



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní



ŘÍDICÍ SYSTÉM PRO PŘESNÉ POLOHOVÁNÍ LASEROVÝCH TERČŮ

Diplomová práce

Studijní program: N2301 – Strojní inženýrství
Studijní obor: 2301T049 – Výrobní systémy a procesy
Autor práce: **Bc. Jakub Horáček**
Vedoucí práce: Ing. Michal Moučka, Ph.D.



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní
Akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Horáček**
Osobní číslo: **S13000488**
Studijní program: **N2301 Strojní inženýrství**
Studijní obor: **Výrobní systémy a procesy**
Název tématu: **Řídicí systém pro přesné polohování laserových terčů**
Zadávací katedra: **Katedra výrobních systémů a automatizace**


Zásady pro vypracování:

1. Proveďte průzkum dostupných řídicích modulů pro přesné polohování.
2. Navrhněte systém řízení polohovacího stroje laserových terčů.
3. Systém zrealizujte a naprogramujte. Jako vývojového prostředku využijte prostředí LabVIEW.
4. Ověřte funkčnost a správnost vámi zvoleného řešení.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: cca 45 stran + přílohy
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

- [1] BALÁTĚ, J. *Automatické řízení*. Praha: BEN - technická literatura, 2004. ISBN 978-80-7300-148-3.
[2] KLOBOUČEK, J. *Snímače v průmyslu (skriptum)*. Liberec: Technická univerzita v Liberci, 2012. ISBN 978-80-7372-821-1.
[3] HAVLÍČEK, J., J. VLACH, M. VLACH a V. VLACHOVÁ. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.
[4] MARTÍNEK, R. *Senzory v průmyslové praxi*. Praha: BEN - technická literatura, 2004. ISBN 80-7300-114-4.
[5] ŠVARC, I., M. ŠEDA a M. VÍTEČKOVÁ. *Automatické řízení*. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4398-3.
[6] TRAVIS, J. a J. KRING. *LabVIEW for Everyone - Graphical Programming Made Easy and Fun (third edition)*. NJ(USA): Prentice Hall, 2008. ISBN 0-13-185672-3.

Vedoucí diplomové práce: Ing. Michal Moučka, Ph.D.
Katedra výrobních systémů a automatizace
Konzultant diplomové práce: Ing. Jaroslav Nejdli, Ph.D.
Fyzikální ústav AV ČR
Datum zadání diplomové práce: 5. března 2015
Termín odevzdání diplomové práce: 5. června 2016


prof. Dr. Ing. Petr Lenfeld
děkan




Ing. Petr Zelený, Ph.D.
vedoucí katedry

V Liberci dne 5. března 2015

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 17. 5. 2015

Podpis: 

ANOTACE A KLÍČOVÁ SLOVA

Autor práce: Bc. Jakub Horáček
Vedoucí práce: Ing. Michal Moučka, Ph.D.
Studijní program: N2301 Strojní inženýrství
Studijní obor: 2301T049 – Výrobní systémy a procesy
Datum zadání: 05.03.2015

Téma:

Řídicí systém pro přesné polohování laserových terčů

Anotace:

Cílem práce je navrhnout a zrealizovat řídicí systém Frekvenčního zásobníku terčů vyvinutý ve Fyzikálním ústavu AV ČR. Návrh se skládá z řešerše modulů pro řízení v reálném čase a z popisu použitého zařízení. K realizaci řídicího systému je použito programovací a vývojové prostředí LabVIEW. Pro správnou funkci systému je provedena identifikace soustavy a určení stavitelných parametrů regulátoru.

Klíčová slova:

řídicí systém, řízení v reálném čase, LabVIEW, stavový automat, identifikace soustavy, astatická soustava, PSD regulátor, seřízení regulátoru, stavitelné parametry regulátoru

Theme:

Control system for precision positioning laser targets

Abstract:

The goal of this work is design and implementation of a control system for the Target delivery system which was developed in the Institute of Physics of the Czech Academy of Sciences. The control system design consists of a list of real-time modules and a list of used equipment. The graphical programming platform LabVIEW is used for the implementation of the control system. System identification and tuning of the controller was made for a proper function.

Keywords:

control system, real-time control, LabVIEW, state machine, system identification, astatic process, PSD controller, tuning of controller, parameters of controller

SEZNAM ZKRATEK A SYMBOLŮ

CVS	compact vision system	$e(t)$	regulační odchylka
DC	direct current	$g(t)$	impulsní charakteristika
EC	electronically commutated	$G(s)$	přenos
ELI	extreme light infrastructure	$h(t)$	přechodová charakteristika
FPGA	field programmable gate array	$hm(t)$	modelová přech. charakteristika
FZT	frekvenční zásobník terčů	J	kvadratická plocha
GPIB	general purpose interface bus	k	zesílení
HMI	human machine interface	r_o	proporcionální konstanta
I/O	input/output	r_{-1}	integrační konstanta
NI	national instruments	r_0	derivační konstanta
PAC	programmable automation controller	s	operátor Laplaceovy transformace
PC	personal computer	T	vzorkovací čas
PID	proporcionálně integračně derivační	$T_{1,2,w}$	časová konstanta
PLC	programmable logic controller	T_{ar}	průměrná doba odezvy
PSD	proporcionálně sumačně diferenční	T_d	dopravní zpoždění
RT	real-time	$u(t)$	akční veličina
RTS	real-time system	$w(t)$	žádaná hodnota
RTOS	real-time operating system	$y(t)$	výstupní veličina
SRAM	static random access memory		
qc	quadrature counts		
QMC	queued message handler		
UDP	user datagram protocol		
USB	universal serial bus		
VBAI	vision builder for automated inspection		

OBSAH

ÚVOD	9
1 POLOHOVACÍ STROJ LASEROVÝCH TERČŮ	10
2 NAVRŽENÍ SYSTÉMU ŘÍZENÍ.....	12
2.1 Specifikace systému řízení FZT	13
2.2 Verifikace systému řízení FZT	14
2.3 Řídicí moduly pro přesné polohování	15
2.3.1 Hardwarová platforma PXI	15
2.3.2 NI Compact Vision System.....	16
2.3.3 NI CompactRIO	18
2.4 Zvolené zařízení	19
2.4.1 Pohonná jednotka	19
2.4.2 Řídicí jednotka	20
2.5 Architektura řídicího systému	22
3 REALIZACE SYSTÉMU ŘÍZENÍ.....	23
3.1 Stavový automat	24
3.1.1 Konečný automat Mooreova typu	24
3.1.2 Stavový automat – Camera loop	25
3.1.3 Stavový automat – Motor loop.....	27
3.2 Použité struktury, šablony a návrhové vzory LabVIEW.....	30
3.2.1 Event structure.....	30
3.2.2 Queued message handler	31
3.2.3 State machine	32
3.3 Řídicí program LabVIEW	33
4 REGULACE POLOHOVÉHO SERVOMECHANISMU.....	36
4.1 Identifikace soustavy z přechodové charakteristiky	37
4.1.1 Tříparametrová identifikace integrační soustavy	39
4.1.2 Identifikace nekmitavé integrační regulované soustavy pomocí dob $t_{0,33}$ a $t_{0,7}$.	40
4.1.3 Identifikace minimální kvadratickou plochou.....	42

4.2	Seřízení regulátoru.....	44
4.2.1	Číslicové regulátory	44
4.2.2	Metoda optimálního modulu	45
4.2.3	Metoda kvadratické regulační plochy	46
4.2.4	Metoda inverze dynamiky	47
4.3	Vyhodnocení regulačního pochodu	49
4.3.1	Ověření simulací soustavy	49
4.3.2	Ověření měřením odezvy	51
	ZÁVĚR.....	53
	POUŽITÁ LITERATURA.....	54

ÚVOD

Diplomová práce byla vytvořena ve spolupráci s projektem ELI Beamlines, jehož realizaci má na starosti Fyzikální ústav Akademie věd České republiky. Projekt má za cíl vybudovat laserové výzkumné centrum s nejintenzivnějším laserem světa. Toto zařízení vzniká v Dolních Břežanech ve Středočeském kraji a po dokončení by mělo dodávat ultrakrátké laserové pulsy trvající několik femtosekund a produkovat výkon až 10PW.

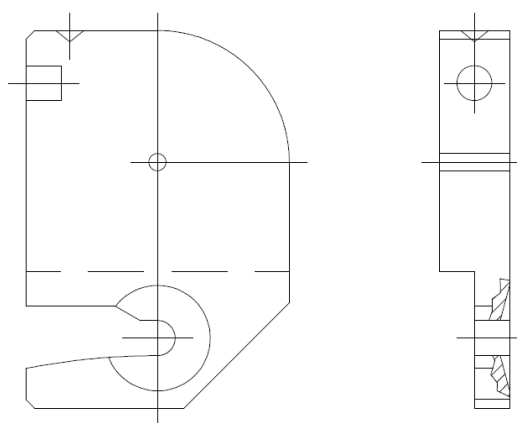
Cílem práce je navrhnout a zrealizovat řídicí systém Frekvenčního zásobníku terčů vyvinutý ve Fyzikálním ústavu AV ČR pro projekty HiPER a ELI Beamlines. Frekvenční zásobník terčů přemísťuje terče ze zásobníku do pracovní polohy, kde dochází k interakci laserového svazku s terčem.

Návrh řídicího systému se skládá z řešení modulů pro řízení v reálném čase, popisu použitých prostředků a návrhu komunikačního diagramu. Aby byl řídicí systém implementovatelný do zbylých systémů řízení ELI Beamlines, je k realizaci použito programovací a vývojové prostředí LabVIEW. Program ovládá pohyb motoru a po přepnutí z manuálního do automatického módu řídí operace bez dalšího nutného zásahu uživatele. Kromě vytvoření programu je pro správnou funkci systému nutné nastavit stavitelné parametry regulátoru. Je provedena identifikace soustavy a určení stavitelných parametrů regulátoru pomocí vhodné metody. Funkčnost nastavení je ověřena simulací a měřením odezvy polohy motoru na skokovou změnu žádané hodnoty.

1 POLOHOVACÍ STROJ LASEROVÝCH TERČŮ

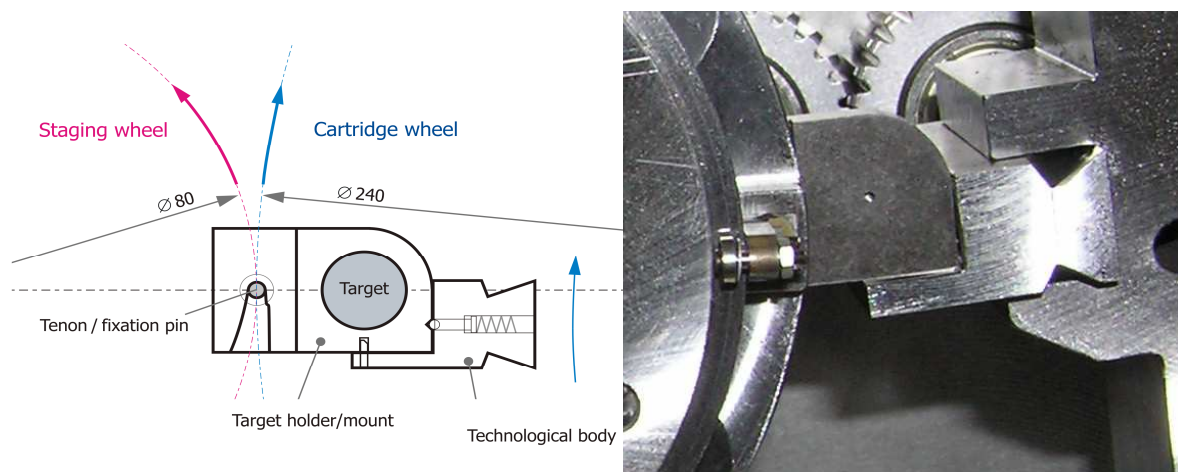
Cílem projektu Frekvenčního zásobníku terčů (FZT) bylo zkonstruovat a vyrobit dopravník, který by manipulačními pohyby přemisťoval terče ze zásobníku do pracovní polohy s vysokou opakovanou přesností polohování a s nominální frekvencí 1Hz, s možností navýšit frekvenci až na 10Hz. Základním principem je, že terč uložený v technologickém tělese (obr.

1) je nejprve dopraven rotorovým kolem zásobníku do aktivní pozice, odkud je odebrán druhým rotorovým kolem do pracovní polohy. Pracovní poloha je poloha, při které dochází k interakci laserového svazku s terčem uloženým v technologickém tělese. První rotorové kolo vykonává současný rotační a suvný pohyb. Na obvodu kola je upevněno 9 lišt, ve kterých jsou pružně uložena technologická tělesa. Druhé



Obr. 1: Technologické těleso terče

rotorové kolo je uloženo rotačně na hřídeli vynášecího rotoru. První a druhé rotorové kolo jsou kinematicky přesně svázány převodem ozubených kol s převodem $i=3$. První rotorové kolo rotuje ve směru pohybu hodinových ručiček, druhé kolo rotuje ve směru proti. V bodě dotyku kružnic (obr. 2) dochází k odebrání terče technologického tělesa z rotorového tělesa

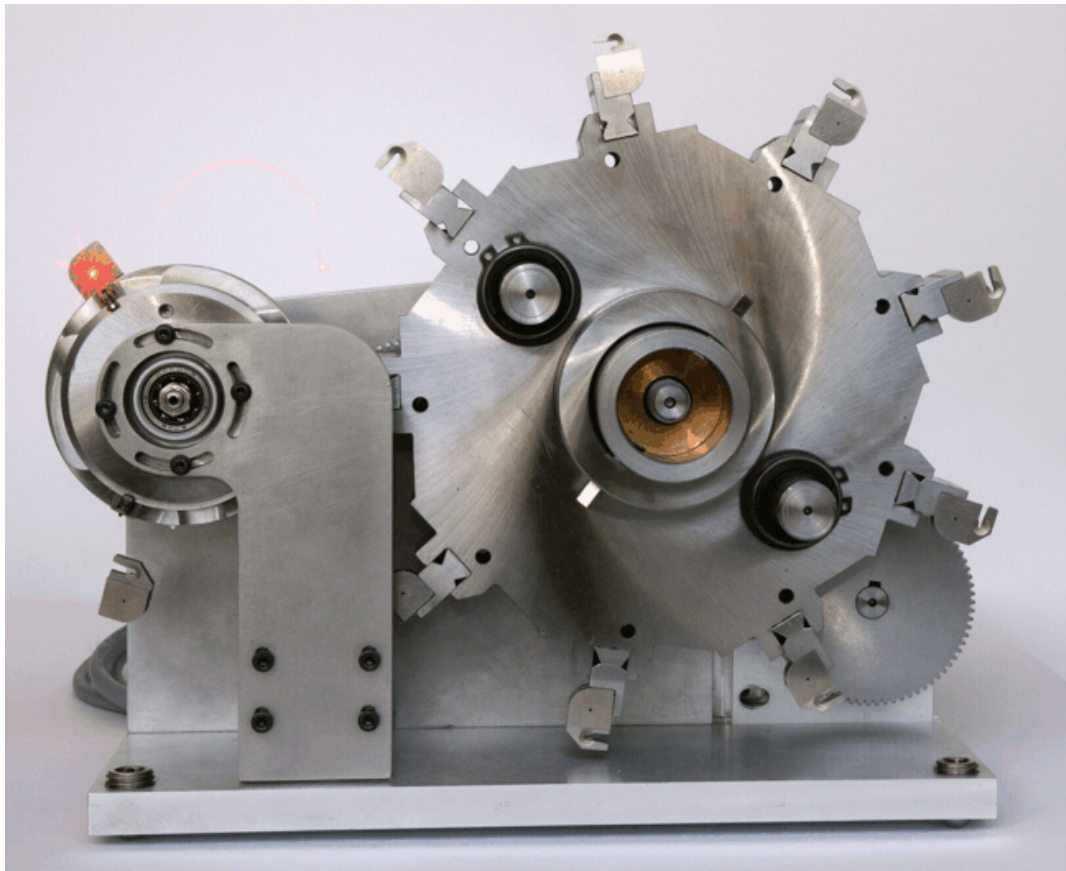


Obr. 2: Odebírání terče technologického tělesa hlavou vynášecího rotoru

zásobníku terčů hlavou vynášecího rotoru. V tomto bodě musí být provedena polohová synchronizace technologického tělesa s aretačními kolíčky v hlavě vynášecího rotoru. Hlava

vynášecího rotoru je opatřena třemi aretačními kolíčky, které jsou rozmístěny po obvodu po 120°.

Konstrukce systému umožňuje navýšit počet terčů dodávaných zásobníkem na devítinásobek množství terčů umístěných v jedné liště. S padesáti terči v liště by byl systém schopný dopravit až 450 terčů. Na obrázku 3 je zobrazen ověřovací prototyp, který byl zkonstruován. Prototyp je vybaven zásobníkem s 8 tělesy uloženými v jedné liště, tedy s celkovým počtem 72 terčů. Je určen k detailnímu testování pojetí konstrukce systému, měření přesnosti doručení terčů a opakovatelnosti. ¹



Obr. 3: Frekvenční zásobník terčů

¹ POLAN, J., HAVLICEK, T., RUS, B. Target delivery system for high repetition rate lasers. In: *Optical Engineering+ Applications*. International Society for Optics and Photonics, 2007. p. 670210-670210-6.

2 NAVRŽENÍ SYSTÉMU ŘÍZENÍ

Provoz frekvenčního zásobníku terčů nebude samostatný systém, ale jeho řízení a pohyb bude závislý na podnětech dalších zařízení laboratoře. Na tyto vstupní podněty bude muset reagovat nejen funkčně správnou odezvou, ale také v předem vymezeném časovém intervalu měřeném od vzniku podnětu. Jedná se tedy o časově kritický systém, označován jako systém pro řízení v reálném čase (dále jen RTS). Odezva v reálném čase je pro tyto systémy důležitým parametrem, a čím vážnější následky může mít nedodržení časových mezí, tím větší jsou požadavky na daný RTS. Podle nedodržení časových mezí kladených na jednotlivé odezvy systému se klasifikují meze odezev do jedné ze tří typových kategorií jako odezva měkká, tvrdá nebo pevná.

V případě měkké odezvy nedodržení mezí vede zpravidla k dočasnému poklesu kvality služeb se zanedbatelnými dopady na okolí systému. Na rozdíl od tvrdé odezvy, kde by nedodržení jediné meze vedlo k nevratným a trvalým následkům v okolí systému (např. zasunování regulačních tyčí do jaderného reaktoru při zjištění poruchy). Pevná odezva se vyznačuje předem danou tolerancí nedodržení mezí. Překročení této meze může mít dopad na okolí systému, ale následky nedodržení mezí tohoto typu jsou typicky vratné.

Podle uvedené klasifikace se mohou rozdělit i RTS, typ je pak určen typem nejkritičtější meze zahrnuté v požadavcích na systém v jeho tzv. specifikaci. Prvotní specifikace RTS je obvykle zapsána s použitím prostředků přirozeného jazyka, protože je typicky používána při komunikaci se zadavatelem. Ta bývá převedena na specifikaci formální, zapsanou např. prostředky logiky reálného času nebo časových automatů. Další etapou vývojového cyklu, tzv. verifikace, se pomocí dotazů ověřuje, zda bude systém vykazovat očekávané vlastnosti nebo naopak, očekávané chování systému nemůže nikdy nastat. Lze se dotazovat na živost systému, bezpečnost, dosažitelnost stavů atd. Tímto způsobem lze odhalit, zda mezi specifikací a ověřovanými vlastnostmi není rozpor a tento rozpor odstranit modifikací či doplněním specifikace.

Po ukončení etap specifikace a verifikace RTS je možné zahájit fázi realizace systému. Před jejím započatím je nutné rozhodnout, které části budou realizovány obvodově, které programově a jaké prostředky se použijí. K hardwarovým základnám používaným při realizaci RTS patří zejména mikroprocesory programované v jazycích C či assembler a programovatelná hradlová pole (FPGA). Je-li specifikace systému jednoduchá, není nutné

oddělit realizovaný RTS od cílového hardwaru. V případě, že je odstiňující vrstva nezbytná, bývá realizována operačním systémem reálného času (RTOS).¹

2.1 Specifikace systému řízení FZT

Řídicí systém FZT by měl mít stejnou koncepci jako ostatní systémy laboratoří ELI Beamlines, aby mohl být implementovatelný do zbylých systémů řízení. K jeho vytvoření by mělo být použito vývojové prostředí LabVIEW. Systém bude tvořen dvěma módy, manuálním a automatickým. V manuálním módu je operátor přímo v laboratoři u frekvenčního zásobníku terčů, který by ve finální verzi měl být umístěn ve vakuu. Pomocí příkazů z řídicího počítače operátor pohybuje s rotorovým kolem zásobníku a osazuje zásobník tělesy terčů. Po zapnutí automatického režimu již není přítomnost obsluhy zapotřebí. Manuální mód:

- Zásobník se přesune do koncové polohy, aby mohl operátor osadit zásobník tělesy terčů.
- Pro lepší přístup operátora při osazování těles terčů do zásobníku prvního rotorového kola je nutné otáčet zásobníkem o nastavitelný úhel.
- Po osazení zásobníku si hlava vynášecího rotoru odebere první terč v řadě a pomocí jemných a přesných pohybů je terč nastaven do dráhy laserového paprsku.
- Po správném nastavení prvního terče zadá operátor počet terčů do systému a spouští automatický mód, tím potvrzuje, že terče byly osazeny a že provedl kontrolu.

