regulační odchylky, popřípadě dobu náběhu či průtahu. Při volbě velikosti těchto hodnot je snahou dosáhnout žádanou hodnotu v co nejkratším čase, ale tak aby systém zůstal stabilní bez překmitu nad žádanou hodnotou.

7.1.1 P – složka

Proporcionální složka v regulátoru má funkci zesílení. Akční veličina $x(t)$ vystupující z regulátoru je tedy přímo úměrná rozdílu žádané veličiny $w(t)$ a výstupní veličiny $y(t)$. To v podstatě odpovídá velikosti odchylky $e(t)$ vynásobené proporcionálním zesílením K_R .

Pro vztah popisující chování akční veličiny P složky poté platí:

$$x(t) = K_R e(t), \quad (7.1)$$

kde $x(t)$ představuje akční veličinu, K_R je zesílení regulátoru a $e(t)$ je velikost regulační odchylky.

Ve většině případech platí, že čím větší se zvolí hodnota zesílení K_R , tím je pravděpodobnější, že dojde k rozkmitání soustavy a to až do možnosti poškození právě používaného zařízení. Pokud v soustavě není přítomna integrační složka, tak samotný P regulátor nedokáže vyregulovat regulační odchylku na nulu pro konstantní žádanou hodnotu ani pro poruchu, která působí na vstupu soustavy [20].

7.1.2 I – složka

Integrační složka v regulátoru slouží k zajištění nulové regulační odchylky regulačního obvodu. Integrační časová konstanta T_i popisuje, jak moc velká regulační odchylka vstupuje do regulace.

Pro vztah popisující chování I složky platí:

$$x(t) = \frac{1}{T_i} \int_0^t e(t) dt + x(0), \quad (7.2)$$

kde $x(t)$ představuje akční veličinu, T_i je integrační časová konstanta a $e(t)$ je velikost regulační odchylky.

Aby byl účinek integrační složky co nejrychlejší, při zvětšující se regulační odchylce, musí být integrační konstanta velmi malá. Jinak řečeno, čím větší je integrační konstanta, tím menší je vliv integrační složky a naopak. Zvolením malé integrační konstanty také dochází k větší kmitavosti regulované soustavy [21]. Nevýhodou této složky je, že zhoršuje dynamické vlastnosti regulátoru a zpomaluje dobu regulace, jelikož nulování regulační odchylky dochází až za určitý čas. Realizace integračního regulátoru se provádí například pomocí stejnosměrného invertujícího zesilovače [22].

7.1.3 D složka

Derivační složka v regulátoru slouží ke zrychlení regulačního děje a ke zlepšení vlastností regulačního pochodu. Derivační konstantou T_D měníme intenzitu D složky, reaguje na rychlost změny skutečné hodnoty. Podle nastavení derivačního času se zvyšuje či snižuje akční veličina [21].

Pro vztah definující chování akční veličiny D složky platí:

$$x(t) = T_D \frac{de(t)}{dt}, \quad (7.3)$$

kde $x(t)$ představuje akční veličinu, T_D je derivační časová konstanta a $e(t)$ je velikost regulační odchylky.

Pokud se regulovaná veličina změní, derivační složka působí vždy proti této změně. Například při působení poruchy v systému v důsledku snižování regulované veličiny D složka působí opačně, takže působí proti snížení regulované veličiny a naopak. Nevýhodou derivační složky je, že nedokáže vyregulovat děj na nízkých frekvencích (šum). Derivační regulátor se nepoužívá samostatně, ale jen ve spojení s proporcionálním regulátorem (PD), případně s integračním a proporcionálním (PID). Při kombinaci těchto dvou regulátorů (PD regulátor) se zvyšuje stabilita systému a zrychluje regulace, což má velký přínos pro odstranění krátkodobých poruch [20] [22].

7.2 PID regulátor – popis

Tento regulátor obsahuje všechny tři regulační složky ve výstupním signálu, které jsou popsány viz výše v podkapitolách 7.1.1, 7.1.2 a 7.1.3. Výsledné chování PID regulátoru ovlivňuje každá ze složek prostřednictvím nastavitelných konstant neboli parametrů regulátoru. Správná volba těchto parametrů se dá získat vícero způsoby, například metodou optimalizace ITAE kritéria, nebo metodou Ziegler-Nicholse a dalšími, vysvětleno níže v podkapitole 7.3. PID regulátor vznikne paralelním spojením jednotlivých regulátorů P, I a D. Obecně úkolem regulátoru je měnit akční veličinu tak,

aby regulovaná veličina byla stejná jako žádaná.

Pro vztah přenosu akční veličiny PID regulátoru platí:

$$x(t) = K_r e(t) + T_D \frac{de(t)}{dt} + \frac{1}{T_I} \int_0^t e(t) dt + x(0), \quad (7.4)$$

kde $x(t)$ představuje akční veličinu a každý člen rovnice představuje jednotlivou složku PID.

Regulátor se soustavou tvoří regulační smyčku, kde na vstup regulátoru je přiveden rozdíl požadované hodnoty od skutečné hodnoty regulované veličiny. Nenulová regulační odchylka může být způsobena poruchou u regulované veličiny nebo změnou žádané hodnoty, která je cílená. V praxi má většina soustav charakter dynamického nekmitavého systému, který většinou bývá prvního nebo druhého řádu s případným dopravním zpožděním [21][22].

7.3 Použité metody pro návrh regulátoru

Pro návrh PID regulátoru byly použity metody, které jsou popsány v následujících podkapitolách 7.3.1 a 7.3.2.

