

Laboratorní model čtyřválcové vodárny

Lukáš Menšík

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

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

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Lukáš Menšík
Osobní číslo: A22197
Studijní program: B0714A150006 Aplikovaná informatika v průmyslové automatizaci
Specializace: Inteligentní systémy s roboty
Forma studia: Prezenční
Téma práce: Laboratorní model čtyřválcové vodárny
Téma práce anglicky: Laboratory Model of a Four-Cylinder Water Tank

Zásady pro vypracování

- Zpracujte literární rešerši na problematiku zabývající se tvorbou laboratorních modelů a také matematických modelů.
- Navrhněte a zrealizujte laboratorní model čtyřválcové vodárny. Komunikační rozhraní pro monitorování a řízení laboratorního modelu bude umožňovat připojení k zadaným typům zařízení. Popište vytvořený laboratorní model včetně jeho I/O komunikačního rozhraní.
- Sestavte a popište matematický model pro vytvořený laboratorní model.
- Změřte statické i dynamické charakteristiky na vytvořeném laboratorním modelu. Proveďte identifikaci laboratorního modelu. Srovnajte výstupy získané na laboratorním modelu a na matematickém modelu. Vyhodnoťte získané výsledky.

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

Seznam doporučené literatury:

1. JOHANSSON, K. „H. The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable zero. IEEE Trans. on Control Systems Technology, 8, (3), 456-465, 2000.
2. NOSKIEVIČ, P. Modelování a identifikace systémů. Vyd. 1 Ostrava: Montanex, 1999, 276 s. ISBN 80-7225-030-2.
3. BOBÁL, Vladimír. Identifikace systémů. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009. ISBN 978-80-7318-888-7.
4. KADLČÍK, Ondřej. Řízení mnohorozměrového laboratorního modelu. Vedoucí Navrátil, Pavel. Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav automatizace a řídicí techniky, 2014. Dostupné také z: <http://hdl.handle.net/10563/30228>.
5. ZAKUCIA, J. Řízení čtyřválcové vodárny. Diplomová práce, ČVUT v Praze, 2009.

Vedoucí bakalářské práce: **Ing. Pavel Navrátil, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **13. prosince 2024**

Termín odevzdání bakalářské práce: **27. května 2025**

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



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

Ve Zlíně dne 13. prosince 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou 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;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské 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 bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....

podpis autora

ABSTRAKT

Cílem této práce je sestavení laboratorního modelu čtyřválcové vodárny, který slouží pro účely řízení. Laboratorní model umožňuje připojení vybraných typů zařízení, prostřednictvím nichž může uživatel laboratorní model řídit a realizovat na něm měření. V teoretické části jsou popsány rovnice z oblasti mechaniky tekutin, ze kterých se následně vychází při matematickém modelování laboratorního modelu v praktické části. V práci jsou rovněž popsány problémy, které nastaly při návrhu a realizaci konstrukce, a jejich řešení. Součástí práce je mimo jiné popis programového vybavení řídicí jednotky laboratorního modelu a způsobu předávání informací s vnějšími systémy.

Klíčová slova: Laboratorní model čtyřválcové vodárny, řídicí jednotka, vnější systémy, matematický model čtyřválcové vodárny, PLC, modul IC-1701

ABSTRACT

The aim of this work is to construct a laboratory model of a four-cylinder water plant for control purposes. The laboratory model allows the connection of selected types of devices, through which the user can control the laboratory model and perform measurements on it. The theoretical part describes equations from the field of fluid mechanics, which are then used for mathematical modelling of the laboratory model in the practical part. The thesis also describes problems, that encountered during the design and construction process and how they were solved. In addition, the work includes a description of the software of the control unit of the laboratory model and the method of data exchange with external systems.

Keywords: Laboratory model of a four-cylinder water plant, control unit, external systems, mathematical model of a four-cylinder water plant, PLC, module IC-1701

Je mou milou povinností poděkovat panu Ing. Pavlu Navrátilovi, Ph.D. za odborné vedení, cenné rady, čas a směřování správným směrem během realizace bakalářské práce. Také bych chtěl poděkovat své rodině a blízkým za podporu během celé délky studia.

OBSAH

ÚVOD.....	13
I TEORETICKÁ ČÁST	14
1 POJEM SYSTÉM.....	15
1.1 MATEMATICKÝ MODEL SYSTÉMU	15
1.1.1 Klasifikace dynamických systémů	16
1.2 POPIS VLASTNOSTÍ SYSTÉMŮ	16
1.2.1 Lineární diferenciální rovnice	17
1.2.2 Přenosová funkce systému	17
2 IDENTIFIKACE	19
2.1 ANALYTICKÝ ZPŮSOB IDENTIFIKACE	19
2.2 EXPERIMENTÁLNÍ ZPŮSOB IDENTIFIKACE	19
2.3 SYSTEM IDENTIFICATION APP	19
2.3.1 Popis okna System Identification App	20
3 MATEMATICKÉ MODELOVÁNÍ FYZIKÁLNÍCH SYSTÉMŮ	23
3.1 MATEMATICKÉ MODELOVÁNÍ MECHANICKÉHO SYSTÉMU	23
3.1.1 Hmotný bod.....	23
3.1.2 Pružina.....	24
3.1.3 Tlumič	24
3.1.4 Těleso pohybující se v pružném prostředí	25
3.2 MATEMATICKÉ MODELOVÁNÍ ELEKTRICKÉHO SYSTÉMU	25
3.2.1 Základní ideální prvky elektrického obvodu.....	26
3.2.2 Výpočet děliče napětí a určení obrazového přenosu	27
4 HYDRAULICKÉ SYSTÉMY	29
4.1 PROUDĚNÍ IDEÁLNÍCH TEKUTIN	29
4.1.1 Zákon zachování hmotnosti – rovnice kontinuity	29
4.1.2 Zákon zachování energie – Bernoulliho rovnice	30
4.2 ROZDĚLENÍ PROUDĚNÍ REÁLNÝCH TEKUTIN	31
4.3 VÝTOK KAPALINY Z NÁDOBY	32
4.3.1 Výtok z uzavřené nádrže s malým otvorem	33
4.3.2 Změna výšky hladiny v otevřené nádrži při volném výtoku.....	34
4.3.3 Změna výšky hladiny nádrže s přítokem	35
5 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY.....	37
5.1 HARDWARE PLC	37
5.1.1 Procesor (CPU)	37
5.1.2 Paměť	38
5.1.3 Vstupně výstupní rozhraní	38
5.1.4 Komunikační rozhraní.....	39

5.1.5	Napájecí zdroj.....	39
5.1.6	Speciální funkce	40
5.2	VYKONÁVÁNÍ PROGRAMU PLC	40
5.2.1	Obrazy vstupů a výstupů	41
6	PLATFORMA ARDUINO	42
6.1	ARDUINO UNO	42
6.2	ATMEGA328P	42
6.2.1	Registry	43
6.2.2	Paměť flash.....	44
6.2.3	Paměť EEPROM	44
6.2.4	Paměť SRAM	44
6.2.5	Aritmeticko-logická jednotka.....	45
6.2.6	Dekodér instrukcí	45
6.2.7	Watchdog časovač	45
6.2.8	Interrupt unit.....	46
6.3	DALŠÍ VYBRANÉ TYPY DESEK ARDUINO	47
6.3.1	Arduino Nano	47
6.3.2	Arduino Mega2560.....	47
II	PRAKTICKÁ ČÁST	49
7	MODEL ČTYŘVÁLCOVÉ VODÁRNY	50
7.1	MATEMATICKÝ MODEL ČTYŘVÁLCOVÉ VODÁRNY.....	51
7.2	LINEARIZACE MATEMATICKÉHO MODELU ČTYŘVÁLCOVÉ VODÁRNY POMOCÍ TAYLOROVA ROZVOJE PRVNÍHO ŘÁDU	52
7.3	UPRAVENÝ MATEMATICKÝ MODEL.....	56
7.4	DALŠÍ MOŽNOSTI ŘÍZENÍ LABORATORNÍHO MODELU	58
8	KONSTRUKCE MODELU ČTYŘVÁLCOVÉ VODÁRNY	61
8.1	PRVOTNÍ NÁVRH KONSTRUKCE.....	61
8.2	FINÁLNÍ NÁVRH KONSTRUKCE.....	62
8.3	REALIZACE KONSTRUKCE	63
9	ŘÍDICÍ JEDNOTKY MODELU.....	66
9.1	HLAVNÍ ŘÍDICÍ JEDNOTKA LABORATORNÍHO MODELU	66
9.2	PLC TECOMAT FOXTROT CP-1015	67
9.2.1	Modul OS-1401	67
9.2.2	Modul IC-1701	68
9.3	TECHNOLOGICKÁ KARTA PCI-1711U.....	69
10	ELEKTRONICKÉ ZAPOJENÍ MODELU	70
10.1	VYUŽITÉ ELEKTRONICKÉ SOUČÁSTKY	70
10.1.1	Ultrazvukové senzory vzdálenosti.....	71
10.1.2	Digitálně analogové převodníky	71

10.1.3	Moduly pro řízení čerpadel	73
10.1.4	Čerpadla	74
10.1.5	Step-down měniče napětí	75
10.2	PROPOJOVACÍ KRABICE	75
10.2.1	Krabice pro připojení PLC s příslušným modulem k řídicí jednotce.....	76
10.2.2	Krabice pro připojení mikropočítače a technologické karty PCI-1711.....	77
10.3	BLOKOVÉ SCHÉMA PROPOJENÍ JEDNOTLIVÝCH ČÁSTÍ	77
10.4	PROPOJOVACÍ KONEKTORY A ROZMÍSTĚNÍ JEDNOTLIVÝCH PINŮ	78
10.4.1	Připojení mikropočítače a technologické karty PCI-1711 k propojovací krabici.....	79
10.4.2	Připojení PLC s modulem IC-1701 nebo OS-1401 k propojovací krabici.....	79
11	PROGRAMOVÉ VYBAVENÍ LABORATORNÍHO MODELU	81
11.1	HLAVNÍ ČÁSTI PROGRAMU ŘÍDICÍ JEDNOTKY	81
11.1.1	Sejmutí vzdálenosti hladiny z ultrazvukového senzoru	81
11.1.2	Vysílání informace o vzdálenosti do vnějších systémů	84
11.1.3	Čtení příchozího PWM signálu o nízké frekvenci	86
11.1.4	Čtení příchozího PWM signálu o vyšší frekvenci.....	89
12	LINEARIZACE CHARAKTERISTIKY NAPĚTÍ NA ČERPADLE V ZÁVISLOSTI NA STRÍDĚ	91
12.1	PRÁCE S NAMĚŘENÝMI DATY V PROGRAMU MICROSOFT EXCEL	91
12.2	APLIKACE ROVNICE V KÓDU PRO VEDLEJŠÍ SYSTÉM	92
13	MĚŘENÍ STATICKÝCH A DYNAMICKÝCH CHARAKTERISTIK.....	94
13.1	KÓD PRO MĚŘENÍ STATICKÝCH A DYNAMICKÝCH CHARAKTERISTIK	94
13.2	NAMĚŘENÁ DATA A NASIMULOVANÁ DATA	95
13.2.1	Měření statických charakteristik	95
13.2.2	Měření dynamických charakteristik.....	97
13.2.3	Simulace a získání přenosové matice.....	99
13.2.4	Identifikace laboratorního modelu	103
13.3	ZHODNOCENÍ ZÍSKANÝCH VÝSLEDKŮ.....	105
	ZÁVĚR	106
	SEZNAM POUŽITÉ LITERATURY.....	107
	SEZNAM OBRÁZKŮ	109
	SEZNAM TABULEK.....	112
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	113
	SEZNAM PŘÍLOH.....	115

ÚVOD

Hydraulické systémy tvoří základ spousty zařízení moderní technické infrastruktury a jsou uplatňovány v širokém spektru průmyslových odvětví. Pro jejich efektivní návrh, analýzu a řízení je nutné vycházet z principů mechaniky tekutin, ovládat metody matematického modelování těchto systémů a umět vybrat vhodné měřicí zařízení.

Tato bakalářská práce se zabývá návrhem, konstrukcí a ověřením funkčnosti laboratorního modelu čtyřválnové vodárny. Tento laboratorní model je koncipován jako pomůcka pro výuku předmětů z oblasti řízení. Laboratorní model může umožnit studentům lépe pochopit aplikované matematické modelování systémů. Zároveň je na něm možno demonstrovat několik jevů a poznatků z teorie řízení, které tak mohou být názorně představeny.

Teoretická část shrnuje poznatky z oblasti teorie systémů, matematického modelování a mechaniky tekutin. Také představuje vybrané typy řídicích systémů. V praktické části se při matematickém modelování laboratorního modelu vychází právě z těchto poznatků.

Cílem práce je nejen samotná realizace modelu, ale také vytvoření matematického popisu systému, jeho linearizace, převod do přenosových funkcí a následné porovnání simulačních a experimentálních dat. Práce rovněž řeší problematiku převodu měřených hodnot do analogových signálů, které jsou pak dopravovány do vnějších jednotek a také popisuje řízení čerpadel vnějšími jednotkami.

Před samotným měřením a porovnáním výsledků matematického modelu se získanými daty z měření jsou podrobně rozepsány dílčí kroky, které vedly k zhotovení celého laboratorního modelu.

I TEORETICKÁ ČÁST

1 POJEM SYSTÉM

Pojem systém lze chápat jako soubor prvků, které jsou propojeny vzájemnými vztahy a jako celek mají také vztahy ke svému okolí. Systém je možno charakterizovat dvěma základními vlastnostmi:

- Chováním systému, čímž lze chápat, jak systém reaguje na podněty z jeho okolí, respektive jaké odezvy se objevují na výstupu systému za určitých podnětů okolí působících na jeho vstup.
- Strukturou systému, která popisuje vazby mezi jednotlivými prvky systému a rovněž popisuje chování těchto prvků. [1]

Zároveň lze pojmu systém rozumět jako množině

$$S = \{P, R, U, Y\} \quad (1.1)$$

kde P označuje prvky systému, R jsou relace mezi prvky systému, U je vstupní veličina systému a Y je výstupní veličina systému. Vstupní veličiny je možno dále dělit na akční (lze je ovlivnit), měřitelné a neměřitelné. Výstupní veličiny lze opět členit jako měřitelné a neměřitelné. [2]

Systém lze popsat matematickým modelem, který je velmi důležitý pro účely řízení. V matematickém modelu systému mohou být struktura i chování systému popsány matematickými a logickými výrazy, rovnicemi a algoritmy. [3]

1.1 Matematický model systému

Matematické modely systémů z pohledu řízení lze podle jejich charakteru rozdělit do dvou velkých skupin. Jsou jimi matematické modely statické a matematické modely dynamické.

U statických matematických modelů systémů je vazba mezi vstupními a výstupními veličinami reprezentována algebraickými rovnicemi. V těchto algebraických rovnicích není čas zahrnut jako nezávislá proměnná, vyskytuje se v nich tedy pouze relace mezi ustálenými hodnotami vstupů a výstupů.

U dynamických matematických modelů systémů je vazba mezi veličinami vstupními a výstupními vyjádřena diferenciálními, resp. diferenčními rovnicemi. [3]

Matematický model systému je důležitý, jelikož je na něm možno provádět experimenty. Místo toho, aby se experimentovalo se samotným systémem, bude

experimentováno s jeho matematickým modelem. Je tedy možno napodobit různé situace nebo vytvořit podmínky, které by mohly nastat u skutečného systému během pracovních podmínek. Tyto činnosti lze nazvat simulace. [1]

1.1.1 Klasifikace dynamických systémů

Dynamický systém je možno dále rozlišit na spojitý nebo nespojitý, lineární nebo nelineární, jednorozměrný nebo mnohorozměrný.

Jako lineární systém lze označit takový, který obsahuje pouze lineární členy a platí pro něj zákon superpozice, Jeho statickou charakteristikou je tedy přímka. Pokud je však alespoň jeden člen systému nelineární, v tom případě se už hovoří o nelineárních systémech, jejichž statická charakteristika je nelineární. Za nelineární systém je možno považovat většinu reálných systémů, avšak práce s nimi je mnohem obtížnější než se systémy lineárními. Snahou je tyto nelineární modely linearizovat. To se provede aproximací v okolí daného pracovního bodu lineárním modelem. [2]

Spojité a nespojité systémy jsou rozlišovány pomocí průběhu vstupních a výstupních veličin systému. U spojitých systémů jsou všechny signály v čase spojité a lze je nazývat také jako analogové systémy. U nespojitých systémů jsou všechny signály v čase nespojité a jsou nazývány také jako diskrétní systémy. [1]

Dělení systémů na jednorozměrné a mnohorozměrné záleží na počtu vstupních a výstupních veličin. U jednorozměrného systému se vyskytuje pouze jedna vstupní veličina, zatímco u mnohorozměrného systému se vyskytuje více vstupních a výstupních veličin. [2]

1.2 Popis vlastností systémů

Způsobů popisu matematického modelu dynamických vlastností lineárního systému existuje hned několik. Lze je rozdělit na dvě kategorie:

- Vnější popis systému, kterým lze vyjádřit dynamické vlastnosti vazeb mezi vstupem a výstupem systému. U vnějšího popisu není potřeba znát strukturu analyzovaného systému. Není tedy známo, co se doopravdy děje uvnitř systému a je sledována pouze reakce systému na vstupní signály.
- Vnitřní popis systému, pomocí něhož jsou popsány dynamické vlastnosti vztahů mezi vstupem, vnitřním vztahem a výstupem systému.

Pro vnější popis systému je taktéž možno použít statickou charakteristiku. Statická charakteristika je grafická závislost výstupní veličiny na vstupní veličině systému po konci přechodového děje, tedy v ustáleném stavu. [1]

V teorii řízení a systému se rozlišuje hned několik způsobů, jak vyjádřit vnější popis systému. Řadí se mezi ně například lineární diferenciální rovnice a přenosová funkce systému. [2]

1.2.1 Lineární diferenciální rovnice

Lineární diferenciální rovnicí je možno popsat chování lineárního spojitého dynamického systému, který obsahuje jednu vstupní a jednu výstupní veličinu

$$\begin{aligned} a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 y'(t) + a_0 y(t) \\ = b_m u^{(m)}(t) + \dots + b_0 u(t) \end{aligned} \quad (1.2)$$

kde a_i a b_j představují konstantní koeficienty, $y(t)$ je výstupní veličina systému a $u(t)$ je vstupní veličina systému.

Pro vyřešení diferenciální rovnice (2) je však nutno znát počáteční podmínky systému a zároveň průběh výstupní veličiny. Aby byl systém fyzikálně realizovatelný, musí platit nerovnost

$$m \leq n \quad (1.3)$$

která porovnává stupně nejvyšších derivací výstupní a vstupní veličiny. Celkový řád systému je určen řádem diferenciální rovnice výstupní veličiny. Pokud platí, že řád nejvyšší derivace výstupní veličiny je vyšší než řád nejvyšší derivace veličiny vstupní, hovoří se o striktně ryzích systémech. V případě, že se nejvyšší řády derivací rovnají, tak se mluví o nestriktně ryzím systému. [1][2]

1.2.2 Přenosová funkce systému

Přenosová funkce (přenos) systému se nejčastěji používá k popisu lineárního spojitého dynamického systému. Jde o poměr Laplaceova obrazu výstupní veličiny k Laplaceově obrazu vstupní veličiny při nulových počátečních podmínkách.

Aby byl získán tvar přenosu systému, je k tomu potřeba lineární diferenciální rovnice. Na lineární diferenciální rovnici, pro kterou platí výše uvedené podmínky, jsou aplikována pravidla Laplaceovy transformace. Po transformaci je získána

$$\begin{aligned} [a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s^1 + a_0 s^0] Y(s) \\ = [b_m s^m + \dots + b_0 s^0] U(s) \end{aligned} \quad (1.4)$$

z výše uvedeného lze přenos systému napsat ve tvaru

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} \quad (1.5)$$

přičemž musí být zároveň splněna podmínka fyzikální realizovatelnosti systému.[1][2]

2 IDENTIFIKACE

Identifikaci lze chápat jako proces, při kterém jsou o zkoumaném objektu získávány informace a poznatky. Pro identifikaci mohou být zvoleny dva přístupy, jimiž jsou analytický a experimentální způsob. [3]

2.1 Analytický způsob identifikace

Během analytického způsobu je matematický model sestavován pomocí využití matematicko-fyzikální analýzy daného objektu, při které se vychází například z konstrukčních, technologických nebo provozních údajů. Ve zkoumaném objektu mohou probíhat jevy, které lze popsat fyzikálními, chemickými a dalšími zákony. Tyto jevy jsou pak matematicky popisovány a tím pádem jsou získány vztahy mezi sledovanými veličinami, které určují matematický model zkoumaného objektu. Platí, že čím hlubší analýza je provedena, tím přesnější bude výsledný matematický model, ale zároveň se dosáhne obtížnějších výpočtů a vztahů. [3]

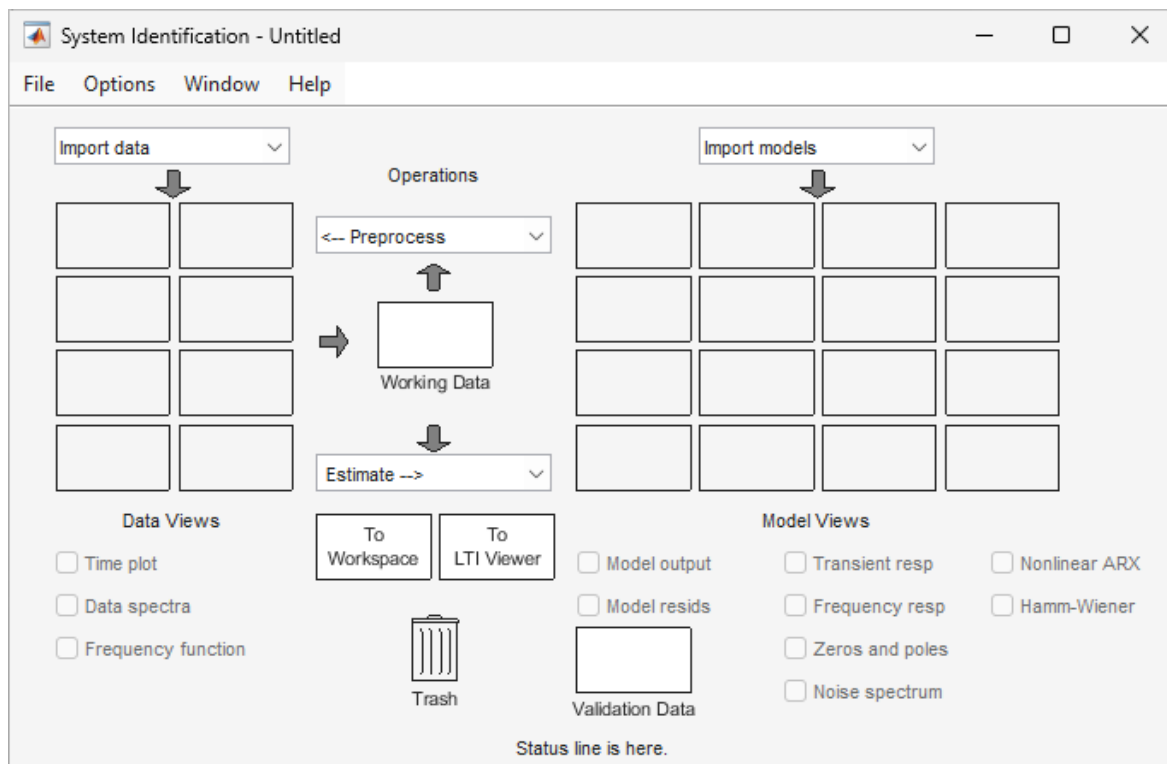
2.2 Experimentální způsob identifikace

Pro experimentální identifikaci je potřeba znát průběhy vstupních a výstupních veličin objektu, prostřednictvím kterých je poté získán matematický model, který představuje vnější popis systému. Možnou nevýhodou je, že matematický model určený experimentální identifikací neumožňuje zjistit vnitřní strukturu objektu.

Pro identifikaci reálných objektů může být použita vhodná kombinace obou z výše uvedených přístupů. Lze například postupovat tak, že model navržený analytickým způsobem je porovnán s reálným objektem pomocí dat získaných simulací matematického modelu a dat získaných experimentálně. [3]

2.3 System Identification App

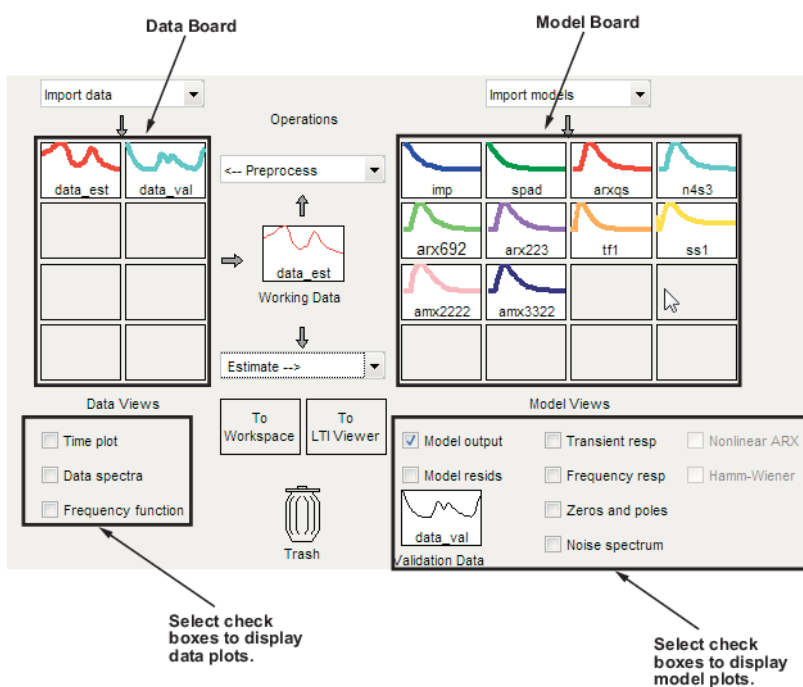
Program MATLAB poskytuje grafické uživatelské rozhraní pod názvem System Identification app, které umožňuje snadno identifikovat měřenou soustavu. Pro identifikaci měřené soustavy je však nutno disponovat naměřenými daty, konkrétně vstupně-výstupními. Po spuštění je uživateli nabídnuta možnost vložení dat, předzpracování a odhad. Pro spuštění se do příkazového okna programu MATLAB napíše příkaz *systemIdentification*. [4][5]



Obrázek 1: System Identification App [5]

2.3.1 Popis okna System Identification App

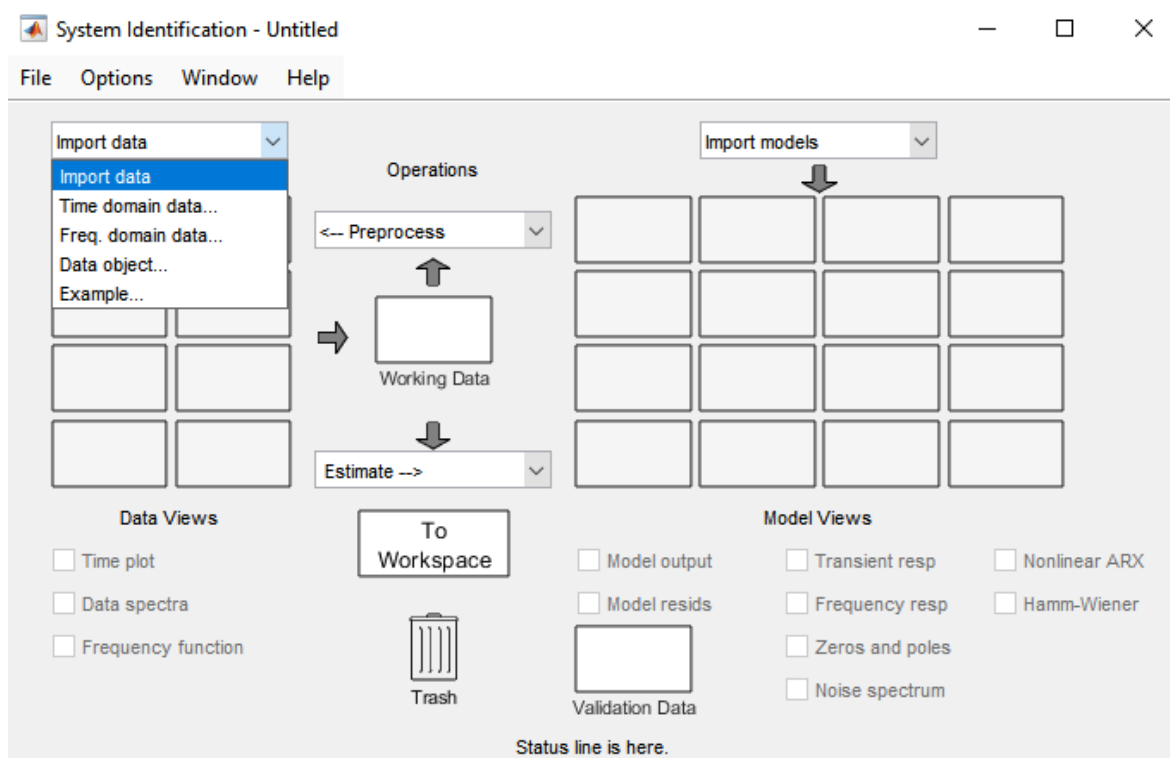
Následující obrázek ukazuje, kde se nachází různé možnosti pro práci se *System Identification App*.



Obrázek 2: Rozmístění jednotlivých funkcí v dialogovém okně System Identification App [5]

Rozvržení okna organizuje úlohy a informace zleva doprava, které zároveň odpovídá pracovnímu postupu. Práce se *System Identification App* začíná vložením pomocí nabídky *Import data* v levém horním rohu a končí vykreslením charakteristik odhadnutého modelu. [5]

Nabídka *Import data* poskytuje uživateli možnost vybrat si, jakým způsobem chce data vložit (viz Obrázek 3). Prvním možným způsobem, jak lze data vložit, je vložení dat do časové oblasti (*Time domain data...*), při kterém jsou ještě zadány vektorová či maticová vstupní a výstupní data, která byla naměřena, počáteční čas a perioda vzorkování. Následující způsob, jak je data možno vložit je vložení dat ve frekvenční oblasti (*Freq. Domain data...*). Zde mohou být vložena frekvenční data obdobným způsobem, jako data v časové oblasti. Dále se vyskytuje možnost vložení vektoru či matice komplexních hodnot s danou frekvencí, anebo vložení vektoru či matice amplitud a fází s danou frekvencí. Poslední nabízený způsob, jak lze data vložit, je datový formát (*Data object...*), pro který však musí být předem definován objekt za pomoci příkazů *IDDATA* či *IDFRD*. [4]



Obrázek 3: Možnosti vložení dat v dialogovém okně System Identification [5]

Dále si lze všimnout oblasti *Data Board*, umístěné hned pod nabídkou *Import Data*, která obsahuje obdélníkové ikony, reprezentující data, která byla naimportována.

