

Univerzita Pardubice
Fakulta chemicko-technologická

Matematický model zařízení „BATYSKAF“

Petra Wasserbauerová

Diplomová práce

2008

Na tomto místě bych ráda poděkovala vedoucímu mé diplomové práce, Doc. Ing. Františku Duškovi, CSc., za jeho odborné vedení, trpělivost, ochotu a čas, který mi věnoval. Rovněž děkuji svým rodičům a svému příteli Miroslavu Pipkovi za velikou podporu a trpělivost po celou dobu mého studia.

ABSTRAKT

Tématem této diplomové práce bylo sestavit matematický model zařízení „Batyskaf“. Spojitý dynamický model byl sestaven na základě matematicko-fyzikální analýzy. Pro komunikaci se zařízením byly vytvořeny potřebné uživatelské funkce. Parametry matematického modelu byly určeny pomocí statického a dynamického experimentu.

KLÍČOVÁ SLOVA

matematický model, experimentální identifikace, kalibrace, MATLAB

TITLE

Mathematical model of device "BATHYSCAPHE"

ABSTRACT

Thesis deals with the mathematical model of device „Bathyscaphe“ derivation. Continuous dynamical model was derived on basis of first principle analysis. Required user functions were programmed for communication with the device. First principle model parameters were calculated from static and dynamic experiment data.

KEYWORDS

mathematical model, experimental identification, calibration, MATLAB

OBSAH

Seznam použitých symbolů a zkratek	7
Seznam tabulek	8
Seznam grafů	9
Seznam obrázků	10
1 Úvod	11
1.1 Popis zařízení „Batyskař“	11
1.2 Princip funkce zařízení	12
1.3 Problémy současného stavu	13
2 Cíl práce	15
3 Teoretická část	16
3.1 Komunikace se zařízením „Batyskař“	16
3.1.1 Fyzické připojení zařízení „Batyskař“	16
3.1.2 Parametry komunikace	16
3.1.3 Podpora sériové komunikace na úrovni OS Windows	17
3.1.4 Podpora sériové komunikace v MATLABu	18
3.2 Fyzikální analýza „Batyskař“	22
3.3 Matematický model	23
4 Experimentální část	26
4.1 Řešení komunikace s úlohou „Batyskař“ v MATLABu	26
4.2 Kalibrace čidla tlaku	27
4.3 Experimentální identifikace	31
4.3.1 Statický experiment a jeho vyhodnocení	31
4.3.2 Dynamický experiment a jeho vyhodnocení	38
4.3.3 Výsledný matematický model s uvedením hodnot parametrů	43
5 Závěr	45
6 Seznam použité literatury	47
7 Dodatek	48

Seznam použitých symbolů a zkratek

a	$[\text{m}\cdot\text{s}^{-2}]$	zrychlení plováčku
F_a	$[\text{N}]$	setrvačná síla
F_g	$[\text{N}]$	tíhová síla
F_o	$[\text{N}]$	odporová síla
F_{vz}	$[\text{N}]$	vztlková síla
g	$[\text{m}\cdot\text{s}^{-2}]$	gravitační zrychlení
h	$[\text{m}]$	vzdálenost plováčku od úrovně čidla
h_x	$[\text{m}]$	výška hladiny nad úrovní čidla
m_k	$[\text{kg}]$	korigovaná hmotnost plováčku
m_p	$[\text{kg}]$	hmotnost plováčku
p	$[\text{Pa}]$	aktuální absolutní tlak v plováčku
p_a	$[\text{Pa}]$	absolutní tlak barometrický
Δp	$[\text{Pa}]$	přetlak nad hladinou
v	$[\text{m}\cdot\text{s}^{-1}]$	rychlost plováčku
V	$[\text{m}^3]$	aktuální objem vzduchu v plováčku
V_0	$[\text{m}^3]$	objem vzduchu v plováčku při p_a
V_p	$[\text{m}^3]$	objem plováčku
α	$[\text{N}\cdot\text{m}^{-1}\cdot\text{s}^{-1}]$	konstanta odporu prostředí
ρ	$[\text{kg}\cdot\text{m}^{-3}]$	měrná hustota kapaliny

Seznam tabulek

Tabulka 1: Funkce MATLABu pro práci se sériovým portem.....	19
Tabulka 2: Vybrané vlastnosti a metody objektu serial.....	21
Tabulka 3: Přehled základních funkcí a jejich význam.....	27
Tabulka 4: Výsledné hodnoty pro důkaz nerovnice (14).....	33
Tabulka 5: Průměrné naměřené hodnoty a vypočtený statický člen $V_0 \cdot \frac{\rho}{m_k}$ pro atmosférický tlak $p_a = 101,325$ kPa.....	35

Seznam grafů

Graf 1 - Závislost údaje z tlakového čidla a odečteného tlaku.....	29
Graf 2 - Závislost údaje z tlakového čidla na otevření ventilu.....	30
Graf 3 - Průběh regulace na žádanou hodnotu $w = 0,2$ m.....	35
Graf 4 - Grafické porovnání experimentálních dat a vypočtené hodnoty statického členu.....	37
Graf 5 - Závislost pro první průběh žádané hodnoty w	39
Graf 6 - Závislost pro druhý průběh žádané hodnoty w	40
Graf 7 – Průběh naměřené a vypočtené výšky pro první průběh žádané hodnoty w	42
Graf 8 – Průběh naměřené a vypočtené výšky pro druhý průběh žádané hodnoty w	43

Seznam obrázků

Obr.1 - Schéma zařízení „Batyskař“.....	12
Obr.2 - Schéma komunikace počítače se zařízením „Batyskař“	18
Obr.3 - Schéma působení sil na plováček.....	22
Obr.4 - Schéma „Batyskařů“	23
Obr.5 - Rozměrové schéma plováčku.....	32

1 Úvod

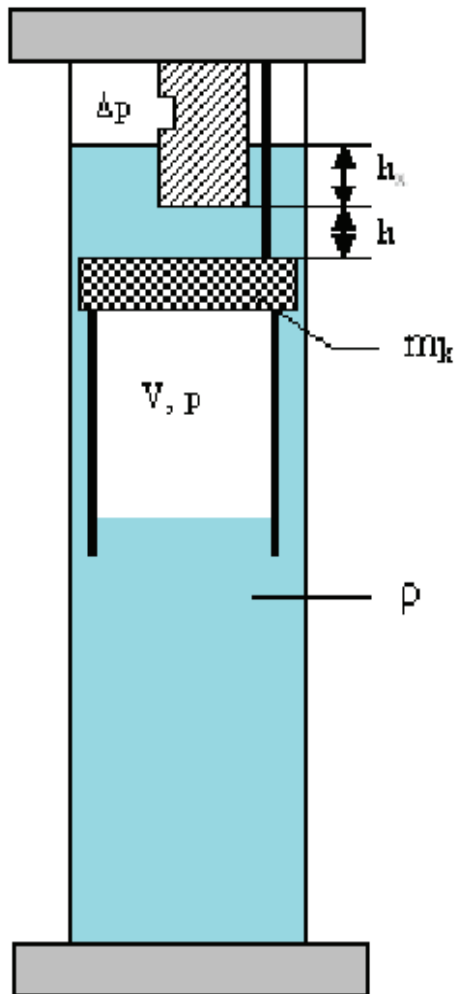
Základním předpokladem každého návrhu složitějšího způsobu řízení je informace o chování řízené soustavy. Tato informace je nejčastěji ve formě matematického modelu dynamického chování. Kvalita matematického modelu pak zcela zásadně ovlivňuje dosažitelnou kvalitu řízení.

Tato práce se zabývá sestavením dynamického spojitého matematického modelu vycházejícího z matematicko-fyzikální analýzy a určením jeho parametrů takových, aby chování popsané modelem se co nejvíce blížilo k chování určenému experimentálně na reálném zařízení.

1.1 Popis zařízení „Batyskař“

Laboratorní zařízení „Batyskař“ bylo vytvořeno v rámci grantového projektu GAČR 102/03/0625 „Konsorcionální přístup k vývoji experimentálních modelů“ jehož cílem bylo vytvořit sadu modelů využitelných při výuce v oblasti řízení.

Zařízení umožňuje ovlivňovat polohu plováčku vznášejícího se ve vodním sloupci pomocí změny tlaku vzduchu nad hladinou. Zařízení tvoří svisle umístěný průhledný válec (plexisklová trubka) naplněný destilovanou vodou. Válec je shora i zdola uzavřený a utěsněný. V horní části je umístěno měřicí zařízení polohy plováčku (ultrazvuková sonda) a přívod tlakového vzduchu. Ve válci je umístěn plováček. Je to kalíšek z tvrzeného plastu otočený dnem vzhůru, v jehož svislé ose je připevněn šroub s maticemi pro zvýšení jeho hmotnosti. Do boku plováčku je u spodního okraje vyvrtán otvor. Při ponoření se naplní plováček až po otvor vodou a ve zbylém objemu je uzavřen vzduch. Schéma zařízení laboratorního zařízení „Batyskař“ je na obr. 1.



Obr. 1 - Schéma zařízení „Batyskaf“

Kde Δp je přetlak nad hladinou, h_x je výška hladiny nad úrovní čidla, h je vzdálenost plováčku od úrovně čidla, m_k je korigovaná hmotnost plováčku, ρ je měrná hustota kapaliny, V je aktuální objem vzduchu v plováčku a p je aktuální absolutní tlak v plováčku.

1.2 Princip funkce zařízení

Pohyb plováčku (a zároveň tím i jeho poloha) je závislý na aktuální síle, která na něj působí. Tato síla je výslednicí síly tíhové, vztlakové, odporové a setrvačné. Tlakem nad hladinou je možné ovlivňovat sílu vztakovou. Při zvýšení tlaku nad hladinou se objem vzduchu uzavřený v plováčku zmenší. Tím dojde k poklesu objemu vody vytlačeného plováčkem, a zároveň i k poklesu vztakové síly, čímž se změní výslednice sil.

Důsledkem je změna polohy plováčku. Hmotnost plováčku a objem v něm uzavřeného vzduchu je stanoven tak, aby začal klesat přibližně v polovině tlakového rozsahu (ten je 0-36 kPa), tj. při přetlaku kolem 20kPa.

Zdrojem tlakového vzduchu je trvale běžící kompresor Silenta používaný v akvaristice. Maximální přetlak vzduchu na výstupu kompresoru je 36 kPa. Tlakový vzduch z kompresoru je veden nad hladinu ve válci a odtud je prostřednictvím dvou ventilů odpouštěn do okolí. Tlak nad hladinou ve válci je tedy závislý na otevření těchto ventilů. Ventily jsou dva proto, aby bylo možné pomocí jednoho řídit tlak a pomocí druhého simulovat poruchu. Ovládání ventilů je řešeno pomocí modelářských sedmiotáčkových servomotorů. Servomotory jsou připojeny k řídicí jednotce komunikující přes sériové rozhraní RS232C.