7.3.1 Metoda optimalizace ITAE kritéria

Váhové kritérium, které pracuje na principu absolutní hodnoty z odchylky. Tato metoda je specifická tím, že neobsahuje lineární funkci a tudíž pro realizaci je potřeba kritérium realizovat za pomoci simulace.

Pro vztah integrálního kritéria ITAE platí:

$$J_{ITAE} = \int_0^t |e(t) - e(\infty)| t dt, \quad (7.5)$$

kde $e(t)$ představuje časový průběh regulační odchylky a $e(\infty)$ trvalou ustálenou odchylku a t je čas.

Kritérium ITAE funguje tak, že se nejprve udělá absolutní hodnota z průběhu odchylky a poté se tento upravený průběh integruje. Nakonec se hodnota odchylky násobí váhovou funkcí, která s časem lineárně narůstá [20]. V MATLABU se optimalizace kritéria realizuje funkcí `fminsearch`. Tato funkce vyhledává lokální minimum vícerozměrného prostoru. Princip vyhledávání je založen na Nelder-Mead algoritmu. [23].

7.3.2 Metoda Ziegler-Nichols

Slouží pro nalezení optimálních parametrů PID regulátoru. Metoda vyžaduje přesnou znalost soustavy, popřípadě simulaci, pro zjištění kritických parametrů jako je kritické zesílení K_{krit} a kritická perioda T_{krit} .

Princip nalezení parametrů K_{krit} a T_{krit} :

- **První krok** – vyřazení integrační a derivační konstanty PID regulátoru ($T_D = 0$ a $T_I = \infty$)

- **Druhý krok** – zvyšujeme zesílení K do doby, dokud nedosáhne obvod meze stability. Zvýšením zesílení se rozkmitá soustava na netlumené kmity, hodnota tohoto zesílení se označuje jako K_{krit} . Perioda se kterou kmitá výstup regulačního obvodu se nazývá T_{krit} [20][20].

Z předdefinované tabulky viz Tabulka 7.1. a získaných kritických parametrů se poté výpočtem zjistí skutečné zesílení K a časové konstanty T . Pro každý regulátor P, PI nebo PID jsou hodnoty z tabulek rozdílné, a proto pro každý regulátor je jiné zesílení a časová konstanta.

Tabulka 7.1 Tabulka pro výpočet konstant regulátoru pomocí Z-N metody [20]

Typ regulátoru	K	Ti	Td
P	$0,5K_{krit}$	-	-
PI	$0,45K_{krit}$	$0,85T_{krit}$	-
PID	$0,6K_{krit}$	$0,5T_{krit}$	$0,125T_{krit}$

Pokud však není možno rozkmitat soustavu z bezpečnostních důvodů konstrukce používaného zařízení, lze použít k nalezení kritických parametrů i další možnosti:

- **Nalezení kritických parametrů z přechodové charakteristiky** – z přechodové charakteristiky se určí doba náběhu T_n a průtahu T_u se následně dle vzorců (7.7 a 7.8) vypočtou kritické parametry

Výpočet kritického zesílení je definován vztahem

$$K_{krit} \doteq \frac{\pi T_n}{2T_u} + 1, \quad (7.7)$$

kde K_{krit} představuje kritické zesílení, T_n je doba náběhu, T_u reprezentuje dobu průtahu.

Výpočet kritické periody je definován vztahem

$$T_{krit} \doteq 4T_u, \quad (7.8)$$

kde T_{krit} představuje kritickou periodu kmitů, T_u je doba průtahu.

- **Nalezení kritických parametrů ze známého modelu soustavy** – Pokud známe přesný matematický model soustavy, kritické zesílení lze vypočítat za použití algebraických kritérií stability jako je například Hurwitzovo kritérium či Routh-Schurovo kritérium.
- **Nalezení kritických parametrů rozkmitáváním za použití relé bez hystereze** – V některých případech se pro rozkmitání soustavy používá relé bez hystereze, kdy nahradíme tímto relé proporcionální složku. Tím docílíme řízených kmitů a pro nalezení kritických parametrů lze dále postupovat dle obecného principu [20].

8. SIMULACE NAVRŽENÉHO MODELU

V následující kapitole je podrobně popsán návrh řešení celé demo aplikace na regulaci výšky pingpongového míčku za pomoci PSD regulátoru. Navržený PSD regulátor je popsán v podkapitole 8.1. Dále je v podkapitole 8.2 popsán teoretický návrh soustavy. Výsledkem simulace je návrh hodnot parametrů PID regulátoru pomocí dvou zvolených metod.

8.1 Návrh PSD regulátoru

Ve výše uvedeném programovacím prostředí MATLAB byl pro tuto demo aplikaci vytvořen PSD regulátor za pomoci simulačního produktu Simulink (zdrojový kód je v Příloha B - CD). Jelikož je vnitřní zapojení funkčního bloku PID regulátoru realizováno jako PI-D regulátor (Obrázek 2.2), je nutné realizovat PS-D regulátor (Beta PSD regulátor). Vytvořený PS-D regulátor se skládá z několika funkčních bloků. Základem simulačního schématu s PS-D regulátorem je vstupní blok Step, který nám udává požadovanou hodnotu, které chceme během simulace dosáhnout. Velikost této požadované hodnoty byla nastavena na 150, což odpovídá hodnotě jednotkového skoku přivedeného na vstup soustavy. Volba velikosti jednotkového skoku je popsána v podkapitole 8.2. Dalším blokem je sumátor pro zápornou zpětnou vazbu, který slouží k získání regulační odchylky. Odchylka je následně vstupem do regulátoru, který je složen ze tří složek P, S a D. P složka je reprezentována pouze zesílením K. Složka S je sumační složka složená z integrační časové konstanty a integrátoru. Tato složka však prodlužuje přechodový děj, a proto je nutné použít omezení. K tomuto ději dochází, když velikost hodnoty napětí na integrátoru překročí maximální hodnotu signálu. Napětí na integrátoru bez omezení může růst do nekonečna. Aby se zabránilo přebuzení, je zde tedy použit tzv. Antiwindup. D složka je v regulačním obvodu realizována s realizační konstantou ε (T_d/N), která se stará o filtraci [24]. Parametry jednotlivých složek jsou získány z použití metod Ziegler-Nichols a Integrační kritérium ITAE. Vzorkovací perioda je použita stejná jako je perioda PID bloku $T_{vz} = 0,01$ s.

