ZADÁNÍ PRÁCE

LICENČNÍ SMLOUVA

Abstrakt

Tato práce se zabývá číslicovou regulací polohy svisle otáčivého ramene. Pohonem tohoto ramene je stejnosměrný motor k jehož číslicovému řízení je použit programovatelný automat. Výsledkem práce je několik alternativ řešení lišících se především dosažitelnou kvalitou regulace. V práci je řešeno i uživatelské rozhraní ve formě dotykového displeje.

Abstract

This graduation theses deal (diploma paper, thesis) with computer control of the vertically rotative arm. The actuator of this arm is a direct current motor for its numeric control is used a programmable logic controller. The produce is several alternatives of a resolution varying in available quality at first. In this graduation theses is solved a user interface in the form of touch screen.

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu diplomové práce Ing. Zdeňku Němcovi, CSc., za odborné vedení, cenné rady a připomínky při vypracování této diplomové práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, dle pokynů vedoucího diplomové práce. Všechny podklady, ze kterých jsem čerpal, jsou řádně uvedeny v seznamu použité literatury.

V Brně dne 29. 5. 2009

……...

podpis

Obsah

1. Úvod

Tato práce se zabývá tématikou číslicové regulace polohy otáčivého ramene. Rameno je pevně spojeno s hřídelí stejnosměrného motoru a je uzpůsobeno k nesení závaží volitelné hodnoty. Uvedená soustava spolu se snímačem otáček je hlavní regulovanou soustavou, kterou budeme za použití programovatelného automatu firmy Siemens číslicově řídit. Výsledkem této práce je návrh a realizace několika alternativ řešení tohoto problému a bude sloužit jako pomůcka při výuce v laboratoři řídící techniky FSI VUT v Brně.

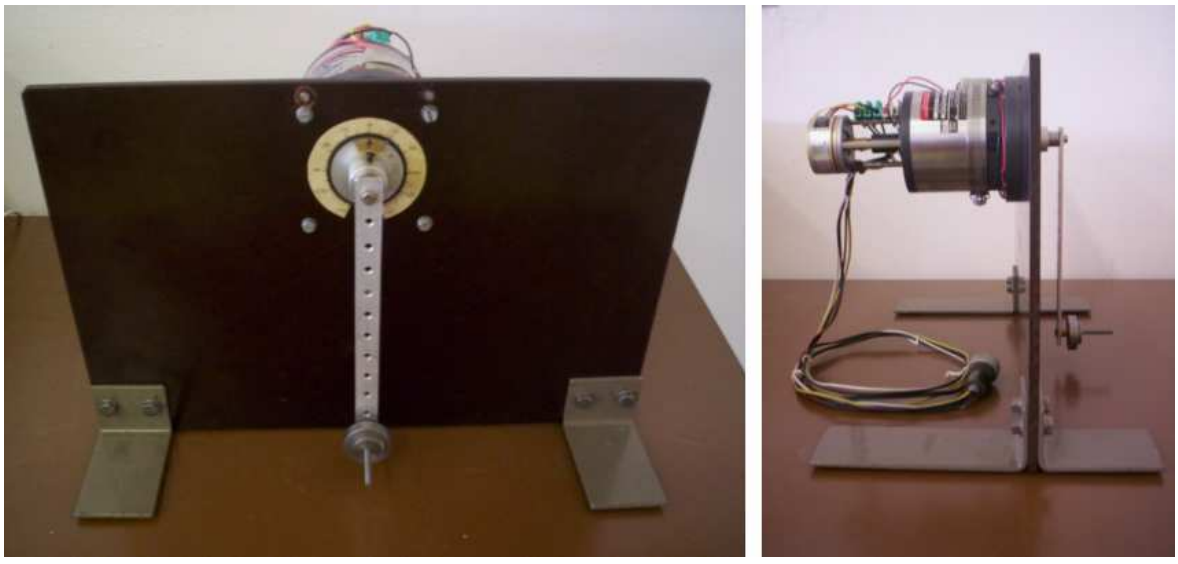
V technické praxi je nasazení programovatelných automatů v oblasti automatizace hojně využíváno a problematika regulace pohonů je velice častá z nichž nejobtížnější je právě řešená regulace polohy. Hlavním problémem je proměnlivost momentu setrvačnosti zátěže a z tohoto plynoucí nutnost adaptivního řízení. Jako pohony jsou nejčastěji využívány stejnosměrné motory a dále také střídavé asynchronní motory s kotvou nakrátko. Tyto se ovšem častěji řídí pomocí frekvenčních měničů, které ovšem nejsou problematikou této práce.

V první kapitole se seznámíme s regulovanou soustavou a s jejím příslušenstvím. Podíváme se na panel analogové regulace [Šarbort 2002], z něhož nově využijeme některých z jeho obvodů. V kapitole další se seznámíme s číslicovými prostředky pro regulaci polohy ramene a s vývojovým prostředím pro jejich programové aplikace. V dalších kapitolách budeme postupně navrhovat řešení pro zadaný problém. Vždy bude popsáno navrhnuté řešení a k němu bude vytvořen patřičný podprogram. Nakonec budou všechny části programu spojeny v komplexní program, který bude navíc využívat uživatelského rozhraní v podobě dotykového displeje. Závěrem práce bude zhodnocení výsledků různých variant regulace polohy ramene na základě měření.

2. Seznámení s regulovanou soustavou a počáteční měření

2.1 Pohon s ramenem

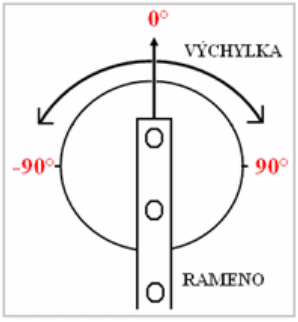
Jelikož se tato práce zabývá regulací polohy svisle otáčivého ramene, je hlavní částí regulované soustavy pohon tohoto ramene. Jedná se o stejnosměrný motor, který má štítkové označení PERMANENT MAGNET SERVO MOTOR – TACH (sériové číslo 05995). Tento motor je mechanicky připevněn ke svislému panelu obdélníkového tvaru. Panel s motorem drží ve svislém směru čtyři konzoly ve tvaru L, které jsou stejně jako motor připevněny k panelu čtyřmi šrouby. Hřídel motoru prostupuje panelem a z čelní strany je k ní připevněno otočné rameno. Toto rameno je z hliníkové slitiny. Délka ramene od osy hřídele k ose šroubu, který nese závaží, je 165mm. Po celé délce ramene jsou otvory, takže tuto délku je možné měnit přemístěním šroubu do jiného otvoru. Této možné změny však nebudeme využívat a hodnota 165mm bude při řešení zadaného problému konstantní. Na hřídeli motoru je umístěna ryska ve tvaru šipky, která ukazuje na stupnici nalepené na svislém panelu. Tato stupnice je v rozsahu od -150˚ po 150˚ a slouží jako orientační údaj o poloze ramene (na obrázku 2.2 můžeme vidět souřadnicový systém této stupnice). Pro přesnější určení polohy ramene nám slouží snímač polohy KINAX 3W2, typ 708 No.0-Serie/4/2. Tento snímač je k hřídeli připevněn ocelovou spojkou. Jeho napájecí napětí je do 30V a odebíraný proud je 160mA. Snímaný rozsah je dle výrobce 0˚ až 270˚ a jeho výstup by měl být v rozsahu 0 až 5mA. Celkové rozložení částí přípravku je patrné z obrázku 2.1.



Obr. 2.1 Mechanické uspořádání přípravku

2.2 Měření motoru

Před veškerým měřením je nutné uvést souřadnicový systém, který odpovídá ocejchování natočení ramene na přípravku. Souřadnicový systém při pohledu na čelo přípravku je uveden na obrázku 2.2.



Obr. 2.2 Souřadnicový systém natočení ramene

Na motoru bylo provedeno měření závislosti kroutícího momentu na proudu. K měření jsme použili stejnosměrný zdroj napětí, který sloužil k napájení motoru. Dále ampérmetr k měření proudu a laboratorní váhy k určení tíhy, kterou vyvíjí rameno při daném proudu. Rameno bylo položeno na „misku“ vah v poloze přesně 90° ve směru kladném, kde bylo provedeno měření v několika měřících bodech. Poté ve směru záporném, kde měření probíhalo v týž bodech ovšem s opačnou polaritou proudu. Jednotlivé naměřené hodnoty pro oba směry orientace kroutícího momentu, respektive proudu, jsou v tabulce 2.1.

kladný směr

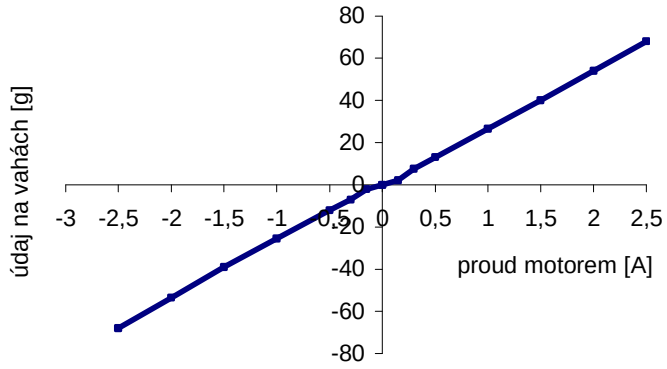
proud do motoru [A]	0	0,15	0,3	0,5	1	1,5	2	2,5
údaj na vahách [g]	0	2	7,5	13	26,5	40	54	68

záporný směr

proud do motoru [A]	0	-0,15	-0,3	-0,5	-1	-1,5	-2	-2,5
údaj na vahách [g]	0	-2	-7	-12	-25,5	-39	-53,5	-68

Tab. 2.1 Naměřené hodnoty proudu a údaje na vahách pro kladný a záporný směr

Při nulovém proudu motorem byl údaj na vahách 15 gramů. Tento údaj odpovídá tíze ramene, která byla zaznamenána laboratorními váhami. Pro určení závislosti kroutícího momentu samotného motoru na proudu bylo nutné tuto hodnotu pro jednotlivé měřící body odečíst. Hodnoty v tabulce odpovídají již korigovaným údajům, tj. údaje odpovídají čistě hodnotám, které vyvíjel samotný motor. Grafická závislost kroutícího momentu na proudu motorem je patrná z obrázku 2.3.



Obr. 2.3 Závislost kroutícího momentu na proudu motorem

Jak je z grafu 2.3 patrné, průběhy v kladném i záporném směru jsou kolem osy x symetrické. Závislost ovšem není ideálně lineární, ale obsahuje jistou nelinearitu, která je zřetelná převážně v oblasti okolí nuly. *Tato nelinearita je způsobena mimo jiné třením, které bude v našem případě jedním z hlavních problémů k dosažení kvalitní regulace.*

2.3 Měření na snímači polohy

Snímač polohy má dle výrobce snímat natočení v rozsahu 0˚ až 270 a jeho výstup by měl být v rozsahu 0 až 5mA. Má ovšem dva stavitelné prvky ve formě trimrů označených jako ZERO a SPAIN, které slouží k jistému nastavení nuly a zesílení. Vzhledem k již v minulosti provedeným zásahům do polohy těchto trimrů bylo nutné provést měření závislosti výstupu snímače na natočení ramene, abychom si ověřili aktuální nastavení snímače.

Snímač má sice proudový výstup, ale byl pro něj vytvořen I/U převodník [Kavan 2003], který byl integrován do ovládacího panelu [Šarbort 2002], popis panelu viz. níže. Tento převodník mění proudový rozsah z 0mA až 5mA na rozsah napěťový od -5V do +5V. Měřená závislost bude tedy výstupní napětí z převodníku na úhlu natočení ramene. Část naměřených hodnot je uvedeno v tabulce 2.2. Grafická závislost je na obrázku 2.4.

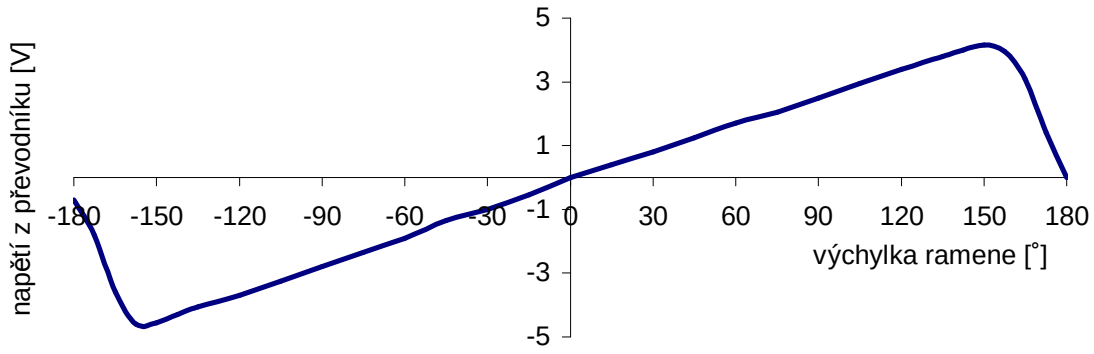
kladný směr

výchylka ramene [˚]	0	30	60	90	120	150	180
výstupní napětí [V]	0	0,8	1,7	2,5	3,4	4,15	0

záporný směr

výchylka ramene [˚]	0	-30	-60	-90	-120	-150	-180
výstupní napětí [V]	0	-1	-1,9	-2,8	-3,7	-4,55	-0,7

Tab. 2.2 Část naměřených hodnot výchylek ramene a jím odpovídající hodnoty napětí



Obr. 2.4 Závislost výstupního napětí snímače natočení ramene

Výstup z převodníku je přibližně symetrický okolo nuly. Nula voltů odpovídá natočení 0˚ a dále se mění podle polarity natočení. Pro naši aplikaci je nejdůležitější oblast, kde snímač přestává být lineární. Jedná se o oblast kde výchylka překračuje hodnotu +150˚ a -150˚. Abychom se vyvarovali nevhodnému chování soustavy při regulaci, musíme dbát přísně na to, abychom se pohybovali vždy v lineární oblasti snímače a nepřekračovali kritickou výchylku. Ke zvětšení možného rozsahu výchylky by pomohl snímač, který má lineární pracovní oblast větší. To ale není předmětem této práce.