Měření polohy plováčku je provedeno bezdotykově pomocí ultrazvukové sondy. Ta vysílá ultrazvukový signál, který prochází vodou, dopadá na horní plochu plováčku a od ní se odráží zpět, kde ho přijímací část sondy přijme. Podle doby mezi vysláním a přijetím ultrazvukového signálu se určí vzdálenost plováčku od sondy. Měření času a jeho přepočítání na hodnotu vzdálenosti zajišťuje měřidlo DIO 570. Pro správnou činnost je nutné, aby sonda vysílající a přijímající ultrazvukový signál byla ponořena do vody. Informace o měřené vzdálenosti je předávána prostřednictvím sériového rozhraní RS232C.

Soustava obsahuje i tlakové čidlo, které je umístěno na zadní straně stojanu spolu s řídicí elektronikou pro servomotory. Tlak je k němu přiváděn odbočkou mezi výstupem tlaku z válce a ventily. Údaje z čidla jsou opět předávány přes sériové rozhraní RS232C.

1.3 Problémy současného stavu

Základním předpokladem funkčnosti měření polohy plováčku je udržování vhodné hladiny vody ve válci. Po klesnutí hladiny vody pod určitou úroveň nelze měřit, protože ultrazvukové čidlo není smočeno ve vodě po celou dobu měření. Voda se musí přibližně jednou za 14 dní doplnit a po určité době se musí i celý objem ve válci vyměnit, protože se tam začínají tvořit řasy i přesto, že se používá destilovaná voda. Řasy mají vliv na celkový pohyb plováčku (zpomalují ho). Voda se doplňuje podle pokynů, které jsou podrobně popsány v dokumentaci k zařízení [1]. Pro obnovení počátečních podmínek se

musí zařízení „Batyskaf“ vrátit do přibližně výchozího stavu tzn., že v plováčku bude objem vzduchu V_0 . Vzduch se opět doplní podle pokynů v dokumentaci k zařízení [1].

Problémem je také tlaková netěsnost, což má za následek to, že v době měření byl maximální dosažitelný tlak jen 32 kPa a ne 36 kPa, jak je uvedeno v kapitole 1. 2. Bylo sice provedeno přetěsnění, ale vyššího tlaku se již nepodařilo dosáhnout. Nižší tlak je zřejmě způsoben opotřebením ventilů (určitý únik tlaku v poloze zavřeno) a opotřebením kompresoru.

Dalším problémem bylo to, že výrobce laboratorního zařízení „Batyskaf“ dodal k tomuto zařízení program, který se dá spustit pomocí programu Simulink. Dále byl tento program udělán tak, že se pomocí něj daly řešit jen jednoduché úlohy a měření (regulace). Proto byla zvolena komunikace zařízení „Batyskaf“ s prostředím MATLAB [2]. Musela být vyřešena otázka společné komunikace a ovládání „Batyskafu“ tak, aby bylo možné řešit i složitější úkoly.

2 Cíl práce

Cílem této diplomové práce bylo zjistit způsob komunikace laboratorního zařízení „Batyskaf“ s počítačem, vyřešit komunikaci prostřednictvím dvou sériových linek v prostředí MATLAB a vytvořit uživatelské funkce pro komunikaci s „Batyskafem“. Pro vyhodnocení bylo potřeba provést kalibraci tlakového čidla, aby byly získány jeho hodnoty ve fyzikálních jednotkách. Dále bylo požadováno vytvoření spojitého dynamického matematického modelu zařízení na základě matematicko-fyzikální analýzy a určení jeho parametrů pomocí experimentální identifikace.

Potřebná měření včetně návrhu realizace potřebných experimentů a jejich vyhodnocení bylo požadováno provést v prostředí MATLAB.

3 Teoretická část

3.1 Komunikace se zařízením „Batyskaf“

Aby bylo možné vyměnit informace mezi „Batyskafem“ a konkrétním programem na PC je potřeba zajistit fyzické propojení a programovou podporu použitých technických prostředků. Obecná programová podpora je nutná jak na úrovni operačního systému (OS) tak na úrovni použitého programového vybavení. Poslední úrovní jsou uživatelské funkce usnadňující používání konkrétního zařízení s respektováním jeho specifických vlastností a možností použitého programového vybavení.

3.1.1 Fyzické připojení zařízení „Batyskaf“

Zařízení je vybaveno dvěma konektory CANON 9 pin??? female s rozložením signálů dle RS232C (TIA/EIA-232C). Využity jsou pouze signály RxD (pin 2), TxD (pin 3) a Gnd (pin 5). Pro fyzické připojení zařízení „Batyskaf“ k počítači jsou potřeba dva sériové porty s rozhraním RS232. U novějších počítačů, které obsahují pouze jeden nebo také žádný sériový port RS232, lze využít konvertorů mezi RS232C a USB. Nové počítače obsahují několik USB portů a na každý lze teoreticky připojit až 128 zařízení.

3.1.2 Parametry komunikace

Údaje o poloze plováčku, tlaku vzduchu nad hladinou a povely pro nastavení polohy servomotorů jsou předávány v digitální podobě pomocí dvou obousměrných sériových linek s rozhraním RS232C. Typem rozhraní jsou dány napěťové úrovně a asynchronní způsob přenosu dat s využitím START a STOP bitů. Pro připojení jsou použity devítipinové konektory CANON se standardním rozložením signálů. Volitelné parametry jsou pro obě linky stejné: přenosová rychlost 600 Bd, osm datových bitů bez parity, jeden stop bit. Nejsou zde využity žádné řídicí signály ani signál BREAK.

Komunikace je zajištěna pomocí dvou sériových linek. Jedna linka je využita pro přenos informace o poloze plováčku (směr z „Batyskafu“) a přenos požadavku na

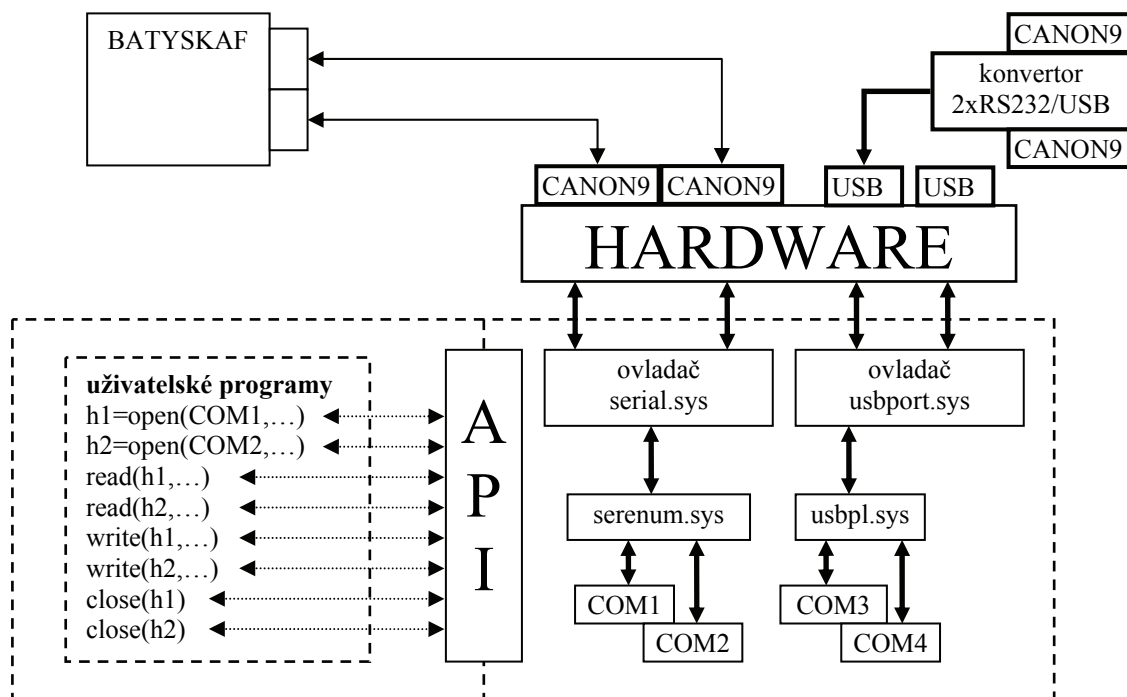
otevření jednoho ventilu (směr do „Batyskafu“). Druhá linka přenáší informace o aktuálním tlaku nad hladinou (směr z „Batyskafu“) a hodnoty požadavku na otevření druhého ventilu (směr do „Batyskafu“).

Číselná informace je přenášena ve formě ASCII řetězce (znaky číslice, tečka, plus, mínus) ukončeného znakem ‚*‘ (hvězdička). Komunikace je zahájena odesláním znaku ‚S‘ a ukončena odesláním znaku ‚F‘ směrem do „Batyskafu“. Informace z „Batyskafu“ jsou po zahájení komunikace přenášeny periodicky asi dvakrát za sekundu až do ukončení komunikace. Neobvyklé je to, že odesláním znaku ‚S‘ do jedné linky se zahájí posílání údajů na lince druhé. Požadavky na otevření ventilů lze posílat kdykoliv tj. i během přestavování servomotoru (přestavení z jedné krajní polohy do druhé trvá několik sekund).

Požadavky na otevření ventilů jsou ve formě celých čísel v rozsahu 0-500. Informace o měřeném tlaku jsou v jednotkách převodníku tj. celá čísla v rozsahu cca 0-1023. O čidle tlaku nejsou žádné informace a je tedy nutná kalibrace pro převod na tlak. Naopak ultrazvukové čidlo je kalibrováno a vysílané údaje jsou přímo v centimetrech vzdálenosti plováčku od čidla.

3.1.3 Podpora sériové komunikace na úrovni OS Windows

Operační systém Windows podporuje sériový port jako standardní zařízení tj. obsahuje pro něj ovladač a knihovnu dovolující přístup k portu na úrovni standardních služeb API (Application Program Interface). Ovládání přenosu informace je pak přístupné pomocí stejných funkcí používaných např. pro disk. Existují samozřejmě určité rozdíly (speciální parametry při inicializaci zařízení), ale principy používání jsou stejné. Zmíněné funkce (služby API) jsou beze změny použitelné i pro připojení prostřednictvím konvertoru RS232/USB. Ke každému konvertoru je dodáván software, který zajišťuje, že se nově připojené linky z programového hlediska chovají stejně jako fyzické porty. Prakticky to vypadá tak, že se nové (virtuální) porty přidají k existujícím. Například má-li PC dva fyzické sériové porty přístupné pod označením COM1, COM2 a připojíme k němu konvertor 2xRS232/USB, pak jsou tyto porty přístupné pod označením COM3 a COM4. Celé řešení lze formálně znázornit schématem na obr. 2.



Obr. 2 - Schéma komunikace počítače se zařízením „Batyskaf“

3.1.4 Podpora sériové komunikace v MATLABu

Programové prostředí MATLAB obsahuje standardně podporu pro sériovou linku od verze 5 – viz téma nápovědy Seriál Port I/O. Vlastní podporu sériové linky obsahuje i Real Time Tbx, kde tato podpora je zaměřena zejména na využití ze SIMULINKu. V demo verzi Real Time Tbx, která je dostupná zdarma, je podpora sériové linky omezena na přenosovou rychlost 600 Bd a proto tuto hodnotu zařízení „Batyskaf“ používá. V práci je využita standardní podpora MATLABu a proto používání v Real Time Tbx nebude dále zmiňováno. V následující tabulce 1 jsou shrnuty základní příkazy MATLABu využitelné pro práci se sériovým portem.