- Získané parametry – Ziegler – Nichols
 - $K = 87,08 \%$
 - $T_i = 0,39$ s
 - $T_d = 0,097$ s
- Získané parametry – ITAE kritérium
 - $K = 64,08 \%$
 - $T_i = 0,87$ s
 - $T_d = 0,05$ s
- Získané parametry – Upravené ITAE kritérium v PLC
 - $K = 56 \%$

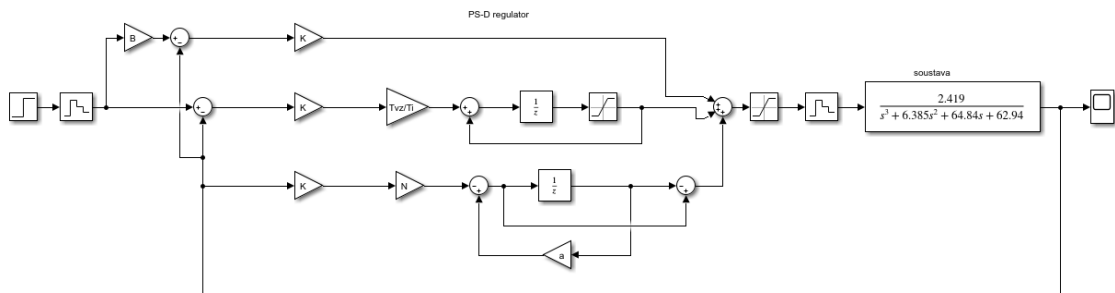
- $T_i = 0,2 \text{ s}$
- $T_d = 0,68 \text{ s}$

Výsledný přenos PID regulátoru je definován vztahem:

$$F_R = K \left(\beta + \frac{T_{vz}}{T_i} \frac{z^{-1}}{1-z^{-1}} - N \frac{(1-z^{-1})}{1 - e^{-\frac{T_{vz}N}{T_d}} z^{-1}} \right), \quad (8.1)$$

kde K je zesílení soustavy, T_i a T_d jsou časové konstanty regulátoru, N je filtr derivační složky PS-D regulátoru a z je operátor přenosu. Parametr β označuje o jakou strukturu regulačního obvodu se jedná (S-PD nebo PS-D). Dále T_{vz} je perioda vzorkování.

Průběhy porovnání návrhů jednotlivých metod s průběhy v PLC jsou vyneseny v grafech v příloze (Příloha A1).



Obrázek 8.1 Simulační schéma regulačního obvodu v MATLABU

Jelikož se jedná o PS-D regulátor je parametr β roven 1. Saturace je nastavena od 0 do 6000 což odpovídá omezenému rozlišení analogového výstupu (4–10 mA). Omezení je nastaveno z důvodu konstrukce soustavy, jelikož není nutné dosahovat vyšších otáček motoru a velké výšky míčku. Parametr a v simulačním obvodu představuje $e^{-\frac{T_{vz}N}{T_d}}$ [24]. Zpracování naměřených dat je uvedeno v příloze (Příloha B – CD).

8.2 Návrh soustavy

Předpoklad pro přenos soustavy je systém 3. řádu, jelikož soustava je ovlivněna hmotností míčku, která způsobuje určitou dobu průtahu. Z tohoto předpokladu soustava nemůže být prvního ani druhého řádu. Dále nemůže nastat jev, že by se při minimálních otáčkách motoru stejně zvyšovala výška míčku, a proto je průběh nelineární. Další vlastností průběhu je doba náběhu, která se ustálí na požadované hodnotě, jelikož víry z vrtule, které obtékají povrch míčku, se rozprostírají do okolí, a i při zvyšujících se otáčkách motoru již míček nebude stoupat.

Návrh soustavy je vytvořen pomocí funkce System identification ve vývojovém prostředí MATLAB. Soustava je navržena ze změřených dat chování soustavy, která jsou získávána pomocí PLC. Data jsou ukládána přes FTP protokol na server. Tyto data se ukládají do 100 .csv souborů v 5 min intervalu. Aktuální data o soustavě jsou ukládána

do těchto souborů s periodou 10 ms. Po naplnění všech 100 souborů se začnou soubory postupně přepisovat novými daty (od nejstaršího záznamu).

Vstupem do soustavy je jednotkový skok, který v daném případě znamená skokovou změnu napětí na motoru. Zvýšením napětí na motoru se zvýší i otáčky a tím docílíme skokové změny vstupu. Data pro identifikaci soustavy byla změřena při jednotkovém skoku nastaveném na 6000 (hodnota na analogovém výstupu karty PLC). Při této hodnotě se míček ustálil ve výšce 150 mm a z tohoto důvodu je v simulaci nastaven jednotkový skok 150 mm pro nalezení optimálních parametrů.

