

Technická univerzita v Liberci

Fakulta strojní

Katedra aplikované kybernetiky

Studijní obor: 23-40-8 Automatizované systémy řízení ve strojírenství

Zaměření: Automatizace inženýrských prací

Laboratorní model řízení suportu NC stroje

Laboratory model of controlling of the NC machine slide rest

Alena Bartešová

Vedoucí diplomové práce: Ing. Slavomír Němeček

Konzultant diplomové práce: Ing. Petr Oliva

UNIVERZITNÍ KNIHOVNA
TECHNICKÉ UNIVERZITY V LIBerci



3146072681



ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení **Alena Bartešová**
obor **M2301 Strojní inženýrství – automatizované systémy řízení ve strojírenství**
zaměření **automatizace inženýrských prací**

Ve smyslu zákona č. 111/1998 Sb. o vysokých školách se Vám určuje diplomová práce na téma:

Laboratorní model řízení suportu NC stroje

Zásady pro vypracování:

(uved'te hlavní cíle diplomové práce a doporučené metody pro vypracování)

1. Seznamte se s problematikou vícehmotových systémů a PLC automatů. Proveďte analýzu požadavků na jejich řízení a navrhňte laboratorní model řízení dvouhmotového systému (dynamický model pohonu posuvu NC stroje).
2. Navrhňte vhodné technické a programové zabezpečení navrženého laboratorního modelu.
3. Realizujte řídicí obvod s využitím PLC automatu SIMATIC S7-300 a dalších nutných řídicích a výkonových jednotek.
4. Navrhňte a ověřte řídicí algoritmy pro řízení laboratorního modelu pohonu posuvu NC stroje.

Prohlášení

Prohlašuji, že diplomovou práci jsem vypracovala samostatně s použitím uvedené literatury.

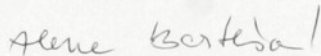
Byla jsem seznámena s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že s o u h l a s í m s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědoma toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Beru na vědomí, že si svou diplomovou práci mohu vyzvednout v Univerzitní knihovně TUL po uplynutí pěti let po obhajobě.

V Liberci dne 15. května 2004



Alena Bartešová

Poděkování

Na tomto místě bych chtěla poděkovat vedoucímu diplomové práce Ing. Slavomíru Němečkovi a konzultantovi Ing. Petrovi Olivovi za ochotu, pomoc a odborné vedení, které mi poskytovali při zpracování této diplomové práce.

Abstrakt

Tato práce se zabývá laboratorním modelem řízení dvouhmotového systému - modelem pohonu posuvu NC stroje. Model je řešen modulárně, řídicí činnost vykonává programovatelný automat. Práce je doplněna simulačním modelem, na kterém byly ověřeny tři modelové úlohy. Na závěr jsou uvedeny dva řídicí algoritmy, realizující jednoduchý pohyb saní suportu, z nichž první je také popsán sekvenčním funkčním grafem (SFC).

Abstract

This work deals with the laboratory model of controlling of the two-mass system – the model of the NC machine feed drive. The model is composed of moduls, the controlling is done by the programmable logic controller. The simulation model on which three model examples were checked, is a part of the work. Finally, two control algorithms performing a simple slide motion are shown. The first one is also described by the sequential function chart (SFC).

Obsah

1 Úvod	7
2 Vícehmotové systémy	9
2.1 Úvod	9
2.2 Dvouhmotový systém	10
2.3 Elektrické servomechanismy	11
2.4 Návrh laboratorního modelu	14
2.5 Popis laboratorního modelu	17
3 Programovatelné logické automaty	22
3.1 Návrh algoritmu	23
3.2 Programovací jazyky	24
3.3 Programování v systému STEP 7 Siemens	26
4 Simulační model	28
4.1 Model	28
4.2 Analýza výsledků	31
5 Řídící algoritmy laboratorního modelu	38
5.1 Varianta A – rychlostní servomechanismus	38
5.2 Varianta B – polohový servomechanismus	41
6 Závěr	44
Literatura	46
Seznam obrázků	47
Přílohy	48

Kapitola 1

Úvod

Řešená úloha je jednou z projektu modernizace laboratoře Katedry aplikované kybernetiky, zpracovaném J. Diesnerem (2001) – Model technologického zařízení – servopohon. Jejím cílem je prostudovat problematiku řízení polohového servomechanismu v interakci se zátěží a ověřit simulační model, reprezentující tuto soustavu.

Typickým představitelem polohového servomechanismu je pohon suportu NC stroje. Suport vykonává posuvný pohyb a zároveň nese nástroj, přes který se na něj přenášejí zatěžující síly, ovlivňující jeho polohu. Na přesnosti polohování suportu se ovšem podílí i pohon a možnosti řídicího systému. Proto byla tato soustava vybrána jako vzor, vhodný k demonstraci činnosti polohového servomechanismu v praktické části výuky.

V následující kapitole je po teoretické stránce rozebrán podrobněji systém dvouhmotový, jehož realizací je navržený laboratorní model. Rovněž je přiblížen rychlostní a polohový servomechanismus a je upozorněno na některé parametry, jež by měly být splněny. Dále je nastíněna problematika regulátorů – členů regulované soustavy. Následuje návrh laboratorního modelu ve dvou variantách, v jedné je použit pohon s rychlostním servomechanismem, ve druhé s polohovým. Součástí jsou také výpočty pro návrh pohonu. Nakonec je popsána struktura laboratorního modelu pro obě varianty.

V kapitole o programovatelných automatech je kladen důraz na programovou stránku – jsou uvedeny možné přístupy k návrhu algoritmů a stručný přehled programovacích jazyků. Rozpracována je partie o návrhovém jazyku SFC, jehož princip je popsán na návrhu algoritmu pro variantu s rychlostním servomechanismem. Dále je přiblíženo programovací prostředí STEP firmy Siemens, struktura programu programovatelného automatu a nakonec specifika při použití polohovacího modulu.

Další kapitola je věnována simulačnímu modelu. Vedle jeho provedení jsou popsány také parametry nutné pro nastavení modelu tak, aby odpovídal třem modelovým úlohám. Na závěr je provedeno shrnutí chování modelu na základě výstupů z modelových případů, z nichž lze usuzovat na chování laboratorního modelu.

V navazující části je uvedeny a popsány řídicí algoritmy pro obě varianty servomechanismů zapsané v jazyce STL.

V závěru je zhodnocen přínos celé práce s výhledem na další vývoj úlohy.

Kapitola 2

Vícehmotové systémy

2.1 Úvod

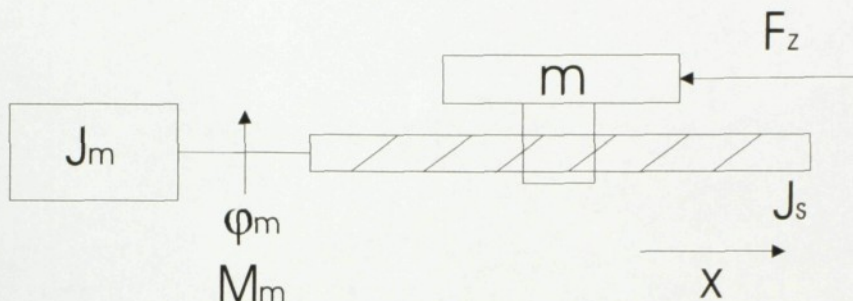
Zkoumání reálného systému lze provádět na výpočtovém modelu, který má zachytit vlastnosti systému podstatné pro výzkum. V systému pohonu posuvu dochází k interakci sil a momentů, proto je mým cílem zkoumání dynamických vlastností a to především během přechodových dějů, které vznikají při změně silových účinků. Systém se skládá z několika různých částí, které na sebe vzájemně působí, a tím i na výsledné chování systému.

Mezi základní součásti s mechanickým působením patří motor, spojka, vodící šroub a suport. Pro návrh pohonu lze uvažovat jednohmotový systém, kdy jsou setrvačné hmoty zátěže redukovány na šroub. Takové zjednodušení ovšem neumožňuje zkoumat interakci sil v suportu. Ideálem by bylo uvažovat všechny poddajnosti převodů, zde motor – spojka – vodící šroub – suport. Výsledkem by tak byl vícehmotový systém – diskrétní soustava složená z hmotných bodů, případně tuhých hmotných těles propojených nehmotnými pružinami a nehmotnými viskózními tlumiči – který by pravdivěji odrazil reálný systém, ovšem za cenu významného nárůstu složitosti.

V našem případě, kdy je vodící šroub poháněn servomechanismem, lze použitím torzně tuhé pevné spojky omezit počet poddajných převodů na jeden – propojení suportu se šroubem. Tak vzniká dvouhmotový systém, jehož výpočtový model je jednodušší.

2.2 Dvuhmotový systém

Na obr. 2.1 je schematicky zachycen motor se suportem. Saně se zátěží a zátěžnou silou působí na pružný spoj, z druhé strany motor. Ten uvádí do pohybu vodící šroub, přes který je pomocí matice rotační pohyb převáděn v translační.

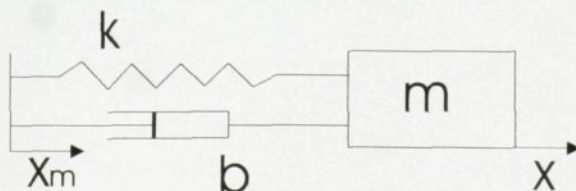


Obrázek 2.1: Model pohonu posuvu suportu

Legenda:

- M_m je hnací moment motoru,
- J_m moment setrvačnosti motoru,
- φ_m natočení hřídele motoru,
- J_s moment setrvačnosti šroubu,
- m zátěž (včetně hmotnosti suportu),
- F_z zátěžná síla.

Na obr. 2.2 je pak zobrazen příslušný výpočtový model suportu. Síly vyvozované pružinou a viskózním tlumičem vyjadřují charakteristické vlastnosti pružného spoje šroub – matice. Protože se jedná o lineární prvky, můžeme hovořit o lineární diskrétní soustavě a to o jednom stupni volnosti, jenž je dán polohou $x(t)$.



Obrázek 2.2: Výpočtové schéma suportu

Legenda:

b vizkózní tlumení převodu šroub – matice,

k torzní tuhost převodu šroub – matice.

Pro tento dvouhmotový model pohonu posuvu suportu platí následující pohybová rovnice

$$m\ddot{x} + b(\dot{x}_m - \dot{x}) + k(x_m - x) = F_z(t).$$

Po Laplaceově transformaci platí pro výslednou polohu suportu $X(s)$

$$X(s) = \frac{G_1(s)G_2(s)X_m(s) + G_2(s)F_z(s)}{1 + G_1(s)G_2(s)}$$

kde přenos $G_1(s) = bs + k$ a přenos $G_2(s) = 1/ms^2$.

Dále platí přenosy – pro $F_z(t) = 0$

$$\frac{X(s)}{X_m(s)} = \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)} = \frac{bs + k}{ms^2 + bs + k}$$

a pro $x_m(t) = 0$

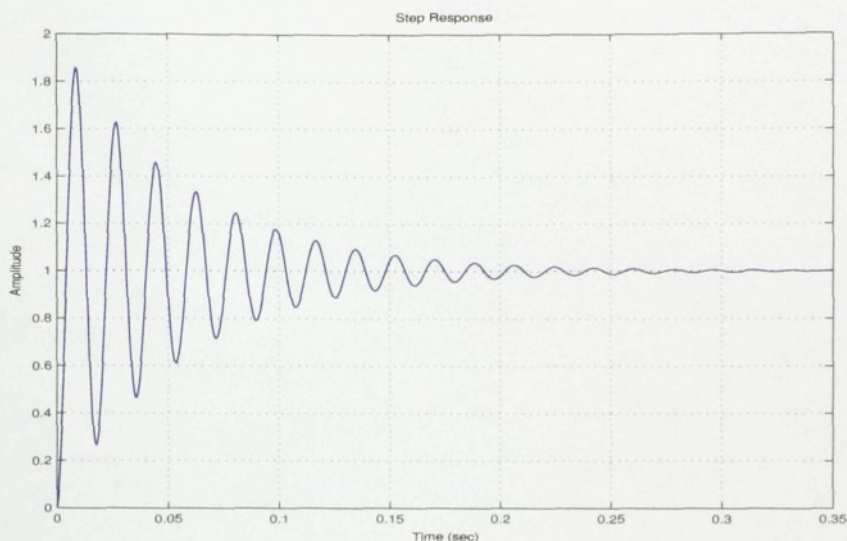
$$\frac{X(s)}{F_z(s)} = \frac{1}{1 + G_1(s)G_2(s)} = \frac{1}{ms^2 + bs + k}.$$

Z přechodové charakteristiky přenosu X/X_m uvedené na obr. 2.3 je patrné, že suport je proporcionální kmitavý člen 2. řádu.