Tabulka 1: Funkce MATLABu pro práci se sériovým portem

Funkce	Popis
<i>obecné funkce</i>	
s = serial (...)	vytvoření objektu třídy serial; obsahuje specifické informace o vlastnostech vybraného portu
fopen(s,...)	inicializace portu určeného objektem serial
fclose (s)	ukončení práce s portem a jeho uvolnění pro použití jinou úlohou OS
delete (s)	zrušení objektu MATLABu
<i>funkce pro zápis dat</i>	
fprintf (s,...)	zápis formátovaných dat (s konverzí binárních dat do textového řetěze)
fwrite (s,...)	zápis neformátovaných dat (zápis binárních dat)
<i>funkce pro čtení</i>	
t = fget1 (s)	čtení textu (data bez ukončovače textu)
t = fgets (s)	čtení textu (data včetně ukončovače textu)
fscanf (s,...)	čtení formátovaných dat
fread (s,...)	čtení binárních dat

Základní schéma programu pracujícího se sériovou linkou v MATLABu je pak následující:

```
s = serial('COM1'); % vytvoření objektu s implicitními parametry
fopen(s);          % inicializace portu podle parametrů v objektu
...
fprintf(s,...);   % požadavek na vyslání dat na inicializovaný port
o = fgets(s);     % požadavek na přečtení textové odpovědi
...
fclose(s)        % ukončení práce s portem a jeho uvolnění
delete(s)        % zrušení objektu
```

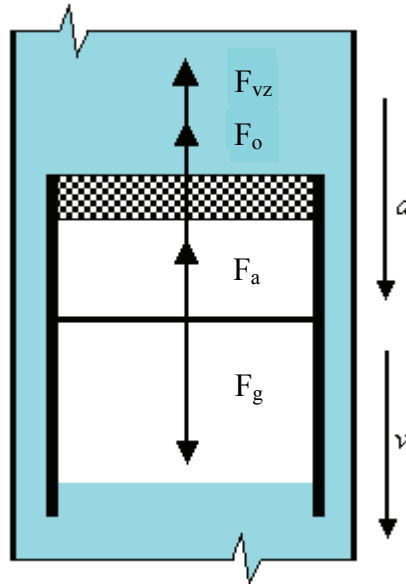
Důležité je rozlišit pojem objekt a identifikátor objektu. Je rozdíl mezi zrušením objektu (příkaz `delete`) a zrušením identifikátoru objektu (příkaz `clear`). Zrušíme-li identifikátor objektu a objekt zůstane, nemáme možnost dál ho používat. Pokud zůstal port např. neuzavřený, není možné ho dál znovu používat. Aby nebylo nutné ukončit a potom znova MATLAB spustit je možné objekty typu *serial* vyhledat a odkazy obnovit pomocí funkce **`instrfind`**. Koncepce použití objektů dovoluje, kromě shrnutí parametrů na jedno místo (vlastnosti), i snadnou práci s událostmi (metody). Objekt *serial* umožňuje zajistit, aby se uživatelem definovaná funkce (M-function callback) spustila asynchronně při výskytu příslušné události spjaté s činností sériového portu. Vlastnosti a metody objektu lze nastavovat buď při vytvoření objektu (funkce *serial*) nebo pomocí standardní funkce **`set`**. Seznam vybraných vlastností objektu *serial* se stručným popisem je v tabulce 2.

Tabulka 2: Vybrané vlastnosti a metody objektu serial

Vlastnost	Hodnota	Popis
<i>obecné vlastnosti</i>		
Port	COM1	identifikace portu (konvence dle OS)
Status	open closed	indikuje, zda je port inicializován či nikoliv
UserData		umožňuje uživateli připojit svoje data
<i>vlastnosti komunikace</i>		
BaudRate	300	definuje použitou přenosovou rychlost
DataBits	8	definuje použitý počet datových bitů
StopBits	1	definuje použitý počet stop bitů
Parity	none	definuje typ použité parity
Terminator	*	definice znaku určujícího konec čtení (fgetl, fgets) či přidaného na konec formátovaného zápisu (fprintf)
<i>vlastnosti související se zápisem dat</i>		
OutputBufferSize	512	definuje velikost výstupního bufru
BytesToOutput		indikuje počet nedeslaných byte ve výstupním bufru
<i>vlastnosti související se čtením dat</i>		
InputBufferSize	512	definuje velikost vstupního bufru
BytesAvailable		indikuje počet nevyčtených byte ve výstupním bufru
<i>metody objektu – události a jejich parametry</i>		
BytesAvailableFcnMode		definuje, zda událost nastane při definovaném počtu byte v bufru nebo při příjmu znaku Terminator
BytesAvailableFcn		odkaz na funkci, která se má při příslušné události spustit

3.2 Fyzikální analýza „Batyskafu“

Působení jednotlivých sil na plováček je znázorněno na obr. 3.



Obr. 3 - Schéma působení sil na plováček

Kde F_a je setrvačná síla, F_o síla odporu (předpokládá se, že při laminárním proudění je úměrná rychlosti), F_g je gravitační síla a F_{vz} je síla vztlaková.

Pro jednotlivé síly platí následující vztahy (1) až (4).

$$F_a = m \cdot a = m \cdot \frac{d^2 h}{dt^2} \quad (1)$$

$$F_o = \alpha \cdot v = \alpha \cdot \frac{dh}{dt} \quad (2)$$

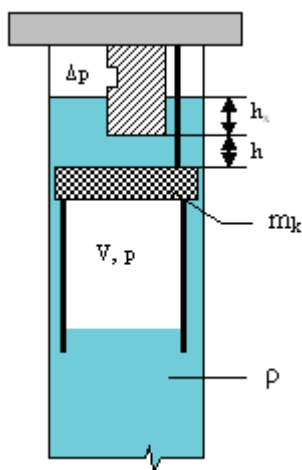
$$F_g = m_k \cdot g \quad (3)$$

$$F_{vz} = V \cdot \rho \cdot g \quad (4)$$

Kde α je konstanta odporu při obtékání plováčku v $[\text{N} \cdot \text{m}^{-1} \cdot \text{s}^{-1}]$, m je hmotnost plováčku v $[\text{kg}]$ a m_k je korigovaná hmotnost plováčku v $[\text{kg}]$.

3.3 Matematický model

Byla uvažována zjednodušená varianta zanedbávající změny výšky hladiny h_x v důsledku změny objemu vzduchu v plováčku. Matematický model pohybu plováčku zařízení vychází ze schémat uvedených na obr. 3 a 4, kde



Obr. 4 - Schéma „Batyskafu“

veličiny (časově proměnné hodnoty):

Δp [Pa] přetlak nad hladinou

h [m] vzdálenost plováčku od úrovně čidla

p [Pa] aktuální absolutní tlak v plováčku

V [m³] aktuální objem vzduchu v plováčku

parametry:

p_a [Pa] absolutní tlak barometrický

h_x [m] výška hladiny nad úrovní čidla

m_k [kg] korigovaná hmotnost plováčku

$$(m_k = m_p - \rho \cdot V_p)$$

m_p [kg] hmotnost plováčku

V_0 [m³] objem vzduchu v plováčku při p_a

konstanty:

ρ [kg.m⁻³] měrná hustota kapaliny

g [m.s⁻²] gravitační zrychlení

Z těchto schémat na obr. 3 a 4 vyplývá, že všechny vektory sil působí ve směru osy válce a změna směru působení síly je vyjádřena pouze změnou znaménka. Výchozí rovnice pro určení matematického modelu je pak rovnováha všech působících sil na plováček. Součet všech těchto sil musí být nulový. Toto vyjadřuje vztah (5).

$$F_a + F_o + F_g + F_{vz} = 0 \quad (5)$$

Po dosazení vztahů (1) až (4) s uvažováním směrů působících sil do rovnice (5) byla postupnou úpravou získána následující diferenciální rovnice (6).

$$m \frac{d^2 h}{dt^2} + \alpha \frac{dh}{dt} = m_k \cdot g - V \cdot \rho \cdot g \quad (6)$$

Pro tlak a objem v plováčku platí Boyleův zákon, který je popsán rovnicí (7).

$$p_a V_0 = p \cdot V = konst \quad (7)$$

Z tohoto zákona byla dosazením za celkový tlak získána následující rovnice (8).

$$p_a V_0 = [p_a + \Delta p + \rho g(h_x + h)] \cdot V \quad (8)$$

Dále byl z rovnice (8) vyjádřen objem V , což vyjadřuje rovnice (9).

$$V = V_0 \cdot \frac{p_a}{p_a + \Delta p + \rho \cdot g(h_x + h)} \quad (9)$$

Rovnice (6) byla upravena vydělením hmotností m a poté bylo za objem V dosazeno z rovnice (9) a tím byla získána rovnice (10).

$$\frac{d^2 h}{dt^2} + \frac{\alpha}{m} \frac{dh}{dt} = g \cdot \left[\frac{m_k}{m} - V_0 \cdot \frac{\rho}{m} \cdot \frac{p_a}{p_a + \Delta p + \rho \cdot g(h_x + h)} \right] \quad (10)$$

Konečné řešení pro matematický model bylo získáno ve formě diferenciálních rovnic (11) a (12) postupnou úpravou rovnice (10).

$$\frac{dh}{dt} = v \quad (11)$$

$$\frac{dv}{dt} = g \cdot \frac{m_k}{m} \cdot \left[1 - V_0 \cdot \frac{\rho}{m_k} \cdot \frac{p_a}{p_a + \Delta p + \rho \cdot g(h_x + h)} \right] - \frac{\alpha}{m} v \quad (12)$$

Aby zařízení splňovalo svoji funkci musí platit, že při ponoření plováčku do maximální hloubky a maximálním atmosférickém tlaku nad hladinou je vztlaková síla větší než síla gravitační, aby plováček začal stoupat vzhůru. Musí tedy platit nerovnost popsaná rovnicí (13).

$$1 > V_0 \cdot \frac{\rho}{m_k} \cdot \frac{p_a}{p_a + \rho \cdot g(h_x + h_{\max})} \quad (13)$$

4 Experimentální část

V experimentální části byl řešen problém komunikace mezi laboratorním zařízením „Batyskař“ a počítačem. Bylo potřeba také naměřit rozměry plováčku, objem vzduchu v něm obsažený a jeho celkový objem pro výpočet matematického modelu. Dále bylo nutné provést kalibraci tlakového čidla, aby hodnoty jím naměřené získaly rozměr fyzikální jednotky. Samotná experimentální identifikace byla rozdělena na dvě části. První část představuje statický experiment, kterým se ověřila hodnota členu $V_0 \cdot \frac{\rho}{m_k}$ v diferenciální rovnici (12) získaná přímým měřením vlastností plováčku. Druhou částí je dynamický experiment, jehož úkolem bylo zjistit hodnotu dalšího parametru α v rovnici (12), který ovlivňuje dynamické vlastnosti zařízení. Těmito experimenty byly určeny hodnoty potřebných parametrů. Po jejich dosazení do diferenciální rovnice (12) byl určen matematický model laboratorního zařízení „Batyskař“.

