



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

HIL TESTOVACÍ STAV PRO SOUSTAVU UNIVERZÁLNÍCH ELEKTRONICKÝCH ŘÍDÍCÍCH JEDNOTEK

HIL TEST STAND FOR UNIVERSAL ELECTRONIC CONTROL UNITS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Štěpán Zouhar

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Brabc

BRNO 2019

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Bc. Štěpán Zouhar
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	Ing. Martin Brablc
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

HIL testovací stav pro soustavu univerzálních elektronických řídicích jednotek

Stručná charakteristika problematiky úkolu:

Problematika testování řídicích jednotek je velmi aktuálním tématem, které se z automotive a aerospace aplikací šíří napříč všemi průmyslovými odvětvími. Tato práce cílí na vývoj Hardware-in-the-loop přípravku pro testování soustavy řídicích jednotek vyvíjených v Mechatronické laboratoři. Přípravek bude schopen testovat jak HW, tak SW vybavení jednotek pomocí metodiky HIL a podobných in-the-loop testů.

Cíle diplomové práce:

- 1) Proveďte literární rešerši v oblasti metodiky testování řídicích jednotek a pokuste se zjištěné poznatky shrnout a zasadit do kontextu různých variant in-the-loop simulačních testů. U každé varianty (např. MIL,SIL,PIL,HIL ...) se pokuste navrhnout vhodnou definici a blokové schéma algoritmu.
- 2) Realizujte testovací přípravek, který umožňuje jednoduché připojení a výměnu testované jednotky pomocí ICT – In Circuit Testing v provedení Bed of Nails, a který umožňuje měřit i simulovat všechny vstupní a výstupní signály řídicí jednotky. Realizujte jednoduchý, ale plně automatizovaný test HW řídicí jednotky.
- 3) Realizujte komplexnější testovací sestavu kompatibilní s testovacím přípravkem, která umožňuje detailní testování jedné jednotky i připojení dalších jednotek, pro které budou připraveny vybrané periferie, senzory a akční členy. U jedné jednotky je možné provádět in-the-loop testy, u ostatních pouze funkční testy, pro které jsou připravené reálné periferie. Sestava musí obsahovat alespoň 2 DC motory s enkodery, 2 potenciometry, dostatečné množství tlačítek a indikačních LED.
- 4) Pomocí I/O karty MF624 a prostředí Matlab realizujte základní HIL test, který umožní plně otestovat schopnost řídicí jednotky řídit DC motor. Test bude plně automatizovaný, včetně nahrávání firmware řídicí jednotky.
- 5) Navrhněte a realizujte HIL test, který testuje chování jedné řídicí jednotky v rámci soustavy alespoň pěti řídicích jednotek.

Seznam doporučené literatury:

NELLES, Oliver. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Berlin: Springer, 2011. ISBN 978-364-2086-748.

LJUNG, Lennart. System identification: theory for the user. 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999. ISBN 978-0136566953.

VALÁŠEK, Michael. Mechatronika. Dot. 1. vyd. Praha: České vysoké učení technické, 1996. ISBN 80-010-1276-X.

NOSKIEVIČ, Petr. Modelování a identifikace systémů. Ostrava: Montanex, 1999. ISBN 80-722-50-0-2.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.

ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.

děkan fakulty

ABSTRAKT

Diplomová práce se zabývá testováním elektronických řídicích jednotek, a to jednak funkčními testy, při kterých je ověřován hardware a software, a jednak testovacími metodami Model in the Loop, Software in the Loop, Processor in the Loop a Hardware in the Loop.

V rámci praktické části byl vytvořen testovací přípravek pro testování funkčnosti dané řídicí jednotky. Přípravek je připojen k PC pomocí vstupně výstupní karty, testování je řízeno a vyhodnocováno skriptem v MATLABu. Celý proces testování je automatizován od počátečního nahrání testovacího firmware do testované jednotky, přes průběh celého testu až po nahrání bootloaderu. Dále byl vytvořen Hardware in the Loop test, při kterém řídicí jednotka plní roli regulátoru a DC motor je simulován v reálném čase na PC v prostředí MATLAB.

KLÍČOVÁ SLOVA

HIL, PIL, SIL, MIL, testování, Bed of Nails tester, funkční testy, Model Based Design, modelování, simulace

ABSTRACT

This diploma thesis is focused on testing of Electronic Control Units, especially functional testing in which hardware and software is verified and also Model in the Loop, Software in the Loop, Processor in the Loop and Hardware in the Loop testing methods.

Within practical part of this thesis testing stand for functional test of the ECU was developed and manufactured. It is connected to PC via Input/Output card, testing is controlled by MATLAB script. Whole process of testing is automated from initial upload of testing firmware to tested ECU over all phases of test up to bootloader flashing. Hardware in the Loop test was also created, in which ECU works as controller and DC motor is simulated in real time with PC in MATLAB environment.

KEY WORDS

HIL, PIL, SIL, MIL, testing, Bed of Nails tester, Functional testing, Model Based Design, modelling, simulation

BIBLIOGRAFICKÁ CITACE

ZOUHAR, Štěpán. HIL testovací stav pro soustavu univerzálních elektronických řídicích jednotek. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/116768>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Ing. Martin Brablec.

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci na téma HIL testovací stav pro soustavu univerzálních elektronických řídicích jednotek vypracoval samostatně s použitím odborné literatury a pramenů, uvedených v seznamu, který tvoří přílohu této práce.

V Brně dne

.....
Bc. Štěpán Zouhar

PODĚKOVÁNÍ

Děkuji tímto vedoucímu práce Ing. Martinovi Brablcovi za cenné připomínky a rady poskytnuté při vypracovávání závěrečné práce. Dále bych rád poděkoval Bc. Tomášovi Kirchnerovi, Bc. Romanovi Adámkovi za spolupráci na dílčích částech vytvořeného zařízení a Mechatronické laboratoři za poskytnutí zázemí pro tvorbu praktické části práce.

OBSAH

1	ÚVOD.....	11
1.1	Motivace	12
2	TEORETICKÁ ČÁST.....	13
2.1	Elektronické řídicí jednotky Dino (Master a Slave)	13
2.1.1	Testovaná řídicí jednotka (Slave).....	13
2.2	Modely vývoje	15
2.2.1	Vodopádový model (Watterfall)	15
2.2.2	V model.....	15
2.3	Model Based Design (MBD)	16
2.3.1	Model in the Loop (MIL)	18
2.3.2	Software in the Loop (SIL)	19
2.3.3	Processor in the Loop (PIL)	19
2.3.4	Hardware in the Loop (HIL)	20
2.3.5	HIL testovací stavy.....	21
2.4	Testování funkčnosti desek plošných spojů.....	23
2.4.1	In Circuit Testing	23
2.5	Spolupráce na testovacím přípravku.....	25
2.5.1	Elektronika	25
2.5.2	Testovací skript	25
2.5.3	Komplexní testovací soustava.....	26
3	PRAKTICKÁ ČÁST	27
3.1	Mechanická konstrukce testovacího přípravku.....	27
3.1.1	Pogo piny.....	27
3.1.2	Návrh mechanické konstrukce	29
3.1.3	Výpočet zatížení zavírací západky	31
3.1.4	Výroba konstrukce	33
3.2	Elektronika.....	34
3.2.1	SW pro automatizovaný test	37
3.3	HIL test regulátoru DC motoru.....	40
3.3.1	Model DC motoru	42
3.3.2	Aproximace funkce $\text{sgn}\omega$	45
3.3.3	Estimace parametrů modelu	46
3.3.4	Srovnání výsledků HIL testu a reálného experimentu	48
3.4	Komplexní testovací soustava.....	51
3.5	HIL test chování jednotky v rámci soustavy více jednotek	51
4	ZÁVĚR.....	53

SEZNAM POUŽITÝCH ZDROJŮ	54
SEZNAM OBRÁZKŮ	56
SEZNAM TABULEK.....	57
SEZNAM PŘÍLOH.....	58

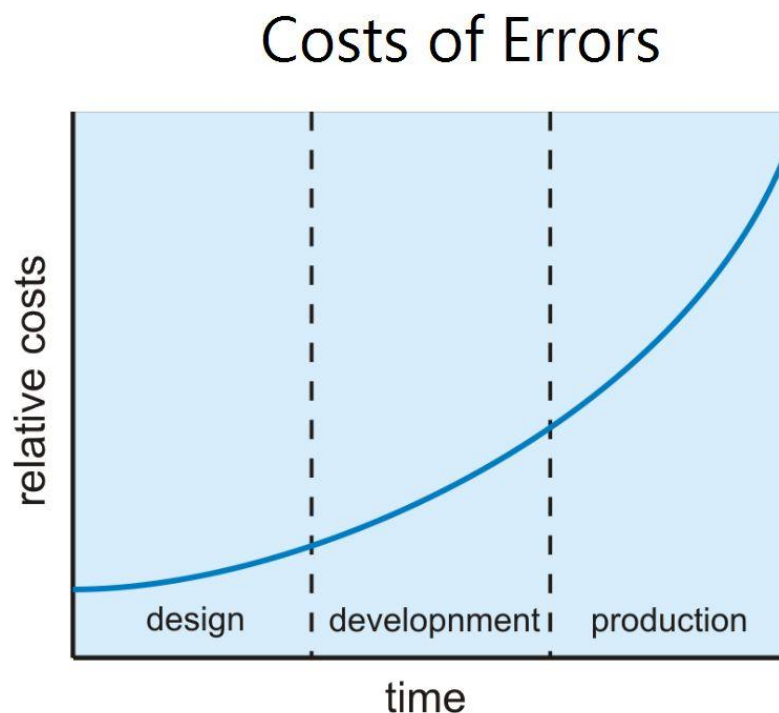
1 ÚVOD

HIL testování se používá ve velkém měřítku v automobilovém průmyslu. Důvod je prostý, současné automobily jsou již natolik komplexní, že klasické testovací metody selhávají.

Zatímco před půlstoletím bylo auto tvořeno výhradně mechanickými komponenty, obří pokrok v oblasti elektroniky umožnil nasazení mikrokontrolerů pro řízení všech významných funkcí od bezpečnostních systémů až po komfortní a multimediální výbavu současných automobilů. Aby automobilky udržely svoji konkurenceschopnost a přitom plnily všechny nové předpisy a nařízení, stala se inovace klíčem k úspěchu nejen v tomto odvětví průmyslu. Díky tomu, že cena za hardware se rok od roku snižuje, zatímco jeho výkon narůstá (podle Moorova zákona se každé dva roky zdvojnásobí počet tranzistorů na jednom integrovaném obvodu) je nyní snazší inovovat, než tomu kdy bylo v minulosti.

Podle Alliance of Automobile Manufacturers a jejich zprávy z roku 2014 se uvnitř průměrného vozidla nachází 60 mikroprocesorů, což je čtyřikrát víc, než to mu bylo před deseti lety. Uvnitř těch mikroprocesorů se v rámci jednoho vozidla nachází zhruba 10 milionů řádků zdrojového kódu. Výrobci aut neustále přidávají nové technologie do svých vozidel, každý rok je zaregistrováno několik tisíc patentů pouze v automobilovém průmyslu. Tyto technologie umožňují zvýšit bezpečnost, výkon, efektivitu, snížit spotřebu a emise, ale současně také velmi zvyšují složitost celého produktu, díky čemuž se moderní vozidlo stává velmi komplexním systémem. Vlivem tohoto zesložitování není inovace vůbec jednoduchý proces, protože všechny komponenty celého systému musí fungovat nejen každá samostatně, ale především ve spojení s ostatními do jednoho celku. Právě metoda Model Based Design umožňuje nejen výrobcům vozidel vyvíjet takto komplexní systémy a přitom přinášet na trh nové inovace. [1] [2]

Pokud se při návrhu nového zařízení objeví chyba, platí, že čím později je odhalena, tím větší jsou náklady na její odstranění, jak je znázorněno na obr. 1.1.



Obrázek 1.1: Závislost ceny odstranění chyby na čase jejího objevení [3]

Současně však rostoucí komplexnost systémů zvyšuje pravděpodobnost výskytu chyby. Problém u klasického způsobu návrhu zařízení spočívá v tom, že řada chyb se objeví až při implementaci řízení na soustavu, tedy téměř na konci vývojového procesu. Jejich odstranění je vlivem toho poněkud komplikované. Proto začaly vznikat nové způsoby návrhu zařízení, které zavádějí testování již v raných fázích vývoje, díky čemuž je možné chyby odhalit mnohem dříve. Jedním z těchto způsobů je metoda Model Based Design. V rámci validace a verifikace uvnitř této metody jsou používány různé druhy testování jako Model in the Loop, Software in the Loop, Processor in the Loop a Hardware in the Loop.

1.1 Motivace

Aby vůbec bylo možné provádět Hardware in the Loop testy s finálním zařízením, které bude nasazeno v běžném provozu, je nutné nejprve otestovat hardware a software tohoto zařízení. Při tomto testování je ověřována funkčnost vstupů a výstupů testované jednotky. Jedná se o činnost poněkud stereotypní a v případě testování velkých sérií i časově náročnou. Proto vznikla potřeba na vývoj a výrobu testovacího přípravku, který celý test automatizuje. Současně umožní snadnou výměnu testované jednotky bez nutnosti neustálého připojování a odpojování konektorů.

Výše popisovaný funkční test však nestačí pro ověření, zda řídicí jednotka je schopna řídit DC motor. Pokud bychom to chtěli ověřit na reálném motoru, je k tomu třeba výkonová elektronika a právě reálný motor, čímž se testovací řetězec rozšíří o další zařízení. Existuje však nespočet variant motoru a enkodéru, pokud by měla být jednotka testována s daným motorem, je nutné všechny typy držet skladem, což je ekonomicky nevýhodné. Testování na reálném motoru navíc neumožňuje sledovat chování regulátoru za zvláštních okolností a mezních stavů. Z toho důvodu vznikl požadavek na rozšíření testovacího přípravku o HIL test DC motoru, který umožní nahradit testování s reálným motorem právě jeho HIL simulací.

2 TEORETICKÁ ČÁST

2.1 Elektronické řídicí jednotky Dino (Master a Slave)

Jedná se o elektronické řídicí jednotky vyvinuté Mechatronickou laboratoří, které jsou určeny pro řízení složitých pohybů. Hlavní řídicí jednotka (Master) k sobě může mít připojených až 16 podřízených jednotek (Slave) prostřednictvím komunikační sběrnice RS 485. Současně je možné připojit hlavní jednotku do sítě Internet a vzdáleně k ní přistupovat. Jedna podřízená jednotka může v kombinaci s výkonovou elektronikou řídit jeden až dva pohony pomocí PWM či analogového výstupu. Podřízená jednotka znázorněná na obr. 2.1 je vybavena vstupy pro určování polohy motoru na základě digitálního enkodéru či analogového senzoru polohy.

Pohon může být řízen bez i se zpětnou vazbu. Buď je možné do jednotky naprogramovat požadovanou trajektorii nebo lze například nastavovat požadovanou polohu pomocí potenciometru připojeného na analogový vstup jednotky. Jednotky umožňují regulaci na polohu, na rychlost i momentovou regulaci při připojení čidla proudu.

Programování probíhá přes PC, ke kterému se připojuje hlavní jednotka prostřednictvím USB kabelu. Do této jednotky jsou nahrávány i programy pro podřízené jednotky a jejich přenos z hlavní jednotky do cílové podřízené probíhá po sběrnici RS 485. Jednotky jsou programovány grafickou formou, což nevyžaduje od uživatele pokročilé znalosti programování. Po naprogramování již PC k provozu zařízení není potřeba.

Původně byly jednotky vyvinuty pro řízení pohybů modelů dinosaurů v Dinoparku, nicméně je možné je nasadit v libovolném stroji. Jedná se o zajímavou alternativu k PLC. [4]

2.1.1 Testovaná řídicí jednotka (Slave)

Testovaná jednotka je vybavena následující piny: [4]

Vstupy:

- 4 digitální vstupy (DI)
- 2 digitální vstupy pro koncové spínače (LS)
- 1 analogový vstup pro čidlo otáček (AIM)
- 1 digitální vstup pro kanál A enkodéru (QE A)
- 1 digitální vstup pro kanál B enkodéru (QE B)
- 1 analogový vstup pro referenční napětí z čidla proudu (REF)
- 1 analogový vstup pro výstup z čidla proudu (CUR)
- 1 analogový obecný vstup (AIG)

Výstupy:

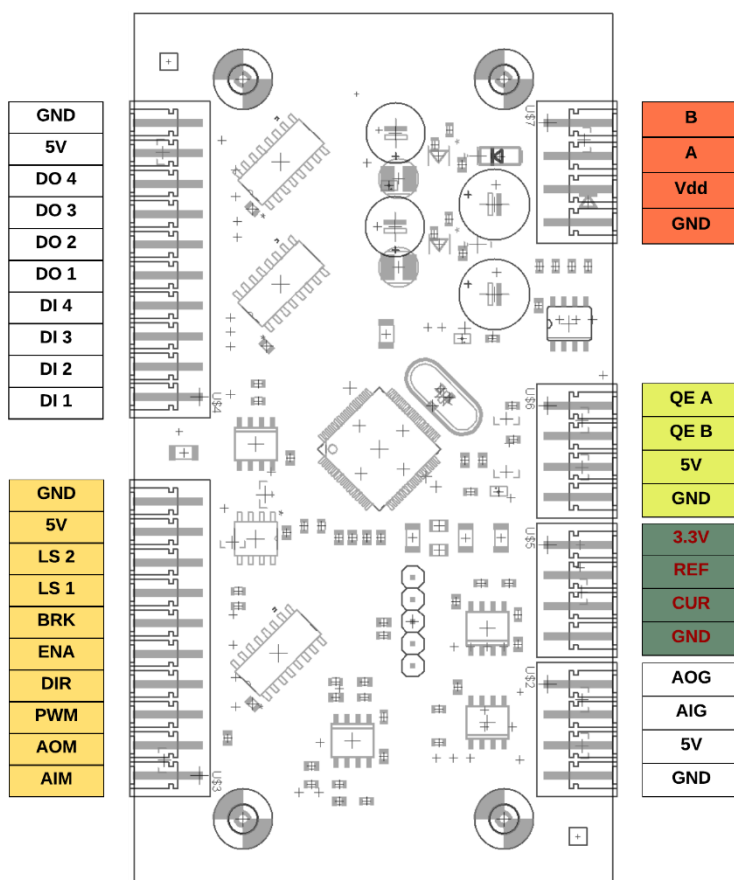
- 4 digitální výstupy (DO)
- 1 digitální výstup pro ovládání brzdy pohonu (BRK)
- 1 digitální výstup pro ovládání řízení pohonu (ENA)
- 1 digitální výstup pro řízení směru pohonu (DIR)
- 1 výstup pulzně šířkové modulace pro motor (PWM)
- 1 analogový výstup pro motor (AOM)
- 1 analogový obecný výstup (AOG)

Komunikace:

- 1 pin pro kanál A komunikace RS 485 (A)
- 1 pin pro kanál B komunikace RS 485 (B)

Napájení:

- 1 pin napájecího napětí 24 V (Vdd)
- 1 pin pro zem (GND)
- 4 piny pro napájení připojených komponent (5V)
- 4 piny pro zem připojených komponent (GND)



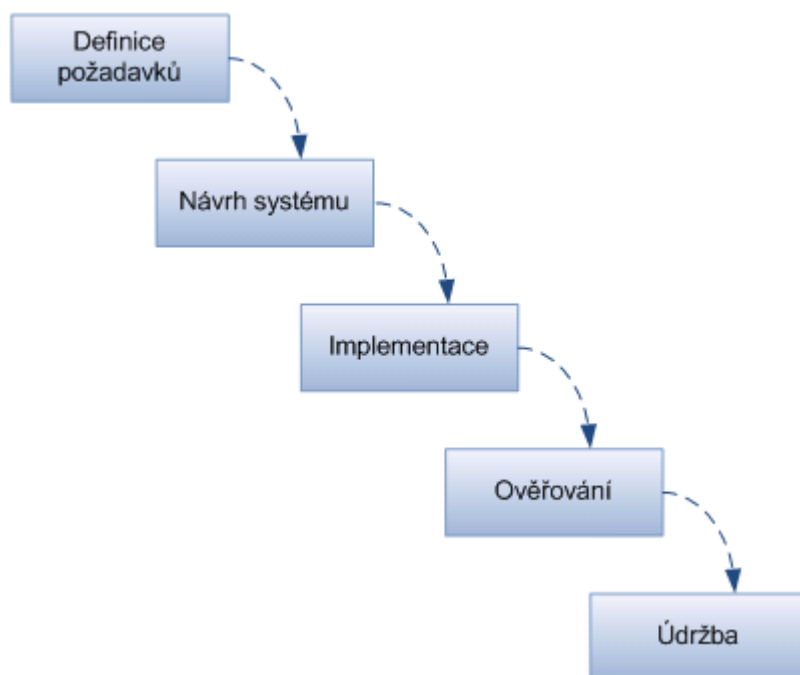
Obrázek 2.1: Testovaná řídicí jednotka (Slave) [4]

2.2 Modely vývoje

2.2.1 Vodopádový model (Waterfall)

Tato vývojová metoda byla standardem hlavně v 70. až 90. letech minulého století. Celý projekt je u ní rozložen na 5 částí, jako je sepsání požadavků, návrh, implementace, verifikace a údržba. Postupuje se chronologicky tím stylem, že nová fáze se zahájí až při ukončení té předcházející, jak je naznačeno na obrázku 2.2.