Oblast *Model Board* rovněž obsahuje obdélníkové ikony, které byly uživatelem odhadnuty nebo naimportovány do aplikace. [5]

3 MATEMATICKÉ MODELOVÁNÍ FYZIKÁLNÍCH SYSTÉMŮ

3.1 Matematické modelování mechanického systému

Matematické modelování mechanických systémů, které se zpravidla skládají z hmotných těles a vzájemných vazeb mezi nimi, spočívá ve stanovení vztahů popisujících jejich pohyb a síly, které na ně působí. K tomu jsou využity zákony dynamiky. Při analýze dynamických vlastností zařízení bude matematický model nejčastěji popisovat soustavu dokonale tuhých těles, které nepodléhají deformacím a nemění tak svůj rozměr. Dále bývá také často předpokladem, že veškerá hmotnost a působící síly na těleso jsou soustředěny do jediného bodu, zvaného hmotný bod. [6]

3.1.1 Hmotný bod

Jak bylo zmíněno výše, hmotný bod má veškerou hmotnost tělesa umístěnou do jeho těžiště a lze pomocí něj vyjádřit setrvačné účinky tělesa. Pokud na hmotný bod působí vnější síla F , lze ji popsat pohybovou rovnicí ve tvaru

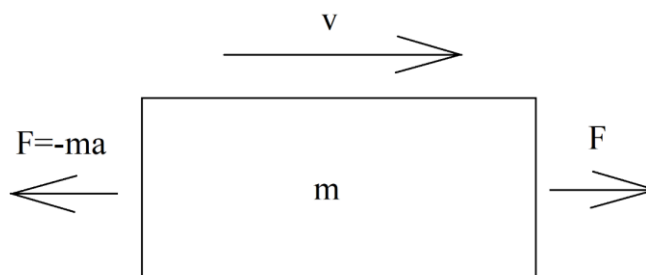
$$m \cdot a = F \quad (3.1)$$

kde m je hmotnost a a představuje zrychlení. Zrychlení je možno vyjádřit jako změnu rychlosti za jednotku času

$$m \cdot \frac{dv}{dt} = F \quad (3.2)$$

rychlost je dále možno vyjádřit jako derivaci dráhy podle času, tím pádem zrychlení je rovno druhé derivaci dráhy podle času [6]

$$m \cdot \frac{d^2x}{dt^2} = m \cdot x'' = F \quad (3.3)$$



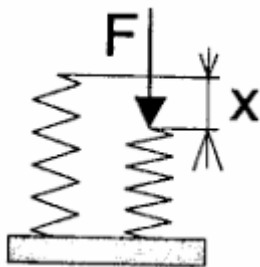
Obrázek 4: Hmotný bod [6]

3.1.2 Pružina

Síla F působící na pružinu, která je přenesena prostřednictvím pružiny je přímo úměrná její deformaci, tedy stlačení nebo prodloužení pružiny x

$$F = k \cdot x \quad (3.4)$$

kde k představuje tuhost pružiny. [6]



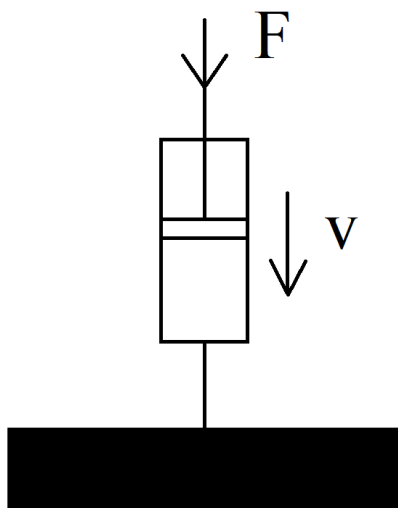
Obrázek 5: Pružina [6]

3.1.3 Tlumič

Tlumič je prvek, který pohybovou energii mechanické soustavy přeměňuje v jinou formu energie, zpravidla na tepelnou. Tlumič přenáší tlumicí sílu F , která je úměrná rychlosti

$$F = b \cdot v = b \cdot \frac{dx}{dt} = b \cdot x' \quad (3.5)$$

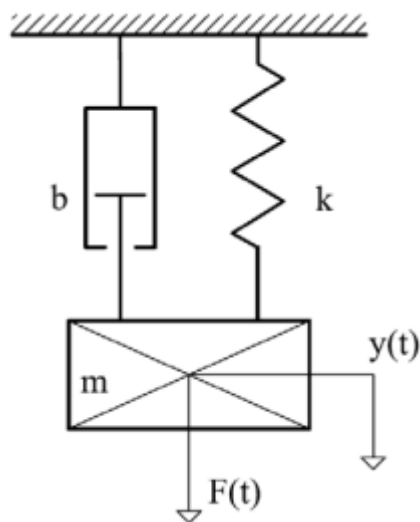
kde b představuje koeficient tlumení. [6]



Obrázek 6: Tlumič [6]

3.1.4 Těleso pohybující se v pružném prostředí

Pokud je vyšetřován pohyb tělesa o hmotnosti m , které je upevněno k tuhému rámu a na které v čase $t=0$ začne působit síla F , je vhodné těleso zjednodušit na hmotný bod. Upevnění lze pak dále nahradit pružinou s danou konstantou tuhosti k a tlumičem s koeficientem tlumení b .



Obrázek 7: Těleso připevněné k tuhému rámu [7]

Na těleso působí síly, které musí být v rovnováze. Pomocí sil jednotlivých prvků, které byly popsány výše, lze tuto mechanickou soustavu popsat rovnováhou sil ve tvaru [7]

$$mx''(t) + bx'(t) + kx(t) - F(t) = 0 \quad (3.6)$$

3.2 Matematické modelování elektrického systému

Elektrické systémy sestávají z jednotlivých komponent, které jsou mezi sebou propojeny tak, aby mezi nimi byl umožněn průchod elektrického proudu. V oblasti automatizace je analýza elektrických obvodů stěžejní, zejména při modelování elektrických pohonů využívajících elektromechanické akční členy.

Při studiu elektrických systémů jsou využívány elektrické obvody, které jsou znázorněny pomocí ideálních aktivních a pasivních prvků. Ideální prvky elektrického obvodu mají pouze jednu vlastnost-odpor, kapacitu nebo indukčnost. Aktivní prvky poskytují obvodu energii. Může se jednat o napěťové či proudové zdroje. Pasivní prvky spotřebovávají elektrickou energii. Řadí se mezi ně například rezistory s odporem R , kondenzátory s kapacitou C a induktory s indukčností L .

Mezi sledované veličiny při analýze elektrických obvodů patří elektrické napětí $u(t)$ a elektrický proud $i(t)$. [6]

Pro analýzu elektrických obvodů jsou dále využívány zákony, které mohou usnadnit určení napětí a proudu tekoucí různými částmi obvodu. Mezi základní zákony patří Ohmův zákon, První Kirchhoffův zákon a Druhý Kirchhoffův zákon.

Ohmův zákon říká, že proud I v uzavřeném obvodu je roven podílu elektrického napětí U a odporu R .

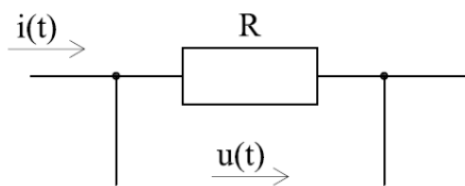
První Kirchhoffův zákon popisující zákon zachování elektrické náboje říká, že v každém uzlu elektrického obvodu je součet proudů vstupujících do uzlu je roven součtu proudů vystupujících z uzlu.

Druhý Kirchhoffův zákon vyjadřuje zákon zachování energie v elektrických obvodech. Stanovuje, že v každé uzavřené smyčce elektrického obvodu je součet všech napětíových úbytků na spotřebičích roven součtu elektromotorických napětí zdrojů, vyskytujících se v této smyčce. [7]

3.2.1 Základní ideální prvky elektrického obvodu

Odpor (rezistor) mění elektrickou energii v teplo. Podle Ohmova zákona je napětí $u(t)$ vyskytující se na odporu R přímo úměrné proudu $i(t)$, z něj je dále vzorec vyjádřen v Laplaceově oblasti

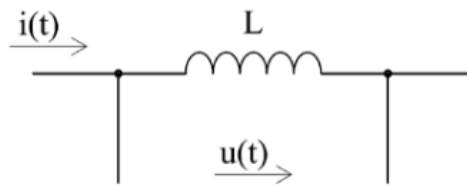
$$u(t) = i(t) \cdot R, \quad U(s) = I(s) \cdot R \quad (3.7)$$



Obrázek 8: Zapojení odporu [7]

Cívka (induktor) akumuluje energii magnetického pole. Okamžité napětí na induktoru je pomocí protékajícího proudu $i(t)$ a indukčnosti L vyjádřeno vztahem

$$u(t) = L \frac{di(t)}{dt}, \quad U(s) = Ls \cdot I(s) \quad (3.8)$$



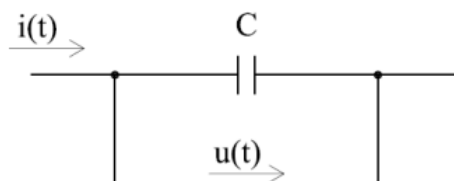
Obrázek 9: Zapojení cívky [7]

Kondenzátor (kapacitor) je schopen akumulovat energii elektrického pole. Proud protékající kondenzátorem je popsán vztahem

$$i(t) = C \frac{du(t)}{dt} \quad (3.9)$$

kde C označuje kapacitu. Z tohoto vzorce je pak dále možno vyjádřit napětí $u(t)$

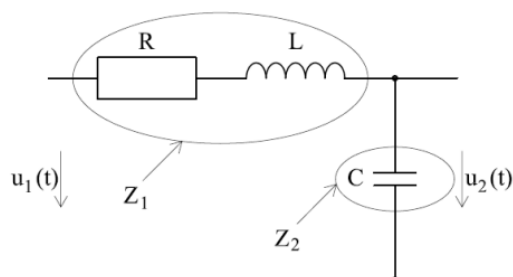
$$u(t) = \frac{1}{C} \int_0^t i(\tau) d\tau, \quad U(s) = \frac{1}{Cs} \cdot I(s) \quad (3.10)$$



Obrázek 10: Zapojení kondenzátoru [7]

3.2.2 Výpočet děliče napětí a určení obrazového přenosu

Uvažuje-li se napěťový dělič uvedený na obrázku níže



Obrázek 11: Schéma děliče napětí [7]

Jeho vstupní impedance Z_1 je vyjádřena vztahem

$$Z_1(s) = R + Ls \quad (3.11)$$

a výstupní impedance Z_2 je popsána vztahem

$$Z_2(s) = \frac{1}{Cs} \quad (3.12)$$

Pro určení obrazového přenosu je bráno v potaz, že vstupní a výstupní napětí je úměrné impedancím v obvodu. [7] Lze tedy napsat

$$G(s) = \frac{U_2(s)}{U_1(s)} = \frac{Z_2(s)}{Z_1(s) + Z_2(s)} = \frac{\frac{1}{Cs}}{R + Ls + \frac{1}{Cs}} \quad (3.13)$$

4 HYDRAULICKÉ SYSTÉMY

Hydraulické systémy lze popsat jako systémy, u nichž dochází k pohybu kapalin a ve kterých jsou kapaliny pod tlakem schopny konat práci. Při jejich modelování je cílem matematicky popsat tyto vlastnosti. S požadavky pro sestavení matematických modelů hydraulických systémů je možno se setkat například při řízení zásoby kapaliny v nádržích, řízení toku kapalin a při řízení pohybů mechanismů využívajících hydraulické pohony.

Tvorba rovnic rovnováhy za klidu a pohybu kapalin vychází z obecně platných vět z mechaniky podmínek rovnováhy sil a momentů, zákonů zachování hmotnosti a energie.[6]

4.1 Proudění ideálních tekutin

4.1.1 Zákon zachování hmotnosti – rovnice kontinuity

Pro proudící kapaliny je podmínkou splnění fyzikálního zákona o zachování hmotnosti. Ten říká, že pro kontrolní objem, kterým kapalina protéká, musí být hmotnost kapaliny neměnná. Hmotnost kapaliny však může být měněna dvěma způsoby. Prvním ze způsobů je lokální změna hmotnosti dm_t , která je zapříčiněna stlačitelností kapaliny. Druhým je pak konvektivní změna hmotnosti dm_s , která je dána rozdílem v hmotnosti kapaliny, jež do kontrolního objemu přiteče a následně odeče. Ve výsledku se tyto dvě změny musí vyrovnat, aby byla zachována hmotnost

$$dm = dm_s + dm_t \quad (4.1)$$

Při jednorozměrném ustáleném proudění, u kterého platí $\frac{\partial \rho}{\partial t} = 0$, se dále bere v potaz, že hustota, průřez a rychlost jsou pouze funkcí souřadnice s . Rovnici kontinuity lze tedy zapsat ve tvaru

$$\frac{d}{ds}(\rho, S, v) = 0 \quad (4.2)$$

Po integrování dle souřadnice je získán tvar

$$Q_m = \rho S v = konst \quad (4.3)$$

kde Q_m je označení pro hmotnostní tok.

V proudové trubici musí v každém průřezu platit rovnost

$$\rho_1 S_1 v_1 = \rho_2 S_2 v_2 = \rho S v = konst \quad (4.4)$$

kterou lze dále zjednodušit, jelikož pro nestlačitelné kapaliny platí, že hustota je konstantní. Získává se rovnice ve tvaru

$$Q = S v = konst \quad (4.5)$$

kde Q je průtok, jenž určuje objem kapaliny, který protekl za jednotku času. [6]

4.1.2 Zákon zachování energie – Bernoulliho rovnice

Proudění kapaliny, na kterou působí tíhové zrychlení, vyjadřuje zákon o zachování energie, který lze popsat Bernoulliho rovnicí, kde proudící kapalina je uvažována jako dokonalá.

Pro dokonalou kapalinu, kterou nelze stlačit, má za působení tíhového zrychlení Bernoulliho rovnice tvar

$$\frac{v^2}{2} + \frac{p}{\rho} + gh = konst. \quad (4.6)$$

V této rovnici, která reprezentuje zákon o zachování energie, představuje první člen měrnou kinetickou energii, druhý člen odpovídá měrné tlakové energii a třetí člen je roven polohové energii pro hmotnostní jednotku kapaliny. Pokud by byly sečteny všechny energie, získala by se tím celková mechanická energie kapaliny, která je podle Bernoulliho rovnice pro každý průřez jedné a té samé trubice konstantní. [6][8] Lze ale použít i jiný zápis, který lze získat vydělením výše uvedené rovnice tíhovým zrychlením

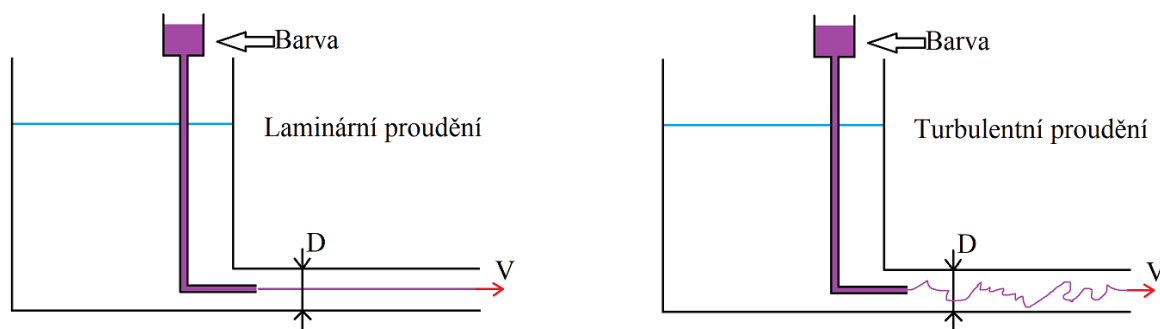
$$\frac{v^2}{2g} + \frac{p}{\rho g} + h = konst, \quad (4.7)$$

kde se první člen označuje jako rychlostní výška, druhý jako tíhová výška a třetí člen určuje polohovou výšku. Každý člen v rovnici reprezentuje energii, která je vztažena na tíhovou jednotku kapaliny. [6]

Bernoulliho rovnice může být aplikována pro výpočet neznámé veličiny pomocí dvou průtočných průřezů. V prvním průtočném průřezu jsou veličiny tlak p_1 , rychlost v_1 a výška h_1 známy, ve druhém se vyskytuje neznámá veličina, například tlak p_2 a ostatní veličiny jsou známy. Hledanou veličinu lze pak po úpravách vyjádřit a vyčíslit. [6]

4.2 Rozdělení proudění reálných tekutin

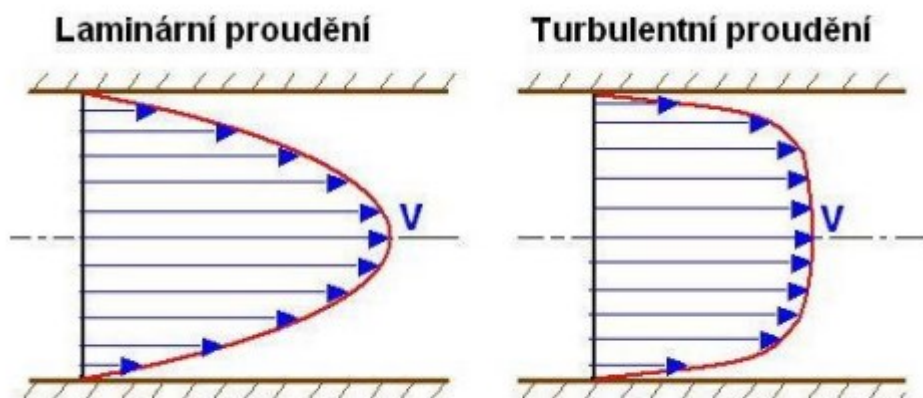
Existují dva druhy proudění reálné tekutiny. Jsou jimi laminární proudění a turbulentní proudění. Existenci těchto dvou druhů proudění potvrzuje Reynoldsův pokus, který je znázorněn na následujícím obrázku



Obrázek 12: Reynoldsův pokus [7]

Reynoldsův pokus spočívá v přivedení obarvené tekutiny tenkou trubičkou do tekutiny proudící kruhovým potrubím. Pokud je rychlost proudění malá, tak nedochází k porušení barevného vlákna, což svědčí o tom, že pohyb je realizován ve vrstvách a jednotlivé částice tekutiny nejsou promíchávány. [8]

Pokud však dochází ke zvětšení rychlosti nad kritickou hodnotu, částice tekutiny se začínají intenzivně promíchávat v důsledku jejich turbulentních pohybů ve všech směrech. Částice tekutiny tak přecházejí mezi jednotlivými vrstvami, přičemž je mezi částicemi vyměňována kinetická energie a dochází k vyrovnávání rychlosti částic v celém průřezu. Takové proudění bývá označováno jako turbulentní. Důsledkem přemísťování částic je změna hybnosti, která se projevuje jako brzdicí účinek. Proto je výsledný odpor proti pohybu větší než při laminárním proudění. [8]



Obrázek 13: Rozdíly rychlostních profilů v potrubí [8]

Jednotlivé druhy proudění nejsou od sebe odlišeny jen rychlostním profilem, ale také velikostí hydraulických ztrát. Při laminárním proudění hydraulický odpor roste úměrně s rychlostí, při turbulentním proudění je závislý na druhé mocnině rychlosti.

Laminární proudění však může přejít na turbulentní proudění. Tento přechod je spojován se změnami rychlostního profilu a závislosti odporu. Tyto změny jsou pro určité potrubí a tekutiny ovlivněny kritickou rychlostí. Experimenty i teorie podobnosti ukazují, že přechod laminárního proudění v turbulentní je charakterizováno Reynoldsovým číslem, které je definováno vztahem

$$Re = \frac{vd}{\nu} \quad (4.8)$$

kde v je střední rychlost tekutiny, d je charakteristický rozměr a ν je kinematická viskozita proudící tekutiny. V kruhovém potrubí je kritická hodnota Reynoldsova čísla stanovena na $Re = 2320 [-]$. [8]

4.3 Výtok kapaliny z nádoby

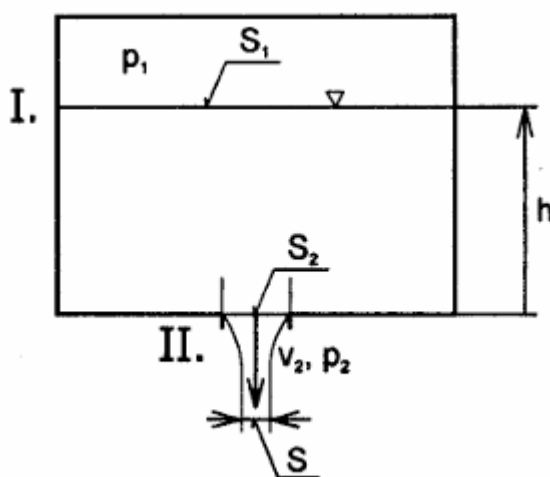
Problematikou výtoku kapaliny z nádoby je možno se zabývat například při regulaci výšky hladiny v nádržích. Výtokové otvory mohou být obecně umístěny na stěně nebo dně nádoby. Jedná-li se o otvor ve stěně nádoby, pak kapalina vytéká různou rychlostí, která je určena polohovou výškou uvažovaného místa. U otvorů malých rozměrů je možno nelineární průběh výtokové rychlosti zanedbat, u velkých otvorů je naopak nutností uvažovat polohovou závislost. V případě umístění otvoru ve dně nádoby je rychlost rozložena rovnoměrně. [6]

4.3.1 Výtok z uzavřené nádrže s malým otvorem

Zde je uvažován výtok kapaliny otvorem ve dně nádoby. Aplikací Bernoulliho rovnice na průřez hladiny a otvoru ve dně nádoby získáváme rovnici ve tvaru

$$\frac{p_1}{\rho} + \frac{v_1^2}{2} + gh = \frac{p_2}{\rho} + \frac{v_2^2}{2} + gh_z \quad (4.9)$$

kde h_z je ztrátová výška a v_2 je hledanou výtokovou rychlostí.



Obrázek 14: Výtok kapaliny z uzavřené nádrže s otvorem ve dně [6]

Ztrátová výška charakterizuje místní ztrátu ve výtokovém otvoru a je popsána vztahem

$$h_z = \xi \frac{v_2^2}{2g} \quad (4.10)$$

kde ξ je ztrátový součinitel.

Aplikací rovnice kontinuity na průřez hladiny a průřez výtokového otvoru je získána rovnice

$$S_1 v_1 = S_2 v_2 \quad (4.11)$$

pomocí které lze jednoduše vyjádřit rychlost v_1 jako

$$v_1 = \frac{S_2 v_2}{S_1} \quad (4.12)$$

Vyjádřenou rychlost v_1 nyní lze snadno dosadit do Bernoulliho rovnice uvedené výše. Dosazením rychlosti v_1 do Bernoulliho rovnice je zajištěno, že rovnice bude obsahovat pouze výtokovou rychlost v_2

$$\frac{p_1}{\rho} + \frac{1}{2} \frac{S_2^2}{S_1^2} v_2^2 + gh = \frac{p_2}{\rho} + \frac{v_2^2}{2} + g\xi \frac{v_2^2}{2g} \quad (4.13)$$

která je rovna

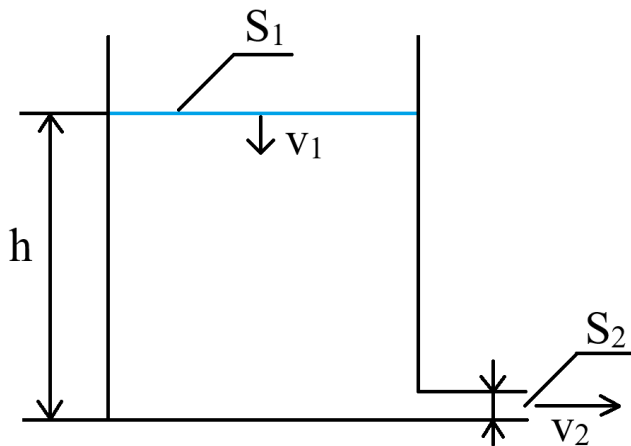
$$v_2 = \sqrt{\frac{2gh + \frac{p_1 - p_2}{\rho}}{1 + \xi - \frac{S_2^2}{S_1^2}}} \quad (4.14)$$

Pokud je uvažováno, že průřez výtokového otvoru je velmi malý v porovnání s průřezem nádrže, lze z rovnice vyloučit člen $\frac{S_2^2}{S_1^2}$. Jelikož je uvažována dokonalá kapalina, tak ztrátový součinitel $\xi = 0$. Dále pokud bude nádoba otevřená, pak tlaky vyskytující se nad hladinou a uvnitř nádoby si budou rovny $p_1 = p_2$. Při uvažování těchto okolností bude výtoková rychlost reprezentována Torricelliho vzorcem [6]

$$v_2 = \sqrt{2gh} \quad (4.15)$$

4.3.2 Změna výšky hladiny v otevřené nádrži při volném výtoku

Zde je změna výšky hladiny zapříčiněna volným výtokem z nádrže. V tomto případě jsou potřebné údaje pouze o obsahu průřezu nádrže a výtokového otvoru. Další potřebným údajem je výška hladiny v nádrži.



Obrázek 15: Volný výtok z otevřené nádrže [6]

Pro výtokovou rychlost v_2 je možno uvažovat Torricelliho vzorec vyjádřený z Bernoulliho rovnice, který je uveden výše

$$v_2 = \sqrt{2gh} \quad (4.16)$$

rychlost v_1 je vyjádřena jako

$$v_1 = -\frac{dh}{dt} \quad (4.17)$$

Vyjádřené rychlosti je možno dosadit do rovnice kontinuity ve tvaru

$$S_1 v_1 = S_2 v_2 \quad (4.18)$$

po dosazení je získán tvar

$$S_1 \left(-\frac{dh}{dt} \right) = S_2 \sqrt{2gh} \quad (4.19)$$

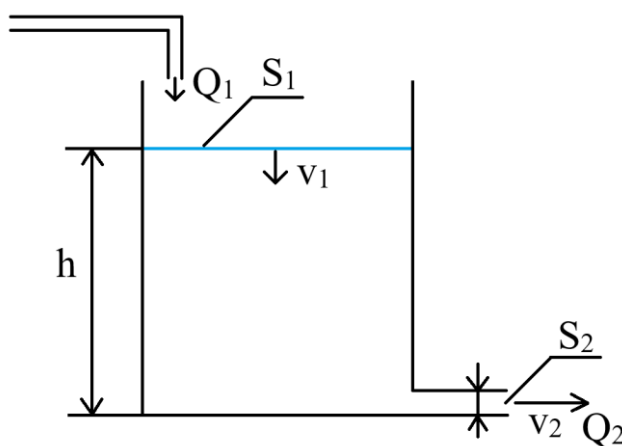
Výšku hladiny h je tedy možno při volném výtoku z otevřené nádrže popsat homogenní nelineární rovnicí prvního řádu ve tvaru:

$$\frac{dh}{dt} + \frac{S_2}{S_1} \sqrt{2gh} = 0 \quad (4.20)$$

Je však možno ještě zahrnout počáteční podmínku, která by představovala počáteční výšku hladiny v nádrži.

4.3.3 Změna výšky hladiny nádrže s přítokem

Nyní je uvažována nádrž s přítokem Q_1 a volným odtokem z nádrže Q_2 . V případě, že přítok Q_1 není roven odtoku Q_2 , pak bude docházet ke změně výšky hladiny uvnitř nádoby. Jestliže $Q_1 \geq Q_2$, dojde ke stoupání hladiny, naopak při $Q_1 \leq Q_2$ bude hladina klesat.



Obrázek 16: Výtok z otevřené nádrže s přítokem [6]

Pro změnu výšky hladiny bude tedy platit

$$S_1 \frac{dh}{dt} = Q_1 - Q_2 \quad (4.21)$$

po vyjádření volného odtoku z otevřené nádrže Q_2 je získán tvar

$$S_1 \frac{dh}{dt} = Q_1 - S_2 \sqrt{2gh} \quad (4.22)$$

který lze upravit na tvar

$$\frac{dh}{dt} = \frac{Q_1}{S_1} - \frac{S_2}{S_1} \sqrt{2gh} \quad (4.23)$$

V případě potřeby vyšetření ustáleného stavu výšky hladiny při konstantním přítoku Q_1 , by bylo z definice ustáleného stavu uvažováno $\frac{dh}{dt} = 0$. Platilo by tedy

$$0 = \frac{Q_1}{S_1} - \frac{S_2}{S_1} \sqrt{2gh} \quad (4.24)$$

Odkud lze po úpravách vyjádřit výšku, na které bude hladina ustálena

$$h_\infty = \frac{Q_1^2}{S_2^2 2g} \quad (4.25)$$

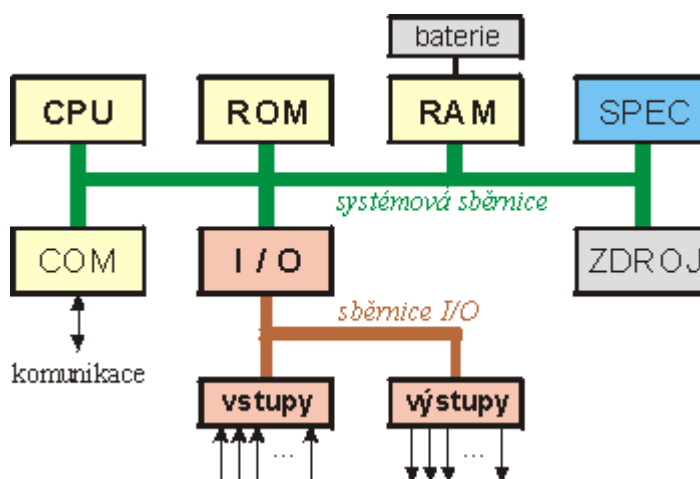
Hladina se po klesání či stoupání ustaluje díky dosažení rovnováhy $Q_1 = Q_2$. Systém tedy vykazuje vlastnost zvanou samoregulace. [6][8]

5 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

Programmable Logic Controller (PLC), česky programovatelný logický automat se skládá z hardwarové a softwarové části. Hardware PLC jsou fyzické součásti, které jako celek tvoří PLC systém. Naproti tomu software PLC se vztahuje k operačnímu systému PLC a aplikačním programům uloženým v jeho paměti. [9]

5.1 Hardware PLC

Hardware PLC sestává z několika fyzických součástí, které jako celek tvoří PLC systém. Každá hardwarová součást je navržena tak, aby vykonávala specifickou činnost. Bez přítomnosti těchto hardwarových komponent nemůže PLC systém spolehlivě pracovat [9]. Mezi základní hardwarové komponenty patří procesor (CPU), paměti, vstupně výstupní rozhraní, komunikační rozhraní, napájecí zdroj a speciální funkce.