2.3 Elektrické servomechanismy

Na pohon posuvu jsou kladeny vysoké nároky, protože přímo ovlivňuje přesnost, řezný výkon a produktivitu. Mezi hlavní požadavky patří rychlost a přesnost polohování. To je zajištěno použitím negativních zpětných vazeb – proudové, rychlostní a pokud je zapotřebí polohové. Proudová a rychlostní smyčka regulují statické a dynamické vlastnosti vlastního pohonu, rychlostní navíc spolu s polohovou mají přímý vliv na výsledné parametry servomechanismu. Regulace je zajištěna regulátory (jeden v každé smyčce) s takovými vlastnostmi, aby co nejlépe vyhovovaly kladeným požadavkům.

Proudová smyčka má za úkol přesné a rychlé řízení proudu motoru a potlačení nelinearity zatěžovacích charakteristik a vlivu indukčnosti. To je umožněno vysokým zesílením proudové smyčky K_i , která je tak odolná vůči vlivu poruchových signálů. Její kvalitní činnost je základem pro správnou funkci nadřazené rychlostní smyčky. Smyčka je regulována PI regulátorem.



Obrázek 2.3: Přechodová charakteristika přenosu X/X_m

Rychlostní servomechanismus

Cílem rychlostní smyčky je co nejpřesnější sledování řídicího signálu polohové odchylky a co nejmenší závislost pohonu na poruchách. Smyčka je též regulována PI regulátorem. Zesílení rychlostní smyčky K_r klesá s rostoucími setrvačnými hmotami, proto rychlostní servomechanismus vyžaduje adaptivní nastavování parametrů regulátoru. Pro rychlou regulaci se volí velké zesílení, ovšem volbou příliš velkého zesílení může docházet k překmitům větším než 30% ustálené hodnoty, což je nežádoucí. Rychlost motoru v ustáleném stavu je dokonale otáčkově tuhá.

Mezi základní požadavky na rychlostní smyčku patří jednak rovnoměrnost pohybu ($\Delta\varphi \leq 0,1$), jednak velikost propustného pásma (nejvyšší frekvence vstupního signálu, kterou obvod dokáže sledovat bez zkreslení), která se má pohybovat od 50 Hz výše.

Polohový servomechanismus

Doplnění soustavy polohovou vazbou umožňuje zajistit maximální přesnost polohování, danou odměřovacím systémem, protože žádaná poloha je dosažena v mo-

mentě, kdy rozdíl žádané a aktuální polohy je roven nule. Významně se na tom podílí i požadavek dojezdu bez překmitu, který je splněn vhodnou volbou zesílení polohové smyčky K_v (v rozmezí $(20 \div 50) s^{-1}$). Polohová smyčka je vybavena pouze P regulátorem, protože použitím PI regulátoru dochází k nerovnoměrnosti pohybu. Polohová smyčka je rovněž staticky dokonale tuhá, navíc nezávislost zesílení K_v na změnách setrvačné zátěže nevyžaduje na rozdíl od rychlostního servomechanismu použití adaptivního regulátoru.

Pro polohovou smyčku je významné dosažitelné zesílení K_v , které by mělo být co největší, ovšem musí být splněna již dříve zmíněná podmínka dojezdu bez překmitu, dále podíl kmitavých složek v přechodovém ději by měl být menší než 20% a nakonec musí být také zajištěna dostatečná dynamická tuhost k .

Regulátory

Regulátory se zařazují do soustavy se zpětnou zápornou vazbou. Podle druhu signálu se dělí na spojitě a nespojitě. Spojitě obsahují jen členy se spojitou statickou charakteristikou, jsou složitější, zato s kvalitnějším regulačním pochodem. Nespojitě mají nejméně jeden člen s nespojitou statickou charakteristikou a jsou jednodušší.

Spojitě regulátory jsou v různé kombinaci složeny ze tří složek: proporcionální P, integrační I a/nebo derivační D. Mezi nejvíce používané patří regulátory P, PI, PD a PID. Mezi regulátory bez zpoždění (ideální) patří všechny elektrické a některé pneumatické, mezi ty se zpožděním 1. řádu některé pneumatické a všechny hydraulické.

Hodnota akční veličiny $u(t)$, vycházející z PID regulátoru, je určena následujícím vztahem:

$$u(t) = k_R \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right) + u(0),$$

kde $e(t)$ je odchylka,

k_R je proporcionální zesílení všech složek,

T_I je integrační časová konstanta,

T_D je derivační časová konstanta.

V případě, že regulátor nemá I složku, je konstanta T_I nekonečně velká, v případě, že nemá D složku, konstanta T_D je rovna nule.

Po Laplaceově transformaci má vztah ideálního regulátoru tvar

$$R(s) = k_R \left(1 + \frac{1}{T_I s} + T_D s \right) = r_0 + \frac{r_1}{s} + r_2 s,$$

kde parametry r_0, r_1, r_2 jsou použity v simulačním modelu, a vztah regulátoru se

zpožděním 1. řádu

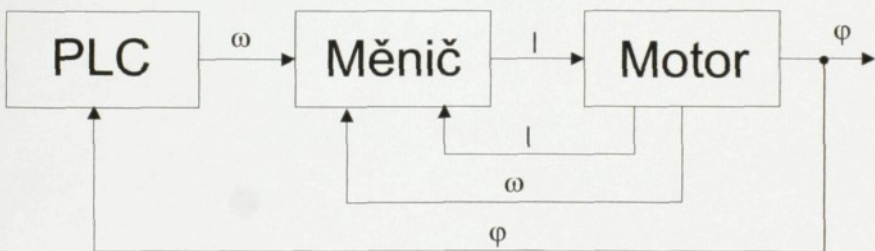
$$R(s) = k_R(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + T_V s}),$$

kde T_V je časová konstanta zpožďujícího členu.

Nejobtížnější částí regulační úlohy je nastavení parametrů regulátoru (zde r_0, r_1, r_2) tak, aby byl regulátor dostatečně dynamický a zároveň stabilní a nekmitavý. Navíc v reálném prostředí dochází ke změně podmínek, např. vlivem nelinearit, které vyžadují přenastavení parametrů. Pak je vhodné zvážit použití adaptivních regulátorů (průběžně se samočinně přizpůsobují) nebo robustních (málo závislých na změnách soustavy).

2.4 Návrh laboratorního modelu

Suport NC stroje je poháněn vodicím šroubem – v modelu bude představován lineární osou s kuličkovým šroubem a saněmi. Na saních bude umístěna zátěž, modelující zatížení řeznou silou F_C , která je největší z působících sil a může dosahovat velikosti až stovek newtonů. Pohon bude zajištěn polohovým servomechanismem, složeným z měniče a motoru. S osou bude motor spojen pevnou spojkou. Motor pro laboratorní úlohu musí být vhodný pro pohony posuvů, stačí s malým trvalým momentem, ke kterému bude z modulárních jednotek (regulační, výkonový a napájecí) sestaven měnič. Odměřovací systém je pro začátek zvolen jednoduchý (integrováný v motoru), v případě potřeby je vždy možno navrhnout jeho rozšíření.



Obrázek 2.4: Schéma vedení zpětných vazeb modelu

Řídící činnost bude zajišťovat programovatelný automat, který bude ovládat servomechanismus a zpracovávat zpětnovazební signály. Nejvhodnější by bylo univerzální

řešení, vhodné i pro jiné úlohy nalézající se v laboratoři, proto bylo zvoleno modulární řešení firmy Siemens. S využitím pouze modulu PLC je realizován rychlostní servomechanismus – varianta A, po doplnění polohovacím modulem je vytvořen jednoduchý NC stroj (polohový servomechanismus) – varianta B.

Blokové schéma na obr. 2.4 ukazuje provedení zpětných vazeb polohového servomechanismu modelu (varianta B). Vyplyvá z něho, že zatímco proudová a rychlostní smyčka se samočinně reguluje v měniči, polohová vazba musí být zpracována v programu. Toto řešení je výhodné pro experimentování.

Výpočty pro návrh pohonu

Velikost zátěže jsem zvolila $m = 400 \text{ kg}$.

Katalogové parametry lineární osy nutné pro výpočet:

zdvih kuličkového šroubu $l = 0,5 \text{ m}$,

průměr kuličkového šroubu $D = 0,012 \text{ m}$,

střední průměr kuličkového šroubu $d_S = 0,010 \text{ m}$,

stoupání šroubu $h_S = 0,01 \text{ m/ot}$,

poloměr valivého tělíška $r = 0,003 \text{ m}$.

Pro převod šroubu i_S platí $i_S = h_S/2\pi$.

Torzní tuhost k je definována vztahem

$$k = 1/c, \quad (2.1)$$

kde c je poddajnost, pro kterou platí

$$c = c_S + c_{TAH} + c_{KRUT}, \quad (2.2)$$

kde c_S je poddajnost šroub–matice,

c_{TAH} poddajnost šroubu v tahu,

c_{KRUT} poddajnost šroubu v krutu.

Poddajnost šroub–matice $c_S = 5,26 \cdot 10^{-9} \text{ mN}^{-1}$, protože tuhost matice–šroub je pro daný šroub $k_S = 190 \cdot 10^6 \text{ Nm}^{-1}$.

Dále platí

$$c_{TAH} = \frac{l}{4ES}, \quad (2.3)$$

kde E modul pružnosti v tahu,

S průřez jádra kuličkového šroubu.

a také

$$c_{KRUT} = \frac{i_S^2 l_{max}}{G J_K}, \quad (2.4)$$

kde l_{max} je vzdálenost místa s největší poddajností od počátku na ose,

G modul pružnosti v krutu,

J_K modul tuhosti v krutu šroubu.

Pro $S = 7,854 \cdot 10^{-5} m^2$ je $c_{TAH} = 7,59 \cdot 10^{-9} m/N$,

pro $l_{max} = 0,25 m$ a $J_K = 9,82 \cdot 10^{-10} m^4$ je $c_{KRUT} = 7,57 \cdot 10^{-9} m/N$.

Tedy $c = 20,42 \cdot 10^{-9} m/N$ a $k = 48,97 \cdot 10^6 N/m$.

Pro přirozenou frekvenci f_0 platí vztah

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (2.5)$$

a její hodnota se má pohybovat v rozmezí $(50 \div 150) Hz$.

Zde je $f_0 = 55,69 Hz$, hodnota se tedy pohybuje v požadovaném intervalu.

Po zkontrolování tuhosti nyní můžeme přistoupit k návrhu motoru. Pro klidový moment na šroubu M_{oS} platí vztah

$$M_{oS} = \frac{k_T m g i_S}{\eta_S} + 0,005 k_T m g r, \quad (2.6)$$

kde k_T je konstanta mezi třecí a normální silou pro valivé vedení,

η_S účinnost kuličkového šroubu,

a pro pracovní moment při obrábění M_{pS}

$$M_{pS} = \frac{F i_S}{\eta_S} + 0,005 F r, \quad (2.7)$$

kde F je celková řezná síla, přitom $F = \sqrt{F_C^2 + F_f^2}$, kde F_C je řezná síla a F_f posuvová síla.

Pro $k_T = 0,005$, $\eta_S = 0,95$ je $M_{oS} = 0,03 Nm$,

pro $F_C = 500 N$ a $F_f = 125 N$, tudíž $F = 515 N$, a proto je $M_{pS} = 0,87 Nm$.

Pro trvalý moment bezkartáčového motoru M_{trv} musí platit, že

$$M_{trv} \geq 1,7 \frac{M_{oS}}{i}, \quad (2.8)$$

kde i je převod mezi motorem a kuličkovým šroubem, a zároveň

$$M_{trv} \geq M_{pS}. \quad (2.9)$$

Podle (2.8) pro $i = 1$ proto musí být $M_{trv} \geq 0,05 \text{ Nm}$ a rovněž podle (2.9) $M_{trv} \geq 0,87 \text{ Nm}$.