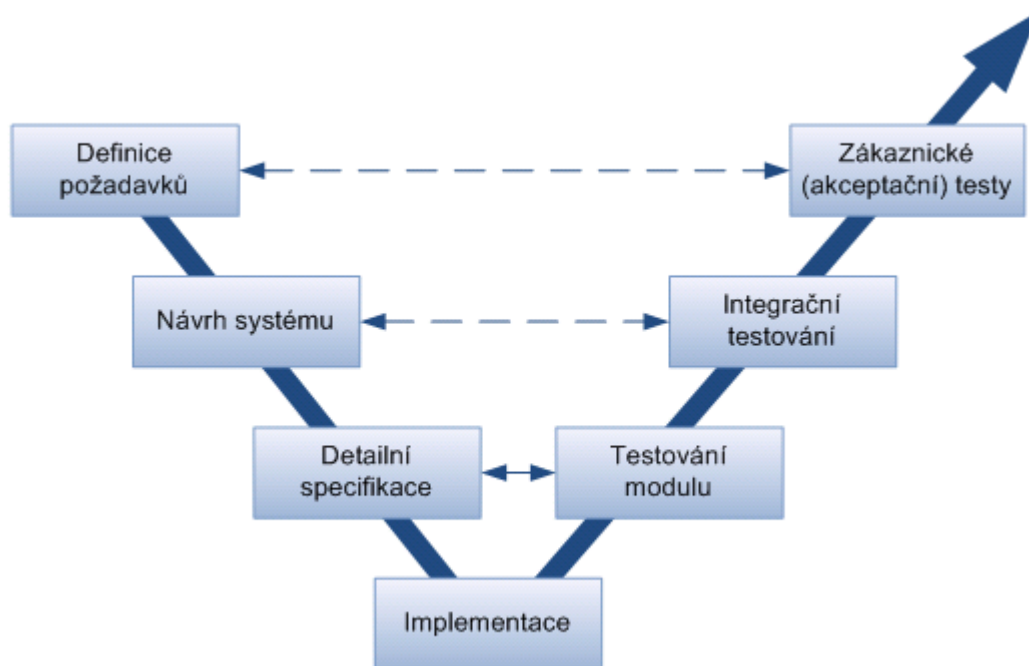
Projekt vedený tímto stylem lze snadno řídit a orientovat se v něm, nicméně použití tohoto stylu je značně limitující. Požadavky na zařízení totiž musí být definovány na začátku vývoje a následně již téměř není možné je měnit, v důsledku čehož se tato metoda velmi špatně přizpůsobuje na případné změny.



Obrázek 2.2: Vodopádový model vývoje [5]

2.2.2 V model

Základním kamenem V modelu je vodopádový přístup, který byl modifikován tak, že každé úrovni specifikace systému v levé části diagramu odpovídá určitý způsob testování v pravé větvi V diagramu, jak je naznačeno na obr. 2.3. Vodorovné čáry spojující jednotlivé úrovně V modelu představují verifikaci a validaci. K otestování každé fáze vývoje je učený jiný typ testu, přičemž se začíná testováním nejmenších částí a postupuje se až k testům celku. Díky testování v každé fázi vývoje je možné odhalit případné chyby včas.



Obrázek 2.3: V model [5]

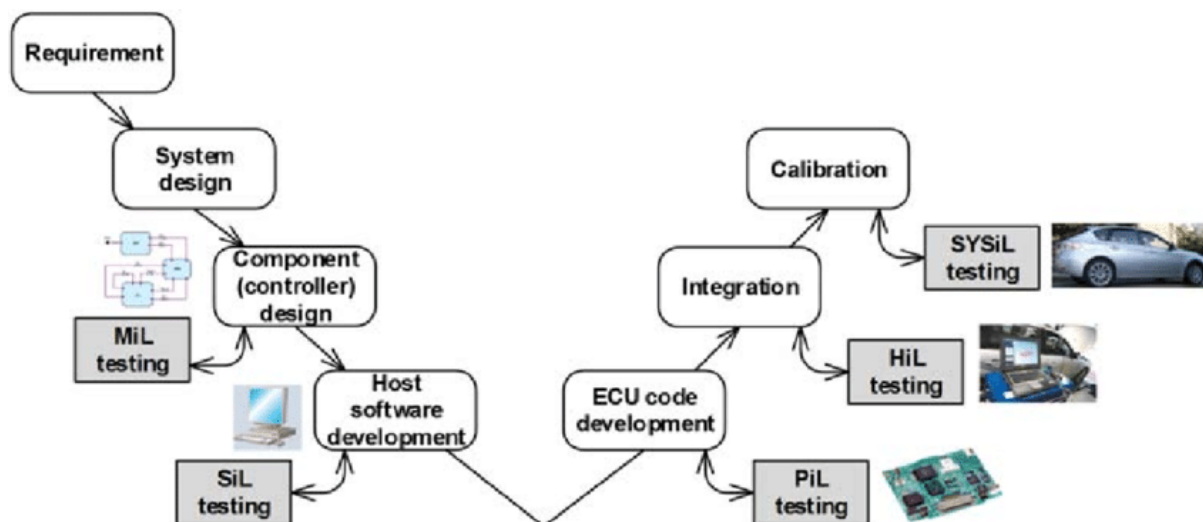
2.3 Model Based Design (MBD)

Model Based Design je metoda návrhu elektronických zařízení založená na použití matematických modelů. Modely musí zahrnovat všechny komponenty, které ovlivňují chování celého systému a rovněž musí být popsáno prostředí, ve kterém celý systém pracuje.

Namísto specifikací v textovém dokumentu, na kterých je založen tradiční vývojový proces, používá metoda Model Based Design modely k implementaci specifikací a parametrů komponent, tzn. dokumentace (datasheet) dané komponenty ve formě textu je nahrazen modelem komponenty. Modely mohou zahrnovat mnohem více detailů než psané specifikace, umožňují popsat charakteristiky, mohou být sdíleny mezi inženýry pracujícími v různých oblastech a na rozličné úrovni návrhového procesu.

Modely jsou používány ve všech 4 fázích vývoje: výzkumu, návrhu, implementaci i verifikaci a validaci. V každé fázi vývoje je model aktualizován. [6]

Vodopádový i V model slouží primárně k vývoji software, nicméně se ukázalo, že V diagram lze snadno používat i k vývoji vestavěných zařízení např. v automobilovém průmyslu. Model Based Design aplikovaný na V diagram je znázorněn na obr. 2.4.



Obrázek 2.4: Model Based Design aplikovaný na V diagram [7]

V důsledku toho, že kontrolér je navržen a odladěn během simulací MIL, SIL, PIL a HIL, v okamžiku implementace na reálném HW je již většina logických chyb odstraněna. Chyby se však mohou objevit v případě, že model není dostatečně přesný.

Díky metodě založené na modelování systémů a automatickému generování kódu je umožněno navrhovat systémy a jejich kontroléry i inženýrům, kteří znají fyzikální podstatu daného systému, ale nejsou příliš zdatní v programování. [6]

Pokud srovnáme metodu Model Based Design s CAD softwarem, zjistíme, že mají mnoho společného. Obě techniky umožní vytvořit virtuální prototyp již v raných fázích vývoje ještě dlouho předtím, než bude vůbec vyroben první reálný prototyp. Zatímco návrhem v CAD prostředí může konstruktér ověřit, zda budou zvolené mechanické komponenty vzájemně kompatibilní a budou tedy splněny požadavky na vlastnosti a rozměry výsledného produktu, metoda Model Based Design umožňuje vývojářům softwaru díky simulacím ověřit, že zvolené řídicí algoritmy budou fungovat v kombinaci s řízenou fyzikální sestavou.

Nespornou výhodou Model Based Design je, že model systému není omezen pouze na jednu oblast návrhu, ale může popisovat komplexní systémy skládající se např. z mechaniky, elektřiny, elektroniky, hydrauliky, pneumatiky, software atd. Snadno tedy je možné modelovat multidiménové systémy.

Velkou výhodou Model Based Design také je, že pomocí této metody lze rychle prototypovat a jednou vytvořené modely lze v budoucnu znovu použít pro obdobné aplikace.

Z modelů lze automaticky přímo vygenerovat zdrojový kód pro jednotlivé komponenty systému, což ušetří spoustu práce v porovnání s ručním psáním software. Díky tomu lze navíc eliminovat chyby, které by mohly vzniknout při ručním psání kódů. Na základě modelů lze provádět různé varianty testů již během samotného návrhu a není tak nutné čekat s návrhem řízení až do okamžiku, kdy bude fyzicky k dispozici první verze celého systému.

Výhody, které Model Based Design přináší, jsou následující: [1]

- Zkrácení času potřebného pro vývoj produktu
- Snížení rizik při testování
- Možnost testování rizikových stavů
- Automatizace časově náročných úkolů a testů

- Snížení počtu prototypů potřebných pro vývoj produktu
- Zvýšení kvality produktu
- Matematické modelování představuje jednoznačný jazyk pro komunikaci mezi inženýry
- Testování metodou HIL přináší v porovnání s testováním na reálné soustavě vyšší bezpečnost, snadnou opakovatelnost testů

Verifikace a validace

Pro verifikaci a validaci se používají metody Software in the Loop, Processor in the Loop a Hardware in the Loop. Je navržena sada opakovatelných testů, které jsou spouštěny a výsledky zaznamenávány. Na základě nich jsou modely vylepšovány. Po každé změně kontroléru nebo modelu soustavy je nutné opakovat stejné testy znovu a ověřit, zda se systém chová podle očekávání. [6]

Filozofie pro tvorbu modelů

Nejdříve je nutné začít s jednoduchým modelem jedné komponenty. Následně porozumět chování dané komponenty. Poté je možné navrhnout jednoduchý kontrolér. Před samotným spuštěním simulace by inženýr měl předpovědět, jak se bude systém chovat. Jednoduchý systém včetně kontroléru je tedy možné simulovat a ověřit, že se chová podle očekávání. Pouze pokud je splněna tato podmínka, by měl být celý proces opakován a model zesložitován až na úroveň potřebnou pro dostatečný popis celé soustavy. Vždy je však nutné provést jednu změnu, model odsimulovat, ověřit, že se chová podle očekávání a až poté provádět změnu další. [6]

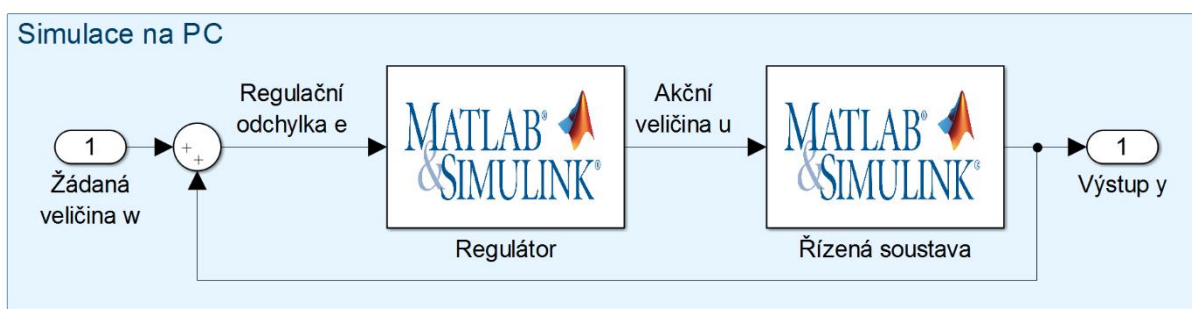
Aby model nebyl příliš výpočtově náročný, provádí se při modelování určitá zjednodušení, v důsledku kterých model sice nepopisuje realitu dokonale, nicméně pro prvotní fáze návrhu regulátoru je to bohatě dostačující.

2.3.1 Model in the Loop (MIL)

Jedná se o první testovací metodu používanou při návrhu nového zařízení. Metoda Model in the Loop slouží k rychlému ověření, zda je možné řídit danou soustavu navrhaným typem regulátoru.

Regulátor i řízená soustava jsou modelovány a simulovány v simulačním prostředí na PC (v našem případě Simulink), jak je znázorněno v blokovém schéma na obr. 2.5.

Simulace během MIL nemusí běžet v reálném čase a je možné použít variabilní délku kroku simulace. [8]



Obrázek 2.5: Blokové schéma Model in the Loop

2.3.2 Software in the Loop (SIL)

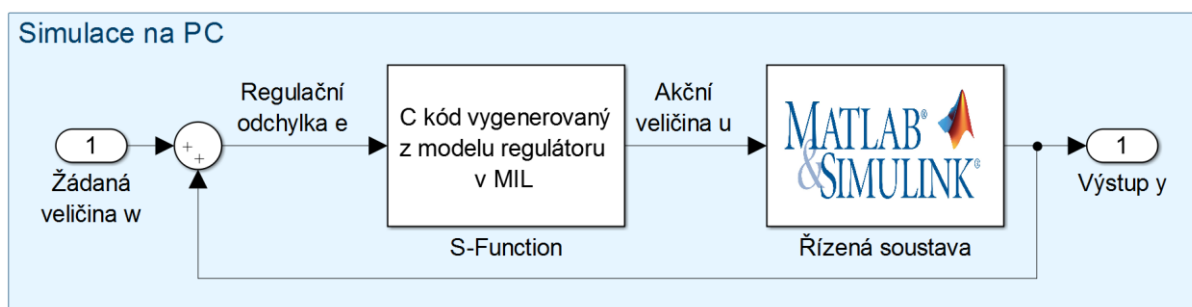
Software in the Loop testování přímo navazuje na Model in the Loop. Z modelu MIL se převezme blok regulátoru, nastaví se u něj pevný časový krok, odpovídající řešič a jeho parametry, pevná šířka všech signálů a regulátor se převede do aritmetiky s pevnou řádovou čárkou.

Kompilátor z modelu regulátoru následně vygeneruje Embedded C kód, blok regulátoru se nahradí blokem S-Function s tímto vygenerovaným kódem. Ten je totožný s kódem, který bude následně použit při implementaci na reálném HW. Jediný rozdíl spočívá v tom, že reálný HW má analogové vstupy a výstupy, zatímco kód uvnitř regulátoru reprezentovaného blokem S-Function pracuje se vstupy a výstupy v modelu v prostředí Simulink, jak je znázorněno v blokovém schéma na obr. 2.6.

Jedná se tedy o verifikaci generovaného kódu. SIL simulace současně slouží pro ověření, že nedochází ke kvantizačním chybám, saturaci či přetečení proměnných. [9]

Pro ověření správnosti srovnáváme regulaci původního modelu a nového modelu. Výstupy systému by se pro stejné vstupy neměly lišit.

Software in the Loop simulace běží v reálném čase a na základě ní je snazší určit potřebný výkon HW pro danou aplikaci. [10]



Obrázek 2.6: Blokové schéma Software in the Loop

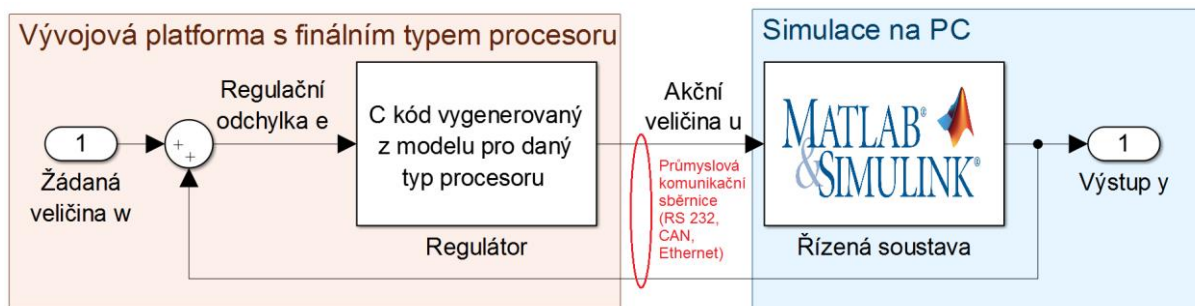
2.3.3 Processor in the Loop (PIL)

V metodě testování Processor in the Loop je model kontroléru kompilovaný pro daný typ procesoru, který bude použit ve finálním produktu. Regulátor již není pouze simulován na PC, ale jeho kód běží na stejném procesoru, který bude ve finálním zařízení. Pro tuto fázi testování však není nutné mít dokončený návrh elektroniky řídicí jednotky, kód regulátoru obvykle běží na vývojové platformě s daným typem procesoru. K automatickému generování C kódu ze Simulinku slouží nástroj Simulink Coder (dříve Real-Time Workshop).

Testování Processor in the Loop může, ale nemusí probíhat v reálném čase. Testování v reálném čase je však výhodnější, protože umožňuje ověřit, zda je HW pro danou aplikaci dostatečný nejen po stránce výkonu. Dále lze stanovit minimální možnou periodu řídicího cyklu i posoudit, jak kvalitní je regulace řízené soustavy.

U Processor in the Loop testů probíhá komunikace mezi kontrolérem a řízenou soustavou prostřednictvím komunikačních sběrnic jako RS 232, CAN či Ethernet, jak je naznačeno v blokovém schéma na obr. 2.7. Simulace se spouští z PC, na kterém je vytvořen model řízené soustavy. [11]

Testováním Processor in the Loop se provede verifikace daného procesoru například z pohledu velikosti zabírané paměti, výpočtové náročnosti vygenerovaného kódu atd. [9]



Obrázek 2.7: Blokové schéma Processor in the Loop

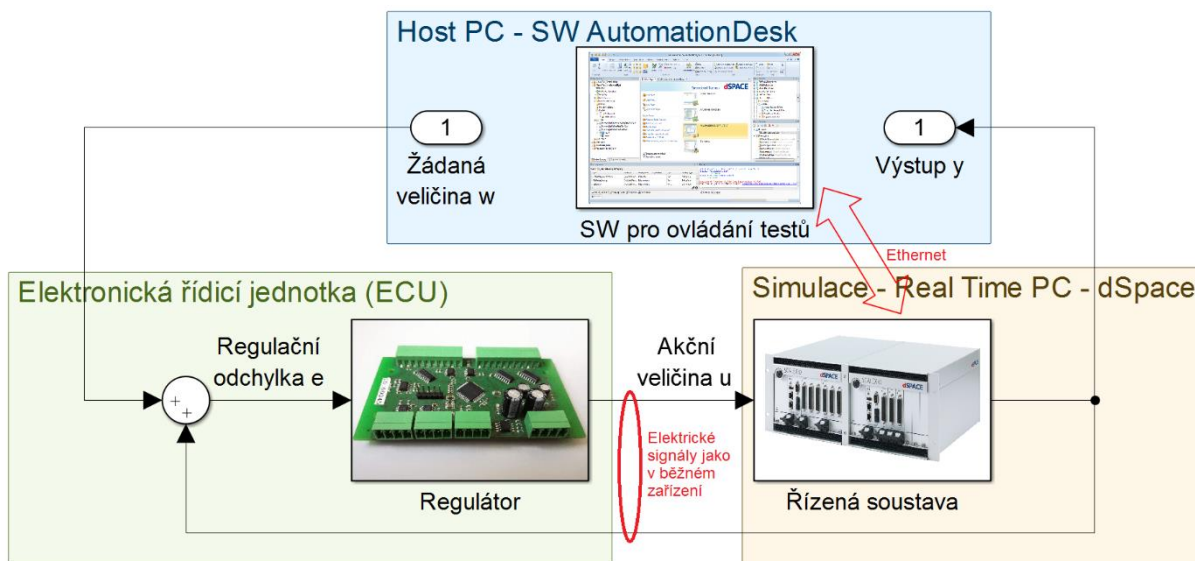
2.3.4 Hardware in the Loop (HIL)

Testování metodou Hardware in the Loop navazuje na PIL s tím rozdílem, že regulátor je implementován na hardware, který bude použit ve finální aplikaci, zatímco reálná soustava je i v tomto případě nahrazena modelem, nyní však musí být simulován v reálném čase. Schema testování metodou Hardware in the Loop je znázorněno na obr. 2.8.

Mezi regulátorem a řízenou soustavou jsou přenášeny elektrické signály jako v běžném zařízení, Real Time PC musí tedy na svém vstupu mít mimo jiné A/D převodníky pro čtení vstupních signálů do řízené soustavy a D/A převodníky pro posílání výstupních signálů z řízené soustavy zpátky do regulátoru. [11]

Testy se spouští z PC se speciální softwarem určeným pro tyto účely např. AutomationDesk od firmy dSpace. V software jsou vytvořeny nejrůznější testovací případy, SW nastavuje požadované veličiny a zaznamenává výstupní veličiny, aby mohlo být následně vyhodnoceno, zda se celý systém chová dle očekávání. Komunikace mezi PC pro ovládání testů a Real Time počítačem, na kterém běží model řízené soustavy, probíhá prostřednictvím Ethernetu, díky tomu je možné spouštět, zastavovat simulaci, ale i měnit model řízené soustavy. Pomocí Ethernetu může být připojena i elektronická řídicí jednotka s regulátorem, díky čemuž bývá zpravidla možné například měnit regulační parametry během testování.

Testování HIL se nejvíce přibližuje reálnému provoznímu stavu, jedná se o poslední fázi před vývojem prototypu. Díky tomu, že je však řízená soustava pouze simulována, lze u tohoto testování dosáhnout kritických stavů, které si při testování řízení reálné soustavy není možné dovolit, ať už z důvodu, že by došlo k destrukci zařízení, újmě na zdraví či významným finančním ztrátám.



Obrázek 2.8: Blokové schéma Hardware in the Loop

2.3.5 HIL testovací stavy

HIL testovací stav je speciální zařízení, které umožňuje simulovat řízenou soustavu v reálném čase. Používají se především v automobilovém, leteckém a energetickém průmyslu, nicméně v posledních letech se tento přístup k navrhování komplexních výrobků rychle šíří i do dalších průmyslových odvětví.