Obrázek 17: Blokova struktura PLC [10]

5.1.1 Procesor (CPU)

Central Processing Unit (CPU), česky centrální výpočetní jednotka, lze považovat za nejdůležitější hardwarový komponent PLC. Na rozdíl od procesoru v klasických stolních počítačích, procesor PLC není navržen pro multitasking, ale spíše pro vykonávání specializovaných úloh, ke kterým je řazeno například skenování a provádění programu. [9] Mezi jeho základní činnosti například patří:

- Aktualizování stavu vstupů a výstupů. PLC tak zjistí stav na vstupních a výstupních svorkách. Umožňuje také změnu stavu na výstupních svorkách.
- Provádění všech logických a aritmetických operací.

- Komunikace s pamětí. Díky procesoru je PLC schopno přečíst nebo pozměnit obsah paměťových míst, kde jsou uloženy programy a data.[11]

V dnešní době jsou nejvíce používány PLC s jedním mikroprocesorem. Avšak některé PLC mohou mít dva mikroprocesory, které si mezi sebou rozdělují jednotlivé úlohy. Dva mikroprocesory je možno využít například tak, že jeden mikroprocesor by se staral o manipulaci s daty a prováděl složité výpočty, zatímco druhý by vykonával logické operace, časování, čítání a další funkce aplikačního programu. [9]

5.1.2 Paměť

Paměť stojí za uchováváním informací, programů a dat v PLC. Pro práci s pamětí jsou využívány dva základní procesy, jimiž jsou zápis a čtení. Zápis je proces, při kterém se nová informace zapíše do paměti. Při čtení jsou získány informace z paměti. V PLC je možno se setkat se dvěma druhy pamětí. Jsou nimi Read Only Memory (ROM) a Random Access Memory (RAM). [11]

Paměť RAM lze označit jako operační paměť, která umožňuje zápis i čtení. Slouží pro uložení řídicího programu, ale i hodnot vstupních, výstupních a pomocných proměnných.

Na druhou stranu paměť ROM slouží pouze pro čtení informací a obsahuje systémové programy, které jsou důležité pro činnost automatu. Uživateli tedy není dovoleno tyto programy modifikovat. [10]

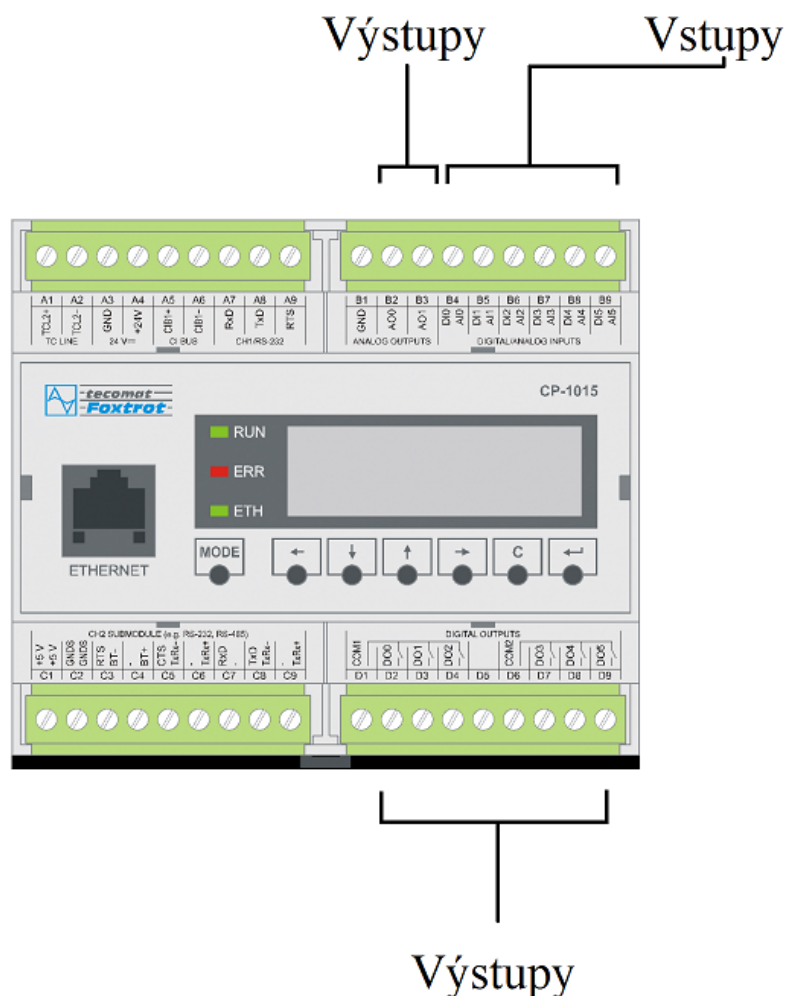
5.1.3 Vstupně výstupní rozhraní

PLC je používáno jako řídicí zařízení, které na základě informací na vstupních svorkách a logiky běžícího programu rozhoduje o stavech na výstupních svorkách.

Vstupní a výstupní signály lze odlišit podle analogového a digitálního charakteru. Hodnota analogového signálu se mění v čase v rozmezí daných hodnot, například u elektrického napětí (0-10) V nebo u elektrického proudu (4-20) mA, zatímco digitální signály mohou nabývat dvou hodnot, logické 0 a logické 1, pod nimiž si lze představit přechod mezi stavy zapnuto-vypnuto.

Za vstupní zařízení lze považovat například tlačítka, koncové spínače, teplotní senzory atd. Vstupně výstupní rozhraní zabezpečuje převod vstupních signálů na stejnosměrné napětí, které je akceptovatelné pro CPU.

Zařízení, která jsou řízená pomocí PLC jsou nazývána jako výstupní. Řadí se mezi ně například alarmy, relé, motory, ventilátory atd. [11][9]



Obrázek 18: Vstupy a výstupy Tecomat Foxtrot CP-1015 [12]

5.1.4 Komunikační rozhraní

PLC mohou komunikovat s mnoha zařízeními pomocí komunikačního rozhraní. Komunikační rozhraní se skládá z portů, které umožňují kabelové propojení s dalšími zařízeními jako PLC, dotykové obrazovky, počítačem a podobně. Mezi oblíbené prostředky pro komunikaci se řadí RS-232, RS-485, Modbus RTU a Ethernet. [9]

5.1.5 Napájecí zdroj

Hlavním účelem napájecího zdroje je přeměna dostupného napětí na takovou úroveň napětí, se kterou je schopen procesor a ostatní moduly PLC pracovat. Pokud je na vstupu zdroje střídavé napětí, tak zdroj jej přemění na napětí stejnosměrné, aby s ním mohly pracovat

ostatní komponenty PLC. Nejčastější hodnoty vstupního napětí do napájecího zdroje jsou 230V, 120V pro střídavé napětí a 24V pro stejnosměrné napětí.[9][11]

5.1.6 Speciální funkce

PLC mohou také nabízet výkonné instrukce pro realizaci komplexních operací, například pro realizaci regulátorů a jejich automatické seřízení, uložení dat do zásobníku a fuzzy logiku a podobně. [13] Speciální funkce mohou dále zahrnovat modul hodin, funkční prvky jako časovače, čítače, sekvenční registry, matematické funkce apod.

- Časovače slouží pro přesné řízení doby, po kterou trvá nějaký proces. Jejich čas běhu lze nastavit, lze je spouštět a zastavovat pomocí binárního signálu. Po uplynutí definovaného času mají na výstupu binární signál.
- Čítače počítají počet pulzů, které se objeví na jejich vstupu. Mohou také vysílat určitý počet pulsů, který je volitelný na jejich výstup.
- Sekvenční registry, též známy jako posuvné registry, je možno si představit jako posloupnost bitů, ve které lze každý bit adresovat a odkazovat se tak na něj. Po vložení nové informace na vstup sekvenčního registru dochází k umístění této informace do prvního bitu. Všechny ostatní bity jsou poté posunuty o jeden směrem dolů, kde končí registr. [10]

5.2 Vykonávání programu PLC

Program, který PLC vykonává, je možno si představit jako posloupnost instrukcí a příkazů. Pro PLC je charakteristické cyklické vykonávání programu v programové smyčce. Systémový program zajišťuje, že se programátor nemusí starat o vrácení vykonávání programu na začátek po jeho konci. Pokud program dlouho setrvává v programové smyčce, systém to zaznamená jako chybu a ohlásí překročení doby cyklu.

Při cyklickém vykonávání programu je vždy po dokončení poslední instrukce uživatelského programu předáno řízení systémovému programu, který zahajuje nový cyklus. V novém cyklu se nejdříve aktualizují hodnoty na vstupech a výstupech, doposud uložené hodnoty v paměti jako obrazy výstupů se přepíší do registrů výstupních periferních modulů. Hodnoty ze vstupních modulů jsou zkopírovány do paměťových obrazů vstupů. Také dochází k aktualizaci časových údajů potřebné pro časovače a systémové registry,

ošetření komunikace a provedení celé řady režijních úkonů. Po dokončení otočky cyklu je řízení znovu předáno první instrukci uživatelského programu. [13]

5.2.1 Obrazy vstupů a výstupů

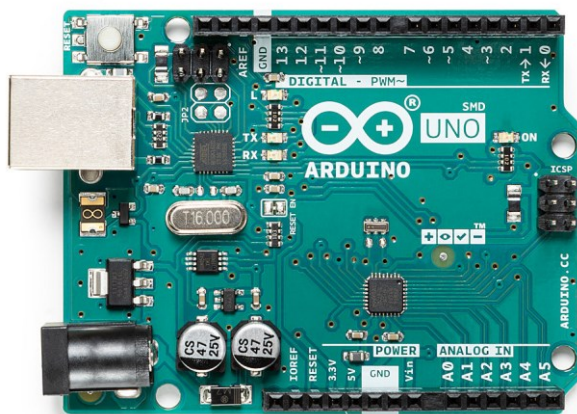
Program PLC nepracuje s aktuálními hodnotami na vstupech a výstupech, ale s jejich obrazy, které jsou uloženy v zápisníkové paměti. Přenos obrazů výstupů k řízenému objektu a načtení aktuálních vstupních hodnot od řízeného objektu je provedeno výhradně ve fázi otočky cyklu. Díky tomu je zajištěna synchronizace vstupních a výstupních dat s během programu, čímž se zamezí pravděpodobnosti vyskytnutí chyb, způsobených nevhodným souběhem měnicích se hodnot. Po dobu otočky cyklu jsou taktéž zmrazeny časové údaje a hodnoty většiny systémových proměnných.[13]

6 PLATFORMA ARDUINO

Arduino lze řadit mezi jednu z nejoblíbenějších platforem pro vývoj elektronických projektů, která je hojně využívána jak mezi začátečníky, tak i pokročilejšími uživateli. Umožňuje rychlé prototypování a ověřování různých konceptů, ale je vhodná i pro realizaci komplexnějších projektů. [14]

6.1 Arduino UNO

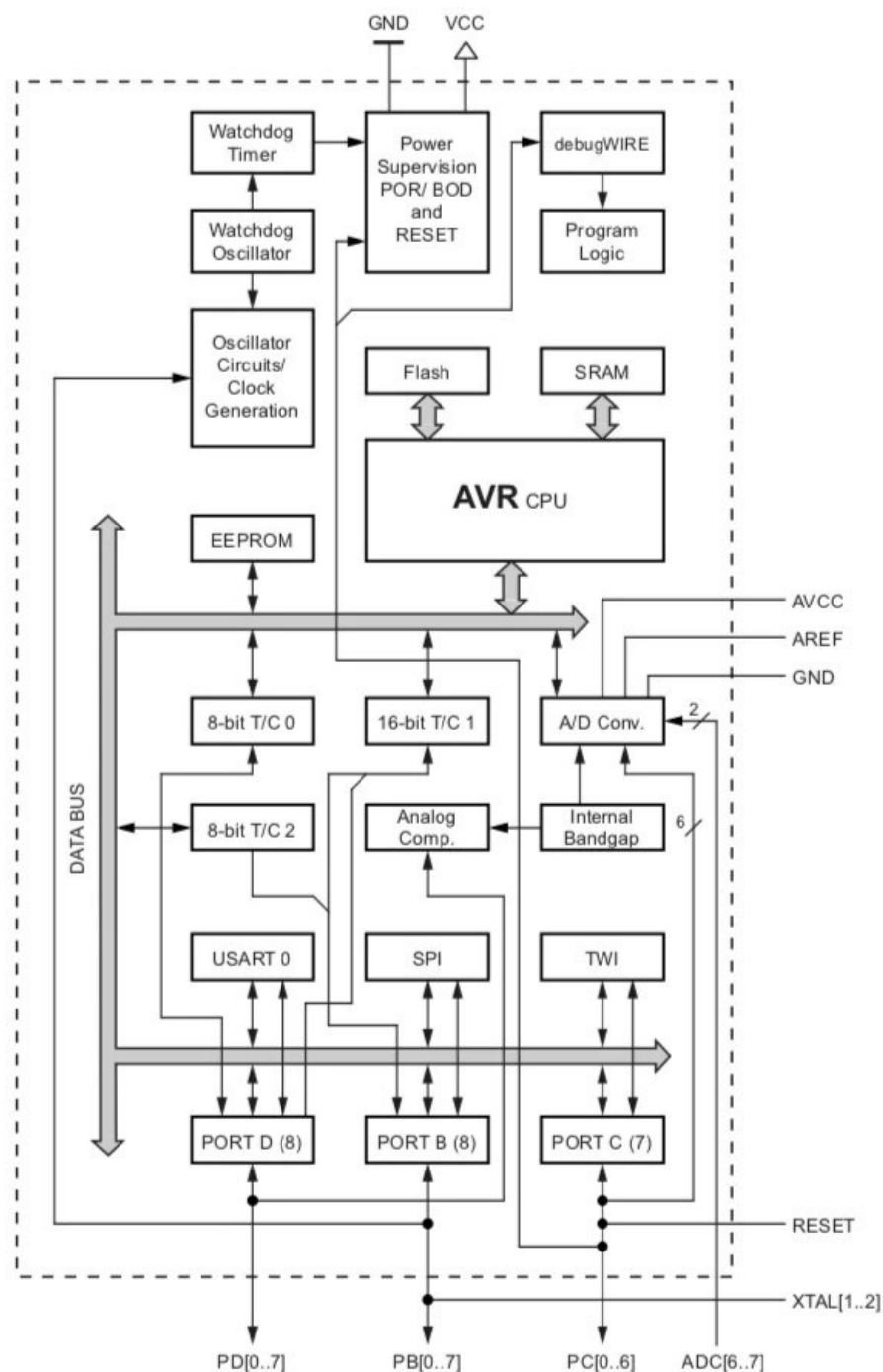
O běh programů na mikropočítači Arduino UNO se stará mikrokontrolér ATmega328. Aby bylo možné nahrát program do samotného čipu, je k tomu zapotřebí programátor, který je připojen k určeným pinům čipu. Pro usnadnění nahrávání programů je deska Arduino UNO vybavena USB převodníkem, jenž zajišťuje převod dat mezi počítačem a mikrokontrolérem ATmega328.[14]



Obrázek 19: Arduino Uno Rev3 [15]

6.2 ATmega328P

Čip ATmega328 obsahuje mimo procesor řady AVR i další komponenty, mezi které se řadí například paměti, časovače a podobně. Samotný procesor se stará o správný běh programu. Při provádění programu však může využívat i právě připojené komponenty.



Obrázek 20: Blokové schéma mikrokontroléru ATmega328 [14]

6.2.1 Registry

Registry jsou nejrychlejší a zároveň nejmenší paměťové jednotky, které je možno nalézt přímo na čipu. Jejich funkce je klíčová, jelikož obstarávají správné řízení celého čipu a jeho periférií. Je možno se setkat s následujícími registry:

- Status and control registry, které řídí čip a jeho periferie, zároveň z nich získávají zpětnou vazbu. Jedná se například o registry pro řízení časovačů, uspávání čipu a další.
- Vstupně-výstupní registry sloužící pro řízení vstupně výstupních operací. Prostřednictvím nich lze ovládat digitální piny.
- Program counter uchovávající adresu instrukce, jež má být provedena. Ukazuje do paměti Flash, kterou informuje o tom, jaká část programu je právě vykonávána. Má velikost 14 bitů.
- Instruction register obsahuje instrukci z paměti flash s adresou aktuálně uloženou v program counter registru.
- General purpose registers (GPR) jsou osmibitové registry, které může využívat jak programátor při vytváření programů, tak i kompilátor, když překládá program z C/C++ do assembleru. GPR registrů je 32 a jsou značené r0 až r31. Jelikož se jich nabízí jen omezený počet, je nutné s nimi šetřit, o což se snaží i kompilátor. Vyskytuje-li se v registru hodnota, se kterou se bude dále pracovat, může se v registru nechat, nebo se může uložit do paměti SRAM, ze které může být načtena v případě potřeby.[14]

6.2.2 Paměť flash

V této paměti zůstávají data i po odpojení napájení, jedná se tedy o non-volatilní paměť. Obsahuje také bootloader, což je speciální program, který je vykonáván po restartu desky. Bootloader na straně čipu čte kompilovaný program ze sériové linky a stará se o jeho uložení na správné místo do flash paměti.[14]

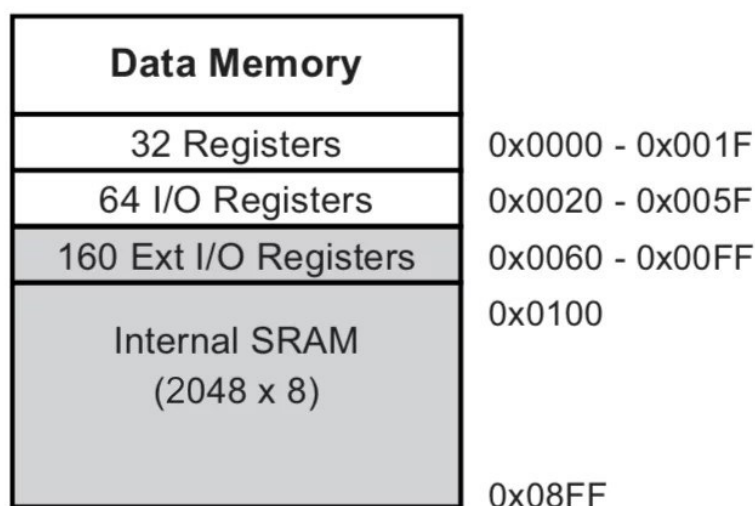
6.2.3 Paměť EEPROM

Jedná se o paměť, která je využívána pro zapisování hodnot v čase běhu programu. Uložené hodnoty v ní zůstávají i při přerušení napájení, jedná se tedy o druhou non-volatilní paměť na čipu. Slouží spíše jako pomocná paměť a mikrokontrolér ji pro svůj běh nezbytně nepotřebuje.[14]

6.2.4 Paměť SRAM

Tato paměť představuje hlavní uložistiště, které je využíváno programem během svého běhu. Vyskytují se na ní například uložené proměnné, které jsou dle potřeby kopírovány do

registrů a zpět, aby se umožnilo s nimi provádět výpočty. Velikost této paměti je 2 kB a její adresní rozsah je od 0x0100 do 0x08FF. Mikrokontrolér ATmega328 tuto paměť spojuje s registry do jednoho adresového prostoru s adresami 0x0000 až 0x08FF. Tyto adresy se tedy mohou jevit z pohledu programu jako jednotlivý prostor, i když je ve skutečnosti rozdělen na dvě části.



Obrázek 21: Adresový prostor registrů a paměti SRAM čipu ATmega328 [14]

Na nejnižších adresách v adresovém prostoru lze nalézt výše uvedené registry r0 až r31, hned za nimi se vyskytují registry pro řízení vstupně-výstupních operací. [14]

6.2.5 Aritmeticko-logická jednotka

Arithmetic logic unit (ALU) je zodpovědná za provádění matematických operací (sčítání, odčítání a další), bitových operací, výpočtů jednotkových a dvojkových komplementů. Jako vstupy pro ALU jsou využity GPR registry.

6.2.6 Dekodér instrukcí

Jedná se o hardware, který je zodpovědný za zapojení všech komponent potřebných pro provedení aktuální instrukce. Instruction registr obsahuje informace o instrukci, která má být provedena, a podle toho jsou aktivovány patřičné signály v procesoru.[14]

6.2.7 Watchdog časovač

Mikropočítače Arduino s mikrokontrolérem ATmega328 obsahují implementaci Watchdog časovače, který má vlastní 128kHz oscilátor. Jedná se o časovač, sloužící k hlídání správné funkce čipu. Sleduje běh programu a v případě, že doba běhu programu je příliš dlouhá a

nepřichází informace, že program není zaseknut, Watchdog časovač mikrokontrolér restartuje. Princip Watchdog časovače je založen na odpočítávání, které když dojde ke svému konci, dojde k automatickému restartu systému. Během běžného provozu je odpočet pravidelně resetován, což zabraňuje restartu systému. [14][16]

6.2.8 Interrupt unit

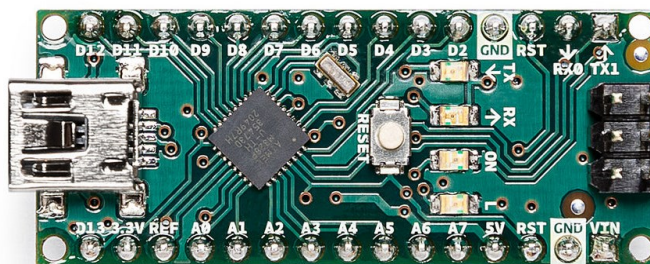
Jednotka přerušení (Interrupt unit) zpracovává signály, které jsou schopny přerušit procesor ve vykonávání aktuálního programu a požadovat po něm zpracování kódu zajišťující zpracování výjimky. Takové signály nazýváme přerušení a o jejich zpracování se stará funkce s názvem Interrupt Service Routine (ISR). Zdroje přerušení mohou být generovány hardwarem, ale i softwarem. Mezi jejich zdroje tedy řadíme například digitální piny, vnitřní časovače, nebo reset tlačítko. Mimo jiné může být přerušení kupříkladu využito k informování procesoru o konci převodu analogové hodnoty na hodnotu digitální. [14][16]

6.3 Další vybrané typy desek Arduino

Obvykle se u desky Arduino dá očekávat, že je v ní implementován procesor od firmy Atmel. Lze však narazit například na desku Arduino Intel Galileo, která disponuje procesorem od firmy Intel, konkrétně Quark SoC X1000, což je 32bitový procesor s pracovní frekvencí 400 MHz. Při vyhledávání daného typu desky si lze všimnout různých doplňků za názvem desky. Může to být například doplněk *Rev3* či *R3*. V tomto případě se jedná o číslo verze určité desky. V jednotlivých verzích desky se může lehce pozměnit umístění určitých komponent či design, k větším úpravám však nedochází. [17]

6.3.1 Arduino Nano

Jedná se o velmi malou verzi Arduino desky, která je právě díky svým malým rozměrům vhodná pro použití v projektech, které vyžadují co nejmenší rozměr řídicí jednotky. Arduino Nano disponuje procesorem ATmega328 s frekvencí 16 MHz. Ačkoli má tato deska malé rozměry, obsahuje USB port a převodník, tím pádem uživatel není nucen pořizovat externí USB-Serial převodník. [17]

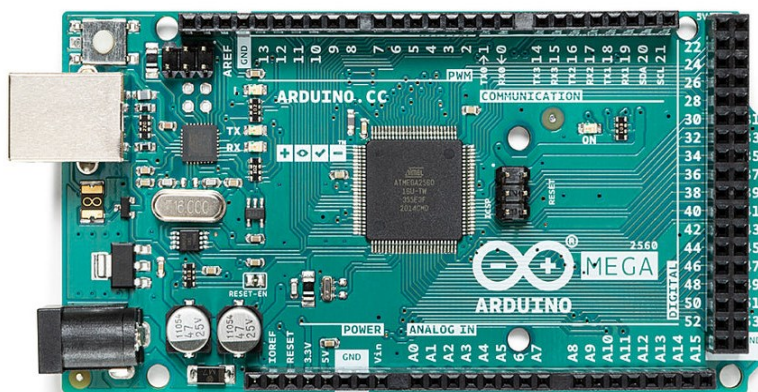


Obrázek 22: Arduino Nano [15]

6.3.2 Arduino Mega2560

Arduino Mega2560 vzniklo prodloužením desky Arduino Uno. Změna rozměrů umožnila integraci výkonnějších procesorů a přidání většího počtu zdírek. Při použití desky Arduino Mega2560 není uživatel tolik omezován počtem pinů a projekty tak mohou být výrazně

rozsáhlejší, než za použití desek Arduino Uno a Arduino Nano. [17]

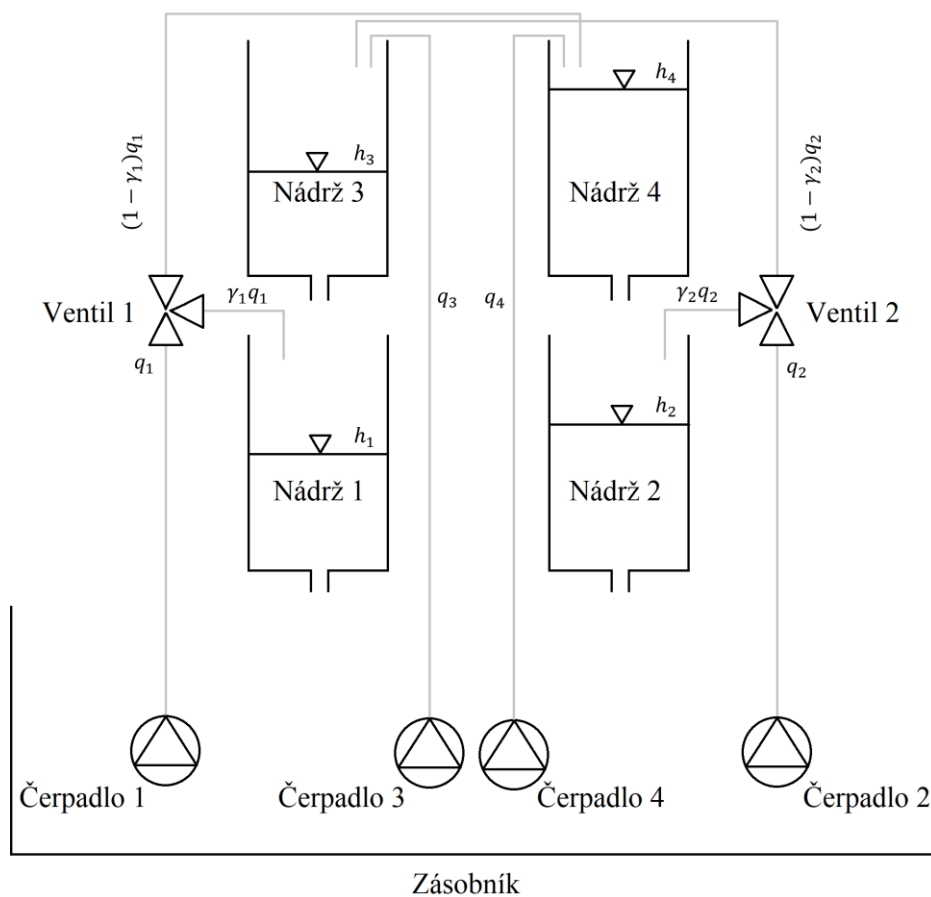


Obrázek 23: Arduino Mega2560 Rev3 [15]

II PRAKTICKÁ ČÁST

7 MODEL ČTYŘVÁLCOVÉ VODÁRNY

Model čtyřválcové vodárny se skládá ze čtyř nádrží uspořádaných ve dvou řadách a dvou sloupcích, třicestných ventilů, čtyř čerpadel a zásobníku. Nádrže mezi sebou vzájemně sdílejí kapalinu přivedenou přes určité čerpadlo. Čerpadlem 1 je tekutina dopravována skrz třicestný ventil do nádrže 1 a nádrže 4, ze které následně odtéká do nádrže 2 a poté zpět do zásobníku. Obdobné platí pro čerpadlo 2, které zajišťuje dopravu tekutiny do jiné kombinace nádrží. Nakonec čerpadlo 3 dopravuje kapalinu do nádrže 3 a čerpadlo 4 dopravuje kapalinu do nádrže 4 (viz Obrázek 24). Jedná se tak o laboratorní model, který slouží k demonstraci různých metod řízení a analýze systémů s více vstupy a výstupy (MIMO – Multiple Input Multiple Output).



Obrázek 24: Schéma čtyřválcové vodárny

Na obrázku uvedeném výše označení h_1 až h_4 představuje výšku hladiny kapaliny v jednotlivých nádržích. Dále parametry γ_1 a γ_2 označují nastavené poměry ventilů. Nastavení těchto ventilů může zásadně ovlivnit chování systému, zejména jestli bude

vykazovat minimálně fázové, či neminimálně fázové chování. Zjednodušeně řečeno, jde o to, zda se hladiny v nádržích po změně vstupních veličin ihned začnou pohybovat směrem k žádané hodnotě (minimální fázovost), nebo jestli se nejprve od žádané hodnoty budou oddalovat a až poté se k ní začnou přibližovat (neminimální fázovost).

Pro řízení systému jsou jako vstupní veličiny využity napětí přiváděné na čerpadle, přičemž výstupními veličinami jsou výšky hladin v jednotlivých nádržích, které jsou v tomto případě měřeny ultrazvukovými senzory, které poskytují bezkontaktní, poměrně přesné a rychle reagující měření výšky hladiny.

Matematický popis chování tohoto modelu vychází ze základních principů mechaniky tekutin. Přesná dynamika systému je pak popsána soustavou nelineárních diferenciálních rovnic, které je možno linearizovat, aby bylo dosaženo usnadnění analýzy systému a případného návrhu regulátorů.