Při požadavku $n_N = 3000 \text{ ot/min}$ podmínce (2.9) vyhovuje motor Siemens 1FT5042-1AF71-3AA0, jehož trvalý moment má hodnotu $M_0(100K) = 1 \text{ Nm}$ a jeho jmenovitý moment $M_N = 1 \text{ Nm}$. Moduly měniče byly vybrány adekvátně podle katalogu.

2.5 Popis laboratorního modelu

Laboratorní model je složen z modulů firem Siemens a Festo, blokové schéma zapojení varianty A je uvedeno na obr. 2.5, varianta B na obr. 2.6.

řídící jednotka (Siemens)

1. **řídící modul** (controller) SIMATIC S7-300 (CPU314 IFM), 6ES7 314-5AE03-0AB0 - řídí celý laboratorní model, zpracovává uživatelský program, komunikuje s okolním prostředím. Je spojen s osobním počítačem, na kterém je nainstalováno programovací prostředí STEP.
2. **polohovací modul** (function module) FM 357-2, 6ES7 357-4AH01-0AE0
- pouze varianta B,
- umožňuje komunikaci mezi řídícím modulem a pohonem.
3. **napájecí modul** (power supply module) PS 307, 6ES7 307-1BA00-0AA0
- napájí řídící a polohovací modul.

lineární osa (Festo)

DGEL-25-500-SP-HD25-GK (Part No.193746) – je určena pro elektromechanický pohon s vřetenem, se zabudovanou spojkovou skříní a vedením pro velké zátěže, s motorem spojena pevnou spojkou.

Pro plnou funkčnost je třeba osu doplnit třemi přibližovacími čidly Festo SMT-8-PS-K-LED-24-B – spínačem referenčního bodu souřadnicového systému a dvěma koncovými spínači polohy (na každém konci osy jeden). Jedná se o bezdotyková indukční čidla, která se připevní přímo na osu do předpřipravených drážek. Při použití polohovacího modulu FM 357-2 lze na ose programově definovat pracovní prostory a doplnit ji softwarovými spínači koncové polohy.

pohon (Siemens)

SIMODRIVE 611 s analogovým řídicím rozhraním pro pohony posuvů.

1. měnič (drive converter)

(a) modul pohonu

Modul pohonu vytváří kompaktní celek, protože nastavovací modul je vložen do regulačního modulu a ten do výkonového.

- **výkonový modul** (power module), 6SN1123-1AA00-0HA1
– je řízen regulačním modulem napájí motor; napájí regulační modul.
- **regulační modul** (control module), 6SN1118-0AA11-0AA1
– řídí v otevřené smyčce motor přes výkonový modul, přijímá zpětnovazební signály z motoru.
Verze je s uživatelsky přívětivým rozhraním, které vyžaduje nastavovací modul. Má stejnou regulační kvalitu jako klasická verze, oproti ní je ovšem určeno pouze pro řízení jedné osy, zato úroveň řízení otevřené smyčky a operátorská úroveň řízení je vyšší.
- **nastavovací modul** (parameter module), 6SN1114-0AA01-0AA0
– jsou v něm trvale uloženy parametry specifické pro konkrétní osu stroje.

(b) napájecí modul (infeed module), 6SN1146-1AB00-0BA1

– transformuje třífázové síťové napětí na stejnosměrné 600 V, které napájí výkonový modul přes napájecí vedení (DC link), a zároveň poskytuje elektronická napětí (např. ± 24 V, ± 5 V) modulu pohonu přes sběrnici (equipment bus); při brzdění se v rezistorech brzdy energie přeměňuje v teplo, které se rozptyluje do okolního prostředí, (tj. energie se nevrací zpět do sítě), proto vyhovuje úlohám s máločetnými a krátkými brzdnými cykly.

2. servomotor, Siemens 1FT5042-1AF71-3AA0

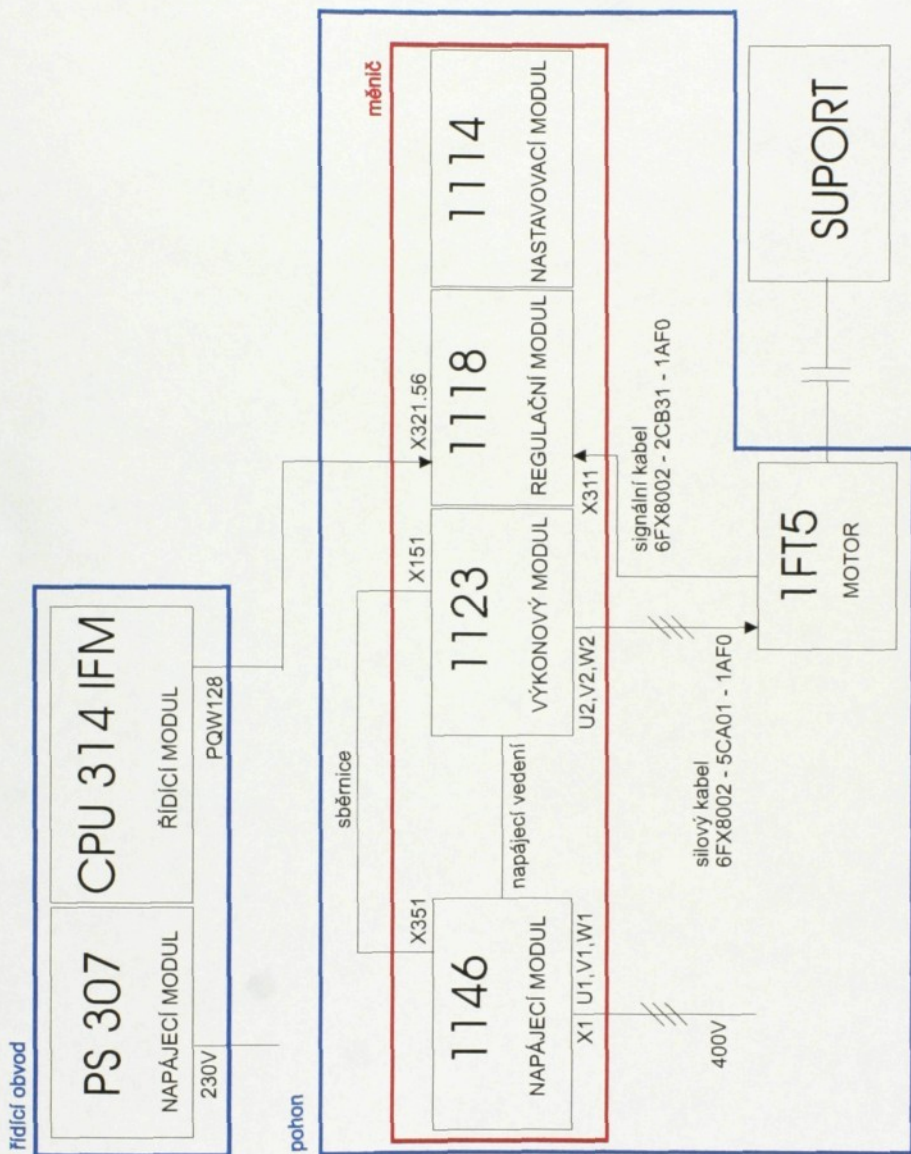
– rotační synchronní motor s integrovaným odměřovacím systémem je napájen stejnosměrným napětím 600 V se čtvercovým průběhem. Je bez větráku - stav přehřátí je zachycen PTC termistorem ve statorovém vinutí a reakci na něj, tj. okamžité vypnutí motoru, je třeba zahrnout do programu. Může být připojen přímo na hřídel stroje nebo na převodovku; je vhodný pro pohony posuvů mj. soustruhů a frézek, robotů, manipulačních zařízení.

3. odměřovací systém – integrovaný v motoru, analogový, Siemens 1FU1030

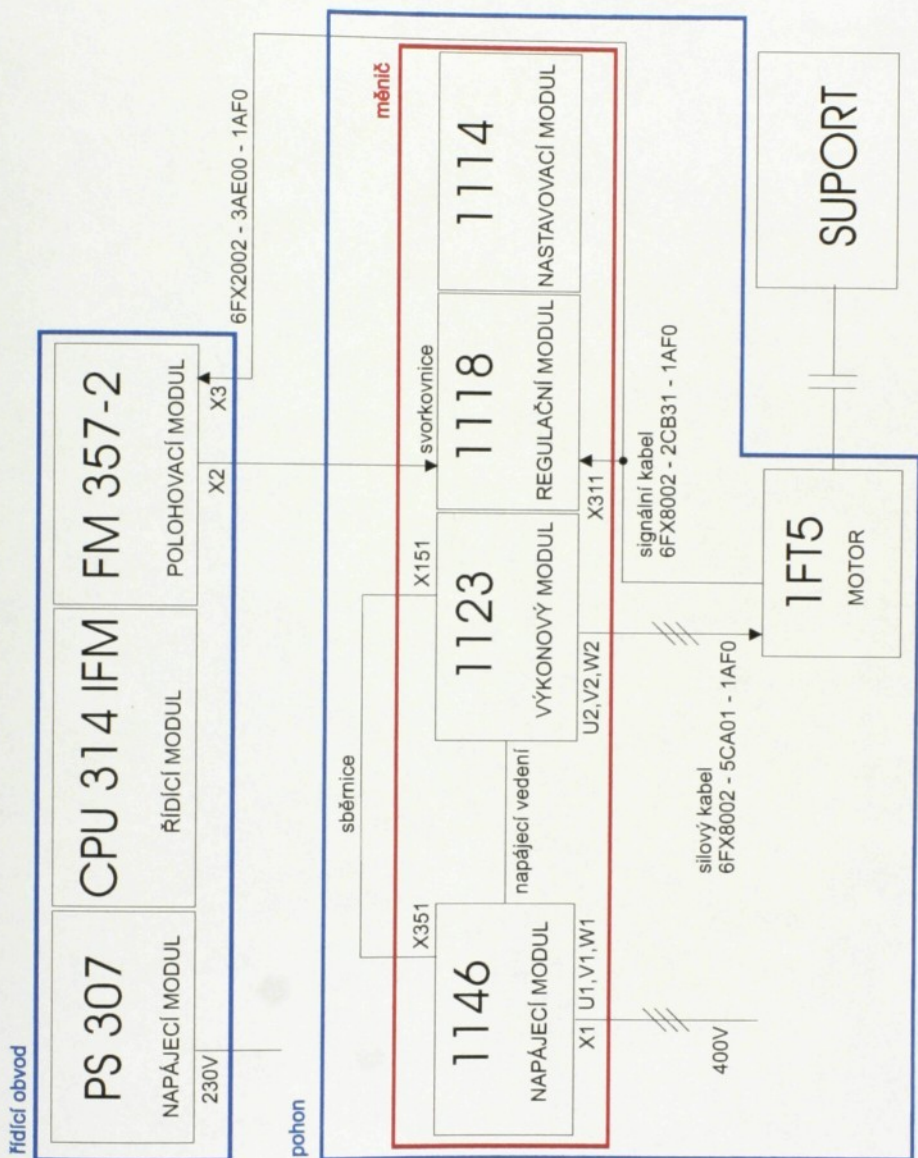
– tachometr pro odměřování rychlosti poskytuje lichoběžníkový napěťový signál; Hallova sonda pro odměřování polohy rotoru poskytuje signál s 18 pulsy na otáčku motoru.

propojení (Siemens)

1. **silový kabel** (power cable), 6FX8002-5CA01-1AF0 – spojuje výkonový modul s motorem, slouží k napájení.
2. **signální kabel** (signal/encoder cable), 6FX8002-2CB31-1AF0 – spojuje motor s regulačním a polohovacím modulem, přenáší zpětnovazební signál z motoru.
3. **signální kabel** (setpoint cable), 6FX2002-3AE00-1AF0 – spojuje polohovací modul s regulačním modulem, analogová hodnota signálu udává rychlost otáčení motoru.