Důležitou vlastností každé regulované soustavy jsou její dynamické vlastnosti, vyjádřené časovými konstantami v jejím přenosu. Pomocí tohoto přenosu můžeme mimo jiné určit vhodné parametry pro regulátor dané soustavy. Přenos je v našem případě ovšem silně závislý na závaží, které je připevněno na otočném ramenu. Čím těžší závaží bude připevněno na rameno, tím bude větší hlavní časová konstanta, neboť s rostoucí hmotností roste i setrvačnost hmoty.
Pro určení časové konstanty (která by byla funkcí hmotnosti závaží) respektive přenosu celé soustavy, by jsme mohli použít diferenciální pohybovou rovnici soustavy. V dynamice soustavy ovšem hraje významnou roli tření, které zdaleka není zanedbatelné a matematický popis vystihující jeho chování určit nelze.
Dále by bylo možné pro jednotlivá závaží změřit přechodové charakteristiky soustavy a z ní určit časové konstanty a celkový přenos. Tato metoda by ovšem vyžadovala měření o počtu všech možných kombinací závaží na rameni.

Vdaném případě bude výhodné, při určování vhodných parametrů regulátorů, pro tuto soustavu volit experimentální Ziegler-Nicholsonovu metodu.

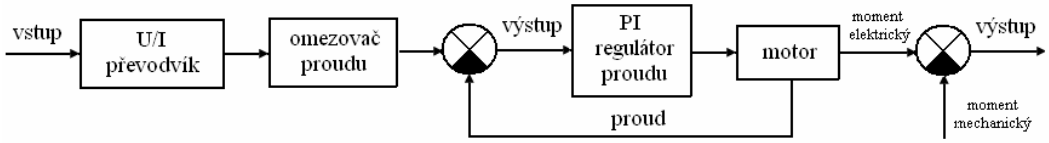
2.4 Panel pro analogovou regulaci

Další částí regulované soustavy je PANEL PRO ANALOGOVOU REGULACI POLOHY RAMENE [Šarbort 2002]. Celkový pohled na tento panel je na obrázku 2.6, kde písmeny A, B a C jsou označeny místa panelu, na které se budeme pomocí tohoto značení později odkazovat. Tento panel slouží k analogové regulaci výše popsané soustavy a v této práci jej budeme nahrazovat číslicovými prostředky. Jelikož ovšem některé jeho části se nám budou hodit i při regulaci číslicové (signál z PLC půjde přes část obvodů tohoto panelu), je nutné na některé jeho části nahlížet jako na další část regulované soustavy.

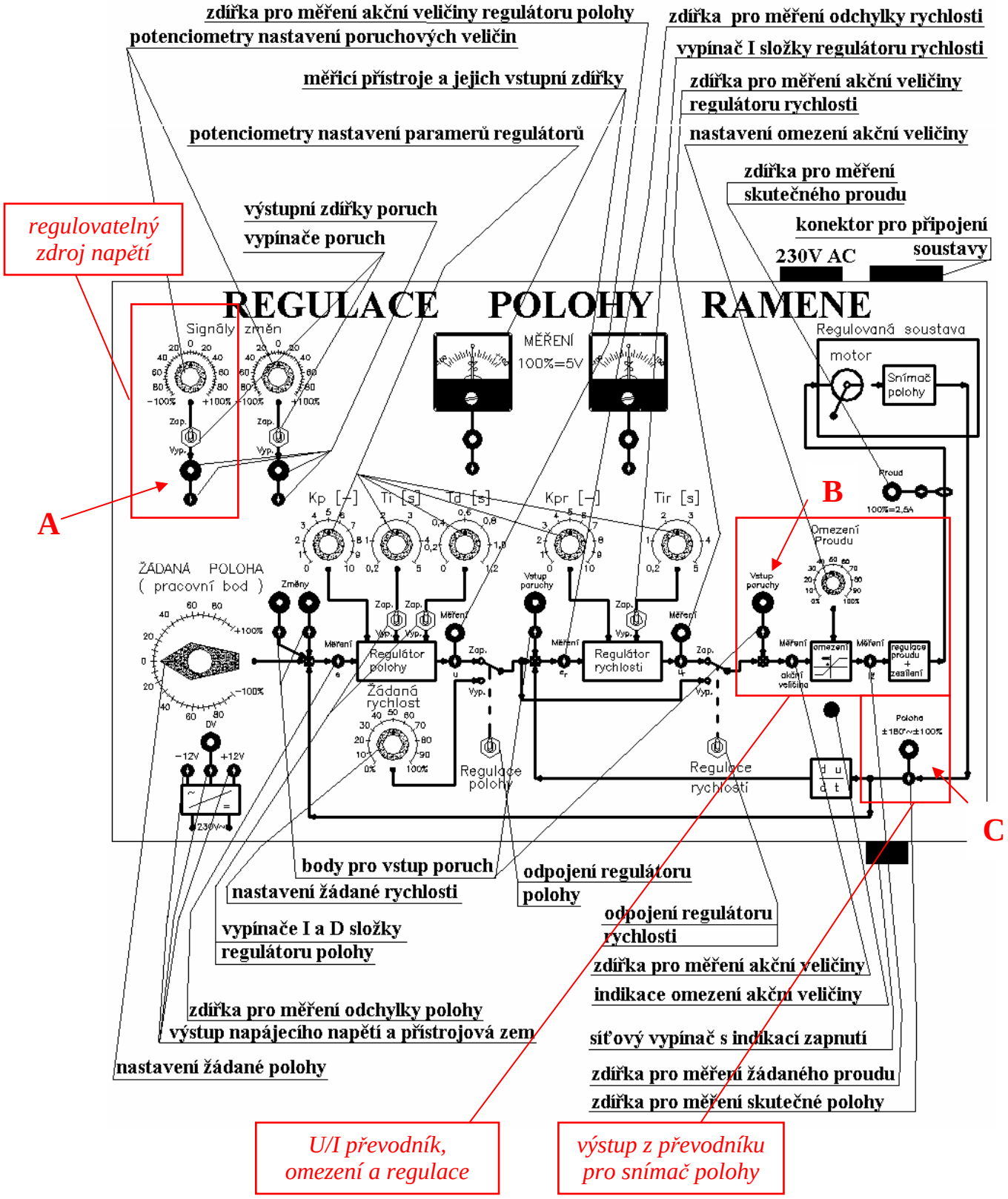
První z využitých bloků panelu je *regulovatelný zdroj napětí v rozsahu od +5V do -5V*. Tento zdroj nepovažujeme za část regulované soustavy, bude nám sloužit pouze jako pomocný obvod. Jelikož je výstup z I/U převodníku také +5V až -5V, bude vhodné rozsahy všech budoucích vstupů do PLC volit v tomto rozsahu. Zdroj je regulovatelný potenciometrem, který je umístěn na čele panelu. Ocejchování potenciometru je od -100% do 100%, jak je vidět na obrázku 2.6. Pro ověření výstupních hodnot toho zdroje bylo provedeno měření v rozsahu 0% až 100% a 0% a -100% v několika měřících bodech. Výstupní napětí je symetrické kolem nuly a má na obě strany od 0% do +-100% lineární průběh, kde 100% odpovídá hodnotě 4,6V, a -100% odpovídá hodnotě -4,6V. U zdroje je též dvoupolohový přepínač, který slouží k vypnutí a zapnutí napětí do výstupní svorky. Tento přepínač bude výhodné využívat pro vytváření skokových změn. Zdroj nám bude sloužit jako signál žádané hodnoty „w(t)“ natočení ramene.

Další obvod, který při číslicové regulaci z panelu využijeme, je U/I převodník. Tento převodník nám bude sloužit k převodu výstupního napětí z PLC na proud. Za převodníkem je dále řazen omezovač proudu s nastavitelným parametrem velikosti omezení. Cejchování u potenciometru nastavení omezení je od 0% do 100%, což odpovídá hodnotě proudu v procentech celého rozsahu, který omezovač „propustí“. Autor panelu [Šarbort 2002] uvádí jako maximální dovolený proud motoru pro dlouhodobý provoz 2,5A. Tato hodnota proudu odpovídá tedy poloze potenciometru na hodnotě 100%, kde omezovač propouští 100% maximálního dovoleného proudu. Za omezovačem je dále regulátor proudu a jeho zesilovač. Celou tuto větev tedy využijeme pro číslicovou regulaci polohy ramene, kde napěťový výstup z PLC bude přiveden na vstup do U/I převodníku, dále přes omezovač proudu na PI regulátor proudu, zesilovač a dále do motoru s ramenem.

Celou tuto soustavu budeme dále ve schématech označovat jako blok regulované soustavy. Ve skutečnosti bude tímto jedním blokem myšleno schéma, které je na obrázku 2.5.



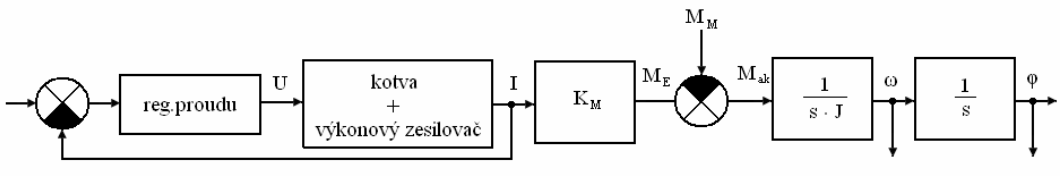
Obr. 2.5 Rozkreslený subsystém regulované soustavy



Obr. 2.6 Ovládací panel [Šarbort 2002] s vyznačenými částmi které byly nově využity pro číslicovou regulaci

2.5 Přenosy soustavy

Regulovanou soustavu lze také zobrazit pomocí bloků dílčích soustav, jak je tomu na obrázku 2.7. Blok K_M značí konstantu motoru, M_E elektrický moment, M_M mechanický moment a $M_{ak,}$ který je rozdílem těchto dvou, je momentem akceleračním.



Obr. 2.7 Dílčí blokové schéma regulované soustavy

Zde si uvedeme jednotlivé důležité přenosy soustav. První přenos bude pro regulační smyčku proudu, která je realizována pomocí panelu analogové regulace.

$$G_{SI}(s) = \frac{I_M(s)}{U_{\check{R}}(s)} = \frac{K_{IU}}{1+T_{SI}} \tag{2.1}$$

Kde $I_M(s)$ je Laplaceův obraz výstupního proudu do motoru, $U_{\check{R}}(s)$ je Laplaceův obraz vstupního řídícího napětí z regulátoru, K_{IU} je konstanta určující vztah mezi proudem do motoru a řídícím napětím z regulátoru proudu a T_{IS} je časová konstanta závislá převážně na časové konstantě obvodu kotvy motoru.

Další z přenosů bude přenos pro regulační smyčku rychlosti:

$$G_{S\omega}(s) = \frac{\omega(s)}{I(s)} = \frac{K_M}{s \cdot J} \tag{2.2}$$

Kde $\omega(s)$ je Laplaceův obraz výstupní úhlové rychlosti, $I(s)$ je Laplaceův obraz vstupního proudu, K_M je konstanta motoru a J je moment setrvačnosti závislí na hmotnosti závaží.

Poslední z uváděných přenosů bude přenos pro regulační smyčku proudu:

$$G_{S\varphi}(s) = \frac{\varphi(s)}{\omega(s)} = \frac{1}{s} \tag{2.3}$$

Kde φ(s) je Laplaceův obraz vstupní polohy a ω(s) je Laplaceův obraz výstupní úhlové rychlosti. V tomto případě se jedná o astatickou soustavu.

3. Popis prostředků číslicové regulace otáčivého ramene

3.1 Hardwarové prostředky

3.1.1 SIMATIC S7-200 s CPU 226

Jelikož při realizaci číslicové regulace otáčivého ramene jsem musel vycházet z dostupného hardwarového vybavení školy, byl jako prostředek k číslicovému řízení celé soustavy vybrán programovatelný automat vyráběný firmou Siemens SIMATIC S7-200 s CPU 226. Automat disponuje rozsáhlým instrukčním souborem a je vybaven i silnými komunikačními funkcemi. Pro lepší splnění požadavků aplikace má řada S7-200 širokou škálu rozšiřovacích modulů. Těmito rozšiřovacími moduly můžeme do S7-200 přidat další funkce nebo rozšířit počet vstupů a výstupů, čehož v naší aplikaci využijeme. PLC Simatic S7-200 s CPU 226 je vidět na obrázku 3.1, jeho základní specifické informace nalezneme v tab.3.1 [manual S7-200].

Obr. 3.1 PLC Simatic S7-200 s CPU 226

Model CPU	Napájení (jmenovité)	Digitální vstupy	Digitální výstupy	Komunik. porty	Analogové vstupy	Analogové výstupy	Odnímatelná svorkovnice	Rozměry (mm) (š x v x h)	Hmotnost
CPU 226	24 VDC	24 x 24 V DC	16 x 24 V DC	2	Ne	Ne	Ano.	196 x 80 x 62	550 g

Tab.

Tab. 3.1 Základní specifické informace o PLC Simatic S7-200 s CPU 226

3.1.2 rozšiřující modul EM235

Jak je vidět z tabulky specifických informací, automat obsahuje pouze digitální vstupy a výstupy. Pro naši aplikaci budeme ale potřebovat dva vstupy analogové (pro žádanou a skutečnou hodnotu polohy ramene) a jeden analogový výstup (pro akční veličinu). Je tedy nutné k automatu připojit rozšiřující modul. Z dostupných modulů od firmy Siemens byl vyhovující rozšiřující modul EM235. Jeho základní specifické informace jsou: 4 analogové vstupy / 1 analogový výstup, rozměry 71.2mm x 80mm x 62mm a jeho hmotnost 186g. Použitý rozšiřující modul je vidět na obrázku 3.2.