Tabulka 8.1 Tabulka srovnání identifikovaných přenosů soustav

Počet kořenů		Přenos soustavy	Přesnost identifikace [%]
Póly	Nuly		
2	0	$F_s = \frac{0.3102}{s^2 + 8.199s + 8.077}$	90.3
2	1	$F_s = \frac{-0.06308s + 0.5965}{s^2 + 14.7s + 15.53}$	90.72
3	0	$F_s = \frac{2.419}{s^3 + 6.385s^2 + 64.84s + 62.94}$	91.66
3	1	$F_s = \frac{-0.4962s + 4.236}{s^3 + 5.756s^2 + 105.8s + 110.3}$	92.11
4	0	$F_s = \frac{52.94}{s^4 + 18.31s^3 + 166.8s^2 + 1415s + 1378}$	92

Z tabulky 8.1 byla vybrána soustava pro tuto aplikaci 3.řádu s přesností identifikace 91.66 %. Soustava má 3 póly a žádnou nulu.

Přenos soustavy je definován vztahem

$$F_s = \frac{K}{as^3 + bs^2 + cs + d} = \frac{2.419}{s^3 + 6.385s^2 + 64.84s + 62.94} \quad (8.2)$$

kde s je operátor přenosu soustavy. Parametry a , b , c , d jsou hodnoty parametrů soustavy a K je hodnota zesílení.

9. ZÁVĚR

Cílem bakalářské práce bylo navrhnout a následně realizovat reálnou soustavu, na které je reprezentována funkčnost PID regulátoru. Základem práce bylo seznámit se s hardwarem a softwarem programovatelného logického automatu od Společnosti Mitsubishi, a to konkrétně s PLC IQ-F. Dále bylo zapotřebí seznámit se v PLC s implementací PID regulátoru v programu GX Works3 a naučit se správně nastavit a nalézt ideální parametry PID regulátoru na soustavě za pomoci vhodně zvolených metod. Dále bylo úkolem porovnat parametry nalezené pomocí vhodně zvolených metod s parametry vypočtenými vnitřní funkcí PLC, Auto-Tuning.

Prvním krokem v návrhu dané aplikace byl výběr vhodných součástí pro řízení soustavy. Hlavní součástky jako je PLC a operátorský panel byly automaticky přiděleny bez možnosti výběru, nicméně další komponenty byly vybírány dle uvážení a parametrů. Mezi ty nejdůležitější patří například optický snímač vzdálenosti Sharp GP2Y0A41SK0F, který je svojí konstrukcí a snímatelnou vzdáleností v porovnání s cenou nejadekvátnější. I přes nevhodný výstupní signálu (napětí, PLC potřebuje proudovou smyčku), byl po kalibraci použit pro tuto aplikaci. Dalšími komponenty byly motor s regulátorem pro řízení otáček vrtule a Raspberry Pi Pico pro převod výstupního signálu z PLC na PWM signál pro řízení motoru. Jelikož vstupní signál regulátoru je PWM s frekvencí 50 Hz a šířkou impulsu v rozsahu 1 až 2 ms (popsáno v podkapitole 6.2), tak PLC není schopno takový signál standartně vygenerovat (nutná hlubší znalost programování konkrétního PLC), protože stav výstupů je aktualizován vždy na konci programového scanu. Z tohoto důvodu je v aplikaci použito Raspberry Pi Pico, které na vstupu dostává signál z PLC (0-16000) a na výstup generuje potřebný PWM signál. Software pro Raspberry Pi Pico byl předložen jako blackbox s předchozí funkčností.

Pro realizaci soustavy bylo zapotřebí vytvořit reálný model vytisknutý na 3D tiskárně (Obrázek 6.3 a 6.4), na kterém PID regulátor reguluje a řídí výšku pingpongového míčku. Po sestavení a připojení modelu k PLC byla pomocí protokolu FTP změřena data. Na vstup soustavy byl přiveden jednotkový skok popsáný v kapitole 8.2 a na výstupu byla měřena aktuální výška míčku v závislosti na čase. Z těchto naměřených dat byl pomocí funkce Ident v MATLABU nalezen přenos soustavy (podkapitola 8.2). Přenos byl použit pro realizaci simulačního obvodu (Obrázek 8.1) PID (z důvodu vnitřního zapojení v PLC – PS-D) regulátoru. Pro návrh PID regulátoru a nalezení parametrů byly použity výpočetní metody (Ziegler – Nichols a optimalizace ITAE kritéria). Vypočtené parametry PID regulátoru byly nahrány do PLC.

V PLC byl vytvořen software realizující PID regulátor, za pomoci knihovního bloku PID, jehož parametry byly získány ze simulace. Porovnání simulačních a reálných výsledků výpočetních metod jsou vyneseny do grafu (Příloha A1 – Graf 2, Graf 3). Hodnoty z PLC v grafech mají offset 60 mm z důvodu konstrukce modelu. Kromě výpočetních metod pro získání parametrů byly použity vnitřní funkce PID bloku (Auto-

Tuning). Auto-Tuning je vybaven dvěma metodami (Step Response a Limit Cycle), obě tyto metody byly implementovány v PLC. Z výsledku grafu (Příloha A1 – Graf 4) je patrné, že metoda Limit Cycle není schopna vyregulovat soustavu, a tak dosáhnout požadované výšky (kmitá v okolí). Příčinnou může být, že tato metoda je vhodná pouze pro pomalejší soustavy. Druhá metoda, Step Response (Příloha A1 – Graf4), dokáže vyregulovat soustavu na požadovanou výšku. Jelikož ustálení dosahuje až v 80 s (velmi malá derivační složka), je tato metoda vhodná pro soustavy kde by překmit soustavy mohl způsobit poškození zařízení. V PLC byly dále nalezeny pro tuto soustavu optimální parametry vycházející z metody Ziegler-Nichols, kde průběh chování soustavy je uveden v Příloha A1 – Graf 4. Tento návrh nemá výrazný překmit a ustaluje se na požadované hodnotě ve 4 s.