Obrázek 2.5: Schéma zapojení modulů – rychlostní servomechanismus (var. A)



Obrázek 2.6: Schéma zapojení modulů – polohový servomechanismus (var. B)

Kapitola 3

Programovatelné logické automaty

Trendem posledních let ve vývoji programovatelných automatů je volba distribuovaného řešení automatizačních systémů. To na jedné straně umožňuje vytvořit jednoduchý řídicí systém PLC, na druhé straně je vyžadována snadná integrovatelnost do informačních sítí a s tím související schopnost komunikace alespoň v některém z komunikačních standardů.

V oblasti hardwaru je situace na trhu přehledná. K běžnému vybavení programovatelného automatu patří schopnost zpracovat jak digitální, tak analogové signály, přístup ke komunikační sběrnici, montáž do běžných průmyslových rozváděčů nebo simulační a testovací nástroje. Naopak v oblasti programového vybavení probíhal vývoj v 70. a 80. l. 20. st. poměrně nezávisle na hlavním proudu vývoje (univerzálních) počítačů. Byly vyvinuty vlastní programovací postupy a nástroje, které vycházely z reléové logiky (liniová schémata), později doplněné o typové logické funkce, které umožnily sestavovat program z bloků. Objektově orientované programování, strukturované programování, algoritmické jazyky pro aplikace reálného času, systémová analýza neměly na tuto oblast významný vliv.

Změna nastala až v pol. 90. l., kdy byla veřejností přijata metoda SFC – jazyk sekvencních funkčních grafů. Její počátek ovšem sahá do 70. l., kdy byly objeveny, dá-li se to tak nazvat, tzv. Petriho sítě. Pod tímto pojmem se skrývá matematický nástroj pro zkoumání řízení procesů. Na jeho základě byla vyvinuta grafická metoda GRAFCET – funkční graf vazeb kroků a přechodů (IEC 848), a z ní pak právě metoda SFC. Rozdíl mezi těmito dvěma metodami nejsou výrazné.

Dalším významným krokem bylo také přijetí normy IEC 1131-3, kde jsou specifikovány programovací jazyky pro PLC a to včetně návrhového jazyka SFC.

3.1 Návrh algoritmu

Východiskem pro návrh algoritmu je analýza systému. Tu je možné dekompozicí struktury(stavů) – metodou stavů a přechodů, nebo dekompozicí chování – metodou kroků a přechodů.

U dekompozice stavů je složitý systém rozložen na jednodušší subsystémy – moduly tak, aby byla vytvořena rovnováha mezi počtem modulů a jejich složitostí. Každý modul má obecně méně stavových veličin než celek. Chování modulu pak musí být vyjádřeno algoritmem, který popíše všechny možné přechody mezi stavy všech stavových veličin modulu. To u složitějšího systému vede ke ztrátě přehledu. Chování systému se znázorní stavovým diagramem, což je osvědčený analytický nástroj, vhodný ale právě pro analýzu a již méně pro návrh algoritmů a jejich implementaci.

Při dekompozici chování je vhodné navázat na stavovou dekompozici. Chování systému je v tomto případě chápáno jako souhrnný proces, chování modulů jako dílčí procesy. Na modulu jsou pak definovány všechny procesy (případně ještě rozdělené do dílčích) a zároveň se určí vztahy mezi procesy (základní nebo doplňkový proces, procesy obsluhující nestandardní situace, restarty a přerušení). Dílčí proces zahrnuje pouze ty stavové proměnné v těch stavech, které jsou danou činností ovlivněny. V každém časovém okamžiku algoritmus sleduje pouze aktivní kroky, čímž je počet operací značně redukován. Je zřejmé, že tento přístup je efektivnější než přechozí. Jako příklad lze uvést typickou úlohu PLC – automatizaci ovládání, kde je systém složen z velkého počtu vzájemně souvisejících procesů.

V jednoduchém systému, kde vystačíme s kombinační logickou funkcí, se používá při návrhu algoritmu několik metod - např. S-mapa, která je zobecněním pravdivostní tabulky a pravdivostní řady. Čísla jsou řazena ve vzestupné posloupnosti a kódována ve dvojkové soustavě narozdíl od dalšího způsobu návrhu pomocí K-mapy, kde čísla jsou řazena rovněž ve vzestupné posloupnosti, ale v Grayově kódu.

Algoritmus systému, ve kterém je nutno použít sekvenční logickou funkci, lze navrhnout již zmíněným stavovým způsobem popisu s využitím stavového grafu nebo přechodového diagramu konečného automatu (Mooreova nebo Mealyho typu) nebo metodou SFC, která je vhodná právě k popisu procesů. Tento grafický jazyk si dále popíšeme blíže.

3.2 Programovací jazyky

Programovací jazyky pro systémy PLC mají za sebou svůj rozmanitý vývoj, který vedl v normě IEC 1131-3 k definici dvou jazyků textových – jazyku mnemokódů (IL, Instruction List) a strukturovaného textu (ST, Structured Text) – a dvou grafických – jazyku kontaktních schémat (LD, Ladder Diagram) a jazyku logických schémat (funkčních bloků)(FBD, Function Block Diagram). Též byl rozpracován jazyk SFC (Sequential Function Chart), který slouží k návrhu sekvenčních logických algoritmů.

Podle J. Cendelína (2003) budoucnost patří strukturovanému textu v kombinaci s metodou SFC. Jazyk ST, který má blízko ke skriptovacím jazykům, spolu se SFC, který umožňuje přehledně definovat procesy, nejlépe vyhovuje současným nárokům na programovací jazyk.

Sekvenční funkční graf SFC

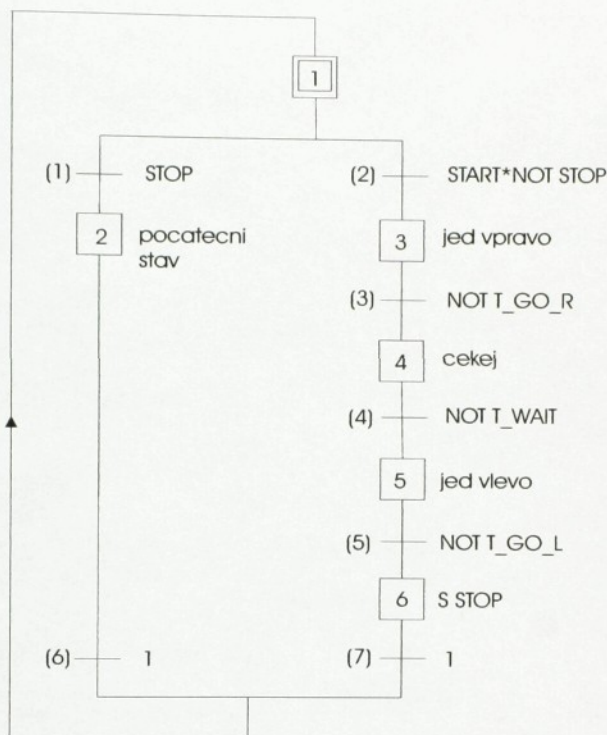
Je vhodný k hrubé analýze nebo návrhu algoritmu. Jeho kvality jsou vystupují až ve spojení s konkrétním programovacím jazykem, kterým je zapsán každý jednotlivý krok. Výhodou je, že každý krok může být naprogramován v jiném programovacím jazyce.

Tato metoda pracuje se širokým spektrem proměnných: logické, analogové, časové a textové řetězce, a požadavků: respektování časových závislostí, koordinace dílčích procesů, paralelismus a synchronizace procesů. Mezi jeho nedostatky patří způsob vyjádření větvení procesů, ke kterému dochází v závislosti na splnění podmínek, které se ovšem nemusejí vzájemně vylučovat. Způsob, jakým jsou pak větvení implementována, záleží na programátorovi. Další komplikace se objevují při zavádění cyklů, protože SFC nesplňuje požadavek strukturovaného programování na neexistenci provázaných smyček v algoritmu.

Pomocí SFC lze program strukturovat do posloupnosti kroků a přechodů navzájem spojených orientovanou úsečkou. Ke každému kroku je vázána skupina akcí a je buď aktivní, nebo neaktivní. Každý přechod je spojen s podmínkou a je buď uvolněný, nebo neuvolněný. Přechod je uvolněn právě tehdy, pokud krok před ním je aktivní. Pokud je splněna podmínka přechodu, dojde k přeskočení na další krok, který je tak aktivován.

Konkrétní návrh algoritmu pro variantu A laboratorního modelu je uveden na obr. 3.1. Na něm si popíšeme základní mechanismy jazyka SFC. Krok 1 v dvojitém čtverečku je počáteční krok. Následuje výlučné větvení – divergentní OR, který tak, jak je zapsán, splňuje požadavek, aby proměnná *STOP* měla prioritu. Tudiž,

pokud $STOP=1$ – přechod (1), je aktivován krok 2. V něm je systém uveden do počátečního stavu nastavením příslušných proměnných.



Obrázek 3.1: Varianta A zapsaná jazykem SFC

V případě $STOP=0$ a zároveň $START=1$ – přechod (2), je aktivován krok 3, jehož sada akcí uvádí osu do pohybu směrem vpravo. To trvá tak dlouho, dokud běží časovač pohybu vpravo. Pak dojde k přeskočení přechodu (3) do kroku 4 – čekání následované krokem 5 – jed vlevo. Na závěr je v kroku 6 nastavena proměnná $STOP=1$. Přechody (6) a (7) jsou formální, aby byla splněna podmínka střídání kroků a přechodů. Následuje spojení – konvergentní OR.

V příkladu není použito paralelní větvení – divergentní AND, kdy při splnění podmínky přechodu je aktivováno více paralelních kroků a které je graficky označeno dvojitou čarou za přechodem, ani synchronizace – konvergentní AND, kdy k přeskočení dojde v případě, že všechny připojené kroky jsou aktivní a podmínka splněna.

Graficky je označena dvojitou čarou před přechodem.

Implementace algoritmu zapsaná jazykem STL fy Siemens je uvedena v kap. 5.1.

3.3 Programování v systému STEP fy Siemens

Programovací prostředí STEP fy Siemens je určeno pro systém MS Windows a odpovídá nárokům kladeným současnou dobou. Na výběr jsou tři programovací jazyky – LAD, FBD a STL. Je doplněno o simulační nástroj.

Struktura programu

Program se skládá z procesů (OB), které mají specifické vlastnosti (funkce, doba volání), funkcí, funkčních bloků (funkce s pamětí) a datových bloků. Nezbytné procesy jsou OB1, který zajišťuje cyklické vykonávání programu, dále OB82, obsluhující chyby, a OB100, vykonávající inicializaci.

Pro PLC je charakteristická práce se vstupy a výstupy – v průběhu jednoho cyklu procesu OB1 řídicí jednotka nepracuje s přímými vstupy a výstupy, ale nýbrž s jejich obrazy. Řídicí moduly fy Siemens mají průběh cyklického procesu OB1, který vstupy a výstupy obsluhuje, následující:

1. operační systém začne odpočítávat monitorovací čas cyklu,
2. CPU přenesne hodnoty z tabulky obrazů výstupů na výstupy modulů,
3. CPU aktualizuje tabulku obrazů vstupů,
4. CPU zpracuje uživatelský program,
5. operační systém vykoná zbylé úlohy (např. stáhnutí nebo smazání bloků),
6. CPU zajistí návrat na začátek programu.

V programu lze adresovat podle potřeby jak obrazy, tak přímé vstupy/výstupy. Jejich značení je následující:

- I obraz vstupu,
- PI přímý vstup,
- Q obraz výstupu,
- PQ přímý výstup.

Přiřazením datové jednotky (b = X bit, B byte, W word, D double word) je určena velikost, např. PQW - přímý výstup o velikosti word. Při adresování vstupů a výstupů se specifikace bitu neuvádí, narozdíl od práce s datovými bloky (viz např. Komunikace s FM dále v této kapitole).

Úprava programu pro obsluhu FM

Pro zajištění správné funkce polohovacího modulu FM 357-2 je třeba nejprve projekt vybavit objekty k tomu určenými (konkrétně viz příloha č. 3 – Standard function blocks for the FM 357-2 overview). V založeném projektu nahradíme všechny implicitně vytvořené bloky a zdrojové soubory těmi z knihovny FM357-2L, která je dostupná v knihovnách SIMATIC Manageru.