Obr. 3.2 Rozšiřující modul Siemens EM235

Modul má možnost výběru proudového nebo napěťového vstupu. Pro naši aplikaci využijeme možnost vstupu napěťového. Z dostupných rozsahů pro tento vstup nám nejvíce bude vyhovovat rozsah od +10V do -10V. Výstup z modulu je možno volit také jako proudový nebo napěťový. Využijeme opět možnost napěťového výstupu rovněž v rozsahu od +10V do -10V (napěťový rozsah od +5V do +5V, který by se nám více hodil pro vstup do U/I převodníku bohužel není k dispozici).

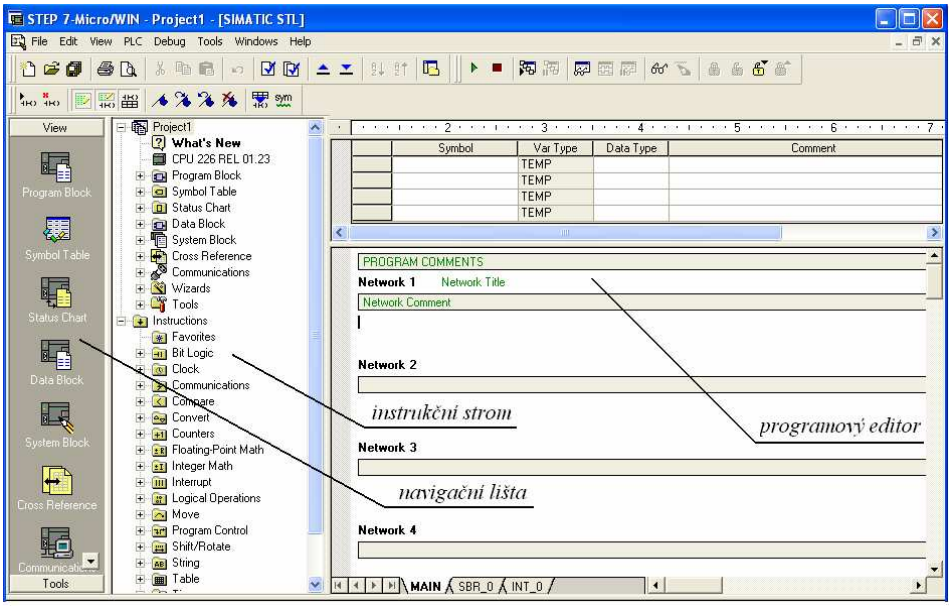
3.1.3 napájecí zdroj Siemens LOGO! Power

PLC i rozšiřující modul mají napájecí napětí 24V. K jejich napájení byl určen napájecí zdroj rovněž od firmy Siemens. Jedná se o napájecí zdroj LOGO! Power, který má výstup 24V/2,5A, což bude pro naši aplikaci plně dostačující. Zdroj má proudové omezení 2,8A a ochranu proti zkratu. Jeho rozměry jsou 55mm x 90mm x 55mm a jeho hmotnost 300g.

Obr. 3.3 Napájecí zdroj Siemens LOGO! Power

3.2 Programovací prostředí STEP7 – Micro/WIN

Pro vytváření, editaci a monitorování programu nutného k řízení aplikace, poskytuje firma Siemens programovací balík STEP7 – Micro/WIN. V tomto programovacím prostředí bude vyvinut program pro PLC, který bude sloužit k regulaci dané soustavy. Na obrázku 3.4 je pohled na interface tohoto prostředí.



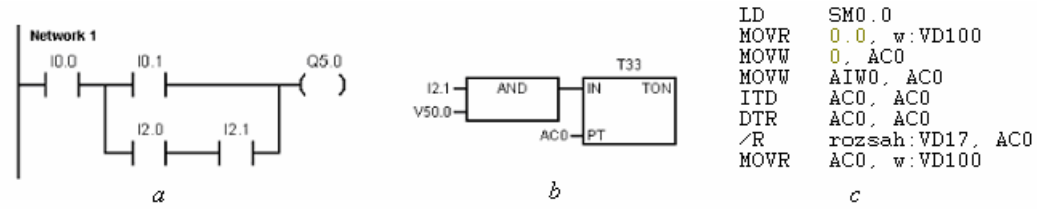
Obr. 3.4 Pracovní prostředí STEP7 – Micro/WIN

- *nástrojové lišty* obsahují tlačítka pro zkratky k často používaným příkazům menu
- *navigační lišta* nabízí ikony pro přístup k různým programovacím prvkům
- *strom s instrukcemi* zobrazuje všechny objekty projektu a instrukce pro tvorbu řídicího programu
- *programový editor* obsahuje program a tabulku lokálních proměnných, ve které můžeme přiřadit symbolické názvy dočasným lokálním proměnným
- *podprogramy a přerušení* jsou zobrazeny jako záložky ve spodní části okna programového editoru

Toto prostředí umožňuje výběr ze tří programových editorů.
První z nich je Editor LAD. Ten zobrazuje program v grafické formě podobné schématům. Programy v kontaktním schématu umožňují simulovat tok elektrického proudu z napájecího zdroje přes řadu logických vstupních podmínek, které následně aktivují výstupní logické podmínky. Program LAD obsahuje levou napájecí lištu, která je pod napětím. Kontakty, které jsou sepnuté, umožňují tok energie do dalšího prvku; kontakty, které jsou rozepnuté, tok energie blokují.

Druhým z editorů je Editor FBD. Tento zobrazuje program v grafické formě, která připomíná běžná logická schémata.Neobsahuje kontakty ani cívky, které se nalézají v editoru LAD, ale ekvivalentní instrukce, které se objevují jako blokové instrukce.

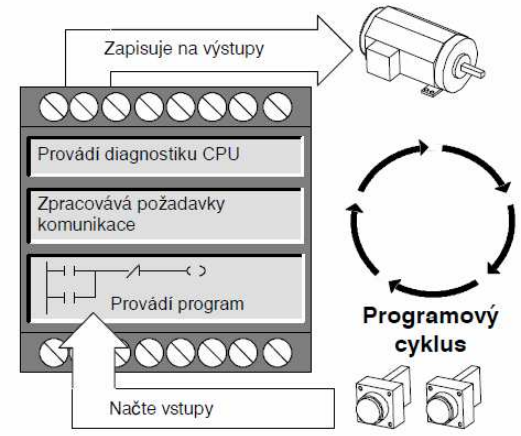
Poslední z nabízených editorů je Editor STL. Tento zobrazuje program jako znakově orientovaný programovací jazyk. Umožňuje vytvářet řídicí programy vkládáním textových instrukcí. Editor STL také umožňuje tvorbu programů, které by pomocí výše popsaných editorů nešly vytvořit. Je to proto, že v STL programujete v jazyku S7-200 a nikoli v jazyku grafického editoru, kde platí určitá omezení, aby byly diagramy správně nakresleny. Tato znakově orientovaná koncepce je velmi podobná programování ve strojovém kódu.



Obr. 3.5 Příklady programu v editorech: a – editor LAD, b – editor FBD, c – editor STL

Jelikož editory LAD a FBD jsou určeny spíše pro začátečníky a editor STL někdy umožní řešit problémy, které se nedají snadno řešit pomocí editoru LAD nebo FBD, zvolil jsem pro vývoj zadané aplikace editor STL.

Před začátkem tvorby programu je nezbytné si uvědomit, jak S7-200 pracuje. Na obrázku 3.6 [manual S7-200] je naznačeno, jakým způsobem automat provádí programový cyklus.



Obr. 3.6 Programový cyklus automatu S7-200

Nejprve načte stav vstupů a tyto hodnoty uloží do tzv. registru obrazu vstupu. Poté provede instrukce programu, který má ve své paměti a zpracuje případné požadavky komunikace. Dále provádí vlastní autodiagnostiku, ve které zkoumá, zda paměť a jeho rozšiřující

moduly pracují správě. Nakonec zapíše na fyzické výstupy hodnoty z registru výstupů. Tuto činnost provádí cyklicky po celou dobu, kdy je v režimu RUN.

Během programového cyklu začíná program vykonávat instrukce od první až po poslední. Pokud jsou v programu použity přerušení, jsou podprogramy přerušení uložené jako součást programu. Podprogramy přerušení se neprovádí jako součást běžného programového cyklu, ale provedou se, je-li splněna podmínka pro dané přerušení (což může nastat v libovolném okamžik programového cyklu).

Při práci s daty lze volit mezi třemi možnými velikostmi místa v paměti. Jedná se o rozsahy Byte „B“, Word „W“ a Double word „D“. V tabulce 3.2 [manual S7-200] jsou uvedeny rozsahy v desítkové a šestnáctkové soustavě pro dané velikosti dat.

Reprezentace	Byte (B)	Word (W)	Double word (D)
Unsigned integer	0 až 255 0 až FF	0 až 65 535 0 až FFFF	0 až 4 294 967 295 0 až FFFF FFFF
Signed integer	−128 až +127 80 až 7F	−32 768 až +32 767 8000 až 7FFF	−2 147 483 648 až +2 147 483 647 8000 0000 až 7FFF FFFF
Real IEEE 32bitová pohyblivá řádová čárka	*Nelze*	*Nelze*	+1.175495E−38 až +3.402823E+38 (kladné) −1.175495E−38 až −3.402823E+38 (záporné)

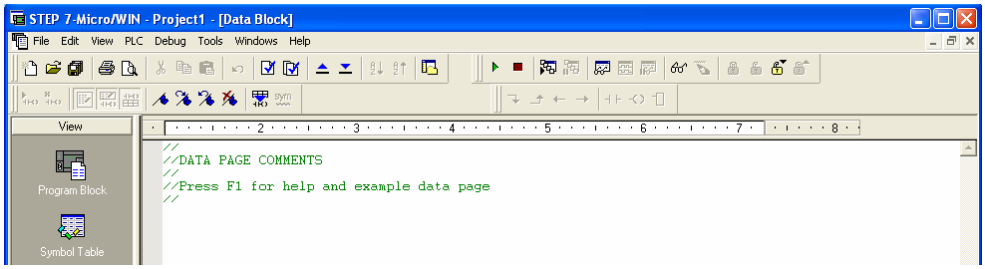
Tab. 3.2 Desítkové a šestnáctkové rozsahy pro různé velikosti dat

Automat obsahuje řadu paměťových oblastí. Pro nás bude nejdůležitější paměťová oblast „V“, kterou lze využívat pro ukládání mezivýsledků i pro ukládání ostatních dat souvisejících s programem. Dále budeme využívat akumulátory. Ty se používají na čtení a zápis podobně jako paměť nebo pro předávání parametrů do podprogramů a nebo pro ukládání mezivýsledků. Lze využívat čtyři 32bitové akumulátory AC0, AC1, AC2 a AC3. Pro výměnu informací mezi CPU a programem slouží tzv. SM bity. Tyto bity se používají pro nastavení a řízení některých speciálních funkcí CPU S7-200. Například jde o bit prvního programového cyklu, který budeme využívat pro volání podprogramu, který se provede pouze v prvním cyklu programu a provádí počáteční úkony a nastavení.

Také je důležité si uvést, jakým způsobem zpracovává automat údaje na svých analogových vstupech a výstupech. Analogové vstupy automat převádí na digitální hodnoty o délce „W“ tedy 16 bitů. Přístup k těmto hodnotám je přes identifikátor AI s následným číslem bytu. Jelikož je velikost dat 2 byty, jsou tyto číselné hodnoty vždy sudé (AI0, AI2 …). Hodnoty na těchto adresách se dají pouze číst. Analogové výstupy mají identifikátor AQ a jejich velikost je analogicky 2 byty (mají tedy také označení bytu pouze sudou hodnotou). Hodnoty na danou adresu jdou pouze zapisovat. Digitální 16bitovou hodnotu automat převádí na proud nebo napětí úměrné její velikosti.

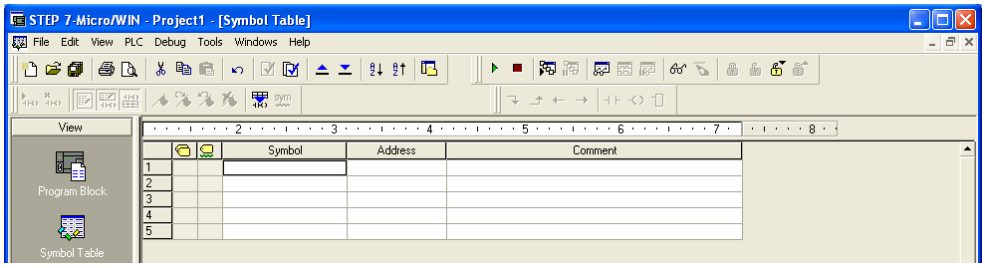
Program se skládá ze tří základních prvků. Z hlavního programu, dalších podprogramů a přerušení. Hlavní program obsahuje instrukce, které řídí aplikaci. S7-200 vykonává tyto instrukce postupně za sebou v časovém sledu jednou za programový cyklus. Podprogramy se provádějí pouze tehdy, jsou-li volány. Podprogramy jsou užitečné v případě, kdy chcete nějakou funkci provádět opakovaně. Poslední z prvků, tzv. podprogramy přerušení, reagují na konkrétní události přerušení. V našem případě budeme podprogramy přerušení volat v opakovaných časových sledech, které budou odpovídat vzorkovací periodě.

Při vytváření aplikace se pracuje s proměnnými, které je možno před spuštěním programu jistým způsobem předem nadefinovat. Přiřazení adres a počátečních hodnot se provádí v editoru datového bloku. Tento umožňuje provést přiřazení počátečních hodnot, ale pouze pro paměť V.