4.1 Řešení komunikace s úlohou „Batyskař“ v MATLABu

Prostředí MATLAB bylo zvoleno proto, že podporuje sériový port a umožňuje provádět všechny běžné matematické operace. Pomocí grafiky v MATLABu se snadno zobrazí výsledky získané výpočtem. Ovládání MATLABu je celkem jednoduché.

Uživatelská podpora komunikace se zařízením „Batyskař“ se skládá z pěti základních funkcí: **OpenB**, **StartB**, **MerB**, **ZadB** a **CloseB** (výpis jednotlivých funkcí je v Dodatku). Jelikož tyto funkce jsou určeny pro použití s konkrétním zařízením, jsou všechny parametry a vlastnosti specifické pro zařízení zahrnuty přímo ve funkcích. Protože zařízení posílá data periodicky, je nutno řešit důsledky přetečení bufru při nedostatečně častém vyčítání. MATLAB řeší problém možného přetečení bufru tím, že vstupní bufer se chová jako cyklický. Důsledkem je, že pokud žádáme o přečtení přijatých dat méně často, než data přicházejí, pak dostaneme hodnoty nejstarší a pravděpodobně navíc neúplná. To nastane v případě, že velikost vstupního bufru nebude odpovídat součtu délek v něm uložených přijatých hodnot. V použitých funkcích je tento problém řešen obsluhou události při příjmu ukončovacího znaku zprávy (součást funkce **OpenB**). V rámci této obsluhy je přijatá zpráva vždy vyčtena z bufru při výskytu ukončovacího

znaku a uložena do vlastnosti UserData příslušného objektu. Funkcí MerB pak je tato, vždy poslední (aktuální), hodnota pouze čtena. Důsledkem je, že se funkce MerB okamžitě provede a nečeká na příchod dat. Obdobně funkce ZadB pouze zařadí požadavek na vyslání dat do výstupního bufru a nečeká na dokončení. Zde by mohl nastat problém při zařazení velkého množství požadavků, které by se nestačily vysílat a nevešly se ani do výstupního bufru. V tomto případě by se návrat z funkce ZadB pozdržel do doby než by se bufer uvolnil. Stručný přehled všech těchto pěti funkcí a jejich význam je shrnut v tabulce 3.

Tabulka 3: Přehled základních funkcí a jejich význam

Syntaxe	Význam
s=OpenB('COMx')	Vytvoření objektu na portu 'COMx'
Start(s)	Inicializace a zahájení komunikace
ZadB(s,u)	Vyslání hodnoty x s kontrolou na meze (např. nastavení ventilů na určitou hodnotu)
y=MerB(s)	Vyčtení poslední přijaté hodnoty
CloseB(s)	Ukončení komunikace, uvolnění portu a zrušení objektu

4.2 Kalibrace čidla tlaku

Bylo zjištěno, že kalibrace ultrazvukového čidla není nutná, protože toto čidlo nám poskytuje již hodnoty přímo ve formě fyzikální jednotky (v cm). Ale není tomu tak u čidla tlakového. Proto bylo nutné provést jeho kalibraci. Nejdříve bylo potřeba získat hodnoty tlaku ve fyzikálních jednotkách. Pro získání těchto hodnot byl k zařízení připojen manometr. Z něho byly hodnoty odečteny vizuálně vždy v ustáleném stavu. Ty pak byly ručně zadány do předem vytvořeného a spuštěného skriptu s názvem Kalibrace (jeho výpis je uveden níže). Tento skript zajišťuje postupné otevírání jednoho ventilu. Nejdříve se muselo počkat, až soustava bude v ustáleném stavu a potom se v něm odečetla z manometru hodnota tlaku. Dále se vzaly hodnoty aritmetického průměru z deseti měření pro otevření ventilu a změřený tlak z tlakového čidla, které byly uloženy do matice *v* na příslušné pozice. Toto bylo provedeno celkem ve 33 cyklech, kde v každém cyklu se ventil otevřel o hodnotu +10. Tento skript byl sestaven tak, aby ukládal do jednoho

souboru jak ručně zadané hodnoty odečtené z manometru, tak i naměřené hodnoty z tlakového čidla.

Výpis funkce Kalibrace

```
%% TITLE inicializace
Oo=0;
s1=OpenB('com1');    s2=OpenB('com2'); % vytvoření objektu serial
Start(s1);           Start(s2);      % inicializace a zahájení komunikace
ZadB(s1,Oo);         ZadB(s2,Oo)
%% TITLE experiment
Omax=320;
krok=10;
N=Omax/krok;
v=zeros(N+1,3);

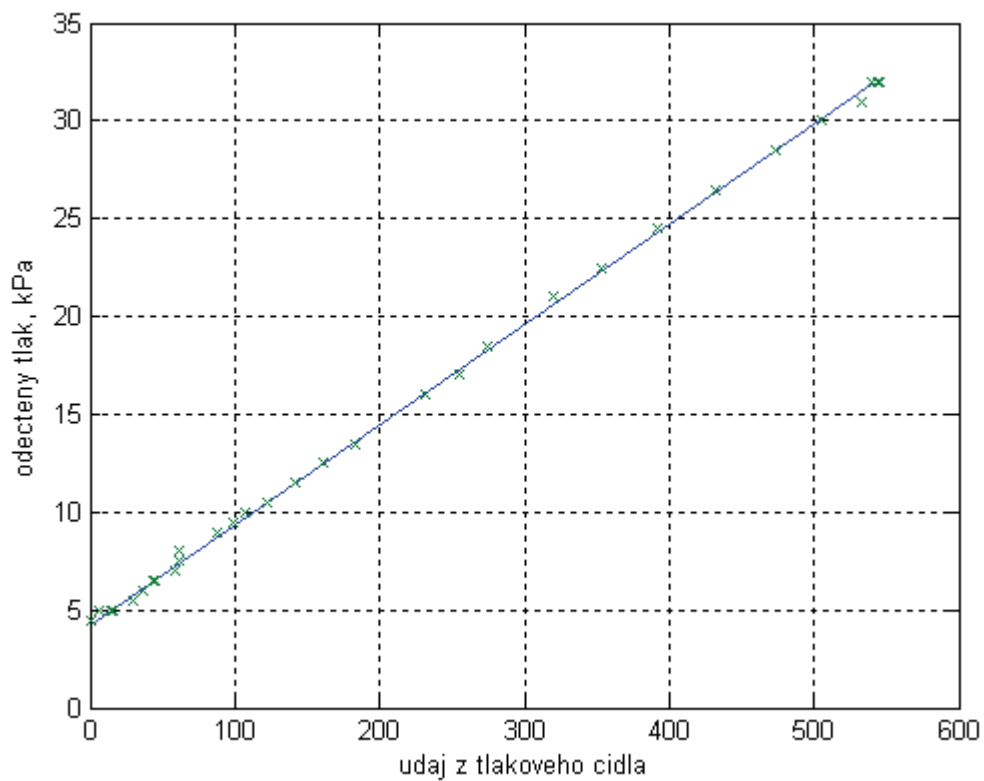
v(1,3)=input('Po ustaleni odeckti a zadej tlak: ');
s=0;
for k=1:10;
    s=s+MerB(s2);
    pause(1),
end
v(1,1)=Oo;          v(1,2)=s/10;
fprintf('O=%4d pm=%7.3f\n',v(1,1:2));

for i=2:N+1;
    Oo=Oo+krok;
    ZadB(s1,Oo);
    v(i,3)=input('Po ustaleni odeckti a zadej tlak: ');
    s=0; for k=1:10; s=s+MerB(s2); pause(1), end
    v(i,1)=Oo;          v(i,2)=s/10;
    fprintf('O=%4d pm=%7.3f\n',v(i,1:2));
end
%% TITLE ukonceni
CloseB(s1)
CloseB(s2)
```

Naměřená data pomocí skriptu Kalibrace byla konvertována z MATLABu do programu Excel, kde byla tato data zpracována a vyhodnocena. Vynesením hodnot tlaku odečtených z manometru a údajů získaných z tlakového čidla byl vytvořen graf 1. Z tohoto grafu 1 je vidět, že mezi těmito dvěma tlaky platí lineární závislost. Ta je vyjádřena rovnicí (14), která byla získána z grafu 1.

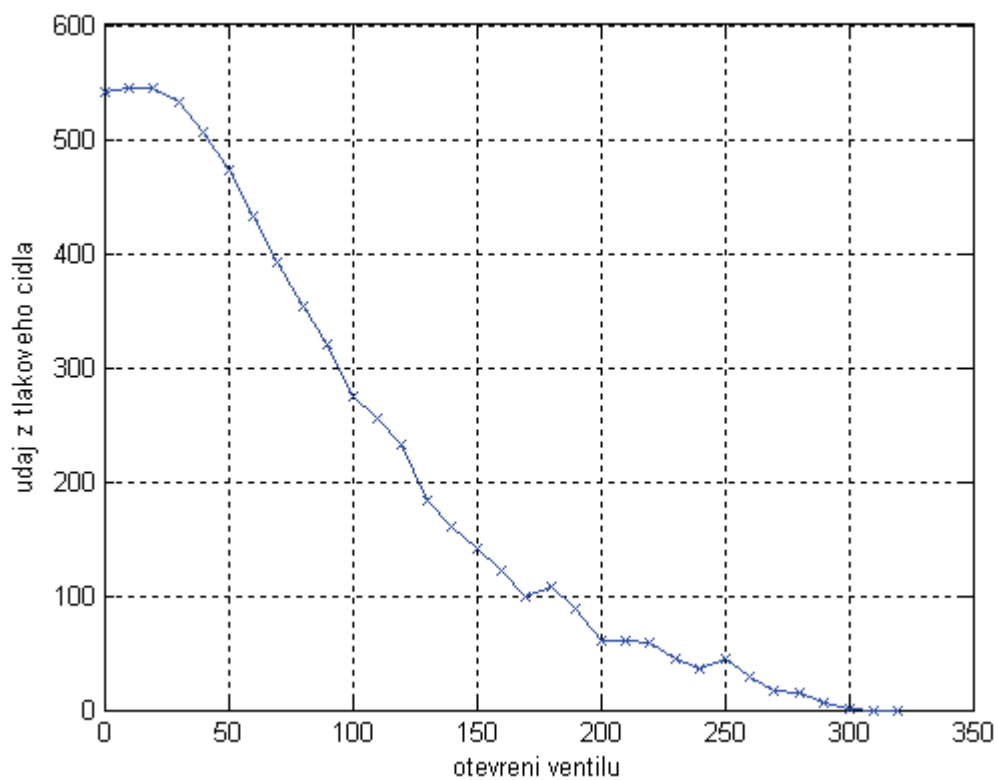
$$y = 0,0511x + 4,2762 \quad [\text{kPa, j.p.}] \quad (14)$$

Pomocí této rovnice (14) byla přepočítávána všechna naměřená data z tlakového čidla, aby byla získána ve fyzikálních jednotkách (v kPa).