7.1 Matematický model čtyřválcové vodárny

Při matematickém modelování čtyřválcové vodárny je uplatněn zákon zachování hmoty (rovnice kontinuity) a zákon zachování energie (Bernoulliho rovnice), které jsou popsány výše (viz 4.1.1 a 4.1.2). Matematický model popisuje změnu výšky vody $h_i(t)$ ve čtyřech nádržích jako soustavu diferenciálních rovnic:

$$\begin{aligned}\frac{dh_1(t)}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} + \frac{\gamma_1}{A_1}q_1(t) \\ \frac{dh_2(t)}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} + \frac{\gamma_2}{A_2}q_2(t) \\ \frac{dh_3(t)}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3(t)} + \frac{(1-\gamma_2)}{A_3}q_2(t) + \frac{1}{A_3}q_3(t) \\ \frac{dh_4(t)}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4(t)} + \frac{(1-\gamma_1)}{A_4}q_1(t) + \frac{1}{A_4}q_4(t)\end{aligned}\tag{7.1}$$

kde $h_i(t)$ označuje výšku hladiny v dané nádrži, A_i představuje plochu průřezu dané nádrže, a_i je plocha výstupního otvoru z dané nádrže, g je gravitační zrychlení, q_1 a q_2 jsou průtoky z hlavních čerpadel, q_3 a q_4 jsou průtoky vody z přídatných čerpadel, γ_1 a γ_2 představují poměrové koeficienty rozdělovací proudění vody mezi dolní a horní nádrže.

7.2 Linearizace matematického modelu čtyřválcové vodárny pomocí

Taylorova rozvoje prvního řádu

Pro účely řízení je vhodné pracovat s lineárním modelem systému, který lze získat linearizací nelineárních diferenciálních rovnic. Linearizaci je možno provést pomocí Taylorova rozvoje prvního řádu. Obecně lze změnu výšky hladiny v i -té nádrži vyjádřit jako funkci výšek hladin a průtoků

$$\begin{aligned} f_k(h_i, q_j)(t) &= \frac{dh_k(t)}{dt} = \dot{h}_k(t) \\ &\approx f_k(h_i^0, q_j^0) + \sum_{i=1}^4 \left. \frac{\partial f_k(h_i, q_j)(t)}{\partial h_i(t)} \right|_{h_i=h_i^0} \Delta h_i(t) \\ &\quad + \sum_{j=1}^4 \left. \frac{\partial f_k(h_i, q_j)(t)}{\partial q_j(t)} \right|_{q_j=q_j^0} \Delta q_j(t) \end{aligned} \quad (7.2)$$

kde indexy $i, j, k \in \{1, 2, 3, 4\}$, $\{h_i^0, q_j^0\}$ reprezentují stacionární pracovní body, $\Delta h_i(t) = h_i(t) - h_i^0$ je odchylka hladiny od pracovní hodnoty, $\Delta q_j(t) = q_j(t) - q_j^0$ reprezentuje odchylku průtoku od pracovní hodnoty. Pak linearizované rovnice mají tvar

$$\begin{aligned} \Delta \dot{h}_1(t) &= -\frac{a_1}{A_1} \sqrt{\frac{g}{2h_1^0}} \Delta h_1(t) + \frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}} \Delta h_3(t) + \frac{\gamma_1}{A_1} \Delta q_1(t) \\ \Delta \dot{h}_2(t) &= -\frac{a_2}{A_2} \sqrt{\frac{g}{2h_2^0}} \Delta h_2(t) + \frac{a_4}{A_2} \sqrt{\frac{g}{2h_4^0}} \Delta h_4(t) + \frac{\gamma_2}{A_2} \Delta q_2(t) \\ \Delta \dot{h}_3(t) &= -\frac{a_3}{A_3} \sqrt{\frac{g}{2h_3^0}} \Delta h_3(t) + \frac{(1-\gamma_2)}{A_3} \Delta q_2(t) + \frac{1}{A_3} \Delta q_3(t) \\ \Delta \dot{h}_4(t) &= -\frac{a_4}{A_4} \sqrt{\frac{g}{2h_4^0}} \Delta h_4(t) + \frac{(1-\gamma_1)}{A_4} \Delta q_1(t) + \frac{1}{A_4} \Delta q_4(t) \end{aligned} \quad (7.3)$$

Na základě linearizovaných rovnic uvedených výše lze sestavit stavový model pro získání stavových rovnic ve tvaru

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{v}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \quad (7.4)$$

kde $x(t)$ je vektor stavů $[\Delta h_1(t), \Delta h_2(t), \Delta h_3(t), \Delta h_4(t)]^T$ (přírůstky hladin v jednotlivých nádržích), $u(t)$ je vektor manipulovaných vstupů $[\Delta q_1(t), \Delta q_2(t)]^T$ (přírůstky průtoku hlavními čerpadly), $v(t)$ představuje vektor měřitelných poruch $[\Delta q_3(t), \Delta q_4(t)]^T$ (přírůstky průtoků přídatnými čerpadly), $y(t)$ je vektor výstupních veličin $[\Delta h_1(t), \Delta h_2(t)]^T$ (přírůstky hladin ve spodních nádržích). Matice A , B , E , C a D pak popisují různé vlastnosti systému.[18][19]

Matice A popisuje dynamiku systému (jaký vliv mají hladiny na sebe navzájem), matice B reprezentuje vliv hlavních čerpadel (čerpadlo 1 a čerpadlo 2), jejichž průtok je rozdělen ventily a matice E představuje poruchové vstupy do systému, které jsou zapříčiněny průtoky z přídatných čerpadel (čerpadlo 3 a čerpadlo 4).

Stavový popis je tedy ve tvaru

$$\begin{aligned}
 \begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \\ \Delta \dot{h}_3 \\ \Delta \dot{h}_4 \end{bmatrix} &= \begin{bmatrix} -\frac{a_1}{A_1} \sqrt{\frac{g}{2h_1^0}} & 0 & \frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & -\frac{a_2}{A_2} \sqrt{\frac{g}{2h_2^0}} & 0 & \frac{a_4}{A_2} \sqrt{\frac{g}{2h_4^0}} \\ 0 & 0 & -\frac{a_3}{A_3} \sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & 0 & 0 & -\frac{a_4}{A_4} \sqrt{\frac{g}{2h_4^0}} \end{bmatrix} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} \\
 &+ \begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{(1-\gamma_2)}{A_3} \\ \frac{(1-\gamma_1)}{A_4} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{A_3} & 0 \\ 0 & \frac{1}{A_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta q_3 \\ \Delta q_4 \end{bmatrix} \\
 \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot [\Delta h_1 \quad \Delta h_2 \quad \Delta h_3 \quad \Delta h_4]^T
 \end{aligned} \tag{7.5}$$

Pro zjednodušení zápisu stavového popisu byly zavedeny časové konstanty, které jsou definovány vztahem

$$T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_i^0}{g}} \quad (7.6)$$

kde $i \in \{1, 2, 3, 4\}$. Aby byla zachována rovnost, je zavedena převrácená hodnota časové konstanty. Pak platí

$$\frac{1}{T_i} = \frac{a_i}{A_i} \sqrt{\frac{g}{2h_i^0}} \quad (7.7)$$

Zjednodušený zápis stavového popisu má tedy tvar

$$\begin{aligned} \begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \\ \Delta \dot{h}_3 \\ \Delta \dot{h}_4 \end{bmatrix} &= \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1} \frac{1}{T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2} \frac{1}{T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} \\ &+ \begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{(1-\gamma_2)}{A_3} \\ \frac{(1-\gamma_1)}{A_4} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{A_3} & 0 \\ 0 & \frac{1}{A_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta q_3 \\ \Delta q_4 \end{bmatrix} \quad (7.8) \\ \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot [\Delta h_1 \quad \Delta h_2 \quad \Delta h_3 \quad \Delta h_4]^T \end{aligned}$$

Člen $\frac{A_3}{A_1} \frac{1}{T_3}$ v prvním řádku a třetím sloupci matice A vznikl vynásobením původního zápisu

$\frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}}$ jedničkou, která má tvar $\frac{A_3}{A_3}$. Pak tedy platí rovnost

$$\frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}} = \frac{A_3}{A_3} \frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}} = \frac{A_3}{A_1} \frac{1}{T_3} \quad (7.9)$$

Tento postup byl aplikován i na další členy, u kterých byly použity jiné, příslušné koeficienty.

Nyní lze získat matici přenosů $\mathbf{G}_S(s)$ a $\mathbf{G}_{SV}(s)$, kde symbol (s) představuje komplexní proměnnou. Předtím nejprve musí být na stavový popis aplikována Laplaceova transformace, platí

$$\begin{aligned} s\mathbf{X}(s) &= \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{E}\mathbf{V}(s) \\ \mathbf{Y}(s) &= \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \end{aligned} \quad (7.10)$$

Dále lze vyjádřit stavovou rovnici $\mathbf{X}(s)$ jako

$$\begin{aligned} (s\mathbf{I} - \mathbf{A})\mathbf{X}(s) &= \mathbf{B}\mathbf{U}(s) + \mathbf{E}\mathbf{V}(s) \\ \mathbf{X}(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}\mathbf{V}(s) \end{aligned} \quad (7.11)$$

Dosazením do výstupní rovnice $\mathbf{Y}(s)$ je následně získáno

$$\mathbf{Y}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}\mathbf{V}(s) + \mathbf{D}\mathbf{U}(s) \quad (7.12)$$

Matice $\mathbf{D}$ je nulová, takže výstupní rovnice je ve tvaru

$$\mathbf{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}\mathbf{V}(s) = \mathbf{G}_S(s) + \mathbf{G}_{SV}(s) \quad (7.13)$$

Následně byl použit program MATLAB k výpočtu maticových přenosů $\mathbf{G}_S(s)$ a $\mathbf{G}_{SV}(s)$, které byly vypočteny po naplnění jednotlivých matic $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ a $\mathbf{E}$. Tvar maticových přenosů byl určen pomocí příkazů

```
Gs=simplify(C*inv(s*eye(4)-A)*B+D);
Gsv=simplify(C*inv(s*eye(4)-A)*E+D)
```

ze kterých bylo získáno

$$\begin{aligned} \mathbf{G}_S(s) &= \begin{bmatrix} \frac{T_1\gamma_1}{A_1(T_1s+1)} & \frac{-T_1(\gamma_2-1)}{A_1(T_1s+1)(T_3s+1)} \\ \frac{-T_2(\gamma_1-1)}{A_2(T_2s+1)(T_4s+1)} & \frac{T_2\gamma_2}{A_2(T_2s+1)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{T_1\gamma_1}{A_1(T_1s+1)} & \frac{T_1(1-\gamma_2)}{A_1(T_1s+1)(T_3s+1)} \\ \frac{T_2(1-\gamma_1)}{A_2(T_2s+1)(T_4s+1)} & \frac{T_2\gamma_2}{A_2(T_2s+1)} \end{bmatrix} \end{aligned} \quad (7.14)$$

$$G_{SV}(s) = \begin{bmatrix} \frac{T_1}{A_1(T_1s + 1)(T_3s + 1)} & 0 \\ 0 & \frac{T_2}{A_2(T_2s + 1)(T_4s + 1)} \end{bmatrix}$$

Matice $G_S(s)$ popisuje dynamiku mezi vstupy a výstupy (jak vstupy Δq_1 a Δq_2 ovlivňují výstupy Δh_1 a Δh_2). Matice poruch $G_{SV}(s)$ pak popisuje vliv přídavných čerpadel na výstupy systému.

7.3 Upravený matematický model

Z pohledu řízení lze matematický model přepsat, jelikož uživatel zadává, kolik procent z rozsahu napětí by chtěl na čerpadla aplikovat. Lze tedy zavést

$$q_i = k_i \cdot v_i = k_i \cdot v \cdot p_i \quad (7.15)$$

kde index $i \in \{1, 2, 3, 4\}$, q_i označuje výsledný průtok daného čerpadla, k_i představuje zisk čerpadla [cm^3/Vs], v_i je napětí přivedené na čerpadlo [V], v je rozsah napětí a p_i jsou procenta zadaná uživatelem.[20]

Pak je matematický model ve tvaru

$$\begin{aligned} \frac{dh_1(t)}{dt} &= -\frac{a_1}{A_1} \sqrt{2gh_1(t)} + \frac{a_3}{A_1} \sqrt{2gh_3(t)} + \frac{\gamma_1 k_1}{A_1} v_1(t) \\ \frac{dh_2(t)}{dt} &= -\frac{a_2}{A_2} \sqrt{2gh_2(t)} + \frac{a_4}{A_2} \sqrt{2gh_4(t)} + \frac{\gamma_2 k_2}{A_2} v_2(t) \\ \frac{dh_3(t)}{dt} &= -\frac{a_3}{A_3} \sqrt{2gh_3(t)} + \frac{(1 - \gamma_2)k_2}{A_3} v_2(t) + \frac{k_3}{A_3} v_3(t) \\ \frac{dh_4(t)}{dt} &= -\frac{a_4}{A_4} \sqrt{2gh_4(t)} + \frac{(1 - \gamma_1)k_1}{A_4} v_1(t) + \frac{k_4}{A_4} v_4(t) \end{aligned} \quad (7.16)$$

Linearizované rovnice by změnilly tvar na

$$\frac{dh_i(t)}{dt} = -\frac{a_i}{A_i} \sqrt{2gh_i(t)} + \frac{a_j}{A_i} \sqrt{2gh_j(t)} + \frac{\gamma_j k_j}{A_i} v_j(t) + \frac{k_i}{A_i} v_i(t) \quad (7.17)$$

$$\Delta \dot{h}_3(t) = -\frac{a_3}{A_3} \sqrt{\frac{g}{2h_3^0}} \Delta h_3(t) + \frac{(1-\gamma_2)k_2}{A_3} \Delta v_2(t) + \frac{k_3}{A_3} \Delta v_3(t)$$

$$\Delta \dot{h}_4(t) = -\frac{a_4}{A_4} \sqrt{\frac{g}{2h_4^0}} \Delta h_4(t) + \frac{(1-\gamma_1)k_1}{A_4} \Delta v_1(t) + \frac{k_4}{A_4} \Delta v_4(t)$$

A stavový popis má tedy následující strukturu

$$\begin{aligned} \begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \\ \Delta \dot{h}_3 \\ \Delta \dot{h}_4 \end{bmatrix} &= \begin{bmatrix} -\frac{a_1}{A_1} \sqrt{\frac{g}{2h_1^0}} & 0 & \frac{a_3}{A_1} \sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & -\frac{a_2}{A_2} \sqrt{\frac{g}{2h_2^0}} & 0 & \frac{a_4}{A_2} \sqrt{\frac{g}{2h_4^0}} \\ 0 & 0 & -\frac{a_3}{A_3} \sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & 0 & 0 & -\frac{a_4}{A_4} \sqrt{\frac{g}{2h_4^0}} \end{bmatrix} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} \\ &+ \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{k_3}{A_3} & 0 \\ 0 & \frac{k_4}{A_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta v_3 \\ \Delta v_4 \end{bmatrix} \\ \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot [\Delta h_1 \quad \Delta h_2 \quad \Delta h_3 \quad \Delta h_4]^T \end{aligned} \quad (7.18)$$

Následně lze opět zavést časovou konstantu T_i , kde $i \in \{1, 2, 3, 4\}$. Tím je získán stavový popis ve zjednodušeném tvaru

$$\begin{aligned}
\begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \\ \Delta \dot{h}_3 \\ \Delta \dot{h}_4 \end{bmatrix} &= \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1} \frac{1}{T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2} \frac{1}{T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} \\
&+ \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{k_3}{A_3} & 0 \\ 0 & \frac{k_4}{A_4} \end{bmatrix} \cdot \begin{bmatrix} \Delta v_3 \\ \Delta v_4 \end{bmatrix} \quad (7.19) \\
\begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot [\Delta h_1 \quad \Delta h_2 \quad \Delta h_3 \quad \Delta h_4]^T
\end{aligned}$$

Ze stavového popisu v tomto tvaru lze vyjádřit maticové přenosy $\mathbf{G}_S(s)$ a $\mathbf{G}_{SV}(s)$. Jejich tvar je

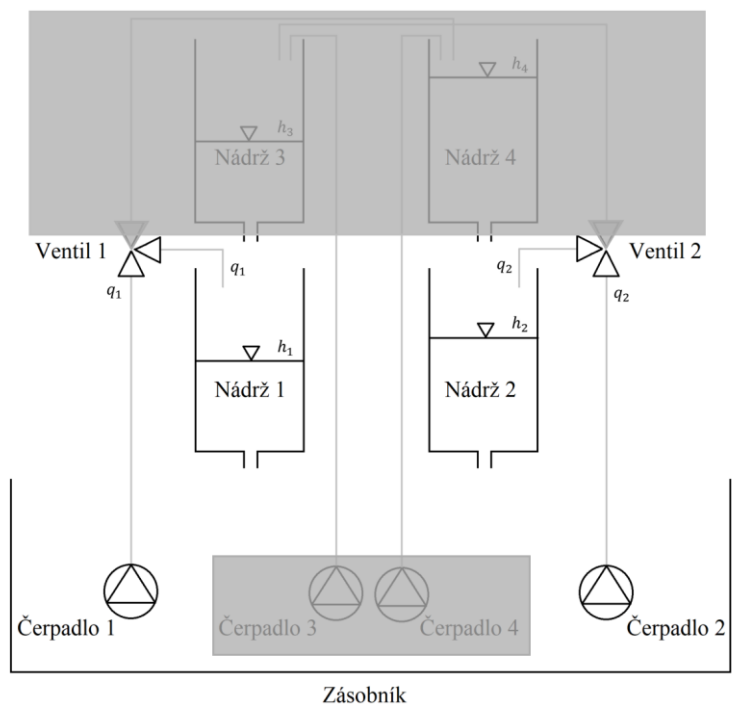
$$\mathbf{G}_S(s) = \begin{bmatrix} \frac{T_1 \gamma_1 k_1}{A_1(T_1 s + 1)} & \frac{T_1 k_2 (1 - \gamma_2)}{A_1(T_1 s + 1)(T_3 s + 1)} \\ \frac{T_2 k_1 (1 - \gamma_1)}{A_2(T_2 s + 1)(T_4 s + 1)} & \frac{T_2 \gamma_2 k_2}{A_2(T_2 s + 1)} \end{bmatrix} \quad (7.20)$$

$$\mathbf{G}_{SV}(s) = \begin{bmatrix} \frac{T_1 k_3}{A_1(T_1 s + 1)(T_3 s + 1)} & 0 \\ 0 & \frac{T_2 k_4}{A_2(T_2 s + 1)(T_4 s + 1)} \end{bmatrix} \quad (7.21)$$

7.4 Další možnosti řízení laboratorního modelu

Laboratorní model může být řízen i odlišnými způsoby než základní konfigurací. Jednou z nabízených možností je kombinace aktivních a neaktivních čerpadel. Lze například řídit pouze napětí přiváděné na čerpadlo 1 a čerpadlo 2 a i přesto by se stále naplňovaly všechny čtyři nádrže a bylo by možné sledovat výšku hladiny v každé z nich.

Dále lze různě manipulovat s ventily, a to například tak, aby nedocházelo k rozdělování průtoku do horních nádrží. V tomto případě by veškerá kapalina z ventilů vcházela pouze do spodních nádrží. Kdyby zároveň byla zanedbána činnost čerpadel 3 a 4, získala by se následující upravená struktura systému

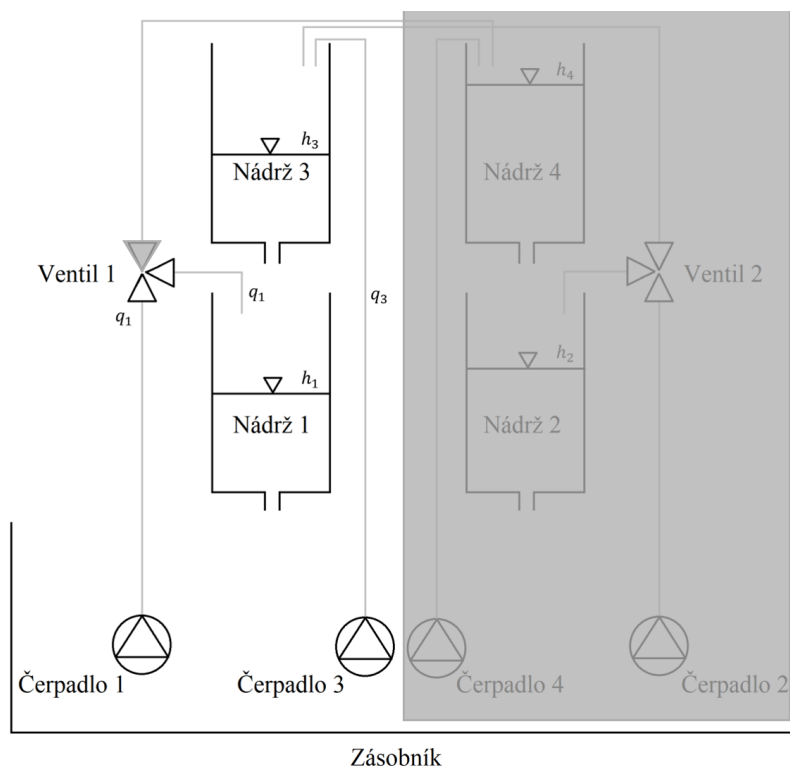


Obrázek 25: Jedna z možných změn struktury laboratorního modelu

Z obrázku výše je patrné, že se změnily i veličiny, které byly ve schématu výše (viz Obrázek 24). Například průtoky q_1 a q_2 přímo vstupují do spodních nádrží a nerozdělují se přes ventil do horních nádrží. Dále nejsou uvažovány průtoky z čerpadla 3 a čerpadla 4, neboť v tomto režimu nejsou aktivní. Pro tuto modifikaci by soustava nelineárních diferenciálních rovnic měla tvar

$$\begin{aligned}\frac{dh_1(t)}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{q_1}{A_1}(t) \\ \frac{dh_2(t)}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{q_2}{A_2}(t)\end{aligned}\tag{7.22}$$

Další nabízenou modifikací je například neuvažování průtoků z čerpadel 2 a 4, tím pádem by nádrže 2 a 4 a výšky hladiny v nich byly ignorovány. Dále kdyby bylo uvažováno, že ventil 1 je nastaven tak, že propouští průtok q_1 přímo do nádrže 1 a nerozděluje průtok do nádrže 4, byla by získána níže uvedená struktura systému



Obrázek 26: Další možná změna struktury laboratorního modelu

A její matematický model by měl tvar

$$\begin{aligned} \frac{dh_1(t)}{dt} &= -\frac{a_1}{A_1} \sqrt{2gh_1(t)} + \frac{a_3}{A_1} \sqrt{2gh_3(t)} + \frac{q_1}{A_1}(t) \\ \frac{dh_3(t)}{dt} &= -\frac{a_3}{A_3} \sqrt{2gh_3(t)} + \frac{q_3}{A_3}(t) \end{aligned} \quad (7.23)$$

Laboratorní model čtyřválcové vodárny tedy nemusí být provozován v jednom konkrétním uspořádání. Díky jeho konstrukci a vybraným komponentům je možné jej nakonfigurovat různými způsoby a realizovat tak různé kombinace aktivních vstupů do systému. Tím je umožněno sledovat odlišné výstupní veličiny a testovat chování systému při různých podmínkách řízení.

8 KONSTRUKCE MODELU ČTYŘVÁLCOVÉ VODÁRNY

Při návrhu konstrukce čtyřválcové vodárny bylo bráno v úvahu několik požadavků. Nejdůležitější bylo, aby model byl snadno přenositelný a tím pádem se s ním dalo jednoduše manipulovat nebo ho přenášet mezi laboratořemi. Přihlíželo se také k rozumnému kompromisu mezi cenou a praktičností řešení. Proto také vzniklo několik návrhů řešení. Snahou tedy bylo zhotovit model z co nejméně komponent, které by bylo v budoucnu potřeba vyměnit, či by bylo složité je znovu koupit v případě jejich selhání.

Během návrhu konstrukce se nabízelo hned několik možných řešení, jak model fyzicky sestavit. Každé řešení neslo s sebou několik výhod, ale také několik nevýhod, musel se tedy najít kompromis.

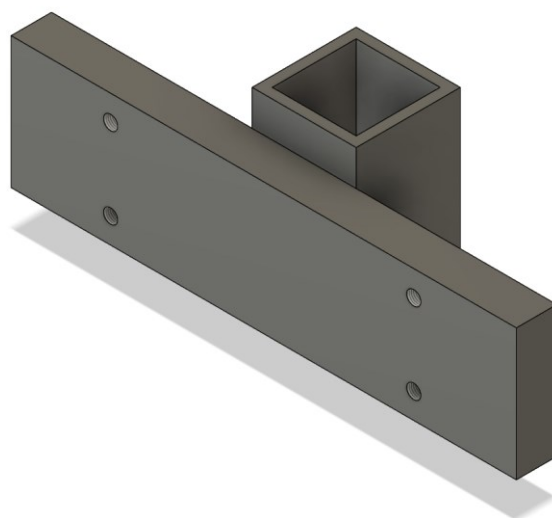
Realizace konstrukce obnášela několik vytištěných 3D dílů. Pro tyto potřeby byly navrženy 3D díly v softwaru Autodesk Fusion 360.

Součástí přílohy je také seznam použitých komponent, které byly zakoupeny za účelem sestavení laboratorního modelu.

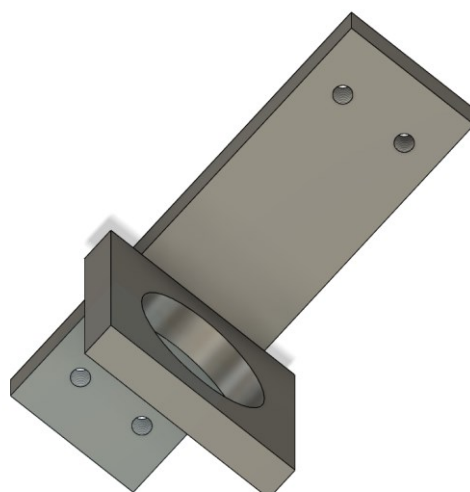
8.1 Prvotní návrh konstrukce

Jako první byla uvažována konstrukce z hliníkových profilů, jenže tato konstrukce s sebou nesla několik nevýhod. K nim se řadila například cena a následné uchycení nádrží, které by bylo problematictější. Uchycení držáků mohlo být vyřešeno pomocí využití 3D tisku, jenže výtisk by byl celkem namáhán, jelikož by musel nést tíhu nádrže a kapaliny obsažené v ní, což by z dlouhodobého hlediska mohlo být problematické. Na druhou stranu mezi kladné stránky využití hliníkových profilů se řadila nízká hmotnost, mechanická odolnost materiálu a zaručená stabilita a možnost úpravy.

Jedno z možných řešení uchycení nádrží bylo nasunutí výtisku na profily (viz Obrázek 27) a následné přišroubování druhé části k nasunutému výtisku (viz Obrázek 28). Během návrhu se však objevila jiná možnost řešení realizace konstrukce, a tak podoba modelů nebyla dotažena až do konce a jednalo se pouze o vizualizace.



Obrázek 27: První část návrhu řešení uchycení nádrží za použití hliníkových profilů [21]



Obrázek 28: Druhá část návrhu řešení uchycení nádrží za použití hliníkových profilů [21]

8.2 Finální návrh konstrukce

Původní záměr postavit konstrukci z hliníkových profilů se ukázal jako nevhodný, jelikož bylo vyžadováno pořízení velkého množství dalších dílů (drážkové kameny, matice, úhelníky apod.), což by znatelně zvýšilo celkové náklady. Proto se hledala jiná varianta, kterou se nakonec stal regál. Použití regálu s sebou neslo stejné výhody, jako použití hliníkových profilů, tedy nosnost, nízkou hmotnost, stabilitu a možnost pro případné úpravy do budoucna, zatímco cena oproti konstrukci z hliníkových profilů výrazně klesla. Jediná překážka byla opět v uchycení nádrží.



Obrázek 29: Návrh konstrukce za použití regálu z více pohledů [21]

K uchycení nádrží by byl opět využit 3D tisk, kde by do polic byly vyvrtány díry, na které by byly nasazeny a přišroubovány vytisklé 3D díly (viz Obrázek 29). Vytisknuté 3D díly by musely mít vespod díry, aby byl umožněn průtok kapaliny z vyšších nádrží do nižších nádrží a následně do zásobníku.

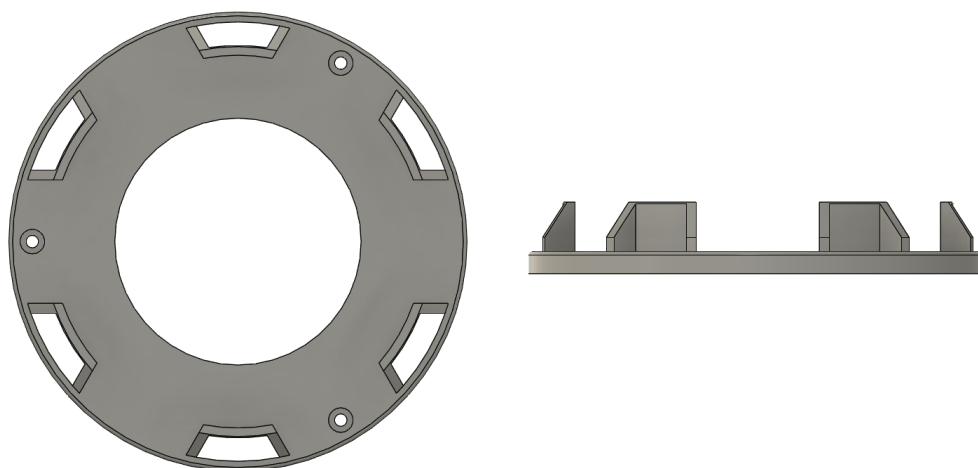
8.3 Realizace konstrukce

Finální podoba konstrukce byla zrealizována z regálu o výšce 120 cm, hloubce 50 cm a šířce 60 cm a 3D vytištěných dílů do nichž jsou dále zasazeny nádrže o průměru 15 cm a výšce 13 cm. Při návrhu 3D modelů bylo dbáno na několik aspektů, například aby díl odolával při různé manipulaci s ním a na precizní dosednutí nádrží, čímž byl zamezen možný pohyb nádrží. Dále bylo přihlíženo na množství použitého materiálu, který oproti prvnímu návrhu držáků (viz Obrázek 29) byl zredukován několikanásobně. Pro vytištění držáků nádrží byl použit materiál PLA a pro všechny ostatní vytištěné komponenty byl využit materiál PLA+.