Obr. 3.7 Editor datového bloku

Dále je vhodné používat tabulku symbolů pro symbolické adresování těchto i jiných proměnných. Tato tabulka umožňuje definovat a editovat symboly, ke kterým je odkudkoliv z programu možné přistupovat pomocí symbolických názvů. Tabulek symbolů je možné vytvořit více, například pro různé části programu, kde se proměnné navzájem tématicky liší.



Obr. 3.8 Tabulka symbolů pro symbolické adresování

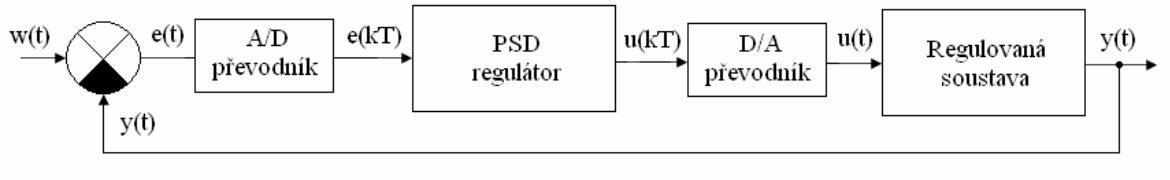
Mimo jiné také toto programovací prostředí obsahuje nástroj pro komunikaci s automatem, kde je možné za běhu programu nahlížet na hodnoty jednotlivých proměnných a také zobrazovat jejich průběhy. Této možnosti lze využít například při odlaďování jednotlivých částí programu a při kontrole správnosti jejich funkce.

Toto by byly základní informace, jak automat zpracovává program, jakým způsobem přistupovat ke vstupním datům, jakým způsobem nastavovat data výstupní a také s jakými formáty velikosti paměti lze pracovat. Nyní již můžeme začít vytvářet aplikaci pomocí jednotlivých instrukcí, které se zapisují do programového editoru. S výhodou využijeme i výše zmiňovaný editor datového bloku spolu s tabulkami symbolů pro symbolické adresování.

4. Realizace PSD regulace polohy ramene a řešení programu automatu

4.1 Řešení programu

Základní a nejjednodušší variantou pro řízení polohy otáčivého ramene bude regulační smyčka s jedním PSD regulátorem. Tento regulátor bude realizován pomocí zvoleného PLC a bude regulovat polohu ramene. Na jeho vstup bude přivedena regulační odchylka *e(kT)*, která bude vypočítána jako rozdíl žádané hodnoty *w(kT)* a skutečné hodnoty *y(kT)*. Z této odchylky pomocí PSD algoritmu vypočítáme hodnotu akčního zásahu *u(kT)*.



Obr. 4.1 Schéma regulační smyčky s PSD regulátorem polohy ramene

Jelikož se jedná o číslicovou regulaci a regulovaná soustava spolu s žádanou hodnotou jsou reprezentovány spojitým signálem, je potřeba vstupní signál nejprve „vzorkovat“ pomocí A/D převodníku a po výpočtu hodnoty akčního zásahu tento signál nutné zase „natvarovat“ do spojité podoby pomocí D/A převodníku, jak je vidět na schématu regulační smyčky. Proces vzorkovaní budeme provádět pomocí cyklického spouštění programu v intervalu vzorkovací periody. Proces tvarování bude realizován výstupem automatu, kde aktuální hodnota výstupu zůstane na svorce až do doby dalšího cyklu, který tuto hodnotu nahradí novou.

Nejdůležitějším blokem celé regulační smyčky je PSD regulátor. Stejně jako u PID spojitého regulátoru požadujeme od číslicového regulátoru aby vstupní regulační odchylku zesiloval, integroval a derivoval. Jelikož ovšem vstupní údaj není spojitý, není možné jej integrovat a derivovat. Je nutné přejít k substituci a to místo integrace používat tzv. sumaci a místo derivace tzv. diferenci. Existují dva základní typy algoritmu pro výpočet hodnot akčního zásahu pro číslicové regulátory a to „polohový “ a „přírůstkový“. Po úpravě prvního z nich dostaneme vztah 4.1 který budeme k řešení zadaného problému využívat.

$$u(k) = K_p e(k) + \left[u_i(k-1) + C_i e(k)\right] + C_d\left[e(k) - e(k-1)\right] \tag{4.1}$$

kde $$C_i = K_p \frac{T_{vz}}{T_i} \tag{4.2}$$

a $$C_d = K_p \frac{T_d}{T_{vz}} \tag{4.3}$$

V rovnici (4.1) je u(k) akční veličinou, kterou budeme v každém cyklu programu počítat. Kp značí proporcionální konstantu, která bude volena na základě Ziegler-Nicholsonovi metody stejně jako Ti a Td, které značí integrační a derivační časové konstanty. Za proměnnou e(k) budeme dosazovat regulační odchylku, která bude rozdílem žádané hodnoty w(k) a hodnoty skutečné y(k). Hodnota ui(k-1) bude odpovídat velikosti minulé akční veličiny (pozn. minulou akční veličinou rozumíme akční veličinu vypočtenou v minulém cyklu programu), ale pouze od integrační složky. Analogicky proměnná e(k-1) značí minulou regulační odchylku.

První program pro PSD řízení polohy ramene si probereme trochu podrobněji. Jak již bylo řečeno, základním stavebním kamenem projektu v Step7 jsou tzv. *instrukce*, kterých má PLC celou řadu. K vytváření tohoto projektu budeme používat převážně instrukce matematické. Každá instrukce má až na výjimky dva parametry, za které dosazujeme patřičné proměnné. Operace dané instrukce se provede mezi první a druhou proměnnou a výsledek se uloží do proměnné druhé. Instrukce se provádějí jedna za druhou, od vrchu dolů, jak jsou napsány, případně se volá podprogram nebo se obslouží přerušení.

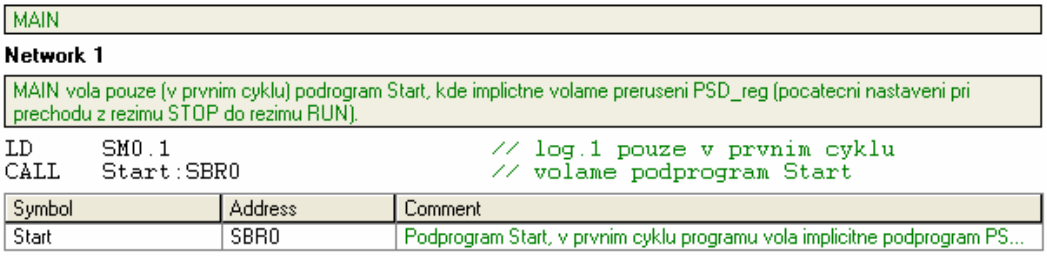
První věcí, kterou začneme, je vytvoření sady globálních proměnných, které budeme v programu potřebovat. Jak již bylo zmíněno v kapitole 3.2, pro globální proměnné je vhodné použít tabulku symbolů a pomocí ní deklarovat proměnné, přiřazovat jim adresy a také si k nim psát krátké komentáře. Na obrázku 4.2 je tabulka symbolů pro symbolické adresování našeho prvního programu. Tato tabulka obsahuje všechny proměnné, které budou v tomto první programu pro PSD regulaci polohy ramene potřeba. V prvním sloupci označeném jako „Symbol“ se nachází symbolický popis proměnné. V dalším sloupci je adresa proměnné a v posledním je místo na krátký popis.

	Symbol	Address	Comment
1	Kp	VD0	proporcionalni konstanta (zesileni)
2	Ti	VD4	casova integracni konstanta
3	Td	VD8	casova derivacni konstanta
4	Tvzms	VB12	vzorkovaci perioda v ms
5	Tvz	VD13	vzorkovaci perioda v s
6	rozsah	VD17	hodnota rozsahu analogovych vstupu
7	w	VD100	zadana hodnota
8	y	VD104	skutecna hodnota
9	ek	VD108	regulacni odchylka e=w-y nynejsi
10	ek_1	VD112	regulacni odchylka minula
11	uk	VD116	akcni zasah nynejsi
12	uik_1	VD120	akcni zasah od integracni slozky minuly
13	Ci	VD124	hodnota integracni slozky
14	Cd	VD128	hodnota derivacni slozky
15	PropSl	VD200	proporcionalni slozka rovnice
16	IntSl	VD204	integrcni slozka rovnice
17	DerSl	VD208	derivacni slozka rovnice

Obr. 4.2 Tabulka proměnných pro první program PSD regulace polohy ramene

Jelikož doba trvání programového cyklu není pro každý cyklus konstantní (záleží např. zda je obsluhován požadavek komunikace či nikoliv), není možné tuto dobu brát jako vzorkovací periodu. Hodnota vzorkovací periody musí být konstantní neboť vystupuje v rovnici pro výpočet akčního zásahu. Proto je nutné v programu volat přerušení, které je možné volat v pravidelných časových intervalech a při každém přerušení počítat aktuální hodnotu akční veličiny. Každý program začíná svůj cyklus podprogramem, který

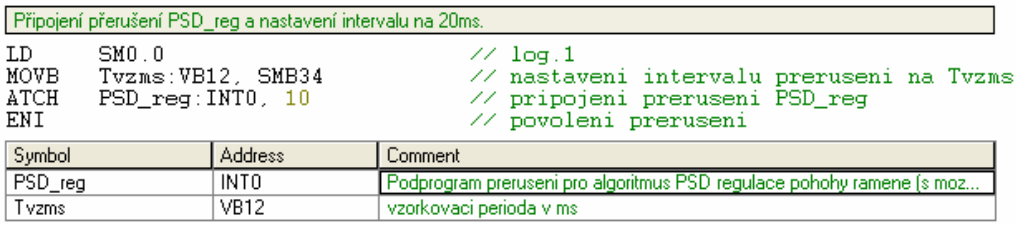
se nazývá MAIN. Jelikož tedy není nutné počítat hodnotu akční veličiny každý programový cyklus, ale pouze při každém přerušení, bude nám MAIN sloužit pouze pro zavolání prvního, námi vytvořeného podprogramu, a to pouze v prvním programovém cyklu. Tohoto lze dosáhnout použitím speciálního paměťového bitu SM0.1, který nabývá logické jedničky pouze v prvním cyklu programu. Dále bude nabývat hodnoty nula a instrukce pod ním se již neprovedou. Jak vypadá MAIN, pro náš první program, je vidět na obrázku 4.3.



Obr. 4.3 Main prvního programu pro PSD regulaci polohy ramene

Instrukce LD načte hodnotu ze speciálního bitu SM0.1 a pokud bude rovna jedné, přijde na řadu další instrukce, která volá podprogram nazvaný „Start“. Jelikož, jak již bylo řečeno, podprogram „Start“ bude volán pouze v prvním cyklu programu, je v něm nutné provést veškeré potřebné počáteční úkony.

Vytvoříme tedy nový podprogram, který si nazveme Start. Nejdůležitější počáteční úkon je nastavení intervalu přerušení. Tento se nastavuje pomocí instrukce pro připojení přerušení ATCH, kde jako první argument instrukce je název přerušení a druhý je v našem případě číselná hodnota „10“, udávající, že se jedná o časové přerušení, kde je hodnota času uložena na adrese SMB34 v milisekundách.



Obr. 4.4 Připojení přerušení PSD_reg

Jak je z obrázku 4.4 zřejmé, nejprve načteme hodnotu bitu na adrese SM0.0. Hodnota tohoto bitu je vždy rovna jedné a instrukce pod ním se tedy provedou vždy. Dále následuje instrukce MOV, která přesune hodnotu z proměnné Tvzms na místo s adresou SM34. Na této adrese se nachází údaj o časovém intervalu přerušení. V našem případě tedy přerušení bude v intervalu, který bude odpovídat uložené hodnotě v proměnné Tvzms. Na dalším řádku je výše zmiňovaná instrukce ATCH pro připojení přerušení, konkrétně přerušení nazvané PSD_reg. Na posledním řádku je instrukce ENI, která slouží k povolení přerušení.

Dále je třeba provést výpočty konstant Ci a Cd podle rovnice 4.2 a 4.3. Jelikož tyto hodnoty budou konstantní a bylo by zbytečné je počítat při každém dalším přerušení

znovu. Na obrázku 4.5 je naznačeno, jakým způsobem je prováděn výpočet rovnice 4.1 v prostředí Step7.

Vypocet integracni konstanty pro PSD regulator Ci = Kp * Tvz/Ti:

```
LD      SM0.0                   // log.1
MOVR    0.0, Ci:VD124           // vynulejeme promennou Ci
MOVR    Tvz:VD13, Ci:VD124      // Tvz do Ci
/R      Ti:VD4, Ci:VD124        // Tvz/Ti
*R      Kp:VD0, Ci:VD124        // Ci = Kp * Tvz/Ti
```

Symbol	Address	Comment
Ci	VD124	hodnota integracni slozky
Kp	VD0	proporcionalni konstanta (zesileni)
Ti	VD4	casova integracni konstanta
Tvz	VD13	vzorkovaci perioda v s

Obr. 4.5 Vypočet integrační konstanty

Nejprve se načte bit s hodnotou logická 1. Potom se proměnná Ci „vynuluje“ a dále do ní vložíme hodnotu proměnné Tvz, která obsahuje vzorkovací periodu v sekundách. Na dalších řádcích počítáme pomocí instrukcí pro dělení a násobení hodnotu integrační konstanty podle vzorce 4.2. Hodnotu derivační konstanty počítáme obdobným způsobem, kde se ovšem při výpočtu držíme vzorce 4.3.