Automatický mód:

- FZT zůstává ve stejné pozici, dokud nepřijde řídicí signál laboratoře (tzv. trigger), tento řídicí signál je generovaný v časovacím systému laboratoře.
- Obdrží-li systém tento signál, uvede do pohybu zásobník o jeden krok, tedy otočí hlavu vynášecího rotoru o 120°. Po dokončení pohybu zůstává zásobník na místě, pohyb musí být dokončen před obdržení dalšího signálu.
- Při dokončení pohybu musí být kromě správné polohy, kterou udává pohonná jednotka, nezávisle ověřena poloha tělesa terče kamerou. Systém uloží obrázek pro další analýzu.
- Po dokončení pohybu čeká FZT na další signál k pohybu (trigger) nebo ukončí automatickou sekvenci, pokud provedl předání všech těles terčů. Při ukončení automatické sekvence přechází zpět do manuálního módu.

¹ STRNADEL, Josef. Návrh časově kritických systémů I: specifikace a verifikace. *Automa*. roč. 2010, č. 10, s. 42-44

2.2 Verifikace systému řízení FZT

Na tomto systému lze ověřovat, zda bude při obdržení signálu z časovacího systému hlava vynášecího rotoru nebo těleso terče na správném místě nebo zda systém rozpozná, pokud terč úplně chybí. Verifikace je uvedena v tabulce 1, v jejímž prvním sloupci je dotaz na vlastnost systému zapsaný v přirozeném jazyce, v druhém sloupci výsledek ověření dotazu a v posledním pak doplnění specifikace.

Tab. 1: Verifikace systému řízení

Dotaz na vlastnost systému	Výsledek ověření dotazu	Doplnění specifikace
Platí, že při pohybu do koncové polohy stroje je zásobník terčů prázdný?	Nesplněno	Obsluha bude vyzvána ke kontrole zásobníku před pohybem do koncové polohy.
Platí, že se zásobník při přijmutí signálu nepohybuje?	Splněno	
Platí, že je hlava vynášecího rotoru při přijmutí signálu na správném místě?	Splněno	
Platí, že je těleso terče při průchodu laserového svazku na správném místě?	Nesplněno	Nejedná se o chybu specifikace ale o rys systému.
Platí, že při chybějícím terči v zásobníku tuto chybu systém eviduje?	Splněno	
Platí, že je možné automatickou sekvenci předčasně ukončit?	Nesplněno	Pohyb v manuálním režimu i automatický režim bude možné ukončit příkazem z řídicího systému.
Platí, že po předání všech terčů se ukončí automatický mód?	Splněno	

Modifikace specifikace proběhne u pohybu do koncové polohy, kde nesmí být při reversním pohybu v zásobníku umístěny terče, a bude přidán bod specifikace o možnosti přerušení pohybu či ukončení automatické sekvence před jejím dokončením. Řídicí systém nemůže zaručit, že bude těleso terče vždy na správném místě, to musí být zajištěno konstrukcí frekvenčního zásobníku terčů. Řídicí systém nese odpovědnost za správnou pozici hlavy vynášecího rotoru a pozici terče nezávisle kontroluje kamerou. V případě špatné pozice terče se může na základě této informace prohlásit měření vzorku za neplatné.

2.3 Řídicí moduly pro přesné polohování

Pro řízení strojů a technologických zařízení se v současné době nejčastěji používají specializované řídicí systémy v podobě logických automatů – PLC (Programmable Logic Controller). Často jsou doplňovány kartami analogových vstupů nebo výstupů a dalšími kartami pro regulaci nebo řízení pohonů. Některé aplikace však vyžadují funkce, které jsou za hranicemi možností PLC. Pro tyto aplikace byla vyvinuta nová generace řídicího systému PAC (Programmable Automation Controller). S tímto systémem není problém vytvořit řídicí systém s pokročilými schopnostmi pro analýzu signálu, řízení pohybu, zpracování obrazu apod.

Dle specifikace v kapitole 2.1 by měl řídicí systém FZT kromě řízení pohybu zpracovávat data z kamer a vše by mělo být řízeno pomocí vývojového prostředí LabVIEW. Firma National Instruments nabízí několik zařízení typu PAC, jejichž architektura vychází z možností grafického vývojového prostředí LabVIEW pod operačním systémem reálného času: zdokonalený průmyslový počítač PXI, Compact Vision system, určen především pro zpracování obrazu, a CompactRIO, který obsahuje konektory interní sběrnice pro vložení dalších modulů. Tato zařízení budou podrobněji rozebrána v následujících kapitolách.¹

2.3.1 Hardwarová platforma PXI

PXI (PCI eXtensions for Instrumentation) – (obr. 4)², je odolná platforma založená na PC určená pro měření a automatizaci. Její hlavní výhodou je využití běžných standardů sběrnice PCI, procesorů Intel a operačního systému Windows. Klíčovým prvkem architektury PXI zaručujícím její kompatibilitu a vysoký výkon je zmíněná komunikační sběrnice PCI. Výhodou použití této sběrnice pro měření a řízení je ideální kombinace velkého přenosového výkonu a malé latence. Malá latence je výhodou u úloh, při nichž se posílá velké množství krátkých zpráv, jako je konfigurování digitálního multimetru a přepínačů a měření těmito přístroji.³

Platforma PXI se skládá z šasi, řídicí jednotky a zásuvných modulů. Šasi zajišťuje napájení, chlazení a komunikační sběrnici PCI nebo PCI Express. Řídicí jednotka je postavena na procesorech Intel, od nejnovějšího NI PXIe-8880 s osmijádrovým procesorem Xeon E5-

¹ VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. 1. vyd. Praha: BEN - technická literatura, 2008, 247 s. ISBN 9788073002459.

² [online]. [cit. 2015-04-03]. Dostupné z: <http://rtcmagazine.com/articles/view/101671>

³ ŠTEFAN, Radim. Minulost, současnost a budoucnost standardu PXI. *Automa*. roč. 2006, č. 3

2618L s frekvencí 2,3GHz, po nejlevnější variantu NI PXIe-8820 osazenou dvoujádrovým procesorem Celeron 1020e s taktem 2,2GHz. Platforma PXI umožňuje zvolit si operační systém mezi Windows 7 Professional (64-bit) a NI LabVIEW Real-Time. Operační systém LabVIEW Real-Time je ideálním prostředím pro aplikace náročné na rychlost, které vyžadují deterministické chování. Díky přesnému časování a možnosti přidělit úlohám priority lze v



Obr. 4: Platforma NI PXI

LabVIEW snadno vyvíjet deterministické úlohy, jako je řízení pohybu v uzavřené smyčce. OS Windows pak umožňuje připojení velkého množství I/O periférií od Gigabit Ethernet portů, přes USB3.0 a USB2.0 pro připojení klávesnice, myši, tiskárny nebo paměťových zařízení, až po GPIB rozhraní nebo SMB trigger. Jako zásuvné moduly lze zvolit např. NI PXIe-8234 pro pořizování obrazu z kamer s rozhraním GigE Vision nebo NI PXI-7342 pro řízení krokových nebo servo pohonů.¹

2.3.2 NI Compact Vision System

Compact Vision System – CVS (obr. 5)² je kompaktní průmyslový systém pro zpracování obrazu v reálném čase. CVS umožňuje připojení až dvou kamer pomocí rozhraní USB3 Vision nebo GigE Vision. Řídicí jednotka je postavena na dvoujádrovém procesoru Intel Atom N455 s frekvencí 1,66GHz, výkonnější verze je postavena na čtyřjádrovém procesoru Intel Atom E3845 s taktem 1,91GHz. Operační systém je u nejnovějších verzí CVS-1458RT a

¹ National Instruments. [online]. [cit. 2015-04-03]. Dostupné z: <http://www.ni.com/products/>

² [online]. [cit. 2015-04-03]. Dostupné z: <http://www.ni.com/vision/systems/cvs/>

CVS-1459RT NI Linux Real-Time. Software od firmy National Instruments pro zpracování obrazu nabízí uživateli dvě možnosti přístupu. První a jednodušší spočívá v použití intuitivního prostředí programu Vision Builder for Automated Inspection (VBAI), druhou



Obr. 5: NI Compact Vision System CVS-1457RT

možností je využití flexibility a funkcí programovacího jazyka LabVIEW. V prostředí VBAI je programátorské prostředí nahrazeno interaktivními průvodci, uživatel zde najde nástroje pro výběr zkoumané oblasti, měření doby trvání inspekce, I/O komunikaci a vytvoření uživatelského rozhraní. Koncepce CVS byla navržena s ohledem na všestranné propojení s periferiemi. Kromě již zmíněných rozhraní je možné komunikovat s dalšími zařízeními prostřednictvím sériového rozhraní RS-232, další externí zařízení lze připojit pomocí 24 digitálních vstupů a výstupů. Ty lze použít k připojení synchronizačních a spouštěcích signálů, inspekce tak může být spouštěna asynchronním signálem nebo ji lze synchronizovat s dalšími procesy. Pro zpracování těchto signálů je možné využít programovatelné hradlové pole Spartan 6-LX25.¹

Programovatelná hradlová pole, všeobecně známá pod zkratkou FPGA, jsou integrované obvody s pravidelnou strukturou logických buněk schopných realizovat jednoduché logické funkce. Volitelným propojením těchto buněk lze dosáhnout rozsáhlých komplexních funkcí, k jejichž realizaci by bylo nutné použít mnoho různých obvodů. Technika programování používaná v FPGA je založena na programovatelném přepínači, který je obvykle tvořen buď tzv. antipojistkou nebo buňkou SRAM. Softwarový modul LabVIEW FPGA umožňuje vytvářet vlastní měřicí a řídicí hardware pomocí grafického programování, aniž by bylo třeba umět nižší programovací jazyky běžně používané pro programování hradlových polí.²

¹ ŠTEFAN, Radim. Kompaktní systém pro zpracování obrazu. *Automa*. roč. 2006, č. 5

² DANĚK, Martin. Programovatelná hradlová pole - FPGA. *Automa*. roč. 2006, č. 2

2.3.3 NI CompactRIO

Platforma CompactRIO (obr. 6)¹ v sobě spojuje vestavěnou řídicí jednotku s procesorem, programovatelné FPGA pole a šasi s vstupně výstupními moduly. Nejnovější verze (cRIO-9033) je postavena na dvoujádrovém procesoru Intel Atom E3825 o frekvenci 1,33GHz. Vyšší výpočetní výkon je vykoupen i vyšší pořizovací cenou, proto mohou zákazníci vybírat i



Obr. 6: NI cRIO-9030

z nabídky platforem s dvoujádrovými procesory Xilinx Zynq-7000 s taktem 667MHz (cRIO-9065). Řídicí jednotka cRIO-9082 je pak osazena procesorem Intel Core i7-660UE s vlastností zvanou Hyper-Threading, která zlepšuje práci s několika jádry tím, že dává každému procesoru schopnost zpracovat současně dvě vlákna, čímž se zdvojnásobí počet virtuálních jader, která má software k dispozici. Inženýři a integrátoři systémů si mohou vybrat mezi determinismem a spolehlivostí operačního systému LabVIEW Real-time nebo možností využívat rozsáhlou paletu softwaru a integrovaných funkcí uživatelského rozhraní systému Microsoft Windows Embedded Standard 7. Na NIDays 2014 pak byla představena platforma využívající RTOS NI Linux.² Pomocí gigabitových ethernetových portů lze připojit různé kamery s rozhraním GigE Vision a velkou rychlostí pořizovat a zpracovávat obrazy. Pomocí ovladače NI-IMAQdx a algoritmů v knihovně Vision Development lze vytvářet systémy strojového vidění s pokročilými funkcemi, jako je porovnávání vzorů a měření objektů. Největší výhodou platformy CompactRIO je sběrnice pro vložení 4 až 8 rozšiřujících modulů díky kterým můžeme vytvořit systém přesně „na míru“, např. modul NI 9502 pro řízení bezkartáčkových stejnosměrných motorů či modul NI 9503 pro řízení krokových motorů.³

¹ [online]. [cit. 2015-04-03]. Dostupné z: <http://sine.ni.com/psp/app/doc/p/id/psp-1195/lang/cs>

² NIDays 2014. *Automa*. roč. 2015, č. 1, s. 54

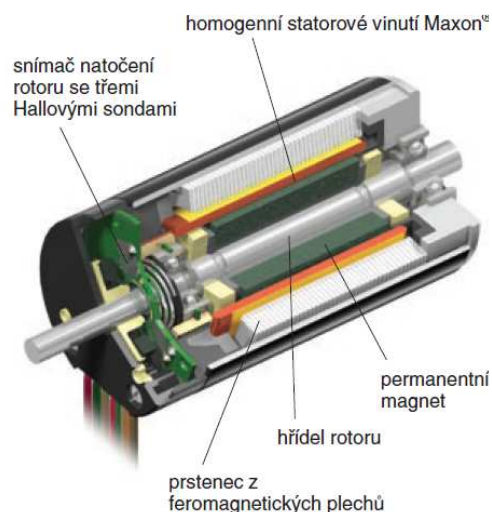
³ Představení výkonného vícejádrového řídicího systému NI CompactRIO. *Automa*. roč. 2012, č. 8-9, s. 50-51

2.4 Zvolené zařízení

Již při vývoji projektu Frekvenčního zásobníku terčů byl pro pohon dopravníku zakoupen stejnosměrný motor Maxon EC45 o výkonu 250W a ovladač Maxon EPOS 70/10 určený pro motory o výkonech 80 až 700W. Jednotka je koncipována pro řízení s komunikací po průmyslové sběrnici CAN nebo přes sériovou linku RS232 a je připojena k řídicímu nadřazenému počítači jak při vkládání parametrů pohybu motoru, tak i během jeho provozu.¹ Pro realizaci řídicího systému bylo rozhodnuto tyto prostředky využít.

2.4.1 Pohonná jednotka

Motor Maxon EC45 (obr. 7)² je stejnosměrný motor s elektrickou komutací. Zatímco u motoru DC se rotor otáčí okolo stojícího permanentního magnetu vloženého do vnitřku vinutí, u motoru EC je splétané vinutí součástí statoru a na rotoru jsou upevněny permanentní magnety. Výhodou těchto motorů je vyšší životnost a dosažitelnost vyšších otáček, v tomto případě může motor dosahovat 5250ot/min bez zatížení, jmenovité otáčky jsou pak 4520ot/min. Mechanická komutace je na rozdíl od DC motorů nahrazena vhodnými elektronickými spínacími obvody. Pro zajištění správného napájení



Obr. 7: Motor EC v částečném řezu

konkrétní fáze vinutí je ale nutné znát relativní polohu rotoru vůči poli statoru. Za tím účelem se u motoru používají Hallové sondy, rozložené po 120 stupních na obvodu statoru.³

Do soustavy je za motorem připojena keramická planetová převodovka GP42C s trvalým přípustným momentem 15Nm, krátkodobým pak až 22,5Nm. Převodovka má 3 stupně, v každém stupni jsou v záběru 3 planetová kola a proto kroticí moment planetové převodovky několikanásobně převyšuje momenty přenášené převodovkami s předlohou. Další výhodou planetové převodovky je odstranění radiální zátěže hřídele motoru, který přenáší pouze mechanický moment. Keramické čepy, které zkracují délku převodovky a zvyšují

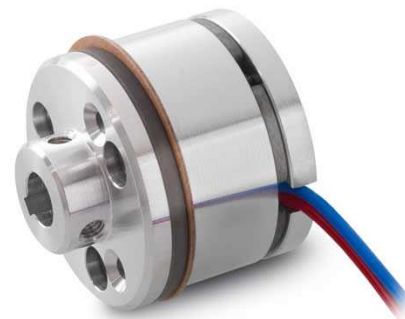
¹ BROŽ, Václav. Programovatelný regulátor rychlosti a polohy pro sběrnici CAN. *Automa*. roč. 2006, č. 1, s. 34.

² BROŽ, Václav. Řízení otáček motorů EC s výkony do 400W. *Automa*. roč. 2007, č. 3.

³ SINGULE, Vladislav. Vlastnosti a použití mikromotorů. *Automa*. roč. 2008, č. 3, s. 62-64

doporučenou rychlost vstupního pastorku se zachováním stejného mechanického momentu, jsou uloženy pouze na koncových stupních. Převod planetové převodovky GP42C je 91:1, takže na výstupu snižujeme jmenovité otáčky motoru na 49,7ot/min.¹

K motoru je také připojena jednostranná stejnosměrná brzda s permanentním magnetem Maxon AB28 (obr. 8)², která zabraňuje rotaci hřídele při klidu nebo při vypnutí napájení motoru. Brzda není vhodná pro brždění otáčející se hřídele motoru, proto je nutné tuto brzdu při zapnutí napájení vždy odbrzdit, jinak by mohlo dojít k poškození motoru a brzdy.³



Obr. 8: Brzda AB28

Poloha hřídele motoru je snímána inkrementálním optickým snímačem Maxon HEDL 9140 s 500 impulsy na otáčku na 3 kanálech. Optické mřížky před průhledným kotoučem s čárovým rastrem po obvodu (obr. 9)⁴ jsou prosvěcovány fotodiodami. Fototranzistor snímá intenzitu světla, jejíž přibližně sinusový průběh se elektronicky tvaruje. Protože rozlišení pozice je čtyřnásobek rozlišení snímače, výsledné rozlišení pozice je 2000qc/otáčku. Nevýhoda inkrementálního snímače je pouze relativní informace o poloze, tzn. že po vypnutí řídicí jednotky se vynuluje i čítač polohy.⁵



Obr. 9: Snímač HEDL

2.4.2 Řídicí jednotka

Jednotka Maxon EPOS 70/10 obsahuje programátor příkazů, který zpracovává požadavky nadřazeného počítače na pohyb. Je možné využít generátor dráhy, vytvářející optimální způsob pohybu do zadané cílové polohy. Regulátory polohy nebo rychlosti jsou zapojeny do smyček zpětné vazby, porovnávají skutečný pohyb s požadovaným a zpracovávají regulační odchylku. Způsob zpracování regulační odchylky regulátorem se určí nastavením jeho

¹ UZIMEX PRAHA. Malé stejnosměrné motory Maxon. [online]. 2002 [cit. 2014-06-04]

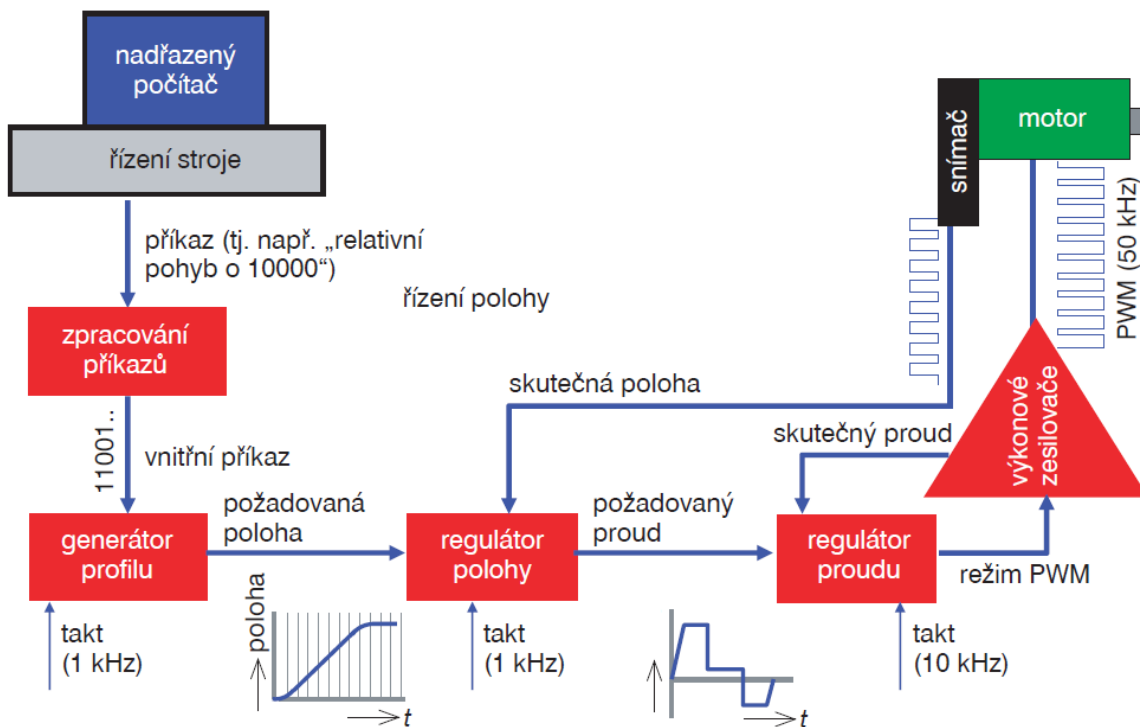
² [online]. [cit. 2014-06-04]. Dostupné z: <http://www.maxonmotor.com/maxon/view/product/228390>

³ [online]. [cit. 2014-06-04]. Dostupné z: <http://www.uzimex.cz/Sortiment/Elektricke-pohony/maxon-motor-ag/Brzdy/AB-28.html>

⁴ [online]. [cit. 2014-06-04]. Dostupné z: <http://www.maxonmotor.com/maxon/view/product/137959>

⁵ BROŽ, Václav. Snímače stejnosměrných motorů do 400W. *Technika*. roč. 2008, č. 4, s. 34-35.

proporcionálního zesílení, integrační a derivační časové konstanty. Výstup z regulátoru polohy je přiváděn do regulátoru proudu, který ovládá výkonové zesilovače koncového výkonového stupně napájecího motoru. Na vstupy podsystému I/O řídicí jednotky se mohou přivést další informace o stavu pohonu jako koncové a referenční spínače. Schéma řídicí jednotky EPOS je na obrázku 10¹. Jednotka může pracovat v několika nastavitelných módech.