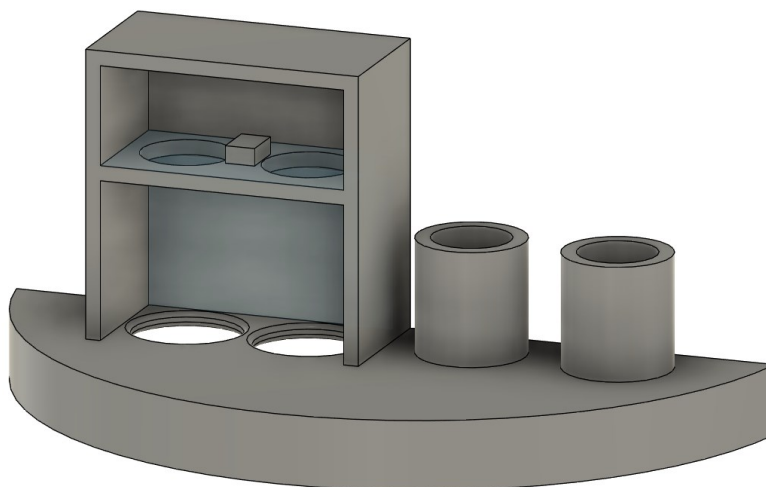
Obrázek 30: Realizace konstrukce

Aby bylo možné odlišit nádrže na první pohled, tak kromě číselného označení je možné použít i barevnou odlišnost držáku pro nádrže. Při modelování držáku pro nádrže bylo také dbáno na to, aby se nádrž jednoduše nepřevrátila, a proto jsou držáky po obvodu patřičně vysoké.



Obrázek 31: Pohled shora a z boku na držák pro nádrže [21]

Dále bylo potřeba namodelovat víko na jednotlivé nádrže. Víko slouží pro umístění senzoru vzdálenosti a pro vsunutí hadiček, kterými je kapalina dopravována do nádrží.



Obrázek 32: Finální podoba víka na nádrže [21]

Na modelu víka proběhlo několik úprav, aby bylo dosaženo co nejpřesnějšího měření. Bylo zjištěno, že se musí brát v úvahu rozptýlení ultrazvukového signálu, vycházejícího ze senzorů, a proto je možno si všimnout na obrázku výše, že vespod jsou umístěny díry o větším průměru, aby jimi mohl ultrazvukový signál spolehlivě procházet a zpátky se do něj vracet. Bylo zjištěno, že zaoblením výstupních otvorů pro ultrazvukový signál je dosaženo potlačení rušivých signálů, které se do senzoru vrací.

Ke kompletaci konstrukce bylo zapotřebí zvolit vhodné ventily. Byly vybrány ventily od firmy ESBE, konkrétně typ VRG 131 15-0.63. Tento ventil umožňuje průtok tekutiny až $0,63 \text{ m}^3/\text{h}$. Je vyroben z mosazi a poskytuje ochranu proti vyluhování zinku. Ventil je možno provozovat jak pro účely směšování, ale i pro účely rozdělování.



Obrázek 33: Použité rozdělovací ventily

9 ŘÍDICÍ JEDNOTKY MODELU

Laboratorní model sestává z hlavní řídicí jednotky, ke které je možno připojit různé externí řídicí systémy. Jedná se například o PLC Tecomat Foxtrot CP 1015 s moduly, Arduino Uno a technologickou kartu PCI-1711U. V této kapitole bude kladen důraz na vysvětlení jednotlivých částí kódu nahraného v hlavní řídicí jednotce. Následně budou popsány vnější řídicí systémy.

9.1 Hlavní řídicí jednotka laboratorního modelu

Pro řídicí jednotku musel být vhodně zvolen mikropočítač. Nakonec bylo vybráno Arduino Mega 2560, které disponuje dostatkem pinů (digitálních i analogových), časovačů a výstupů časovače, které jsou vhodné například pro generování vlastní pulzně šířkové modulace (PWM).

Řídicí jednotka modelu obstarává dvě hlavní činnosti. První z činností je čtení příchozího PWM signálu, díky kterému je generováno napětí na čerpadla pomocí modulů L298N. Další hlavní činností je vysílání informace o vzdálenosti hladiny vody od ultrazvukových senzorů v jednotlivých nádržích do vnějších systémů pomocí digitálně-analogových převodníků.

Hlavním požadavkem na řídicí jednotku bylo dosažení co nejmenší latence mezi jednotlivými instrukcemi, aby se minimalizovalo zpoždění při měření a případné regulaci na laboratorním modelu. Pro programování řídicí jednotky bylo využito prostředí Arduino IDE, které poskytuje již řadu zabudovaných funkcí, mezi něž se řadí například *delay()* a *analogWrite()*. Nevýhodou při použití těchto funkcí je délka jejich režie, kde se oproti přímému zápisu do registrů vykonává několik instrukcí navíc, což může zpomalovat reakci, třeba i o několik stovek taktů čipu. Dalším omezením u funkce *analogWrite()* je použití pevně daného děliče hodin a frekvence PWM, tím pádem uživateli není umožněno jemné ladění PWM signálu.

Tento problém je však řešitelný přímou prací s registry čipu ATmega2560. Otevírá se tak možnost generování vlastní PWM s nastavitelnou frekvencí, vytvoření pravidelných přerušení generovaných časovačem, či rychlé čtení příchozího PWM signálu z vnějšího systému.

9.2 PLC Tecomat Foxtrot CP-1015

PLC Tecomat Foxtrot CP-1015 je výkonný programovatelný automat (PLC), který je možno využít pro široké spektrum aplikací, mezi které je možno zařadit průmyslovou regulaci, řízení a měření, ale i například oblast automatizace budov. Mezi hlavní vlastnosti CP-1015 lze zařadit 6 analogových vstupů s rozlišením převodu 12 bitů, 2 analogové výstupy s rozsahem (0-10) V, 6 binárních vstupů, 6 reléových výstupů (s dobou sepnutí 10 ms a dobou rozepnutí 4 ms), TCL2 sběrnice pro připojení až 10 rozšiřujících modulů, podpora online programování a integrovaný displej s možností ovládání pomocí tlačítek.

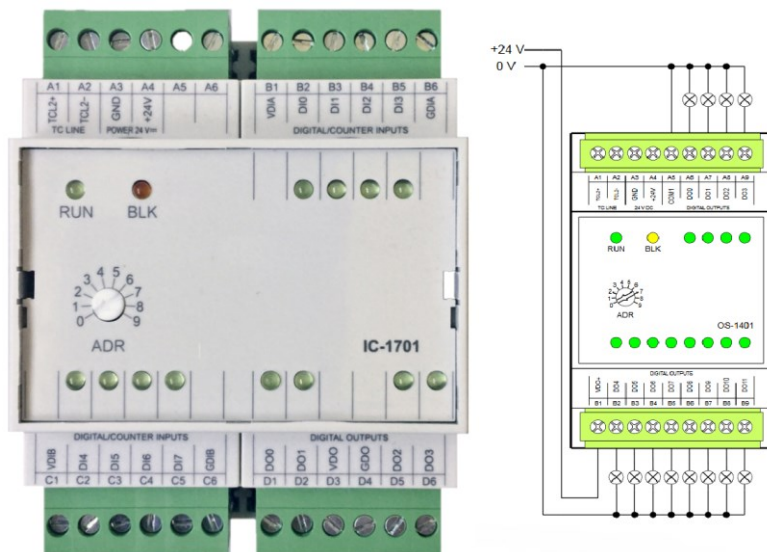
CP-1015 je tedy vhodné použít v aplikacích využívajících kombinaci analogových a digitálních signálů. [22]

9.2.1 Modul OS-1401

Modul Tecomat OS-1401 je rozšiřující výstupní modul, který je navržen ke spínání binárních výstupů v různých aplikacích. Může se jednat například o automatizaci budov, průmyslových procesů a jiných řídicích systémů, kde jsou očekávány různé proudové nároky.

Poskytuje 12 binárních tranzistorových výstupů, rozdělených do dvou skupin. Skupina A zahrnuje výstupy DO0-DO3, které zvládají poskytnout proud až 2 A, přičemž součet proudů ze všech těchto čtyř výstupů nesmí překročit hranici 2,5 A. Další skupina B sestává z výstupů DO4-DO11, kde každý z těchto výstupů zvládá proud pouze do 0,5 A. Doba sepnutí binárního výstupu je udávána jako 400 mikrosekund a doba rozepnutí je udávána také jako 400 mikrosekund. Tyto časy platí pro obě skupiny výstupů.

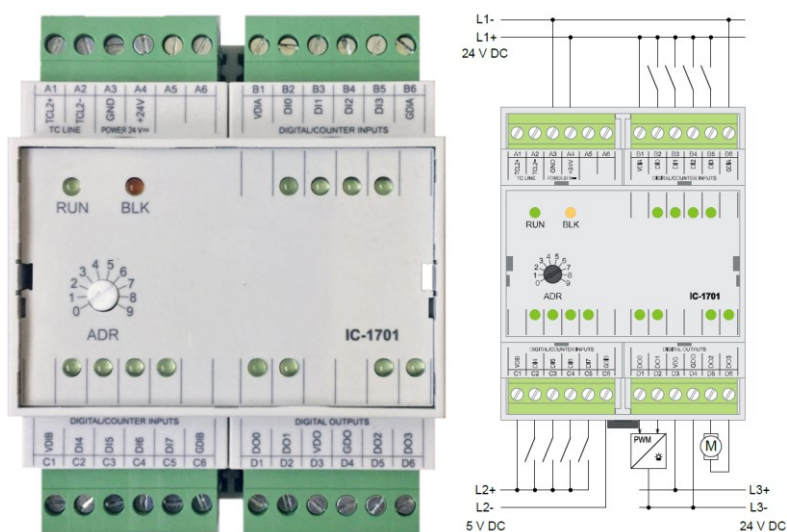
Během práce s tímto modulem si lze všimnout, že obsahuje indikační LED, které poskytují informaci o aktivitě binárních výstupů. Modul je připojován k základnímu modulu (například CP-1015) pomocí TCL2 sběrnice. [23]



Obrázek 34: Modul OS-1401 a příklad jeho zapojení [20]

9.2.2 Modul IC-1701

Modul IC-1701 poskytuje osm binárních vstupů zvládajících vyšší frekvence signálu. Kromě toho modul nabízí čtyři rychlé tranzistorové výstupy, které lze využít pro výstupní PWM signál. Binární tranzistorové výstupy tohoto modulu dokážou rychleji přecházet mezi logickými stavy, než výstupy výše zmíněného modulu OS-1401. Modul je tedy například vhodnější pro generování PWM signálu pro plynulé řízení motorů. Maximální proud, který lze těmito tranzistorovými výstupy spínat v pulsním režimu, je 4 A. Pokud je vyžadováno trvalé dodávání proudu, výstupy jsou schopny poskytnout proud až 2,7 A.

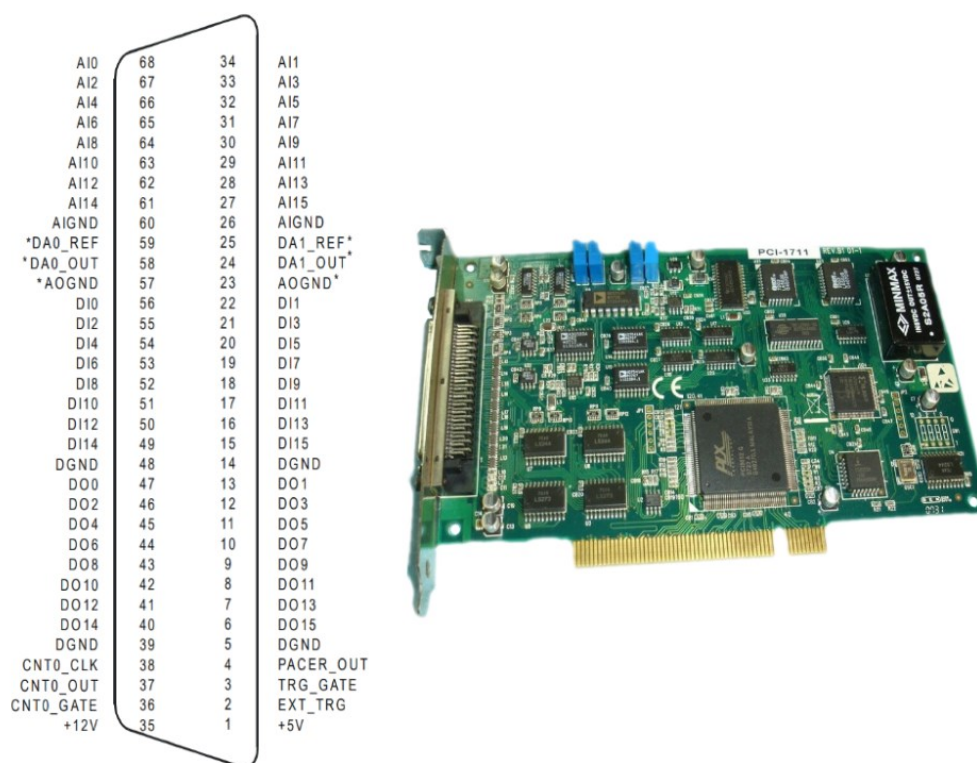


Obrázek 35: Modul IC-1701 a příklad jeho zapojení [24]

9.3 Technologická karta PCI-1711U

Technologická karta PCI-1711U poskytuje potřebné funkce pro měřicí a řídicí systémy, které uživatelé nejčastěji požadují. Poskytuje 16 analogových vstupů, 2 analogové výstupy, 16 digitálních vstupů a 16 digitálních výstupů s 5 V logikou. Analogově-digitální převod probíhá s rozlišením 12 bitů.

Analogové výstupy lze nastavit v rozmezí (0-5) V nebo (0-10) V, tím pádem uživatel může posílat analogové signály do zařízení, jako je Arduino Uno, ale také třeba do PLC a nemusí nijak externě upravovat signál. Na této kartě je možno také nastavit rozsah analogových vstupů. Využití této karty může být například v průmyslu, pro který je tato technologická karta standardizována, ale také například v měřicích aplikacích nebo právě v laboratorních modelech. [25]

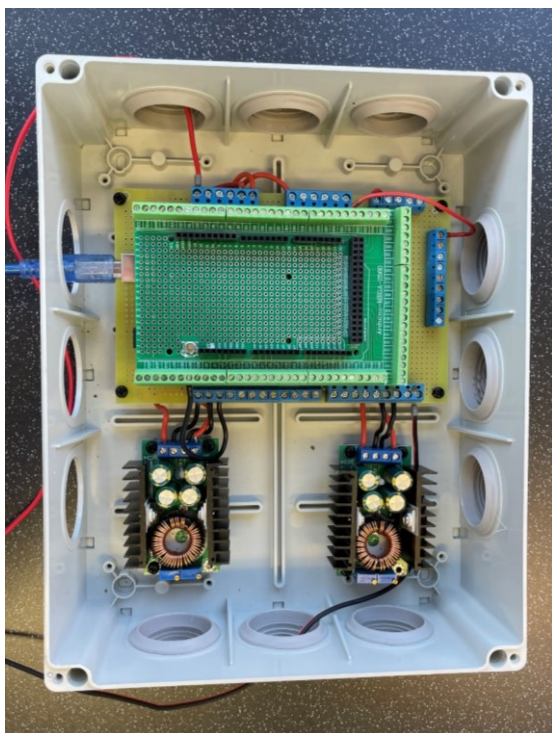


Obrázek 36: Rozmístění pinů vstupně-výstupního konektoru technologické karty PCI-1711 (levá část obrázku) a technologická karta PCI-1711 (pravá část obrázku) [25]

10 ELEKTRONICKÉ ZAPOJENÍ MODELU

Během návrhu elektronické části laboratorního modelu byl zohledněn na počet použitých součástek, jednoduchost vyměňování informací a hlavně na frekvenční a napěťovou kompatibilitu mezi součástkami a řídicími systémy (hlavním a vedlejšími). V této kapitole bude popsáno, jaké elektronické součástky byly využity, řídicí jednotky a jak jsou informace posílány mezi zařízeními.

Pro umístění komponent byly použity elektroinstalační krabice, do kterých byly následně vyvrtány otvory pro umístění distančních sloupků. Na distanční sloupky byly poté připevněny jednotlivé využití komponenty laboratorního modelu. Ve větší elektroinstalační krabici je umístěno Arduino Mega 2560 společně se step-down měniči. V menší elektroinstalační krabici jsou pak umístěny moduly L298N a digitálně-analogové převodníky (viz Obrázek 37).



Obrázek 37: Umístění řídicí jednotky a step-down měničů napětí ve voděodolné krabici před kompletací zapojení

10.1 Využití elektronické součástky

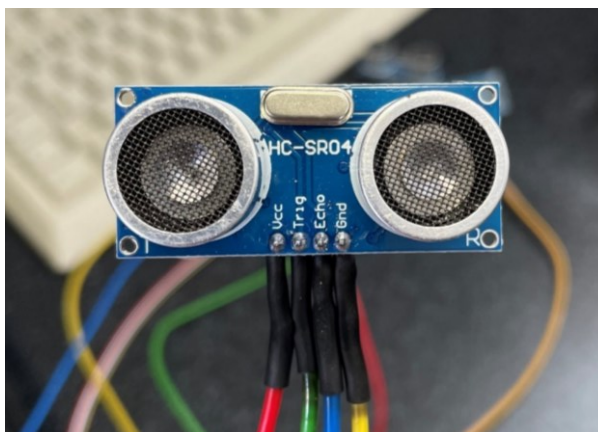
Laboratorní model sestává celkově ze čtyř odlišných komponent, které zajišťují sejmutí vzdálenosti hladiny, převod signálu z digitální formy na analogovou, ovládání výkonu

čerpadel a samotné poskytnutí energie tekutině pro dopravu z reservoáru do nádrží. Dále byly použity step-down měniče napětí, pro úpravu vstupního napětí na žádané hodnoty.

10.1.1 Ultrazvukové senzory vzdálenosti

Pro měření vzdálenosti hladiny byl zvolen cenově dostupný a snadno vyměnitelný ultrazvukový senzor HC-SR04. Senzor obsahuje dvě hlavní části – vysílač a přijímač.

Měření pomocí těchto senzorů probíhá tak, že řídicí jednotka (v tomto případě Arduino Mega 2560) aktivuje na dobu 10 μ s vstupní pin senzoru, který je označen jako *TRIG*. Tato aktivace způsobí, že senzor vygeneruje krátký ultrazvukový signál o frekvenci 40 kHz. Pokud ultrazvukový signál narazí na překážku, odrazí se zpět k senzoru. Jakmile je odražený signál detekován, dojde k deaktivaci výstupního pinu senzoru, který je označen jako *ECHO*, který byl aktivován ihned po vyslání ultrazvukového signálu. Doba trvání signálu logické 1 na *ECHO* pinu je úměrná vzdálenosti překážky. Senzor je připojen ke společné zemi a je napájen pomocí 5 V z řídicí jednotky.



Obrázek 38: Ultrazvukový senzor HC-SR04

Ultrazvukové senzory byly zvoleny proto, že poskytují bezkontaktní, poměrně přesné a rychle reagující měření výšky hladiny. Byly také zvažovány laserové (ToF – Time of Flight) senzory vzdálenosti, ale ty by nebyly tak přesné, jelikož vysoká odrazivost hladiny čiré kapaliny by mohla způsobovat chybné odečty. Hodí se tak spíše pro aplikace, kde je zabarvená, či kalná kapalina.

10.1.2 Digitálně analogové převodníky

Tyto převodníky zajišťují převod PWM signálu se střídou (0-100) % na napětí v rozmezí (0-10) V. Aby byl zajištěn správný převod, musí být vstupní PWM signál do těchto převodníků v rozmezí frekvencí 1-3 kHz.

Řídicí jednotka obstarává vysílání PWM signálu o frekvenci 2 kHz do digitálně analogových převodníků, ze kterých poté analogová hodnota putuje do vnějších systémů. Vzdálenost hladiny od ultrazvukových senzorů je přepočítána na hodnotu, která je uložena do *OCRnx*, díky čemuž je poté generován PWM signál o střídě, pomocí které je na výstupu digitálně-analogových převodníků analogový signál odpovídající vzdálenosti.

Tyto převodníky potřebovaly kalibraci. Ta se provádí ručně pomocí potenciometru umístěného na převodníku. Bylo zjištěno, že nejpřesnější výsledky jsou dosaženy, pokud je napájecí napětí převodníků 13,5 V, protože bylo zjištěno, že při této hodnotě vstupního napětí mají převodníky nejmenší chybu převodu. Nevýhodou je, že tyto převodníky mají přípustnou chybu až 5 %. Měřením však bylo zjištěno, že tato chyba je poměrně malá.

Testování bylo navrženo tak, že převodník měl na výstupu takovou hodnotu napětí, která odpovídá vzdálenosti. První testování probíhalo při frekvenci 500 Hz vstupního PWM signálu do převodníku, tedy menší frekvence, než je doporučena výrobcem. Výsledky převodů jsou uvedeny v tabulce níže

Tabulka 1: Testování chyby převodu digitálně-analogových převodníků při frekvenci 500 Hz

Testování při 500 Hz		
Vzdálenost [cm]	Napětí [V]	Chyba
9,2	9,103	1,054348
5,3	5,266	0,641509
2,62	2,581	1,48855
3,69	3,66	0,813008
3,04	3,021	0,625
6,82	6,766	0,791789
5,95	5,907	0,722689
4,01	3,999	0,274314
7,23	7,194	0,497925
7,87	7,811	0,749682
Průměrná chyba převodu [%]		0,765881

Poté bylo otestováno, jestli se průměrná chyba převodu sníží při takové frekvenci vstupního signálu, kterou výrobce doporučuje. Frekvence vstupního signálu pro druhé testování byla 1000 Hz a výsledek testu je uveden v tabulce níže

Tabulka 2: Testování chyby převodu digitálně-analogových převodníků při frekvenci 1000 Hz

Testování při 1000 Hz		
Vzdálenost [cm]	Napětí [V]	Chyba
8,42	8,336	0,997625
5,63	5,601	0,515098
4,34	4,322	0,414747
3,06	3,078	0,058824
8,09	8,042	0,593325
5,3	5,28	0,377358
7,23	7,192	0,525588
3,37	3,355	0,445104
6,8	6,774	0,382353
9,5	9,495	0,052632
Průměrná chyba převodu [%]		0,436265

Z tabulky je patrné, že při dosažení hodnoty frekvence, kterou doporučuje výrobce, bylo dosaženo zmenšení chyby převodu. Z tohoto důvodu byly v řídicí jednotce vybrány určité piny a poté naprogramovány tak, aby poskytovaly výstupní PWM signál o frekvenci 2 kHz.

10.1.3 Moduly pro řízení čerpadel

Bylo otestováno pět různých modulů pro řízení čerpadel. Z hlediska dlouhodobého provozu a přihlížení k faktu, že měření může trvat i v řádech hodin, se jako nejlepší zdály moduly L298N. Tyto moduly obsahují dva H-můstky, umožňující řízení výkonu na čerpadle. Jsou ovládány logickým napětím 5 V a jsou schopny poskytnout na výstupu napětí v rozmezí (5-35) V a maximálním proudem 2 A na kanál.

Pro každé čerpadlo byl využit jeden modul L298N, který byl zapojen paralelně. Byly tedy propojeny výstupy OUT1-OUT4, OUT2-OUT3, ENA-ENB, IN1-IN4 a IN2-IN3. Tímto bylo dosaženo, že se modul tolik nepřehříval a výstupní výkon z modulu byl tak stabilnější. Hodnota výstupního výkonu z modulu je ovládána pomocí příchozího PWM signálu na propojené piny ENA a ENB.



Obrázek 39: Paralelní zapojení výstupů modulu L298N

Vstupní napětí do každého modulu bylo testováním stanoveno na hodnotu 11,05 V. Při tomto vstupním napětí je na výstupu k čerpadlům napětí o maximální hodnotě 9 V. Tento pokles napětí je zapříčiněn úbytkem napětí, který je způsoben tranzistory modulu.

10.1.4 Čerpadla

Pro poskytnutí energie kapalině byla vybrána čerpadla od firmy Barwig, konkrétně Barwig Typ 04 0444. Toto čerpadlo je možno provozovat při napětí až 12 V, avšak pouze krátkodobě. Při napájecím napětí do 9 V je možný trvalý provoz, ale čerpadlo musí být kompletně ponořeno. Maximální průtok čerpadla je 10 l/min a čerpací výška dosahuje až 6 m. [26]



Obrázek 40: Ponorné čerpadlo Barwig Typ 04 0444

10.1.5 Step-down měniče napětí

Laboratorní model vyžaduje více napětových úrovní, jelikož do modulů L298N a digitálně-analogových převodníků vstupují různé hodnoty napětí. Pro poskytnutí více napětových úrovní byly využity měniče napětí XL4016, které se vykazují vysokou účinností při přeměně podstatně vyššího vstupního napětí na napětí nižší.

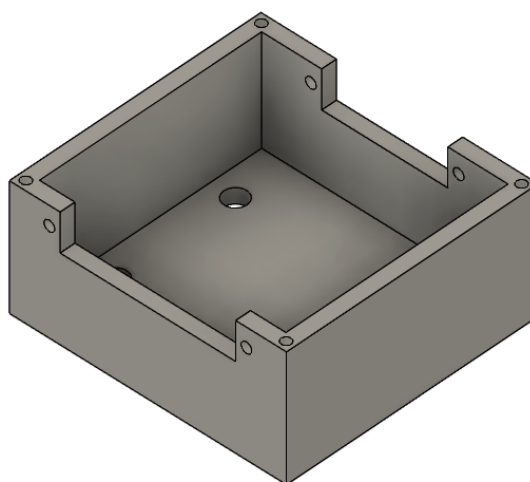
Regulace výstupního napětí i proudu je umožněna pomocí trimrů. Měnič je schopen pracovat se vstupním napětím až 40 V a na výstupu poskytnout napětí v rozmezí (1,25-36) V při maximálním výstupním proudu až 8 A.

10.2 Propojovací krabice

Aby bylo možno připojit vnější systém k řídicí jednotce a vyměňovat tak signály mezi nimi, je zapotřebí použít propojovací krabici. Důvod použití krabice je kvůli tomu, aby nedošlo k poškození řídicí jednotky.

Řídicí jednotka totiž potřebuje příslušné napětové úrovně (maximálně 5 V). Pokud by bylo PLC s příslušným modulem připojeno k řídicí jednotce napřímo, došlo by k poškození řídicí jednotky, jelikož PLC s příslušným modulem poskytuje na digitálních výstupech napětí 24 V. Tím pádem by řídicí jednotka byla nevratně poškozena. Pro PLC s příslušným modulem, mikropočítač i technologickou kartu jsou využity odlišné moduly.

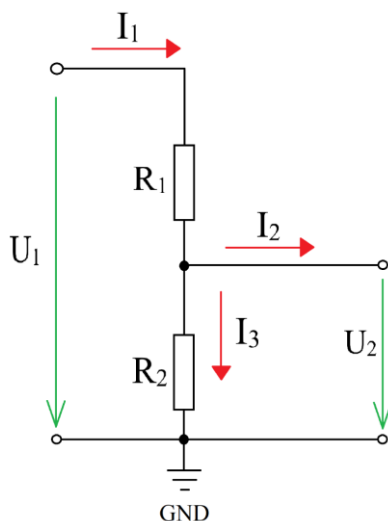
V každé propojovací krabici je umístěno pájivé pole, které je připevněno distančními sloupky. Krabice byla opět vymodelována a vytištěna na 3D tiskárně.



Obrázek 41: 3D model propojovací krabice

10.2.1 Krabice pro připojení PLC s příslušným modulem k řídicí jednotce

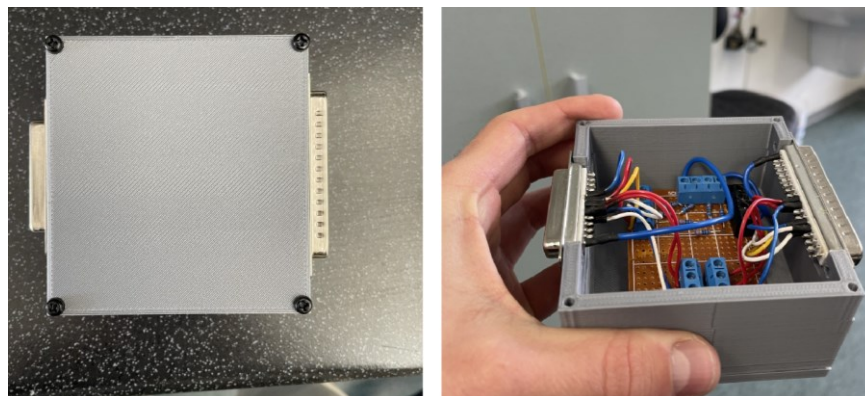
Tyto krabice sráží napětí digitálních výstupů z modulu (IC-1701 nebo OS-1401) z 24 V na 4,8 V. Při této napěťové úrovni řídicí jednotka (Arduino Mega2560) je stále schopna detekovat výskyt stavu logická 1 na digitálních vstupech. Pro sražení napětí na takovou hodnotu byl použit napěťový dělič, pro který byly vypočteny hodnoty rezistorů.



Obrázek 42: Znázornění použitého napěťového děliče

Hodnoty rezistorů byly stanoveny na $R_1=3300\ \Omega$ a $R_2=820\ \Omega$. Napětí U_1 označuje napětí ze zdroje, což je PWM signál vycházející z příslušného modulu PLC. Napětí U_2 představuje sražené napětí, které se dostává do řídicí jednotky. Hodnoty proudů jsou zanedbatelné, jelikož se jedná o signálové výstupy, které se pro oba moduly pohybují okolo hodnoty 5 mA.

Výstupní analogové napětí z digitálně-analogových převodníků, nesoucí informaci o vzdálenosti hladiny, není potřeba upravovat. PLC je totiž schopno pracovat s analogovým napětím o maximální hodnotě 10 V a digitálně-analogový převodník vyšší analogové napětí nedokáže poskytnout.