Abychom využili výhod editoru datového bloku, který je popsán v kapitole 3.2, nadefinujeme si v něm počáteční hodnoty proměnných Kp, Ti a Td, pro které bude nutno později určit hodnoty optimální. Prozatím jim budou přiřazeny hodnoty pouze smyšlené. Protože tyto hodnoty proměnných budou nadefinovány pouze na jednom místě a to právě v editoru datového bloku, po určení jejich optimálních hodnot bude jednoduché tyto hodnoty upravit. V editoru jsou také nastaveny hodnoty proměnných „Tvz“ (hodnota vzorkovací periody v sekundách), „Tvzms“ (hodnota vzorkovací periody v milisekundách) a proměnné „rozsah“ (hodnota udávající číselný rozsah vstupů a výstupů PLC). Dále jsou nastaveny na nulu proměnné reprezentující akční zásah, minulý akční zásah od integrační složky a regulační odchylka.

```
// NASTAVENI POCATECNICH HODNOT GLOBALNICH PROMENNYCH
// PRO PRVNÍ PROGRAM PSD REGULACE POLOHY RAMENE

// vzorkovaci perioda v ms
Tvzms:VB12 20

// vzorkovaci perioda v s
Tvz:VD13 0.02

// rozsah vstupu a vystupu
rozsah:VD17 32000.0

// proporcialni konstanta
Kp:VD0 1.0

// integracni casova konstanta
Ti:VD4 1.0

// derivacni casova konstanta
Td:VD8 1.0

// akcni zasah
uk:VD116 0.0

// akcni zasah od integracni slozky minuly
uik_1:VD120 0.0

// regulacní odchylka
ek:VD108 0.0
```

Obr. 4.6 Nastavení hodnot proměnných v editoru datového bloku

Podprogram „Start“ je tedy hotový. V prvním cyklu nám vypočítá hodnoty Ci a Cd a připojí nám přerušení PSD_reg, u kterého nastaví interval přerušení na 20ms. Nyní je tedy třeba toto přerušení vytvořit. Přerušení v daných intervalech sejme údaje o žádané hodnotě a skutečné hodnotě natočení ramene na vstupech, vypočítá regulační odchylku, z ní dále pomocí rovnice 4.3 určí akční veličinu, a tu vhodně převedenou na výstupní rozsah pošle na výstup automatu.

V první řadě je tedy nutné sejmout hodnoty ze vstupů automatu. Dále je nutné tyto hodnoty převést na real (datový typ s desetinou čárkou vhodný k „přesným“ výpočtům), neboť obrazy vstupů a výstupů jsou pouze o velikosti 2 byty. Nakonec tyto hodnoty vložit do patřičných proměnných. Hodnotu z prvního vstupu do proměnné „w“ a hodnotu druhého vstupu do proměnné „y“. Na obrázku 4.7 je ukázka této operace pro první vstup. Instrukce ITD slouží ke konverzi z integeru na double integer a instrukce DTR ke konverzi double integeru na real. Jelikož je obraz vstupu číslo nabývající hodnot od -32000 po 32000, je výhodné tento rozsah převést na rozsah od -0,5 do 0,5 s nímž se dále pracuje názorněji (pokud bude žádaná hodnota a skutečná hodnota v rozsahu od +0,5 po -0,5, bude regulační odchylka vzniklá jejich rozdílem v rozsahu od -1 do 1). Ke změně rozsahu slouží řádek s instrukci /R, kde dělíme číslo, které odpovídá žádané hodnotě rozsahem.

```
Vypocet aktualni regulacni odchylky "e" a urceni hodnoty ek_1
LD      SM0.0                   // log.1
MOVR    0.0, w:VD100            // vynulujeme promenne w
MOVW    0, AC0                  // vynulujeme akumulator AC0
MOVW    AIW0, AC0               // do AC0 vlozime obraz vstupu AI0
ITD     AC0, AC0                // konverze integer na double integer
DTR     AC0, AC0                // konverze double integer na real
/R      rozsah:VD17, AC0        // podelime rozsahem (ziskame hodnotu -1 az 1)
MOVR    AC0, w:VD100            // hodnotu vlozime promenne w
```

Obr. 4.7 Sejmutí žádané hodnoty ze vstupu, konverze na real

Druhý vstup je sejmut a zpracován analogicky. Hodnota v rozsahu od -0,5 do 0,5 je uložena do proměnné y. Dále následuje vložení hodnoty e(k) do proměnné e(k-1), protože v proměnné e(k) máme hodnotu z minulého cyklu a ta je v tomto cyklu již e(k-1). Provedením rozdílu mezi hodnotou žádanou a skutečnou získáme regulační odchylku, kterou uložíme do proměnné e(k).

Nyní již máme všechny koeficienty potřebné k dosazení do rovnice 4.1 a k výpočtu akční veličiny.

```
// vypocet integracni slozky
LD      SM0.0                       // log.1
MOVR    IntSl:VD204, uik_1:VD120    // minulou integracni slozku vlozi do uik_1
MOVR    Ci:VD124, IntSl:VD204       // Ci do IntSl
*R      ek:VD108, IntSl:VD204       // Ci * e(k)
+R      uik_1:VD120, IntSl:VD204    // IntSl = [Ci * e(k)] + uik_1
```

Obr. 4.8 Část kódu pro výpočet integrační složky akční veličiny

Na obrázku 4.8 je uvedena část podprogramu, kde se počítá integrační složka akční veličiny. Na druhém řádku obrázku je vidět, jak hodnotu IntSl vkládáme do proměnné uik_1, neboť stejně jako tomu bylo u regulační odchylky, i zde je v proměnné IntSl hodnota z minulého cyklu a ta je nyní pro nás akčním zásahem od integrační složky minulým.

Takto vypočítané jednotlivé složky akčního zásahu (proporcionální složka, integrační složka a derivační složka) jsou sečteny a vloženy do proměnné uk. Předtím je ještě hodnota integrační složky podrobena omezení, pokud je její hodnota větší než 1,05 nebo menší než -1,05. Pokud tomu tak je, nastavíme natvrdo hodnotu +1,05 nebo -1,05 vzhledem k polaritě přesahující hodnoty. Omezení integrační složky děláme pro případ, kdy rameno z nějakého důvodu mělo delší dobu výraznější nenulovou regulační odchylku jedné polarity. Při tomto stavu integrační složka, která je počítaná číslicově narůstala do velkých hodnot (teoreticky omezených pouze rozsahem proměnné), a při následném uvolnění ramene by trvalo značnou dobu, než by tato hodnota klesla zpět (než by se opět „odintegrovala“).

Výslednou hodnotu akčního zásahu je třeba opět převést na celé číslo, které bude možné poslat na výstup automatu. K tomu nám slouží instrukce TRUNC která „ořízne “ desetinné číslo za jeho desetinou čárkou a uloží ho jako celé. Dále je hodnota akčního zásahu porovnávána, zda leží v rozsahu od -16000 po 16000. Pokud ne, vnutíme hodnotu -16000 nebo 16000 podle polarity původního akčního zásahu. Hodnota 16000 je poloviční hodnotou výstupního rozsahu, a to proto, aby napěťový rozsah výstupu z automatu byl korigován na hodnoty od -5V do 5V při nastaveném rozsahu od -10V po 10V. Úpravu provádíme z důvodu U/I převodníku, který následuje za automatem a má vstupní rozsah od -5V do 5V.

```
Prepocet u(k) do vystupniho rozsahu -16000 az 16000 a zapis hodnoty na vystup
LD      SM0.0                       // log.1 tzn.
MOVR    uk:VD116, AC0               // vlozi u(k) do akumulatoru
*R      rozsah:VD17, AC0            // vynasobi hodnotou rozsahu prevod do +-16000
TRUNC   AC0, AC0                    // konverze real na double integer
LDW<=   16000, AC0                  // test zda je AC0 vetsi nez 16000
MOVW    16000, AC0                  // pokud ano vnutime hodnotu 16000
LDW>=   -16000, AC0                 // test zda je AC0 mensi nez -16000
MOVW    -16000, AC0                 // pokud ano vnutime -16000


LD      SM0.0                       // log.1 tzn.
MOVW    AC0, AQW0                   // vypis AC0 na vystup
```

Symbol	Address	Comment
rozsah	VD17	hodnota rozsahu analogovych vstupu
uk	VD116	akcni zasah nynejsi

Obr. 4.9 Konverze a přepočet hodnoty do výstupního rozsahu

Na 4.9 obrázku vidíme část kódu, která „zkracuje“ real na na celé číslo a porovnává výstupní hodnotu. Instrukce MOVW s parametrem umístění AQW0 slouží k uložení hodnoty do tzv.obrazu výstupu, ze kterého automat převádí hodnotu na patřičný výstup.

Tímto je první program hotový. Před nahráním aplikace do automatu je ještě nutné uvést, pro jaký automat, respektive pro jaké CPU, byl projekt vytvořen. V našem případě se jedná o CPU 226. Tento údaj se vyplňuje na vrcholu instrukčního stromu, kde Step7 nabízí výběr z podporovaných CPU. Nyní je již projekt připraven k transportu do PLC. K propojení mezi počítačem a automatem slouží PPI multi-master kabel, který umožňuje komunikaci přes rozhraní RS232. K downloadu programu slouží ikona v nástrojové liště. Pokud je vše správně propojeno a nastaveno (přenosové rychlosti, adresy portů atd.), download programu je již záležitostí několika sekund.

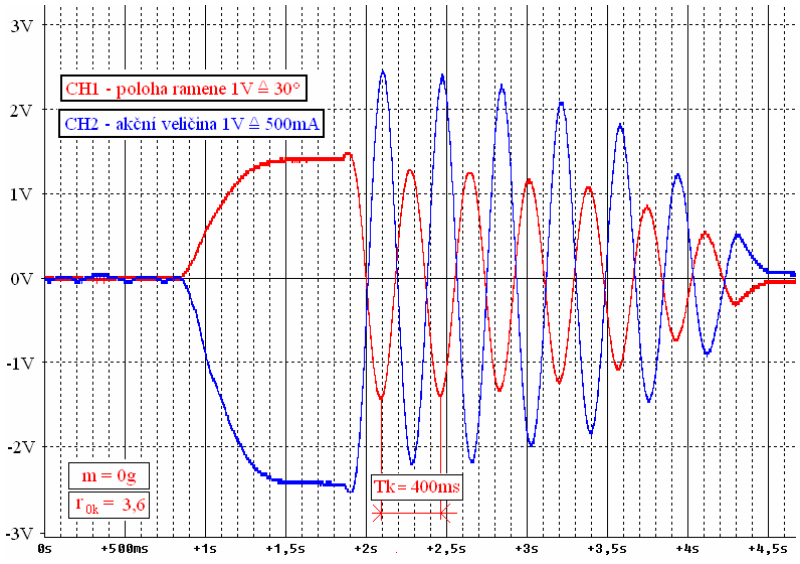
Toto je první a nejjednodušší program pro PSD regulaci natočení ramene (celý zdrojový kód je na přiloženém CD). Program byl odzkoušen na regulované soustavě, kde byl jako vstup žádané hodnoty k automatu připojen signál ze zdířky označené na obrázku 2.6 písmenem A. Dále signál ze zdířky označené na stejném obrázku písmenem C, jako vstup skutečné hodnoty (údaj ze snímače polohy) a napěťový výstup z automatu byl připojen ke zdířce B. Za pomoci funkce „program status“, která slouží k monitorování hodnot proměnných programu za jeho běhu (v režimu RUN), byla zkontrolována správnost polarit a rozsahů jednotlivých proměnných v jednotlivých částech programu. Všechny hodnoty proměnných reagovaly na podněty z regulované soustavy podle očekávání. Aby ale automat mohl soustavu správně řídit, je nutné nastavit parametry PSD regulátoru.

4.2 Nastavení regulátoru

K určení parametrů, které jsou blízké optimálním, využijeme již zmiňovanou experimentální Ziegler-Nicholsonovu metodu. Podstatou této metody je nastavení parametru zesílení regulátoru (při odstavení integrační a derivační složky z provozu) tak, aby se regulovaná soustava dostala na hranici stability. Takové zesílení, jakým dostaneme obvod na hranici stability, se nazývá kritické zesílení a budeme jej značit jako Kpk. Na hranici stability kmitá obvod netlumenými kmity o konstantní amplitudě

a dalším důležitým parametrem ,k určení parametrů blízkým optimálním, je perioda těchto kmitů. Tato perioda se nazývá kritická perioda a budeme ji značit Tk.

Jelikož soustava má proměnné vlastnosti (vlivem přidání nebo odebrání jistého počtu závaží), které jsou závislé na uživateli a tedy svým způsobem náhodné, bylo nutné pro tuto základní a nejjednodušší variantu řízení zvolit parametry, které jsou blízké optimálním pro řízení soustavy bez závaží. Pokud by regulátor byl nastaven na parametry odpovídající jisté hodnotě závaží, stávala by se celá soustava v případě provozu bez tohoto závaží nestabilní. Proto jsem na hranici stability přiváděl soustavu ve stavu, kdy na ramenu nebylo závaží žádné. Prvním krokem Ziegler-Nicholsonovi metody je vyřazení integrační a derivační složky. Toto jsme provedl „vnucením“ nulové hodnoty do proměnných IntSl a DerSl v programu. Postupnou změnou zesílení (respektive změnou hodnoty Kp v programu) jsem soustavu dostal těsně pod hranici stability, kde při vychýlení o 45° z žádané polohy kmitala téměř netlumenými kmity. Průběh polohy ramene při této zkoušce jsem zaznamenal pomocí elektronického záznamníku Hioki.