Obr. 10: Uspořádání řídicí jednotky EPOS

V proudovém módu se uplatní pouze regulátor proudu a výkonové zesilovače, nastavuje se požadovaná hodnota v mA a řízením proudu je řízen moment motoru. V polohovém a rychlostním módu je možné volit mezi dvěma způsoby vytváření odezvy na změnu požadované hodnoty a to buď s vytvořením profilu (hodnota se nejprve zpracuje v generátoru trajektorie nebo rychlosti) nebo s využitím odezvy regulátorů.²

Zařízení je vybaveno digitálními I/O - konektor J5. Na pinu 10 (DigOUT 4) je připojena permanentní magnetická brzda AB28, na pinu 3 (DigIN 6) je připojen signál z negativního koncového spínače, na pinu 4 (DigIN 5) je připojen signál z pozitivního koncového spínače. Signál připojený k těmto digitálním vstupům je generován v optickém senzoru SunX FX-311.³

¹ BROŽ, Václav. Programovatelný regulátor rychlosti a polohy pro sběrnici CAN. *Automa*. roč. 2006, č. 1.

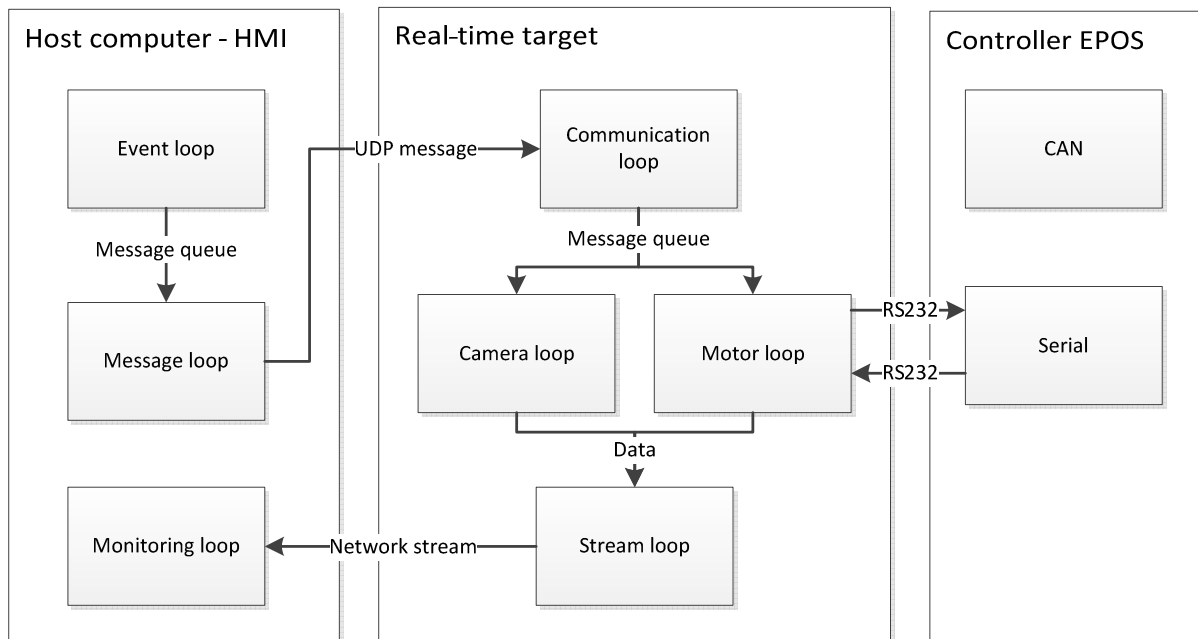
² Tamtéž.

³ MAXON MOTOR. *EPOS 70/10 Positioning controller: Hardware Reference*. 752380-04. Sachseln, 2008.

Dostupné z: http://www.maxonmotor.ch/medias/sys_master/8803613802526/300583-Hardware-Reference-En.pdf

2.5 Architektura řídicího systému

Návrh architektury řídicího systému je zobrazen na diagramu na obrázku 11. Architektura se skládá z uživatelského PC, PAC s operačním systémem reálného času a z pohonné řídicí jednotky EPOS. Na PC i PAC běží samostatné aplikace, které jsou založeny na nezávislých



Obr. 11: Návrh komunikačního diagramu řídicího systému

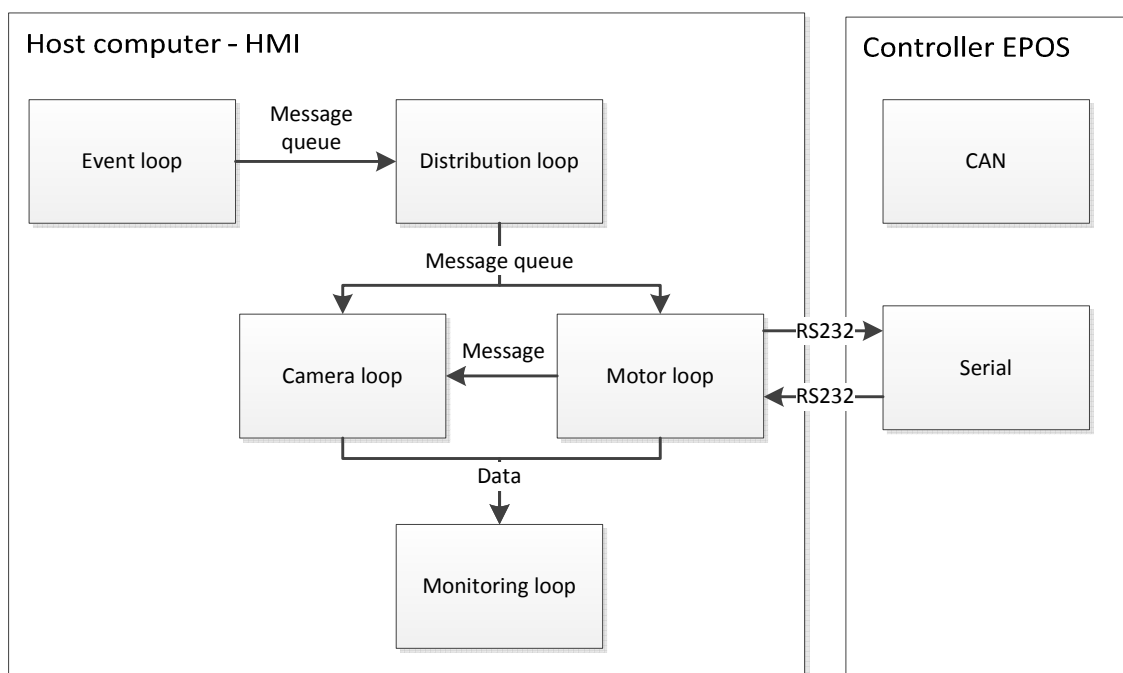
smyčkách, které svou činností a vykonáváním příkazů nemohou zpomalovat běh ostatních smyček. Tím je zajištěna včasná odezva na podněty a dodržení časových mezí pro vykonání příkazu.

HMI (Human Machine Interface), aplikace běžící na hostitelském počítači, pomocí smyčky Event loop zachycuje události vzniklé na základě akce uživatele (obsluhy dopravníku). Tato akce je způsobena stiskem tlačítka myši či klávesou v grafickém rozhraní. Událost je přenesena pomocí zprávy do nezávislé smyčky Message loop, kde je vytvořen příkaz, který je poslán přes síť pomocí protokolu UDP. Komunikační smyčka na Real-time počítači monitoruje příchozí zprávy na portu a předává je dál do procesních smyček. Princip obou procesních smyček na RT target je založen na stavovém automatu. Smyčka Motor loop komunikuje přes sériovou linku RS 232 s jednotkou EPOS a řídí ji. Pozice a informace o motoru nejsou zasílány periodicky řídicí jednotkou, ale pouze jako odpověď na dotaz procesní smyčky. Proto slouží tato smyčka i jako smyčka monitorovací. Smyčka Camera loop nezávisle ověřuje polohu tělesa terče zpracováním snímků z kamery. Data jsou pak zasílána pomocí Network streams do hostitelského počítače, kde jsou zobrazována uživateli.

3 REALIZACE SYSTÉMU ŘÍZENÍ

Architektura řídicího systému bude realizována zjednodušeně podle diagramu na obrázku 12. Skládá se pouze z uživatelského PC a pohonné řídicí jednotky EPOS, se kterou komunikuje přes sériovou linku RS 232. I když je systém realizovaný bez platformy s RTOS, je zachována architektura založená na nezávislých smyčkách a v případě rozšíření řídicího systému o PAC jsou tyto smyčky použitelné pro RT program navržený v kapitole 2.5.

Smyčka Event loop zaznamenává události způsobené uživatelem na předním grafickém panelu programu a vytváří zprávu, která je zasílána do Distribution loop. Tím je zajištěn stejný výstup ze smyčky a mohla by být použita bez modifikací i pro RT verzi řídicího



Obr. 12: Komunikační diagram realizovaného řídicího systému

systému. Smyčka Distribution loop je mezičlánek mezi procesy a Event loop. Rozesílá příchozí zprávy do jednotlivých procesů a pro RT verzi by byla nahrazena UDP komunikací. Přítomnost této smyčky je nutná, protože zpráva z Event loop nemusí být totožná s příkazem, který očekává proces (např. proces kamery očekává příkaz ,Initialize', stejný příkaz ale očekává i proces motoru s jinými daty, takže výstup z Event loop by měl být ,Initialize camera' a ,Initialize motor'), proto nemohou být propojeny napřímo. Procesy Camera loop a Motor loop jsou založeny na stavovém automatu, který bude popsán v následujících kapitolách, a tyto procesy jsou totožné pro RT i zjednodušenou verzi řídicího programu. Smyčka Monitoring loop načítá data z procesních smyček a napřímo je zobrazuje uživateli. V RT verzi by musela být nahrazena Network stream komunikací.

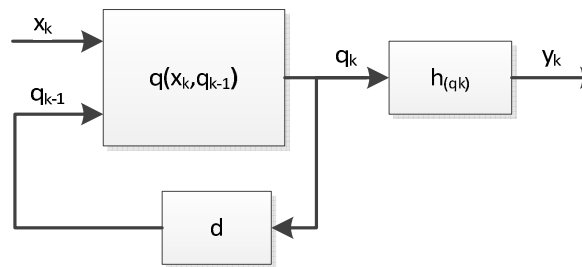
3.1 Stavový automat

Stavový automat se skládá z různých stavů a přechodových funkcí, které vedou do dalších stavů. Je mnoho druhů konečných stavových automatů, z nichž nejčastějším bývá Mooreův automat a Mealyův automat. Pro naprogramování stavového automatu se v LabVIEW používá návrhového vzoru State machine, který vykonává automat Mooreova typu. Velkou výhodou návrhového vzoru State machine v LabVIEW je velikost blokového diagramu, což souvisí s jeho čitelností a poměrně jednoduchou rozšiřitelností o další stavy či možnosti úpravy stavů na rozdíl od kaskádově vytvářeného programu.

3.1.1 Konečný automat Mooreova typu

S využitím automatu Mooreova typu (obr. 13) je možné popsat požadovanou činnost, postup řízení, ale i zásady doporučené obsluhy nebo řešení technické diagnostiky většiny mechanismů nebo výrobních procesů.

Jejich přechodová funkce $q_k = g(x_k, q_{k-1})$ zpracovává současný vzorek vstupních proměnných x_k spolu s informací o aktuálním kroku q_{k-1} a generuje informaci o novém kroku q_k . Takto je realizován



Obr. 13: Automat Mooreova typu

přechod. Symbol d znázorňuje časové

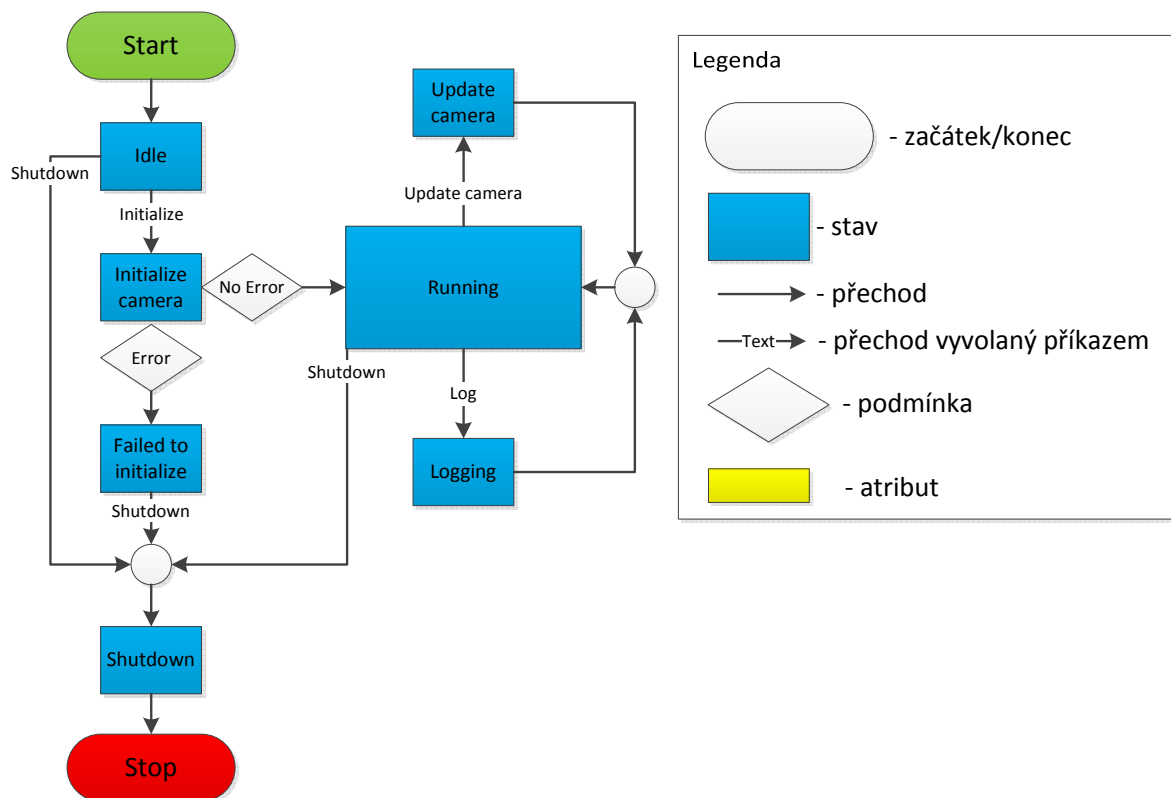
zpoždění o jeden výpočetní krok mezi právě generovaným q_k a minulým krokem q_{k-1} .

Výstupní funkce zpracovává informaci o kroku a generuje hodnoty výstupních proměnných $y_k = h(q_k)$. Nemění-li přechodová funkce hodnotu kroku (konečný automat setrvává v nezměněném kroku, nedochází k přechodu), nemění se ani hodnoty výstupních proměnných. Takovéto přechody obvykle ani nejsou zakreslovány v přechodovém diagramu – kresleny jsou jen přechody, které směřují do jiného kroku. Konečné automaty Mooreova a Mealyho typu jsou někdy souhrnně nazývány jako stavové automaty. Mají stejnou algoritmickou mohutnost a mohou realizovat i velmi složité sekvenční funkce.¹

¹ ŠMEJKAL, Ladislav, ČERNÝ, Josef. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 19). *Automa*. roč. 2014, č. 3, s. 40-43.

3.1.2 Stavový automat – Camera loop

Stavový stroj smyčky Camera loop je zobrazen pomocí stavového diagramu na obrázku 14 včetně legendy, která vysvětluje použité bloky. Po startu programu přejde stavový stroj automaticky do stavu Idle a v něm nečinně setrvává. V tomto stavu je možné příkazem ‚Shutdown‘ program ukončit nebo příkazem ‚Initialize‘ přejít do stavu Initialize camera, kde se otevře reference ke kameře, nastaví se typ akvizice a alokuje se paměť pro obrázky. Pokud došlo k chybě (např. není připojena kamera), setrvává automat ve stavu Failed to initialize, dokud není příkazem ‚Shutdown‘ smyčka ukončena. Pokud vše proběhlo bez chyby, začne



Obr. 14: Stavový diagram smyčky Camera loop

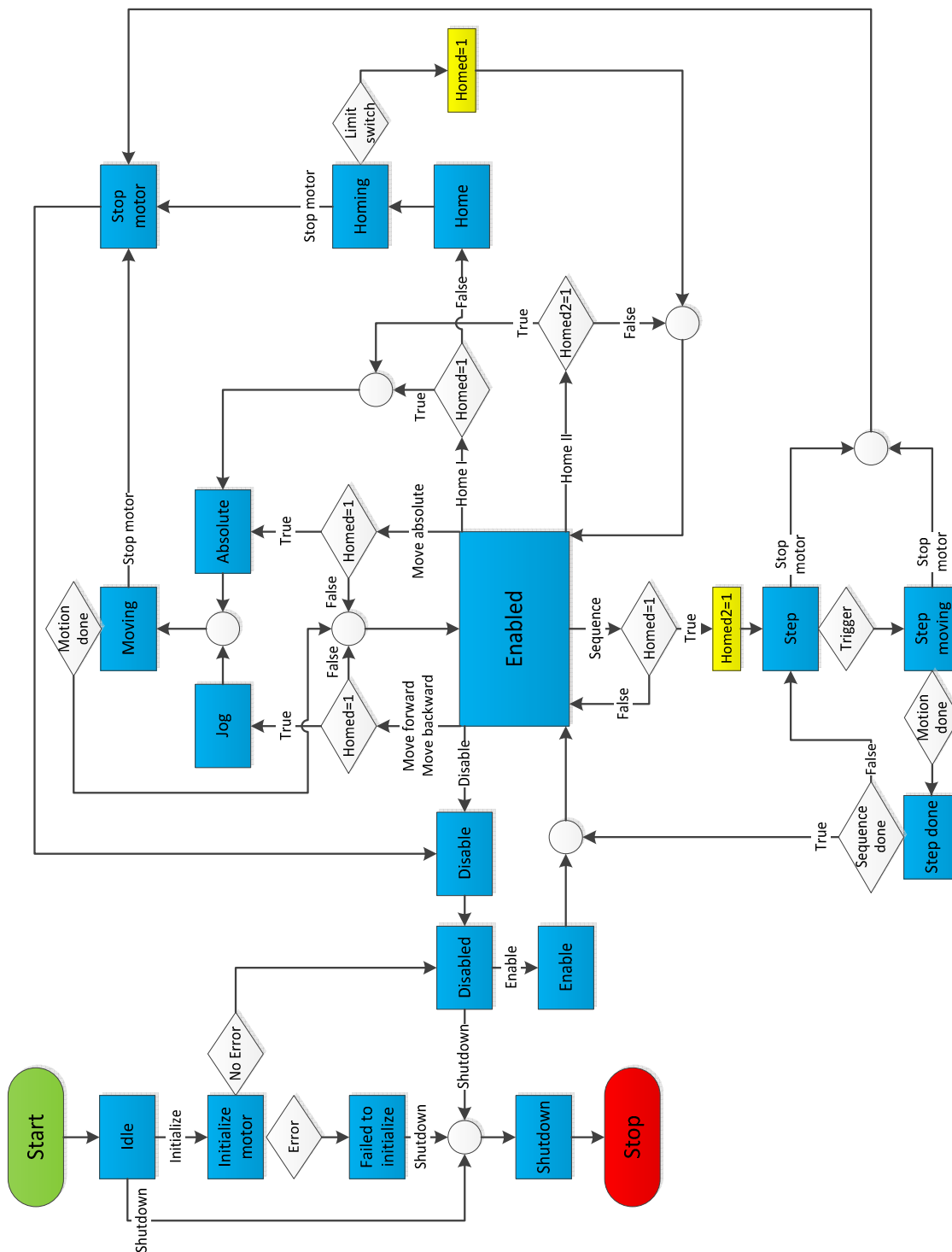
kamera ve stavu Running snímat obrázky z kamery a předávat je do další smyčky, která je odesílá k zobrazení do uživatelského rozhraní. Zprávou ‚Update camera‘ lze ve stavu stejného jména měnit nastavení kamery – dobu expozice a zesílení citlivosti. Poté se automat vrátí do stavu Running. Při příkazu ‚Log‘ automat přejde do stavu Logging, kde dojde k uložení aktuálního obrázku do souboru na disku. V tabulce 2 je uveden popis všech stavů, výpis příkazů, kterými do daných stavů můžeme přejít, a všechny předchozí a následující stavy pro jednotlivý stav.