Graf 1 - Závislost údaje z tlakového čidla a odečteného tlaku

Po získání lineární závislosti bylo přepočítáno otevření ventilu na jednotky převodníku. Jednotlivé rozsahy otevření ventilů a jednotek převodníku jsou uvedeny v kapitole 3.1.2. Pak byla vynesena do jednoho grafu závislost odečteného tlaku z manometru a údaje z tlakového čidla na otevření ventilů. Tato závislost je znázorněna v grafu 2. Z něho bylo vyčteno, že má smysl otevřít ventily jen v rozmezí hodnot 20 až 300, což po přepočítání představuje rozmezí 40,9 až 613,8 jednotek převodníku.



Graf 2 - Závislost údaje z tlakového čidla na otevření ventilu

4.3 Experimentální identifikace

Experimentální identifikace byla rozdělena na dvě části, a to na statickou a dynamickou. Statickým experimentem byly měřeny hodnoty v ustáleném stavu tzn., že vypadl dynamický člen α , a tudíž mohla být dohledána hodnota členu $V_0 \cdot \frac{\rho}{m_k}$ v diferenciální rovnici (12). Tímto experimentem bylo tedy ověřeno, zda hodnoty získané přímým měřením vlastností plováčku odpovídají hodnotám získaným vyhodnocením experimentálních dat v ustáleném stavu. Vyhodnocením dynamického experimentu byl určen člen α . Ten byl získán pomocí naměřeného časového průběhu jednotlivých parametrů (výšky h a tlaku Δp).

4.3.1 Statický experiment a jeho vyhodnocení

Tímto experimentem (jak už bylo řečeno v kapitole 4.3) byl ověřen člen $V_0 \cdot \frac{\rho}{m_k}$. Měření bylo provedeno v ustáleném stavu, protože v něm musí naměřené dvojice pro tlak a výšku splňovat rovnici (15).

$$\left[1 - V_0 \frac{\rho}{m_k} \cdot \frac{p_a}{p_a + \Delta p_i + \rho g(h_{xi} + h_i)} \right] = 0 \quad (15)$$

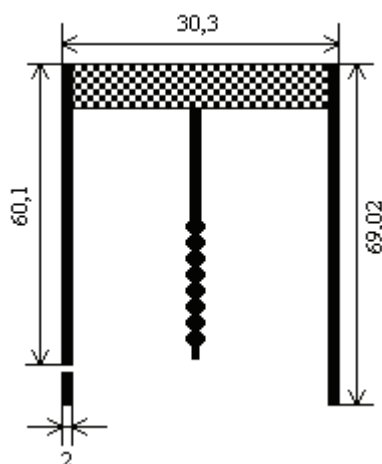
Z této rovnice (15) byla úpravou získána následující rovnice (16).

$$\Delta p_i + \rho \cdot g(h_{xi} + h_i) = \left(V_0 \cdot \frac{\rho}{m_k} - 1 \right) \cdot p_a \quad (16)$$

Z rovnice (16) byl matematickou úpravou vyjádřen hledaný statický člen, což ukazuje rovnice (17).

$$V_0 \cdot \frac{\rho}{m_k} = \left(\frac{\Delta p_i + \rho \cdot g(h_{xi} + h_i)}{p_a} \right) + 1 \quad (17)$$

Hodnota tohoto statického členu byla určena i teoreticky numerickým výpočtem. Za hodnoty V_0 a m_k byly dosazeny hodnoty, které byly zjištěny přímým měřením vlastností plováčku, protože dosažení těchto hodnot matematickým výpočtem je velmi složité. Potřebné rozměry plováčku byly změřeny pomocí posuvného měřítka. Rozměrové schéma plováčku v milimetrech je uvedeno na obr. 5.



Obr. 5 - Rozměrové schéma plováčku

Dále byl změřen objem plováčku V_0 , ve kterém je stlačený vzduch. Výsledné změřené hodnoty jsou:

$$V_0 = 25,5 \text{ ml} = 25,5 \cdot 10^{-6} \text{ m}^3$$

$$V_p = 18,9 \text{ ml} = 18,9 \cdot 10^{-6} \text{ m}^3$$

$$m_p = 0,0393 \text{ kg}$$

Pomocí těchto zjištěných hodnot byla vypočtena hodnota korigované hmotnosti m_k podle rovnice (18).

$$m_k = m_p - \rho \cdot V_p \quad (18)$$

Výsledná hodnota je tedy $m_k = 0,0204 \text{ kg}$.

Vypočtená hodnota hledaného statického členu $V_0 \cdot \frac{\rho}{m_k}$ je 1,2449.

Ve všech výpočtech byl atmosférický tlak zvolen konstantní a to $p_a = 101,325$ kPa.

Pomocí statického experimentu bylo také dokázáno, proč se plováček pohybuje směrem dolů pomaleji oproti pohybu nahoru. Zároveň tím byla dokázána i platnost nerovnice (13). To vše dokazuje následující tabulka 4.

Tabulka 4 – Výsledné hodnoty pro důkaz nerovnice (14)

w (m)	$\Delta p = 0$ kPa	$\Delta p = 30$ kPa
0,05	1,2392	0,9565
0,35	1,2043	0,9356

Z tabulky 4 je vidět, že pokud je člen na pravé straně nerovnice (13) větší než 1, pak plováček klesá, pokud je menší než 1, tak stoupá.

Pro změření hodnot potřebných pro výpočet statického členu byl vytvořen v programu MATLAB skript, který byl pojmenován Regulátor (jeho výpis je uveden níže). V něm se zadávala požadovaná žádaná hodnota w . Bylo zde nadefinováno i zesílení r_0 a interval vzorkování T . Ten byl zvolen $T=1,5$ s. Kdyby byl zvolen menší, tak by mohlo dojít k chybě při komunikaci se zařízením „Batyskaf“ a naopak při větším intervalu vzorkování by došlo ke ztrátě informací o chování soustavy.

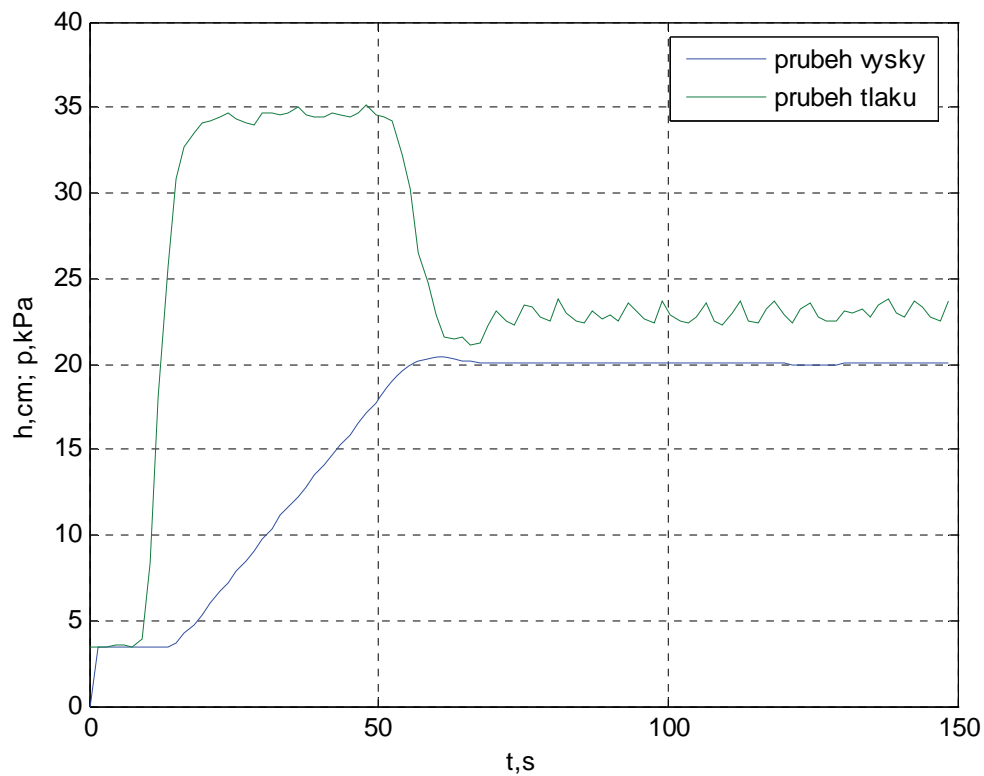
Měření bylo provedeno celkem ve 100 cyklech proto, aby bylo dosaženo ustáleného stavu. Tento stav byl získán pomocí diskrétního P-regulátoru se vzorkováním $o(k) = P^* [h_w(k) - h(k)]$. Další hodnoty pro P-regulátor byly zvoleny ručně. Naměřená data byla uložena do souboru data.txt.

Výpis funkce Regulátor

```
% inicializace
clear all
s1=OpenB('com1');   s2=OpenB('com2'); % vytvoření objektu serial
Start(s1);          Start(s2);       % inicializace a zahájení komunikace
ZadB(s1,0);         ZadB(s2,0);
w=10; r0=-25; N=100;
T=1.5;
Y=zeros(N,1);      tt=Y;      P=Y;   U=Y;
t=0; tic
for k=1:N,
    y=MerB(s1)/1000;
    p=MerB(s2);
    e=w-y;
    u=r0*e+75;
    ZadB(s1,u);
    fprintf('t=%6.1f    h=%6.4f    p=%6.4f    u=%6.4f\n',[t,y,p,u])
    tt(k)=t;      Y(k)=y;      P(k)=p;      U(k)=u;
    while t+T>toc, pause(0.01), end
    t=t+T;
end
CloseB(s1);        CloseB(s2);
clear s1 s2
XXX=[tt,Y,P,U];
save data.txt XXX -ascii
```

Měření bylo provedeno celkem pro pět různých žádaných hodnot ($w = 0,05; 0,1; 0,2; 0,3$ a $0,35$ m). Každé měření bylo provedeno dvakrát. Na ukázkou je v grafu 3 znázorněn průběh žádané hodnoty $w = 0,2$. Z tohoto grafu 3 je vidět, že P-regulátor doreguluje výšku plováčku na žádanou hodnotu bez trvalé regulační odchylky. To je způsobeno tím, že tato soustava je astatická. Je zde i vidět, že výška mírně kmitá kolem žádané hodnoty, nebo-li bylo dosaženo tzv. metastabilního stavu. O tomto stavu bude dále zmiňováno jako o ustáleném, poněvadž se pro další výpočty berou průměrné hodnoty z těchto naměřených dat. Průměrné naměřené hodnoty výšky h_{xi} a h_i a hodnoty tlaku Δp_i pro zadané žádané hodnoty w pomocí skriptu Regulátor jsou uvedeny v tabulce 5. Průměrné naměřené hodnoty byly odečteny ručně v programu Excel. Pro změřené průměrné hodnoty výšky h_{xi} a h_i a hodnoty tlaku Δp_i pro jednotlivé žádané hodnoty w byla určena hodnota výrazu $V_0 \cdot \frac{\rho}{m_k}$ z rovnice (17). Tyto výsledné hodnoty ověřovaného členu

$V_0 \cdot \frac{\rho}{m_k}$ pro jednotlivé žádané hodnoty w jsou také uvedeny v tabulce 5.