Obr. 4.10 Průběh polohy ramene a akční veličiny na hranici stability soustavy

Jak je na obrázku 4.10 vidět, kritická doba kmitu Tk je rovna 400ms. Toho jsme dosáhli při hodnotě zesílení Kp=3,6. Proto je tedy hodnota 3,6 pro nás zesílení kritické Kpk. V tabulce 4.1 pro výpočet parametrů regulátoru pro metodu Ziegler-Nicholsonovu jsou vidět vztahy pro určení hodnot parametrů regulátoru blízké optimálním [Švarc 2007].

typ regulátoru	Kp	Ti	Td
P	0,5Kpk	x	x
PI	0,45Kpk	0,83Tk	x
PD	0,4Kpk	x	0,05Tk
PSD	0,6Kpk	0,5Tk	0,12Tk
S	x	2Tk	x

Tab. 4.1 Vztahy pro výpočet parametrů regulátorů

Podle vztahů v této tabulce nám tedy parametry PSD regulátoru polohy ramene vychází následovně:

- *Zesílení Kp=0,6Kpk, tedy Kp=0,6*3,6=* ***2,16 [-]***

- *Časová integrační konstanta Ti=0,5Tk, tedy 0,5*0,4=* ***0,2 [s]***

- *Časová derivační konstanta Td=0,12Tk tedy 0,12*0,4=* ***0,048 [s]***

Těmito hodnotami patřičně upravíme proměnné našeho programu v editoru datového bloku. Po této úpravě je již program hotový a je schopen regulovat polohu natočení ramene.

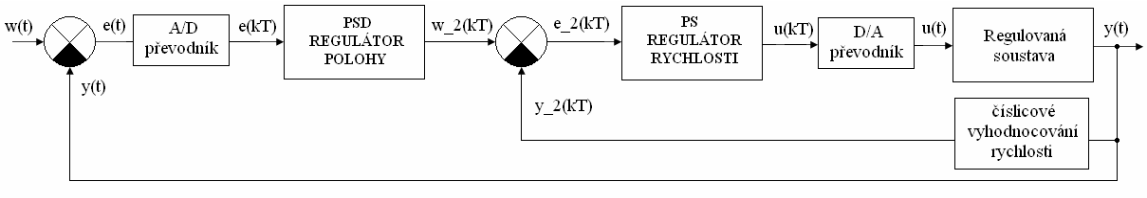
```
// proporcialni konstanta
Kp:VD0 2.16

// integracni casova konstanta
Ti:VD4 0.2

// derivacni casova konstanta
Td:VD8 0.048
```

Obr. 4.11 Upravené hodnoty v editoru datového bloku

5. Realizace druhé regulační smyčky s PS regulátorem polohy ramene a řešení programu automatu

5.1 Řešení programu

Jelikož zadáním této práce je vytvořit několik alternativ řešení, lišících se dosažitelnou kvalitou regulace, je nutné vytvořit program další, který by měl mít výsledky při regulaci ramene lepší. Návrh nové regulační smyčky je na obrázku 5.1.



Obr. 5.1 Schéma regulační smyčky s PSD regulátorem polohy a PS regulátorem rychlosti

Do původní regulační smyčky je v této navíc vložen další regulátor, který bude regulovat rychlost ramene. Jeho vstupem bude výstup z prvního regulátoru polohy. Tedy akční zásah první regulátoru bude odpovídat žádané hodnotě rychlosti. Hodnotu skutečnou bude třeba dopočítávat z hodnoty skutečné polohy. Jelikož se jedná o číslicové zpracování, nebude možné údaj o poloze derivovat, jak by se nabízelo u spojitého systému, kde derivací polohy získáme hodnotu rychlosti. Opět budeme muset přejít k tzv. diferenci, kde ze dvou hodnot rychlosti, které budou od sebe vzdáleny pouze o krátký časový úsek, určíme jejich rozdíl a tento rozdíl podělíme časovým úsekem. Tímto algoritmem získáme hodnotu rychlosti ramene.

Při řešení programu budeme vycházet z již vytvořeného prvního programu, který pouze upravíme a rozšíříme o PS regulátor rychlosti. K proměnným z minulého programu, které jsou vidět na obrázku 4.2, přibudou proměnné další, které uvádím jako tabulku symbolických proměnných na obrázku 5.2.

	Symbol	Address	Comment
1	Kp2	VD300	proporcionalni casova konstanta (zesileni) pro PS reg.
2	Ti2	VD304	casova integracni konstanta pro PS reg.
3	w2	VD400	zadana hodnota pro PS reg.
4	y_1	VD404	skutecna hodnota minula
5	y2	VD408	skutecna hodnota
6	ek2	VD412	regulacni odchylka pro PS reg.
7	uk2	VD416	akcni zasah 2 - vystup z PS reg.
8	uik_12	VD420	akcni zasah (integracni slozky) minuly
9	Ci2	VD424	hodnota integracni slozky pro PS reg.
10	PropSI2	VD500	proporcionalni slozka rovnice pro PS reg.
11	IntSI2	VD504	integracni slozka rovnice pro PS reg.

Obr. 5.2 Tabulka proměnných potřebných k rozšíření prvního programu

Jelikož se nyní jedná o PS regulátor, který derivační složku neobsahuje, bude nutné podprogram „Start“ doplnit o výpočet integrační konstanty pro tento regulátor. Hodnota

integrační konstanty se počítá opět podle vzorce 4.2, ovšem hodnoty Ti2 (integrační časová konstanta pro PS regulátor) a Kp2 (zesílení pro PS regulátor) se budou lišit od konstant pro PSD regulátor polohy. Navíc, jelikož je za PSD regulátorem řazen další PS regulátor, není možné ponechat nastavení konstant určených metodou Ziegler-Nicholsonovu pro minulý program. Tyto hodnoty jsou blízké optimálním pro samotný PSD regulátoru polohy. V tomto případě bude nutné po naprogramování algoritmu provést nový experiment, kde budeme zjišťovat nové hodnoty blízké optimálním pro oba regulátory. Doplněním o výpočet integrační konstanty pro PS regulátor je úprava podprogramu Start hotová. Další korekce budeme provádět na podprogramu přerušení, který je nutné nejprve přejmenovat z PSD_reg na PSD_PS_reg. Začátek podprogramu přerušení bude stejný. Nejprve načteme hodnoty vstupů, potom z nich určíme regulační odchylku. Regulační odchylku spolu s dalšími koeficienty dosadíme do rovnice 4.1 a určíme akční zásah od PSD regulátoru polohy, jehož hodnota bude pro náš druhý PS regulátor rychlosti hodnotou žádanou. Tato hodnota by měla ležet v rozsahu od -1 do +1. Pro jistotu ale ještě doplníme porovnání hodnoty, zda se opravdu v tomto rozsahu nachází, a pokud ne, přiřadíme jí hodnotu -1 nebo +1 podle polarity její hodnoty.

```
Vypocet regulacni odchylky ek2 pro PS reg z rovnice ek2 = w2 - y2   kde   y2 = (y - y_1)/Tvz:
LD      SM0.0                       // log.1
MOVR    uk:VD116, w2:VD400          // uk z PSD reg je w2 pro PS reg
MOVR    y:VD104, y2:VD408           // y2 = (y - y_1)/Tvz zde priprava y do y2
-R      y_1:VD404, y2:VD408         // y2 = (y - y_1)/Tvz zde odecteni y_1
/R      Tvz:VD13, y2:VD408          // y2 = (y - y_1)/Tvz zde deleni Tvz
MOVR    y:VD104, y_1:VD404          // nastaveni y jako y_1 pro pristi cyklus
MOVR    w2:VD400, ek2:VD412         // ek2 = w2 - y2 zde priprava w2 do ek2
-R      y2:VD408, ek2:VD412         // ek2 = w2 - y2 zde odecteni y2
```

Obr. 5.3 Část programu, ve které určujeme velikost regulační odchylky pro PS regulátor.

Na obrázku je vidět část programu, která se stará o určení regulační odchylky pro PS regulátor rychlosti. Vychází ze známé rovnice, kde regulační odchylka je rovna rozdílu mezi žádanou hodnotou a skutečnou hodnotou. Skutečnou hodnotu rychlosti v tomto případě je třeba dopočítat, neboť máme k dispozici pouze údaj o momentální poloze ramene. Pokud si v programu budeme pamatovat údaj o minulé poloze, lze hodnotu rychlosti určit jako rozdíl současné polohy a minulé polohy ramene, který podělíme časovým intervalem, který dělí tyto dva údaje. Nakonec po určení hodnoty regulační odchylky je tato uložena do proměnné ek2.

Dále je nutné dodělat algoritmu pro výpočet akčního zásahu PS regulátoru. Bude vycházet rovněž z rovnice 4.1, ve které ovšem bude poslední člen (derivační složky) roven nule, tedy se vynechá. Z regulační odchylky a dalších členů rovnice se vypočítá proporcionální složka a integrační složka akční veličiny. Kvůli již zmiňovaným důvodům teoreticky nekonečného nárůstu integrační složky, který se projevuje pouze u číslicové regulace (u analogové regulace bylo přirozeným „stropem“ nárůstu akční veličiny napájecí napětí analogových zesilovačů), je nutné provést opět porovnání a případnou korekci hodnoty do rozsahu od -1,05 do +1,05.

```
// omezeni integracni slozky (aby nemohla narustat do "nekonecna" za jistych podminek)
LD      SM0.0                       // log.1 tzn.
LDR>=   IntSl2:VD504, 1.05          // test zda je integracni slozka2 vetsi nez 1.05
MOVR    1.05, IntSl2:VD504          // pokud ano vnutime honotu 1.05
LDR<=   IntSl2:VD504, -1.05         // test zda je integracni slozka2 mensi nez -1.05
MOVR    -1.05, IntSl2:VD504         // pokud ano vnutime hodnotu -1.05
```

Obr. 5.4 Část programu pro omezení integrační složky

Porovnání se provádí pomocí instrukce „LDR<=" a „LDR>=", kde operátor pro porovnání je možné si představit mezi následujícími dvěma parametry instrukce. Pokud je výraz roven logické jedničce, provede se instrukce na dalším řádku, pokud ne, následující řádek se vynechá.

Jakmile máme integrační složku omezenou, je nutné ji dále sečíst se složkou proporcionální, jak je tomu ve vztahu 4.1. Jejich součet nám dá výslednou hodnotu akčního zásahu. Tato hodnota je v číselném rozsahu „real", proto je nutné ji převést na celé číslo a toto číslo následně do výstupního rozsahu od -16000 do 16000 (jak již bylo zmíněno dříve, převádíme do polovičního rozsahu kvůli změně výstupního napětí z +-10V na +-5V). Ještě před výpisem do obrazu výstupu hodnotu porovnáme, zda-li opravdu leží v daném rozsahu a případně provedeme korekci. Po této operaci již hodnotu přesuneme na adresu AQW0, ze které si automat bere hodnoty pro svůj první analogový výstup (označený nulou).

Na závěr v editoru datového bloku, kde máme nastaveny počáteční hodnoty proměnných, doplníme nastavení proměnných nových.

```
//akcni zasah od PS regulatoru
uk2:VD416 0.0

//regulacni odchylka pro PS regulator
ek2:VD412 0.0

//akcni zasah od integracni slozky minuly
uik_12:VD420 0.0

//minula hodnota polohy ramene
y_1:VD404 0.0
```

Obr. 5.5 Nové hodnoty v editoru datového bloku

Tímto je druhý program hotov (celý zdrojový kód je na přiloženém CD). Byl přenesen do PLC a různými podněty z regulované soustavy bylo ověřeno chování jednotlivých proměnných, které se chovaly podle očekávání. Tímto byla odzkoušena funkčnost programu a dále bylo nutné provést nastavení konstant obou regulátorů pomocí Ziegler-Nicholsonovi metody.

5.2 Nastavení regulátorů

Jelikož na rozdíl od programu prvního máme zde regulátory dva, je nutné provést nejprve určení parametrů podle Ziegler-Nicholsonovi metody pro PS regulátor rychlosti, kdy bude PSD regulátor polohy vyřazen z provozu. Po nastavení PS regulátoru budeme provádět nastavení regulátoru PSD, které ovšem bude již s PS regulátorem v provozu. Začneme tedy nastavením regulátoru pro rychlost. V programu vyřadíme PSD regulátor z provozu tím, že jeho výstup nastavíme „natvrdo" na nulu. Výstupní hodnota z tohoto regulátoru nám ovšem reprezentuje žádanou hodnotu pro PS regulátor. Je tedy nutné i přes vyřazený regulátor polohy na vstup regulátoru rychlosti přivádět jistou nenulovou hodnotu, která nám bude reprezentovat žádanou nenulovou rychlost. Jelikož snímač polohy ramene má pracovní rozsah přibližně od -150° do 150° není možné aby tato hodnota žádané rychlosti respektive její doba trvání nám dostala rameno z této oblasti. Je tedy nutné aby

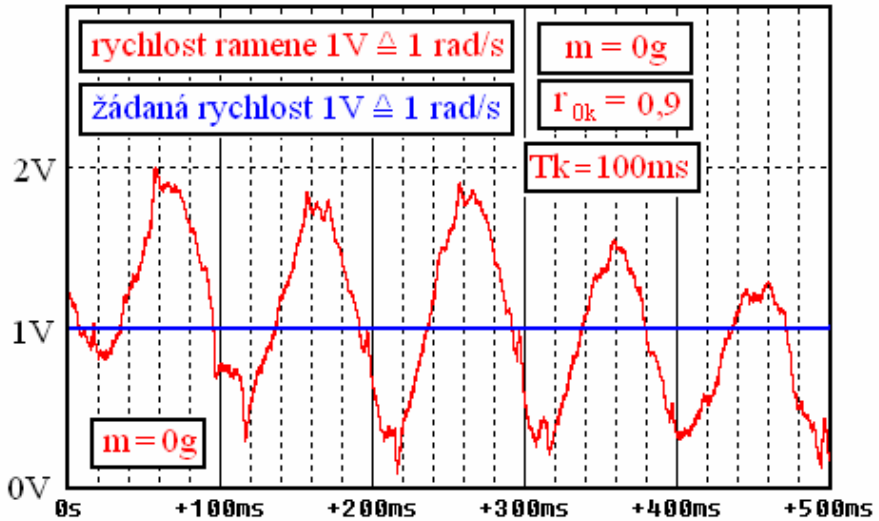
žádaná hodnota rychlosti se v čase měnila a to tak, aby se rameno kývalo danou rychlostí přibližně kolem nulové hodnoty a jeho výchylka nepřesáhla pracovní rozsah snímače. K tomu nám poslouží pár řádků instrukcí.

```
LD      SM0.0                   // log.1
LD      SM0.5                   // specialni bit, log 1/0 po 0,5s
+I      1, pocitadlo:VW508      // pricteni hodnoty 1
LDW=    50, pocitadlo:VW508     // porovnani s hodnotou 50
*R      -1.0, ukazatel:VD512    // zmena znamenka
MOVW    0, pocitadlo:VW508      // vynulovani

LD      SM0.0                   // log. 1
LDD>    0, ukazatel:VD512       // urceni kladnosti hodnoty
MOVR    0.2, uk:VD116           // prirazeni hodnoty 0,2
LDD<    0, ukazatel:VD512       // urceni zapornosti hodnoty
MOVR    -0.2, uk:VD116          // prirazeni hodnoty -0,2
```