Stav	Příkaz	Předchozí stav	Následující stav	Popis
Idle	X	X	Initialize camera Shutdown	Nečinný.
Initialize camera	Initialize	Idle	Running Failed to initialize	Vytvoření reference pro kameru, nastavení typu akvizice a alokace paměti.
Failed to initialize	X	Initialize camera	Shutdown	Nečinný, inicializace se nezdařila
Shutdown	Shutdown	Idle Failed to initialize Running	X	Ukončení akvizice a reference kamery, uvolnění paměti a ukončení běhu smyčky.
Running	X	Initialize camera Update camera Logging	Update camera Logging Shutdown	Snímání obrazu z kamery.
Update camera	Update camera	Running	Running	Změna nastavení – expozice a gain.
Logging	Log	Running	Running	Uložení obrázku na disk.

Tab. 2: Popis stavů smyčky Camera loop

3.1.3 Stavový automat – Motor loop

Na obrázku 15 je zobrazen stavový diagram smyčky Motor loop. Stejně jako předchozí automat přejde po startu programu do stavu Idle. Po bezchybné inicializaci přejde do stavu Disabled. Příkazem ,Enable' se jednotka EPOS nastaví do stavu Enable a je deaktivována permanentní brzda motoru. Ze stavu Enabled je možné přejít do stavů, které pohybují se



Obr. 15: Stavový diagram smyčky Motor loop

zásobníkem terčů. Jak je uvedeno v kapitole 2.4.1, motor je vybaven inkrementálním snímačem, a proto je nutné po restartu programu nejdříve najet s dopravníkem na koncový snímač. Ve stavu Home se nastaví rychlostní mód řídicí jednotky EPOS a motor se začne otáčet v negativním směru. Automat přejde do stavu Homing. Ve chvíli, kdy je sepnut limitní snímač, se motor zastaví, do proměnné Homed typu boolean se uloží hodnota jedna (True) a uloží se také softwarové limity pro následný pohyb. Pokud během běhu programu přijde znovu příkaz ‚Home‘, jsou stavy Home a Homing nahrazeny absolutním pohybem do negativního softwarového limitu, který byl ve stavu Homing uložen. Po osazení zásobníku tělesy terčů a nastavení prvního terče do dráhy laserového paprsku je příkazem ‚Sequence‘ spuštěn automatický mód. Do proměnné Homed2 typu boolean je uložena hodnota True a pozice prvního terče je uložena do paměti (program nesmí být ukončen). Příkazem ‚Home II‘ je pak možné první terč dopravit do dráhy laserového paprsku, aniž by musela obsluha znovu nastavovat první terč do správné polohy v manuálním módu. Ve stavu Step je čten digitální vstup, na který je napojen řídicí signál trigger. Pokud je hodnota True, je zadán pohyb o jeden krok (terč) a ve stavu Step moving je načítána pozice motoru, dokud není pohyb dokončen. Stav Step done pak zašle zprávu (příkaz) ‚Log‘ do smyčky Camera loop k uložení obrazu. Počet dokončených kroků je navýšen o jeden, a pokud je počet požadovaných kroků větší než počet dokončených kroků, přechází automat do stavu Step a postup se opakuje. Pohyb v manuálním i automatickém režimu je možné ukončit příkazem ‚Stop motor‘. Všechny stavy jsou podrobně popsány v tabulce 3.

Stav	Příkaz	Předchozí stav	Následující stav	Popis
Idle	X	X	Initialize motor, Shutdown	Nečinný
Initialize motor	Initialize	Idle	Failed to initialize, Disabled	Otevření a nastavení komunikace, nastavení regulátorů
Failed to initialize	X	Initialize motor	Shutdown	Nečinný, selhala inicializace
Shutdown	Shutdown	Idle, Failed to initialize, Disabled	X	Ukončení komunikace přes sériový port
Enable	Enable	Disabled	Enabled	Odbrzdění permanentní brzdy, nastavení EPOS na enable
Enabled	X	Enable, Moving, Homing, Step done	Disable, Jog, Absolute, Home, Step	Nečinný, motor je připraven k pohybu
Disable	Disable	Enabled, Stop motor	Disabled	Nastavení EPOS na disable, aktivování permanentní brzdy
Disabled	X	Initialize motor, Disable	Enable, Shutdown	Nečinný, motor je zabrzděn
Home	Home I	Enabled	Homing	Aktivování koncového snímače, nastavení EPOS na rychlostní mód, zadání pohybu konstantní rychlostí
Homing	X	Home	Stop motor, Enabled	Načítání polohy, čekání na aktivaci limitního snímače
Stop motor	Stop motor	Moving, Homing, Step, Step moving	Disable	Nouzové zastavení motoru, popř. automatické sekvence
Jog	Move forward Move backward	Enabled	Moving	Zadání relativního pohybu motoru
Absolute	Move absolute	Enabled	Moving	Zadání absolutního pohybu motoru
Moving	X	Jog, Absolute	Stop motor, Enabled	Načítání polohy, čekání na dokončení pohybu
Step	Sequence	Enabled	Stop motor, Step moving	Při načtení signálu Trigger zadání pohybu o jeden krok
Step moving	X	Step	Stop motor, Step done	Načítání polohy, čekání na dokončení pohybu
Step done	X	Step moving	Step, Enabled	Zaslání příkazu kamerové smyčce k uložení obrazu, kontrola počtu kroků a opakování či ukončení sekvence

Tab. 3: Popis stavů smyčky Motor loop

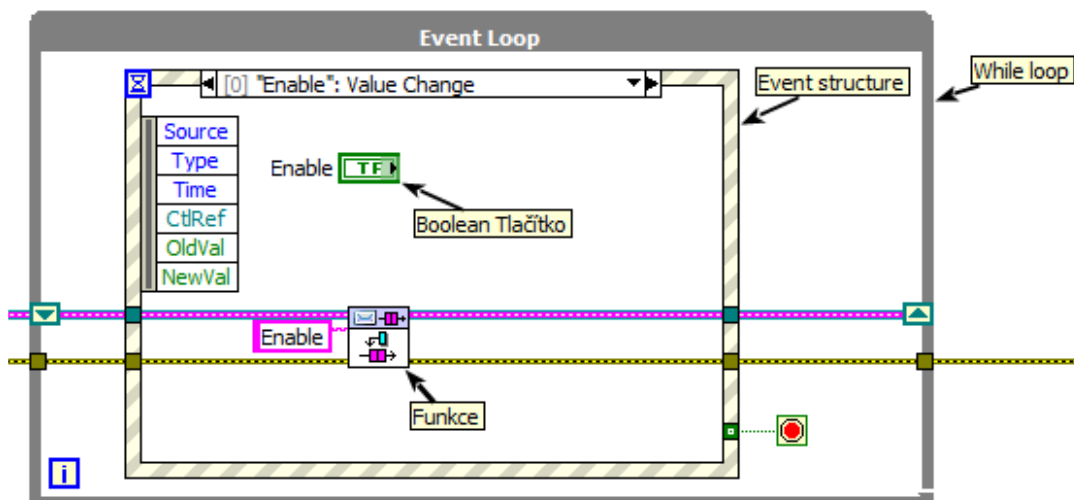
3.2 Použité struktury, šablony a návrhové vzory LabVIEW

Programovací a vývojové prostředí LabVIEW je produktem americké firmy National Instruments, která je průkopníkem a největším výrobcem v oblasti virtuální instrumentace. Prostředí LabVIEW, někdy nazývané též jako G-jazyk (tedy „grafický“ jazyk), je vhodné nejen k programování systémů pro měření a analýzu signálů, řízení a vizualizaci technologických procesů, ale také k programování složitých systémů.¹

Vysvětlit a popsat všechny použité funkce v řídicím programu je mimo rámec této diplomové práce, pro podrobnější studium viz literatura *Začínáme s LabVIEW*², *LabVIEW for everyone: graphical programming made easy and fun*³ a *LabVIEW graphical programming*⁴. Zde budou popsány pokročilejší struktury a šablony, které jsou pro funkci řídicího systému podstatné.

3.2.1 Event structure

Struktura Event structure je založena na metodě vykonání funkce (funkcí) a změny datového toku na základě události. Takovou událostí může být kliknutí myši či stisk klávesnice na čelním panelu. To samozřejmě naznačuje použití této struktury pouze v grafickém rozhraní HMI na uživatelském PC, protože RT target nemá žádný front panel, na kterém by mohla událost vzniknout. Event structure je většinou umístěna do smyčky while loop, jinak by



Obr. 16: Smyčka Event loop zaznamenávající události Front panelu

¹ VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. 1. vyd. Praha: BEN - technická literatura, 2008, 247 s. ISBN 9788073002459.

² Tamtéž

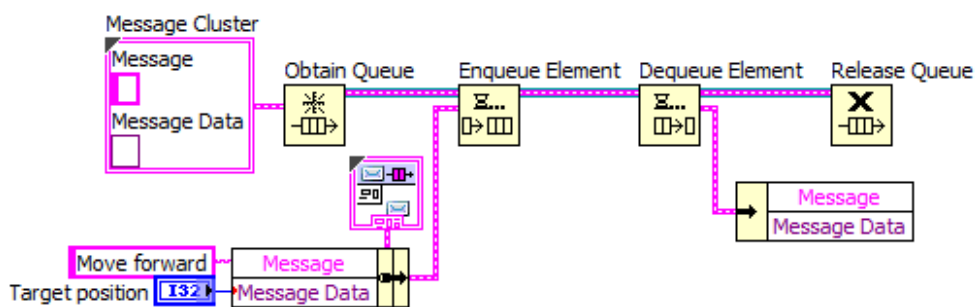
³ TRAVIS, Jeffrey a Jim KRING. *LabVIEW for everyone: graphical programming made easy and fun*. 3rd ed. /. Upper Saddle River, NJ: Prentice Hall, c2007, xli, 981 p. ISBN 9780131856721.

⁴ JOHNSON, Gary W a Richard JENNINGS. *LabVIEW graphical programming*. 4th ed. New York: McGraw-Hill, c2006, xv, 608 p. ISBN 0071451463.

zaznamenala pouze jednu událost. Příklad použití Event structure je na obrázku 16. V tomto případě dojde změnou hodnoty boolean tlačítka Enable ke vzniku události a k provedení vnitřku struktury, tedy funkce, jejíž vstup je proměnná string s hodnotou ‚Enable‘. Event structure funguje podobně jako struktura Case, může tedy obsahovat mnoho případů a každý případ může kontrolovat více událostí najednou. Každý Event nese informaci o čase, kdy se vyskytl, jakou událostí byl spuštěn či nese informaci o nové a původní hodnotě proměnné. Event structure používá k uložení událostí frontu. Ve chvíli, kdy je událost vložena do této fronty, je uzamčen přední panel, dokud není událost zpracována. V tu chvíli LabVIEW nezpracovává žádné akce na předním panelu, ale ukládá je do bufferu pro pozdější zpracování.

3.2.2 Queued message handler

Šablona Queued message handler slouží ke komunikaci paralelních nezávislých smyček pomocí zpráv (Message). Tyto zprávy se skládají z příkazu a dat nutných k provedení tohoto příkazu. Princip je založen na použití funkcí fronty (Queue), tím umožňuje smyčkám předávat si data, aniž by byl omezen nebo zpomalen jejich průběh. Funkce používané pro QMH jsou zobrazeny na obrázku 17. Před prvním použitím je nutné vytvořit frontu pomocí funkce Obtain Queue. Vstupem této funkce je element data type, což je v případě QMH Message cluster skládající se z příkazu (většinou typu string nebo enum) a dat typu variant. Příkazce,

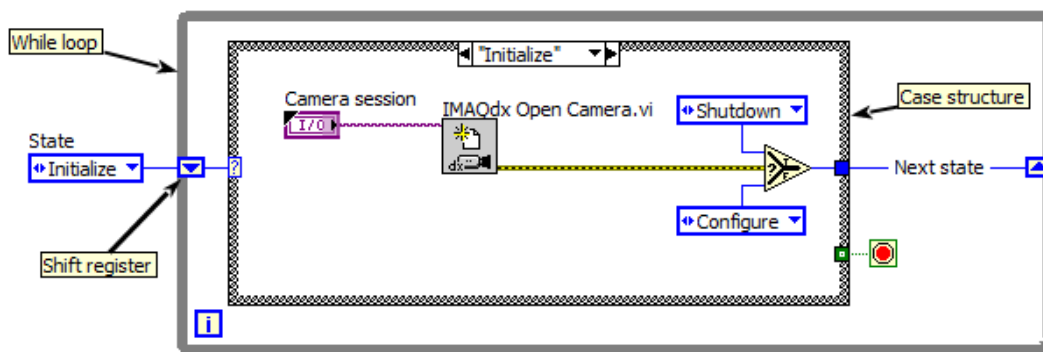


Obr. 17: Funkce používané pro queued message handler template

např. smyčka Event loop, přidá do fronty pomocí funkce Enqueue Element příkaz ‚Move forward‘ a požadovanou pozici typu integer. Příjemce, v našem případě proces motoru, pak při každé iteraci funkcí Dequeue element zkontroluje frontu a vyjme message z fronty a příkaz s daty zpracuje. V případě, že je fronta prázdná, výstupem funkce je výchozí hodnota, tedy prázdný cluster. Na konci programu je pak vhodné frontu ukončit a tím dealokovat paměť, která byla pro frontu vyhrazena.

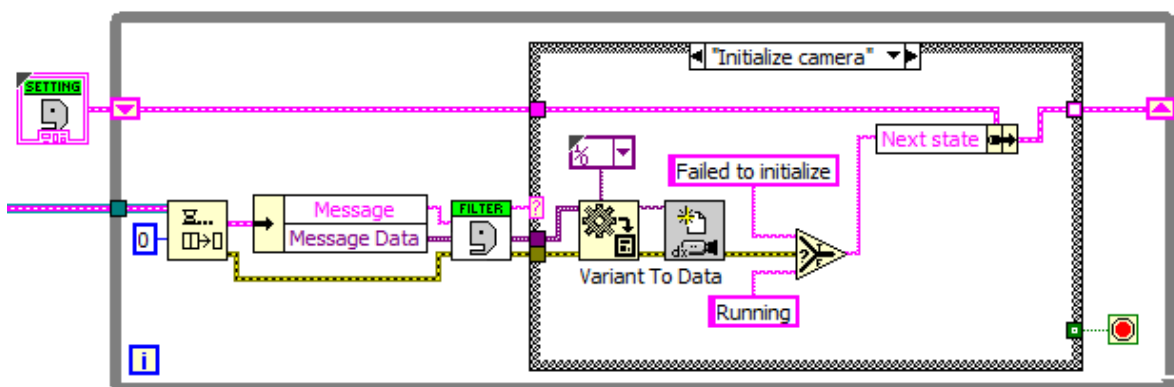
3.2.3 State machine

Návrhový vzor State machine se skládá ze smyčky while loop, case structure a shift register, který může nést informaci o následujícím přechodu či další potřebná data pro následující stav. Princip State machine umožňuje provádět při každé iteraci různý kód programu v závislosti na dané podmínce. Na obrázku 18 je ukázka jednoduchého stavového stroje s takovou podmínkou. Prvním stavem je stav Initialize definovaný jako výchozí hodnota pro shift register. Poté je proveden kód uvnitř case structure, tedy otevření reference kamery. Rozhodovací podmínkou je error, který může být touto funkcí vygenerován. Pokud je Status Error pravda, pak následující stav je Shutdown, pokud je nepravda, následující stav je Configure. Vstupem Case structure nemusí být jen výčetový typ enum, ale také datový typ



Obr. 18: Návrhový vzor State machine

string. To nabízí včlenění šablony Queued Message handler do návrhového vzoru State machine. Tím vznikne možnost řídit stavové stroje uvnitř paralelních nezávislých smyček, viz obrázek 19. Filtr stavů rozhodne o tom, je-li příchozí příkaz mezi povolenými pro tento stav a dojde k přechodu nebo stavový stroj setrvá ve stejném stavu. Např. při selhání inicializace a přechodu do stavu Failed to initialize je mezi povolenými příkazy pouze ‚Shutdown‘.