Na modelu byla ověřena funkčnost výpočetních návrhových metod, které mají téměř shodné chování v simulaci jako v reálné soustavě. Metoda optimalizace ITAE kritéria je pro tuto soustavu neoptimálnější, jelikož má menší překmit než metoda Ziegler-Nichols a ustaluje se na požadované výšce rychleji v čase 3 s.

V práci je dále možné ručně nastavovat parametry bloku PID, pomocí nichž lze reprezentovat chování jednotlivých složek PID regulátoru. Tato možnost je realizována z důvodu použití modelu pro výukové účely. Ukázka funkčnosti modelu je v podobě videa, které je vloženo do přílohy CD.

LITERATURA

- [1] MELSEC iQ-F Series. Mitsubishi Electric [online]. [cit. 2022-01-01]. Dostupné z: <https://www.mitsubishielectric.com/fa/products/cnt/plcf/items/index.html>
- [2] PLC procesor řada FX5U, výstup: Relé, Tranzistor Komunikace měniče, komunikační protokol MELSEC (rámy 3C/4C), připojení. RS [online]. [cit. 2022-01-01]. Dostupné z: <https://docs.rs-online.com/4ee4/0900766b813df617.pdf>
- [3] MELSEC iQ-F FX5-4AD-ADP. Mitsubishi Electric [online]. In: . s. 2 [cit. 2022-01-01]. Dostupné z: <https://www.mitsubishifa.co.th/files/dl/FX5-4AD-ADP%20Hardware.pdf>
- [4] MELSEC iQ-F FX5-4DA-ADP. Mitsubishi Electric [online]. In: . s. 2 [cit. 2022-01-01]. Dostupné z: https://www.konighartman.nl/UserFiles/Product/Datasheet/FX5-4DA-ADP_DATA.pdf
- [5] MELSEC iQ-F FX5 User's Manual. Mitsubishi Electric [online]. In: . s. 306 [cit. 2022-01-01]. Dostupné z: https://www.allied-automation.com/wp-content/uploads/2015/05/MITSUBISHI_manual_plc_fx5_application.pdf
- [6] Infračervený senzor vzdálenosti Sharp GP2Y0A41SK0F. Drátek Návod [online]. [cit. 2022-01-01]. Dostupné z: <https://dratek.cz/arduino/7723-infracerveny-senzor-vzdalenosti-4-30-cm-gp2y0a41sk0f.html>
- [7] SHARP GP2Y0A41SK0F [online]. In: . s. 9 [cit. 2022-05-19]. Dostupné z: <https://dratek.cz/docs/produkty/0/917/gp2y0a41sk0f.pdf>
- [8] Co je stupeň krytí IP?. Goled [online]. 2019 [cit. 2022-05-19]. Dostupné z: <https://www.goled.cz/blog/stupen-kryti-ip/>
- [9] GOT2000 Series HMI GT-2705 [online]. [cit. 2022-01-03]. Dostupné z: <https://tr.mitsubishielectric.com/fa/products/hmi/got/got2000/gt27/gt2705-vtbd.html>
- [10] What Is MATLAB? Cimss [online]. [cit. 2022-01-02]. Dostupné z: <https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>
- [11] Simulink tutorial [online]. In: . s. 18 [cit. 2022-01-02]. Dostupné z: <https://ewh.ieee.org/r1/ct/sps/PDF/MATLAB/chapter8.pdf>
- [12] System Identification Toolbox. MathWorks [online]. [cit. 2022-05-19]. Dostupné z: <https://www.mathworks.com/products/sysid.html>
- [13] MELSEC iQ-F series. Mitsubishi Electric [online]. [cit. 2022-01-01]. Dostupné z: https://www.mitsubishielectric.com/fa/products/cnt/plcf/pmerit/concept/gx_works3.html
- [14] Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic [online]. In: . s. 101 [cit. 2022-01-03]. Dostupné z: http://www.edumat.cz/texty/Programovani_IEC61131-3.pdf
- [15] GT Designer3 [online]. In: . s. 1152 [cit. 2022-05-19]. Dostupné z: <https://www.mitsubishifa.co.th/files/dl/sh080866engan.pdf>