Obr. 5.6 Podprogram pro generování pulzů žádané hodnoty

Jak je vidět z obrázku 5.6, ke „generování“ obdélníkových pulzů žádané hodnoty rychlosti, je využit speciální bit SM0.5. Tento bit generuje hodinový pulz, který je v log. 1 po dobu 0,5 sekundy. Následně je vypnutý po 0,5 sekundy a pak v log. 0 po dobu 0,5 sekundy v pracovním cyklu 1 sekunda. K funkci generátoru potřebujeme ještě dvě nové proměnné „pocitadlo“ a „ukazatel“. Proměnná počítadlo, jak je vidět z obrázku, v každém cyklu přerušení je inkrementována o hodnotu jedna, pokud je bit SM0.5 v logické jedničce. Jakmile má počítadlo hodnotu 50, vynásobí se proměnná ukazatel hodnotou -1. Tímto je zajištěna její změna znaménka a počítadlo se vynuluje.V další části je porovnáváno, zda je ukazatel kladný nebo záporný. Pokud je kladný, výstupem bude kladná hodnota žádané rychlosti o velikosti 0,2. V opačném případě je hodnota -0,2. Jelikož je vzorkovací perioda 20ms, generuje tento algoritmus obdélníkový signál s periodou přibližně 4s. Signál je symetrický kolem nuly a má velikost amplitudy 0,4 jednotky žádané rychlosti.

Tento signál byl přiváděn jako žádaná hodnota rychlosti na vstup PS regulátoru, který měl vyřazenu integrační složku. Pomocí změny proporcionální konstanty jsem soustavu, která neobsahovala žádné závaží (ze stejného důvodu jako tomu bylo i u předchozího programu) na rameni, přivedl na hranici stability.



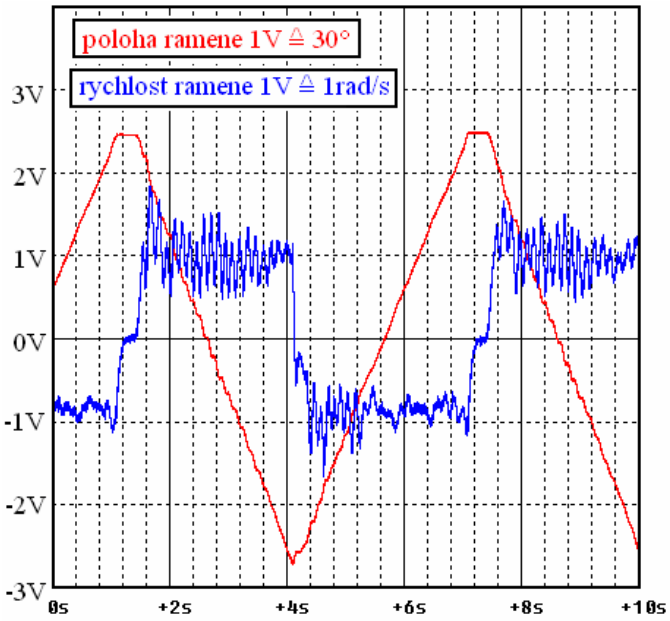
Obr. 5.7 Průběh rychlosti ramene na hranici stability

Na obrázku 5.7 je výřez z průběhu rychlosti, na kterém vidíme rychlostní kmity. Tyto kmity mají periodu 100ms a tato hodnota periody odpovídá periodě kritické.

Nastává při zesílení 0,9. Podle tabulky 4.1 [Švarc 2007] jsou parametry blízké optimálním pro tento regulátor tedy rovny:

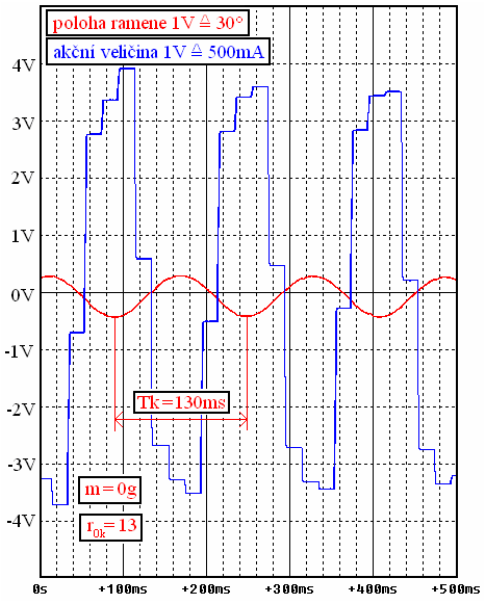
- Zesílení Kp=0,6Kpk, tedy Kp=0,6*0,9= **0,54 [-]**
- Časová integrační konstanta Ti=0,5Tk, tedy Ti= 0,5*0,1= **0,05 [s]**

Pro tyto hodnoty proporcionální a časové integrační konstanty byla soustava nestabilní. Bylo nutné tedy přistoupit ke korekci těchto hodnot a najít tak parametry vhodnější. Jako blízké optimálnímu se nakonec ukázalo nastavení při **Kp=0,6** a **Ti=0,1**. Na obrázku 5.8 je vidět průběh polohy ramene i jeho rychlosti při tomto nastavení regulátorů. Žádanou hodnotu rychlosti jsme opět generovali pomocí podprogramu na obrázku 5.6.



Obr. 5.8 Průběh polohy a rychlosti ramene při nastaveném regulátoru rychlosti

Parametry PS regulátoru tedy máme. Nyní je nutné určit parametry blízké optimálním pro PSD regulátor polohy. V programu vyřadíme z funkce generátor pulzů žádané rychlosti a obnovíme funkci PSD regulátoru. Použijeme opět Ziegler-Nicholsonovu metodu, při které vyřadíme derivační a integrační složku regulátoru a změnou proporcionální konstanty dostaneme soustavu na hranici stability. Žádaná hodnota polohy bude nastavena na hodnotu 0˚. Pro rozkmitání soustavy budeme rameno vychylovat do polohy 45˚. Kritické zesílení, při kterém nastalo kmitání přibližně na hranici stability bylo 13. Hodnota kritické periody kmitu byla 130ms.

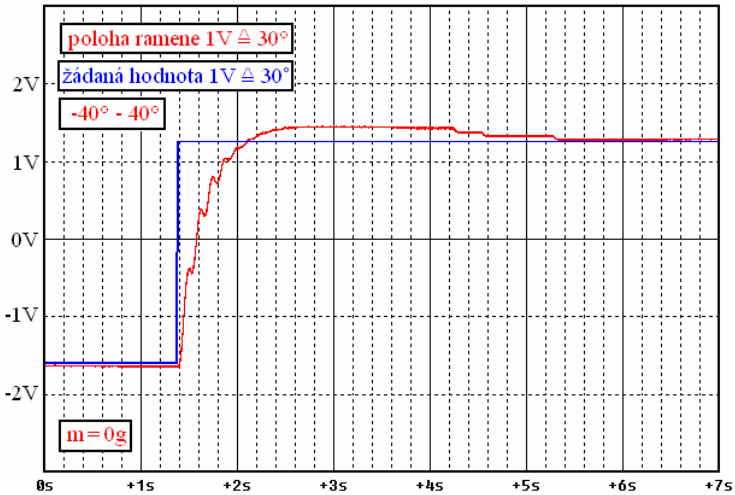


Obr. 5.9 Průběh polohy ramene a akční veličiny na hranici stability

Podle naměřené hodnoty kritického kmitu a odpovídajícímu kritickému zesílení určíme parametry blízké optimálním pro PSD regulátor polohy, který je v regulačním obvodu zapojen spolu s PS regulátorem rychlosti. Opět využijeme tabulky 4.1 [Švarc 2007] k určení vhodných parametrů:

- Zesílení Kp=0,6Kpk, tedy Kp=0,6*13= **7,8 [-]**
- Časová integrační konstanta Ti=0,5Tk, tedy 0,5*0,13= **0,65 [s]**
- Časová derivační konstanta Td=0,12Tk tedy 0,12*0,13= **0,0156 [s]**

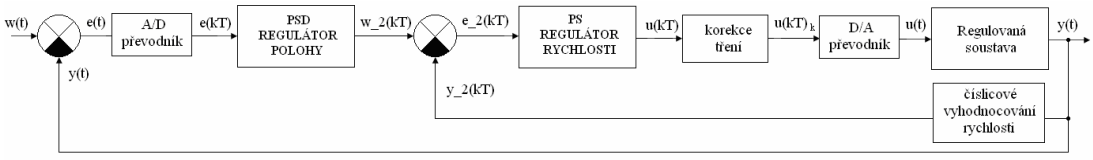
Pro tyto parametry byla soustava opět nestabilní. Bylo nutné tedy parametry upravit. Chování soustavy se zdálo být nejlepší při hodnotách Kp=5, Ti=2,5 a Td=0,01. Na obrázku 5.10 je vidět přechodová charakteristika soustavy pro nově nastavené regulátory. Skoková změna žádané polohy ramene byla z -40˚ na 40˚.



Obr. 5.10 Přechodová charakteristika pří optimálním nastavení regulátorů

6. Podprogram korekce tření

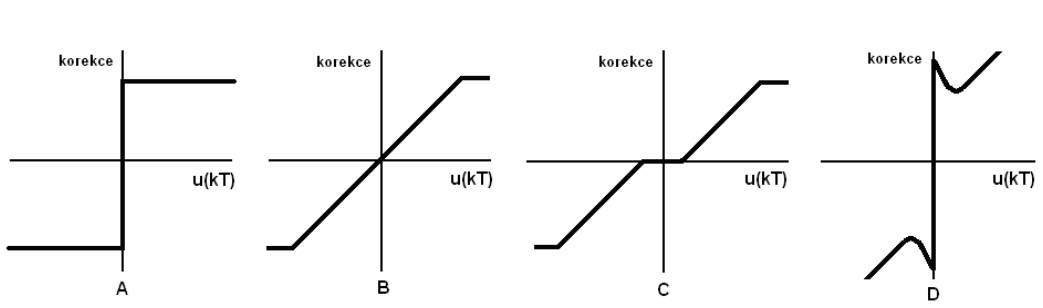
V kapitole 4 jsme vytvořili první program pro PSD regulaci polohy ramene. V kapitole 5 jsme tento program rozšířili o další regulátor rychlosti. Existují však i další možnosti, jak program rozšířit a tím zvýšit dosažitelnou kvalitu regulace. Jednou z možností je do regulační smyčky přidat blok, který bude mít za úkol korekci tření. Tření hraje v řízení naší soustavy významnou roli a je vhodné se jej jistým způsobem snažit minimalizovat nebo alespoň částečně eliminovat.



Obr. 6.1 Regulační smyčka s korekcí tření

Abychom mohli nějakým způsobem odhadnout velikost tření, respektive velikost potřebného proudu k jeho překonání, bylo nutné provést měření. Rameno bylo ponecháno v poloze 0° natočení a neobsahovalo žádné závaží. S postupným zvyšováním proudu do motoru jsem sledoval, kdy se rameno „utrhne“ ze stacionární polohy (při jakém proudu se začne pohybovat). Tento pokus jsem opakoval pro obě polarity (oba směry natočení) několikrát. Proud pro překonání tření ,pro oba směry, byl přibližně 150mA, kde v záporném směru tento proud byl záporné polarity.

Pomocí těchto hodnot můžeme vytvořit podprogram, který bude částečně eliminovat nepříznivou vlastnost tření. Hodnota 150mA proudu motoru odpovídá výstupní hodnotě 1200 jednotek na výstupu automatu. Analogicky hodnota -150mA odpovídá hodnotě -1200 jednotek. Možných řešení korekce tření je celá řada. Vhodnost jednotlivých řešení závisí na regulované soustavě. Na obrázku 6.2 jsou uvedeny možné průběhy korekce tření, kde na ose x je akční zásah, který je třeba korigovat a na ose y je korekce.