Následuje parametrizace – přizpůsobení jednotky aplikaci nastavením příslušných parametrů (např. stoupání pohybového šroubu, převod mezi motorem a pohybovým šroubem, maximální rychlost rotoru atd.) v hardwarové části projektu. Poté v programové části po úpravě vstupních parametrů v procesech OB 100 a OB 82 tak, aby vyhovovaly požadované aplikaci, vložení uživatelského programu do procesu OB1 a kompilaci lze přistoupit ke zkopírování bloků do PLC.

Komunikace s FM

Komunikace s FM probíhá formou řídicích a kontrolních příkazů. Řídící příkaz mění stav, v jeho názvu se vyskytuje řetězec CTR (control), kontrolní příkaz stav ověřuje, v jeho názvu se vyskytuje řetězec CHB (checkback). Dále se příkazy rozdělují na příkazy týkající se kanálů („FMx“) a na příkazy týkající se os („AXy“).

Osy lze řídit jednotlivě, ale zároveň je možno je seskupovat do skupin a ty řídit podle konkrétních potřeb. Např. příkaz FM1.CH[3].CHB.CH_RDY je kontrolní příkaz pro kanál č. 3, při log. 1 znamená, že kanál je připraven k použití, při log. 0, že není. Obdobně příkaz AX1.AX[2].CTR.PULSE_EN je řídicí příkaz, aktivující osu č. 2.

Uvedené názvy příkazů jsou symbolické, za nimi se skrývá přímé pojmenování datové jednotky v datovém bloku. Např. výše uvedený příkaz pro osu je totožný s názvem DB „AXy“ DBX13.7, což označuje 7. bit v 13. bajtu uživatelského datového bloku „AXy“.

Příkazy a jejich hodnoty jsou uloženy v uživatelských datových blocích FMx, AXy, které jsou automaticky generovány funkcí FC1 ze šablon UDT1 a UDT2 během startu.

Kapitola 4

Simulační model

4.1 Model

Simulační model jsem vytvořila v programu Matlab Simulink. Jeho struktura pohonu je uspořádána tak, aby přesně odpovídala struktuře laboratorního modelu, jak je naznačena na obr. 2.4. Realizace modelu je uvedena na obr. 4.1. K pohonu je přes převod, který převádí rotační souřadnici na posuvnou, připojen suport. Ze schématu je patrné, že síla vyvozená pružnou vazbou F_{pr} ovlivňuje motor. Zátěžná síla F_z představuje zatížení posuvovou silou F_f .

V simulačním modelu jsou použity tyto parametry –

konstanty motoru: momentová konstanta $C_u = 0,95 \text{ Nm/A}$,

odpor kotvy $R = 28,2 \Omega$,

indukčnost kotvy $L = 48,4 \text{ mH}$,

moment setrvačnosti motoru $J_m = 1,73 \cdot 10^{-4} \text{ kgm}^2$,

stálý moment $M_0(100K) = 1,0 \text{ Nm}$,

stálý proud $I_0(100K) = 1,1 \text{ A}$,

a konstanty vztahující se k suportu: zátěž m ,

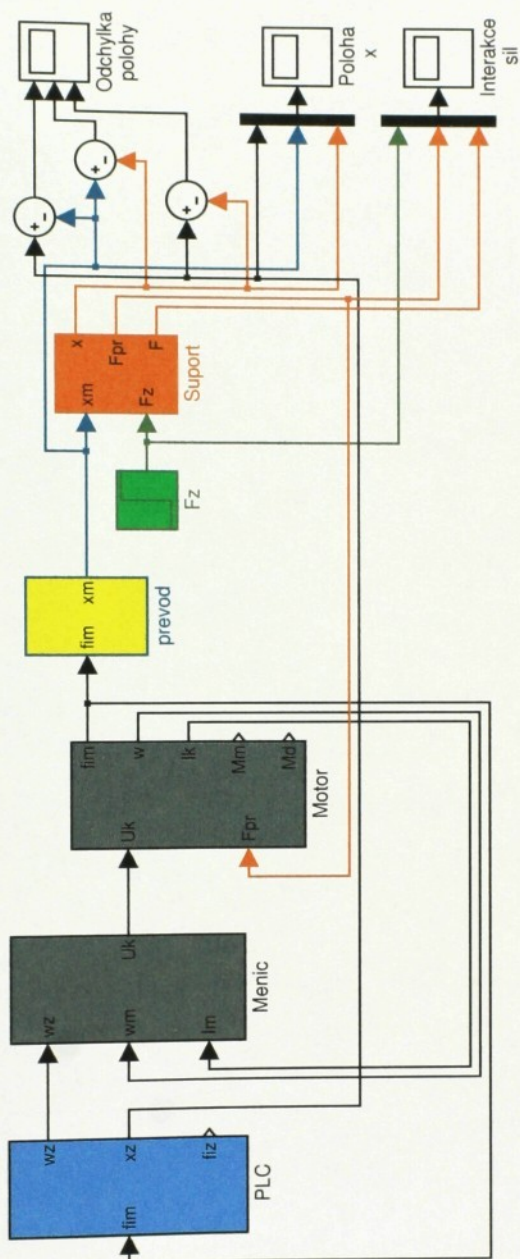
moment setrvačnosti šroubu $J_s = 2,32 \cdot 10^{-4} \text{ kgm}^2$,

koeficient tlumení b ,

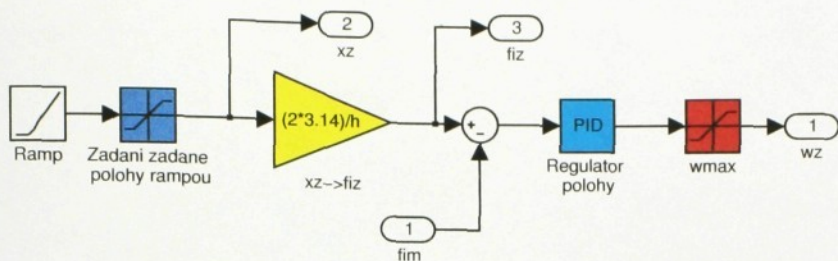
torzní tuhost je dle (2.1) $k = 48,97 \cdot 10^6 \text{ N/m}$.

Úprava modelu

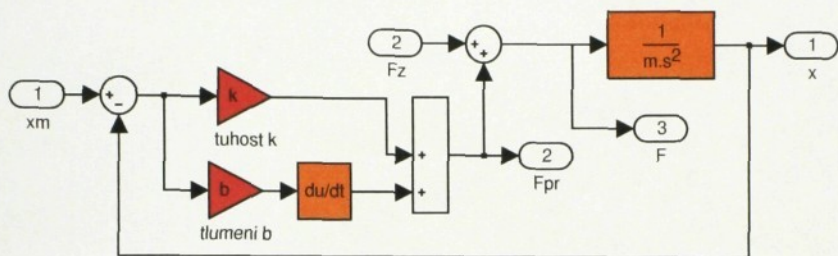
Přestože v laboratorním modelu je použit bezkartáčový EC motor, lze ho v simulaci po úpravě příslušných konstant nahradit motorem kartáčovým (Souček 1992, s. 32).



Obrázek 4.1: Schéma simulačního modelu – celek



Obrázek 4.2: Schéma simulačního modelu – PLC



Obrázek 4.3: Schéma simulačního modelu – Suport

Tuto variantu jsem zvolila, protože splňuje požadavky na výstup a jeho model je jednodušší. Jak P. Souček dále uvádí, parametry momentová konstanta C_u , odpor R a indukčnost L je nutno přepočítat takto: pro zapojení do hvězdy platí $\sqrt{3}C_u$, $2R$ a $2L$, pro zapojení do trojúhelníku C_u , $2/3 R$, $2/3 L$.

V našem případě jde o zapojení do hvězdy, proto parametry mají tyto hodnoty:

$$C_u = 1,65 \text{ Nm/A},$$

$$R = 56,4 \Omega,$$

$$L = 96,8 \text{ mH}.$$

Výpočty pro model a další parametry

Koeficient tlumení b lze vyjádřit vztahem

$$b = 2\xi\sqrt{km}, \quad (4.1)$$

kde ξ je poměrné tlumení.

Pro zátěž $m = 400 \text{ kg}$ a poměrné tlumení $\xi = 0,05$ je velikost tlumení $b = 14,00 \cdot 10^3 \text{ kg/s}$.

Pro zátěž $m = 10 \text{ kg}$ a poměrné tlumení $\xi = 0,05$ je velikost tlumení $b = 2,21 \cdot 10^3 \text{ kg/s}$.

V modelu jsou použity tři regulátory (viz kap. 2.3): PI regulátor v proudové smyčce má parametry $P = 15$ a $I = 1,5$, PI regulátor v rychlostní smyčce $P = 30$ a $I = 0,75$ a P regulátor v polohové smyčce $P = 30$.

Popis modelových případů

Simulace byla provedena na třech modelových případech, ve kterých je zkombinován pracovní posun s rychloposunem a dvě různé zátěže – 400 kg z důvodů uvedených v kap. 2.4 a 10 kg, která je použita na laboratorním modelu.

Modelový případ 1: Pracovní posun, zátěž 400 kg

Příklad představuje soustružení obrobku za těchto podmínek: $n = 360 \text{ ot/min}$, $f_{ot} = 0,3 \text{ mm/ot}$, $l = 0,01 \text{ m}$. Pro simulační model byla vypočtena rychlost $\omega = 1,13 \text{ s}^{-1}$. Zátěžná síla $F_z = 125 \text{ N}$ začne působit v čase $t = 0,4 \text{ s}$.

Modelový případ 2: Rychloposun, zátěž 400 kg

Příklad představuje rychloposuv suportu za těchto podmínek: $n = 360 \text{ ot/min}$, $v_f = 0,2 \text{ m/s}$, $l = 0,25 \text{ m}$. Pro simulační model byla vypočtena rychlost $\omega = 126 \text{ s}^{-1}$. Během pohybu nepůsobí zátěžná síla F_z .

Modelový případ 3: Rychloposun, zátěž 10 kg

Příklad je určen pro srovnání s laboratorním modelem a představuje rychloposuv suportu za těchto podmínek: $n = 360 \text{ ot/min}$, $v_f = 0,2 \text{ m/s}$, $l = 0,25 \text{ m}$. Pro simulační model byla vypočtena rychlost $\omega = 126 \text{ s}^{-1}$. Během pohybu nepůsobí zátěžná síla F_z .

4.2 Analýza výsledků

Pro každý případ jsou uvedeny tři grafy. První znázorňuje průběh pohybu suportu během najíždění do zadané polohy, druhý průběh odchylky konečné polohy suportu od polohy vynucené motorem a třetí interakci sil v suportu, které jsou důsledkem pružného spoje šroub – matice. Zároveň jsou pro srovnání uvedeny trvalé odchylky polohy v jednotlivých případech.

Modelový případ 1: Pracovní posun, zátěž 400 kg

Velikost trvalé odchylky přepočítané polohy motoru od žádané hodnoty je $x_z - x_m = -0,2 \mu\text{m}$, velikost trvalé odchylky polohy suportu od žádané hodnoty je $x_z - x = -2,8 \mu\text{m}$.

Modelový případ 2: *Rychloposun, zátěž 400 kg*

Velikost trvalé odchytky přepočítané polohy motoru od žádané hodnoty je $x_z - x_m = 0,3 \mu m$, velikost trvalé odchytky polohy suportu od žádané hodnoty je $x_z - x = 0,3 \mu m$.

Modelový případ 3: *Rychloposun, zátěž 10 kg*

Velikost trvalé odchytky přepočítané polohy motoru od žádané hodnoty je $x_z - x_m = 0,2 \mu m$, velikost trvalé odchytky polohy suportu od žádané hodnoty je $x_z - x = 0,2 \mu m$.