Mezi nejznámější výrobce testovacích stavů patří společnosti dSpace, National Instruments, MicroNova a řada dalších. Testovací stavy se vyrábí v různých velikostech od nejmenších, které umožňují testovat například jednu řídicí jednotku až po velké testovací stavy, na kterých je možné simulovat celé zařízení. Větší testovací stavy jsou zpravidla montovány do standardizovaných skříní (racků). Celý systém bývá obvykle velmi modulární, takže je například možné pospojovat tři testovací stavy do jednoho celku, přičemž jeden může sloužit k simulaci spalovacího motoru, druhý k simulaci bezpečnostních funkcí a třetí je určen pro komfortní elektroniku. Takto vzniklou kombinací simulátorů lze simulovat podstatnou část vozidla. S rozvojem elektromobility se již delší dobu na trhu objevují také HIL simulátory systému baterií určené pro elektromobily. Ukázka skříňového provedení HIL testovacího stavu je na obrázku 2.9.



Obrázek 2.9: Hardware in the Loop simulátor ve skříňovém provedení (racku) [12]

Pokud se HIL simulace doplní o vizualizaci a vstupní zařízení, kterým ovládá uživatel zařízení v běžném provozu, je možné používat HIL simulátor i jako trenažér například pro výcvik operátorů či ladění řídicího algoritmu. Díky tomu, že jsou operátoři zaučováni na virtuálním prototypu, se výrazně snižuje riziko poškození drahého strojního zařízení. Operátoři si navíc mohou natrénovat zvládnutí krizových situací, jejichž nácvik by byl v běžném prostředí buď velmi nákladný, protože by mohlo dojít k destrukci celého zařízení, nebo by při nich mohlo dojít k újmě na zdraví. Na obrázku 2.10 je znázorněn HIL test řízení jeřábu na lodi. Díky připojení dálkového ovládacího zařízení, které je určeno pro ovládací jeřábu, k HIL simulaci a věrohodné animaci celého procesu lze tento HIL test používat jako trenažér právě například pro výcvik operátorů. Velkou výhodou tohoto trenažéru je také to, že lze snadno měnit podmínky testu.



Obrázek 2.10: HIL simulátor jako testovací trenažér [13]

2.4 Testování funkčnosti desek plošných spojů

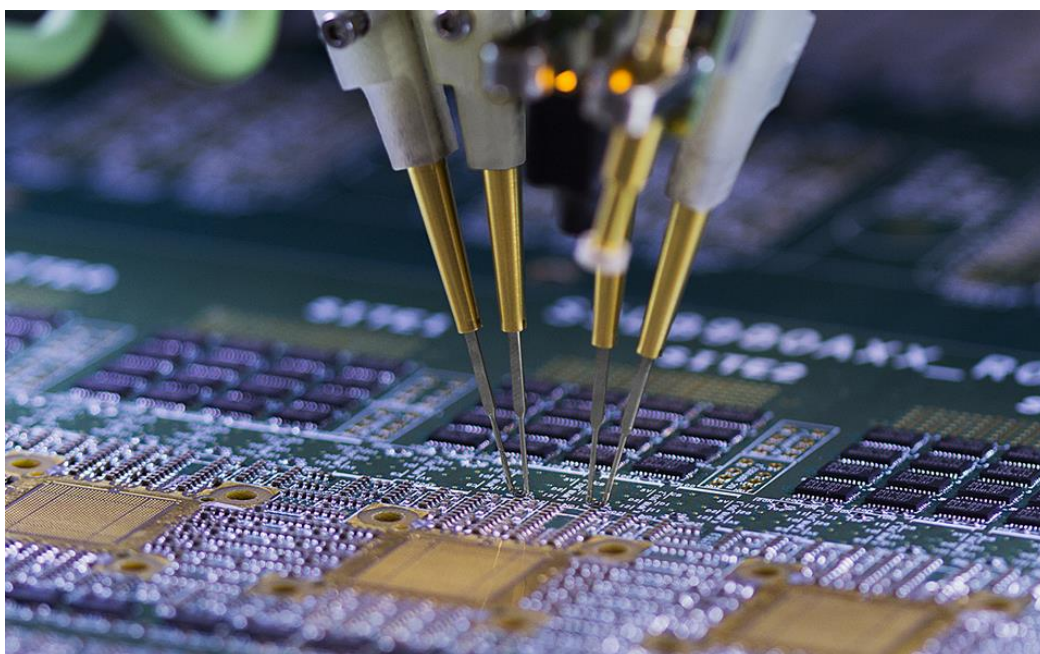
Testovací zařízení desek plošných spojů se obecně používají pro detekci výrobních defektů, jako je například zkrat či přerušená vodivá cesta, měření pasivních a aktivních součástek, orientace součástek, záměna součástek atd.

2.4.1 In Circuit Testing

Kontaktní testování osazených desek plošných spojů se označuje zkratkou ICT – In Circuit Testing. Mimo ICT existují ještě bezkontaktní metody testování jako například AOI – Automated Optical Inspection, AXI – Automated X-ray Inspection aj. U bezkontaktních metod je snímána deska pomocí kamery a obraz je vyhodnocován tak, že se v něm vyhledává, zda na desce nechybí nějaká komponenta či není nesprávně orientovaná nebo zdeformovaná.

Zařízení sloužící k testům ICT se dají rozdělit do dvou základních kategorií:

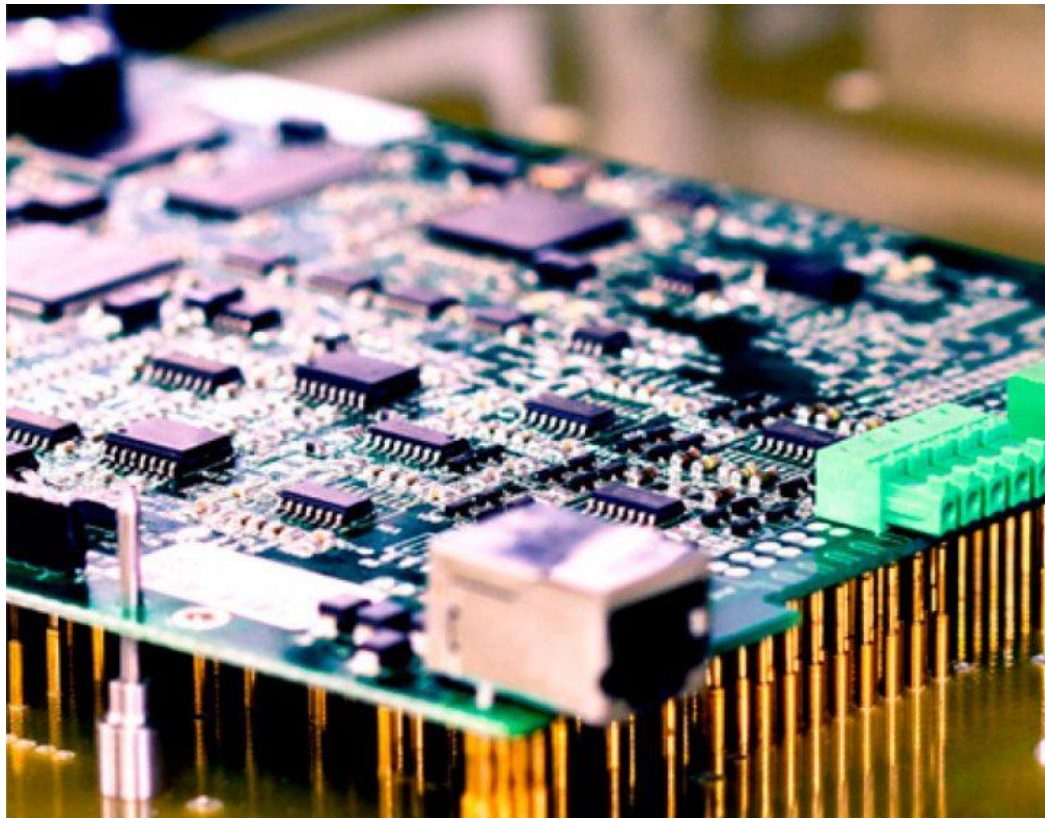
- a) **Testovací zařízení typu Flying Probe** – používají pohyblivé sondy, které se vždy přemístí do místa nad DPS, kde testovací zařízení bude aktuálně měřit, následně se sondy spustí níž, aby došlo ke kontaktu testované DPS se sondami, a dojde k přeměření vodivé cesty mezi sondami viz obrázek 2.11. Tento typ testovacího zařízení je sice univerzálnější, protože není určen pro testování jedné konkrétní DPS, hodí se však spíše pro testování malých až středních sérií, protože cena za test jedné DPS je vyšší než u druhé varianty, což je dáno tím, že test metodou Flying Probe trvá zpravidla déle, protože je nutné testovat postupně jednu komponentu po druhé. Výhodou metody je však, že v případě nového typu testované desky stačí přeprogramovat testovací zařízení na základě CAD návrhu DPS a není tak nutné vyvíjet celé zařízení od začátku, jako je tomu u druhé metody. [14]



Obrázek 2.11: Testování DPS metodou Flying Probes – pohyblivé sondy [15]

- b) **Testovací zařízení typu Bed of Nails** – je určené pro jeden konkrétní typ DPS. Na začátku jsou sice vyšší nároky na vývoj zařízení, nicméně následné náklady na test jednoho kusu desky jsou již nízké a test relativně rychlý. Vzájemná poloha

testované desky vůči testovacímu přípravku je zajištěna pomocí vodících kolíků, na které se nasune testovaná deska. V místech na testované desce, na které se chceme připojit (například tam, kde jsou konektory, vývody, prokovy, pájící plošky), jsou umístěny v testovacím přípravku pogo piny, což jsou základní konstrukční prvky testovacích zařízení tohoto typu viz obrázek 2.12.



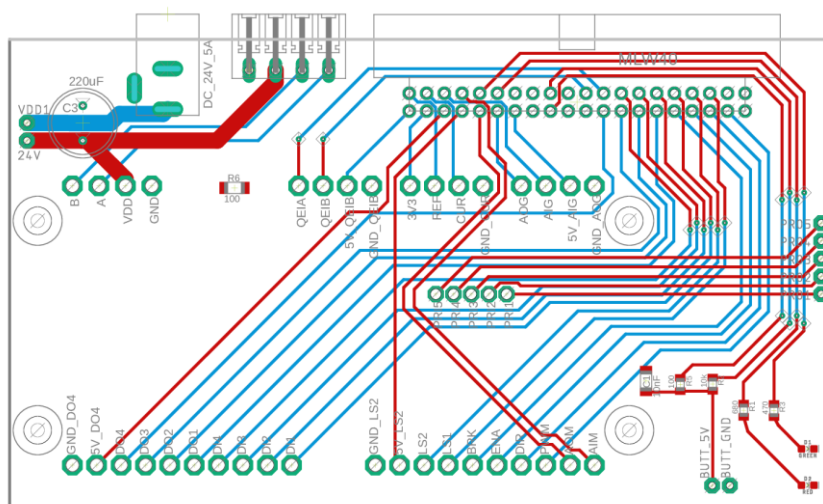
Obrázek 2.12: Testování DPS metodou Bed of Nails [16]

2.5 Spolupráce na testovacím přípravku

Při návrhu testovacího přípravku bylo na dvou dílčích úkolech spolupracováno s kolegy. Návrhem elektroniky se zabýval Tomáš Kirchner, při tvorbě testovacího algoritmu jsem skript, ze kterého jsem následně vycházel a rozšiřoval jej, obdržel od Romana Adámka.

2.5.1 Elektronika

Návrh desky plošných spojů testovacího přípravku byl vytvořen Tomášem v prostředí EAGLE. Následně jsme opakovaně konzultovali polohu jednotlivých součástek, montážních otvorů aj., dokud jsme se nedostali k finální verzi DPS, ta se následně nechala vyrobit, poté ji Tomáš osadil. DPS testovacího přípravku je znázorněna na obrázku 2.13.



Obrázek 2.13: Návrh DPS testovacího přípravku

2.5.2 Testovací skript

Software pro testování funkčnosti testované jednotky byl vytvořen v prostředí MATLAB. V době kompletnosti přípravku měl již Roman připraven program pro základní test vstupů a výstupů.

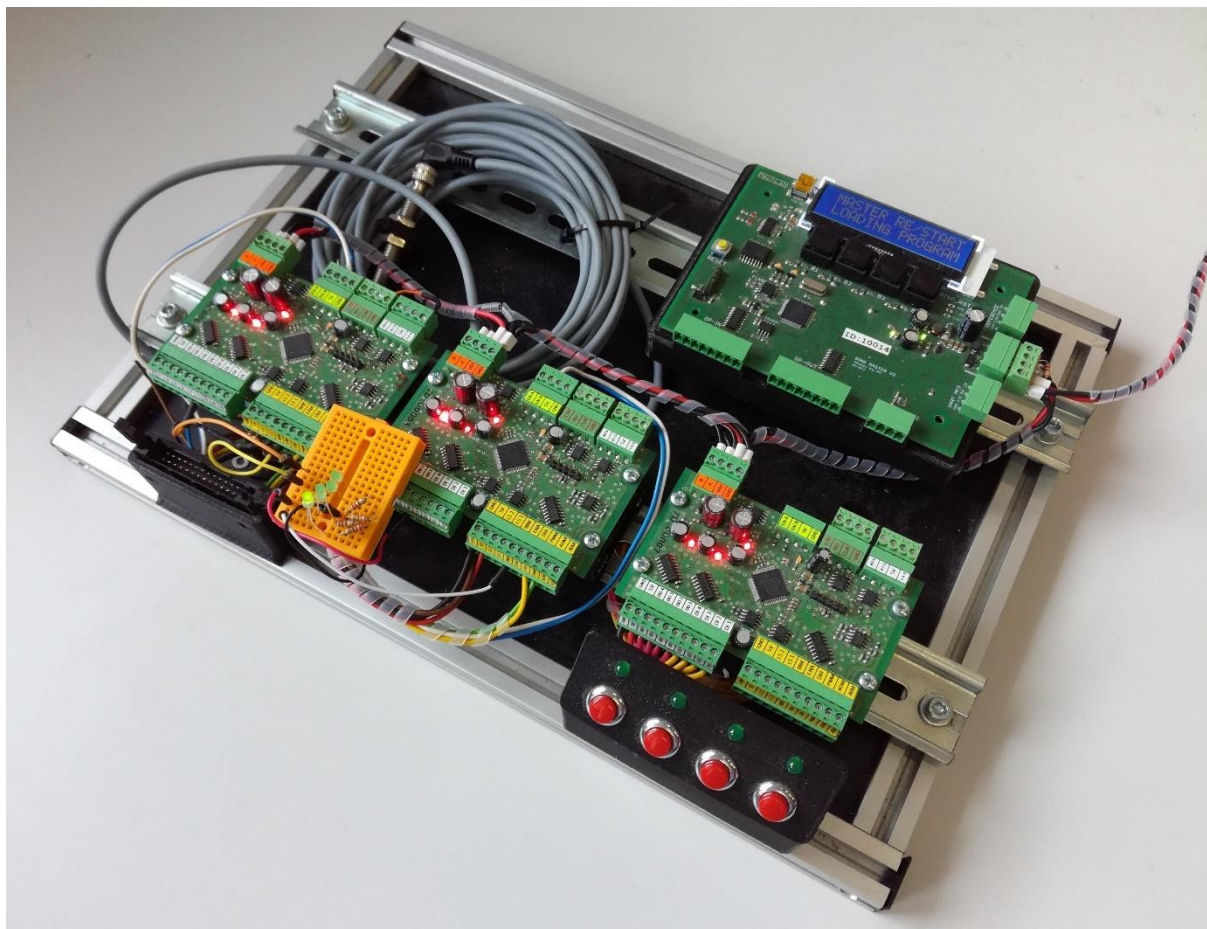
Na začátku skriptu proběhne inicializace karty a nastaví se časový krok na 1 ms, test tedy běží na frekvenci 1 kHz. Nejprve se testují digitální vstupy a výstupy. Vygenerují se náhodné testovací digitální signály o délce 1 s a test spočívá v tom, že se na jednotlivé výstupy karty, tedy vstupy testované jednotky, postupně zapisují hodnoty testovacích signálů a současně jsou vyčítány digitální vstupy karty, tedy výstupy testované jednotky. Po proběhnutí testu digitálních vstupů a výstupů se vyhodnotí počet promeškaných tiků časovače, vykreslí se srovnání digitálních signálů poslaných a přijatých kartou a pro všechny testované piny se spočítá střední kvadratická odchylka. Pokud vyjde u některého odchylka větší, než je nastavená povolená mez, vyhodnotí se daný pin jako vadný.

Na test digitálních vstupů a výstupů navazuje test analogových vstupů a výstupů. Průběh této části testu je shodný s předchozím testováním digitálních vstupů a výstupů. Nejprve se vygenerují sinusové testovací signály, následně se zapisuje na výstupy karty a čte ze vstupů karty. Protože má jednotka víc analogových vstupů než výstupů, je v testovacím firmwaru naprogramováno, že signál na analogovém vstupu CUR se má sečíst se signálem z obecného analogového vstupu AIG a hodnota vzniklá součtem vstupů se nastaví na obecný analogový výstup AOG. Po skončení této části testu se opět provede vyhodnocení funkčnosti pinů na základě střední kvadratické odchylky.

Poslední část základního testu tvoří ověření funkčnosti digitálních vstupů pro kanály enkodéru A, B a PWM výstup pro řízení motoru. Testovací signály enkodérů trvají 5 s, přičemž v první polovině předbíhá signál kanálu A signál kanálu B a ve druhé naopak signál z kanálu B předbíhá signál kanálu A. Tyto signály se postupně zapisují na odpovídající vstupy testované jednotky, jednotka na to reaguje PWM signálem generovaným na pinu PWM, tento signál je filtrován dolní propustí a její výstup je snímán kartou MF624. Po skončení poslední části testu proběhne vyhodnocení funkčnosti testovaných pinů QEA, QEB a PWM, ale i celého testu jednotky. Pokud jsou všechny piny v pořádku, jednotka je označena jako funkční, pokud se objeví jedna či více chyb, jednotka je označena jako vadná.

2.5.3 Komplexní testovací soustava

K testování řídicích jednotek již v minulosti byl v Mechatronické laboratoři vyvinut testovací přípravek, který umožňuje testovat 1 jednotku Master a 3 jednotky Slave. Přípravek však neumožňuje snadnou výměnu testované jednotky, proto vznikl požadavek na vývoj a výrobu nového testovacího zařízení typu Bed of Nails. Fotka testovacího zařízení je zobrazena na obr. 2.14.



Obrázek 2.14: Komplexní testovací soustava

3 PRAKTICKÁ ČÁST

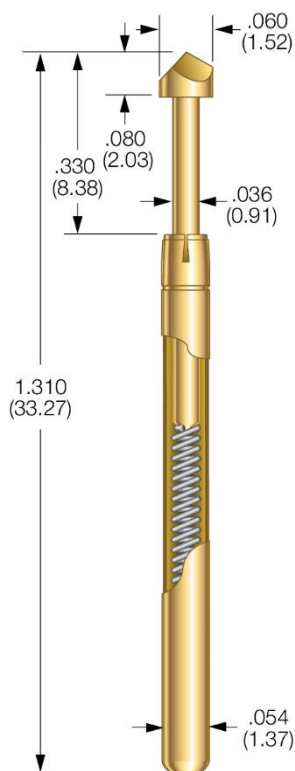
Hlavním cílem praktické části této diplomové práce je výroba přípravku, jenž slouží k otestování funkčnosti hardware elektronické řídicí jednotky Slave popsané v teoretické části práce a znázorněné na obr. 2.1. Test spočívá ve zkoušce vstupů a výstupů, přičemž se jedná o vstupy a výstupy jednak analogové, jednak digitální.

3.1 Mechanická konstrukce testovacího přípravku

Protože je testovací přípravek v mém případě určen pouze pro jeden typ testované DPS a jedním z požadavků je, aby test probíhal rychle, bylo rozhodnuto, že bude vyroben testovací přípravek typu Bed of Nails.

3.1.1 Pogo piny

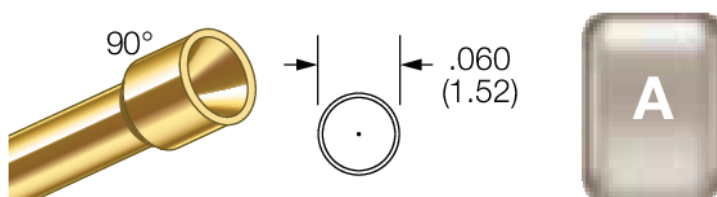
Pogo pin použitý na testovacím přípravku je znázorněn včetně rozměrů v palcích (v závorkách jsou uvedeny rozměry v mm) na obrázku 3.1, označení použitého pogo pinu je ICT-100-A-5.5-G S/C.



Obrázek 3.1: Náčrtes pogo pinu použitého na testovacím přípravku včetně rozměrů [17]

Spodní část pogo pinu je pevná a připevňuje se do testovacího přípravku buď pájením – v případě, že testovací přípravek má vlastní DPS, nebo je uložen v testovacím přípravku nalisováním do otvoru, jedná se tedy o uložení s přesahem. Obvykle se pogo pin nepřipevňuje přímo do testovacího přípravku, ale použijí se upevňovací pouzdra, což jsou speciální trubičky, které se připevní do testovacího přípravku a do nich se potom nasadí samotný pogo pin. Výměna pogo pinu v případě poškození je tedy velmi snadná, pogo pin stačí vytáhnout a vyměnit za jiný. I na mém testovacím přípravku byl použit tento způsob připevnění prostřednictvím upevňovacích pouzder s označením R-100-CR.