Obrázek 43: Propojovací krabice pro připojení PLC s modulem IC-1701

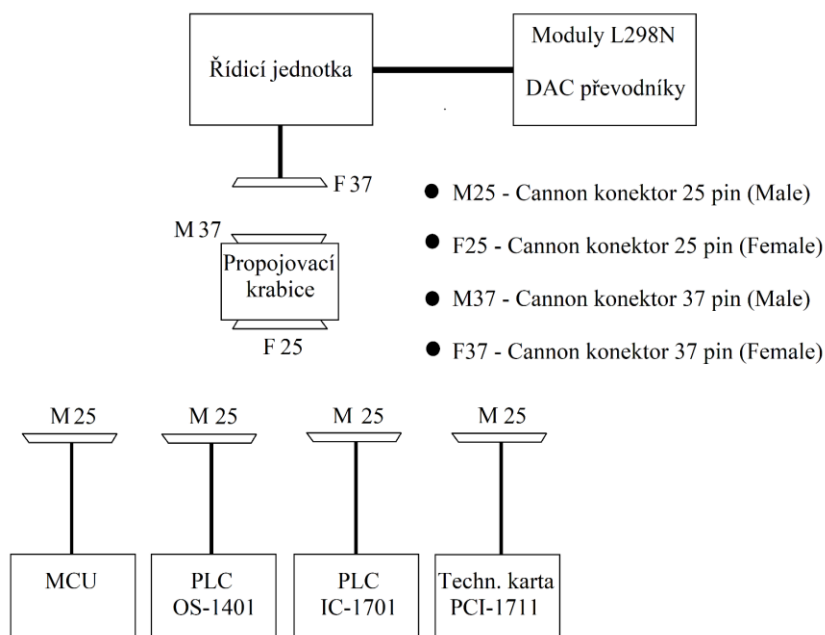
10.2.2 Krabice pro připojení mikropočítače a technologické karty PCI-1711

Pro mikropočítač i technologickou kartu PCI-1711 je možno použít stejnou krabici. Je to z důvodu, že oba systémy poskytují na digitálních výstupech maximálně napětí 5 V, které je řídicí jednotka schopna bez problémů zpracovat a nemusí tedy být na digitální výstupy z vedlejších systémů použit dělič napětí. Tato propojovací krabice však upravuje napěťovou úroveň analogových signálů.

Z digitálně-analogových převodníků se do propojovací krabice dostává analogové napětí v rozmezí (0-10) V, a tak je pro mikropočítač potřeba upravit napěťovou úroveň na 5 V. Pro tento účel byl použit dělič napětí s odpory $R_1=R_2=10\text{ k}\Omega$, který spolehlivě dělí napětí na polovinu a do mikropočítače se dostává takové napětí, které jej není schopno poškodit. Technologická karta PCI-1711 poskytuje možnost nastavit rozsah vstupních analogových systémů z rozmezí (0-10) V na (0-5) V, ale takováto úprava vstupního analogového signálu by mohla být řešena i programově. Avšak například při použití Arduino Uno jako mikropočítače, je podmínkou použití externího napájení. Například USB step-up měniče napětí z 5 V na 9 V.

10.3 Blokové schéma propojení jednotlivých částí

Způsob připojení vnějšího systému k řídicí jednotce je realizován prostřednictvím propojovacích krabic, které mají na straně propojení k vnějšímu systému 25 pinový konektor. Řídicí jednotka se k propojovací krabici připojí pomocí 37 pinového konektoru, takže je znemožněno připojit vnější zařízení napřímo k řídicí jednotce a tím ji případně poškodit.

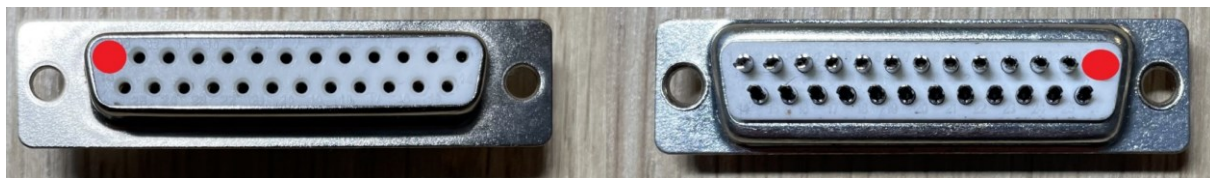


Obrázek 44: Blokové schéma propojení jednotlivých částí modelu čtyřválnové vodárny

Jak bylo popsáno výše, připojením vnějšího systému ke krabici je docíleno toho, že se do řídicí jednotky nedostanou napěťové úrovně, které by řídicí jednotku mohly poškodit. Například pokud chce uživatel připojit k řídicí jednotce PLC s příslušným modulem, musí nejprve vybrat vhodnou propojovací krabici, která je schopna upravit napěťové úrovně digitálních signálů na takovou napěťovou úroveň, kterou je řídicí jednotka schopna zvládnout.

10.4 Propojovací konektory a rozmístění jednotlivých pinů

Pro připojení jednotlivých vedlejších řídicích systémů k hlavní řídicí jednotce jsou přiřazeny jednotlivé piny, které vedou příslušné signály na konektoru Cannon 25. Jednotlivé číslování na konektoru je bráno z pohledu, když je konektor pájen.



Obrázek 45: Ukázka číslování pinů na konektoru Cannon 25

Pin, který je za kruhem, vyznačeným červenou barvou, je považován jako pin 1. Z toho plyne, že v levém horním rohu z pohledu pájení (pravá část obrázku) se nachází pin 13 a v levém dolním rohu je pin 25.

10.4.1 Připojení mikropočítače a technologické karty PCI-1711 k propojovací krabici

Z vysvětlení číslování jednotlivých pinů konektoru, je tedy pro připojení technologické karty PCI-1711 a mikropočítače na konektoru umístěném na krabici, využita první skupina pinů 3, 4, 16, 17 pro připojení příchozího PWM signálu pro řízení čerpadel, druhá skupina pinů 11, 12, 24, 25 pro připojení analogových signálů nesoucích informaci o vzdálenosti a pin 13 slouží k propojení GND mezi hlavní řídicí jednotkou a vedlejším řídicím systémem.

Pokud chce tedy uživatel propojit vedlejší řídicí systém k propojovací krabici, musí na konektoru vedoucím z vedlejšího systému použít skupinu pinů 10, 11, 22, 23 pro odchozí PWM signály, skupinu pinů 2, 3, 14, 15 pro příchozí analogové signály a pin 1 pro propojení GND.

10.4.2 Připojení PLC s modulem IC-1701 nebo OS-1401 k propojovací krabici

PLC spolehlivě zvládá čist vstupní analogové napětí v rozsahu (0-10) V a není jej tedy potřeba nijak upravovat. Aby však řídicí jednotka zvládla přečíst PWM signál z modulů pro PLC, napětí muselo být rozděleno v určitých poměrech (viz 10.2.1).

Z toho plyne, že analogové signály mohou sdílet stejné piny konektoru, avšak PWM signály musí být rozděleny do dvou skupin, jelikož vstupují na odlišné piny řídicí jednotky. Konektor umístěný na krabici má analogové signály umístěny na pinech 7, 8, 20, 21, PWM signály z modulu OS-1401 jsou umístěny na pinech 1, 2, 14, 15 a PWM signály z modulu IC-1701 jsou umístěny na pinech 5, 6, 18, 19. Pro propojení GND vedlejšího a hlavního řídicího systému je opět využít pin 13.



Obrázek 46: Umístění pinu na propojovací krabici pro PLC s modulem IC-1701

Pokud chce uživatel připojit PLC s modulem OS-1401 systém k propojovací krabici, musí být na konektoru vedoucím z tohoto vedlejšího systému rozmístění pinů tak, že PWM

signály jsou připojeny k pinům 12, 13, 24, 25 a analogové signály k pinům 6, 7, 18, 19 a GND k pinu 1.

Jestliže chce uživatel připojit PLC s modulem IC-1701 systém k propojovací krabici, musí být na konektoru vedoucím z tohoto vedlejšího systému rozmístění pinů tak, že PWM signály jsou připojeny k pinům 8, 9, 20, 21. Připojení analogových signálů a GND je stejný jako za použití modulu OS-1401.

11 PROGRAMOVÉ VYBAVENÍ LABORATORNÍHO MODELU

Jak již bylo zmíněno výše, o běh laboratorního modelu se stará kód nahraný v hlavní řídicí jednotce. V této kapitole bude popsán postup tvorby programu, přičemž bude kladen důraz na vysvětlení práce s registry čipu ATmega2560.

11.1 Hlavní části programu řídicí jednotky

Mezi hlavní části programu řídicí jednotky se řadí sejmутí vzdálenosti hladiny z ultrazvukového senzoru, výpočet mediánu vzdálenosti, čtení příchozího PWM signálu a vysílání PWM signálu. Během programování bylo často pracováno s registry, pro které je zásadní umět pracovat s datasheetem daného čipu. Ten obsahuje informace o uspořádání bitů v jednotlivých registrech a o tom, čeho všeho lze dosáhnout zapsáním logické jedničky do daného bitu registru. Dále je potřeba znát logické operace, ve většině případů však stačí znát pouze logickou operaci *AND* a *OR*.

Při programování kódu pro řídicí jednotku bylo využito prostředí Arduino IDE, které poskytuje již řadu zabudovaných funkcí, mezi které se řadí například *delay()* a *analogWrite()*. Nevýhodou při použití těchto funkcí je délka jejich režie, kde se oproti přímému zápisu do registrů vykonává několik instrukcí navíc, což může zpomalovat reakci, třeba i o několik stovek taktů čipu. Dalším omezením u funkce *analogWrite()* je použití pevně daného děliče hodin a frekvence PWM, tím pádem uživateli není umožněno jemného ladění PWM signálu.

Tento problém je však řešitelný přímou prací s registry čipu ATmega2560. Otevírá se tak možnost generování vlastní PWM s nastavitelnou frekvencí, vytvoření pravidelných přerušení generovaných časovačem, či rychlé čtení příchozího PWM signálu z vnějšího systému.

11.1.1 Sejmутí vzdálenosti hladiny z ultrazvukového senzoru

Pro snímání vzdálenosti hladiny od ultrazvukového senzoru je využito přerušení generované každých 200 ms pomocí 16bitového časovače 1 (Timer 1). Aby byla zaručena přesná perioda přerušení každých 200 ms, což odpovídá frekvenci 5 Hz, je nejprve potřeba vypočítat, kolik taktů čipu uběhne za tuto dobu

$$\text{počet taktů} = \frac{f_{\text{clock}}}{\text{prescaler} * f_{\text{žádaná}}} - 1 [-] \quad (11.1)$$

kde f_{clock} je frekvence čipu ATmega2560, $prescaler$ je hodnota nastavené děličky hodinového signálu a $f_{žádaná}$ je frekvence, které chceme dosáhnout. Po dosazení je získáno:

$$počet\ taktů = \frac{16000000}{256 * 5} - 1 = 12499 \quad (11.2)$$

Timer 1 čipu ATmega2560 je 16bitový, čítač tedy přeteče při hodnotě 65535. Od této hodnoty se odečte počet taktů čipu za 200 ms. Tímto způsobem je získána počáteční hodnota Counteru 1. [27]

$$TCNT1 = 65535 - 12499 = 53036 \quad (11.3)$$

Výsledný kód pro nastavení generování přerušení každých 200 ms je tedy:

```
TCCR1A = 0; //nastavení všech bitů v registru pro Timer1 na hodnotu 0
TCCR1B = 0; //nastavení všech bitů v registru pro Timer1 na hodnotu 0

TCNT1 = 53036; // Counter 16 bit, tedy čítá do 65535, abychom získali
přesných 200ms, nastavíme jeho počáteční hodnotu na 53036.

TCCR1B |= (1 << CS12); // Prescaler nastavena na 256 zápisem logické 1
do bitu CS12

TIMSK1 |= (1 << TOIE1); // povolení timer overflow přerušení
```

Dále bylo potřeba naprogramovat samotnou obslužnou rutinu přerušení pro Timer 1, jinak by bylo generováno přerušení každých 200 ms, ale nic by se nedělo. Snahou je, aby obslužná rutina byla co nejkratší, nejlépe aby byl pouze kontrolován stav jednotlivých pinů na portech, či nastavení nějakého příznaku a řízení tedy bylo co nejrychleji vráceno zpět do hlavního programu. Pokud by bylo v obslužné rutině přerušení použito volání funkcí, či složitější matematické operace, mohlo by dojít k blokaci dalšího příchozího přerušení.

```
ISR(TIMER1_OVF_vect) {
    zmer = true;
    TCNT1 = 53036; // Aby čas do dalšího přetečení byl přesně 200ms
}
```

V tomhle případě je ISR (Interrupt Service Routine) pro Timer 1 velmi jednoduchá, dochází pouze k nastavení příznaku a opětovnému nastavení počáteční hodnoty čítače.

Následně bylo potřeba vytvořit úsek kódu pro samotné ultrazvukové senzory vzdálenosti, aby řídicí jednotka měla informaci o vzdálenosti hladiny vody. Pro tyto účely byl nejprve použit následující úsek kódu

```
void zapsani_do_pole1(void) {
    if (i1 < 5) {
        digitalWrite(trigPin1, LOW);
        //PORTB &= ~(1 << 5); // zapsání logické nuly do pinu
        delayMicroseconds(5);
        digitalWrite(trigPin1, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin1, LOW);
        duration1 = pulseIn(echoPin1, HIGH);
        // Rychlost šíření zvuku ve vzduchu je závislá na teplotě a
        vlhkosti, hodnota je pro 20 stupňů Celsia
        distance1 = (duration1 * 0.0343) / 2;
        retezec1[i1] = distance1;
        i1++;
    }
}
```

Vypočtená vzdálenost byla tedy uložena do pole *retezec1*. Každým voláním této funkce se také zvýšila hodnota proměnné *i1*. Pokud tato proměnná dosáhla hodnoty uvedené v podmínce *if()*, tak byl vypočten medián z těchto uložených hodnot. Medián byl vypočten pomocí následujícího úseku kódu

```
void serazeni_pole1(void) {
    for (int m = 0; m < 5; m++) { // Opakované průchody pro úplné
        seřazení
        for (int s = 4; s > m; s--) { // Procházíme pole od konce
            if (retezec1[s] < retezec1[s - 1]) {
                float d = retezec1[s];
                retezec1[s] = retezec1[s - 1];
                retezec1[s - 1] = d;
            }
        }
    }
    pwm_1 = retezec1[2];
    if (pwm_1 <= 17.0) {
        OCR3A = pwm_1 * 29.41177; // pokud je hodnota menší nebo rovna 17,
        zůstane zapsaná v bitu OCR3A do doby, než se načte nová hodnota <=17
        cm
    }
    for (int z = 0; z <= 4; z++) {
        retezec1[z] = 0.0;
    }
}
```

Tímto bylo dosaženo, že se hodnoty uložené v poli seřadily od nejmenší po největší a následně byla uložena prostřední hodnota do proměnné *pwm_1*, která představuje vzdálenost hladiny od senzoru. Do registru *OCR3A* je poté uložena hodnota, představující požadovanou výstupní frekvenci. Maximální hodnota, která může být do registru uložena je 500. Při této hodnotě je generován výstupní PWM signál se střídou 100 %. Dále je ošetřeno, se do proměnné *pwm_1* uloží nežádoucí hodnota, tak nedojde k zápisu nové hodnoty do registru *OCR3A* a čeká se, dokud se do proměnné *pwm_1* neuloží vhodná hodnota.

11.1.2 Vysílání informace o vzdálenosti do vnějších systémů

Údaj o vzdálenosti hladiny získaný z ultrazvukových senzorů, je do vnějších systému předáván prostřednictvím PWM signálu, generovaného řídicí jednotkou. Pokud je vnějším systémem PLC nebo technologická karta, PWM signál z řídicí jednotky putuje nejdříve skrz digitálně-analogový převodník. Do PLC nebo technologické karty se tedy dostává analogové napětí v rozmezí (0-10) V.

Aby bylo Arduino Mega 2560 zabudované v řídicí jednotce schopno generovat vhodný PWM signál pro digitálně analogový převodníky, bylo nutno nastavit příslušné registry časovačů. Pro generování PWM byl zvolen režim 8, tedy *Phase and Frequency Correct* s nejvyšší možnou dosažitelnou hodnotou čítače uloženou v registru *ICRn* (Input Capture Register), kde *n* označuje číslo příslušného časovače. V tomto režimu čítač čítá nejdříve nahoru a poté dolů.

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM

Obrázek 47: Možnosti generování PWM pro ATmega2560 [28]

Pro generování PWM byly použity čítače číslo tři a čtyři. V datasheetu čipu ATmega2560 bylo potřeba nalézt, v jakém registru příslušného čítače se nastavují *WGM* (Waveform generation mode) bity, aby bylo dosaženo požadovaného PWM signálu.

Bit	7	6	5	4	3	2	1	0	
(0x90)	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obrázek 48: Timer/Counter 3 Control Register A (TCCR3A) [28]

Z obrázku výše plyne, že je potřeba zapsat na bit *WGM30* a *WGM31* logickou nulu. V registru *TCCR3A* nejsou k dispozici bity *WGM32* a *WGM33*, ty se nacházejí v následujícím osmibitovém registru *TCCR3B* časovače 3 (Timer 3).

Bit	7	6	5	4	3	2	1	0	
(0x91)	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Obrázek 49: Struktura Timer/Counter 3 Control Register B (TCCR3B) [28]

Po nalezení potřebných informací o struktuře registrů byl do nich zapsán kód:

```
TCCR3A = 0;
TCCR3B = 0;
TCCR3B |= (1 << WGM33) | (1 << CS31);
```

Tímto kódem bylo zajištěno požadované nastavení registru *TCCR3B*, jelikož na začátku byly registry vynulovány a až poté do nich byly zapsány logické jedničky. Zapsáním logické jedničky do bitu *CS31* bylo dosaženo nastavení prescaleru na hodnotu 8.

Dále bylo zapotřebí nastavit *Compare Output Mode*, díky kterému je zajištěno žádané nastavování příslušných *OCnx* (Output Compare) pinů časovače na stav logická nula, nebo stav logická jedna.

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 =9 or 11: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match when up-counting Set OCnA/OCnB/OCnC on compare match when downcounting
1	1	Set OCnA/OCnB/OCnC on compare match when up-counting Clear OCnA/OCnB/OCnC on compare match when downcounting

Obrázek 50: Compare Output Mode pro Phase and Frequency Correct PWM [28]

Potřebné bity *COM3A1*, *COM3A0* a *COM3B1*, *COM3B0* jsou umístěny v registru *TCCR3A* (viz Obrázek 48). Byl tedy do nich proveden potřebný zápis ve tvaru:

```
TCCR3A |= (1 << COM3A1) | (1 << COM3B1);
```

```
TCCR3A &= ~(1 << COM3A0);
TCCR3A &= ~(1 << COM3B0);
```

Tímto zápisem bylo docíleno, že pokud čítač čítá nahoru a narazí na hodnotu uloženou v *OCR_n* (Output Compare register), konkrétně registrech *OCR3A* a *OCR3B*, dochází k nastavení porovnávacích pinů (*OC3A* a *OC3B*) do stavu logická nula. Pokud čítač načítal do maximální hodnoty uložené v registru *ICR3*, začne od této hodnoty odčítat směrem k nule. V případě, že čítač narazí opět na hodnotu uloženou v *OCR3A* nebo *OCR3B*, nastaví tentokrát příslušné porovnávací piny do stavu logická jedna. Z toho vyplývá, že když do *ICR3* byla zapsána hodnota 500 a do registru *OCR3A* byla zapsána hodnota 400, tak výsledný PWM signál měl střidu 80 %.

Jestliže registr *ICR3* má v sobě uloženou hodnotu 500, výstupní PWM signál na daném pinu by měl frekvenci, stanovenou vzorcem ve tvaru

$$f_{OCnxFPCPWM} = \frac{f_{clock}}{2 * N * TOP} [Hz] \quad (11.4)$$

Po dosazení údajů uvedených výše se získává hodnota frekvence

$$f_{OCnxFPCPWM} = \frac{16000000}{2 * 8 * 500} = 2000 \text{ Hz} \quad (11.5)$$

kde *N* označuje hodnotu děličky a *TOP* představuje hodnotu uloženou v registru *ICR_n*.

Pro správnou funkci je však důležité nastavit příslušné piny jako výstupní, aby byl PWM signál generován správně. To bylo zaručeno kódem:

```
//Nastavení pinů jako output na portu E
DDRE |= (1 << PE3); // Nastavení bitu PE3 => pin 5 jako output
DDRE |= (1 << PE4); // Nastavení bitu PE4 => pin 2 jako output
```

Stejný postup byl aplikován pro Timer 4, který také generuje PWM signál pro digitálně-analogové převodníky.

11.1.3 Čtení příchozího PWM signálu o nízké frekvenci

Příchozí PWM signál slouží ke generování PWM signálu z řídicí jednotky směrem k čerpadlům. PWM signál na čerpadlech určuje, jaké napětí, tedy i výkon se na čerpadle aktuálně vyskytuje. Při programování čtení příchozího PWM signálu byly využity *PCINT* (Pin change interrupt) piny. Pokud dojde ke změně stavu na těchto pinech, buď z logické nuly na logickou jedničku, či naopak, dojde k vygenerování přerušení, které má svou příslušnou obsluhu *ISR*.

Signál s nižší frekvencí (třeba 5 Hz) je například generován modulem OS-1401, který není schopen poskytnout vysoké frekvence výstupního PWM signálu. Nižší frekvence by však mohly být v podstatě nastavitelné na jakémkoliv zařízení, které poskytuje možnost ladění výstupního PWM signálu.

Při hledání *PCINT* pinů bylo snahou využít takové piny, které jsou umístěny na portu, který ještě neplní žádnou funkci, například negeneruje PWM. Tedy aby přerušení nebylo generováno i v moment, kdy řídicí jednotka vysílá PWM signál směrem k čerpadlům. Pro čtení příchozího PWM o nízké frekvenci byl využit port K na čipu ATmega2560. Byl použit následující kód:

```
DDRK &= ~(1 << PK0); // Vyčištění bitu PK0 => pin A8 jako input
DDRK &= ~(1 << PK1); // Vyčištění bitu PK1 => pin A9 jako input
DDRK &= ~(1 << PK2); // Vyčištění bitu PK2 => pin A10 jako input
DDRK &= ~(1 << PK3); // Vyčištění bitu PK3 => pin A11 jako input

lastPortStateK = PINK; // Uložení počátečního stavu portuK

PCICR |= (1 << PCIE2); // Povolí přerušení pro skupinu PCINT2 (port
K)

// Povolení interrupt jen pro bity PK0-PK3 (Analogové piny A8-A11)
PCMSK2 |= (1 << PCINT16) | (1 << PCINT17) | (1 << PCINT18) | (1 <<
PCINT19);
```

Tímto kódem byly nastaveny vybrané piny pro čtení PWM signálu jako vstupní a uložení aktuálního stavu na portu K. Dále bylo prostřednictvím *PCMSK2* (Pin Change Mask Register 2) registru nastaveno, které piny budou reagovat na změnu stavu.

Následně byla naprogramována obsluha přerušení pro příslušný port:

```
ISR(PCINT2_vect) {
    // Uložení aktuálního stavu na PortuK
    uint8_t current_StateK = PINK;
    // Porovnání předchozího a aktuálního stavu PortuK pomocí XOR operace
    uint8_t changedK = current_StateK ^ lastPortStateK;
    // Uložení času při provedení přerušení (detekovala se změna na jednom
    z pinů na portu K)
    unsigned long current_timeK = micros();

    // Pokud se změnil stav na pinu A8 (bit K0 na portu K)
    if (changedK & (1 << PK0)) {
```

```

// Pokud byl poslední stav na pinu A8 LOW a nyní je na pinu stav
HIGH, vypočti pulse_Period1, nastav flag newPulse1 na TRUE;
if ((current_StateK & (1 << PK0)) && !(lastPortStateK & (1 << PK0)))
{
    // Při úplně první hraně nemá last_RisingEdge1 žádnou hodnotu,
    perioda se tedy vypočte od druhé náběžné hrany
    if (last_RisingEdge1 != 0) {
        pulse_Period1 = current_timeK - last_RisingEdge1;
        newPulse1 = true;
    }
    last_RisingEdge1 = current_timeK; //uložení času poslední
    příchodí náběžné hrany
}
//Pokud je na pinu A8 nyní stav LOW a předchozí stav byl HIGH
else if (!(current_StateK & (1 << PK0)) && (lastPortStateK & (1 <<
PK0))) {
    pulse_HighTime1 = current_timeK - last_RisingEdge1; // Vypočti
    dobu, po kterou byl na pinu stav HIGH a ulož ji do proměnné
    pulse_HighTime1
}
}
}

```

Tento úsek kódu při každém přerušení na portu K uloží jeho aktuální stav. Tento stav je následně porovnán s předchozím stavem portu K pomocí logické operace XOR. Výsledek této operace je uložen do proměnné. Tato proměnná obsahuje logickou jedničku na každé pozici, kde došlo ke změně oproti předchozímu stavu. Obsah této proměnné je následně porovnán s jednotlivými bity na portu K. Pokud se zjistí, že došlo ke změně na příslušném bitu, zjišťuje se, jestli změna byla ze stavu logická nula do stavu logická jedna, nebo naopak. Pokud je aktuální stav příslušného bitu logická jedna a přechodí stav byl logická nula a zároveň již někdy byla zaznamenána nástupní hrana PWM signálu, vypočte se perioda příchodího PWM signálu a nastaví se příznak. Pokud došlo k přechodu ze stavu logická jedna do stavu logická nula, je vypočteno trvání stavu logická jedna příchodího PWM signálu.

Údaje o délce stavu logická jedna a o periodě příchodího signálu jsou poté zpracovány v hlavní smyčce programy, kde je z těchto údajů následně vypočítána střída PWM signálu. Následně je z těchto znalostí generováno soft PWM, které má vyšší frekvenci, než příchodí PWM signál a čerpadlo má tak plynulé otáčky.

11.1.4 Čtení příchozího PWM signálu o vyšší frekvenci

Vyšší frekvenci vstupního PWM signálu je schopna generovat například technologická karta, PLC s modulem IC-1701 a mikropočítač. Aby byla řídicí jednotka schopna takové signály zpracovat a spolehlivě je přepisovat na své výstupní piny, bylo pro tyto účely využito přímé přepisování aktuálního stavu na vstupních pinech na příslušné výstupní piny.

V této části byl použit obdobný postup (viz 11.1.3) s tím rozdílem, že generování PWM s identickou střídou, jakou má vstupní PWM signál, není zajištěno výpočty zpracovávajících délku času jednotlivých stavů. Zde bylo využito porovnání předchozího stavu jednotlivých bitů celého portu s jeho aktuálním stavem pomocí logické operace *XOR*. Pro tyto účely byl prvně použit tento úsek kódu:

```
DDRJ &= ~(1 << PJ0); // Vyčištění bitu PJ0 => pin 15 jako input
DDRJ &= ~(1 << PJ1); // Vyčištění bitu PJ1 => pin 14 jako input
DDRB &= ~(1 << PB4); // Vyčištění bitu PB4 => pin 10 jako input
DDRB &= ~(1 << PB5); // Vyčištění bitu PB5 => pin 11 jako input
DDRA |= (1 << PA0); // Nastavení bitu PA0 => pin 22 jako output
DDRA |= (1 << PA1); // Nastavení bitu PA1 => pin 23 jako output
DDRA |= (1 << PA2); // Nastavení bitu PA2 => pin 24 jako output
DDRA |= (1 << PA3); // Nastavení bitu PA3 => pin 25 jako output
lastPortStateJ = PINJ;
lastPortStateB = PINB;

// Povolení pin change interrupt pro PORTB (piny 0-7)
PCICR |= (1 << PCIE0);
// Povolení pin change interrupt pro PORTJ (piny 8-15)
PCICR |= (1 << PCIE1);

// Povolení interrupt jen pro PB4 (digitální pin 10 - PCINT4) a pro PB5
// (digitální pin 11 - PCINT5)
PCMSK0 |= (1 << PCINT4) | (1 << PCINT5);

// Povolení interrupt jen pro PJ0 (digitální pin 15 - PCINT9) a pro PJ1
// (digitální pin 14 - PCINT10)
PCMSK1 |= (1 << PCINT9) | (1 << PCINT10);
```

PCINT piny zde byly vybrány na portu J a portu B. Výstupní piny byly vybrány na portu A. Každý port, na kterém byly nastaveny piny jako vstupní, má svou vlastní obslužnou rutinu přerušení. Jelikož jsou očekávány vyšší frekvence, bylo zde vhodné rozdělit vstupní *PCINT* piny pro čtení příchozího PWM signálu mezi dva porty. Tím dochází k zamezení možného vzniku opožděného přepisu aktuálního stavu, jelikož obslužné rutiny jsou vykonány rychleji. Pro účely přepisu aktuálního stavu na výstupní pin bylo použito:

```
ISR(PCINT1_vect) {  
    // Načti aktuální stav portu J  
    uint8_t currentStateJ = PINJ;  
    // Porovnání předchozího a aktuálního stavu PortuJ pomocí XOR operace  
    uint8_t changedJ = currentStateJ ^ lastPortStateJ;  
  
    // Pokud se změnil stav na pinu 15 (PJ0)  
    if (changedJ & (1 << PJ0)) {  
        // Podle aktuálního stavu pinu 15 (PJ0) nastav pin 24 (PA2)  
        if (currentStateJ & (1 << PJ0))  
            PORTA |= (1 << PA2); // Nastav pin 24 (PA2) na HIGH  
        else  
            PORTA &= ~(1 << PA2); // Nastav pin 24 (PA2) na LOW  
    }  
    // Aktualizuj uložený stav portu J  
    lastPortStateJ = currentStateJ;  
}
```

Stejná logika byla také použita na další příslušné piny portu a také pro port J a jeho vybrané piny. Za použití tohoto kódu v řídicí jednotce bylo dosaženo stejných výsledků, jako když čerpadla byla napřímo řízena vedlejším systémem. Tím pádem tenhle úsek kódu lze prohlásit za funkční a spolehlivý.

12 LINEARIZACE CHARAKTERISTIKY NAPĚTÍ NA ČERPADLE V ZÁVISLOSTI NA STŘÍDĚ

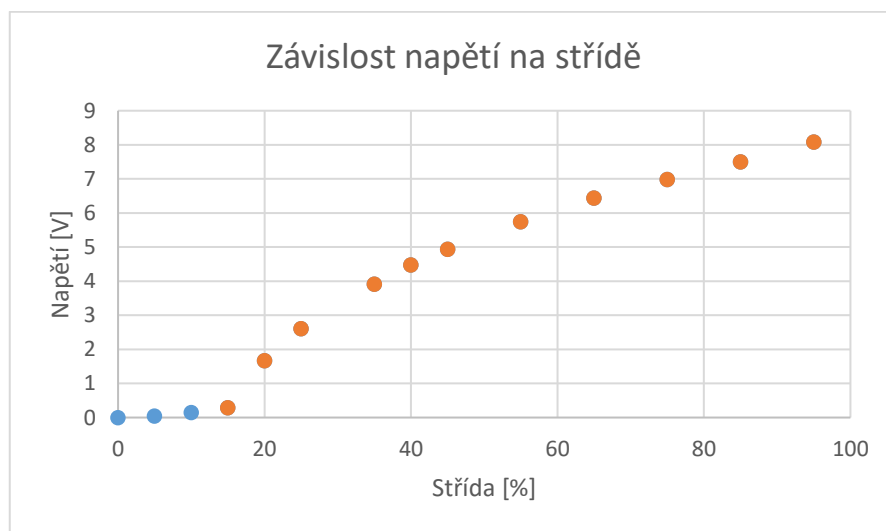
Napětí přivedené na čerpadla pomocí modulu L298N nevykazuje lineární charakteristiku. Tedy pokud je maximální napětí na čerpadle 9 V a na modul L298N je aplikována 50 % střída, uživatel by očekával napětí 4,5 V na výstupu modulu směrem k čerpadlu, ale není tomu tak.