Graf 3 – Průběh regulace na žádanou hodnotu $w = 0,2$ m

Tabulka 5: Průměrné naměřené hodnoty a vypočtený statický člen $V_0 \cdot \frac{\rho}{m_k}$ pro

atmosférický tlak $p_a = 101,325$ kPa

w (m)	h_{xi} (m)	h_i (m)	Δp_i (kPa)	$V_0 \cdot \frac{\rho}{m_k}$
0,05	0,003	0,0479	23,7669	1,2392
0,1	0,0025	0,0986	23,5604	1,2421
0,2	0,0021	0,2003	22,9496	1,2458
0,3	0,0018	0,3019	21,929	1,2456
0,35	0,001	0,3527	21,7038	1,2483

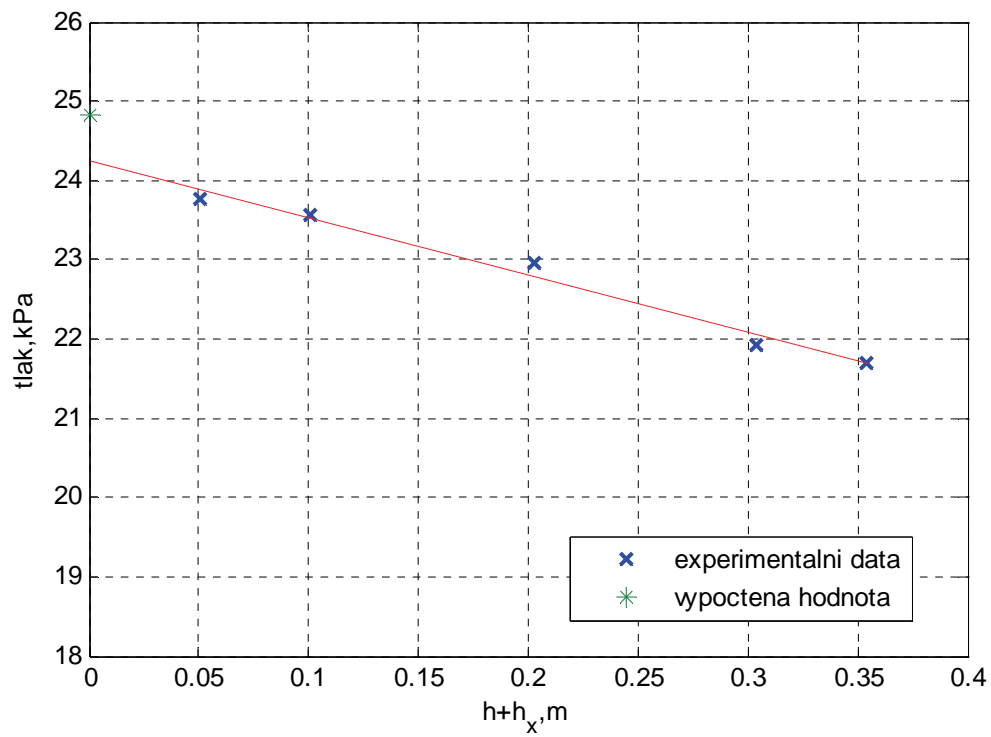
Z tabulky 5 je vidět, že hodnoty statického členu získané experimentálně jsou téměř shodné, a od teoreticky vypočtené hodnoty se liší jen nepatrně.

Poté byla do grafu 4 vynesena průměrná naměřená data z tabulky 5, aby byla získána závislost naměřeného tlaku Δp_i na výšce h_i+h_{xi} . Vynesené body pro jednotlivé výšky byly proloženy přímkou o rovnici (19).

$$\Delta p_i = \left(V_0 \cdot \frac{\rho}{m_k} - 1 \right) \cdot p_a - \rho \cdot g(h_{xi} + h_i) \quad (19)$$

Do stejného grafu byla pro srovnání vynesena i teoreticky vypočtená hodnota pro nulovou výšku. Experimentálně nulové výšky nelze dosáhnout, protože minimální výška plováčku je 3,4 cm. To je také jeden z důvodů proč se hodnoty získané přímým měřením ověřují experimentálně. Proto byly body získané experimentálním měřením proloženy přímkou vyjádřenou rovnicí (19). V místě protnutí přímky s osou y, tzn. při nulové výšce, byla odečtena průměrná experimentálně hledaná hodnota tlaku Δp 24,238 kPa. Vypočtená hodnota je 24,815 kPa. Chyba mezi experimentálně získanou a vypočtenou hodnotou je 2,38 %. Tato chyba je téměř zanedbatelná. Ale toto číslo platí jen pro toto dané vyhodnocení. Atmosférický tlak p_a nebyl totiž během experimentu měřen, a tudíž není znám. Do výpočtu byla použita již dříve zmiňovaná hodnota tlaku $p_a = 101,325$ kPa. Pokud by se v daném měřeném čase skutečný atmosférický tlak lišil o více jak 5 %, byla by chyba větší.

Z těchto výsledků plyne, že hodnoty získané přímým měřením vlastností plováčku byly ověřeny experimentálně a tyto hodnoty spolu souhlasí.



Graf 4 - Grafické porovnání experimentálních dat a vypočtené hodnoty statického členu

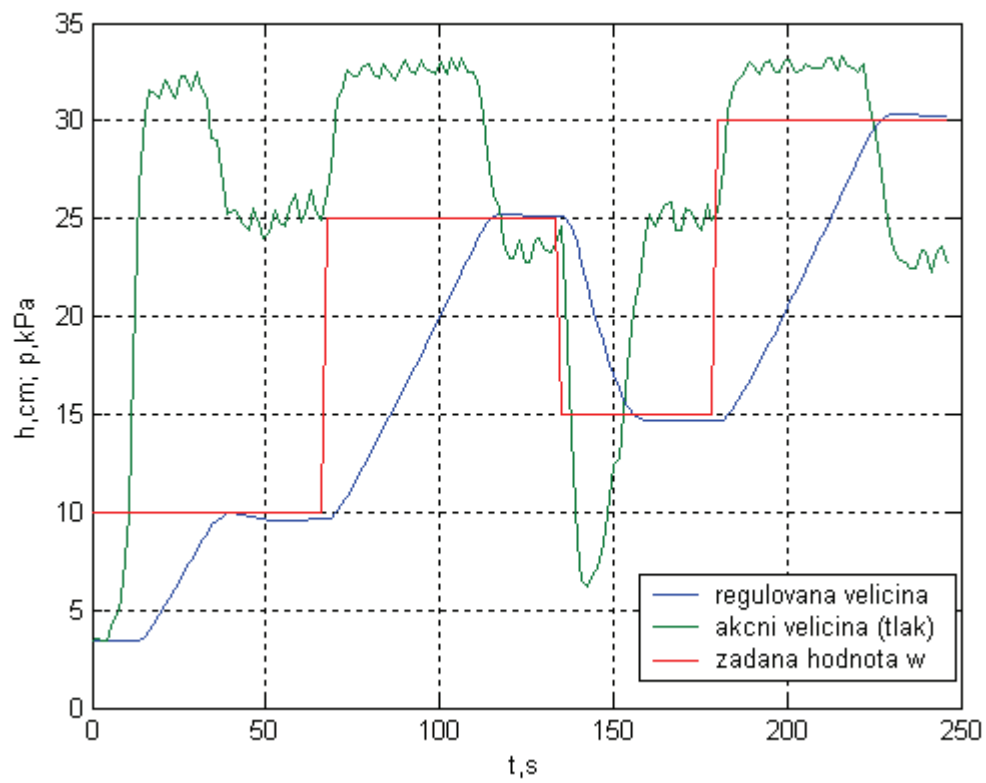
4.3.2 Dynamický experiment a jeho vyhodnocení

Dynamickým experimentem byl určen člen α . Jeho hodnota byla získána pomocí změřeného časového průběhu parametrů výšky h a tlaku Δp . Pro změření dynamické závislosti byl vytvořen skript s názvem Dynamika, jehož výpis je uveden níže. Byl zde zvolen průběh žádané hodnoty w . Skript je sestavený tak, že se vždy počká, až se hodnota výšky dostane na žádanou hodnotu w a poté teprve dojde ke změně žádané hodnoty w na jinou hodnotu. Tato doba byla zadávána ručně, protože při pohybu plováčku k žádané hodnotě směrem vzhůru se pohybuje rychleji, než při pohybu opačným směrem. Také zde muselo být zajištěno to, aby se plováček nedostal do krajních poloh. Tato situace by mohla nastat, protože tato soustava je astatická. Z toho vyplývá, že tento průběh musel být řízen, aby se této situaci zabránilo. Bylo toho docíleno opět pomocí P-regulátoru, který doreguluje výšku plováčku na žádanou hodnotu w .

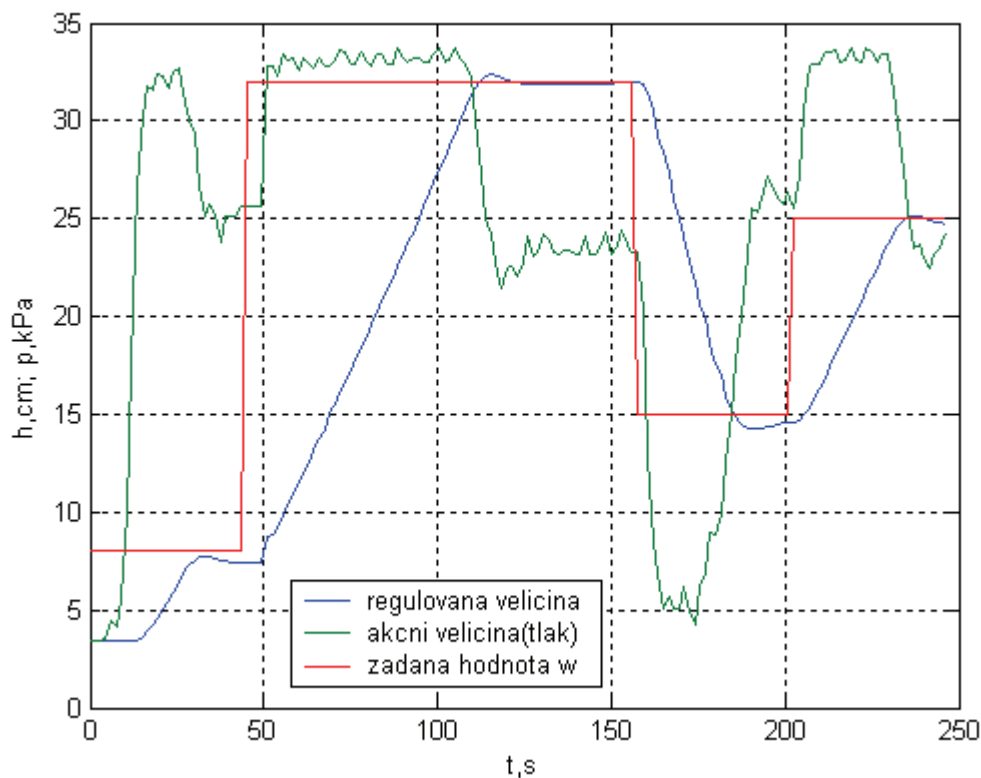
Výpis funkce Dynamika

```
% inicializace
clear all
s1=OpenB('com1');    s2=OpenB('com2');
Start(s1);          Start(s2);
ZadB(s1,0);         ZadB(s2,0);
Tw=30;
w=[10*ones(1.5*Tw,1);25*ones(1.5*Tw,1);15*ones(Tw,1);30*ones(1.5*Tw,1)];
r0=-25; N=165;
T=1.5;
Y=zeros(N,1);    tt=Y;    P=Y;    U=Y;    W=Y;
t=0; tic
for k=1:N,
    y=MerB(s1)/1000;
    p=MerB(s2);
    e=w(k)-y;
    u=r0*e+75;
    ZadB(s1,u);
    fprintf('t=%3.1f h=%6.4f p=%6.4f u=%6.4f\n w=%6.4f\n',t,y,p,u,w(k))
    tt(k)=t;    Y(k)=y;    P(k)=p;    U(k)=u;
    while t+T>toc, pause(0.01), end
    t=t+T;
end
CloseB(s1);        CloseB(s2);
clear s1 s2
XXX=[tt,Y,P,U,w];
save data2.txt XXX -ascii
```