Uvnitř spodní pevné části pogo pinu je pružina, která tlačí horní pohyblivou část, trn, směrem nahoru. Horní konec pohyblivé části přichází do kontaktu s testovanou DPS. Vyrábí se různé varianty provedení horního konce, na obrázku 3.1 je znázorněna varianta se zakončením ve tvaru čtyřbokého jehlanu, ta se hodí, jestliže je na testované DPS průchozí otvor, se kterým má pogo pin přijít do kontaktu. V mém případě je však potřeba vytvořit dočasný elektrický kontakt mezi testovacím přípravkem a testovanou DPS v místech, kde se nachází zapájené piny konektorů, výše uvedené provedení je tedy pro můj případ nevyhovující a proto byl zvolen válcový konec s otvorem ve tvaru kuželu o vrcholovém úhlu 90° viz obrázek 3.2. Tento kužel pomáhá k navedení pogo pinu do správné pozice při dosedání testované desky do testovacího přípravku.



Obrázek 3.2: Válcové provedení konce pogo pinu s otvorem ve tvaru jehlanu [17]

Vzhledem k tomu, že na navrhovaném testovacím přípravku musí být celkem 41 pogo pinů, respektive na 41 míst na testované desce se chceme připojit, je nutné kalkulovat i se silou od pogo pinů. Podle dokumentace (datasheet) jmenovitá síla pružiny závisí na materiálu, ze kterého je pružina vyrobená. Z označení použitého typu pogo pinu lze zjistit, že materiálem pružiny je materiál používaný na výrobu hudebních strun, jmenovitá síla je $M_{jm} = 5,5$ oz a předpětí $M_{pr} = 2,4$ oz. Přepočít jednotky hmotnosti 1 oz na kilogramy je následující:

$$1 \text{ oz} = 0,0283495 \text{ kg}$$

V našem případě je tedy jmenovitá síla pogo pinu:

$$F_{jm} = M_{jm} \cdot g \cdot 0,0283495 = 5,5 \cdot 9,81 \cdot 0,0283495 = 1,53 \text{ N}$$

Předpětí pogo pinu pak vychází:

$$F_{pr} = M_{pr} \cdot g \cdot 0,0283495 = 2,4 \cdot 9,81 \cdot 0,0283495 = 0,67 \text{ N}$$

Jmenovitá síla se uvádí při stlačení pogo pinu o $x = 4,32$ mm, přičemž maximální stlačení je $x_{max} = 6,35$ mm.

Je vhodné spočítat sílu pružiny pogo pinu při maximálním možném stlačení, bude tedy uvažováno, že pružina má lineární charakteristiku a pro sílu pružiny platí vztah:

$$F = k \cdot x$$

V našem případě je neznámá tuhost pružiny, vyjádříme ji tedy z rovnice:

$$k = \frac{F}{x}$$

Za sílu F musíme dosazovat rozdíl sil při jmenovitém stlačení F_{jm} a předpětí F_{pr} :

$$k = \frac{F}{x} = \frac{F_{jm} - F_{pr}}{x} = \frac{1,53 - 0,67}{0,00432} = 199,57 \frac{\text{N}}{\text{m}}$$

Síla pružiny při maximální stlačení $x_{max} = 6,35$ mm se určí vztahem:

$$F_{max} = F_{pr} + k \cdot x_{max} = 0,67 + 199,57 \cdot 6,35 = 1,93 \text{ N}$$

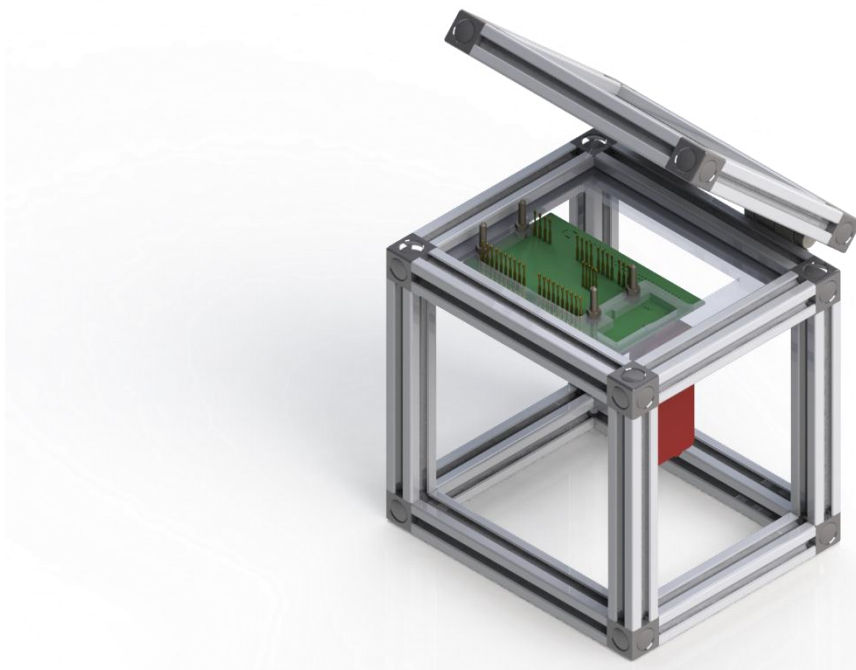
3.1.2 Návrh mechanické konstrukce

Na začátku samotného návrhu mechanické konstrukce testovacího přípravku jsem zvažoval různé varianty provedení. Požadavky na konstrukci byly takové, aby výrobní náklady nebyly příliš vysoké, respektive aby většinu dílů bylo možné vytisknout na 3D tiskárně, kterou máme k dispozici.

První verze konstrukce počítala s tím, že téměř všechny díly budou vytištěné na 3D tiskárně. Ve spodní části přípravku měly být pogo piny, na které se přiloží testovaná deska a přítlačná síla potřebná k dosažení požadovaného elektrického spojení mezi pogo piny a testovanou deskou bude zajištěna prostřednictvím rychloupínky, která se používá například pro zajištění vzájemné polohy dílů ve svařovacím přípravku. U těchto rychloupínek pouhým pohybem madla o 90° dojde k upnutí a odepnutí kusu.

Vzhledem k tomu, že jsem měl však obavy z toho, že by hlavní tělo konstrukce vytištěné na 3D tiskárně nemuselo mít dostatečnou pevnost a postupem času by tedy mohlo dojít k deformaci celého přípravku, jsem se následně rozhodl vytvořit zcela nový návrh.

Hlavním stavebním prvkem druhé verze návrhu měly být hliníkové profily systému Kombi, které mají po stranách drážky určené pro vsunutí matic, prostřednictvím kterých lze spojovat více profilů dohromady či připevňovat k hliníkové konstrukci další konstrukční díly. Vytvořil jsem tedy návrh konstrukce šroubované z hliníkových profilů v CAD software SolidWorks, jak je naznačeno na obr. 3.3.



Obrázek 3.3: Druhá verze konstrukce testovacího přípravku

Spodní část ve tvaru krychle je tvořena rámem z hliníkových profilů, které jsou v rozích spojeny pomocí spojovacích dílů. V horní stěně základní krychle je do drážek profilů vsunutá deska z akrylátu (běžně se nazývá také jako plexisklo). Ze spodní strany na desku dosedá DPS testovacího přípravku, která má v sobě zapájená upevňovací pouzdra pro pogo piny. Protože se

testovaná deska přikládá z horní strany akrylátové desky, musí mít akrylátová deska v sobě otvory pro pogo piny, které současně drží pogo pin ve správné poloze, aby nedocházelo k vybočení pogo pinu při jeho stlačení. Současně musí mít akrylátová deska otvory v místech, kde se nachází vodící trny pro testovací desku, které současně slouží pro připevnění DPS testovacího přípravku k akrylátové desce. V akrylátové desce musí být rovněž otvory v místech, kde jsou na DPS testovacího přípravku z horní strany připájené součástky, aby mohla horní plocha DPS ležet přímo na dolní ploše akrylátové desky.

Protože síla od pogo pinů při jejich stlačení není zanedbatelná, není vhodné, aby veškerá síla od pružin pogo pinů byla přenášena do konstrukce testovacího přípravku pouze skrze pájené spoje. Z tohoto důvodu byl navržen ještě díl sloužící k podpoře pogo pinů ze spodní strany DPS testovacího přípravku. Konstrukce tohoto dílu byla uzpůsobena tomu, že bude díl vyráběn technologií 3D tisku. Jedná se o díl, který má jednak 4 díry skrz, do kterých se vloží vodící trny, které plní současně roli šroubového spojení mezi podporou pogo pinů, DPS testovacího přípravku a akrylátovou deskou, jednak má na horní straně 41 děr, do kterých se vsunou pogo piny a síly od pogo pinů se tedy bude přenášet skrz stěny těchto otvorů do celé konstrukce.

Ze spodní strany DPS testovacího přípravku je připojený programátor PICkit 3, který slouží pro nahrávání testovacího firmware do testované desky plošných spojů a je znázorněn na obrázku 3.3 červenou barvou.

Spodní část konstrukce je prostřednictvím pantů spojena s horním pohyblivým víkem. Po vložení testované DPS na vodící trny se víko zavře, čímž dojde k dostatečnému zatlačení testované DPS do pogo pinů a vznikne tak elektrické spojení mezi testovanou DPS a DPS testovacího přípravku potřebné pro provedení testu.

Druhá verze konstrukce již plnila všechny požadavky na tuhost i mechanickou odolnost, nicméně ani tato varianta nakonec nebyla zvolena, protože by výrobní náklady byly příliš vysoké.

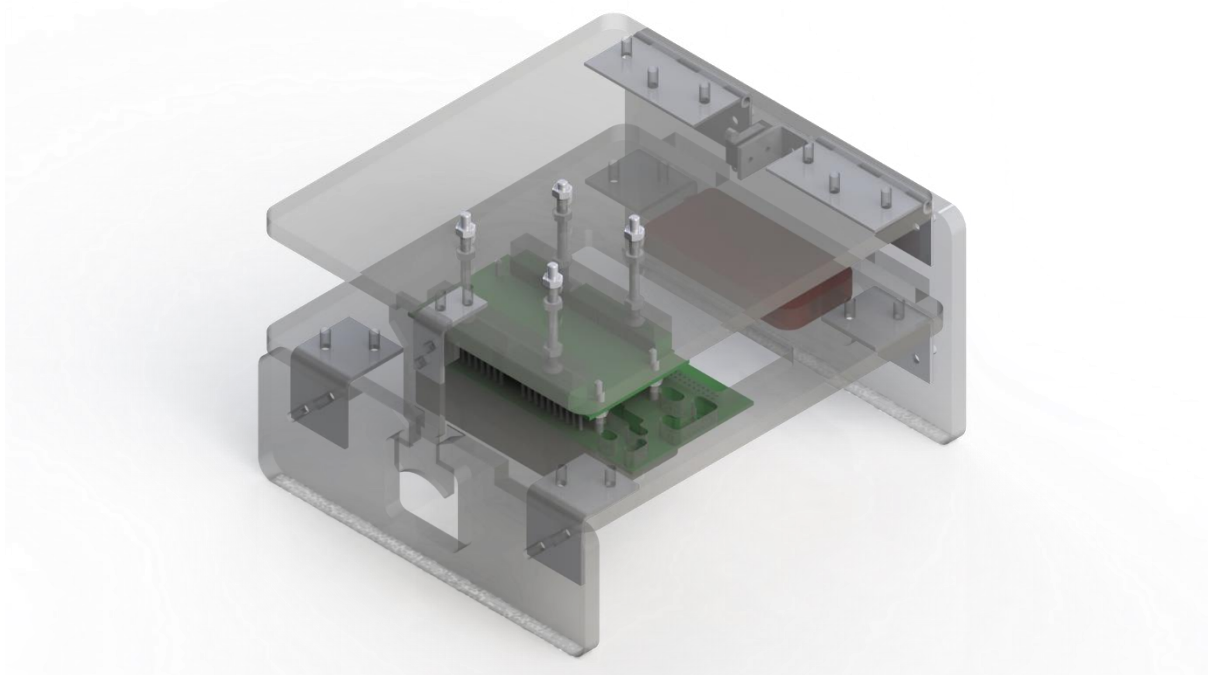
Zvažoval jsem tedy ještě další varianty, které by přicházely v úvahu, a při průzkumu trhu s testovacími zařízeními jsem narazil na zajímavý Kit pro testování DPS od firmy Merifix Electronics. Jedná se o zařízení velmi podobné mé druhé variantě, ale s tím rozdílem, že základním konstrukčním prvkem nejsou hliníkové profily, ale právě akrylátové desky a překližka. [18]

Inspiroval jsem se tedy jejich řešením a vytvořil nový návrh na míru našim potřebám a naší testované DPS. Hlavní součásti byly navrženy z akrylátu o tloušťce 8 mm, který je dostatečně pevný. Na zavírání, které musí být pružné, byl použit akrylát o tloušťce 3 mm. Západka je namyšlená tak, že se při zavírání víko samo zajistí a pro otevření je nutné západku vyhnout směrem dopředu. Spojení akrylátových desek je realizováno úhelníky z hobby marketu, horní víko se pohybuje na pantech. Pro zajištění vzájemné polohy testované desky vůči pogo pinům na testovacím přípravku byly použity šrouby M4, které jsou na konci osoustružené na menší průměr, aby bylo možné testovanou DPS na trny nasunout, protože montážní otvory uvnitř testované DPS mají průměr pouze 3 mm.

Na horním víku jsou přišroubované 2 obdélníkové díly z akrylátu, které dosedají na testovanou jednotku a tlačí ji směrem dolů. Vzdálenost těchto přitlačných dílů od horního víka lze měnit pomocí šroubů a přípravek tak lze tedy nastavit tak, aby stlačení pogo pinů při zavření přípravku bylo dostatečné, ale současně nebyly pogo piny stlačeny na maximum. Tyto obdélníkové díly se dotýkají testované DPS v místě konektorů a tlačí tedy testovanou DPS proti

pogo pinům právě v místech, kde se nachází pogo piny, takže nedochází k nežádoucímu mechanickému namáhání testované DPS a jejímu průhybu či krutu. Díl pro podporu pogo pinů ze spodní strany DPS testovacího přípravku byl stejně jako u druhé varianty konstrukce navržen tak, aby bylo možné jej vytisknout na 3D tiskárně. Spodní část pogo pinů je tedy uložena v podpěrném dílu vyrobeném 3D tiskem, následně jsou pogo piny připájeny do DPS testovacího přípravku a horní konec pevné části je ještě veden ve správném směru otvory v akrylátové desce. Těmito opatřeními je zajištěno, že nemůže dojít k vybočení pogo pinu a dosednutí horního konce pogo pinu do špatného místa na testované desce.

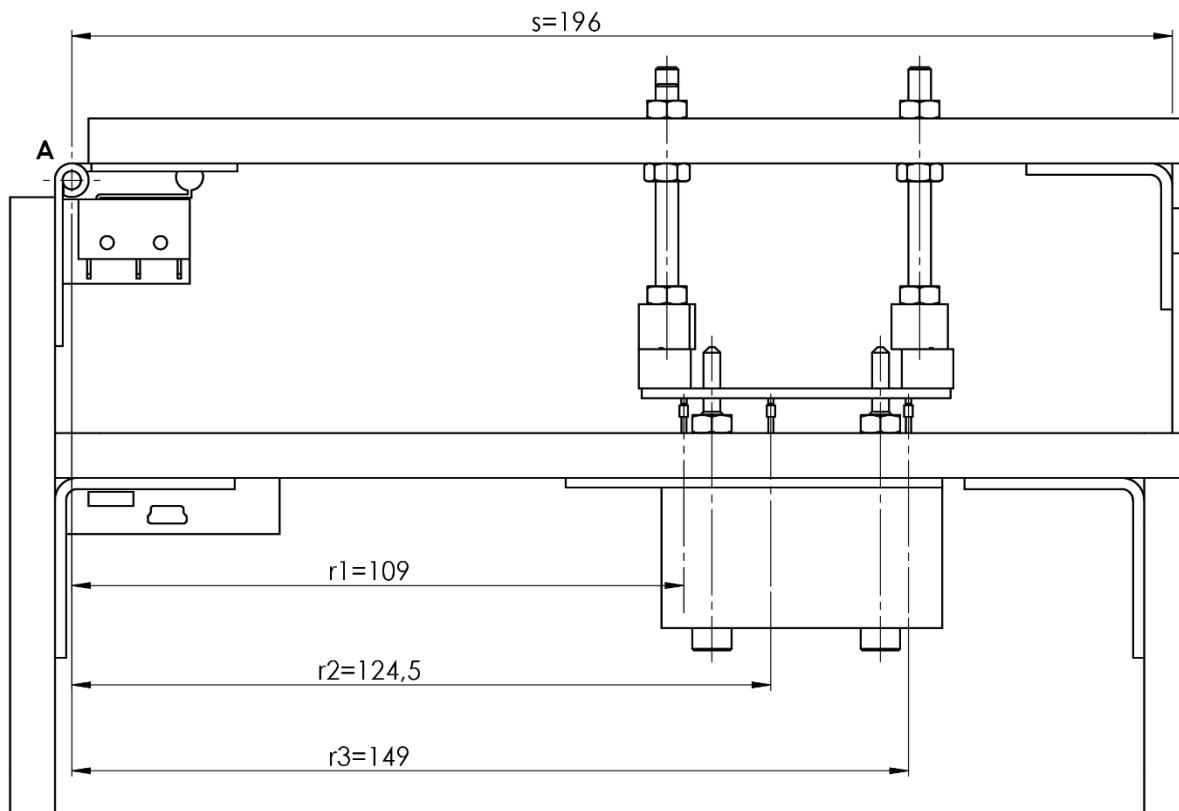
Mezi panty se nachází spínač, který slouží k odpojení napájení DPS testovacího přípravku v případě, že je horní víko otevřeno. Třetí varianta návrhu konstrukce testovacího přípravku je znázorněna na obr. 3.4.



Obrázek 3.4: Třetí verze konstrukce testovacího přípravku

3.1.3 Výpočet zatížení zavírací západky

Pro ověření, jak velkým napětím bude zatíženo horní víko a západka zavírání, byla spočítána síla působící na západku. Zatížení od jednoho pogo pinu budeme uvažovat maximální možné, tedy $F_{max} = 1,93$ N. Levý pohled na testovací přípravek i se zakótováním vzdálenosti pogo pinů a západky od pantů je znázorněn na obrázku 3.5. Ve vzdálenosti $r_1 = 109$ mm od pantů se nachází na testované jednotce $n_1 = 16$ pogo pinů, ve vzdálenosti $r_2 = 124,5$ mm potom $n_2 = 5$ pogo pinů a nejdál od pantů – ve vzdálenosti $r_3 = 149$ mm leží zbývajících $n_3 = 20$ pogo pinů. Vzdálenost západky od osy pantů je $s = 196$ mm.



Obrázek 3.5: Levý pohled na testovací přípravek se zakotovaným umístěním pogo pinů

Momentová rovnice k bodu A (ose pantů) je následující:

$$\Sigma M_A = F_{max} \cdot r_1 \cdot n_1 + F_{max} \cdot r_2 \cdot n_2 + F_{max} \cdot r_3 \cdot n_3 - F_{zav} \cdot s = 0$$

Pokud vyjádříme sílu působící na zavírání F_{zav} , dostaneme:

$$F_{zav} = \frac{F_{max} \cdot r_1 \cdot n_1 + F_{max} \cdot r_2 \cdot n_2 + F_{max} \cdot r_3 \cdot n_3}{s} = \frac{1,93 \cdot 0,109 \cdot 16 + 1,93 \cdot 0,1245 \cdot 5 + 1,93 \cdot 0,149 \cdot 20}{0,196} = 52,78 \text{ N}$$

Šířka západky v nejužším místě je $a = 14 \text{ mm}$, tloušťka akrylátu je $t = 3 \text{ mm}$. Průřez v tomto místě je tedy:

$$S_{zav} = a \cdot t = 14 \cdot 3 = 42 \text{ mm}^2$$

Tahové napětí v západce se určí následovně:

$$p_{zav} = \frac{F_{zav}}{S_{zav}} = \frac{52,78}{42} = 1,26 \text{ MPa}$$

Mez pevnosti akrylátu v tahu je $51,7 \text{ MPa}$ [19], zatížení západky je tedy téměř zanedbatelné a způsobuje mnohonásobně menší napětí, než je maximální možné dovolené napětí.