- [16] BLDC motor outrunnerový 2200 kV A2212/6T. Eclipse RA [online]. In: . s. 1 [cit. 2022-01-02]. Dostupné z: <https://dratek.cz/docs/produkty/0/119/1496225881.pdf>
- [17] Řadič pro bezkartáčové BLDC motory 3,7V-14,8V 30A. LaskaKit [online]. [cit. 2022-05-19]. Dostupné z: <https://www.laskakit.cz/radic-pro-bezkartacove-blde-motory-3-7v-14-8v-30a/>
- [18] Zdroj 12V 8,3A 100W spínaný SANPU YS-100-12. Drátek [online]. [cit. 2022-01-02]. Dostupné z: https://dratek.cz/arduino/74699-zdroj-12v-8-3a-100w-spinany-sanpu-ys-100-12.html?gclid=Cj0KCQiAlMCOBhCZARIsANLid6aUd2gNUqDsks2DivWIsiOE2meHRzT07_6YJ5U-YMNGRJUM12S_Q0kaAjavEALw_wcB
- [19] Raspberry Pi Pico. RPishop [online]. [cit. 2022-01-01]. Dostupné z: <https://rpishop.cz/raspberry-pi/3352-raspberry-pi-pico-0617588405587.html>
- [20] Blaha, P., Vavřín, P.: Řízení a regulace I. Základy regulace lineárních systémů – spojité a diskrétní. Skriptum VUT. [cit. 2022-01-01].
- [21] Co znamená PID. Automa [online]. [cit. 2022-01-01]. Dostupné z: https://automa.cz/cz/casopis-clanky/co-znamená-pid-2003_03_28768_3811/
- [22] Regulátory a vlastnosti regulátorů [online]. In: . s. 14 [cit. 2022-01-02]. Dostupné z: <http://automatizace-issnp.wz.cz/Soubory3/Vlastnosti%20regulatoru.pdf>
- [23] GHANADZADEH GILANI, A, S AHMADIFAR a T TAKI. Experimental and modeling study of liquid phase equilibria for (water + phosphoric acid + sec-alcohols) systems. The Journal of chemical thermodynamics [online]. LONDON: Elsevier, 2019, 135, 305-315 [cit. 2022-05-19]. ISSN 0021-9614. Dostupné z: doi:10.1016/j.jct.2019.04.005
- [24] Pivoňka, P.: Číslicová řídicí technika. Skriptum VUT. [cit. 2022-05-19].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

PID	Proporcionálně integračně derivační
P	Proporcionální
I	Integrační
D	Derivační
PLC	Programovatelný logický automat
USB	Universal Serial Bus
SD	Secure Digital
ROM	Read-Only Memory
CPU	Central processing unit
LED	Light-Emitting Diode
PWM	Pulse Width Modulation
IP	Ingress protection
IEC	International Electrotechnical Commission
ST	Strukturovaný text
LD	Ladder diagram
FB	Funkční blok
OR	Nebo/Součet
AND	Zároveň/Součin
NOT	Negace
POU	Program organization unit
RGB	Red green blue
PC	Personal computer
BLDC	Bezkartáčový outunerový motroe
AD	Analogovědigitální
tzv.	Tak zvaně
MB	Megabyte

Symboly:

U	napětí	(V)
I	proud	(mA/A)
R	odpor	(Ω)
l	délka	(mm/m)

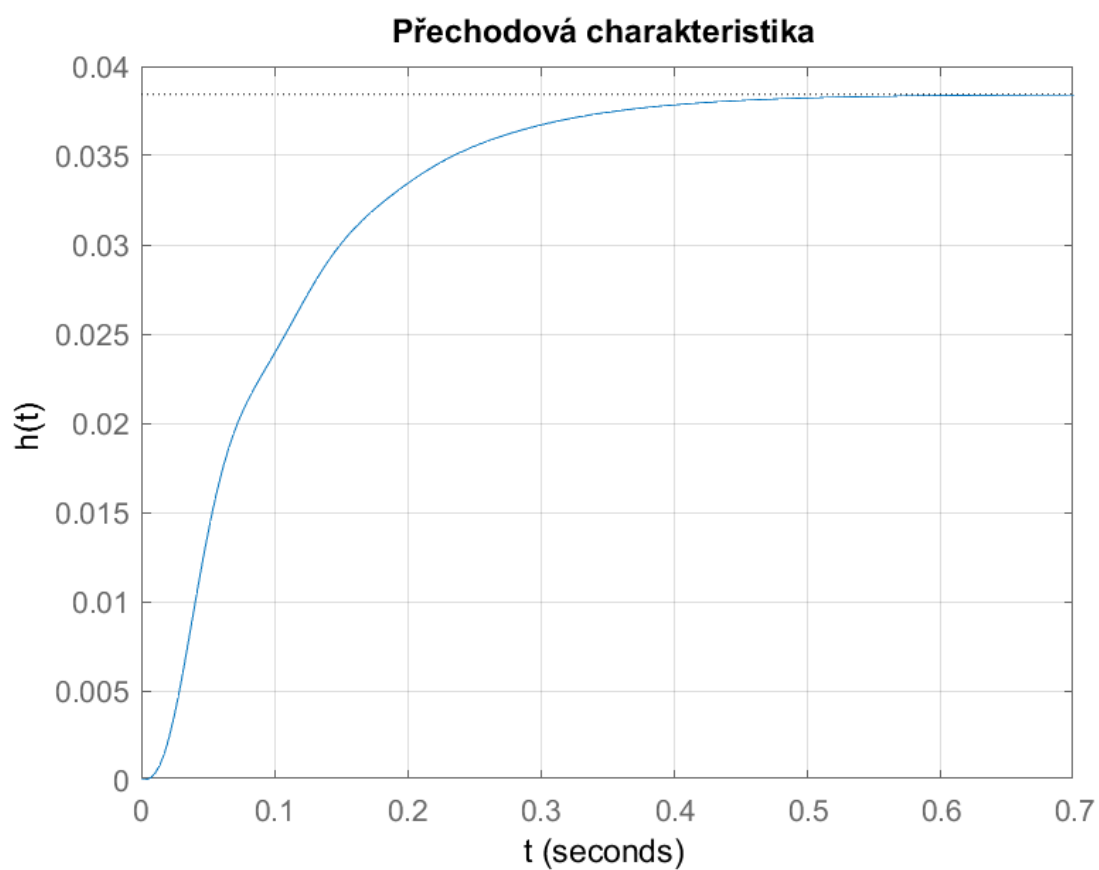
f	frekvence	(Hz)
t	čas	(ms/s)
P	výkon	(W)
$x(t)$	akční veličina	()
$y(t)$	výstupní veličina	()
$e(t)$	regulační odchylka	()
F_R	přenos regulátoru	()
F_s	přenos soustavy	()
s	operátorový přenos	()
K	zesílení	()
T_i	integrační konstanta	()
T_d	derivační konstanta	()
N	reálná složka	()
ξ	reálná složka	()