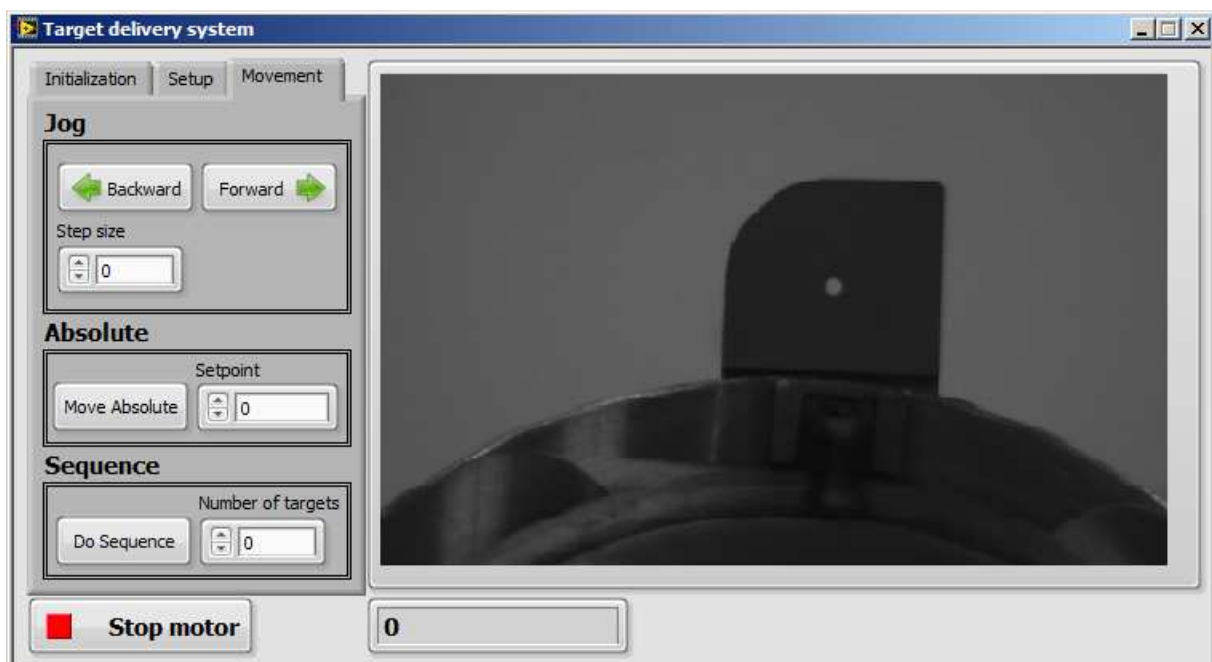


Obr. 19: Návrhový vzor State machine s včleněním QMH šablony

3.3 Řídicí program LabVIEW

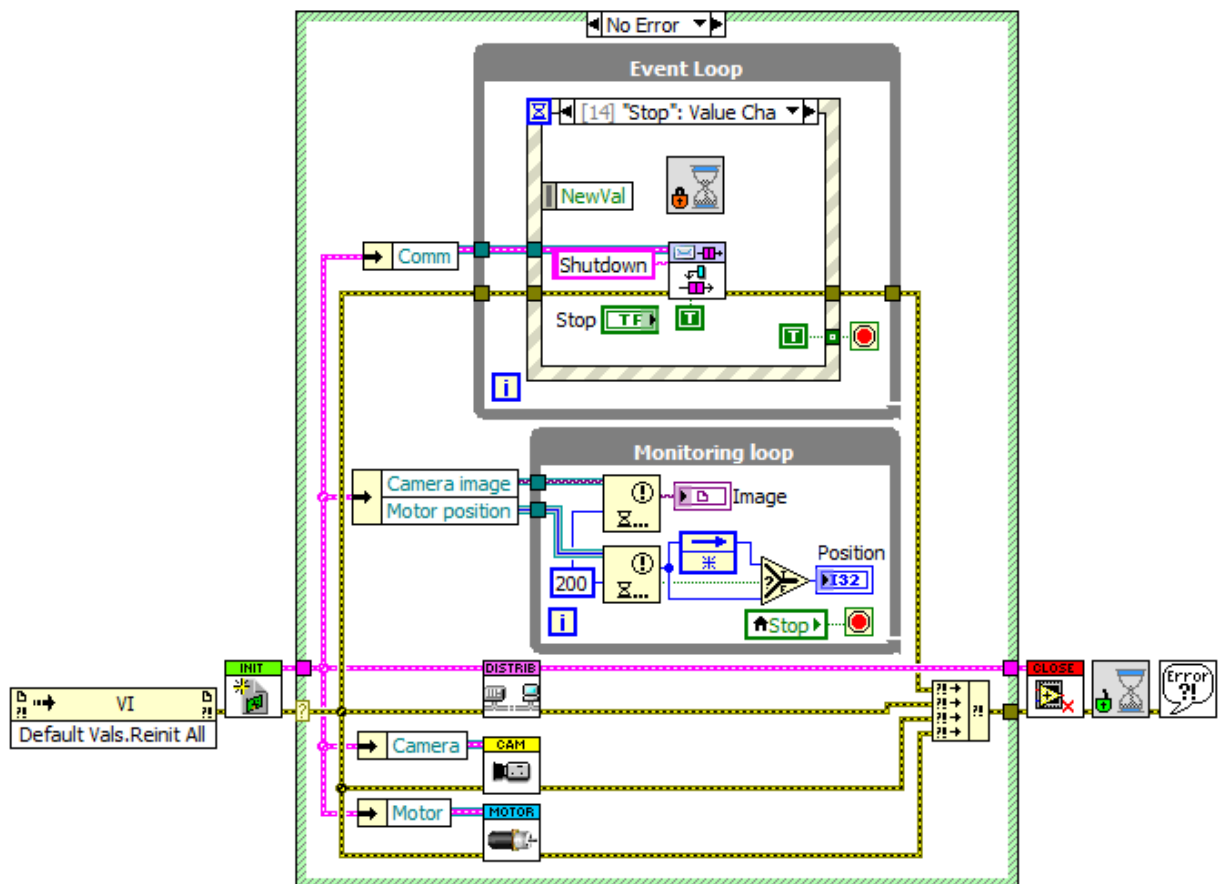
Řídicí systém byl naprogramován v programovacím prostředí LabVIEW 2013 verze 13.0.1f2. Kvůli nekompatibilitě softwaru LabVIEW s jeho staršími verzemi byl program uložen i pro starší verze LabVIEW 2012 a 2010, touto konverzí by měla být zachována funkcionálníta, změní se však zamýšlený design předního panelu. Pro správnou funkci programu je nezbytné mít nainstalovaný LabVIEW Vision modul.

Po spuštění vrchního programu Main.vi se zobrazí ovládací panel, viz obrázek 20. Přední panel se skládá ze záložek, displeje, na kterém se zobrazuje obraz z kamery, tlačítka pro nouzové zastavení motoru a ukazatele polohy motoru. Uživatel může přepínat mezi záložkami Initialization, Setup a Movement. V záložce Initialization je nutné nejdříve inicializovat procesy kamery a motoru. Uživatel vybere z nabídky jméno kamery, zadá cestu pro ukládání obrázků a stiskne tlačítko Initialize camera. Pro inicializaci motoru zadá jméno COM portu, na kterém je připojena řídicí jednotka EPOS, a stiskne tlačítko Initialize motor. V záložce Setup je možné měnit dobu expozice a zesílení citlivosti kamery. Před pohybem motoru se tlačítkem Enable odbrzdí permanentní brzda a tlačítkem Home I se zásobník přesune do referenční polohy. Před tímto pohybem je uživatel vyzván ke kontrole zásobníku. V záložce Movement je pak možné pohybovat s motorem relativním a absolutním pohybem nebo spustit automatický režim. Program se ukončí stisknutím křížku v pravém horním rohu.



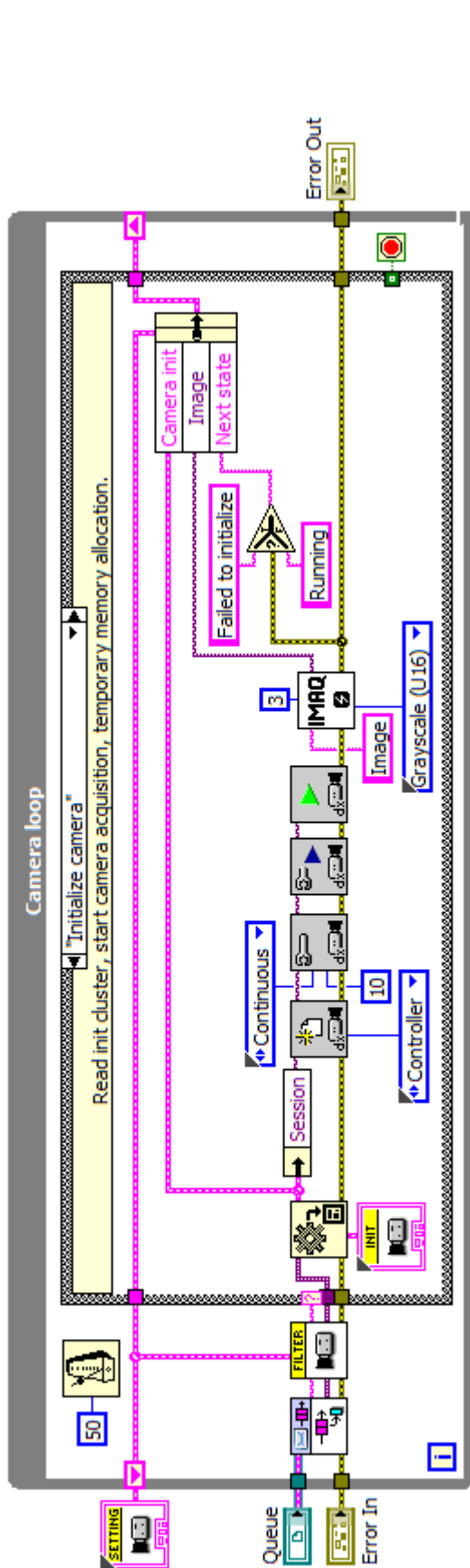
Obr. 20: Front panel programu Main.vi

Blokový diagram vrchního programu Main.vi (zdrojový kód 1) se skládá z 5 smyček dle komunikačního diagramu na obrázku 12 a z podprogramů Main_init.vi a Main_close.vi. Main_init.vi slouží k inicializaci QMH a funkcí Notifier, Main_close.vi slouží k ukončení těchto funkcí a dealokaci paměti. Funkce Notifier je použita k přenosu dat mezi procesními smyčkami a mezi smyčkou Monitoring loop.

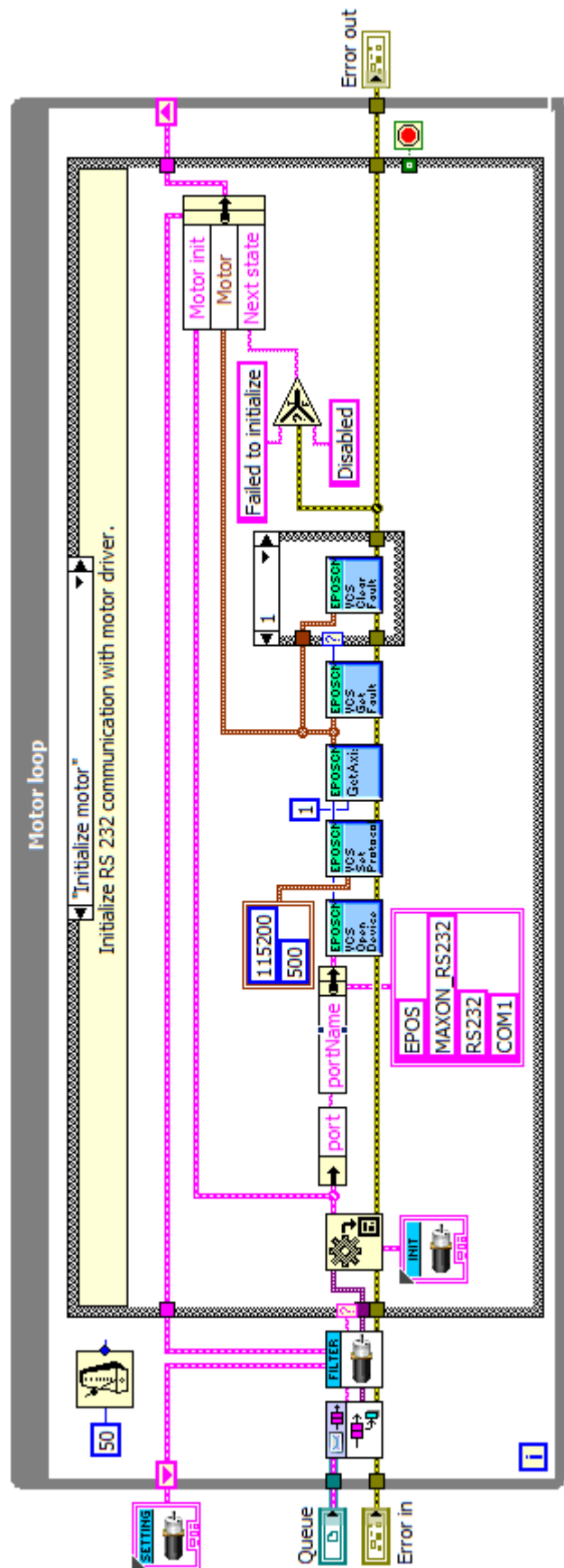


Zdrojový kód 1: hlavního programu Main.vi

Ukázka zdrojového kódu 2 a 3 podprogramů Camera_proces.vi a Motor_proces.vi zobrazuje zpracování stavových strojů popsaných v kapitolách 3.1.2 a 3.1.3 pomocí dříve vysvětlené šablony QMH a návrhového vzoru State machine.



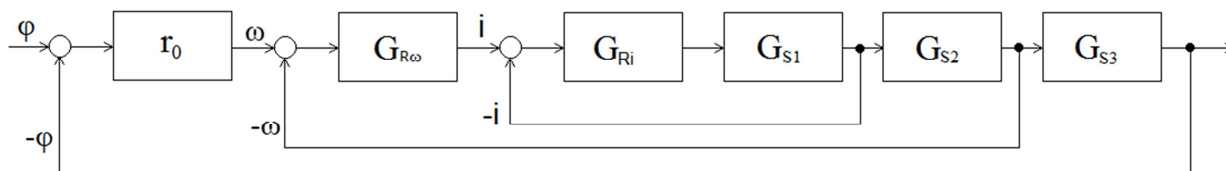
Zdrojový kód 2:
Podprogram Camera_proces.vi



Zdrojový kód 3:
Podprogram Motor_proces.vi

4 REGULACE POLOHOVÉHO SERVOMECHANISMU

Standardní struktura regulace polohového servomechanismu je tvořena polohovou smyčkou s podřízenou rychlostní smyčkou. Jako další podřízená smyčka rychlostní smyčky může být smyčka proudová nebo momentová. (obrázek 21, na kterém je uzavřená proudová smyčka znázorněna svým přenosem G_{Ri}).



Obr. 21: Struktura polohové regulace

Smyčka nejnižší úrovně – momentová (proudová) – slouží k ochraně motoru před přetížením a k omezení maximálního momentu, kterým může působit motor na mechanickou soustavu. Při použití tranzistorových výkonových měničů s pulsní šířkovou modulací s kmitočtem (jednotky až desítky kHz) je proudová smyčka velmi rychlá, s dobou odezvy řádově 0.1-1ms, což je dobrý předpoklad kvality nadřazené rychlostní smyčky. ¹

„Nastavení proudové smyčky závisí na parametrech motoru a je provedeno při výrobě pohonu. Obecně není vhodné toto nastavení měnit a pokud si nechceme přidělat starosti, nezasahujeme do tohoto nastavení.“ ²

Polohová regulační smyčka je nadřazena rychlostní smyčce: výstup polohového regulátoru je tedy žádanou hodnotou rychlosti. Charakteristickým znakem uzavřené polohové smyčky je její integrační charakter (poloha je integrálem rychlosti). Teoreticky je tedy polohová odchylka při nulové rychlosti rovněž nulová. Prakticky je citlivost pohonu na inkrementální odchylku polohy, danou rozlišovací schopností polohového snímače, závislá na citlivosti a rychlosti integrace otáčkového regulátoru. Parametrem, určujícím kvalitu polohové smyčky, je polohové zesílení (rychlostní konstanta) r_0 . ³

¹ SKALICKÝ, Jiří. Elektrické servopohony. Vyd. 1. Brno: PC-DIR Real, 1999, s. 15. Učební texty vysokých škol. ISBN 80-214-1484-7.

² Tamtéž s. 55

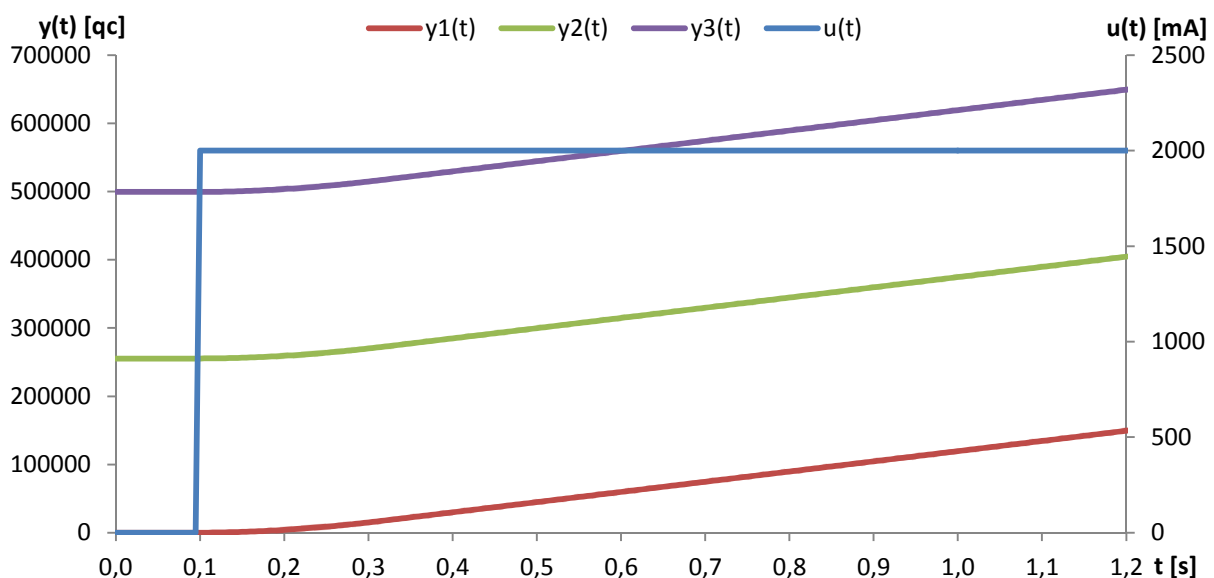
³ Tamtéž s. 15

4.1 Identifikace soustavy z přechodové charakteristiky

„Získání matematického modelu reálného objektu se nazývá identifikace. Identifikace může být analytická nebo experimentální. Praktické metody identifikace leží mezi těmito dvěma krajními případy. Vždy je užitečné teoretickou cestou najít přibližné matematické vztahy popisující daný objekt – regulovanou soustavu a pak experimentálně upřesnit parametry, které v nich vystupují. Čím lépe jsou připraveny analytické vztahy, tím kratší a méně nákladná jsou experimentální měření.“¹

„Měření přechodových charakteristik je velmi často používaným prostředkem ke zjišťování dynamických vlastností soustavy. Měřením se zjišťuje odezva $y(t)$ soustavy při změně vstupního signálu $u(t)$ skokem známé velikosti. Před provedením změny vstupního signálu musí být soustava v ustáleném stavu. Je účelné měření několikrát opakovat a pro vyhodnocení dynamických vlastností soustavy používat průměrných hodnot, nebo provést vyhodnocení všech zdařilých měření a pak určit výsledné hodnoty přechodové charakteristiky.“²

U soustavy FZT je výstupní veličina $y(t)$ poloha motoru a akční veličinou $u(t)$ je hodnota proudu vstupující do proudového regulátoru. Měření bylo provedeno při skoku proudu $\Delta u(t) = 2000\text{mA}$ a vzorkování 5ms. Hodnoty odezev polohy motoru $y_1(t)$ až $y_3(t)$ měřených po ustálení soustavy jsou znázorněny na grafu 1.



Graf 1: Odezva polohy motoru na skok akční veličiny 2000mA

¹ ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 79. ISBN 978-80-214-4398-3.

² OLEHLA, Miroslav, Slavomír NĚMEČEK a Ivan ŠVARC. *Automatické řízení*. Liberec: Technická univerzita, 2009, s. 35. ISBN 978-80-7372-484-9.

„Při výpočtu přechodové charakteristiky (odezvy na jednotkový skok) z několika naměřených odezev (na skok libovolné velikosti), je možno použít při obecně nestejně velkých skokových změnách vstupní veličiny vztahu

$$h(i) = \frac{1}{M} \sum_{k=1}^M \frac{y(i, k)}{\Delta u(k)} \quad 1$$

kde

$h(i)$ je hodnota přechodové charakteristiky v čase $t = i \cdot \Delta t$

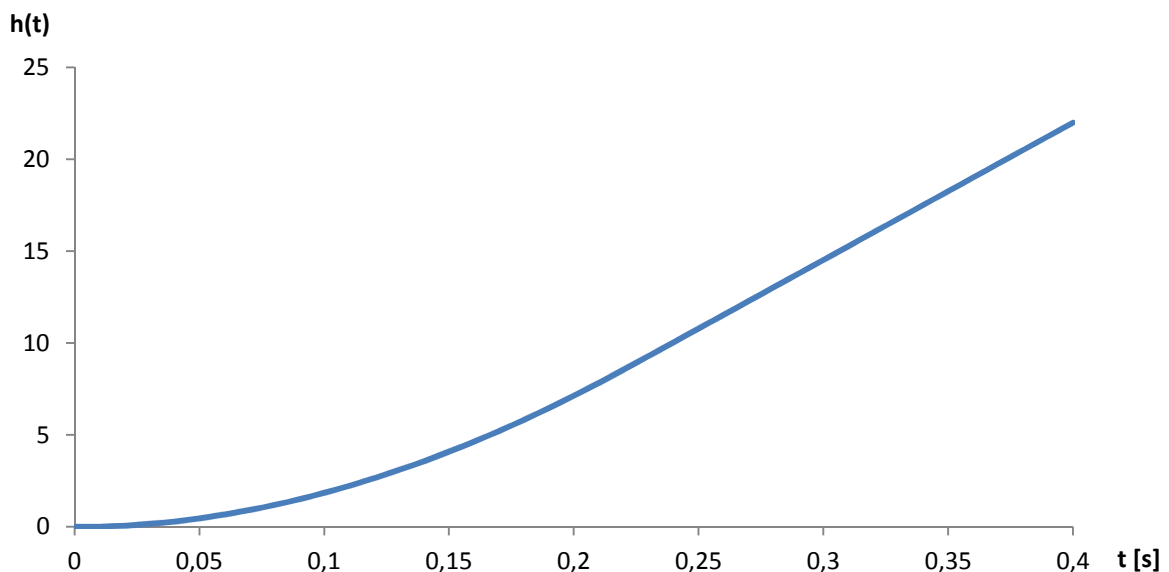
$\Delta u(k)$ skoková změna vstupního signálu při k -tém měření přechodové charakteristiky

$y(i, k)$ odezva výstupní veličiny soustavy v i -tém intervalu vzorkování při k -tém měření, $i = 0, 1, 2, \dots, N$

M počet skokových změn a odpovídajících odezev

N počet vzorků jednotlivých odezev na skokový signál“¹

Pro identifikaci se předpokládá, že „průběhy přechodových charakteristik jsou vhodně upraveny (filtrovány, vyhlazeny) a že se pracuje s přírůstkovými veličinami, tj. s průběhy začínajícími v počátku souřadnic.“² Výsledná přechodová charakteristika $h(t)$ je zobrazena v grafu 2.