3.1.4 Výroba konstrukce

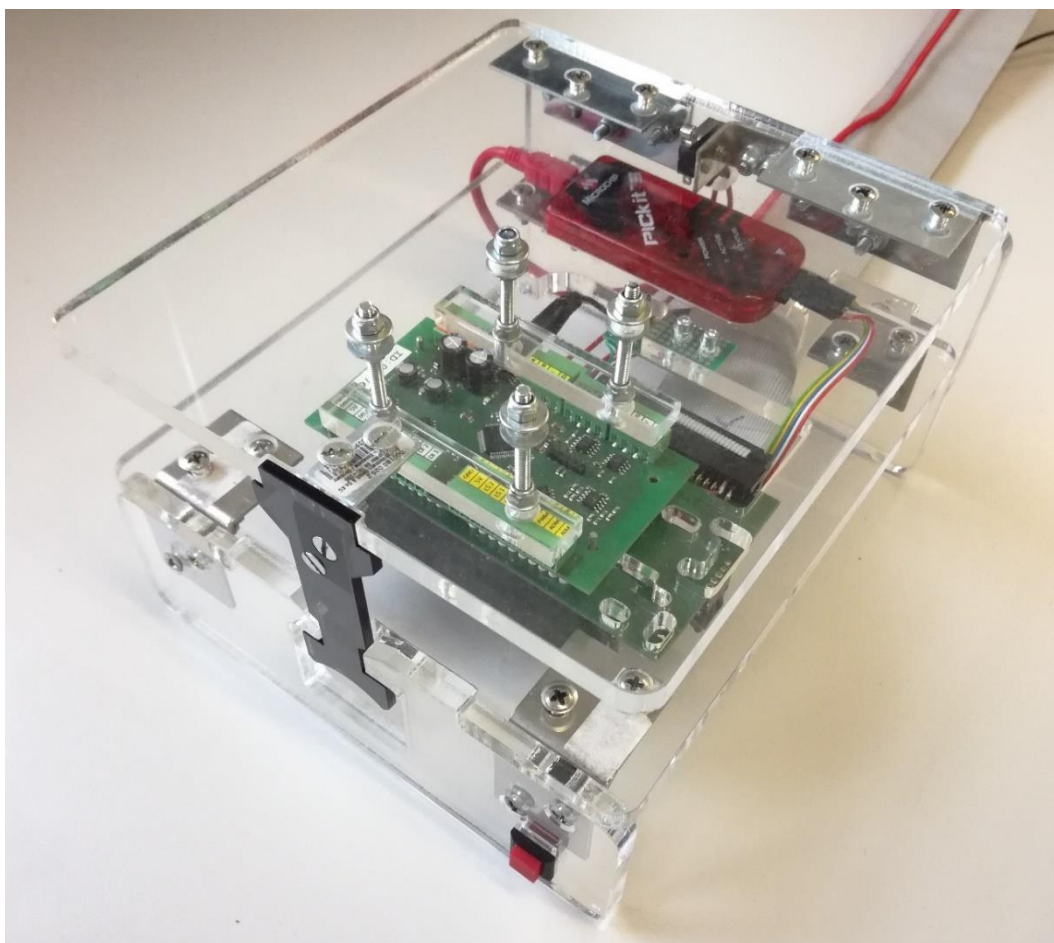
Protože tato varianta již splnila všechny požadavky na tuhost konstrukce i výrobní náklady, byly na základě modelu sestavy přípravku v programu SolidWorks připraveny podklady pro výrobu. U všech dílů z akrylátu jsem vygeneroval jejich geometrii do formátu dxf. Soubory tohoto formátu jsem následně naimportoval do programu Inkscape, ze kterého je již možné je poslat na laserovou řezačku.

Díly z akrylátu jsem vypálil ve Fablabu, kde disponují laserovou řezačkou Trotec Speedy 300 o výkonu 60 W. Tento laser je určen pro řezání a gravírování do dřeva, gumy, plastů, skla aj. Řezání akrylátu o tloušťce 8 mm je sice mírně komplikované, nicméně na více průchodů se to dá zvládnout, jen jsou hrany akrylátů po řezání poněkud ostré, ale to bylo vyřešeno přebroušením hran pálených dílů.

Součást přenášející zatížení od pogo pinů do konstrukce testovacího zařízení a současně sloužící k vedení pogo pinů ve správném směru byla zhotovena technologií 3D tisku v rámci Mechlabu.

Vodící trny plní současně roli spojovacího prvku mezi akrylátovou deskou, DPS testovacího přípravku a součástí k podpoře pogo pinů. Vzal jsem tedy šrouby M4 a konce osoustružil tak, aby na šroubu zůstal dostatečný počet závitů a současně bylo možné na osoustružený konec nasadit testovanou DPS.

Po zhotovení všech komponent byla mechanická konstrukce složena. Výsledek je zachycen na obrázku 3.6.



Obrázek 3.6: Fotka složeného testovacího přípravku

3.2 Elektronika

Spodní strana navržené DPS je podepřena vytištěnou podporou, která v sobě obsahuje otvory pro pogo piny. Tyto slepé díry o průměru 1,6 mm a hloubce 20 mm však není možné vytisknout na 3D tiskárně s dostatečnou přesností, průměry děr byly tudíž navrženy záměrně menší, aby zůstalo na stěnách několik desetin milimetru materiálu na následné převrtání vrtákem, čímž se dosáhne požadovaného rozměru i tvaru díry. Při tomto převrtávání byla na podpěrný díl přiložena DPS testovacího přípravku, čímž se zajistilo, že rozteče otvorů pro pogo piny v obou dílech budou souhlasné. Obdobně bylo nutné svrtat podle DPS i otvory pro pogo piny v akrylátové desce. Následně byla DPS znovu přiložena na podpěrný díl, do kterého byly zasunuty upevňovací pouzdra pogo pinů. Díky spodnímu podpěrnému dílu byla zajištěna správná poloha a upevňovací pouzdra tak mohly být připájeny z horní strany DPS. Poté následovalo vyjmutí podpěrného dílu a zapájení i ze spodní strany. V ten okamžik již nic nebránilo osazení DPS zbývajícími součástkami a konektory.

Protože má samotný test funkčnosti testované DPS běžet v prostředí MATLAB na PC vybaveném vstupně výstupní kartou MF624 od firmy Humusoft, je na DPS testovacího přípravku konektor MLW40. Do tohoto konektoru se připojuje kabel vedoucí do konektoru X1 karty MF624. Popis toho, který pin karty je přiveden na který pin testované DPS, je zaznamenán v tabulce 1.

Tabulka 1: Přiřazení pinů karty MF624 na piny testované DPS

Číslo pinu na kartě MF624	Pin na kartě MF624	Pin na testované DPS (Slave)
1	AD0	5V (QEIB)
2	AD1	3V3
3	AD2	5V (DO4)
4	AD3	5V (LS2)
5	AD4	AOM
6	AD5	AOG
7	AD6	5V (AIG)
8	AD7	Výstup RC filtru
9	AGND	Zem RC filtru
10	DA6	Nevyužito
11	DA7	Nevyužito
12	DIN0	DO4
13	DIN1	DO3
14	DIN2	DO2
15	DIN3	DO1
16	DIN4	BRK
17	DIN5	ENA
18	DIN6	DIR
19	DIN7	Tlačítko na přípravku
20	DA0	REF
21	DA1	CUR

22	DA2	AIG
23	DA3	AIM
24	DA4	LED GREEN
25	DA5	LED RED
26	-12V	Nevyužito
27	+12V	Nevyužito
28	+5V	5V
29	GND	GND
30	DOUT0	QEIB
31	DOUT1	QEIA
32	DOUT2	DI4
33	DOUT3	DI3
34	DOUT4	DI2
35	DOUT5	DI1
36	DOUT6	LS2
37	DOUT7	LS1

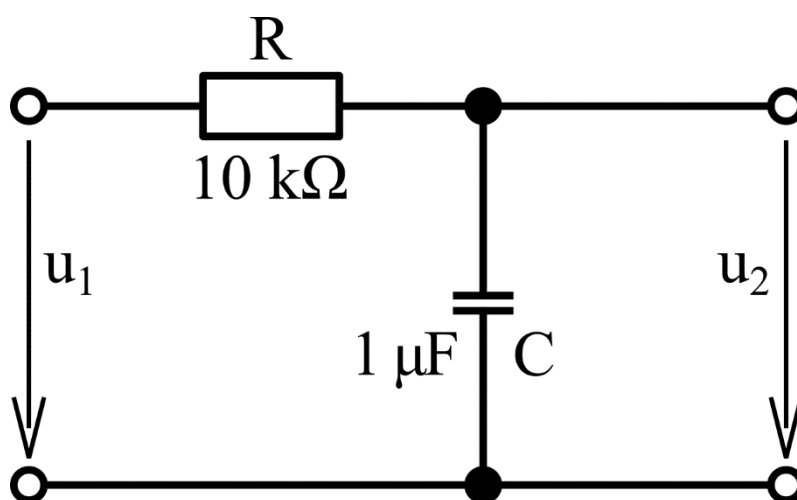
Při návrhu DPS testovacího přípravku bylo uvažováno, že pin PWM testované jednotky bude propojen na digitální vstup DIN7 karty MF624. Po výrobě této DPS však bylo zjištěno, že PWM má nosnou frekvenci $f_{PWM} = 20 \text{ kHz}$, což je příliš vysoká frekvence signálu na to, aby mohl být signál zaznamenán pomocí karty. Bylo tedy rozhodnuto, že signál PWM generovaný testovanou jednotkou bude filtrován pomocí RC článku – dolní propusti.

Parametry dolní propusti byly zvoleny následující:

Odpor rezistoru $R = 10 \text{ k}\Omega$

Kapacita kondenzátoru $C = 1 \mu\text{F}$

Schéma RC členu – dolní propusti – s navrženými parametry je znázorněno na obr. 3.7.



Obrázek 3.7: Schéma dolní propusti s navrženými parametry součástek

Časovou konstantu dolní propusti lze snadno určit:

$$\tau = R \cdot C = 10 \cdot 10^3 \cdot 1 \cdot 10^{-6} = 1 \cdot 10^{-2} \text{ s} = 10 \text{ ms}$$

Pro dolní propust platí následující rovnice:

$$u_2 = u_1 \cdot \frac{\frac{1}{j\omega C}}{\frac{1}{j\omega C} + R} = u_1 \cdot \frac{\frac{1}{j\omega C}}{\frac{1 + j\omega RC}{j\omega C}} = u_1 \cdot \frac{1}{j\omega RC + 1} = u_1 \cdot \frac{1}{j\omega\tau + 1}$$

Přenos dolní propusti je tedy následující:

$$F(p) = \frac{u_2}{u_1} = \frac{1}{\tau \cdot p + 1}$$

Po dosazení parametrů použité dolní propusti:

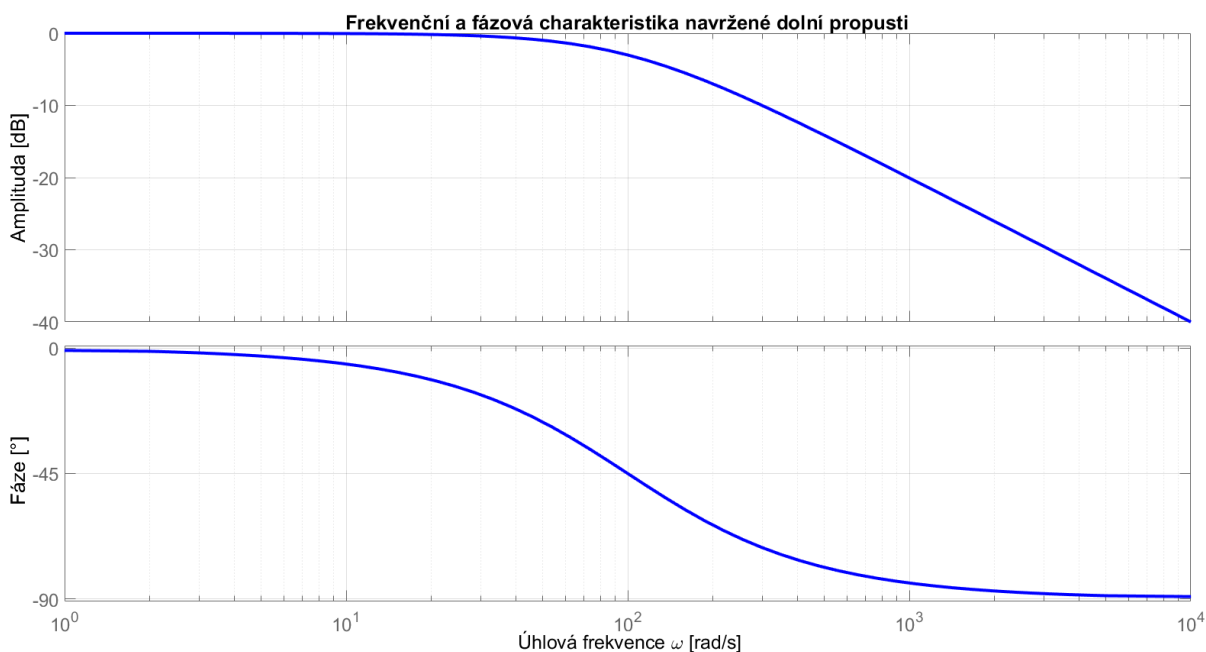
$$F(p) = \frac{1}{\tau \cdot p + 1} = \frac{1}{0,01 \cdot p + 1}$$

Přenosová funkce byla zadána do prostředí MATLAB a následně z ní byla vykreslena frekvenční a fázová charakteristika (Bodeho diagram), které jsou znázorněny na obrázku 3.8. Pro mezní kmitočet ω_0 musí platit:

$$\omega_0 = \frac{1}{R \cdot C} = \frac{1}{\tau} = \frac{1}{0,01} = 100 \frac{\text{rad}}{\text{s}}$$

Tento výpočet se potvrdil i grafem, ze kterého je evidentní, že mezní kmitočet ω_0 , na kterém dochází k útlumu -3 dB, je skutečně $100 \frac{\text{rad}}{\text{s}}$. Mezní frekvenci f_0 určíme vztahem:

$$f_0 = \frac{\omega_0}{2\pi} = \frac{100}{2\pi} = 15,92 \text{ Hz}$$



Obrázek 3.8: Frekvenční a fázová charakteristika navržené dolní propusti

Na vstup dolní propusti je připojen pin PWM testované DPS, na kterém testovaná jednotka generuje pulzně šířkově modulovaný signál pro řízení elektrického motoru. Výstup dolní propusti je přiveden na analogový vstup AD7 karty MF624 viz tabulka 1.

3.2.1 SW pro automatizovaný test

Výše popisovaný testovací skript od Romana jsem začal rozšiřovat o další funkcionality. Aby bylo možné jednotku Slave otestovat, je nutné do ní nejprve nahrát testovací firmware, ve kterém je například naprogramováno, aby se signály z digitálních a analogových vstupů posílaly rovnou na digitální, respektive analogové výstupy.

Pro automatizaci celého testu bylo nutné vyřešit, jak nahrávat firmware testované jednotky, který byl vytvořen ve vývojovém prostředí MPLAB X IDE, pomocí příkazů z MATLABu.

Projekt v MPLABu lze ovládat pomocí příkazové řádky operačního systému Windows a příkazy v MATLABu lze volat příkazovou řádku Windows.

Provést sestavení (build) projektu lze příkazem `make`, k nahrání firmware slouží příkaz `ipcmd`. Aby bylo možné používat tyto příkazy v MATLABu, vytvořil jsem následující návod:

Sestavení (Build) projektu z MATLABu

Pro kompilaci projektu přes příkazovou řádku slouží příkaz `make`. Aby však tento příkaz fungoval, je potřeba do systémové proměnné `Path` (Tento počítač -> Vlastnosti -> Upřesnit nastavení systému -> Proměnné prostředí -> Systémové prostředí -> Path -> Upravit -> Nový) přidat následující cestu:

```
C:\Program Files (x86)\Microchip\MPLABX\v5.10\gnuBins\GnuWin32\bin
```

V proměnné `Path` musí zůstat všechny hodnoty, které tam již jsou, pouze se na konec přidá výše uvedená cesta.

Poté je nutné změnit aktuální pracovní složku MATLABu na cestu k projektu v MPLABu, který má být zkompileován. Do MATLABu je tedy zadáno například:

```
% zmena pracovni_slozky  
cd('C:\Users\Student\MPLABXProjects\vzorovy_projekt.X')
```

Příkazem `make clean` se následně vymažou soubory vzniklé při poslední kompilaci. Příkazovou řádku lze z MATLABu ovládat pomocí funkce `system`:

```
% vymazani starych souboru  
[status,cmdout] = system('make clean')
```

Příkazem `make` je následně provedeno sestavení (build) projektu:

```
% kompilace  
[status,cmdout] = system('make')
```

Nahrání firmware – nahrání hex souboru vzniklého sestavením (buildem)

Pro nahrání firmware se používá příkaz `ipecmd`. Aby bylo možné příkaz použít, do systémové proměnné `Path` musí být přidána ještě následující cesta:

`C:\Program Files (x86)\Microchip\MPLABX\v5.10\mplab_platform\mplab_ipe`

Následně již je možné použít příkaz `ipecmd`:

```
% nahrani hex souboru
[status,cmdout] = system('ipecmd -TPPK3 -P33EP256MU806 -M -
FC:\Users\Student\MPLABXProjects\vzorovy_projekt.X\dist\default\production\
vzorovy_projekt.X.production.hex -OL ')

```

Vysvětlení použitých parametrů příkazu `ipecmd`:

`-TPPK3` = použitý programátor PICKit3, alternativně lze použít např. `-TPPK4` pro PICKit4

Lze však nahradit přímo sériovým číslem použitého programátoru,

např. `-TSBUR1234566` pro programátor se sériovým číslem BUR1234566

`-P33EP256MU806` = cílový kontrolér dsPIC33EP256MU806 (uvádí se bez dsPIC na začátku)

`-M` znamená, že je programována celá paměť

`-FC:\Users\Student\MPLABXProjects\vzorovy_projekt.X\dist\default\production\vzorovy_projekt.X.production.hex` udává za písmenem F cestu k hex souboru, který chceme nahrávat

`-OL` zamezuje tomu, aby čip po přeprogramování zůstal v resetu (pokud tento parametr neuvedeme, po nahrání se kód nezačne vykonávat, ale je nutné provést restart manuálně)

Reference

`C:\Program Files (x86)\Microchip\MPLABX\v5.10\docs\Readme for IPECMD.htm` – manuál k IPECMD, jsou zde podrobně vysvětleny jednotlivé parametry tohoto příkazu

Řešení problémů

Hláška 'Ipecmd' není názvem vnitřního ani vnějšího příkazu...

Pokud po přidání cesty k souborům do systémové proměnné `Path` MATLAB po volání příkazu `ipecmd` zobrazuje varovnou hlášku:

'ipecmd' is not recognized as an internal or external command, operable program or batch file.

Respektive českou alternativu: *'ipecmd' není názvem vnitřního ani vnějšího příkazu, spustitelného programu nebo dávkového souboru*, znamená to, že Matlab stále nemá přístup k souboru `ipecmd`.

Pokud je však příkaz zadán přímo do příkazové řádky, obvykle příkaz `ipecmd` funguje.

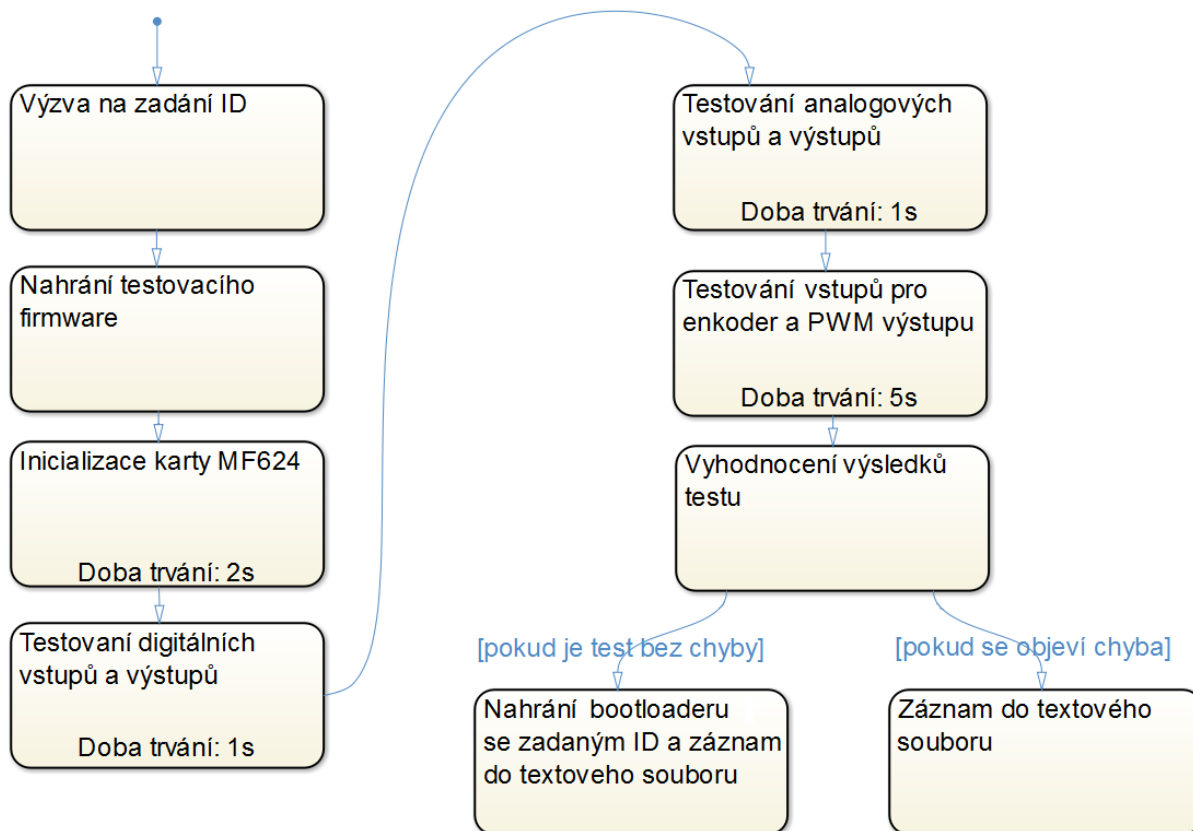
Řešením je **restartování systému** a poté by již měl příkaz `ipecmd` fungovat i při volání z Matlabu.

Protože Roman používal při testování jiné zapojení mezi kartou a testovanou jednotkou, musel být nejprve jeho kód upraven podle schéma DPS testovacího přípravku, která propojuje kartu s testovanou jednotkou.