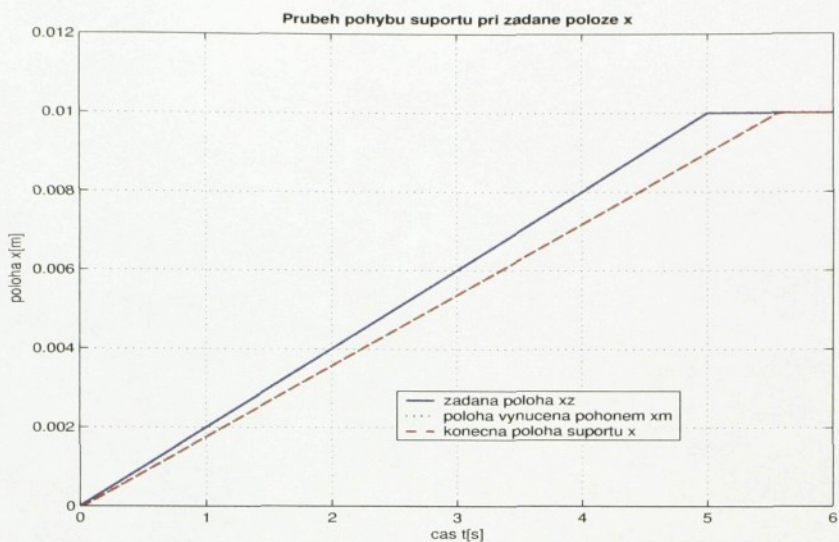
Shrnutí

V případech 2 a 3, kdy nepůsobí zátěžná síla, se obě trvalé odchytky pohybují v řádu desetin mikrometrů; navíc v každém případě mají odchytky tutéž hodnotu. Z toho vyplývá, že zde není nezbytné přímé odměřování polohy na suportu nebo použití korekční konstanty.

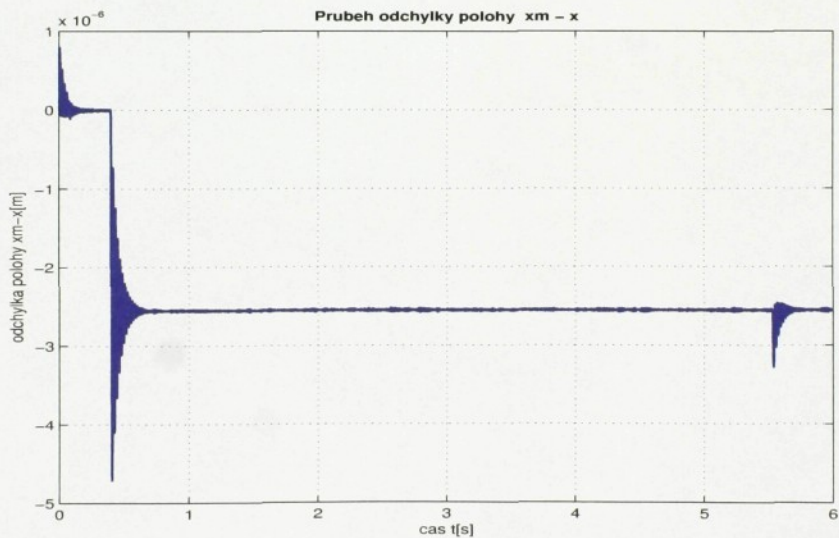
Poněkud odlišná situace je v případě 1, kdy vliv zatěžné síly F_z patrný je – trvalá odchytka skutečné polohy suportu od žádané hodnoty je v řádu mikrometrů. Největší měrou se na ní podílí odchytka polohy suportu od polohy vynucené motorem (obr. 4.5), která vznikla důsledkem pružné vazby šroub – matice. Zde vidím možnost využití buď korekční konstanty, zjištěné pro konkrétní podmínky, nebo přímého odměřování polohy suportu. Z obrázku také vyplývá, že vlivem zátěžné síly se soustava rozkmitá více než náhlou změnou rychlosti, nicméně k ustálení dojde pokaždé poměrně rychle.

Velká rychlost u velké zátěže při náhlé změně rychlosti způsobí nejvýraznější rozkmit soustavy (obr. 4.8), rovněž působící síly jsou značné (obr. 4.9). S klesající zátěží jsou vzniklé síly menší (obr. 4.12) a také trvá kratší dobu, než přechodový děj odezní (obr. 4.11). Tomu odpovídají i přechodové odchytky polohy suportu od polohy vynucené motorem.

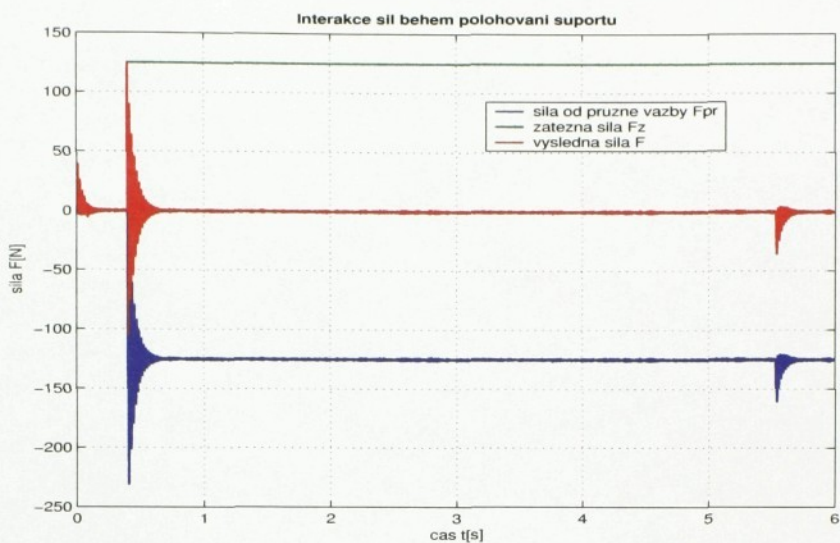
Po srovnání grafů na obr. 4.6 a obr. 4.12 lze usoudit na to, že interval, v němž se pohybuje ustálená velikost výsledné síly F je menší v případě větší zátěže, malé rychlosti a působení zatěžné síly F_z , tedy zvláště během samotné operace obrábění. Nakolik je tato hypotéza pravdivá a do jaké míry mají jednotlivé parametry význam, by mohlo být cílem dalšího zkoumání.



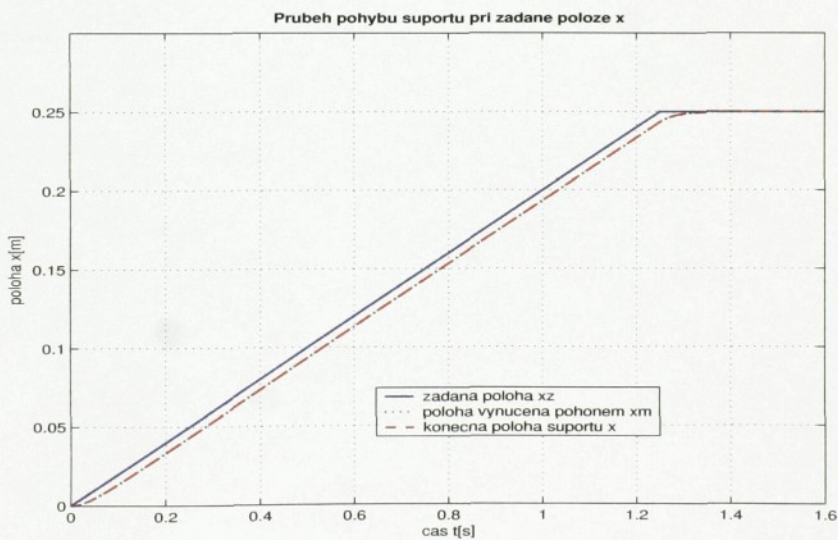
Obrázek 4.4: (1) Průběh pohybu suportu při $x = 0,01\text{ m}$, $m = 400\text{ kg}$, $F_z = 125\text{ N}$ v $t = 0,4\text{ s}$.



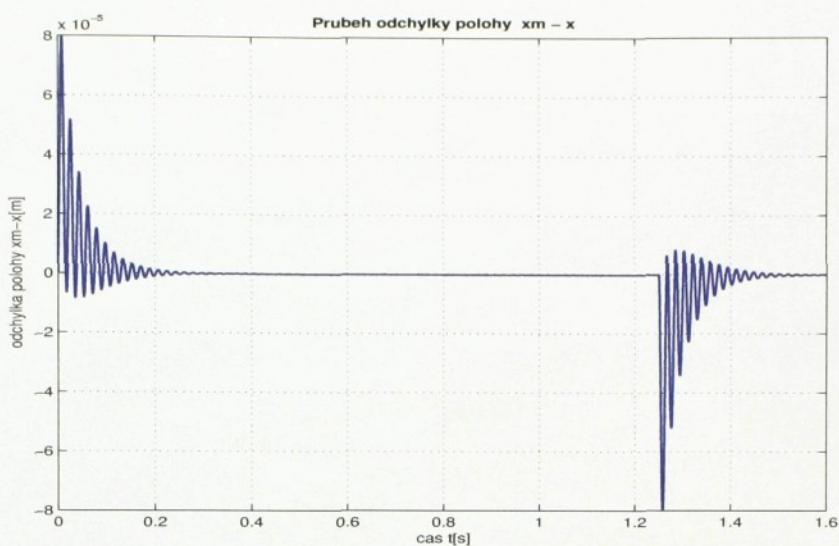
Obrázek 4.5: (1) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,01\text{ m}$, $m = 400\text{ kg}$, $F_z = 125\text{ N}$ v $t = 0,4\text{ s}$.



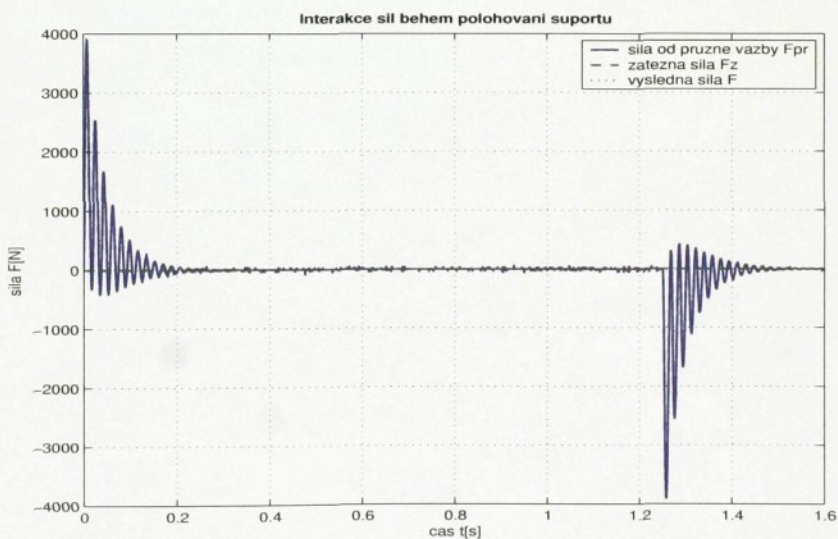
Obrázek 4.6: (1) Interakce sil během pohybu suportu při $x = 0,01 \text{ m}$, $m = 400 \text{ kg}$, $F_z = 125 \text{ N}$ v $t = 0,4 \text{ s}$.



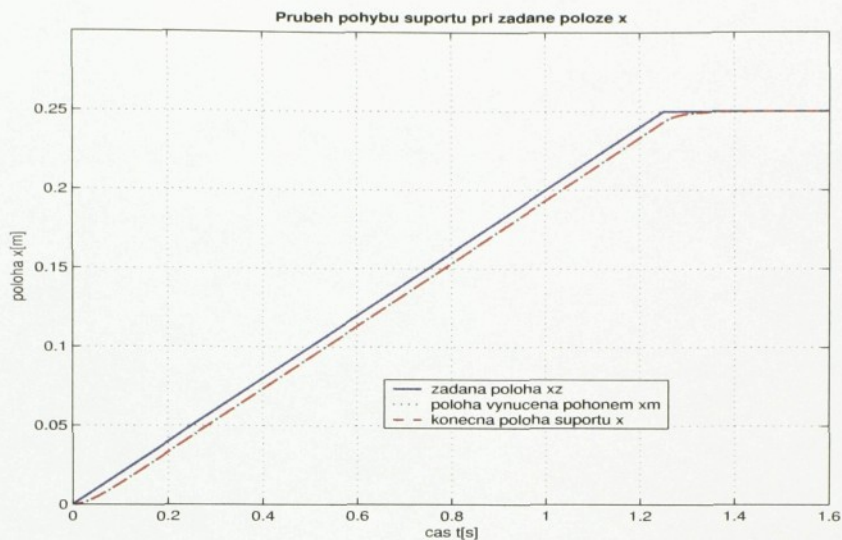
Obrázek 4.7: (2) Průběh pohybu suportu při $x = 0,25 \text{ m}$, $m = 400 \text{ kg}$, $F_z = 0 \text{ N}$.