Časová závislost výšky h a tlaku Δp byla změřena pro dva různé průběhy žádané hodnoty w . Jejich průběh byl vynesena do grafů 5 a 6.



Graf 5 - Závislost pro první průběh žádané hodnoty w



Graf 6 - Závislost pro druhý průběh žádané hodnoty w

Pro vyhodnocení dat dynamického experimentu bylo potřeba, aby experiment začínal z ustáleného stavu. Za ustálený stav nelze považovat stav ihned na začátku experimentu, proto byla data vyhodnocována až od prvního ustáleného stavu. Hodnoty tohoto prvního ustáleného stavu byly určeny ručně tak, že se do vyhodnocení zahrnovaly data až od 40 změřené hodnoty dále.

Dále bylo potřeba najít nulovou hodnotu daného kritéria. To bylo zvoleno jako suma kvadrátů odchylek naměřené a vypočtené výšky, což se matematicky dá zapsat rovnicí (20).

$$Kr = \sum_{i=1}^N (h_{mi} - h_{vi}(p_i))^2 \quad (20)$$

Byla dohledávána i optimální hodnota atmosférického tlaku p_a a hodnota hledaného dynamického parametru α . Tyto hodnoty byly získány optimalizací, která v sobě zahrnovala požadavek najít co nejmenší hodnotu daného kritéria popsáno rovnicí (20). Při této hodnotě kritéria jsou totiž hledané hodnoty parametrů optimální. Kritérium se dá také zapsat vztahem (21). Z něho plyne, že pro získání optimální hodnoty parametru α a atmosférického tlaku p_a musí být známy hodnoty tlaku p_i a výšky h_i . Ty byly získány z časového průběhu těchto parametrů.

$$Kr(\alpha, p_a; p_i, h_i) \quad (21)$$

Všechny tyto myšlenky v sobě zahrnuje vytvořený skript pro vyhodnocení dynamického experimentu, jehož výpis je uveden v Dodatku. Pomocí tohoto skriptu byly získány tyto výsledné hodnoty:

Pro první průběh žádané hodnoty w :

$$Kr = 0,001$$

$$p_a = 104,559 \text{ kPa}$$

$$\alpha = 3,811 \text{ N.m}^{-1} \cdot \text{s}^{-1}$$

Pro druhý průběh žádané hodnoty w :

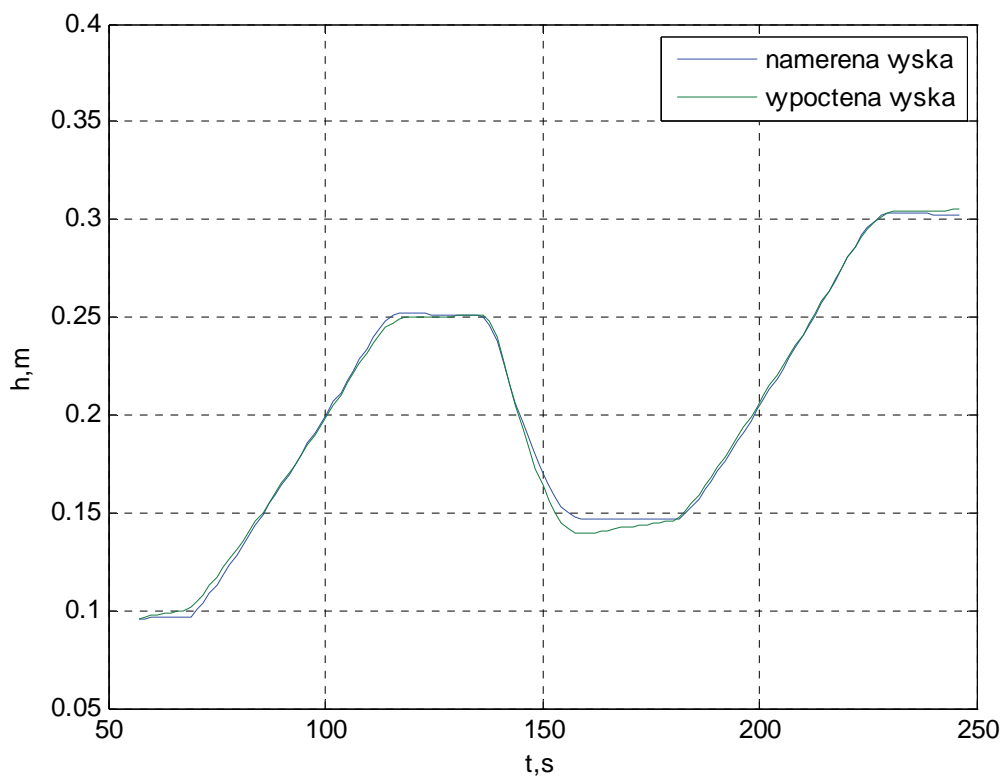
$$Kr = 0,007$$

$$p_a = 104,079 \text{ kPa}$$

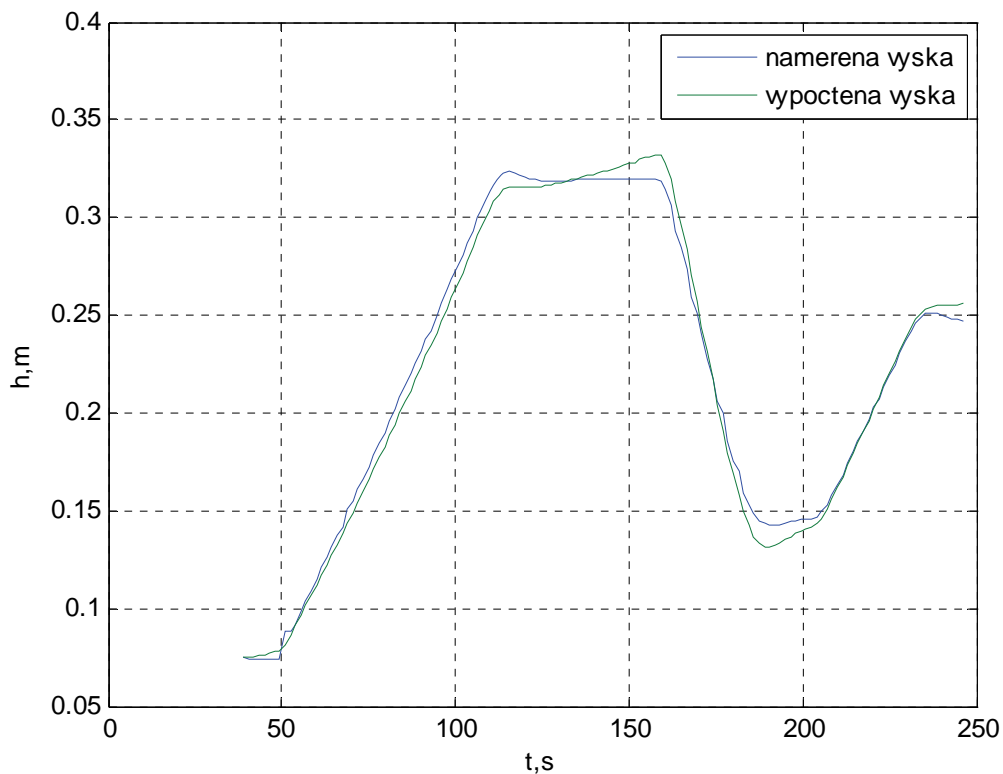
$$\alpha = 3,607 \text{ N.m}^{-1} \cdot \text{s}^{-1}$$

Hodnoty dynamického parametru α se mírně liší. Chyba je však jen 5,3 %. Tato chyba je relativně malá, tudíž ji můžeme zanedbat.

Dále byl vykreslen pro srovnání obou průběhů graf 7 a 8, ve kterých je vidět průběh naměřené a vypočtené výšky pomocí optimalizace. Z těchto grafů je vidět, že platí to, co bylo dokázáno již pomocí statického experimentu v kapitole 4.3.1. Tzn., že plováček klesá pomaleji, a tudíž trvá déle než se ustálí na žádané hodnotě w . Při stoupání plováčku na žádanou hodnotu w se na ní ustálí rychleji. To odpovídá strmější křivce v grafu oproti křivce, která vyjadřuje klesání plováčku.



Graf 7 – Průběh naměřené a vypočtené výšky pro první průběh žádané hodnoty w



Graf 8 - Průběh naměřené a vypočtené výšky pro druhý průběh žádané hodnoty w

4.3.3 Výsledný matematický model s uvedením hodnot parametrů

Z matematicko-fyzikální analýzy byl získán matematický model zařízení „Batyskař“, který je popsán diferenciálními rovnicemi (11) a (12), jejichž odvození je ukázáno v kapitole 3.3. Tyto rovnice se dají napsat souhrnně jako jedna diferenciální rovnice druhého řádu, což ukazuje rovnice (22).

$$\frac{d^2h}{dt^2} = g \cdot \frac{m_k}{m} \cdot \left(1 - V_0 \cdot \frac{\rho}{m_k} \cdot \frac{p_a}{p_a + \Delta p + \rho \cdot g \cdot (h_x + h)} \right) - \frac{\alpha}{m} \cdot \frac{dh}{dt} \quad (22)$$

Hodnota statického členu získaná přímým měřením vlastností plováčku, která byla ověřena statickým experimentem je $V_0 \cdot \frac{\rho}{m_k} = 1,2449$.

Hodnota dynamického členu byla určena pomocí dynamického experimentu. Jeho výsledná hodnota byla získána aritmetickým průměrem dvou hodnot získaných ze dvou vyhodnocení dvou časových průběhů. Tato hodnota je $\alpha = 3,709 \text{ N} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$.