V testovacím programu byly dále provedeny změny tak, že podstatné informace z průběhu testu jsou zaznamenávány do textového souboru, aby bylo možné na průběh testu kdykoli zpětně nahlédnout. Na začátku testování se do tohoto souboru запиše datum, čas a informace o tom, že bylo zahájeno nové testování. Uživatel je následně vyzván k zadání ID testované jednotky. V momentě, kdy tak učiní, se tato informace rovněž zaznamená do textového souboru. Poté následují příkazy pro nahrání testovacího firmware do testované jednotky, aby mohlo být následně zahájeno samotné testování. Informace o průběhu testu se vypisují do příkazového okna programu MATLAB. Pokud jednotlivé dílčí části testu proběhnou úspěšně, je tato informace vypsána pouze v příkazovém okně, ale do výpisu v textovém souboru se úspěšné dílčí kroky nezaznamenávají, aby nebyl výpis příliš dlouhý. V případě, že se objeví nějaká chyba, je tato informace samozřejmě zobrazena v prostředí MATLAB i je zapsána do textového souboru.

Pokud celý test proběhne bez chyby, posledním bodem testovacího skriptu je nahrání bootladeru do testované jednotky. Nejprve je však nutné zapsat ID zadané uživatelem do konfiguračního souboru projektu v MPLABu. Tento soubor se tedy z MATLABu otevře a nahraje, následně se vyhledá příslušný řádek s informací o ID, provede se zápis ID aktuálně testované jednotky a celý soubor se zase uloží. Poté je již možné provést kompilaci celého projektu s bootladerem prostřednictvím příkazů z MATLABu a nahrát vzniklý hex soubor do úspěšně otestované jednotky.

Celý tento test je automatizovaný, jediná interakce s uživatelem spočívá v zadání ID na začátku testu a pokud test dopadne úspěšně, testovací skript se postará i o nahrání bootladeru se správným ID, jak je naznačeno na obr. 3.9.



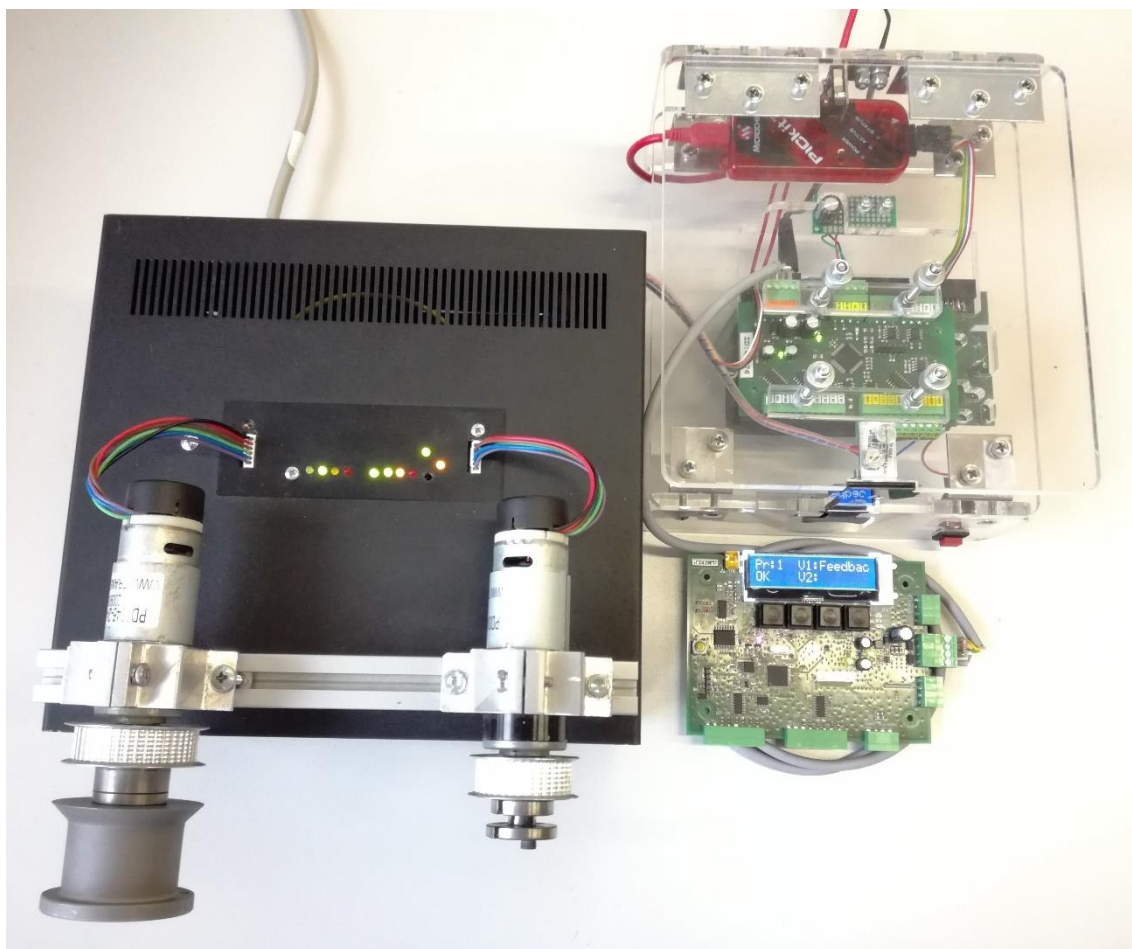
Obrázek 3.9: Vývojový diagram funkčního testu

3.3 HIL test regulátoru DC motoru

Dalším z bodů zadání této diplomové práce je vytvoření HIL testu, který bude ověřovat schopnost testované jednotky řídit DC motor. Doposud byla tato schopnost ověřována připojením reálného DC motoru k testované jednotce. Motor je součástí zařízení Double Drive. Jedná se o platformu určenou pro výuku v Mechlabu, která obsahuje 2 DC motory s planetovou převodovkou a enkodéry, napájecí zdroj, výkonovou elektroniku (H můstek), indikační LED diody a konektor pro připojení ke kartě MF624, prostřednictvím které lze celý Double Drive řídit z MATLABu a Simulinku.

Protože každý motor má mírně odlišné parametry, byl pro účely srovnávání reálného motoru s matematickým modelem vybrán Double Drive s označením DD#01 a konkrétně jeho motor A, tedy ten na levé straně při pohledu zepředu. Tento Double Drive je znázorněn v levé části fotografie na obr. 3.10.

Standardně se model řízené soustavy pro HIL simulaci vytváří v prostředí Simulink, v našem případě však bylo výhodnější vytvořit model v MATLABu stejně jako funkční test, který je rovněž tvořen skripem v MATLABu.



Obrázek 3.10: Zařízení Double Drive, testovací přípravek, jednotky Slave a Master

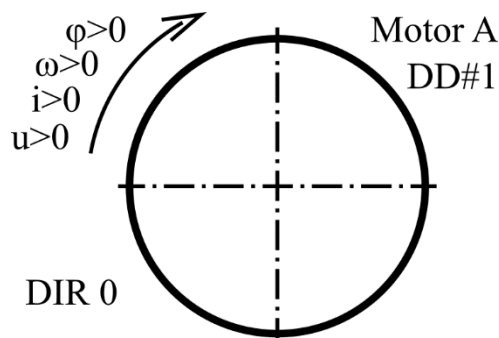
Motor A, který bude modelován, má označení Transmotec PD3046-24-5-BFEC. Jedná se o DC motor s planetovou převodovkou, kuličkovými ložisky, EMC filtrem a enkodérem. Hodnoty z katalogu výrobce jsou uvedeny v tabulce 2.

Tabulka 2: Katalogové parametry motoru s planetovou převodovkou [20]

Veličina	Hodnota
Jmenovité napětí	24 V
Převodový poměr	5
Jmenovitý moment	53 mNm
Jmenovité otáčky	1150 ot/min
Délka	70 mm
Váha	188 g
Počet tiků enkodéru	14 tiků/ot/kanál
Počet kanálů enkodéru	2

Aby bylo možné dokonale simulovat reálný motor, muselo u popisovaného motoru být nejprve provedeno měření, kolik tiků na otáčku výstupní hřídele má tento motor. Double Drive byl tedy připojen ke vstupně výstupní kartě MF624 a prostřednictvím Simulinku bylo změřeno, že enkodér dává $145 \frac{\text{tiků}}{\text{ot}}$. Následovalo určení kladných směrů otáčení, proudu a napětí, jak je

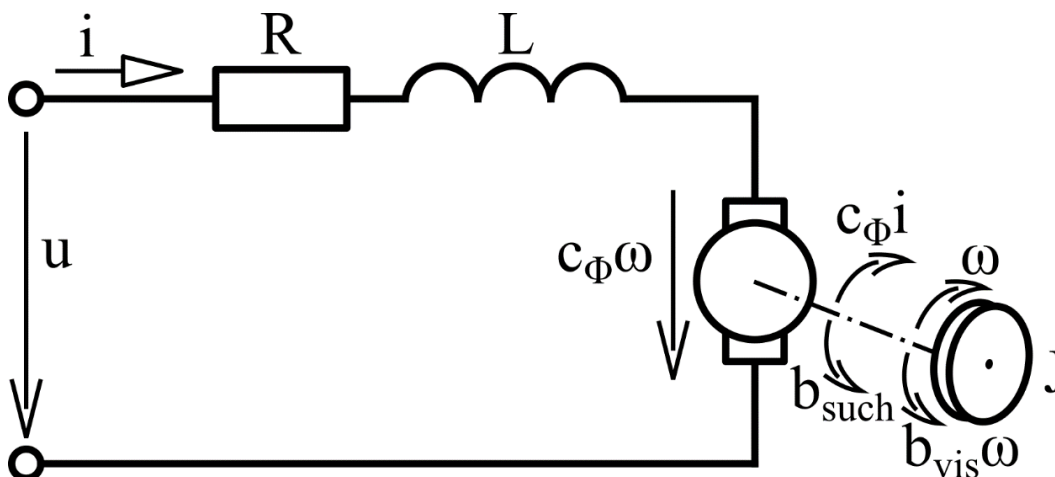
naznačeno na obr. 3.11. Pokud se motor otáčí naznačeným směrem, na výstupu enkodéru se signál kanálu B zpožďuje za signálem kanálu A.



Obrázek 3.11: Vyznačení kladných směrů motoru A na Double Drive DD#01

3.3.1 Model DC motoru

Náhradní schéma DC motoru je znázorněno na obr. 3.12.



Obrázek 3.12: Náhradní schéma DC motoru

Na základě náhradního schéma byly sestaveny následující rovnice popisující DC motor:

$$u = R \cdot i + L \cdot \frac{di}{dt} + c_{\phi} \cdot \omega$$

$$J \cdot \frac{d\omega}{dt} = c_{\phi} \cdot i - b_{vis} \cdot \omega - b_{such} \cdot \text{sgn } \omega$$

Z první rovnice je nutné vyjádřit derivaci proudu $\frac{di}{dt}$ a ze druhé rovnice derivaci úhlové rychlosti $\frac{d\omega}{dt}$:

$$\frac{di}{dt} = \frac{1}{L} \cdot (u - R \cdot i - c_{\phi} \cdot \omega)$$

$$\frac{d\omega}{dt} = \frac{1}{J} \cdot (c_{\phi} \cdot i - b_{vis} \cdot \omega - b_{such} \cdot \text{sgn } \omega)$$

Změna proudu di a úhlové rychlosti $d\omega$ za časový interval dt je tedy následující:

$$di = \frac{1}{L} \cdot (u - R \cdot i - c_\phi \cdot \omega) \cdot dt$$

$$d\omega = \frac{1}{J} \cdot (c_\phi \cdot i - b_{vis} \cdot \omega - b_{such} \cdot sgn \omega) \cdot dt$$

Protože MATLAB řeší diferenciální rovnice v diskrétních časových krocích, bude pro proud i a úhlovou rychlost ω v časovém kroku k platit:

$$i_k = i_{k-1} + di$$

$$\omega_k = \omega_{k-1} + d\omega$$

Pokud dosadíme výše uvedené rovnice pro di a $d\omega$ do rovnic pro i_k a ω_k , dostáváme rovnice potřebné pro vytvoření modelu motoru v MATLABu:

$$i_k = i_{k-1} + \frac{1}{L} \cdot (u - R \cdot i - c_\phi \cdot \omega) \cdot dt$$

$$\omega_k = \omega_{k-1} + \frac{1}{J} \cdot (c_\phi \cdot i - b_{vis} \cdot \omega - b_{such} \cdot sgn \omega) \cdot dt$$

Aby bylo možné počítat natočení hřídele motoru φ , je nutné rovnice doplnit o třetí rovnici:

$$\varphi_k = \varphi_{k-1} + \omega_{k-1} \cdot dt$$

Výše uvedené kinematické veličiny (úhlová rychlost ω a natočení hřídele φ) jsou vztaženy na výstup DC motoru, což je současně vstup do planetové převodovky. Dále budou tedy označovány indexem 1.

Kinematické veličiny vztažené k výstupu převodovky (úhlová rychlost ω a natočení φ výstupní hřídele) budou nadále označovány indexem 2. Úhlová rychlost vstupní hřídele převodovky ω_1 je přímo úměrná výstupní úhlové rychlosti ω_2 přes převodový poměr i :

$$\omega_1 = \omega_2 \cdot i$$

Analogicky platí, že natočení vstupní hřídele φ_1 je přímo úměrné natočení výstupní hřídele φ_2 přes převodový poměr i :

$$\varphi_1 = \varphi_2 \cdot i$$

Protože má model simulovat DC motor včetně enkodéru a generovat signály z kanálů enkodéru A a B, je nutné zahrnout enkodér rovněž do modelu v MATLABu.

Kanály enkodéru jsou dva a každý může nabývat dvou hodnot, celkem tedy mohou nastat 4 stavy během jednoho pulzu enkodéru. Pro jeho správnou činnost je nezbytné určit, ve kterém kvadrantu pulzu se právě nachází a podle toho nastavit odpovídající hodnoty na simulované kanály enkodérů A a B.

Měřením na reálném motoru bylo zjištěno, že enkodér má $CPR_2 = 145 \frac{tiků}{ot}$ výstupní hřídele. Enkodér však snímá polohu vstupní hřídele převodovky, na základě znalosti převodového poměru tedy byl vypočítán počet tiků na otáčku vstupní hřídele CPR_1 :

$$CPR_1 = \frac{CPR_2}{i} = \frac{145}{5} = 29 \frac{tiků}{ot}$$

Podle katalogu má motor $14 \frac{\text{tiků}}{\text{ot} \cdot \text{kan}}$, enkodér má dva kanály, takže dle katalogu má mít motor $28 \frac{\text{tiků}}{\text{ot}}$, což odpovídá naměřené hodnotě $CPR_1 = 29 \frac{\text{tiků}}{\text{ot}}$.

Na základě této znalosti byla určeno natočení vstupní hřídele odpovídající jednomu tiku enkodéru:

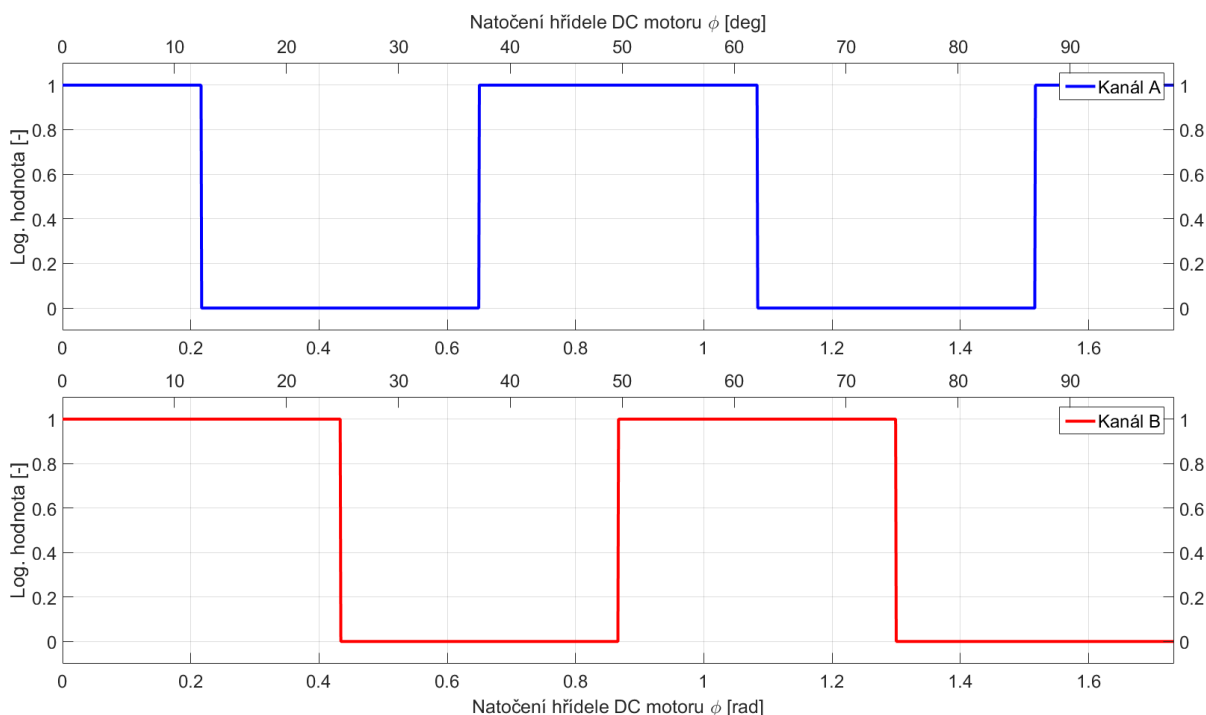
$$\varphi_1 = \frac{2 \cdot \pi}{CPR_1} = 0,217 \text{ rad}$$

Čtyři tiky enkodéru odpovídají jednomu pulzu, pokud tedy dojde k natočení o $4 \cdot 0,217 = 0,867 \text{ rad}$, na výstupu enkodéru se objeví právě jeden pulz.

Během simulace DC motoru je na konci každého kroku simulace natočení hřídele motoru φ_1 poděleno úhlem $0,867 \text{ rad}$ a na základě zbytku po dělení (používá se přímo funkce modulo) se rozhodne, ve kterém kvadrantu pulzu se hřídel právě nachází. Podle toho se nastaví simulované výstupy enkodéru na odpovídající hodnoty, přičemž pokud natočení φ_1 roste, sekvence je taková, že kanál B se zpožďuje za kanálem A. Stav výstupů v jednotlivých kvadrantech pulzu jsou znázorněny v tabulce 3 a grafu na obr. 3.13. [21]

Tabulka 3: Výstupy kanálů enkodérů v závislosti na natočení hřídele

Poloha hřídele	Natočení hřídele φ_1		Výstup kanálu A	Výstup kanálu B
	od	do		
I. kvadrant pulzu	0 rad	0,217 rad	1	1
II. kvadrant pulzu	0,217 rad	0,433 rad	0	1
III. kvadrant pulzu	0,433 rad	0,650 rad	0	0
IV. kvadrant pulzu	0,650 rad	0,867 rad	1	0



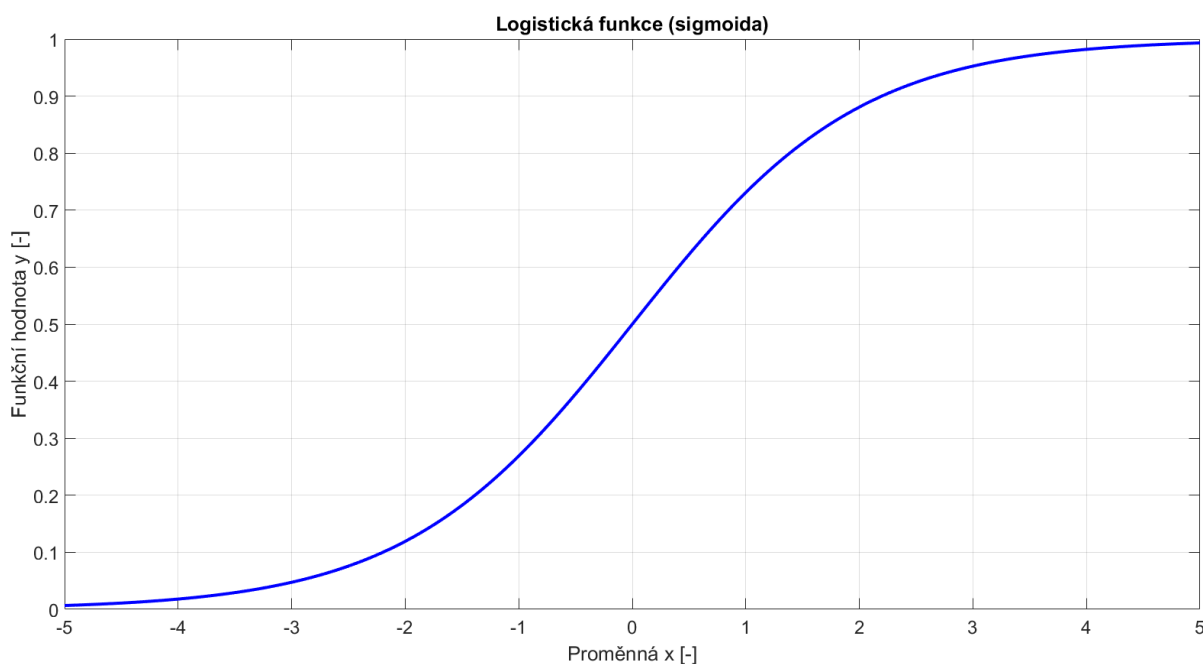
Obrázek 3.13: Závislost výstupu enkodéru na natočení vstupní hřídele převodovky

3.3.2 Aproximace funkce $\operatorname{sgn} \omega$

Na základě výše uvedených rovnic byl vytvořen model DC motoru v prostředí MATLAB, nicméně použití funkce $\operatorname{sgn} \omega$ značně zvyšuje výpočetní náročnost modelu, protože je funkce nespojitá v počátku soustavy souřadnic. Funkci $\operatorname{sgn} \omega$ je tedy vhodné nahradit spojitou funkcí. Pro tyto účely byla vybrána logistická funkce (sigmoida). Ta je obecně definována rovnicí:

$$y = \frac{1}{1 + e^{-x}}$$

Graf této funkce je znázorněn na obr. 3.14.