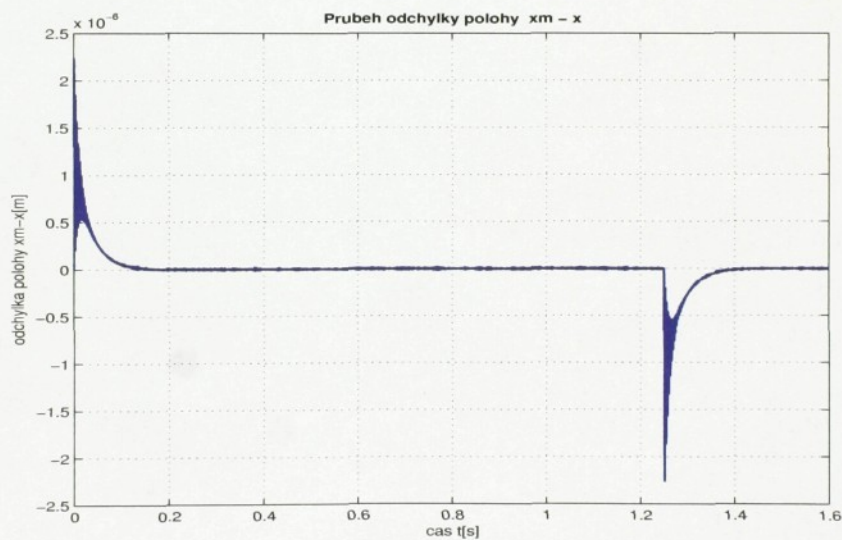
Obrázek 4.8: (2) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,25\text{ m}$, $m = 400\text{ kg}$, $F_z = 0\text{ N}$.



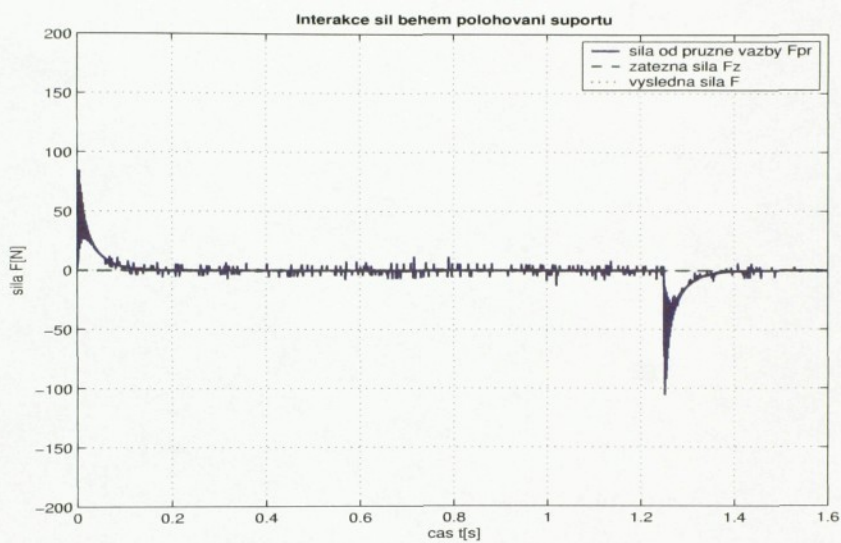
Obrázek 4.9: (2) Interakce sil během pohybu suportu při $x = 0,25\text{ m}$, $m = 400\text{ kg}$, $F_z = 0\text{ N}$.



Obrázek 4.10: (3) Průběh pohybu suportu při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$.



Obrázek 4.11: (3) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$.



Obrázek 4.12: (3) Interakce sil během pohybu suportu při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$.

Kapitola 5

Řídící algoritmy laboratorního modelu

Řídící algoritmy jsou zapsány v jazyce STL. Program pro rychlostní servomechanismus je jednodušší, proto je popsán po celcích, pouze některé příkazy jsou vysvětleny konkrétně. Program pro polohový servomechanismus je rozepsán detailně, protože jsou v něm použity příkazy polohovacího modulu FM 357-2, jejichž význam nemusí být na první pohled zřejmý.

Oba programy jsou uloženy v procesu OB1 (cyklické vykonávání programu). Oba rovněž používají pro přehlednost uživatelský datový blok, jehož symbolický název "USERDB". V něm jsou uloženy uživatelem definované proměnné.

5.1 Varianta A – rychlostní servomechanismus

proměnné typu **BOOL** uložené v datovém bloku "USERDB":

STOP – zakázat posuv

START – povolit posuv

GO_R – jeď doprava

T.GO_R – časovač jízdy vpravo

WAIT – čekej

T.WAIT – časovač čekání

GO_L – jeď doleva

T.GO_L – časovač jízdy vlevo

PQW128 – přímý analogový výstup

Proměnné GO_R, WAIT a GO_L jsou nastaveny na log.1 na znamení toho, co se má vykonat. Po spuštění příslušného časovače jsou resetovány.

Ošetření příkazu STOP - uvedení do počátečního stavu a skok na konec smyčky.

```
A "USERDB".STOP
R "USERDB".START
S "USERDB".GO_R
R "USERDB".T_GO_R
S "USERDB".WAIT
R "USERDB".T_WAIT
S "USERDB".GO_L
R "USERDB".T_GO_L
R "USERDB".STOP
JC END
```

Pohyb vpravo 0,25 m = napětí 0.71 V po dobu 4 s.

```
A "USERDB".START
A "USERDB".GO_R
AN "USERDB".T_GO_R
L S5T#4S // čas 4 s uložen do akumulátoru
SE T 1 // spuštěn časovač T1
R "USERDB".GO_R
```

```
A T 1
A "USERDB".GO_L
= "USERDB".T_GO_R
```

```
A "USERDB".T_GO_R
JCN END1 // pokud není, skočit na návěští END1
L W#16#7AE // napětí o velikosti 7AEh
T PQW 128 // na výstup PQW128
```

```
END1: NOP 0
```

Čekání 5 s = napětí 61.3 mV pro vyrovnání driftu.

```
A "USERDB".START
A "USERDB".WAIT
AN "USERDB".T_WAIT
AN "USERDB".GO_R
```

```
AN "USERDB".T.GO_R
L S5T#5S
SE T 2
R "USERDB".WAIT
```

```
A T 2
= "USERDB".T.WAIT
```

```
A "USERDB".T.WAIT
JCN END2
L W#16#C2
T PQW 128
```

```
END2: NOP 0
```

Pohyb vlevo 0,25 m = napětí -0,59 V po dobu 4 s.

```
A "USERDB".START
A "USERDB".GO_L
AN "USERDB".T.GO_L
AN "USERDB".WAIT
AN "USERDB".T.WAIT
L S5T#4S
SE T 1
R "USERDB".GO_L
```

```
A T 1
AN "USERDB".GO_L
= "USERDB".T.GO_L
```

```
A "USERDB".T.GO_L
JCN END3
L W#16#F9A1
T PQW 128
```

```
END3: NOP 0
```

Na závěr programu aktivován STOP.

```
A "USERDB".START
AN "USERDB".GO_L
```

```

AN "USERDB".T.GO_L
L W#16#C2
T PQW 128
S "USERDB".STOP

```

```

END: NOP 0

```

5.2 Varianta B – polohový servomechanismus

Uvedená funkce reaguje na pokyny předávané pomocí uživatelského datového bloku "USERDB", pomocí kterého je zadána žádaná poloha a rychlost otáček. V případě, že se suport v této poloze nenachází, řízení je předáno CPU, která samostatně do žádané polohy najede.

Úvodní operace.

```

CALL "BFCT" // synchronizace s FM 357-2
FMDB_NO:=30 // přiřazení čísla datovému bloku signálů FM357-2

```

```

A "FM1".STARTUP // probíhá start FM 357-2?
JC LAB1 // pokud probíhá spouštění FM 357-2, skočit na návěští LAB1

```

Vlastní program.

```

A "FM1".CONNECT // FM 357-2 připraveno komunikovat s CPU?
JCN END // pokud není připraveno, skončit na návěští END

L B#16#26 // 100 % override
T "FM1".CH[1].CTR.FD_OVERR // přiřadit hodnotu override pro rychloposuv a posuv kanálu
T "AX1".AX[1].CTR.OVERR // přiřadit hodnotu override ose
S "FM1".CH[1].CTR.FD_OVERR_FDR // při posuvu použít nastavenou hodnotu override
S "AX1".AX[1].CTR.ACT_OVERR // při pohybu osy použít nastavenou hodnotu override

```

Kontrola stavu, která probíhá na začátku každé smyčky programu.

```

A "FM1".CH[1].CHB.CH_RDY // kanál připraven k operaci?
JCN END // pokud není, skočit na návěští END
S "FM1".CH[1].CTR.MODE_JOG // nastavit mód Jog

```

```

R "FM1".CH[1].CTR.MODE_A // zrušit automatický mód
R "FM1".CH[1].CTR.REF_POINT // zrušit mód najetí do referenčního bodu
S "AX1".AX[1].CTR.PULSE_EN // aktivovat osu
A "USERDB".CTR.EN_AX // uzavřena polohová smyčka?
= "AX1".AX[1].CTR.CTR_EN // nastavit stav polohové smyčky

```

SET

```

S "USERDB".INCR // nastavit inkrementální mód
= "AX1".AX[1].CTRCA.INCR // nastavit inkrementální mód

```

Nahrání hodnot zadaných uživatelem do polohovacího modulu.

```

L "USERDB".POS // nahrát žádanou polohu
T "AX1".AX[1].CTRCA.POS // uložit žádanou polohu
L "USERDB".FRATE // nahrát rychlost otáček
T "AX1".AX[1].CTRCA.FRATE // uložit rychlost otáček

```

Plnění požadavků uživatele.

```

AN "USERDB".INPOS // není osa v žádané poloze?
A "USERDB".START // spustit osu?
S "AX1".AX[1].CTRCA.START // spustit osu
S "USERDB".CPUAX_FCT // osu řídit pomocí CPU

```

```

A "USERDB".CPUAX_FCT // řídit osu pomocí CPU?
AN "AX1".AX[1].CTRCA.START // osa není spuštěna?
R "USERDB".START // zrušit start od uživatele
R "USERDB".CPUAX_FCT // zrušit řízení osy pomocí CPU

```

```

A "AX1".AX[1].CHBCA.INPOS // je osa v žádané poloze?
= "USERDB".INPOS
R "AX1".AX[1].CTRCA.START // vypnout osu

```

Kontrola stavu osy.

```

A "AX1".AX[1].CHBCA.ACTIV // je osa aktivní?
= "USERDB".ACTIV

```

Ošetření chybového stavu.

```
A "AX1".AX[1].CHBCA.ERR // hlásí osa chybu?  
JCN NOER // pokud ne, skočit na návěstí NOER  
S "USERDB".ERR // nastavit chybový bit  
R "AX1".AX[1].CTRCA.START // vypnout osu  
L "AX1".AX[1].CHBCA.ERR_NO // nahrát číslo chyby  
T "USERDB".ERR_NO // uložit číslo chyby  
JU END // skočit na návěstí END
```

```
NOER: R "USERDB".ERR // zrušit chybový bit  
L B#16#0  
T "USERDB".ERR_NO // smazat číslo chyby
```

```
LAB1: NOP 0  
END: BE
```

Kapitola 6

Závěr

Na závěr diplomové práce bych zhodnotila přínos laboratorní úlohy pro studenty a naznačila směry, kterými by se úloha mohla dále rozvíjet.

Laboratorní i simulační model lze využít např. při výuce Základů aplikované kybernetiky pro praktickou ukázkou toho, jak typ regulátoru a nastavení jeho parametrů působí na kvalitu regulačního pochodu při dosahování žádané polohy. V Servomechanismech lze názorně ukázat rozdíly mezi rychlostním a polohovým servomechanismem a také, jaký vliv mají tyto rozdíly na přesnost polohování při různých kombinacích parametrů. Simulační model zde může být rozpracován o adaptivní nastavování parametrů regulátorů. Rovněž se lze podrobněji věnovat studiu motoru a v návaznosti na to vytvořit jeho model a nahradit stávající zjednodušenou variantu. Po dovybavení spínači koncové polohy je zde potenciál definovat pracovní prostory a demonstrovat tak názorně prvky bezpečnosti práce.