Graf 2: Přechodová charakteristika $h(t)$

Z grafu přechodové charakteristiky $h(t)$ můžeme vyhodnotit, že soustava, se kterou pracujeme, je astatická (integrační). „Integrační regulovaná soustava je soustava, u níž po změně akční veličiny přibývá nebo ubývá regulované veličiny trvale. Změna probíhá trvale, pokud neuvažujeme omezení regulované veličiny dané konstrukcí některého jejího členu. Je

¹ OLEHLA, Miroslav, Slavomír NĚMEČEK a Ivan ŠVARC. *Automatické řízení*. Liberec: Technická univerzita, 2009, s. 35-36. ISBN 978-80-7372-484-9.

² ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 79. ISBN 978-80-214-4398-3.

tedy zřejmé, že po poruše rovnováhy u těchto soustav nenastane samovolné ustálení na nové hodnotě, jak tomu bylo u soustav proporcionálních, ale odchylka od původního rovnovážného stavu se neustále zvětšuje. Tyto soustavy nemají autoregulaci. Z toho vyplývá, že následky vzniklé poruchou lze odstranit pouze pomocí regulátoru.“¹

4.1.1 Tříparametrová identifikace integrační soustavy²

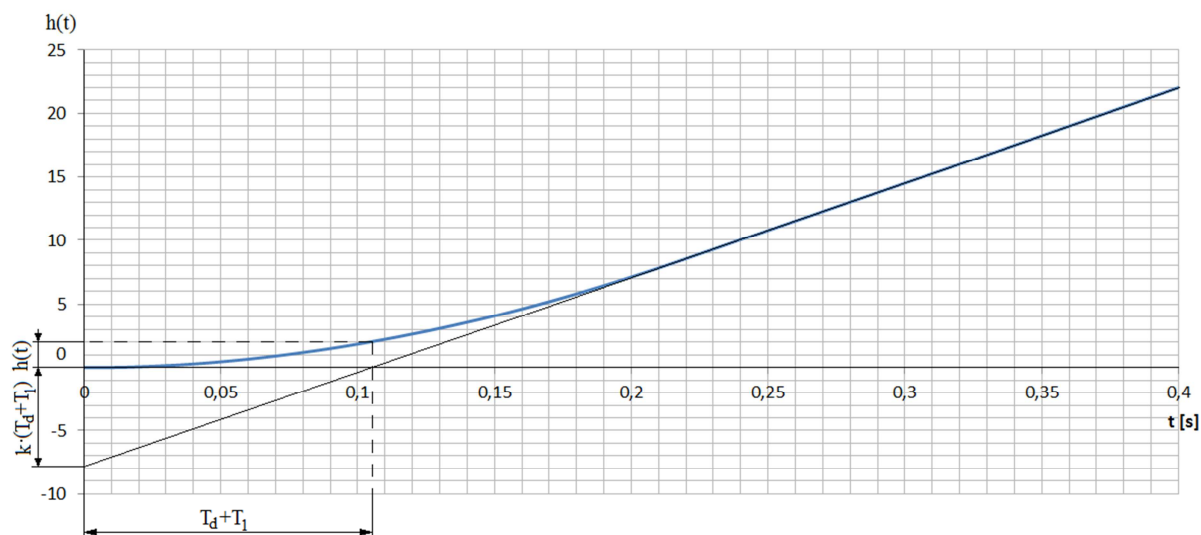
Model je dán přenosovou funkcí

$$G(s) = \frac{k}{s(1 + sT_1)} e^{-sT_d} \quad 2$$

a je charakterizován třemi parametry: rychlostním zesílením k , časovou konstantou T_1 a dopravním zpožděním T_d . Přejchodová funkce modelu je

$$h(t) = k \left(t - T_d - T_1 \left(1 - e^{-(t-T_d)/T_1} \right) \right) \quad 3$$

Zesílení k a průměrná doba odezvy $T_{ar} = T_d + T_1$ mohou být určeny graficky dle obrázku 22.



Obr. 22: Grafická identifikace tříparametrového modelu

$$T_{ar} = T_d + T_1 = 0,105s \quad k \cdot (T_d + T_1) = 7,9 \quad \rightarrow \quad k = \frac{7,9}{0,105} = 75,2qc \cdot mA^{-1}$$

Dopravní zpoždění T_d a časová konstanta T_1 mohou být určeny z rovnice 3 pro jeden bod přechodové funkce, vhodným bodem je $h(T_d + T_1) = kT_1 e^{-1}$.

$$h(T_d + T_1) = h(0,105) = 2qc \cdot mA^{-1} \quad T_1 = \frac{2}{75,2} e^1 = 0,072s$$

$$T_d = T_{ar} - T_1 = 0,105 - 0,072 = 0,033s$$

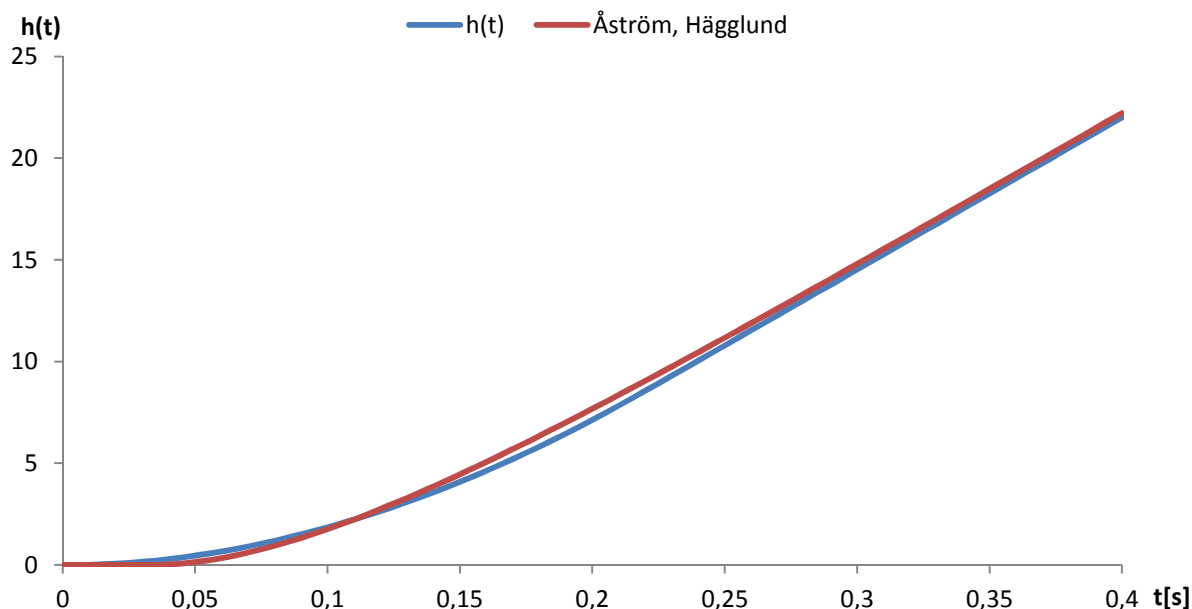
¹ OLEHLA, Miroslav, Slavomír NĚMEČEK a Ivan ŠVARC. Automatické řízení. Liberec: Technická univerzita, 2009, s. 58. ISBN 978-80-7372-484-9.

² ÅSTRÖM, Karl J, Tore HÄGGLUND. PID controllers. 2nd ed. Research Triangle Park, N.C.: International Society for Measurement and Control, c1995, s. 22-23. ISBN 1556175167.

Výsledný přenos určený touto metodou bude

$$G(s) = \frac{75,2}{s(1 + 0,072s)} e^{-0,033s} \quad 4$$

A porovnání naměřené přechodové charakteristiky a modelu je v grafu 3.



Graf 3: Porovnání modelové a naměřené přechodové charakteristiky

4.1.2 Identifikace nekmitavé integrační regulované soustavy pomocí dob $t_{0,33}$ a $t_{0,7}$ ¹

Pro identifikaci nekmitavých integračních regulovaných soustav (1.řádu) lze použít metody identifikace pro nekmitavé proporcionální regulované soustavy, pokud místo jejich přechodových charakteristik $h(t)$ se použije jejich derivace, tj. impulsní funkce

$$\frac{dh(t)}{dt} = g(t) \quad 5$$

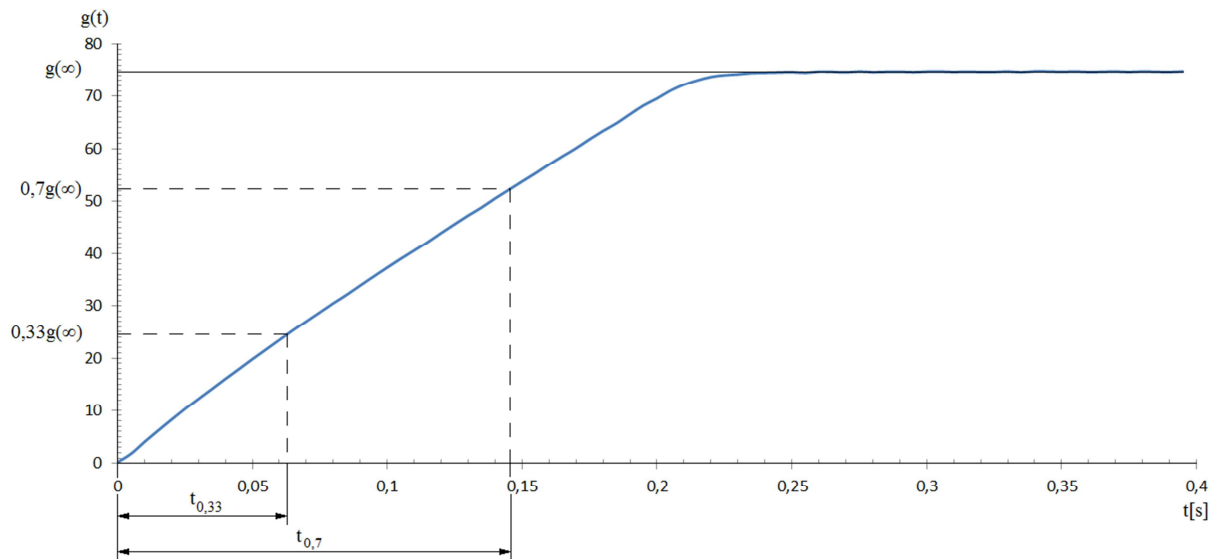
Použijí-li se pro experimentální identifikaci doby $t_{0,33}$ a $t_{0,7}$ (obr. 23), pak lze použít náhradní přenos

$$G(s) = \frac{k}{s(T_1s + 1)} e^{-T_d s} \quad 6$$

Koeficient přenosu k pro jednotkovou skokovou změnu akční veličiny, tj. $\Delta u(t) = \eta(t)$, je dán ustáleným stavem na impulsní charakteristice $k = g(\infty)$. Fyzikální rozměr koeficientu

¹ ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 82. ISBN 978-80-214-4398-3.

přenosu k je dán poměrem fyzikálního rozměru výstupní veličiny $y(t) = h(t)$ k fyzikálnímu rozměru akční veličiny $\Delta u(t)$ násobeného rozměrem času.



Obr. 23: Identifikace z impulsní charakteristiky $g(t)$

$$k = g(\infty) = 74,7 \text{ qC} \cdot \text{mA}^{-1} \text{ s}^{-1} \quad t_{0,33} = 0,063 \text{ s} \quad t_{0,7} = 0,145 \text{ s}$$

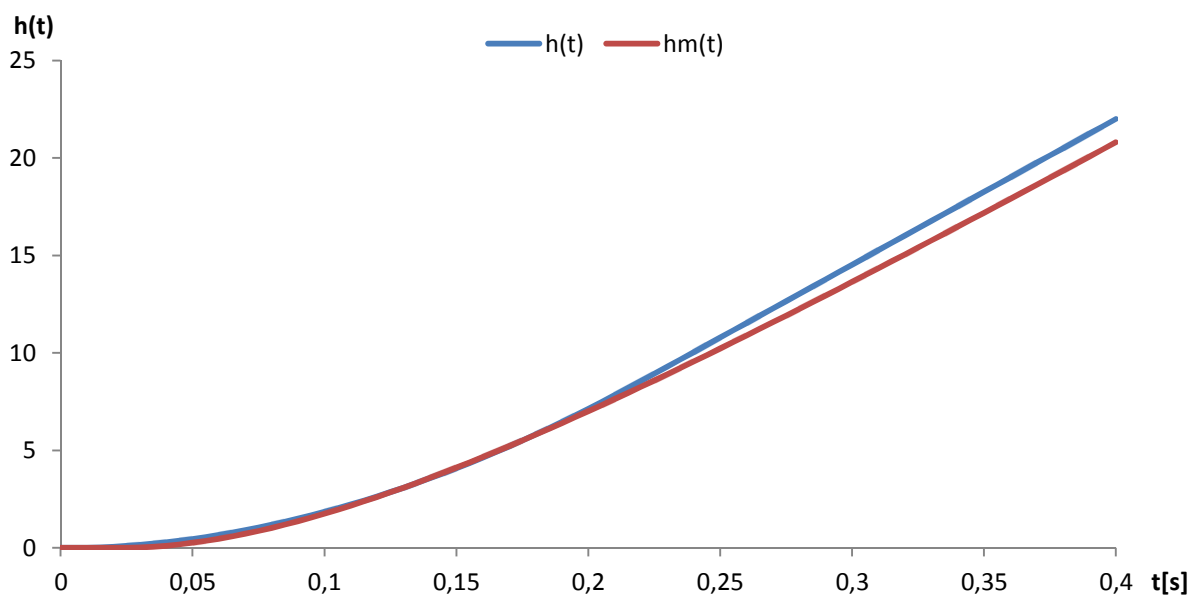
$$T_1 = 1,245(t_{0,7} - t_{0,33}) = 1,245(0,145 - 0,063) = 0,102 \text{ s}$$

$$T_d = 1,498t_{0,33} - 0,498t_{0,7} = 1,498 \cdot 0,063 - 0,498 \cdot 0,145 = 0,022 \text{ s}$$

Výsledný náhradní přenos určený touto metodou je

$$G(s) = \frac{74,7}{s(0,102s + 1)} e^{-0,022s} \quad 7$$

a porovnání naměřené přechodové charakteristiky $h(t)$ a modelu $hm(t)$ je v grafu 4.



Graf 4: Porovnání naměřené a modelové přechodové charakteristiky

4.1.3 Identifikace minimální kvadratickou plochou

Tato metoda je založena na minimalizaci kvadratické plochy mezi naměřenou přechodovou charakteristikou a modelovou charakteristikou

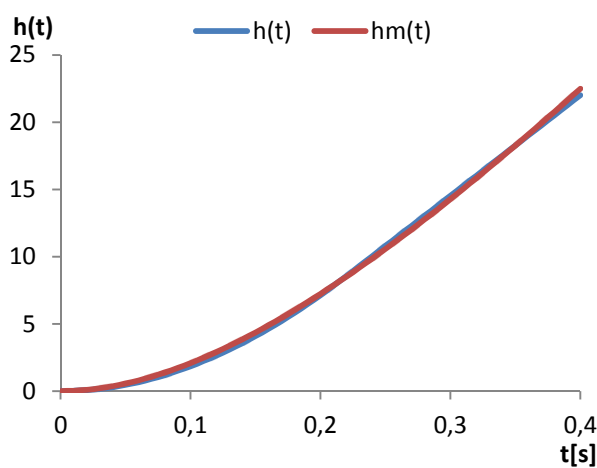
$$J = \int_0^{\tau} (y - y_m)^2 dt \quad 8$$

K výpočtu modelového přenosu a hodnot přechodové charakteristiky bylo použito programovací prostředí MATLAB viz zdrojový kód 4 programu *Identifikace.m* a zdrojový kód volané funkce *Integral.m* pro integrační člen 1. řádu se setrvačností 1. řádu.

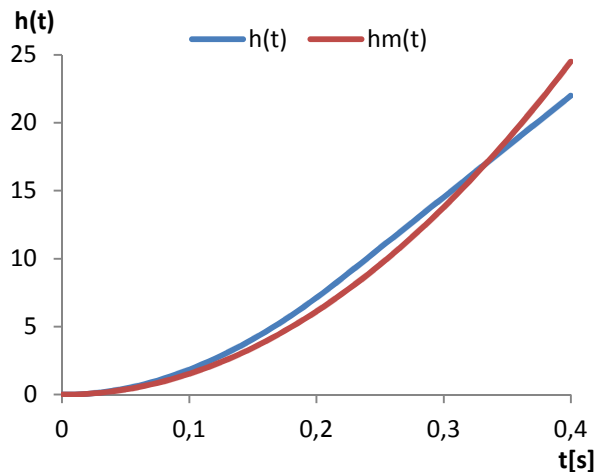
Byly testovány modely s integračním členem a setrvačností různých řádů, nejnižší kvadratickou plochu dosáhl model s integračním členem 1. řádu se setrvačností 2. řádu

$$G(s) = \frac{k}{s \cdot (T_1s + 1) \cdot (T_2s + 1)} = \frac{78,57}{s \cdot (0,07s + 1) \cdot (0,05s + 1)} \quad 9$$

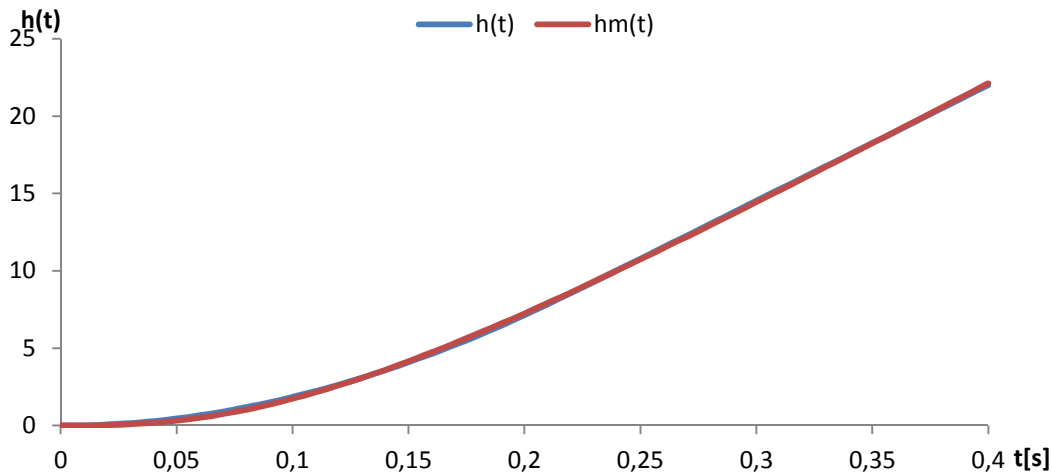
s odchylkou $J = 0,0033$, srovnání charakteristik $h(t)$ a $hm(t)$ jsou v grafu 5 až 7.