Obrázek 3.14: Graf logistické funkce

Z grafu je evidentní, že logistická funkce má obor hodnot $(0; 1)$, funkce $\operatorname{sgn} \omega$ však $(-1; 1)$, logistickou funkcí je tedy nutné vynásobit číslem 2 a následně odečíst -1 . Protože navíc logistická funkce roste příliš pozvolně, byl do exponentu členu e^{-x} přidán koeficient 20.

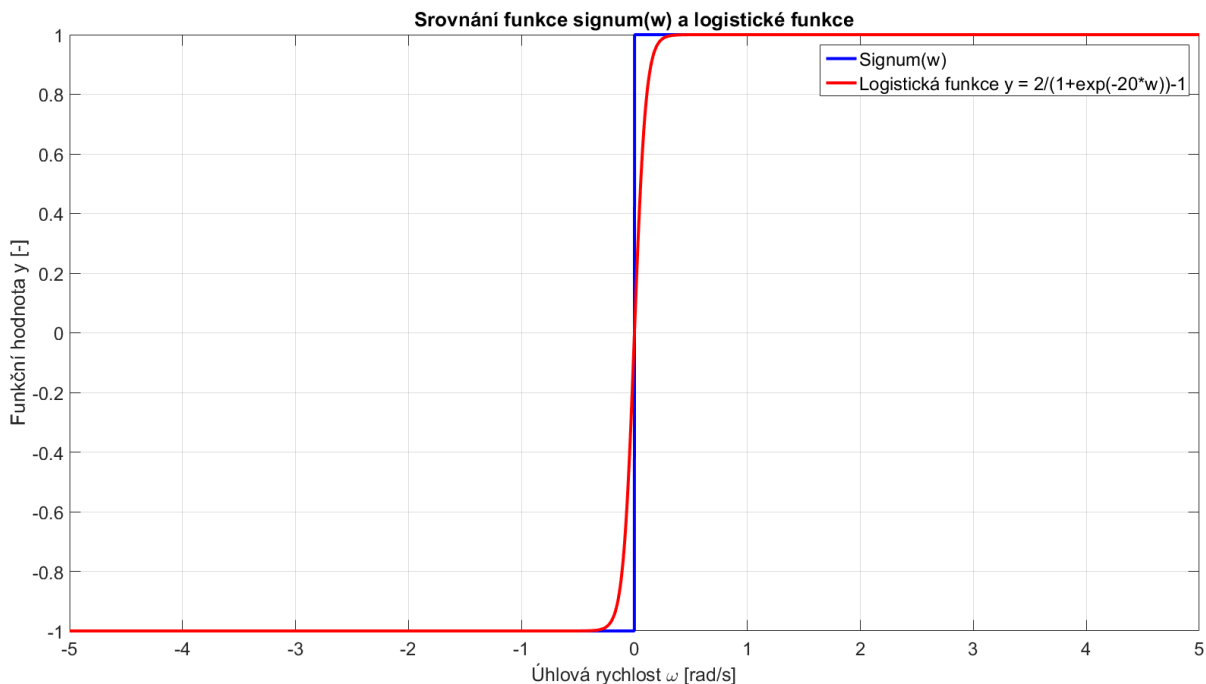
Předpis funkce pro náhradu funkce $\operatorname{sgn} x$ je tedy následující:

$$y = \frac{2}{1 + e^{-20x}} - 1$$

V případě modelu DC motoru je nutné touto funkcí nahrazovat $\operatorname{sgn} \omega$, místo proměnné x se tedy bude psát úhlová rychlost ω :

$$y = \operatorname{sgn} \omega \approx \frac{2}{1 + e^{-20\omega}} - 1$$

Srovnání grafu této funkce společně s grafem funkce $\operatorname{sgn} \omega$ zobrazuje graf na obrázku 3.15, odkud je evidentní, že daná logistická funkce nahrazuje funkci signum s dostatečnou přesností, výpočtová náročnost modelu je však v tomto případě mnohem nižší.



Obrázek 3.15: Srovnání funkce signum a logistické funkce, která ji nahrazuje

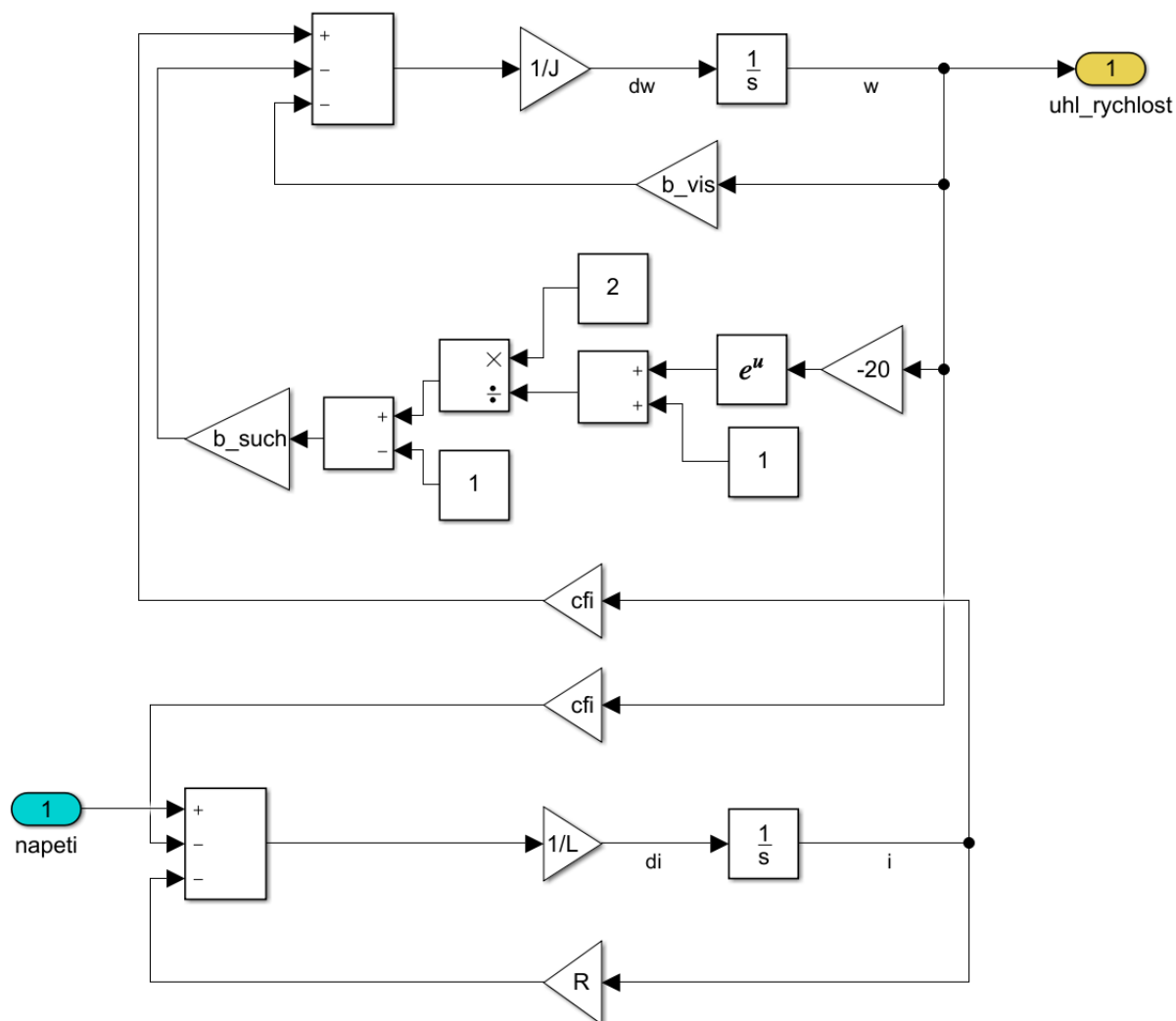
3.3.3 Estimace parametrů modelu

Aby byl vytvořený model DC motoru funkční a odpovídal realitě, je nutné správně zadat jeho parametry. Nejdříve byly nastaveny přibližné parametry motoru pro ověření, zda jsou rovnice správné a tvar výstupních signálů z modelu (úhlová rychlost ω , natočení φ a proud tekoucí motorem i). Poté musel být model upraven tak, aby byly zohledněny zvolené kladné směry veličin. Model však pouze přibližně odpovídal reálnému průběhu. Na řadu tedy musela přijít estimace parametrů, aby model popisoval reálný motor přesněji.

Model pro estimaci znázorněný na obr. 3.16 byl vytvořen v Simulinku na základě následujících rovnic:

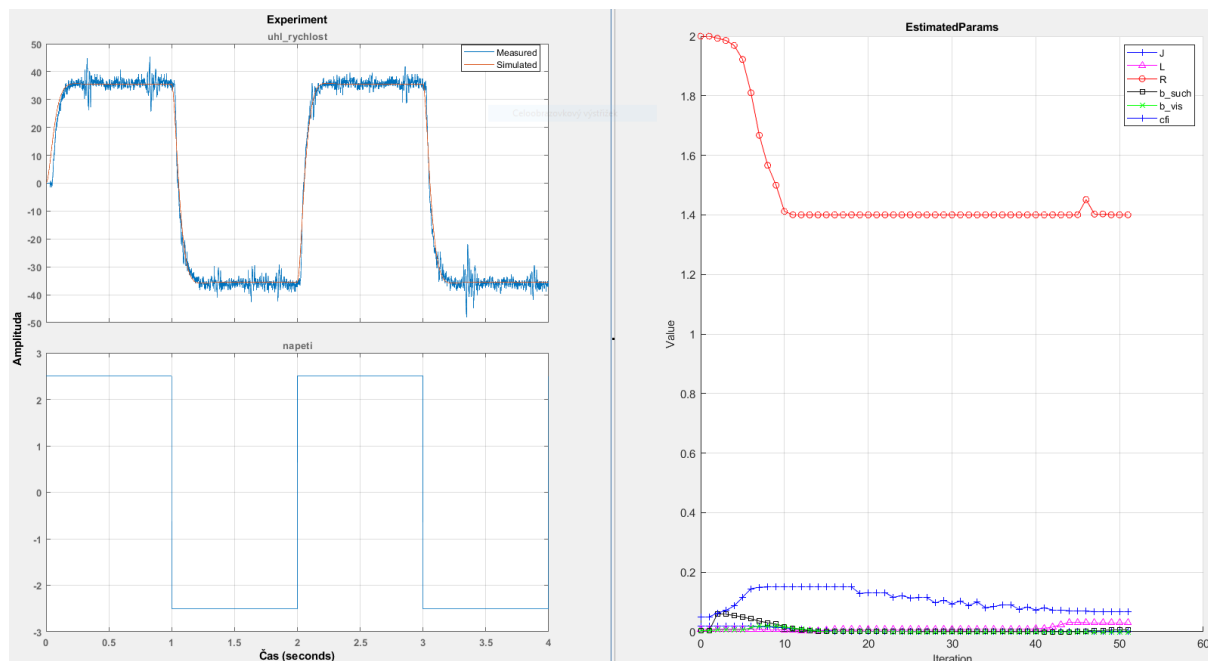
$$\frac{di}{dt} = \frac{1}{L} \cdot (u - R \cdot i - c_{\phi} \cdot \omega)$$

$$\frac{d\omega}{dt} = \frac{1}{J} \cdot \left(c_{\phi} \cdot i - b_{vis} \cdot \omega - b_{such} \cdot \left(\frac{2}{1 + e^{-20\omega}} - 1 \right) \right)$$



Obrázek 3.16: Model DC motoru pro estimaci parametrů

Pro účely estimace je vstupem modelu napětí U a výstupem úhlová rychlost ω . Estimaci lze provést pouze na základě naměřených dat na reálné soustavě, bylo tedy provedeno měření, při kterém byl motor buzen po částech konstantním signálem s periodou 2 s, kdy po dobu 1 s bylo buzeno kladným napětím 2,5V, následně 1 s záporným napětím $-2,5V$, jak je naznačeno na obr. 3.17. Měřeno bylo natočení hřídele motoru φ_1 a z naměřených hodnot byl filtrovanou derivací získán průběh úhlové rychlosti ω_1 potřebný pro estimaci.



Obrázek 3.17: Estimace parametrů – naměřená data pro estimaci a průběh estimace

Parametry získané estimací jsou zaznamenány v tabulce 4.

Tabulka 4: Estimované parametry a jejich hodnoty

Estimovaný parametr	Hodnota získaná estimací
Moment setrvačnosti J	0,000205 kg · m ²
Indukčnost L	0,03012 H
Odpor vinutí R	1,4002 Ω
Koeficient suchého tření b_{such}	0,0062 Nm
Koeficient viskozního tření b_{vis}	1,377 · 10 ⁻¹³ Nms ²
Konstanta motoru c_{ϕ}	0,0667 Vs

Odpor vinutí rotujícího motoru R byl určen estimací na 1,4002 Ω. Pro srovnání bylo provedeno měření na stojícím motoru. RLC metrem CEM DT-9935 byl naměřen odpor motoru 10,91 Ω. Je zřejmé, že odpor stojícího a rotujícího motoru nebude stejný, nicméně rozdíl mezi hodnotou odporu získanou estimací při rotujícím motoru a hodnotou získanou měřením při stojícím motoru je příliš velký.

Stejným RLC metrem byla rovněž změřena indukčnost stojícího motoru. Výsledek měření je 4,76 mH, zatímco estimací byla získána při běhu motoru hodnota indukčnosti 301 mH, tedy více než šestinásobná.

3.3.4 Srovnání výsledků HIL testu a reálného experimentu

Na základě znalosti výše uvedených rovnic a estimovaných parametrů reálného motoru již bylo možné realizovat HIL test. DC motor je simulován skriptem v MATLABu, který provádí následující činnosti v uvedeném pořadí:

Na samotném začátku je iniciována karta MF624 a nastaven časový krok na 1 ms. Simulace tedy běží na frekvenci 1 kHz, což je dostatečně rychlé na to, aby testovaná jednotka nepoznala, že neovládá reálný motor.

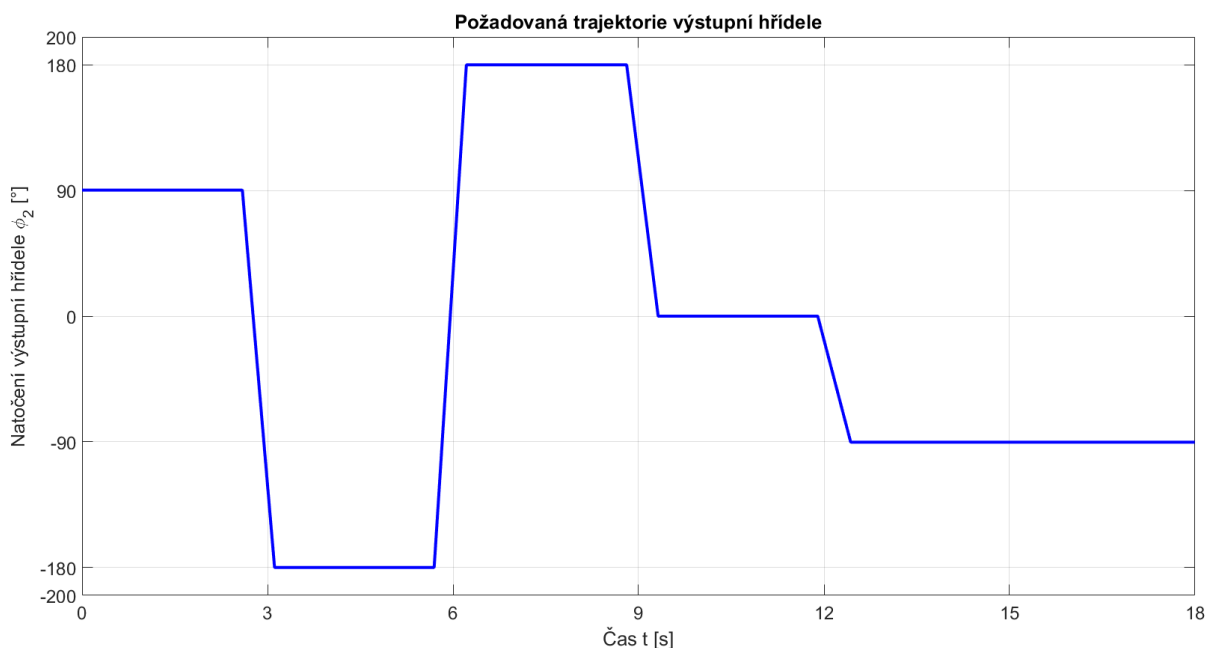
V úvodní pasáži skriptu následuje definice parametrů simulovaného motoru jako je například odpor motoru R , indukčnost L , moment setrvačnosti J , konstanta motoru c_ϕ , ale i počet tiků enkodéru na otáčku hřídele DC motoru.

Při samotné simulaci DC motoru je vždy na začátku kroku simulace přečtena hodnota analogového napětí na pinu AOM, což je akční zásah regulátoru běžícího na testované jednotce. Protože se signál pohybuje v rozsahu od 0 V do 10 V, nenese signál na tomto pinu informaci o směru, kterým se má motor točit. Proto je nutné současně vyčítat digitální hodnotu na pinu DIR, který udává směr otáčení motoru. Pokud je na tomto pinu logická nula, motor se otáčí po směru chodu hodinových ručiček, jak je naznačeno na obr. 3.11.

Na základě akčního zásahu se dle rovnic DC motoru spočítá nejprve úhlová rychlost ω_1 , následně proud i a natočení hřídele DC motoru (respektive vstupní hřídele převodovky) φ_1 . Všechny veličiny jsou vypočítány pro aktuální časový krok.

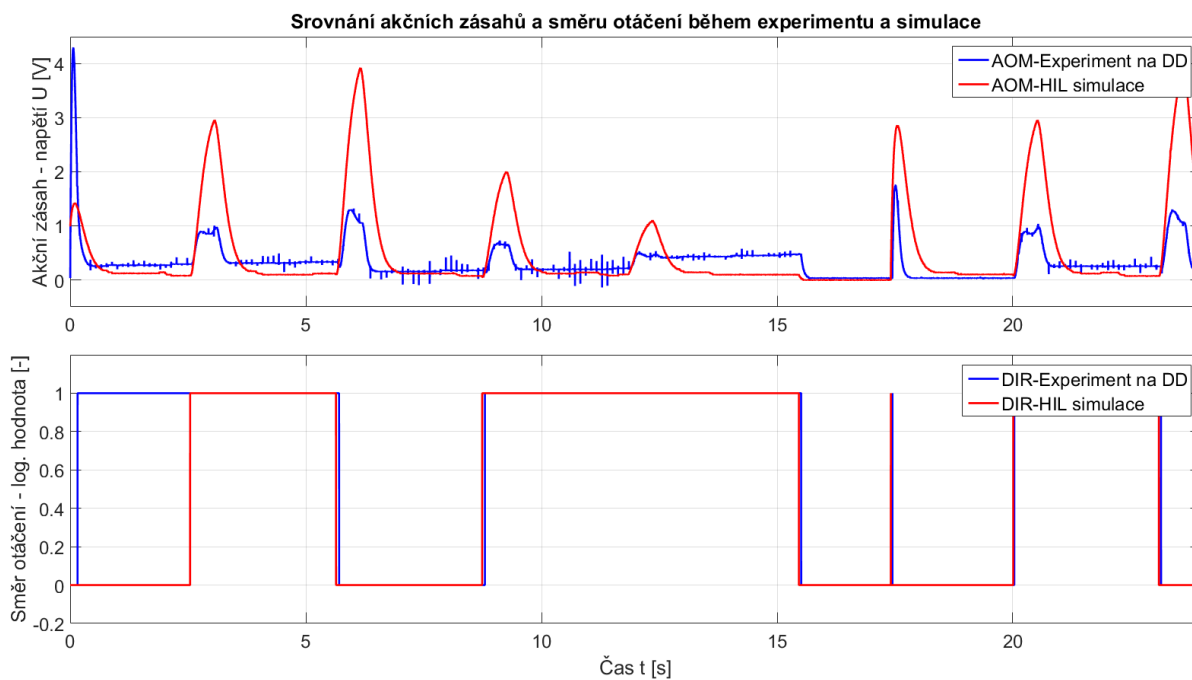
Aby bylo možné simulovat enkodér, následuje výpočet, v jakém kvadrantu pulzu enkodéru se hřídel právě nachází. Tento výpočet je realizovaný na základě natočení hřídele DC motoru a funkce modulo, popis výpočtu je uveden v kapitole 3.3.1. Po výpočtu dojde k nastavení digitálních výstupů reprezentujících kanály enkodéru A a B do správné logické úrovně. Následuje už jen příkaz, který zajišťuje správné časování. Pokud se tedy celý krok simulace stihne vypočítat dostatečně rychle, čeká MATLAB po zbývající dobu kroku simulace, pokud výpočet trval příliš dlouho a došlo k promeškání časového kroku, následuje okamžitě přechod na další krok simulace.