Linearizace probíhala tak, že nejdříve musela být naměřena data, konkrétně jaké napětí se objevuje na svorkách čerpadla při různých hodnotách střídý.

Před samotným měřením však muselo být čerpadlo spuštěno čtyři minuty při střídě 85 %. Pokud by nebylo provedeno spuštění čerpadla po tuto dobu a měřilo se přímo, tak by se neshodovaly hodnoty napětí odebírané čerpadlem s hodnotami při laboratorním měření na modelu.

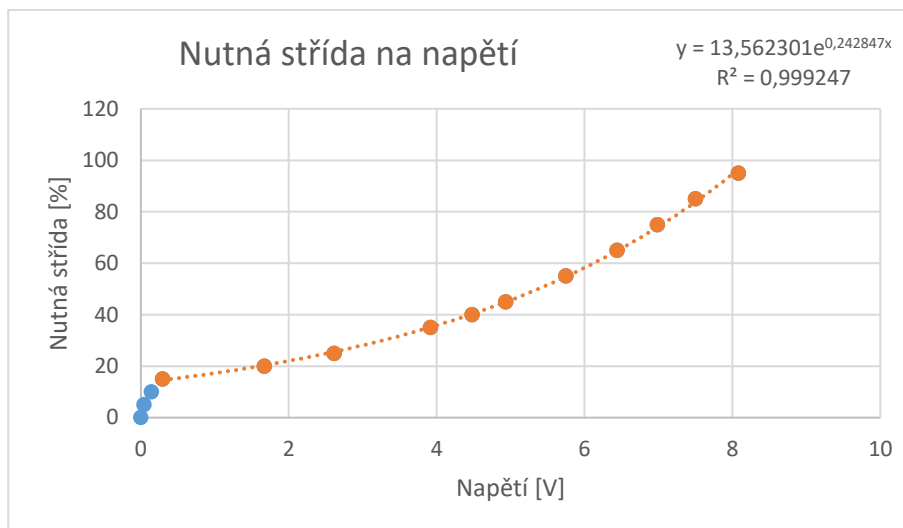
12.1 Práce s naměřenými daty v programu Microsoft Excel

Po správném postupu měření hodnot napětí na čerpadle byly tyto hodnoty zapsány do programu Excel, ve kterém byl vykreslen následující průběh



Obrázek 51: Závislost napětí na střídě

Bylo vypořádováno, že graf má logaritmický průběh, ale je možno si všimnout, že až od čtvrtého bodu v grafu. Z faktu, že graf měl od určité hodnoty logaritmický průběh, byly přehozeny osy grafu (transformace grafu) a byl tak získán exponenciální průběh



Obrázek 52: Závislost nutné aplikované střídě PWM signálu pro požadované napětí

Jak lze vidět, do grafu byla vložena exponenciální spojnice trendu, která slouží jako aproximační model, ale až od hodnot, kde graf vykazuje exponenciální průběh (oranžový průběh). Pro tento průběh byla vygenerována regresní rovnice, kterou lze následně aplikovat v programech pro vedlejší systémy.

Výše uvedená data platí pro první čerpadlo. Pro všechna ostatní čerpadla byl proveden stejný postup.

12.2 Aplikace rovnice v kódu pro vedlejší systém

Pokud je vedlejším řídicím systémem mikropočítač, například Arduino Uno, aplikace rovnice v programovém kódu by vypadala

```
OCR1A = (13.562301 * exp(0.242847 * akcni_zasah_cerpadlo1)) * 10.0;
```

kde *akcni_zasah_cerpadlo1* představuje proměnnou, ve které je uložena hodnota napětí, jež má být přivedena na svorky čerpadla. Do registru *OCR1A*, což je výstupní registr časovače 1 (Timer 1) mikrokontroléru ATmega328P, je následně uložena hodnota, která udává střidu výstupního PWM signálu.

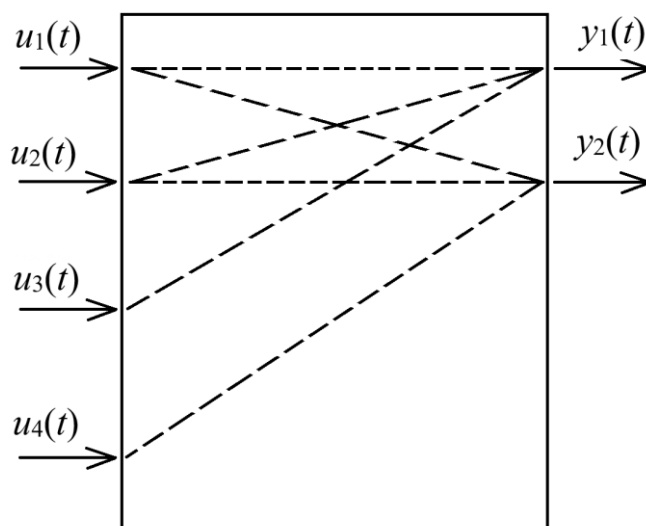
Pomocí výše uvedené rovnice lze tedy generovat PWM signál, který kompenzuje nelineární odezvu čerpadla a je tak umožněno plynulé řízení výkonu čerpadla. Tento kód platí pro první čerpadlo, analogicky jej však lze použít i pro ostatní čerpadla pomocí upravení koeficientů podle jejich specifické charakteristiky.

Program byl naprogramován tak, aby uživatel zadával pouze hodnoty v rozmezí (0-100) a tato hodnota byla dále převedena na napětí na čerpadle. Při testech laboratorního modelu bylo zjištěno, že pro lepší výsledky měření je zapotřebí, aby čerpadla měla v sobě neustále nasátou destilovanou vodu. Bylo zjištěno, že k tomuto dochází při aplikování 25 % střídý PWM signálu na čerpadlo. Tedy pokud uživatel zadá do sériové linky hodnotu 0, tak na modul pro řízení výkonu čerpadel bude aplikována 25 % střída PWM signálu. Pokud uživatel zadá hodnotu 100 do sériové linky, tak se na můstcích pro ovládání čerpadel vyskytuje 95 % hodnota střídý PWM signálu.

13 MĚŘENÍ STATICKÝCH A DYNAMICKÝCH CHARAKTERISTIK

Během měření statických a dynamických charakteristik bylo provedeno několik úprav. Jednalo se například o změnu pozice ventilů, úprava držáků pro senzory a doladování přívodu kapaliny do nádrží tak, aby nevířila hladinu. Hladina tím pádem zůstávala klidná a rovnoměrná a senzory tak odečítaly co nejpřesnější hodnotu. Dále bylo pro měření statických a dynamických charakteristik jako vedlejší řídicí systém použito Arduino Uno, pro které byl naprogramován příslušný program, aby uživatel jen zadával hodnoty střidy, které chce na moduly pro čerpadla aplikovat.

Laboratorní model čtyřválcové vodárny může být uvažován jako mnohorozměrový systém z důvodu toho, že průtok q_1 (u_1) ovlivňuje hladinu h_1 (y_1), ale také hladinu h_2 (y_2). To samé průtok q_2 (u_2) z čerpadla 2. Dále průtok q_3 (u_3) ovlivňuje pouze hladinu h_1 (y_1) a průtok q_4 (u_4) ovlivňuje pouze h_2 (y_2).



Obrázek 53: Vliv vstupních veličin $u_1(t)$, $u_2(t)$, $u_3(t)$, $u_4(t)$ na výstupy $y_1(t)$ a $y_2(t)$

13.1 Kód pro měření statických a dynamických charakteristik

Kód zajišťuje dvě hlavní činnosti, kterými je čtení příchozí hodnoty analogového napětí z digitálně-analogových převodníků a generuje PWM signálu, který vstupuje do řídicí jednotky.

Čtení analogového napětí je zajištěno analogovými piny, konkrétně se jedná o piny A0 - A3. Výstupní PWM signál je opět generován pomocí přerušování časovače, které již bylo jednou vysvětleno (viz 11.1.2). Zde byl použit obdobný postup, pouze s tím rozdílem, že Arduino Uno nabízí pouze jeden 16bitový časovač (Timer 1), který dokáže zajistit vyšší

rozlišení a přesnější šířku pulzu. Zbylé časovače (Timer 0 a Timer 2) jsou pouze 8bitové. Nakonec byl vybrán časovač 2 (Timer 2), jelikož časovač 0 (Timer 0) je vyhrazen pro funkce *millis()* a *delay()* a interní časování Arduina.

Funkce pro čtení vzdálenosti má následující tvar

```
void vypocti_vzdalenost1 () {  
    precteny_analog0 = analogRead(analog_pin0); //přečtení hodnoty na  
    analogovém pinu 0 (10bit)  
    napeti0=precteny_analog0*(5.0/1023.0); //převod hodnoty v rozsahu  
    10bit na napětí (0-5.0) V  
    vzdalenost_1=17.0-((napeti0/5.0)*(17.0)); //převod přečteného napětí  
    na vzdálenost v rozsahu (0-17) cm  
}
```

Tato funkce převádí čtené analogové napětí v rozmezí hodnot (0-1023) na napětí v rozmezí (0-5) V. Následně je toto napětí převedeno na vzdálenost v rozsahu (0-17) cm.

Uživatel zadává, kolik procent výkonu by chtěl aplikovat na čerpadlo. Tento údaj je uživatelem zadáván prostřednictvím sériové linky.

```
if((zadany_zasah_2 >=0 ) && (zadany_zasah_2 <=100)){  
    akcni_zasah_cerpadlo2=2.54+(zadany_zasah_2/100.0)*5.66;  
    OCR1B= (15.026661*exp(0.232970*akcni_zasah_cerpadlo2))*10.0;  
}
```

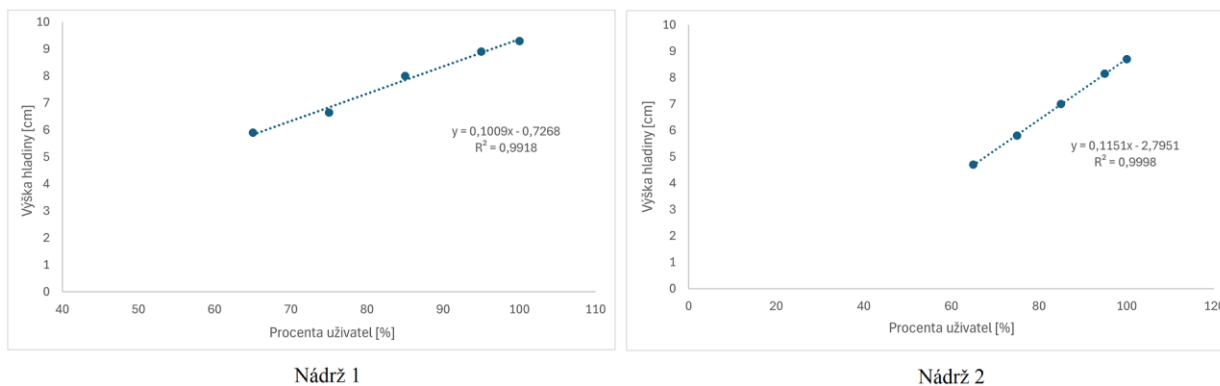
Tento úsek kódu zajišťuje, že čerpadlo jede vždy na nejnižší hodnotu, při které nasaje kapalinu (2,54 V). K této hodnotě je pak připočtena hodnota napětí, která byla přepočtena ze střídy zadané uživatelem. Výsledek je uložen do registru *OCR1B*, který řídí šířku pulzu na pinu 10 (*OC1B*), z něhož PWM signál vstupuje do řídicí jednotky a z ní do modulu L298N, který řídí výkon čerpadla 2. Zároveň je ošetřena situace, kdy by uživatel zadal vyšší hodnotu než jakou lze na čerpadla aplikovat.

13.2 Naměřená data a nasimulovaná data

13.2.1 Měření statických charakteristik

Při měření statických charakteristik bylo postupováno tak, že se do sériové linky mikropočítače zadávala uživatelská procenta, podle nichž byl generován příslušný PWM signál. Následně se čekalo, dokud se hladiny v jednotlivých nádržích neustálily. Tento postup byl opakován pro několik hodnot uživatelských procent pro modul čerpadla 1.

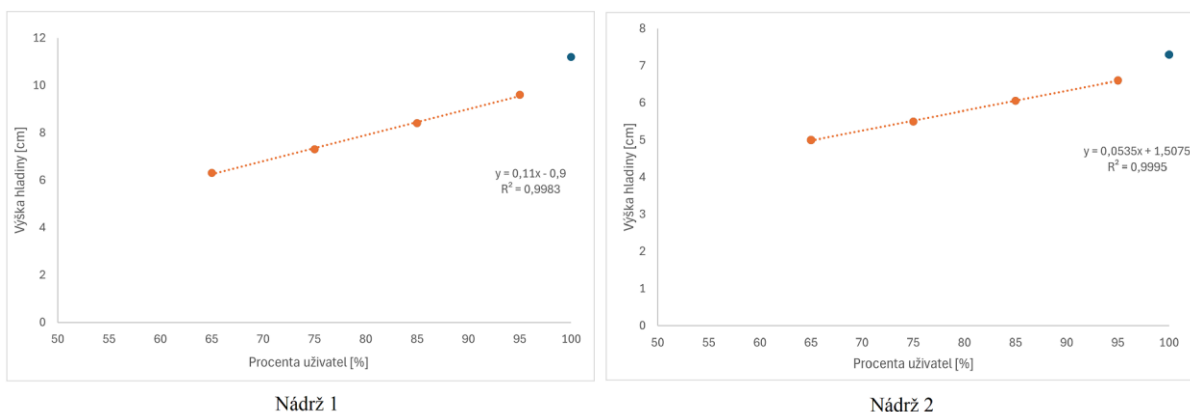
K ustálení hladin dochází díky samoregulaci (viz 4.3.3), při níž dochází k vyrovnání objemových průtoků, které do nádrže vstupují s objemovým průtokem, který z nádrže vystupuje. Bylo zjištěno, že výška hladiny roste nejlineárněji od hodnoty 65 % na čerpadle 1 a při 70 % na čerpadle 2. Totéž platí i obráceně. Od těchto hodnot totiž čerpadla spolehlivě dopraví kapalinu i do vyšších nádrží. Pro čerpadlo 1 má naměřená statická charakteristika následující tvar



Obrázek 54: Naměřená statická charakteristika pro čerpadlo 1

Z výše uvedeného obrázku jasně plyne, že zvýšením uživatelských procent, které mění střidu aplikovanou na modul pro řízení výkonu na čerpadle 1, je dosažen růst výšky hladiny v obou spodních nádržích. Je to z toho důvodu, že čerpadlo 1 dopravuje tekutinu do nádrže 1 a nádrže 4, ze které poté teče do nádrže 2 (viz Obrázek 24).

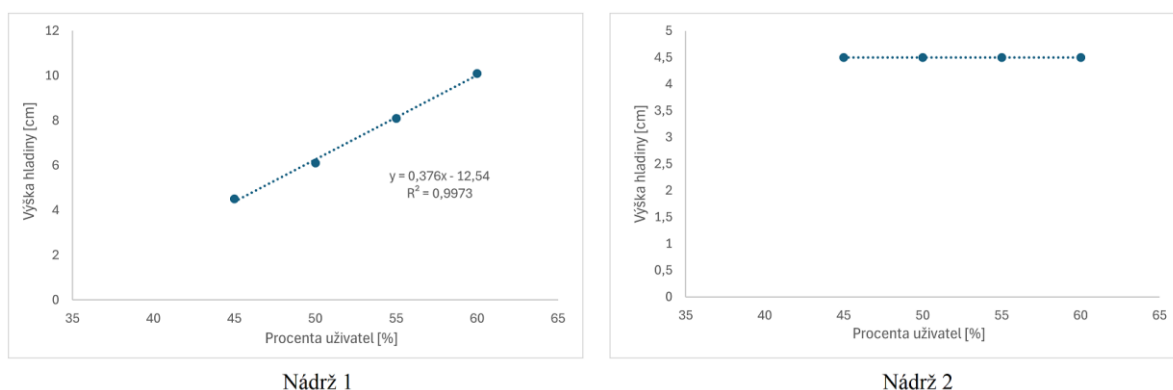
Následně byla změřena statická charakteristika pro čerpadlo 2, která vykazuje obdobné chování



Obrázek 55: Naměřená statická charakteristika pro čerpadlo 2

Lze si všimnout, že pokud by byla brána v potaz oblast uživatelských procent, jako u čerpadla 1, tak by se nezískal tak lineární průběh. Oblast, která vykazuje nejlineárnější chování, byla označena oranžovou barvou.

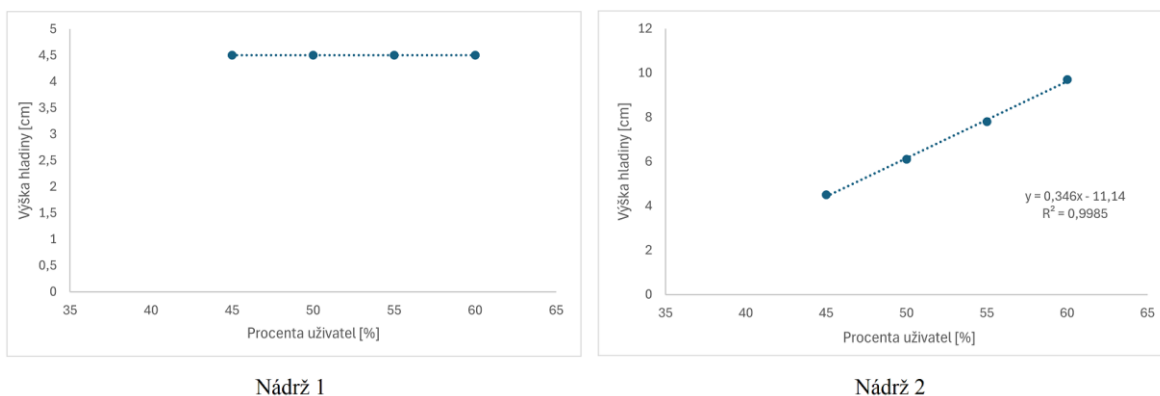
Dále byla změřena statická charakteristika pro čerpadlo 3, která má následující tvar



Obrázek 56: Naměřená statická charakteristika pro čerpadlo 3

Ze statické charakteristiky uvedené výše (levá část je pro nádrž 1 a pravá pro nádrž 2) je zřejmé, že nárůst hladiny v nádrži 1 s uživatelskými procenty opět vykazuje lineární charakteristiku. Hladina v nádrži 2 je konstantní, jelikož se zde nemění objemový průtok, který do nádrže vstupuje a který z ní vystupuje.

Nakonec byla zobrazena statická charakteristika pro čerpadlo 4, která vykazuje následující tvar



Obrázek 57: Naměřená statická charakteristika pro čerpadlo 4

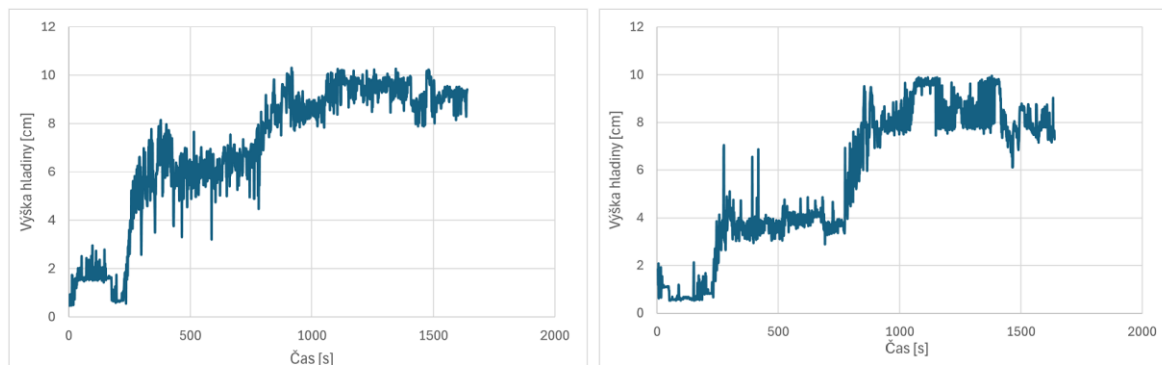
Na základě těchto lineárních charakteristik byly následně vybrány pracovní body pro každé z čerpadel a byly změřeny dynamické charakteristiky.

13.2.2 Měření dynamických charakteristik

Cílem měření dynamických charakteristik bylo zjistit, jak systém reaguje na skokovou změnu řídicího signálu, konkrétně procenta výkonu zadaných uživatelem.

Výška hladiny v nádržích byla nejprve uvedena do pracovního bodu, tedy do bodu, od kterého statická charakteristika vykazovala nejlineárnější chování. Následně došlo ke

skokové změně střidy PWM signálu zadaného uživatelem a čekalo se, než se výška hladiny v nádržích ustálí. Po celou dobu tohoto měření byla zaznamenáván časový průběh výšky hladiny.



Nádrž 1

Nádrž 2

Obrázek 58: Naměřené dynamické odezvy na změnu uživatelských procent pro čerpadlo 1

Z průběhů dynamické odezvy na obrázcích uvedených výše je zřejmé, že ultrazvukové senzory nevracejí přesné hodnoty i přesto, že výsledek, který se dostane z řídicího systému do vnějšího, je filtrován přes výpočet mediánu pěti měřených hodnot. Kolísání hodnot může být ovlivněno hned několika faktory, které se vyskytují během měření. Jedním z nich je tvorba bublin na hladině, které se náhodně pohybují po nádrži. Během měření docházelo k tomu, že vytvořené bublina občas procházely nějakou dobu přímo pod senzorem a ten tedy vrátil nežádoucí hodnotu. Dalším faktorem, který měření mohl ovlivnit, je víření hladiny vody. Toto víření je způsobeno zejména tehdy, pokud do nádrže vtékají vyšší objemové průtoky kapaliny. Toto víření hladiny se však zmenšuje, když hladina v nádrži stoupá blíže k hadici, ze které do nádrže kapalina vstupuje. Posledním faktorem mohou být samotné digitálně-analogové převodníky, které právě údaj o vzdálenosti hladiny poskytují vnějšímu systému prostřednictvím analogového napětí. Toto analogové napětí nemusí přesně odpovídat vzdálenosti, jelikož tyto převodníky mají přípustnou chybu, která je udávána výrobcem (viz 10.1.2).

Pro měření dynamických charakteristik byl vytvořen program pro Arduino Uno, ve kterém uživatel zadával procenta výkonu, které chce aplikovat na čerpadlo. Zadávání procent výkonu uživatelem probíhalo pomocí sériové linky. Zároveň se na sériovou linku vypisovaly údaje o čase, procentech zadaných uživatelem na čerpadla a údaje o výšce hladiny v nádrži 1 a 2. Na obrázku níže je uveden příklad výstupu na sériové lince po čas měření

670,65,70,5.97,4.23
 671,65,70,6.27,4.11
 672,65,70,6.00,4.11
 673,65,70,6.55,4.26
 674,65,70,6.55,4.05
 675,65,70,6.66,4.42
 676,65,70,6.48,3.82
 677,65,70,6.63,4.18
 678,65,70,6.50,4.24

Obrázek 59: Příklad výpisu dat na sériovou linku počas měření dynamické odezvy na čerpadle 1

S vypsányi daty na sériové lince šlo pracovat tak, že byly zkopírovány, poté vloženy do poznámkového bloku a uloženy do soubory. Dále šlo tato data ze souboru vložit do programu EXCEL prostřednictvím karty *Data*, aby s nimi šlo manipulovat a vykreslit tak průběh (viz **Chyba! Nenalezen zdroj odkazů.**).

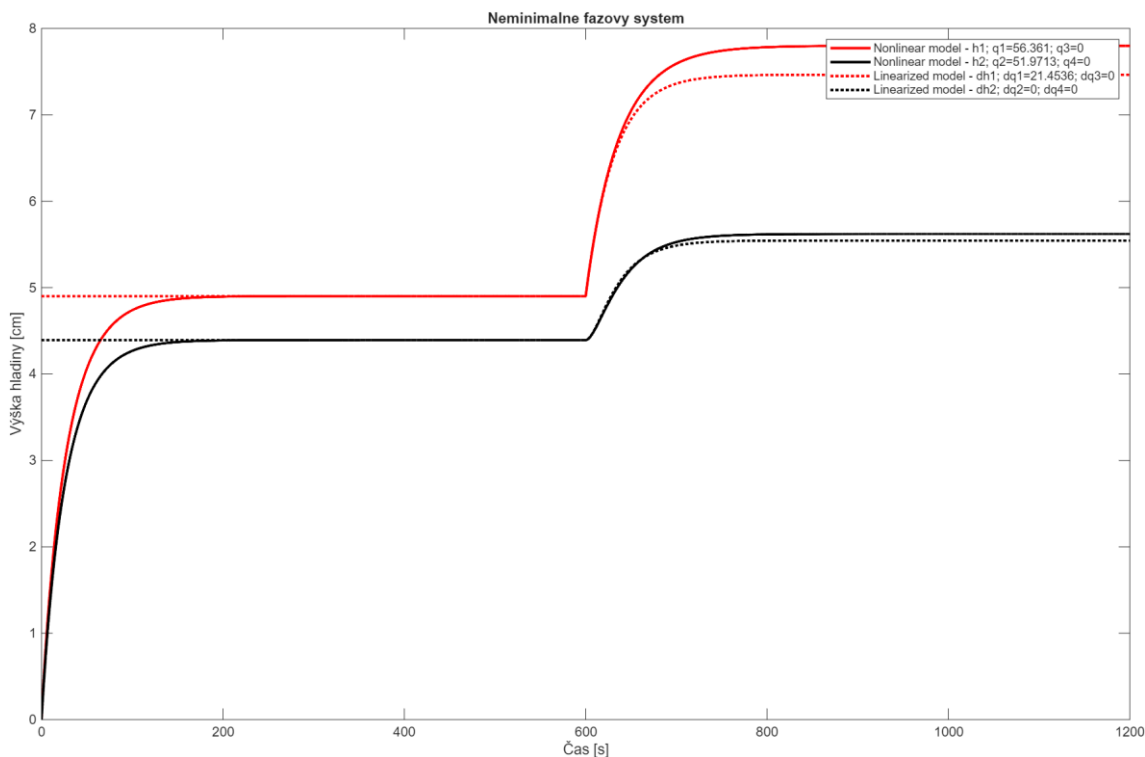
670	65	70	5,97	4,23
671	65	70	6,27	4,11
672	65	70	6	4,11
673	65	70	6,55	4,26
674	65	70	6,55	4,05
675	65	70	6,66	4,42
676	65	70	6,48	3,82
677	65	70	6,63	4,18
678	65	70	6,5	4,24

Obrázek 60: Zobrazení dat v programu EXCEL

13.2.3 Simulace a získání přenosové matice

Nejprve byly na reálném laboratorním modelu naměřeny ustálené hodnoty průtoků q_1 a q_2 , které byly zadány do proměnných q_1 a q_2 . Pro dynamickou odezvu byla na čerpadle 1 a čerpadle 2 provedena skoková změna a opět změřena hodnota průtoků. Tyto hodnoty průtoků byly následně zadány v programu MATLAB do příslušných proměnných q_i , která má v sobě uloženou hodnotu průtoků v pracovním bodě a dq_i , která v sobě obsahuje uloženou hodnotu přírůstek průtoků po skokové změně z pracovního bodu. Byly také nastaveny hodnoty průřezu nádrží, průřezu průtokového otvoru z nich a poměr ventilu.

V programu MATLAB byly zobrazeny průběhy pro nelineární matematický model a také pro linearizovaný matematický model.



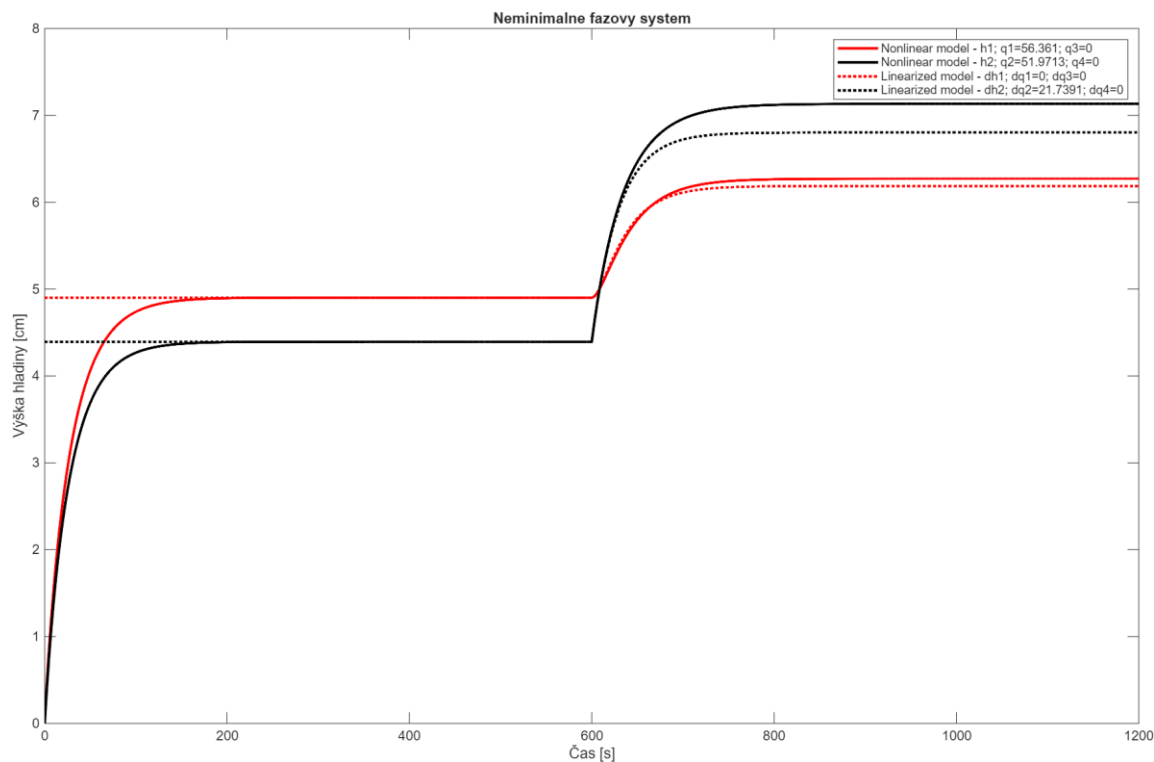
Obrázek 61: Průběhy matematických modelů v programu MATLAB po zadání hodnot změřených při dynamické změně pro čerpadlo 1 [18]

Linearizovaný průběh je rovnou posunut do pracovního bodu a až od něj je pak provedena změna skoková změna. Z průběhu výše lze vidět, že jsou patrné odlišnosti průběhů výšky hladiny v čase pro nelinearizovaný matematický model a pro matematický model linearizovaný.