Graf 5: Integrační člen 1. řádu se setrvačností 1. řádu



Graf 6: Integrační člen 2. řádu se setrvačností 1. řádu



Graf 7: Integrační člen 1. řádu se setrvačností 2. řádu

```

%Identifikace soustavy z prechodove charakteristiky
%vymazani command window
clc
%import casu t a hodnot namerene prechodove charakteristiky h(t) z excelu
t = xlsread('Position_2000mA.xlsx','Sheet 1','A2:A82');
ht = xlsread('Position_2000mA.xlsx','Sheet 1','F2:F82');
%pouziti algoritmu fminsearch pro nalezeni minima kvadraticke plochy
T = fminsearch(@(T) Integral(T,ht,t),[1,1])
%prenos s dosazenými hodnotami z algoritmu fminsearch
Gs = tf([T(1,1)], [T(1,2) 1 0]);
%modelova prechodova charakteristika prenosu Gs
[a,b]=step(Gs,t);
%ulozeni modelove prechodove charakteristiky do excelu
filename='Position_2000mA.xlsx';
hmt=[a];
sheet='Sheet 2';
xlswrite(filename,hmt,sheet)
%otevreni noveho okna
figure
%vytvoreni grafu z vypoctenych hodnot
plot(b,hmt,'r')
hold on
%vytvoreni grafu z namerenych hodnot
plot(b,ht)
legend('model','namereno')
hold off
%Hodnota kvadraticke odchylky
I=Integral(T,ht,t)

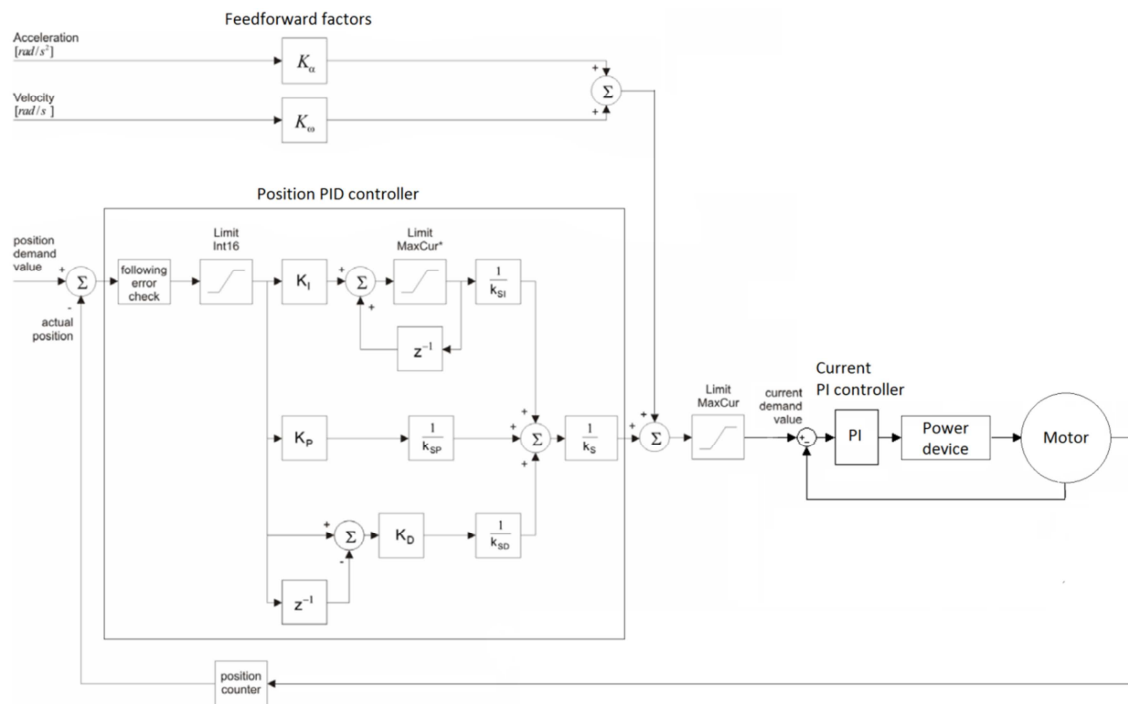
%funkce Intergral
function I = Integral(T,ht,t)
%kontrola vstupnich parametru
N=length(ht);
if N~=length(t)
    error('Vstupni vektory maji ruznou delku')
end
if N<2
    error('Pro vypocet jsou potreba alespon dva body')
end
%Prenos s dosazenými hodnotami T
Gs = tf([T(1)], [T(2) 1 0]);
%modelova prechodova charakteristika prenosu Gs
[a]=step(Gs,t);
hmt=[a];
%numericka integrace
I=0;
for i=1:N-1
    I=I+(t(i+1)-t(i))*((ht(i,1)-hmt(i,1))^2);
end

```

Zdrojový kód 4: Program Identifikace.m a funkce Integral.m programovacího prostředí MATLAB

4.2 Seřízení regulátoru

Pro regulaci pozice je v jednotce EPOS použit diskretní PID regulátor (PSD) využívající digitální signálový procesor se vzorkovacím časem $T = 1ms$. Blokové schéma regulačního obvodu je zobrazeno na obrázku 24¹. Dopředné zesílení rychlosti a zrychlení K_ω a K_α je zapojeno jen při profilovém polohovém módu, které nebude v řídicím systému použito.



Obr. 24: Schéma regulačního obvodu

4.2.1 Číslicové regulátory

Od číslicového regulátoru budeme očekávat stejnou funkci jako od spojitého regulátoru, to znamená zesilovat, integrovat a derivovat vstupující regulační odchylku. Proto při sestavování algoritmu pro číslicový regulátor vyjdeme z funkce a tím i rovnice spojitého PID regulátoru

$$u(t) = r_0 \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right] \quad 10$$

Číslicovou verzi regulátoru získáme z této rovnice diskretizací integrace a derivace. Integraci provedeme náhradou spojitého signálu tzv. stupňovitou náhradou zleva (obdélníky zleva). Určení hodnoty integrálu se provádí jako součet ploch pod náhradním průběhem

$$\int_0^{kT} e(t) dt \cong T \sum_{i=1}^k e(i) \quad 11$$

¹ MAXON MOTOR. *Positioning controller: Position regulation with Feed Forward*. Sachseln, 2008.

Derivaci získáme nahrazením diferencemi (podělenými T)

$$\frac{de}{dt} \cong \frac{e(k) - e(k-1)}{T} \quad 12$$

Po dosazení těchto vztahů do rovnice spojitého PID regulátoru (10), kam současně dosadíme diskrétní čas kT respektive k , dostaneme

$$u(k) = r_0 \left\{ e(k) + \frac{T}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad 13$$

Tomuto algoritmu číslicového regulátoru se říká polohový algoritmus. Hodnota integrálu se zde získává sumací a hodnota derivace se získává pomocí zpětné diference. Proto se tyto regulátory nazývají proporcionálně-sumačně-diferenční a označují se zkratkou PSD.¹

Pro výpočet stavitelných parametrů regulátoru bude použit model s integračním členem 1. řádu se setrvačností 2. řádu určený metodou identifikace minimální kvadratickou plochou s nejnižší kvadratickou odchylkou $J = 0,0033$ ze všech použitých metod identifikace

$$G(s) = \frac{78,57}{s \cdot (0,07s + 1) \cdot (0,05s + 1)}$$

4.2.2 Metoda optimálního modulu²

Metoda optimálního modulu se používá především při regulaci elektrických pohonů, kde se malé časové konstanty (elektrické) zastupují náhradní součtovou časovou konstantou. Často se v praxi používá zjednodušená verze metody optimálního modulu. Ta spočívá v tom, že se integrační nebo derivační konstanta regulátoru (nebo obě u PID regulátoru) zvolí rovny časové konstantě (časovým konstantám) soustavy. V přenosu rozpojeného obvodu $G_0(s)$ dojde pak k vykrácení závorek v čitateli a jmenovateli. Přenos regulované soustavy se upraví na vhodný tvar podle tabulky 4 a pro doporučený regulátor se vypočtou hodnoty jeho stavitelných parametrů. V tabulce jsou uvedeny jen ty kombinace regulátor – regulovaná soustava, které dávají zaručeně stabilní regulační obvod a není třeba proto kontrolovat jeho stabilitu. Pro zvolený model je výpočet parametrů následující:

Tab. 4: Metoda optimálního modulu pro PD regulátor

Regulovaná soustava	Typ	r_0	r_{-1}	r_1
$\frac{k}{s \cdot (T_1s + 1) \cdot (T_2s + 1)}$ $T_1 \geq T_2$	PD	$\frac{1}{2k(T_2 + 0,5T)}$	-	$\frac{r_0 \cdot (T_1 - 0,5T)}{T}$

¹ ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 216-217. ISBN 978-80-214-4398-3.

²Tamtéž, s. 228-230.

$$r_0 = \frac{1}{2 \cdot 78,57(0,05 + 0,5 \cdot 0,001)} = 0,126 \quad r_1 = \frac{0,126 \cdot (0,07 - 0,5 \cdot 0,001)}{0,001} = 8,757$$

Podle schématu na obrázku 24 musí být stavitelné parametry upraveny kvůli konstantám k_S , k_{SP} a k_{SD} a pro zadání do jednotky EPOS musí být zaokrouhleny na celá čísla

$$K_P = r_0 \cdot k_{SP} \cdot k_S = 0,126 \cdot 4 \cdot 2 = 1 \quad K_D = r_1 \cdot k_{SD} \cdot k_S = 8,757 \cdot 1 \cdot 2 \cong 18$$

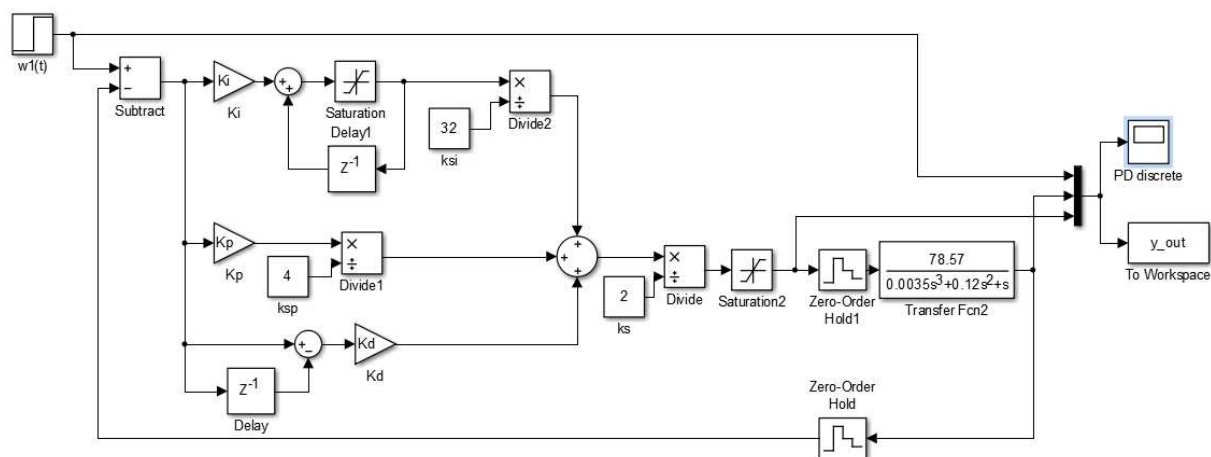
4.2.3 Metoda kvadratické regulační plochy¹

Kritérium minima kvadratické plochy je dáno vztahem

$$J = \int_0^{\infty} [y(t) - y(\infty)]^2 dt = \int_0^{\infty} e^2 dt \quad 14$$

Kvadratická plocha dává větší váhu větším odchylkám a menší ve svém hodnocení potlačuje (kvadratická závislost). Kritérium minima kvadratické regulační plochy je z teoretických kritérií nejrozšířenější.

Pro výpočet stavitelných parametrů regulátoru bylo použito programovací prostředí MATLAB viz následující zdrojový kód 6 programu *Parametry_regulatoru.m* a volané funkce *Regulator.m*. Funkce *fminsearch* mění parametry regulátoru jako globální proměnné a je volán program vytvořený v prostředí Simulink (zdrojový kód 5) na základě blokového schématu dle obrázku 24. Výstupní matice *y_out* předává hodnoty požadované veličiny $w(t)$ a odezvy na skok $y(t)$, z kterých je vypočítána kvadratická plocha. Počet iterací je snížen kvůli omezení maximálních hodnot zesílení regulátoru na 15 iterací.



Zdrojový kód 5: Program PSD_regulator.slx v simulačním prostředí Simulink

¹ ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 132. ISBN 978-80-214-4398-3.

```

%Parametry_regulatoru kritériem minima kvadratické plochy
%vymazani command window
clc
global Kp;
global Kd;
global Ki;
global y_out;
%delka simulace
t_end=1;
%krok
h=0.001;
%pouziti algoritmu fminsearch pro nalezeni minima kvadraticke plochy
T = fminsearch(@(T) Regulator(T,t_end,h),[1 17.5],optimset('MaxIter',15))

%funkce Regulator
function f = Regulator(T,t_end,h)
global Kp;
global Kd;
global Ki;
global y_out;
%zmena globalnich promennych
Kp=T(1);
Kd=T(2);
Ki=0;
tt=(0:h:t_end);
%volani programu simulink
sim('psd_regulator',tt);
f=0;
N=length(tt);
%numericka integrace
for i=1:N-1
f=f+(tt(i+1)-tt(i))*((y_out(i,1)-y_out(i,2))^2);
end

```

Zdrojový kód 6: Program Parametry_regulatoru.m a funkce Regulator.m

Výsledek této metody jsou stavitelné parametry regulátoru

$$K_P = 4.6066 \cong 5 \quad K_D = 63,1914 \cong 63$$

4.2.4 Metoda inverze dynamiky¹

Tato metoda seřizování regulátorů vychází z obecných principů inverze dynamiky a redukuje se na nalezení takového regulátoru s přenosem $G_R(z)$, který zajistí na základě vztahu

$$G_R = \frac{1}{G_S} \frac{G_w}{1 - G_w} \quad 15$$

pro regulovanou soustavu s přenosem G_S požadovaný model uzavřeného regulačního obvodu, tj. požadovaný přenos řízení G_w . V případě, že regulovaná soustava neobsahuje dopravní zpoždění ($T_d = 0$), je uvažován požadovaný přenos řízení pro diskrétní regulační obvody ve tvaru

¹ VÍTEČKOVÁ, M. *Seřizování regulátorů metodou inverze dynamiky*. VŠB-TU Ostrava, Fakulta strojní, 2000.

$$G_w(z) = \frac{1 - e^{-\frac{T}{T_w}}}{z - e^{-\frac{T}{T_w}}} \quad 16$$

V tomto případě regulační proces je vždy aperiodický a časovou konstantu uzavřeného regulačního obvodu T_w je třeba volit s ohledem na omezení akční veličiny a maximální nastavitelnou hodnotu zesílení regulátoru $r_{0\max}$. V případě jiných tvarů přenosů regulovaných soustav je nutné je upravit, viz *Úprava přenosů regulovaných soustav*¹, na základní tvar uvedený v tabulce 5.

Tab. 5: Metoda inverze dynamiky pro PD regulátor

Regulovaná soustava	Typ	r_0 (pro $T_d = 0$)	r_{-1}	r_1
$\frac{k_1}{s \cdot (T_1 s + 1)} e^{-T_d s}$	PD	$\frac{2}{k_1(2T_w + T)}$	-	$\frac{r_0 \cdot (T_1 - 0,5T)}{T}$

$$\frac{k}{s \prod_{i=1}^n (T_i s + 1)} \approx \frac{k}{s(T_\Sigma + 1)} \quad T_\Sigma = \sum_{i=1}^n T_i$$

$$\frac{78,57}{s \cdot (0,07s + 1) \cdot (0,05s + 1)} \approx \frac{78,57}{s \cdot (0,12s + 1)}$$

Výpočet parametrů regulátoru pro zvolenou časovou konstantu $T_w = 0,025$ a jejich úprava kvůli konstantám k_S , k_{SP} a k_{SD} je tedy následující

$$r_0 = \frac{2}{78,57 \cdot (2 \cdot 0,025 + 0,001)} = 0,5 \quad r_{-1} = \frac{0,5 \cdot (0,12 - 0,5 \cdot 0,001)}{0,001} = 59,75$$

$$K_P = r_0 \cdot k_{SP} \cdot k_S = 0,5 \cdot 4 \cdot 2 = 4 \quad K_D = r_1 \cdot k_{SD} \cdot k_S = 59,75 \cdot 1 \cdot 2 \cong 120$$

¹ ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, s. 86-87. ISBN 978-80-214-4398-3.

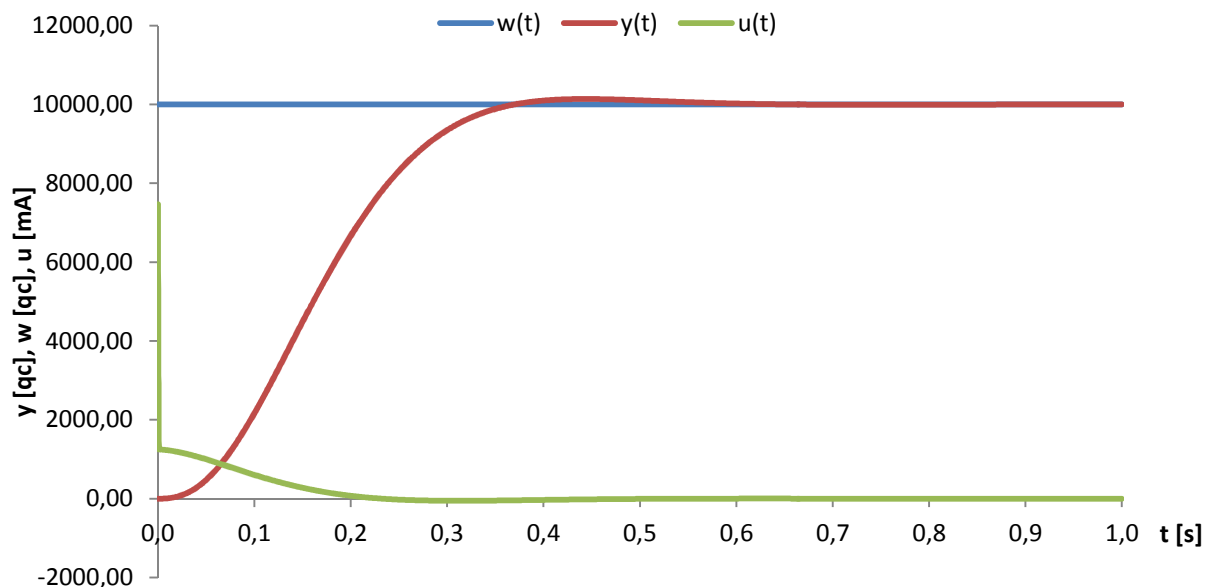
4.3 Vyhodnocení regulačního pochodu

Ověření chování regulované soustavy s vypočtenými stavitelnými parametry regulátoru bude provedeno simulací odezvy polohy motoru $y(t)$ na skok požadované polohy $w(t)$ známé velikosti a měřením odezvy polohy motoru reálné soustavy. Pro správnou funkci FZT je důležitá rychlost dosažení požadované polohy s minimální trvalou regulační odchylkou a bez překmitávání požadované polohy (to je dáno konstrukcí podavače, kdy ve chvíli dosažení pracovní polohy terče dochází zároveň k odebrání terče technologického tělesa z rotorového tělesa zásobníku terčů, viz kapitola 1).

4.3.1 Ověření simulací soustavy

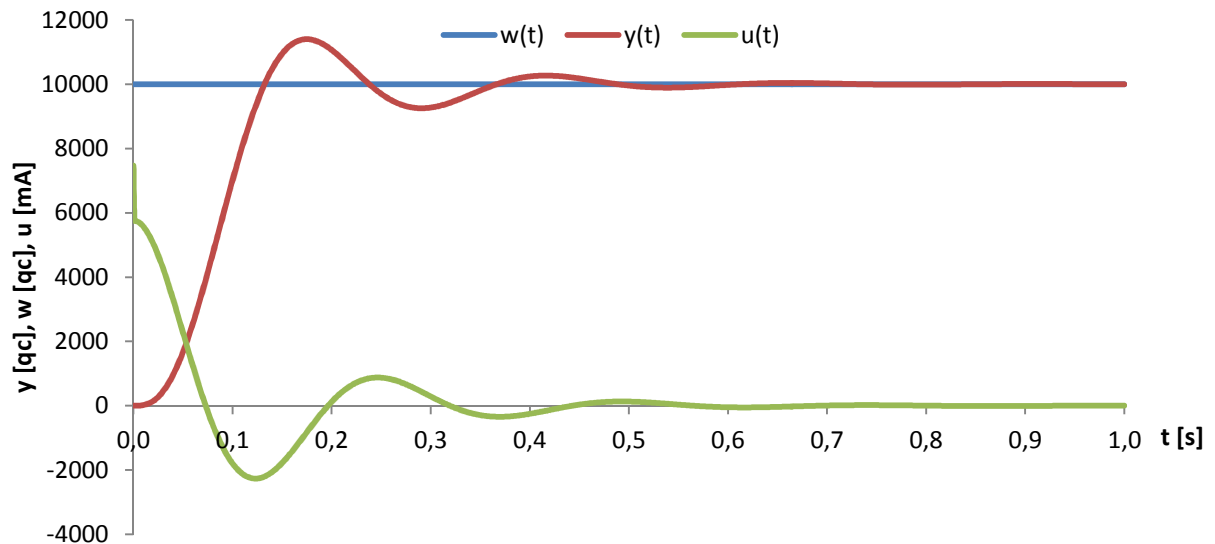
Simulace je provedena v simulačním prostředí Simulink (zdrojový kód 5) skokem požadované polohy o velikosti $\Delta w(t) = 10000qc$.