Předmět Prostředky automatického řízení nebo jiný předmět věnující se automatizaci je možno rozšířit o partii programování PLC, které se poněkud odlišuje od klasického programování a otázka efektivního návrhu algoritmu je zde důležitá. Otestováním programu na konkrétní úloze pak studenti získají vlastní zkušenost s významným prvkem současné automatizační techniky.

A pokud jde o další budoucnost úlohy? Možnosti polohovacího modulu nebyly v této práci zdaleka vyčerpány. Vedle polohování řízeného CPU lze vyvíjet řídicí algoritmy s využitím regulátorů, a tak navázat na zkušenosti získané ze simulačního modelu. Polohovací modul navíc také dokáže zpracovat NC programy. Úlohu by bylo také vhodné doplnit o systém přímého odměřování, aby se mohly ověřit výsledky ze simulačního modelu a prohloubit tak znalosti chování suportu – dvouhmotového systému.

Zajímavé možnosti určitě také přináší chápání úlohy jako jednoho z automatizovaných technologických pracovišť, které komunikuje s ostatními a které může být zařazeno do systému vzdáleného sběru dat. Nakonec se nabízí využití této laboratorní úlohy, resp. celé laboratoře při alternativních formách výuky jako je např. e-learning.

Literatura

- [1] Cendelín, J.: Historie programovatelných automatů a jejich současné efektivní použití. *Automa*, 9, 2003, č. 6, s. 6-9.
- [2] Diesner, J.: *Využití moderních technických a programových prostředků pro řízení laboratorního procesu*. (Diplomová práce.) Liberec, TUL 2001.
- [3] Hanzálek, Z.: Petriho sítě (3). *Automatizace*, 44, 2001, č. 10, s. 648-650.
- [4] Internet – <http://www3.ad.siemens.de/doconweb>
- [5] Internet – <http://www.festo.com>
- [6] Souček, P.: *Servomechanismy NC strojů a průmyslových robotů*. (Skripta.) Praha, ČVUT 1992.
- [7] Šmejkal, L. - Martinásková, M.: *PLC a automatizace*. Praha, BEN 2001.

Seznam obrázků

2.1	Model pohonu posuvu suportu	10
2.2	Výpočtové schéma suportu	10
2.3	Přechodová charakteristika přenosu X/X_m	12
2.4	Schéma vedení zpětných vazeb modelu	14
2.5	Schéma zapojení modulů – rychlostní servomechanismus (var. A) . .	20
2.6	Schéma zapojení modulů – polohový servomechanismus (var. B) . . .	21
3.1	Varianta A zapsaná jazykem SFC	25
4.1	Schéma simulačního modelu – celek	29
4.2	Schéma simulačního modelu – PLC	30
4.3	Schéma simulačního modelu – Suport	30
4.4	(1) Průběh pohybu suportu při $x = 0,01\text{ m}$, $m = 400\text{ kg}$, $F_z = 125\text{ N}$ v $t = 0,4\text{ s}$	33
4.5	(1) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,01\text{ m}$, $m = 400\text{ kg}$, $F_z = 125\text{ N}$ v $t = 0,4\text{ s}$	33
4.6	(1) Interakce sil během pohybu suportu při $x = 0,01\text{ m}$, $m = 400\text{ kg}$, $F_z = 125\text{ N}$ v $t = 0,4\text{ s}$	34
4.7	(2) Průběh pohybu suportu při $x = 0,25\text{ m}$, $m = 400\text{ kg}$, $F_z = 0\text{ N}$. . .	34
4.8	(2) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,25\text{ m}$, $m = 400\text{ kg}$, $F_z = 0\text{ N}$	35
4.9	(2) Interakce sil během pohybu suportu při $x = 0,25\text{ m}$, $m = 400\text{ kg}$, $F_z = 0\text{ N}$	35
4.10	(3) Průběh pohybu suportu při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$. . .	36
4.11	(3) Průběh odchylky polohy suportu od polohy vynucené motorem při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$	36
4.12	(3) Interakce sil během pohybu suportu při $x = 0,25\text{ m}$, $m = 10\text{ kg}$, $F_z = 0\text{ N}$	37

Přílohy

- [1] Standard motor 1FT5042 (speed-torque diagram)
- [2] Analog value representation in output range $\pm 10\text{ V}$
- [3] Standard function blocks for the FM 357-2 (overview)
- [4] Disketa: simulační model, projekt var. A, projekt var. B

3.1 Speed-torque diagrams

Table 3-3 Standard motor 1FT5042

1FT5042					
Technical data	Code	Units	–0AF71	–□AK71	
Engineering data					
Rated speed	n_{rated}	RPM	3000	6000	
Rated torque	M_{rated} (100 K)	Nm	1.0	0.9	
Rated current	I_{rated}	A	1.1	1.6	
Stall torque	M_0 (60 K)	Nm	0.75	0.75	
Stall torque	M_0 (100 K)	Nm	1.0	1.0	
Stall current	I_0 (100 K)	A	0.8	1.3	
Stall current	I_0 (60 K)	A	1.1	1.7	
Moment of inertia (with brake)	J_{mot}	10^{-4} kgm^2	2.11	2.11	
Moment of inertia (without brake)	J_{mot}	10^{-4} kgm^2	1.73	1.73	
Limit data					
Max. speed	n_{max}	RPM	5500	8300	
Max. torque	M_{max}	Nm	4.0	4.0	
Peak current	I_{max}	A	4.5	7.0	
Limiting torque	M_{limit}	Nm	2.5	1.9	
Physical constants					
Torque constant	k_T	Nm/A	0.95	0.60	
Voltage constant	k_E	V/1000 RPM	115	75	
Winding resistance	R_{ph}	Ohm	28.2	11.8	
Three-phase inductance	L_D	mH	48.4	20.3	
Electrical time constant	T_{el}	ms	1.7	1.7	
Mechanical time constant	T_{mech}	ms	11.0	11.0	
Thermal time constant	T_{th}	min	40	40	
Weight with brake	m	kg	3.5	3.5	
Weight without brake	m	kg	3.2	3.2	

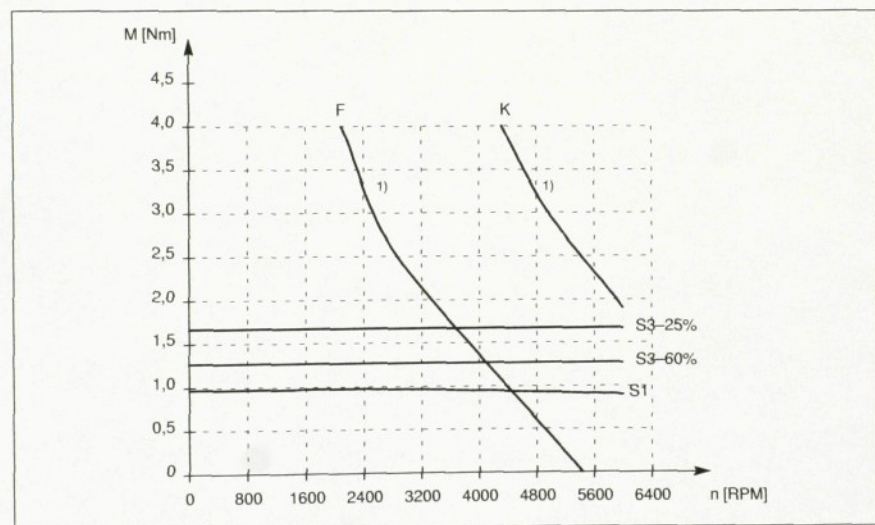


Fig. 3-3 Speed-torque diagram 1FT5042

1) valid for a 600 V DC link voltage

Analog value representation in voltage output ranges

Table 4-32 Analog Value Representation in Output Range ± 10 V

System			Voltage Output Range	
	Dec.	Hex.	± 10 V	
118.5149 %	32767	7FFF	0.00 V	Overflow, off circuit and de-energized
	32512	7F00		
117.589 %	32511	7EFF	11.76 V	Overrange
	27649	6C01		
100 %	27648	6C00	10 V	Rated range
75 %	20736	5100	7.5 V	
0.003617 %	1	1	361.7 μ V	
0 %	0	0	0 V	
	-1	FFFF	-361.7 μ V	
-75 %	-20736	AF00	-7.5 V	
-100 %	-27648	9400	-10 V	
	-27649	93FF		Underrange
-117.593 %	-32512	8100	-11.76 V	Underflow, off circuit and de-energized
	-32513	80FF		
-118.519 %	-32768	8000	0.00 V	

Table 4-33 Analog Value Representation in Output Ranges 0 to 10 V and 1 to 5 V

System			Voltage Output Range		
	Dec.	Hex.	0 to 10 V	1 to 5 V	
118.5149 %	32767	7FFF	0.00 V	0.00 V	Overflow, off circuit and de-energized
	32512	7F00			
117.589 %	32511	7EFF	11.76 V	5.70 V	Overrange
	27649	6C01			
100 %	27648	6C00	10 V	5 V	Rated range
75 %	20736	5100	7.5 V	3.75 V	
0.003617 %	1	1	361.7 μ V	1V+144.7 μ V	
0 %	0	0	0 V	1 V	
	-1	FFFF			Underrange
-25 %	-6912	E500		0 V	
	-6913	E4FF			Not possible. The output value is limited to 0 V.
-117.593 %	-32512	8100			
	-32513	80FF			Underflow, off circuit and de-energized
-118.519 %	-32768	8000	0.00 V	0.00 V	

Table 6-1 Standard function blocks for the FM 357-2 (overview)

Block no.	Block name	Meaning/function	Meaning/function
FC 1 Page 6-17	RUN_UP	Call in OB 100 and OB 86, startup / initialization	required for application, no. can be changed ¹⁾
FC 5 Page 6-19	BF_DIAG	Call in OB 82, FM restart, serious internal FM error	required for application, no. can be changed ¹⁾
FC 22 Page 6-22	BFCT	Call in OB 1, cyclic call (synchronization with FM 357-2) Basic functions and operating modes, interface handling, write requests	required for application, no. can be changed ¹⁾
FB 2 Page 6-24	FM_GET	Call in OB 1, read FM variable according to FM-VAR list (generated with NC-VAR Selector) A user DB is required as an instance DB or a multi-instance DB (for storing the parameters/values of the variables).	Only to be used if "Read variable" function is needed for the application; no. can be changed ¹⁾
FB 3 Page 6-32	FM_PUT	Call in OB 1, write FM variable according to FM-VAR list. A user DB is required as an instance DB or a multi-instance DB (for storing the parameters/values of the variables).	Only to be used if "Write variable" function is needed for the application; no. can be changed ¹⁾
FB 4 Page 6-38	FM_PI	Call in OB 1, "select program", "acknowledge error", "activate MD"; (services stored in DB 15). A user DB is required as an instance DB or a multi-instance DB (for storing the parameters/program name/path).	Only to be used if the function is needed for the application; no. can be changed ¹⁾
DB 15	PI_DI	Internal block, is supplied	Required for operation of FB 6, no. cannot be changed
DB 16	BF_DB16	Internal block, is supplied	Required for application, no. cannot be changed
DB 121 Page 6-45	VAR_ADDR	DB with selected variables, is supplied (generated with NC-VAR Selector)	Only if use of FB 2, FB 3 necessary, no. can be changed ¹⁾
FB 6	BF_FB6	Internal block, is supplied	Only if use of FB 2, FB 3, FB 4 necessary, no. cannot be changed
FC 23	BF_FC23	Internal block, is supplied	Required for application, no. cannot be changed
OB 1	—	Cyclic level	Required for application
OB 82	—	Diagnostic alarm level	Required for application
OB 100	—	Startup level	Required for application
OB 86	—	Subrack failure	for distributed installations
OB 122	—	I/O access error	for distributed installations
Timer 0	—	Assigned internally, cyclic sign-of-life/interface watchdog of FM	—
Timer 1	—	Assigned internally, power-up time-out, internal = 3 min	—

- 1) - **Block no. is default setting**, it is possible to change the block no. in the SIMATIC Manager
 - It is only necessary to change the symbol table entries if symbolic programming is used