K ověření správnosti HIL simulace slouží porovnání chování skutečného motoru a jeho modelu běžícího v rámci HIL simulace. Do testované jednotky byl nahrán PI regulátor na polohu s požadovanou trajektorií výstupní hřídele znázorněnou na obrázku 3.18.

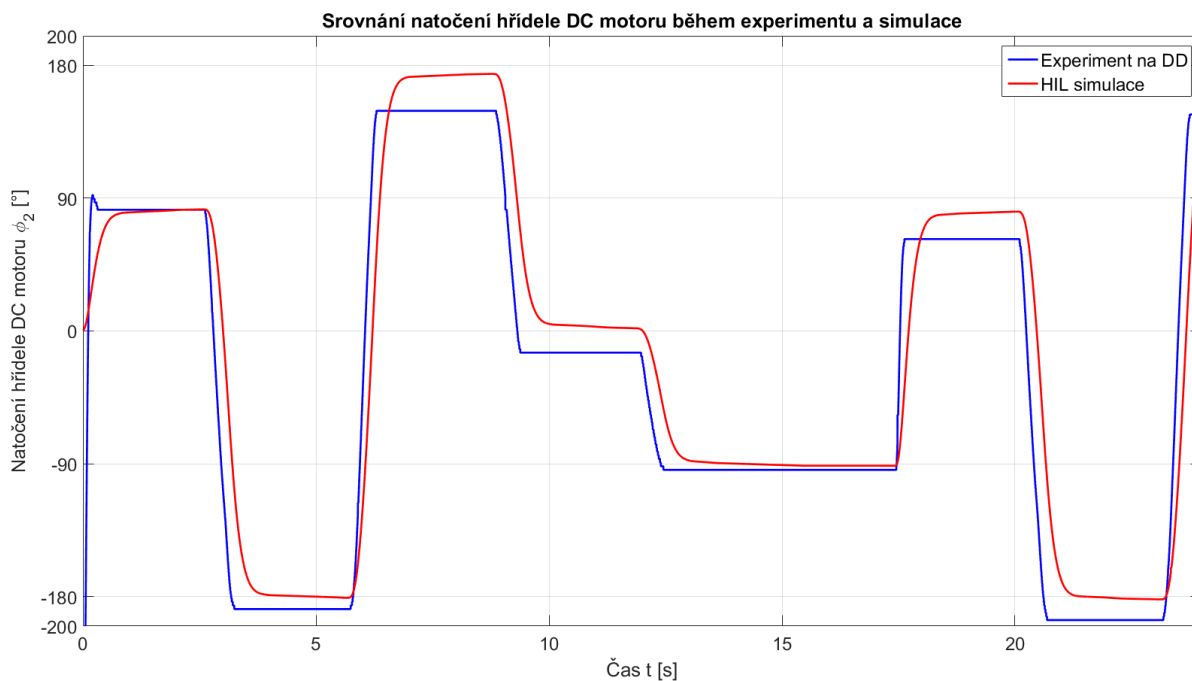


Obrázek 3.18: Požadovaná trajektorie výstupní hřídele

Nejprve byl připojen k testované jednotce Double Drive s reálným motorem a po spuštění testované jednotky byl zaznamenán průběh akčního zásahu (signál AOM), směru otáčení (signál DIR) a skutečné polohy určené z kanálů enkodérů A a B. Následně byl Double Drive odpojen a připojena karta MF624, aby bylo možné DC motor simulovat. Po spuštění HIL simulace byly zaznamenány stejné signály. Srovnání akčních zásahů a směru otáčení je uvedeno na obr. 3.19, natočení rotoru je porovnáno na obrázku 3.20.



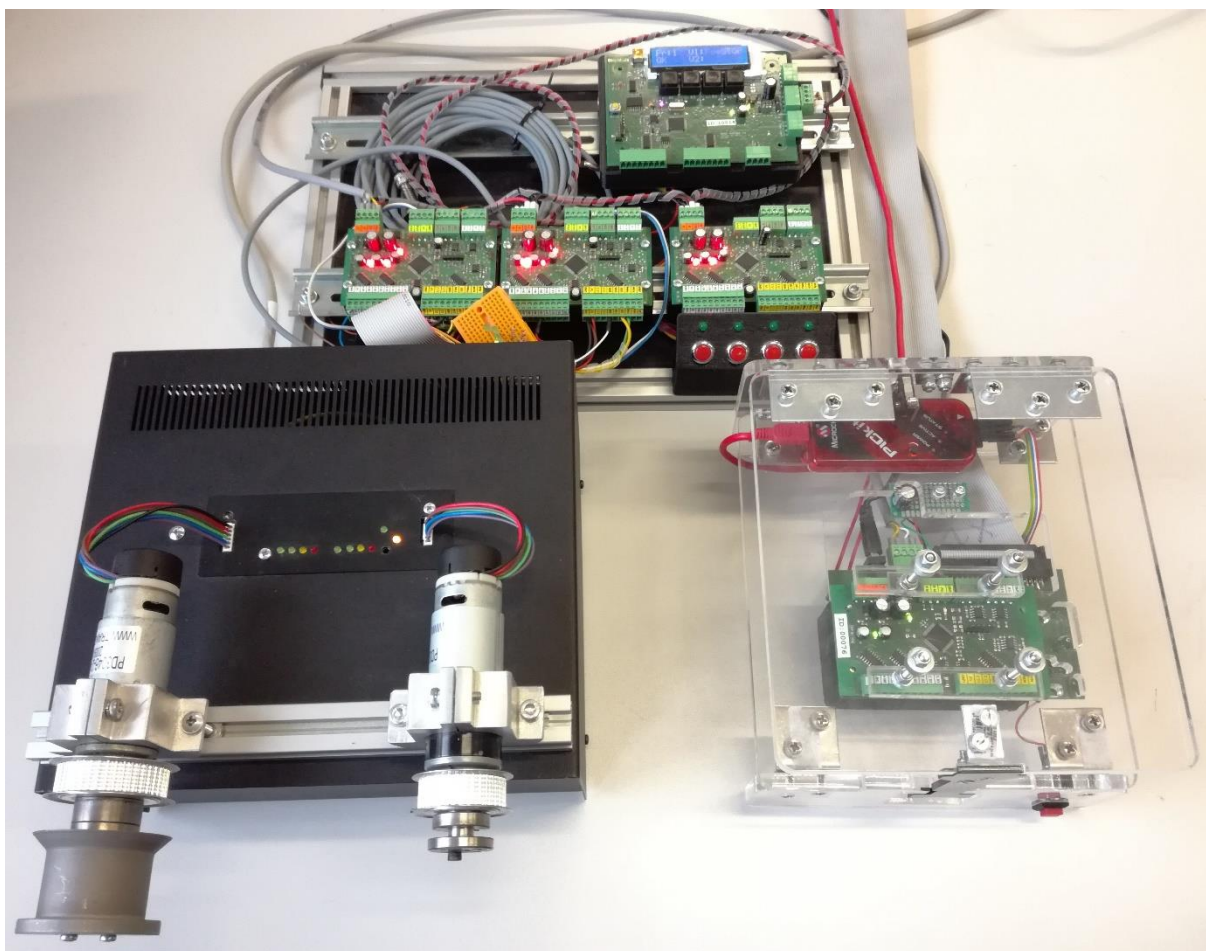
Obrázek 3.19: Porovnání akčních zásahů a směru otáčení



Obrázek 3.20: Porovnání natočení hřídele

3.4 Komplexní testovací soustava

Předposledním bodem zadání byla realizace komplexní testovací soustavy. Pro účely testování chování soustavy více jednotek byl vytvořen kabel, kterým je možné propojit vytvořený testovací přípravek s původním testovacím přípravkem popsáním v bodě 2.5.3. Propojením vznikne testovací soustava, která se skládá z jedné jednotky Master a celkem 4 jednotek Slave, přičemž na 3 jednotkách Slave je možné provádět funkční testy, zatímco na testovací jednotce vložené do navrženého testovacího přípravku v rámci této diplomové práce je možné provádět Hardware in the Loop testy. Fotka celého testovacího řetězce je na obr. 3.21.



Obrázek 3.21: Komplexní testovací soustava pro testování více jednotek

3.5 HIL test chování jednotky v rámci soustavy více jednotek

Po dohodě s vedoucím práce byl tento bod zadání přeformulován tak, že základní HIL test DC motoru je třeba rozšířit o logické řízení. Model pro HIL simulaci byl doplněn o simulaci koncových spínačů pohonu včetně hystereze. Na začátku skriptu se definuje poloha minimálního i maximálního koncového spínače daného pohonu. Protože je v konfiguraci testované jednotky nastaveno, že koncový spínač v okamžiku stlačení připojí digitální vstup testované jednotky k zemi, je na začátku testování nastaven signál z obou koncových spínačů na logickou úroveň 1 (žádný ze spínačů není stlačený, pohon tedy není najetý na koncovém spínači).

Během simulace je v každém kroku na základě natočení rotoru ověřováno, zda nedošlo ke stlačení koncového spínače. Pokud ano, patřičný digitální vstup testované jednotky se nastaví na úroveň logické nuly a vypíše se informace o dosažení koncového spínače do příkazového okna prostředí MATLAB. Vzhledem k tomu, že jsou spínače modelovány včetně hystereze, dochází při najíždění na maximální koncový spínač k sepnutí při větším úhlu, než který je potřeba k jeho rozpojení. Jinými slovy, pokud spínač sepne při poloze rotoru 150° , musí se motor vrátit o několik stupňů, aby došlo k rozepnutí koncového spínače.

4 ZÁVĚR

V teoretické části práce jsou popsány benefity, jež vývojová metoda Model Based Design založená na modelování přináší, pokud je používána pro návrh produktů. Přestože má metoda řadu výhod, není dokonalá a vždy je nutné se zamyslet nad tím, zda je v daném případě její použití výhodné. Úsilí, které musí být vyvinuté pro přechod na metodu Model Based Design, není malé, u některých jednoduchých zařízení se tedy investovaný čas a prostředky nemusí vyplatit. U složitějších systémů však bývá popis pomocí matematických modelů velmi výhodný a umožňuje výrazně zkrátit dobu potřebnou pro vývoj zařízení i náklady s tím spojené.

Při modelování je vždy nutné mít na paměti, že model nikdy nevystihuje realitu dokonale, v důsledku toho není možné mu slepě věřit, ale je nutné si chování modelu vždy ověřit.

Cílem práce bylo provedení rešerše v oblasti testování řídicích jednotek s důrazem na metody MIL, SIL, PIL a HIL. Rešerše byla provedena, jednotlivé metody porovnány, u každé bylo vytvořeno blokové schéma a zasazení do kontextu metody Model Based Design. Výsledky rešerše jsou zaznamenány v teoretické části této práce.

Hlavním cílem práce bylo vytvoření testovacího přípravku pro snadné a rychlé testování elektronických řídicích jednotek. Tento úkol se podařilo splnit, na testovacím přípravku již byla otestována první série testovaných jednotek, při kterém vše fungovalo dle očekávání. Otestovat jednu jednotku zabere přibližně minutu. Na základě proběhlých testů byl mírně přepracován způsob zaznamenávání výsledků testů do textového souboru. Původně byly všechny události zaznamenávány do jednoho textového souboru, nyní je vytvářen nový soubor pro jednotku s novým ID, takže každá otestovaná jednotka má svůj soubor s výsledky testu.

Dalším cílem práce bylo vytvoření HIL testu k otestování schopnosti jednotky řídit DC motor. Model DC motoru byl vytvořen a úspěšně simulován při HIL testu na frekvenci 1 kHz. Nyní je tedy možné ověřovat vlastnosti regulátoru buď na reálném motoru, který je součástí Double Drive, nebo je možné nahradit tento motor jeho simulací v reálném čase.

Ze srovnání průběhů akčních zásahů při reálném experimentu a Hardware in the Loop simulaci na obrázku 3.19 je evidentní, že chování modelu a reálného motoru je téměř shodné. Při skokové změně požadované hodnoty je však akční zásah u HIL simulace větší než u reálné soustavy. To může být způsobeno tím, že rovnice nepopisují reálný motor dokonale, respektive parametry modelu a reálného motoru se mohou mírně lišit.

Průběhy digitálního signálu nesoucího informaci o směru otáčení z experimentu a HIL simulace znázorněné rovněž na obrázku 3.19 jsou až na začátek měření naprosto shodné. Rozdílnost průběhů v čase od 0s do 2,5 s však není významná, protože akční zásah je v tomto časovém intervalu téměř nulový.

Na obrázku 3.20 je znázorněno srovnání natočení hřídele při experimentu a HIL simulaci. Průběhy jsou téměř shodné a odpovídají požadované poloze, je z nich však evidentní, že po skokové změně požadované polohy se reálný motor roztočí a následně zastaví bez překmitu téměř na požadované poloze. Integrovaná složka regulátoru je však příliš slabá, takže nedojde ani během následujících 3 s, po které je požadovaná poloha konstantní, k eliminaci statické odchylky. Důvodem může být Coulombovo tření, které způsobí, že při stojícím rotoru nedokáže malý akční zásah rotor roztočit. To se však dá vyřešit zvýšením odpovídajícího parametru regulátoru v konfiguračním souboru testované jednotky. U HIL simulace dojde po skokové změně požadované polohy k rychlému roztočení, čímž se za krátkou dobu výrazně zmenší regulační odchylka, následně se motor přibrzdí, nicméně díky integrovaní složce má regulátor tendenci eliminovat statickou odchylku a pomalu se tak rotor přibližuje požadované poloze.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] AERENSTRUP, Roger. *Managing Model-Based Design* [online]. Natick: The MathWorks, 2015 [cit. 2019-04-15]. ISBN 978-1512036138. Dostupné z: <https://www.mathworks.com/campaigns/offers/managing-model-based-design.html>
- [2] HILL, Kim, Bernard SWIECKI, Debra MARANGER MENK a Joshua CREGGER. *Just How High-Tech is the Automotive Industry?* [online]. 2014, 2 [cit. 2019-04-17]. Dostupné z: https://autoalliance.org/wp-content/uploads/2017/01/CARReport_Just_How_High_Tech_is_the_Automotive_Industry.pdf
- [3] Save Time. *Controllab Products* [online]. [cit. 2019-05-23]. Dostupné z: <https://www.hil-simulation.com/benefits/save-time.html>
- [4] NAJMAN, Jan. *Dokumentace řídicího systému DINO*. Brno: Mechatronická laboratoř, 2019
- [5] Vývojové modely. *Diagnostika a testování elektronických systémů* [online]. [cit. 2019-04-17]. Dostupné z: <http://www.umel.feec.vutbr.cz/bdts/index.php/embedded-systemy/vyvojove-modely>
- [6] HERNITER, Marc, Zachariah CHAMBERS. *Introduction to Model-Based System Design (Studijní materiály k VŠ kurzu)* [online]. [cit. 2019-04-20]. Dostupné z: <https://www.mathworks.com/academia/courseware/intro-to-model-based-system-design.html>
- [7] Testing embedded software: a survey of the literature. *Information and Software Technology* [online]. 2018, 21 [cit. 2019-04-23]. DOI: 10.1016/j.infsof.2018.06.016. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0950584918301265>
- [8] NIBERT, Jonathan, Marc HERNITER a Zachariah CHAMBERS. Model-Based System Design for MIL, SIL, and HIL. *World Electric Vehicle Journal* [online]. 2012 [cit. 2019-05-06]. ISSN 2032-6653. Dostupné z: <https://www.mdpi.com/2032-6653/5/4/1121>
- [9] Simulating and Testing TargetLink Code. *dSPACE* [online]. [cit. 2019-05-15]. Dostupné z: <https://www.dspace.com/en/inc/home/medien/videos/productvideos/video-tl-simulation.cfm>
- [10] Simulace SIL - Software in the loop. *REX Controls* [online]. [cit. 2019-05-12]. Dostupné z: <https://www.rexcontrols.cz/simulace-sil>
- [11] Simulace Processor In the Loop a Hardware In the Loop. *AUTOMA* [online]. [cit. 2019-05-09]. Dostupné z: http://automa.cz/cz/casopis-clanky/simulace-processor-in-the-loop-a-hardware-in-the-loop-2007_05_34311_2055/
- [12] Hardware-in-the-Loop Testing (HIL) for Real-Time Plant Simulation. *Speedgoat* [online]. [cit. 2019-05-13]. Dostupné z: <https://www.speedgoat.com/applications-industries/applications/plant-simulation-hil>
- [13] KLEIJN, Christian. Introduction to Hardware-in-the-Loop Simulation. *Controllab Products* [online]. [cit. 2019-05-02]. Dostupné z: <https://www.hil-simulation.com/images/stories/Documents/Introduction%20to%20Hardware-in-the-Loop%20Simulation.pdf>
- [14] Picking the Best Electronic Test Solution for Your Product. *Electronic Contract Manufacturer* [online]. [cit. 2019-04-28]. Dostupné z: <https://www.lectronics.net/best-electronic-test-solution/>

- [15] CHATTOPADHYAY, Taisha. How Flying Probe Testing Works for PCB Assembly. Sierra Circuits [online]. [cit. 2019-05-03]. Dostupné z:
<https://www.protoexpress.com/blog/how-flying-probe-testing-works-for-pcb-assembly/>
- [16] Electronic Board Test. SPEA [online]. [cit. 2019-05-03]. Dostupné z:
<https://www.spea.com/electronic-board-test/>
- [17] Kontaktní sondy R-100-CR Technické informace [online]. [cit. 2018-10-18]. Dostupné z: https://cz.mouser.com/datasheet/2/193/72_73-8874.pdf
- [18] Fixture Kits. *Merifix Electronics* [online]. [cit. 2018-12-04]. Dostupné z:
<http://www.merifix.com/products.htm>
- [19] STUHLÍK, Petr. Polymery jako konstrukční materiály. Zlín, 2011. Dostupné také z:
http://digilib.k.utb.cz/bitstream/handle/10563/18081/stuhl%C3%ADk_2011_bp.pdf.
Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta technologická, Ústav výrobního inženýrství. Vedoucí práce doc. Ing. Miroslav Maňas, CSc.
- [20] Planetary Gear DC Motors [online]. *Transmotec* [cit. 2019-04-28]. Dostupné z:
<https://www.transmotec.com/Download/Catalog/Transmotec-EN-PD-0.6W-80W-2019.pdf>
- [21] VALÁŠEK, Michael. *Mechatronika*. Praha: České vysoké učení technické, 1996. ISBN 80-010-1276-X.

SEZNAM OBRÁZKŮ

- Obrázek 1.1: Závislost ceny odstranění chyby na čase jejího objevení [3]
Obrázek 2.1: Testovaná řídicí jednotka (Slave) [4]
Obrázek 2.2: Vodopádový model vývoje [5]
Obrázek 2.3: V model [5]
Obrázek 2.4: Model Based Design aplikovaný na V diagram [7]
Obrázek 2.5: Blokové schéma Model in the Loop
Obrázek 2.6: Blokové schéma Software in the Loop
Obrázek 2.7: Blokové schéma Processor in the Loop
Obrázek 2.8: Blokové schéma Hardware in the Loop
Obrázek 2.9: Hardware in the Loop simulátor ve skříňovém provedení (racku) [12]
Obrázek 2.10: HIL simulátor jako testovací trenažér [13]
Obrázek 2.11: Testování DPS metodou Flying Probes – pohyblivé sondy [15]
Obrázek 2.12: Testování DPS metodou Bed of Nails [16]
Obrázek 2.13: Návrh DPS testovacího přípravku
Obrázek 2.14: Komplexní testovací soustava
Obrázek 3.1: Nákres pogo pinu použitého na testovacím přípravku včetně rozměrů [17]
Obrázek 3.2: Válcové provedení konce pogo pinu s otvorem ve tvaru jehlanu [17]
Obrázek 3.3: Druhá verze konstrukce testovacího přípravku
Obrázek 3.4: Třetí verze konstrukce testovacího přípravku
Obrázek 3.5: Levý pohled na testovací přípravek se zakótovaným umístěním pogo pinů
Obrázek 3.6: Fotka složeného testovacího přípravku
Obrázek 3.7: Schéma dolní propusti s navrženými parametry součástí
Obrázek 3.8: Frekvenční a fázová charakteristika navržené dolní propusti
Obrázek 3.9: Vývojový diagram funkčního testu
Obrázek 3.10: Zařízení Double Drive, testovací přípravek, jednotky Slave a Master
Obrázek 3.11: Vyznačení kladných směrů motoru A na Double Drive DD#01
Obrázek 3.12: Náhradní schéma DC motoru
Obrázek 3.13: Závislost výstupu enkodéru na natočení vstupní hřídele převodovky
Obrázek 3.14: Graf logistické funkce
Obrázek 3.15: Srovnání funkce signum a logistické funkce, která ji nahrazuje
Obrázek 3.16: Model DC motoru pro estimaci parametrů
Obrázek 3.17: Estimace parametrů – naměřená data pro estimaci a průběh estimace
Obrázek 3.18: Požadovaná trajektorie výstupní hřídele
Obrázek 3.19: Porovnání akčních zásahů a směru otáčení
Obrázek 3.20: Porovnání natočení hřídele
Obrázek 3.21: Komplexní testovací soustava pro testování více jednotek

SEZNAM TABULEK

Tabulka 1: Přiřazení pinů karty MF624 na piny testované DPS

Tabulka 2: Katalogové parametry motoru s planetovou převodovkou [20]

Tabulka 3: Výstupy kanálů enkodérů v závislosti na natočení hřídele

Tabulka 4: Estimované parametry a jejich hodnoty

SEZNAM PŘÍLOH

1. Skript v MATLABu pro funkční testování řídicí jednotky
2. Skript v MATLABu pro HIL test řídicí jednotky
3. CAD model testovacího přípravku v SolidWorks
4. Fotografie testovacího zařízení