Na grafu 8 je simulace odezvy polohy motoru při nastavení parametrů $K_P = 1, K_D = 18$ vypočtených metodou optimálního modulu. Ustálené polohy dosáhne soustava v čase 0,71s s mírným překmitem požadované polohy.



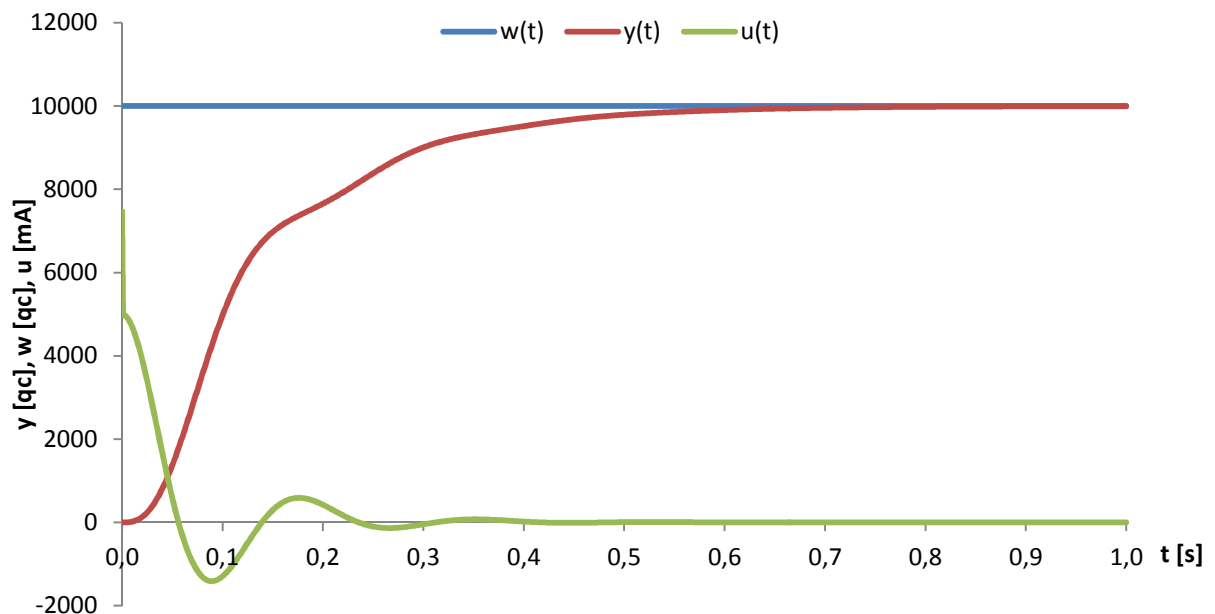
Graf 8: Simulace polohy motoru při použití parametrů z metody optimálního modulu

Simulace odezvy polohy motoru při použití metody minima kvadratické plochy ($K_D = 5$, $K_P = 63$) je zobrazena na grafu 9. Vyšší proporcionální konstanta regulátoru zrychluje odezvu systému, ale způsobuje větší kmitání kolem požadované polohy. Ustálené polohy dosáhne soustava v čase 0,97s.



Graf 9: Simulace polohy při použití parametrů z metody kvadratické regulační plochy

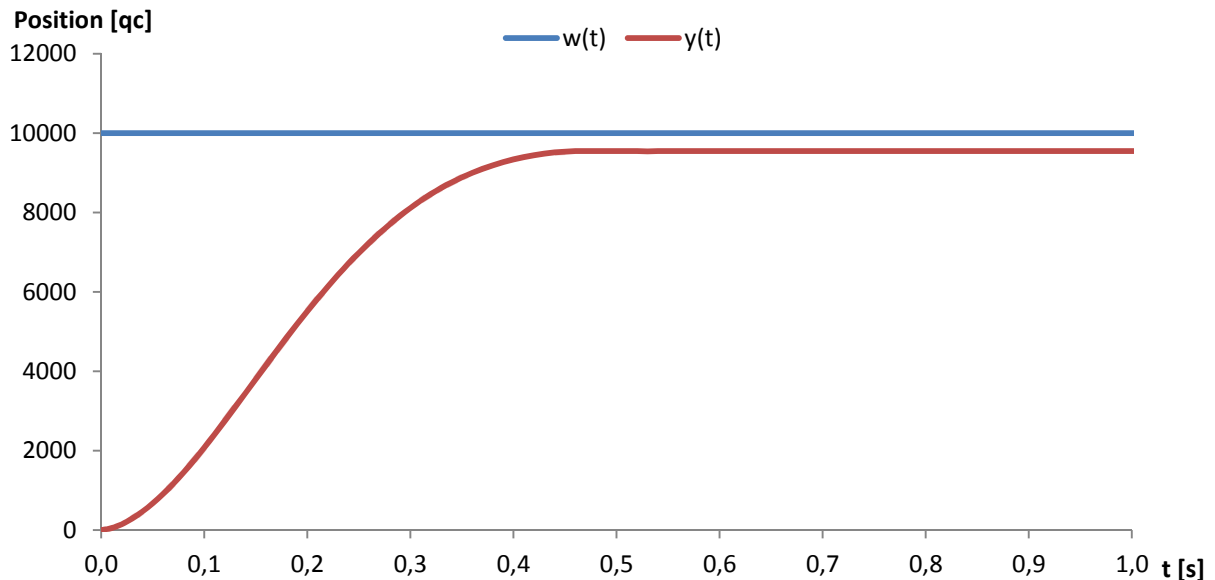
Se stavitelnými parametry regulátoru z metody inverze dynamiky ($K_P = 4$, $K_D = 120$) dostaneme simulaci odezvy dle grafu 10. Soustava dosáhne ustálené polohy bez překmitu v čase 0,9s. Pomalejší odezva systému a neplynulost pohybu jsou způsobeny vysokou hodnotou derivační konstanty regulátoru.



Graf 10: Simulace polohy při použití parametrů z metody inverze dynamiky

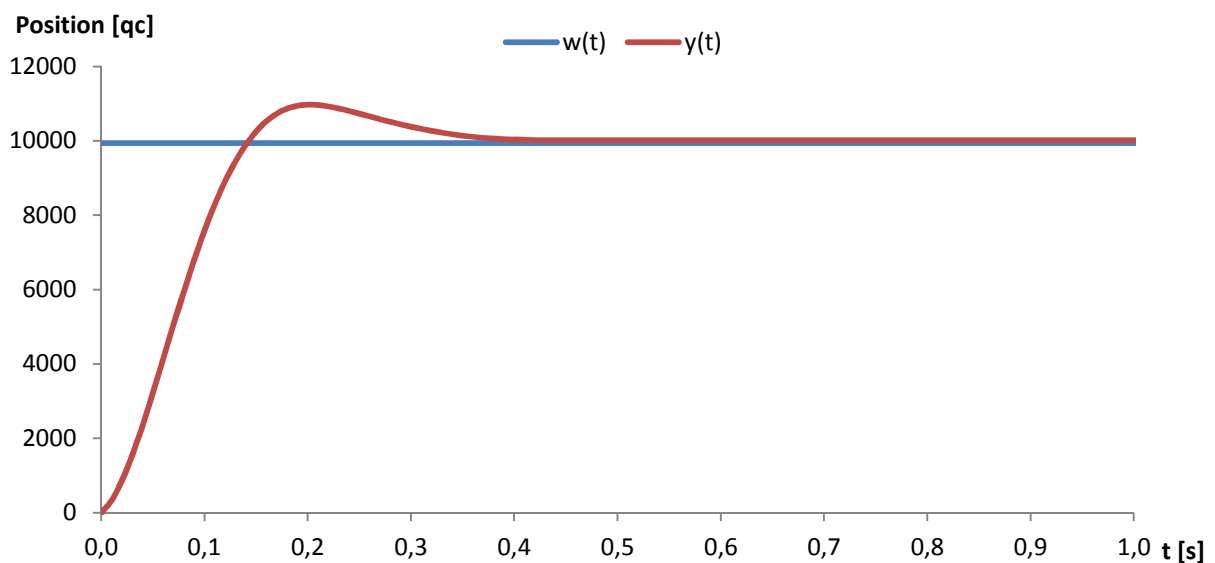
4.3.2 Ověření měřením odezvy

Použitím stavitelných parametrů regulátoru vypočtených metodou optimálního modulu ($K_P = 1, K_D = 18$) dosáhne reálná soustava ustálené polohy v čase 0,53s bez překmitu s trvalou regulační odchylkou 0,024rad. Průběh odezvy polohy motoru je zobrazen v grafu 11.



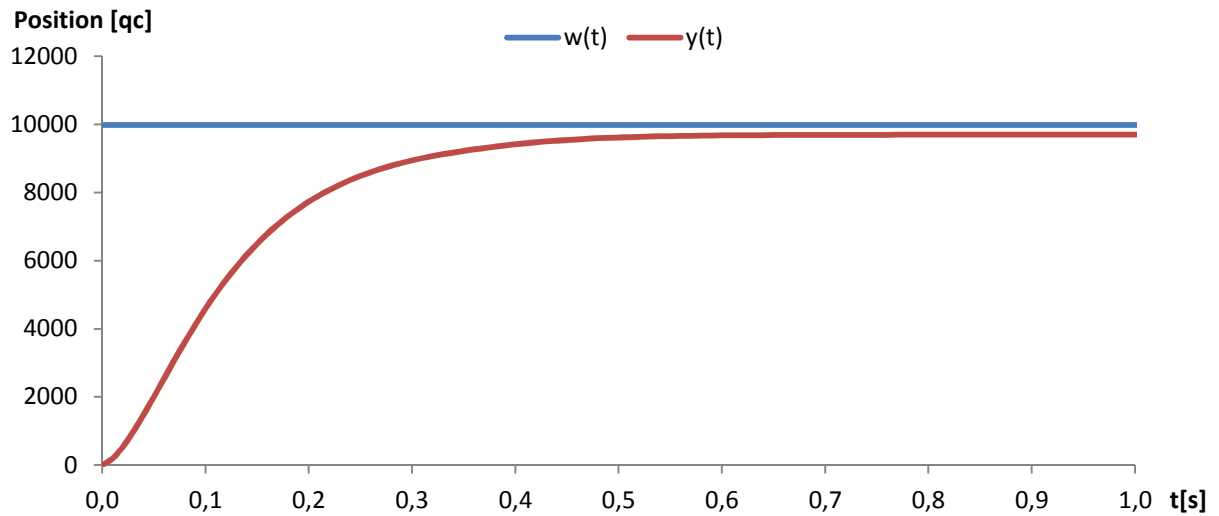
Graf 11: Odezva polohy motoru při použití parametrů z metody optimálního modulu

Se stavitelnými parametry regulátoru ($K_P = 5, K_D = 63$) z metody minima kvadratické regulační plochy dojde k překmitu a návratu regulované soustavy na ustálenou hodnotu v čase 0,45s s trvalou regulační odchylkou $4,2 \cdot 10^{-3}$ rad. Odezva je zobrazena v grafu 12.



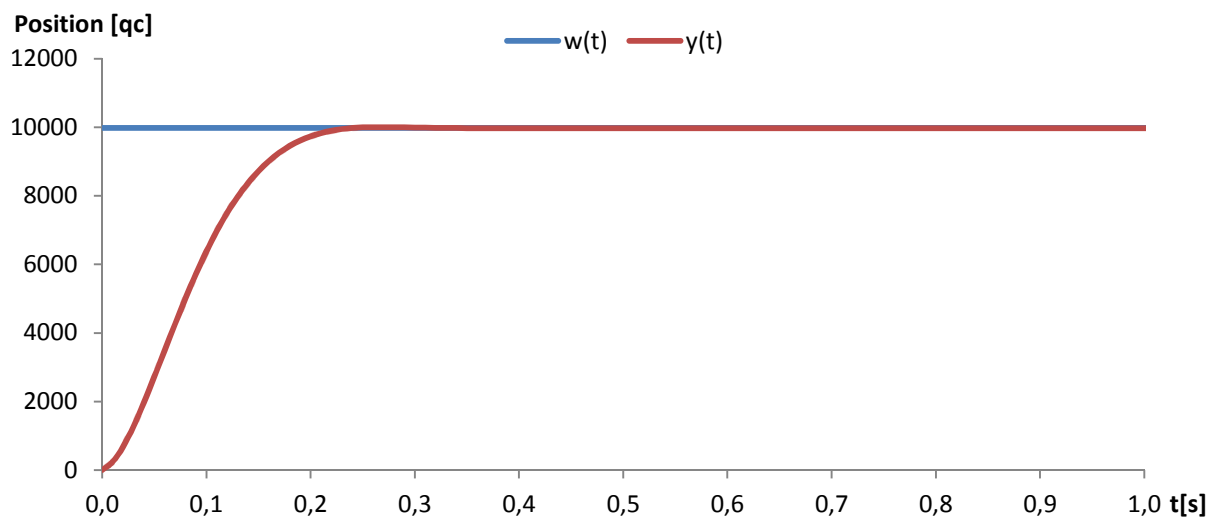
Graf 12: Odezva polohy při použití parametrů z metody kvadratické regulační plochy

S použitím stavitelných parametrů z metody inverze dynamiky ($K_P = 4, K_D = 120$) dosáhne soustava ustálené polohy bez překmitu v čase 0,75s s trvalou regulační odchylkou 0,015rad. Průběh odezvy polohy motoru na skok je zobrazen v grafu 13.



Graf 13: Odezva polohy při použití parametrů z metody inverze dynamiky

U odezvy první a třetí metody je nevyhovující trvalá regulační odchylka způsobená pasivními odpory. Druhá metoda se nejvíce přibližuje požadovanému průběhu svou rychlostí i nízkou regulační odchylkou. Nevyhovující je překmit, který lze jednoduše opravit navýšením derivační složky. Vyjdeme-li z podobnosti parametrů 2. a 3. metody a nastavíme-li zesílení $K_P = 5$ a derivační složku na hodnotu $K_D = 90$, dostaneme ideální průběh dle grafu 14. Ustálené polohy dosáhne soustava v čase 0,35s s regulační odchylkou $5,2 \cdot 10^{-5}$, což je na hranici rozlišitelnosti encoderu.



Graf 14: Odezva polohy při nastavení parametrů $K_P=5, K_D=90$

ZÁVĚR

Dle zadání byl navržen a zrealizován řídicí systém Frekvenčního zásobníku terčů. Součástí návrhu řídicího systému byla specifikace a verifikace řídicího systému, rešerše dostupných řídicích modulů pro řízení v reálném čase společnosti National Instruments a návrh komunikačního diagramu.

Řídicí program byl vytvořen v programovacím a vývojovém prostředí LabVIEW. Byl sestaven z nezávislých smyček, které mezi sebou komunikují pomocí návrhového vzoru Queued message handler. Procesní smyčky pro ovládání pohybu zásobníku a zpracování obrazu kamery byly vystavěny na principu stavového automatu a jejich struktura byla popsána pomocí stavových diagramů.

Pro pohon dopravníku byl použit stejnosměrný motor Maxon EC45 a řídicí jednotka Maxon EPOS 70/10 s implementovaným PSD regulátorem. Pro jeho správnou funkci bylo nutné nastavit stavitelné parametry regulátoru. Byla naměřena odezva soustavy na skok řídicího proudu a z přechodové charakteristiky bylo určeno, že se jedná o soustavu astatickou. Pro získání obrazového přenosu soustavy bylo použito několik metod identifikace, nejpřesnější metodou byla metoda minima kvadratické plochy, z níž byla soustava určena jako astatická soustava 2. řádu. Poté byly různými metodami vypočteny stavitelné parametry regulátoru a pro ověření chování regulované soustavy byla použita simulace a měření odezvy polohy motoru na skok žádané hodnoty. Ke kompenzaci pasivních odporů, které v soustavě vznikají, by bylo vhodné použít dopřednou vazbu, což daný regulátor neumožňuje. Pro správný průběh regulačního pochodu byla upravena derivační složka použitého PD regulátoru tak, aby byla minimalizována trvalá regulační odchylka.

Nedílnou součástí projektu Frekvenčního zásobníku terčů při jeho vývoji mělo být kromě testování konstrukce a mechaniky i elektronické řízení pracovního cyklu. Tato práce se snažila přispět k tomuto bodu vytvořením řídicího programu dle aktuální koncepce řídicího systému ELI Beamlines a může být využita při vývoji finální varianty Frekvenčního zásobníku terčů v provedení do vakua.

POUŽITÁ LITERATURA

- [1] POLAN, J., HAVLICEK, T., RUS, B. Target delivery system for high repetition rate lasers. In: *Optical Engineering+ Applications*. International Society for Optics and Photonics, 2007. p. 670210-670210-6.
- [2] STRNADEL, Josef. Návrh časově kritických systémů I: specifikace a verifikace. *Automa*. roč. 2010, č. 10.
- [3] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. 1. vyd. Praha: BEN - technická literatura, 2008, 247 s. ISBN 9788073002459.
- [4] ŠTEFAN, Radim. Minulost, současnost a budoucnost standardu PXI. *Automa*. roč. 2006, č. 3.
- [5] ŠTEFAN, Radim. Kompaktní systém pro zpracování obrazu. *Automa*. roč. 2006, č. 5.
- [6] DANĚK, Martin. Programovatelná hradlová pole - FPGA. *Automa*. roč. 2006, č. 2.
- [7] NIDays 2014. *Automa*. roč. 2015, č. 1.
- [8] Představení výkonného vícejádrového řídicího systému NI CompactRIO. *Automa*. roč. 2012, č. 8-9.
- [9] BROŽ, Václav. Programovatelný regulátor rychlosti a polohy pro sběrnici CAN. *Automa*. roč. 2006, č. 1.
- [10] SINGULE, Vladislav. Vlastnosti a použití mikromotorů. *Automa*. roč. 2008, č. 3.
- [11] UZIMEX PRAHA. Malé stejnosměrné motory Maxon. [online]. 2002 [cit. 2014-06-04]
- [12] BROŽ, Václav. Snímače stejnosměrných motorů do 400W. *Technika*. roč. 2008, č. 4.
- [13] BROŽ, Václav. Programovatelný regulátor rychlosti a polohy pro sběrnici CAN. *Automa*. roč. 2006.
- [14] MAXON MOTOR. *EPOS 70/10 Positioning controller: Hardware Reference*. 752380-04. Sachseln, 2008. Dostupné z: http://www.maxonmotor.ch/medias/sys_master/8803613802526/300583-Hardware-Reference-En.pdf
- [15] ŠMEJKAL, Ladislav, ČERNÝ, Josef. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 19). *Automa*. roč. 2014, č. 3.
- [16] TRAVIS, Jeffrey a Jim KRING. *LabVIEW for everyone: graphical programming made easy and fun*. 3rd ed. /. Upper Saddle River, NJ: Prentice Hall, c2007, xli, 981 p. ISBN 9780131856721.
- [17] JOHNSON, Gary W a Richard JENNINGS. *LabVIEW graphical programming*. 4th ed. New York: McGraw-Hill, c2006, xv, 608 p. ISBN 0071451463.

- [18] SKALICKÝ, Jiří. Elektrické servopohony. Vyd. 1. Brno: PC-DIR Real, 1999. Učební texty vysokých škol. ISBN 80-214-1484-7.
- [19] ŠVARC, Ivan. *Automatické řízení*. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4398-3.
- [20] OLEHLA, Miroslav, Slavomír NĚMEČEK a Ivan ŠVARC. *Automatické řízení*. Liberec: Technická univerzita, 2009. ISBN 978-80-7372-484-9.
- [21] ÅSTRÖM, Karl J, Tore HÄGGLUND. PID controllers. 2nd ed. Research Triangle Park, N.C.: International Society for Measurement and Control, c1995. ISBN 1556175167.
- [22] MAXON MOTOR. *Positioning controller: Position regulation with Feed Forward*. Sachseln, 2008.
- [23] VÍTEČKOVÁ, M. *Seřízení regulátorů metodou inverze dynamiky*. VŠB-TU Ostrava, Fakulta strojní, 2000.