SEZNAM PŘÍLOH

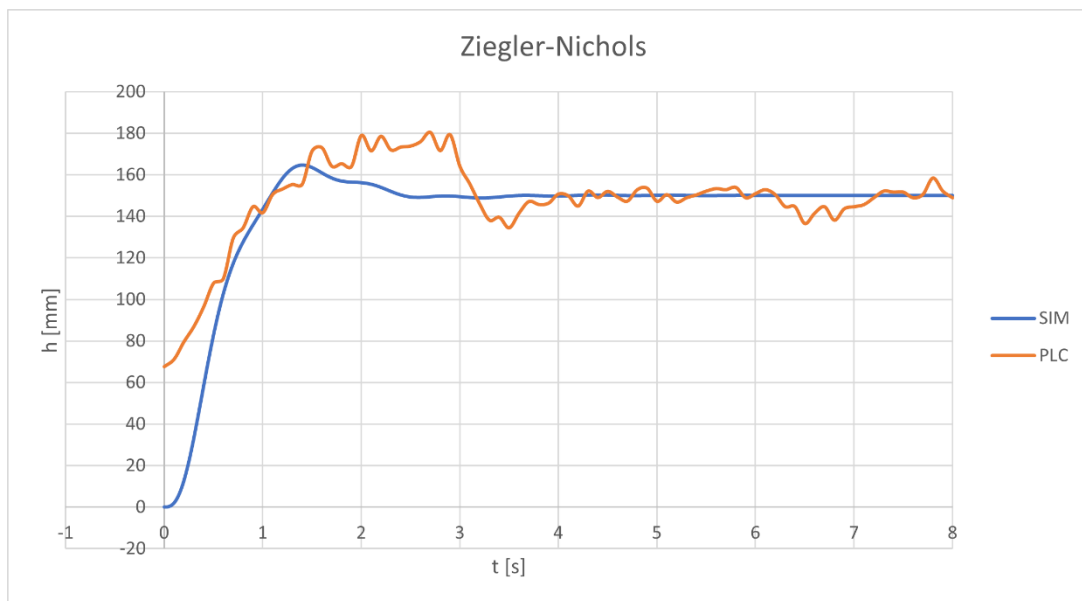
PŘÍLOHA A - ZÍSKANÉ HODNOTY	53
PŘÍLOHA B - ELEKTRONICKÁ PŘÍLOHA	56

Příloha A - Získané hodnoty

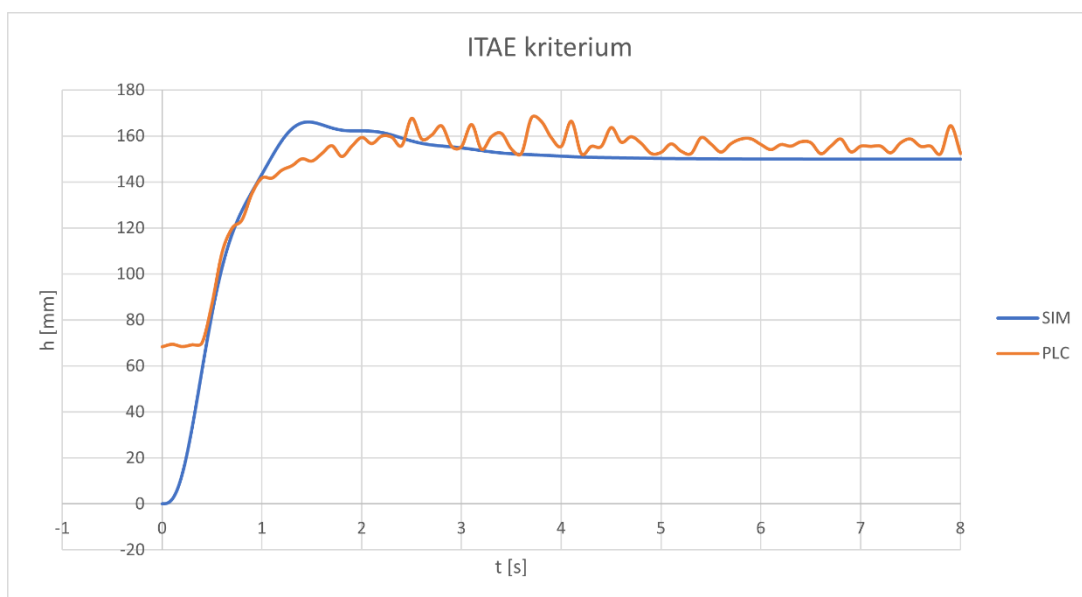
A.1 Naměřené průběhy



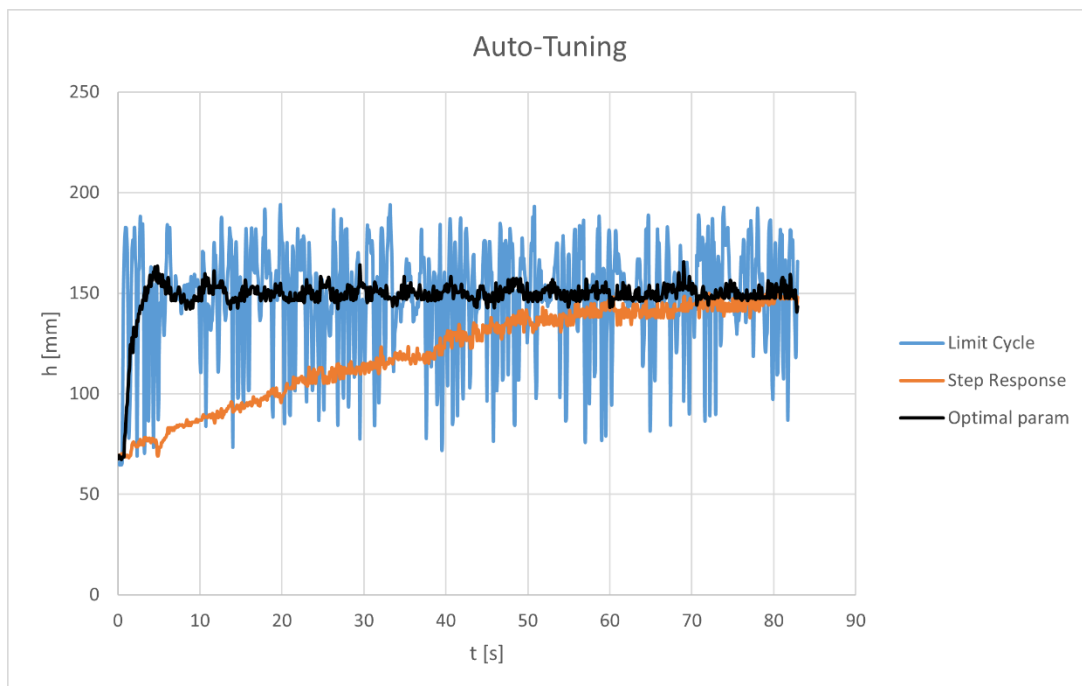
Graf 1 Přechodová charakteristika soustavy



Graf 2 Ziegler-Nichols – simulace/PLC

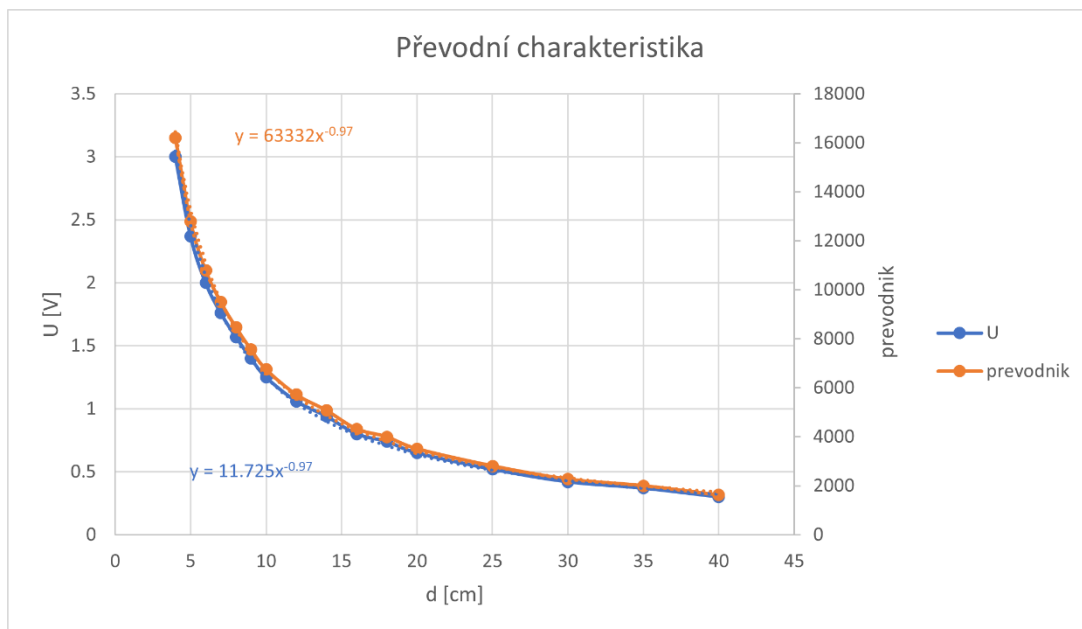


Graf 3 ITAE kritérium – simulace/PLC



Graf 4 Porovnání Auto-Tuningu s optimálními parametry

A.2 Převod snímače



Graf 5 Převodní charakteristika snímače (napětí na analogový rozsah karty)

Příloha B - Elektronická příloha

Obsah přílohy CD:

PID_Demo1.zip

- Bakalářská práce
 - **xkolac16.pdf**
- Záloha SW PLC
 - **PID_Demo1.nvw**
- Zdrojový kód PLC – **PID_Demo1_software**
 - ProgPou.pdf
 - ADConvScale.pdf
 - Init.pdf
 - Optimal_param.pdf
 - Z_Nmethod.pdf
 - ITAE.pdf
 - Autotuning.pdf
 - Global.xml
 - HMI.pdf
- MATLAB – **PSD**
 - data.mat
 - data_ident.m
 - ident.sid
 - log.xlsx
 - itae.m
 - ZN.m
 - soustava.m
 - PSD.slx
- Naměřená data – **Data**
 - Namerene_hodnoty.xlsx
 - Prevod_snimac.xlsx
- Schéma zapojení
 - zapojeni.drawio
- Video
 - **xkolac16_video.mp4**