Z takovéto simulované dynamické odezvy lze určit přenosovou matici $G_S(s)$, která je definována jako

$$G_S(s) = \begin{pmatrix} \frac{\Delta h_1}{\Delta q_1} & \frac{\Delta h_1}{\Delta q_2} \\ \frac{\Delta h_2}{\Delta q_1} & \frac{\Delta h_2}{\Delta q_2} \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \quad (13.1)$$

Tato přenosová matice popisuje, jak se projevuje změna výšky hladiny h_1 a h_2 při změnách průtoku q_1 a q_2 . Je tedy potřeba ještě provést simulaci pro změnu průtoku q_2 , která má následující průběh. Provedla se tedy simulace i pro přírůstek průtoku q_2 a z ní byl získán následující průběh

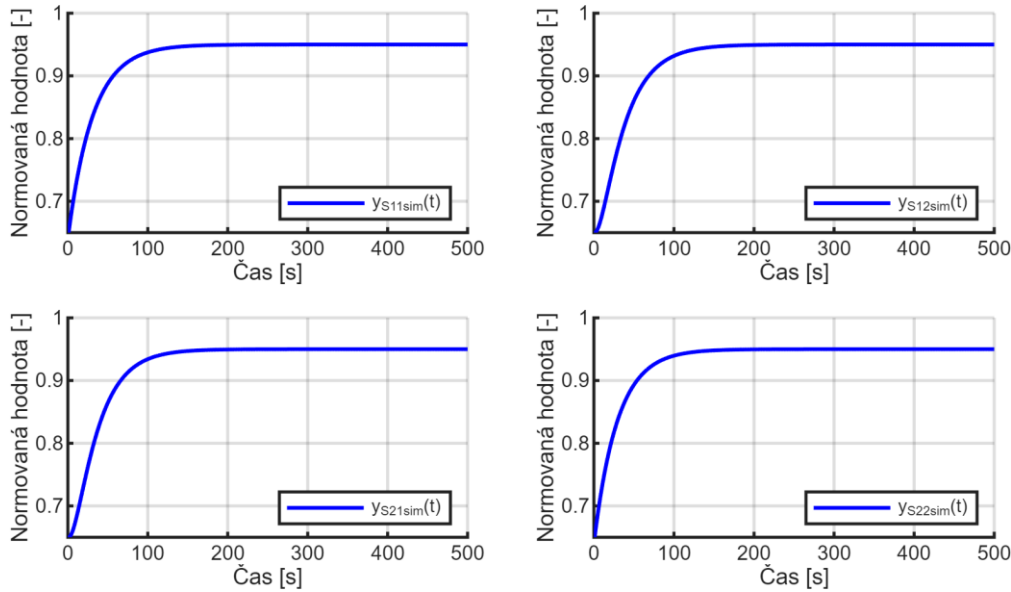


Obrázek 62: Průběhy matematických modelů v programu MATLAB po zadání hodnot změřených při dynamické změně pro čerpadlo 2 [26]

Pro tyto dynamické odezvy je přenosová matice $G_S(s)$

$$G_S(s) = \begin{pmatrix} \frac{0,1194}{31,13s + 1} & \frac{0,05911}{303,7s^2 + 40,89s + 1} \\ \frac{0,0537}{299,2s^2 + 39,62s + 1} & \frac{0,1108}{29,47s + 1} \end{pmatrix} \quad (13.2)$$

Pro jednotlivé přenosy S_{11} , S_{12} , S_{21} a S_{22} získané ze simulace matematického modelu byly vykresleny odezvy na jednotkový skok



Obrázek 63: Odezva simulovaných systémů S_{11} , S_{12} , S_{21} , S_{22} na skokovou změnu

Obdobný postup byl proveden pro čerpadlo 3 a čerpadlo 4. Tedy byl nalezen pracovní bod, následně byly naměřeny průtoky q_1 , q_2 , q_3 , q_4 v pracovním bodě a poté byla přivedena skoková změna pouze na čerpadlo 3. Tím tedy byla získána hodnota dq_3 , která je dána rozdílem průtoku po skokové změně a před ní.

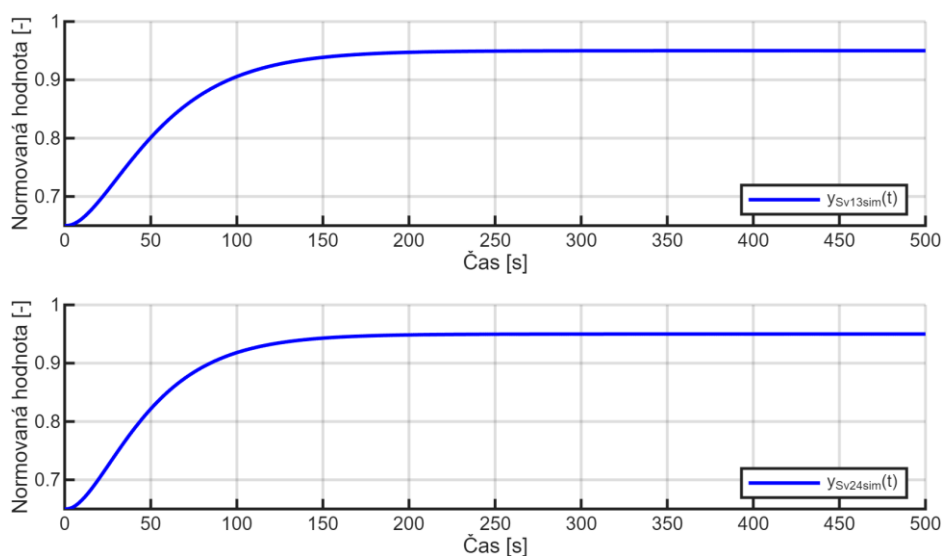
Maticový přenos však má jinou strukturu. V tomto případě se sleduje, jaký vliv má změna průtoku q_3 na výšku hladiny v nádrži 1 a jaký vliv má změna průtoku q_4 na výšku hladiny v nádrži 2 (viz Obrázek 24). Struktura přenosové matice poruch $G_{SV}(s)$ má tvar

$$G_{SV}(s) = \begin{pmatrix} \frac{\Delta h_1}{\Delta q_3} & \frac{\Delta h_1}{\Delta q_4} \\ \frac{\Delta h_2}{\Delta q_3} & \frac{\Delta h_2}{\Delta q_4} \end{pmatrix} = \begin{pmatrix} S_{13} & S_{14} \\ S_{23} & S_{24} \end{pmatrix} \quad (13.3)$$

Jelikož změna průtoku q_3 nemá vliv na změnu výšky hladiny v nádrži 1 a změna průtoku q_4 neovlivňuje výšku hladiny v nádrži 2, tak přenosové matice poruch je

$$G_{SV}(s) = \begin{pmatrix} \frac{0,1894}{831,4s^2 + 58,31s + 1} & 0 \\ 0 & \frac{0,1795}{628s^2 + 51,52s + 1} \end{pmatrix} \quad (13.4)$$

Kde podobně jako pro přenosovou matici $G_S(s)$ byly opět vykresleny odezvy jednotlivých přenosů v matici poruch $G_{SV}(s)$, konkrétně S_{13} a S_{24} , jelikož zbylé přenosy jsou nulové



Obrázek 64: Odezva simulovaných systémů S_{13} a S_{24} na skokovou změnu

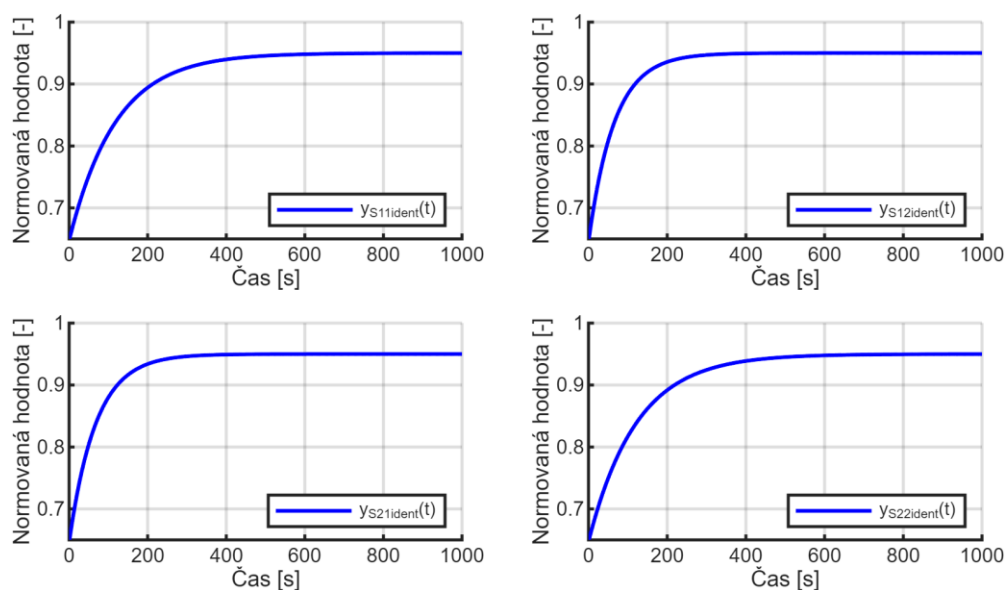
13.2.4 Identifikace laboratorního modelu

Na laboratorním modelu bylo provedeno měření dynamických charakteristik pro všechna čerpadla. Na základě naměřených vstupně-výstupních dat byla určena přenosová matice řízení $G_S(s)$ a přenosová matice poruch $G_{SV}(s)$. Pro účely identifikace bylo použito grafické rozhraní System Identification App, do něhož byla data z provedených měření naimportována. Možnosti práce a zpracování dat byly popsány v práci výše (viz 2.3).

Přenosová matice $G_S(s)$ byla určena jako

$$G_S(s) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} = \begin{pmatrix} \frac{0,007731}{s + 0,008454} & \frac{64,06}{s^2 + 4949s + 75,58} \\ \frac{2,947}{s^2 + 234,5s + 3,449} & \frac{0,007363}{s + 0,008236} \end{pmatrix} \quad (13.5)$$

Pro matici $\mathbf{G}_S(s)$ získanou identifikací byly rovněž vykresleny odezvy jednotlivých systémů na skokovou změnu

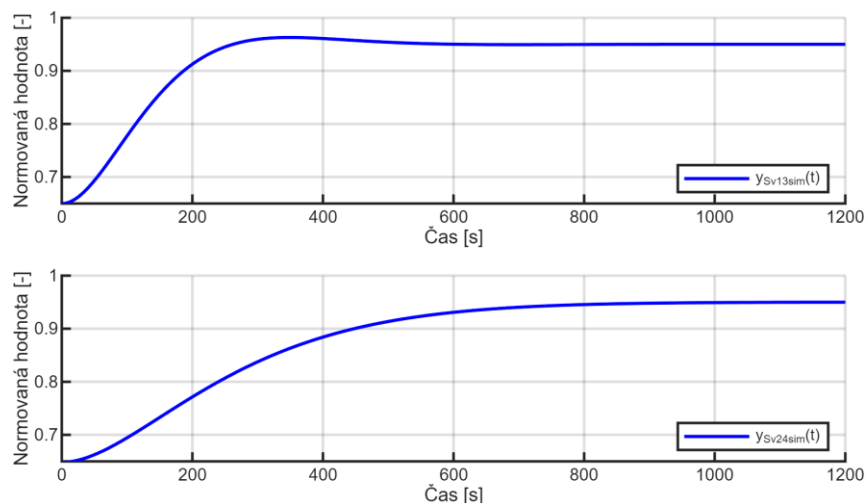


Obrázek 65: Odezva identifikovaných systémů S_{11} , S_{12} , S_{21} , S_{22} na skokovou změnu

Dále byla také identifikována matice poruch $\mathbf{G}_{SV}(s)$

$$\begin{aligned} \mathbf{G}_{SV}(s) &= \begin{pmatrix} S_{13} & S_{14} \\ S_{23} & S_{24} \end{pmatrix} \\ &= \begin{pmatrix} \frac{0,0001236}{s^2 + 0,01807s + 0,000163} & 0 \\ 0 & \frac{0,0000319}{s^2 + 0,01286s + 0,0000464} \end{pmatrix} \end{aligned} \quad (13.6)$$

Průběhy odezev na skokovou změnu jednotlivých systémů v matici $\mathbf{G}_{SV}(s)$ získanou identifikací mají tvar



Obrázek 66: Odezva identifikovaných systémů S_{13} a S_{24} na skokovou odezvu

13.3 Zhodnocení získaných výsledků

Z výše uvedených obrázků a přenosových matic lze konstatovat, že se naměřená data liší od dat získaných simulací. Tento rozdíl může být způsobem hned několika faktory.

Mezi hlavní příčiny lze zařadit například výtlačnou výšku, do které muselo čerpadlo kapalinu dopravovat. Naměřená data se dále mohou od těch získaných simulací odlišovat i vlivem hydraulických odporů, které způsobují energetické ztráty proudící kapaliny. Dále, jak je patrné z průběhů změřených dynamických charakteristik pro čerpadlo 1 (viz Obrázek 58), tak ultrazvukové senzory nevracely příliš stabilní hodnoty během měření, což mohla zapříčinit výše popsaná tvorba bublinek a víření kapaliny v nádrži (viz 13.2.2).

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout, sestavit a ověřit funkčnost laboratorního modelu čtyřválcové vodárny, který umožňuje praktické testování řízení systémů s více vstupy a výstupy (MIMO). Zároveň byl navržen tak, aby byl připojitelný k více vnějším systémům (PLC, mikropočítač, technologická karta PCI-1711).

V teoretické části byly popsány a vysvětleny pojmy z oblastí teorie systémů, matematického modelování a mechaniky tekutin. Z těchto poznatků poté vycházel matematický model laboratorního modelu čtyřválcové vodárny, který byl následně v praktické části linearizován a převeden do přenosových funkcí, čímž bylo umožněno jeho porovnání se simulací a měřením.

Praktická část byla zaměřena na samotnou konstrukci modelu, výběr vhodných komponent a ověření jejich funkčnosti, návrh elektronického propojení a následné programování logiky řídicí jednotky laboratorního modelu. Dále byly popsány metody převodu měřených hodnot výšky hladiny na analogový signál pomocí převodníků PWM na analogové napětí v rozmezí (0-10) V. Zároveň byl kladen důraz na vysvětlení logiky programu řídicí jednotky, konkrétně na přepisování vstupních PWM signálů na výstupní PWM signály směrem k modulům pro řízení výkonu na čerpadlech. Také byl vysvětlen postup při výměně vnější řídicí jednotky, včetně použití příslušných propojovacích krabic. Byly také popsány možné změny konfigurace laboratorního modelu, které vedou k odlišnému matematickému popisu systému.

Během experimentální části bylo zjištěno, že reálný systém se v některých aspektech odchyluje od chování matematického modelu. Tento rozdíl mohl být způsoben hydraulickými ztrátami, nestabilitou vracených hodnot ultrazvukovými senzory a dynamickými jevy v nádržích. Tyto odchylky byly analyzovány a diskutovány včetně jejich možných příčin.

Výsledný model je plně funkční a umožňuje demonstraci řízení MIMO systému, měření statických a dynamických charakteristik a identifikaci systému.

Z pohledu zadání práce byly splněny všechny požadavky – od návrhu modelu, přes jeho realizaci, až po ověření chování prostřednictvím měření. Výsledný model tak může být implementován ve výuce a sloužit jako nástroj pro měření a ověření pojmů z oblasti teorie řízení.

SEZNAM POUŽITÉ LITERATURY

- [1] BALÁTĚ, Jaroslav. *Automatické řízení*. Praha: Ben, 2003. ISBN 80-730-0020-2. Dostupné také z: <http://krameriusndk.nkp.cz/search/handle/uuid:0f4d0930-cbee-11e5-a590-005056827e51>.
- [2] PROKOP, Roman; MATUŠŮ, Radek a PROKOPOVÁ, Zdenka. *Teorie automatického řízení - lineární spojitě dynamické systémy*. Učební texty vysokých škol. Zlín: Univerzita Tomáše Bati ve Zlíně, 2006. ISBN 80-731-8369-2. Dostupné také z: <http://krameriusndk.nkp.cz/search/handle/uuid:3eeb95b0-2fcc-11e7-a38c-005056827e51>.
- [3] BOBÁL, Vladimír. *Identifikace systémů*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009. ISBN 978-80-7318-888-7.
- [4] KADLČÍK, Ondřej. *Řízení mnohorozměrového laboratorního modelu*. Diplomová práce, vedoucí Navrátil, Pavel. Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav automatizace a řídicí techniky, 2014. Dostupné také z: <http://hdl.handle.net/10563/30228>.
- [5] MATHWORKS. *Working with System Identification App*. Online. 2024. Dostupné z: <https://www.mathworks.com/help/ident/ug/working-with-the-system-identification-tool-gui.html>. [cit. 2025-04-08].
- [6] NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. Ostrava: Montanex, 1999. ISBN 80-722-5030-2.
- [7] WAGNEROVÁ, Renata. *Základy automatického řízení*. Online. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2013. ISBN 978-80-248-3054-4. Dostupné z: https://projekty.fs.vsb.cz/463/edubase/VY_01_031/Z%C3%A1klady%20automatického%20řízení%20%C5%99%C3%ADzen%C3%AD%20-%20teoretického%20z%C3%A1klad.pdf. [cit. 2025-05-10].
- [8] JANALÍK, Jaroslav. *Hydrodynamika a hydrodynamické stroje*. Online. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2008. Dostupné z: <https://old.vscht.cz/uchi/ped/hydroplo/materialy/skripta.Janalik.pdf>. [cit. 2025-02-28].
- [9] *PLC Hardware: A Detailed Overview With Component Examples*. Online. 2023. Dostupné z: <https://ladderlogicworld.com/plc-hardware/>. [cit. 2025-02-23].
- [10] *Programovatelné logické automaty*. Online. 2002. Dostupné z: <https://uprt.vscht.cz/kminekm/mrt/F5/F5k53-PLC.htm#k53>. [cit. 2025-02-23].
- [11] *Introduction to Programmable Logic Controllers - Part I*. Online. 2020. Dostupné z: <https://www.pdhonline.com/courses/e116/PLC-module1.pdf>. [cit. 2025-02-23].
- [12] TECO A.S. *Základní modul CP-1015*. Online. 2020. Dostupné z: <https://wiki.tecomat.cz/clanek/276-cp-1015>. [cit. 2025-05-01].
- [13] ŠMEJKAL, Ladislav a MARTINÁSKOVÁ, Marie. *PLC a automatizace*. Praha: BEN - technická literatura, 1999. ISBN 80-860-5658-9.
- [14] *Arduino pod pokličkou: jak funguje mikrokontrolér*. Online. 2019. Dostupné z: <https://bastlirna.hwkitchen.cz/arduino-pod-poklickou-jak-funguje-mikrokontroler/>. [cit. 2025-02-24].
- [15] ARDUINO. *Official Arduino Store*. Online. 2025. Dostupné z: <https://store.arduino.cc/>. [cit. 2025-04-15].

- [16] SELECKÝ, Matuš. *Arduino: uživatelská příručka*. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
- [17] VODA, Zbyšek. *Průvodce světem Arduina*. Vydání druhé. Bučovice: Martin Stříž, 2017. ISBN 978-80-87106-93-8.
- [18] NAVRATIL, Pavel; PEKAR, Libor a MATUSU, Radek. Control of a Multivariable System Using Optimal Control Pairs: A Quadruple-Tank Process. Online. *IEEE Access*. 2020, roč. 8, s. 2537-2563. ISSN 2169-3536. Dostupné z: <https://doi.org/10.1109/ACCESS.2019.2962302>. [cit. 2025-05-25].
- [19] ZAKUCIA, Jozef. *Řízení čtyřválcové vodárny*. Diplomová práce. Praha: ČVUT, 2009.
- [20] JOHANSSON, K.H. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. Online. *IEEE Transactions on Control Systems Technology*. Roč. 8, č. 3, s. 456-465. ISSN 10636536. Dostupné z: <https://doi.org/10.1109/87.845876>. [cit. 2024-12-29].
- [21] AUTODESK. *Fusion 360*. Online. 2025. Dostupné z: <https://www.autodesk.com/products/fusion-360/overview>. [cit. 2025-04-20].
- [22] TECO A.S. *PLC Tecomat Foxtrot – základní moduly*. Online. 2018. Dostupné z: <https://www.tecomat.cz/modules/DownloadManager/download.php?alias=foxtrot-cz-cp-1005>. [cit. 2025-04-27].
- [23] TECO A.S. *OS-1401 Rozšiřující modul*. Online. 2021. Dostupné z: <https://catalog.tecomat.cz/produkt/os-1401#params>. [cit. 2025-04-27].
- [24] TECO A.S. *IC-1701 Rozšiřující modul*. Online. 2021. Dostupné z: <https://catalog.tecomat.cz/produkt/ic-1701#params>. [cit. 2025-04-27].
- [25] ADVANTECH CO., LTD. *ADVANTECH CO., LTD. PCI-1711/1731 User's Manual*. Online. 1999. Dostupné z: https://advdownload.advantech.com/productfile/Downloadfile3/1-FSMPSX/PCI-1711_v3.pdf. [cit. 2025-04-26].
- [26] CONRAD. *Návod k obsluze*. Online. 2014. Dostupné z: <https://asset.conrad.com/media10/add/160267/c1/-/cs/000539090ML01/navod-k-obsluze-a-bezpecnostni-pokyny-539090-barwig-typ-04-0444-nizkonapetove-ponorne-cerpadlo-600-lh-6-m.pdf>. [cit. 2025-05-11].
- [27] *Arduino Timer Interrupts*. Online. 2010. Dostupné z: <https://www.instructables.com/Arduino-Timer-Interrupts/>. [cit. 2025-04-21].
- [28] ATMEL CORPORATION. *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V*. Online. 2014. Dostupné z: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf. [cit. 2025-04-25].

SEZNAM OBRÁZKŮ

Obrázek 1: System Identification App [5]	20
Obrázek 2: Rozmístění jednotlivých funkcí v dialogovém okně System Identification App [5]	20
Obrázek 3: Možnosti vložení dat v dialogovém okně System Identification [5]	21
Obrázek 4: Hmotný bod [6]	23
Obrázek 5: Pružina [6]	24
Obrázek 6: Tlumič [6]	24
Obrázek 7: Těleso připevněné k tuhému rámu [7]	25
Obrázek 8: Zapojení odporu [7]	26
Obrázek 9: Zapojení cívky [7]	27
Obrázek 10: Zapojení kondenzátoru [7]	27
Obrázek 11: Schéma děliče napětí [7]	27
Obrázek 12: Reynoldsův pokus [7]	31
Obrázek 13: Rozdíly rychlostních profilů v potrubí [8]	32
Obrázek 14: Výtok kapaliny z uzavřené nádrže s otvorem ve dně [6]	33
Obrázek 15: Volný výtok z otevřené nádrže [6]	34
Obrázek 16: Výtok z otevřené nádrže s přítokem [6]	35
Obrázek 17: Bloková struktura PLC [10]	37
Obrázek 18: Vstupy a výstupy Tecomat Foxtrot CP-1015 [12]	39
Obrázek 19: Arduino Uno Rev3 [15]	42
Obrázek 20: Blokové schéma mikrokontroléru ATmega328 [14]	43
Obrázek 21: Adresový prostor registrů a paměti SRAM čipu ATmega328 [14]	45
Obrázek 22: Arduino Nano [15]	47
Obrázek 23: Arduino Mega2560 Rev3 [15]	48
Obrázek 24: Schéma čtyřválcové vodárny	50
Obrázek 25: Jedna z možných změn struktury laboratorního modelu	59
Obrázek 26: Další možná změna struktury laboratorního modelu	60
Obrázek 27: První část návrhu řešení uchycení nádrží za použití hliníkových profilů [21]	62
Obrázek 28: Druhá část návrhu řešení uchycení nádrží za použití hliníkových profilů [21]	62
Obrázek 29: Návrh konstrukce za použití regálu z více pohledů [21]	63
Obrázek 30: Realizace konstrukce	64
Obrázek 31: Pohled shora a z boku na držák pro nádrže [21]	64
Obrázek 32: Finální podoba víka na nádrže [21]	65

Obrázek 33: Použité rozdělovací ventily	65
Obrázek 34: Modul OS-1401 a příklad jeho zapojení [20].....	68
Obrázek 35: Modul IC-1701 a příklad jeho zapojení [24].....	68
Obrázek 36: Rozmístění pinů vstupně-výstupního konektoru technologické karty PCI-1711 (levá část obrázku) a technologická karta PCI-1711 (pravá část obrázku) [25]	69
Obrázek 37: Umístění řídicí jednotky a step-down měničů napětí ve voděodolné krabici před kompletací zapojení	70
Obrázek 38: Ultrazvukový senzor HC-SR04.....	71
Obrázek 39: Paralelní zapojení výstupů modulu L298N	74
Obrázek 40: Ponorné čerpadlo Barwig Typ 04 0444.....	74
Obrázek 41: 3D model propojovací krabice	75
Obrázek 42: Znázornění použitého napěťového děliče	76
Obrázek 43: Propojovací krabice pro připojení PLC s modulem IC-1701	77
Obrázek 44: Blokové schéma propojení jednotlivých částí modelu čtyřválnové vodárny ..	78
Obrázek 45: Ukázka číslování pinů na konektoru Cannon 25.....	78
Obrázek 46: Umístění pinu na propojovací krabici pro PLC s modulem IC-1701.....	79
Obrázek 47: Možnosti generování PWM pro ATmega2560 [28]	84
Obrázek 48: Timer/Counter 3 Control Register A (TCCR3A) [28]	85
Obrázek 49: Struktura Timer/Counter 3 Control Register B (TCCR3B) [28].....	85
Obrázek 50: Compare Output Mode pro Phase and Frequency Correct PWM [28]	85
Obrázek 51: Závislost napětí na střídě.....	91
Obrázek 52: Závislost nutné aplikované střídě PWM signálu pro požadované napětí	92
Obrázek 53: Vliv vstupních veličin $u_1(t)$, $u_2(t)$, $u_3(t)$, $u_4(t)$ na výstupy $y_1(t)$ a $y_2(t)$	94
Obrázek 54: Naměřená statická charakteristika pro čerpadlo 1	96
Obrázek 55: Naměřená statická charakteristika pro čerpadlo 2.....	96
Obrázek 56: Naměřená statická charakteristika pro čerpadlo 3	97
Obrázek 57: Naměřená statická charakteristika pro čerpadlo 4.....	97
Obrázek 58: Naměřené dynamické odezvy na změnu uživatelských procent pro čerpadlo 1	98
Obrázek 59: Příklad výpisu dat na sériovou linku počas měření dynamické odezvy na čerpadle 1	99
Obrázek 60: Zobrazení dat v programu EXCEL	99
Obrázek 61: Průběhy matematických modelů v programu MATLAB po zadání hodnot změřených při dynamické změně pro čerpadlo 1 [18].....	100
Obrázek 62: Průběhy matematických modelů v programu MATLAB po zadání hodnot změřených při dynamické změně pro čerpadlo 2 [26].....	101
Obrázek 63: Odezva simulovaných systémů S_{11} , S_{12} , S_{21} , S_{22} na skokovou změnu.....	102

Obrázek 64: Odezva simulovaných systémů S_{13} a S_{24} na skokovou změnu	103
Obrázek 65: Odezva identifikovaných systémů S_{11} , S_{12} , S_{21} , S_{22} na skokovou změnu.....	104
Obrázek 66: Odezva identifikovaných systémů S_{13} a S_{24} na skokovou odezvu	104

SEZNAM TABULEK

Tabulka 1: Testování chyby převodu digitálně-analogových převodníků při frekvenci 500 Hz.....72

Tabulka 2: Testování chyby převodu digitálně-analogových převodníků při frekvenci 1000 Hz.....73

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

$h_i(t)$	Výška hladiny v i-té nádrži [m]
A_i	Plocha průřezu i-té nádrže [m ²]
a_i	Plocha výstupního otvoru z i-té nádrže [m ²]
g	Gravitační zrychlení [m/s ²]
a	Zrychlení [m/s ²]
q_1, q_2, q_3, q_4	Průtoky vody z jednotlivých čerpadel [m ³ /s]
F	Síla [N]
$i(t)$	Elektrický proud v čase [A]
$u(t)$	Elektrické napětí v čase [V]
L	Indukčnost [H]
C	Kapacita kondenzátoru [F]
Z_1, Z_2	Impedance v obvodu [Ω]
s	Komplexní proměnná u obrazu v Laplaceově transformaci
$\mathbf{G}_s(s)$	Matice přenosu soustavy
$\mathbf{G}_{sv}(s)$	Matice přenosu poruch
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$	Matice stavového popisu systému
T_i	Časová konstanta systému [s]
v_i	Napětí přivedené na čerpadlo [V]
p_i	Procentuální hodnota nastavená uživatelem [%]
k_i	Zisk čerpadla [cm ³ /Vs]
MIMO	Multiple Input Multiple Output (systém s více vstupy a výstupy)
GND	Ground (uzemnění)
PLC	Programovatelný logický automat (Programmable Logic Controller)
PWM	Pulzně šířková modulace (Pulse Width Modulation)
CPU	Centrální procesorová jednotka (Central Processing Unit)

ROM	Paměť pouze pro čtení (Read Only Memory)
RAM	Paměť s náhodným přístupem (Random Access Memory)
SRAM	Statická paměť s náhodným přístupem (Static RAM)
DAC	Digitálně-analogový převodník (Digital-Analog Converter)
USB	Univerzální sériová sběrnice (Universal Serial Bus)
OCR	Výstupní porovnávací registr (Output Compare Register)
ATMega2560	Mikrokontroler řady ATMega od firmy Atmel
PCI	Počítačová sběrnice

SEZNAM PŘÍLOH

Příloha P I: Arduino kódy

Příloha P II: CAD modely

Příloha P III: Manuál

Příloha P IV: MATLAB kódy pro vykreslení přechodových charakteristik

Příloha P V: Naměřená data

Příloha P VI: Seznam komponent