5 Závěr

V této diplomové práci byla navržena a realizována komunikace zařízení „Batyskař“ s počítačem využívající dvě sériové linky v prostředí MATLAB. Bylo vytvořeno pět základních uživatelských funkcí na ovládání zařízení. Tyto funkce jsou uvedeny v Dodatku. Dále byla provedena kalibrace tlakového čidla pro získání informace o tlaku ve fyzikálních jednotkách, které jsou potřeba ve fyzikálním modelu. Výsledná rovnice pro přepočítání hodnot je vyjádřena rovnicí (14) v kapitole 4.2. Podle této rovnice byly přepočítávány všechny naměřené hodnoty tlakového čidla.

Poté byla provedena matematicko-fyzikální analýza. Pomocí ní byl odvozen matematický model zařízení „Batyskař“, který je vyjádřen rovnicí (12). Tato diferenciální rovnice obsahuje jednak parametry, které lze určit z geometrických rozměrů a jednak parametry určené pouze experimentálně. Pro tyto parametry byly provedeny dva příslušné experimenty.

Statický experiment znamená, že se hodnoty měří v ustáleném stavu, proto je dynamický člen roven 0 a tudíž vypadne. Pro tento experiment byl vytvořen skript s názvem Regulátor. Tento skript v sobě zahrnuje P-regulátor, který reguluje výšku plováčku na žádané hodnotě w a to i bez trvalé regulační odchylky. To je způsobeno tím, že tato soustava je astatická. Statický člen $V_0 \cdot \frac{\rho}{m_k}$ byl získán přímým měřením vlastností plováčku. Tímto experimentem bylo ověřeno, že tento člen byl určen správně. Zároveň tímto bylo i ukázáno, že se dá tento statický člen získat dvěma způsoby, a to jak přímým měřením, tak i experimentálním.

Pomocí dynamického experimentu byl hledán druhý člen α výsledné diferenciální rovnice (12). Byly změřeny časové průběhy parametrů výšky h a tlaku Δp . Tyto průběhy byly změřeny pro dva různé průběhy žádané hodnoty w . Vyhodnocením těchto časových průběhů a následnou optimalizací byl získán hledaný člen α . Při vyhodnocování hodnot tohoto experimentu musely být hodnoty uvažovány až od prvního ustáleného stavu. Pomocí skriptu pro vyhodnocení těchto naměřených dat byla dohledávána i hodnota atmosférického tlaku p_a , protože bylo zjištěno, že se tento tlak v průběhu experimentu mění a není konstantní, jak se dříve předpokládalo. Hodnoty tlaku p_a a hledaného dynamického členu α byly dohledávány pomocí optimalizační metody, která měla za úkol najít co nejmenší hodnotu daného kritéria. Při této hodnotě byly hledané parametry

optimální. To vše v sobě zahrnuje vytvořený skript pro vyhodnocení dat změřených dynamickým experimentem (jeho výpis je uveden v Dodatku).

Pomocí obou experimentů bylo dokázáno, že se plováček směrem dolů pohybuje pomaleji oproti pohybu směrem vzhůru. Pro vylepšení chování plováčku by se mohla zvýšit hmotnost plováčku Vp přidáním závaží. Tím by se zajistila stejná rychlost plováčku směrem nahoru i dolů.

Dále bylo dokázáno, že získaný matematický model z matematicko-fyzikální analýzy je správný, poněvadž odpovídá svým popisem chování reálné řízené soustavy.

6 Seznam použité literatury

- [1] Grantový projekt GAČR 102/03/0625 „*Konsorcionální přístup k vývoji experimentálních modelů*“. Praha, prosinec 2005.
- [2] DUŠEK, F.; HONC, D. *Matlab a Simulink (úvod do používání)*. Skripta UP-FCHT Pardubice, 2005.

7 Dodatek

Výpis funkce OpenB

```
function s=OpenB(com)
% vytvoření objektu serial na portu 'COMx', jeho otevření a
% nastavení zpracování číselného řetězce ukončeného znakem '*'
% pomocí asynchronně spouštěné funkce. Poslední číselná hodnota je v
% s.UserData

s=serial(com,'Terminator',' ','BaudRate',600);
s.BytesAvailableFcnMode = 'terminator';
s.BytesAvailableFcn = @zprac;
s.Tag='Batyskaf';
s.UserData=-1; % příznak žádná data
fopen(s) % inicializace portu
return % konec funkce OpenB

function zprac(obj,event) % funkce volaná při příjmu znaku '*'
p=obj.BytesAvailable; % aktuální počet znaků v bufru
if p==0, s.UserData=-1;
return;
end
x=fread(obj,p); % všechny vyčti (poslední je *)
x=char(x); % převed' na řetězec
if x(p)~='*', % na konci není *
z=' '; p0=p;
while (z~='*') & (p0>1), % najdi další hvězdičku
p0=p0-1; z=x(p0);
end
p=p0; % pozice další hvězdičky
end % p0 je pozice první * od konce
if p>1, % je něco v bufru?
p1=p-1; z=x(p1); p0=p1; % najdi další hvězdičku
while (z~='*') & (p0>1), p0=p0-1; z=x(p0); end
if x(p0)=='*', p0=p0+1; end
obj.UserData=sscanf(x(p0:p1),'%d');
else % v bufru není * nebo je pouze samotná
obj.UserData=-1; % příznak, že nejsou žádná data
end
return % konec funkce zprac
```

Výpis funkce StartB

```
function Start(s)
% inicializace a zahájení komunikace

fprintf(s,'S') % vyšli znak S = zahájení komunikace
```


Výpis funkce MerB

```
function y=MerB(s)
% vyčtení poslední přijaté hodnoty (-1, žádná data)

y=s.UserData;           % zkopíruj poslední hodnotu
```

Výpis funkce ZadB

```
function ZadB(s,x)
% vyslání hodnoty s kontrolou na meze

if x<0,    x=0;           % kontrola na dolní mez
elseif x>300, x=300;     % kontrola na horní mez
end
x=fix(x);
fprintf(s, '%04d\n',x)   % odeslání požadavku včetně terminátoru
```

Výpis funkce CloseB

```
function CloseB(s)
% ukončení komunikace, uvolnění portu a zrušení objektu

fprintf(s, 'F');        % vyšli znak 'F' = ukončení komunikace
pause(0.1)              % počkej na vyslání
fclose(s)               % uvolni port
delete(s)               % zruš objekt
```

Výpis ukázkového skriptu pro realizaci měření přechodového děje

```
% skript pro zkoušení komunikace s BATYSKAFem

Ts=5.0; N=24;           % interval vzorkování a počet měření
ch1=OpenB('com1'); ch2=OpenB('com2'); % inicializace
StartB(ch1); StartB(ch2); % zahaj komunikaci
ZadB(ch1,0); ZadB(ch2,0); % nastav počáteční stav 0,0
input('Pockej na ustaleni a stiskni Enter');
ZadB(ch1,400);          % nastav nové otevření ventilu na 400
disp('Zahájeno měření')
t=0; tic
for k=1:N,
    h=MerB(ch1)/1000; p=MerB(ch2); % změř výšku a tlak
    fprintf('t=%6.1f h=%6.4f p=%6.4f vl=%6.4f\n',[t,h,p,u(k)])
    tt(k)=t; pp(k)=p; hh(k)=h; % ulož aktuální měření
    while t+T>toc, pause(1), end % čekej na další měření
    t=t+T; % aktualizuj čas dalšího měření
end
CloseB(ch1); CloseB(ch2); % ukonči komunikaci a uvolni porty
clear ch1 ch2
```

Výpis skriptu pro vyhodnocení dynamického experimentu

```
function vyhodnoceniDynKr
% skript pro vyhodnocení dynamického experimentu

load datadynamika.mat;           % načtení dat

d1=ww(:,1);                      % čas v sekundách
d2=ww(:,2)/100;                  % převod na metry
d3=0.0511*ww(:,3)+4.2762;        % převod na pascaly
parD.dp=d3;                      % [Pa]
parD.tt=d1;                      % [s]
parD.hm=d2;                      % [m]

parD.hx=0.005;                  % výška hladiny nad úrovní čidla [m]
parD.V0= 25.5*10^(-6);          % objem vzduchu v plováčku [m^3]
parD.Vp=18.9*10^(-6);          % objem plováčku [m^3]
parD.mp=0.0393;                 % hmotnost plováčku [kg]
parD.mk=parD.mp-998*parD.Vp;    % korigovaná hmotnost plováčku [kg]

% výchozí hodnoty pro odhad hledaných parametrů
p0=[112;                          % atmosférický tlak pa
    10];                          % dynamický koeficient alfa
p=fminsearch(@ (x) funkceKr(x,parD),p0); % optimalizační funkce
[Kr,hv]=funkceKr(p,parD);

fprintf('Kr=%8.3f pa=%8.3f alfa=%8.5f\n',[Kr;p])
plot(parD.tt,parD.hm,parD.tt,hv) % vykreslení naměřených a vypočtených výšek
grid
xlabel('t,s')
ylabel('h,m')
legend('namerena vyska','vypoctena vyska')

return

function [Kr,hv]=funkceKr(p,parD)
% funkce pro hledání kritéria

parD.pa=p(1);                    % hledaný tlak pa
parD.alfa=p(2);                  % hledané alfa
hm=parD.hm;                      % měřená výška
H0=[hm(1);0];                   % ustálený stav na začátku
[t,H]=ode15s(@ (t,y) funkceH(t,y,parD),parD.tt',H0);

hv=H(:,1);                      % vektor vypočtených hodnot výšek
Kr=(parD.hm-hv)'*(parD.hm-hv);   % kritérium

fprintf('Kr=%8.6f pa=%8.3f alfa=%8.3f\n',[Kr;p])

plot(parD.tt,parD.hm,t,hv,'-r') % vykreslení naměřených a vypočtených výšek
grid
xlabel('t,s')
ylabel('h,m')
legend('namerena vyska','vypoctena vyska')

pause(0.1)

return
pokračování skriptu na další stránce
```

Pokračování výpisu skriptu pro vyhodnocení dynamického experimentu

```
function dH=funkceH(t,H,parD)
% diferenciální rovnice Batyskafu

g=9.81;           % gravitační zrychlení [m.s^(-2)]
ro=998;          % hustota kapaliny [kg.m^(-3)]
alfa=parD.alfa;  % hledané alfa

x0=g*parD.mk/parD.mp; % statický koeficient
x1=parD.V0*ro/parD.mk; % statický koeficient
x2=alfa/parD.mp;     % dynamický koeficient
pa=parD.pa;
hx=parD.hx;
tt=parD.tt;
dp=parD.dp;

h=H(1); v=H(2);
k=find(tt>t,1); % první větší index
p=dp(k-1)+(dp(k)-dp(k-1))/(tt(k)-tt(k-1))*(t-tt(k-1));

dv=x0*(1-x1*pa/(pa+p+ro*g*(hx+h)/1000))-x2*v; % diferenciální rovnice
dH=[v;dv]; % naplnění matice dH

return
```