Obr. 6.2 Příklady možných korekci tření

Po vyzkoušení několika variant a následném měření se jako nejlepší pro naši soustavu ukázal průběh, který odpovídá na obrázku 6.2 označení „b“. Tento průběh je spojitý v okolí nuly a nevznikají při něm prudké skokové změny, které by mohly nepříznivě ovlivňovat výslednou dynamiku regulace. Průběh korekce je lineární v okolí nuly a po překročení jistého pásma (pásma nasycení) je dále konstantní. V našem konkrétním případě bude hodnota korekce v tomto pásmu 1200 ,respektive -1200 jednotek,

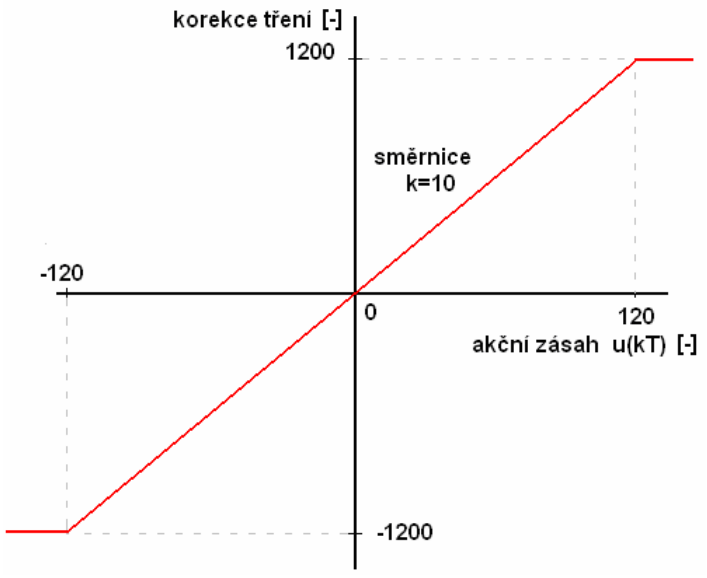
podle polarity akčního zásahu. Velikost korekce v lineárním pásmu budeme počítat pomocí směrnice, která udává sklon této oblasti. Velikost oblasti bude 240 jednotek, tedy od -120 po 120. Směrnice bude mít hodnotu 10. gK realizaci takovéto korekce vytvoříme podprogram, který nazveme podle jeho činnosti jako „Korekce_třeni“.

```
Podprogram pro korekci akcniho zasahu ve smyslu snizeni vlivu treni na regulovonou soustavu
Network 1
Korekce treni:
LD      SM0.0                   // log.1
MOVD    uk:VD116, AC0           // vlozime si uk do AC0
MOVD    AC0, AC1                // a take do AC1
LDD>    -121, AC1               // zjistime jestli uk je mensi nez -161
-I      1200, AC0               // pokud ano provedeme korekci -1600 jednotek
LDD<    121, AC1                // zjistime jestli uk je vetsi nez 161
+I      1200, AC0               // pokud ano provedeme korekci +1600 jednotek
MOVD    AC0, uk:VD116           // AC0 zpet do uk

LDW=    AC0, AC1                // zjistime jestli AC0 je stejne jako AC1
*I      10, AC0                 // vynasobime konstantou 10
MOVD    AC0, uk:VD116           // hodnotu po korekci vlozime do uk
```

Symbol	Address	Comment
uk	VD116	akcni zasah nynejsi - vystup z PSD reg.

Obr. 6.3 Podprogram korekce tření

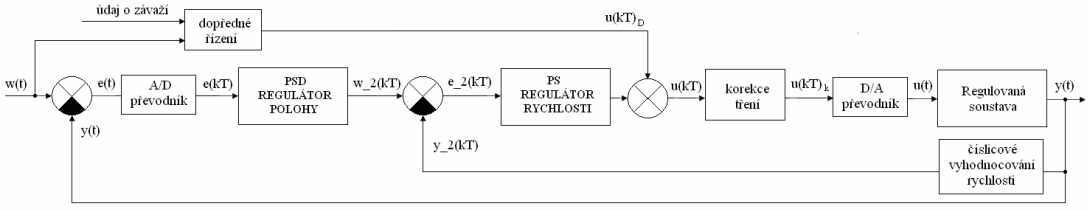
Podprogram bude volán v poslední fázi podprogramu přerušení, kde bude již známá hodnota akčního zásahu. Nejprve určíme, zda tato hodnota neleží v okolí nuly (oblast od -120 do 120 jednotek). Neleží-li v této oblasti, můžeme provést korekci přičtením hodnoty 1200 ,respektive -1200 jednotek, v závislosti na polaritě akční veličiny. Pokud se tato korekce neprovede, znamená to, že hodnota v této oblasti leží a akční veličina zůstane nepozměněna. To si ověříme porovnávací instrukcí a v případě kladného výsledku vynásobíme akční veličinu již zmiňovanou konstantou 10.



Obr. 6.4 Průběh korekce tření

7. Podprogram dopředného řízení

Dalším možným zlepšením dynamiky soustavy při regulaci je využití tzv. dopředného řízení. Jak již název napovídá, jedná se o jistý zásah, který je oproti ostatním v předstihu, respektive k určení jeho hodnoty není zapotřebí zpětné vazby. Vstupními veličinami do tohoto bloku je údaj o žádané hodnotě natočení ramene a údaj o závaží na rameni. Je známo, že proud jdoucí do motoru vyvolává na jeho hřídeli kroutící moment úměrný tomuto proudu. Do protisměru tohoto elektrického momentu působí moment mechanický, který je úměrný délce ramene a hmotnosti závaží umístěného na tomto rameni. Pokud je rameno v nulové poloze a převládá moment elektrický, rameno se vychýlí. Z údajů o požadované hodnotě polohy ramene a velikosti mechanického momentu, který je funkcí hodnoty závaží, budeme schopni bez zpětné vazby dopředu určit, jak velký akční zásah bude třeba provést, aby se rameno vychýlilo do námi požadované polohy.



Obr. 7.1 Blokové schéma regulace s dopředným řízením

Hodnotu „dopředného zásahu“ pro naši soustavu budeme určovat pomocí vztahu 7.1, kde $u(k)_D$ je hodnota dopředného akčního zásahu, který se bude přičítat k akčnímu zásahu z regulátoru, jak je vidět ze schématu na obrázku 7.1. Tuto hodnotu určíme jako násobek proudu potřebného k vychýlení ramene do polohy 90° a sinu žádané hodnoty natočení v radiánech.

$$u(k)_D = I_{90^\circ} \sin w_\varphi \qquad (7.1)$$

Jelikož budeme hodnotu dopředného akčního zásahu počítat v automatu, bude výhodné za proměnnou I_{90° nedosazovat hodnoty v ampérech, nýbrž v těmto hodnotám odpovídajících výstupních jednotkách automatu. Tedy za I_{90° dosadíme takovou hodnotu z výstupního rozsahu automatu, která vyvolá průchodem přes D/A převodník a další obvody řazenými v regulované soustavě před motorem právě takový proud, který je potřebný ke stabilní výchylce ramene 90°.

Tyto hodnoty byly naměřeny pomocí automatu, který obsahoval první program pro PSD regulaci polohy ramene. Žádaná hodnota byla nastavena právě na zmíněných 90°. Po ustálení ramene na této hodnotě jsem odečetl pomocí funkce „program status“, která je integrována v prostředí Step7 a slouží k on line nahlížení do proměnných hodnotu akčního zásahu. Tato hodnota odpovídala nutnému zásahu k „udržení“ ramene v poloze 90°. Toto měření jsem provedl pro všechny kombinace závaží a pro oba směry výchylky. Tedy pro výchylky +90° a -90°. Hodnoty akčních zásahů pro jednotlivá závaží jsem

zaznamenal do tabulky. Tyto hodnoty byly pro obě polarity symetrické. Tedy každá hodnota akčního zásahu pro jednotlivá závaží v kladném směru odpovídala absolutní hodnotě akčního zásahu ve směru záporném. S pomocí těchto hodnot je možné vytvořit podprogram, který nazveme „Dopředné_řízení“. Na obrázku 7.2 je vidět první část tohoto podprogramu.

```
Zjisteni hodnoty zavazi a pocatecni nastaveni ukD (pocatecni nastaveni hodnoty na hodnotu pri natoceni 90 stupnu):
LD      SM0.0                   // log.1
MOVD    0, ukD:VD1018           // vynulujeme si promennou ukD

LDW=    zavazi:VW1010, 0        // jestlize je zavazi 0g
MOVR    3000.0, ukD:VD1018      // prirad ukD hodnotu 3000.0

LDW=    zavazi:VW1010, 10       // jestlize je zavazi 10g
MOVR    5000.0, ukD:VD1018      // prirad ukD hodnotu 5000.0

LDW=    zavazi:VW1010, 20       // jestlize je zavazi 20g
MOVR    6700.0, ukD:VD1018      // prirad ukD hodnotu 6700.0

LDW=    zavazi:VW1010, 30       // jestlize je zavazi 30g
MOVR    8800.0, ukD:VD1018      // prirad ukD hodnotu 8800.0

LDW=    zavazi:VW1010, 40       // jestlize je zavazi 40g
MOVR    10200.0, ukD:VD1018     // prirad ukD hodnotu 10200.0

LDW=    zavazi:VW1010, 50       // jestlize je zavazi 50g
MOVR    12200.0, ukD:VD1018     // prirad ukD hodnotu 12200.0
```

Obr. 7.2 První část podprogramu „Dopředného řízení“

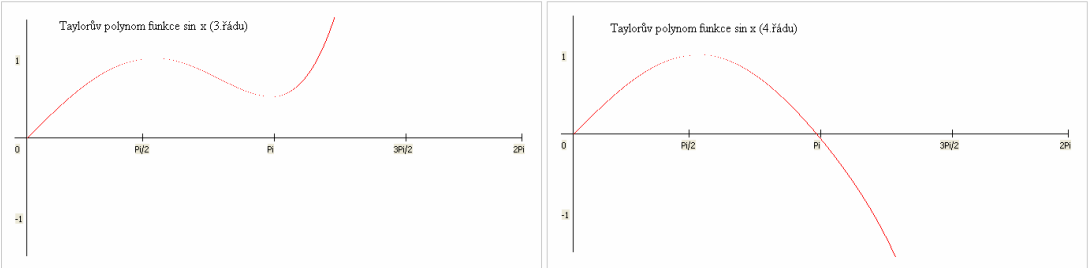
Tato první část porovnává hodnotu proměnné na adrese VW1010 s jednotlivými hodnotami závaží. Na této adrese bude uložena hodnota závaží, která bude nastavena pomocí dotykového displeje (viz kapitola 9 využití dotykového displeje). Z tohoto porovnávání bude vždy pouze jedno s logickým výsledkem jedna a následně pro tuto hodnotu se do proměnné ukD, do které budeme ukládat hodnotu dopředného akčního zásahu, vloží příslušná hodnota na dalším řádku. Zbylé řádky zůstanou neobslouženy. K kódu lze vyčíst jednotlivé hodnoty akčních zásahů, které byly určeny měřením popisovaným výše. Po proběhnutí této části budeme mít tedy hodnotu „$I_{90°}$“ (ve výstupních jednotkách automatu) ze vztahu 7.1 uloženou v proměnné ukD. Dále se ve zmiňovaném vztahu objevuje žádaná hodnota natočení. Tu máme jako globální proměnnou celého programu, proto nebude problém s jejím umístěním do našeho podprogramu. Problém ale nastává s goniometrickou funkcí sinus. Instrukční soubor neobsahuje instrukce pro goniometrické funkce a proto bude nutné tuto funkci vhodným způsobem nahradit. Jako náhrada byl zvolen Taylorův polynom pro funkci sinus x (vztah (7.2)).

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} \tag{7.2}$$

Přesnost aproximace funkce závisí na řádu polynomu. Čím je rozvoj delší (je většího řádu), tím je aproximace přesnější. V našem případě budeme požadovat přesnou

aproximaci funkce v rozsahu od –π po +π (vyjádřeno v radiánech, pokud bychom uvažovali rozsah ve stupních byl by od -180° po 180°). V tomto rozsahu se rameno pohybuje (tento rozsah je pouze teoretický, ve skutečnosti lze efektivně řídit rameno v oblasti přibližně od -150° po 150° viz kapitola 2 – charakteristika snímače polohy).

Abychom nepočítali zbytečně polynom moc velkého řádu nebo naopak neurčili pro aproximaci řád polynomu moc malý a hodnoty by neodpovídaly funkci sinus, bude vhodné si průběhy polynomu pro jednotlivé řády vykreslit.



Obr. 7.3 Průběh funkce Taylorova polynomu 3. a 4. řádu

K vykreslení průběhu funkce jsem vytvořil jednoduchý program v programovacím prostředí Delphi (spustitelný soubor tohoto programu je na přiloženém CD).

Jelikož jak bylo řečeno, budeme náhradu funkce používat pouze v rozsahu –π po +π, bude nám podle důkazu na obrázku 7.3 dostačovat polynom 4.řádu (na obrázku je průběh funkce od 0 do π, jelikož je funkce sin x symetrická, není třeba vykreslovat zápornou oblast).

Nyní můžeme pokračovat ve tvorbě podprogramu. Jelikož při výpočtu jednotlivých členů Taylorova polynomu používáme mocniny a také jednotlivé členy se od sebe střídavě sčítají a odčítají, je nutné žádanou hodnotu natočení ramene (která bude argumentem aproximace sin x) vkládat do výpočtu v kladném tvaru a po provedení celého výpočtu podle vztahu 7.1 teprve rozhodovat o znaménku akčního zásahu (v závislosti na znaménku žádané hodnoty).

```
Vypocet sinu uhlu natoceni a vynasobeni hodnotou ukD (hodnota pri natoceni 90 stupnu):
LD      SM0.0                       // log.1
MOVR    w:VD100, w_d:VD1024         // ulozime si hodnotu w do promenne w_d

// nejprve si urcime znamenko hodnoty w
// hodnotu dale mocnime (v pripade zaporne hodnoty by se znamenko menilo)

LDR<=   0.0, w_d:VD1024             // jestlize je w kladna
MOVW    0, znaminko:VW1016          // znaminko bude 0

LDR>    0.0, w_d:VD1024             // jestlize je w zaporna
MOVW    1, znaminko:VW1016          // znaminko bude 1
*R      -1.0, w_d:VD1024            // udelame z w prozatim kladnou hodnotu
// tj je li znaminko 1 dale budeme uvazovat ze vstupni w je zaporna
```

Obr. 7.4 Určení znaménka pro výsledný dopředný akční zásah

Dále je nutné převést rozsah žádané hodnoty, který je od -0,5 do +0,5 (který odpovídá natočení od-180° po 180°) na rozsah od -3,14 po +3,14, který odpovídá úhlům od -180° po 180° v radiánech (abychom tyto hodnoty mohli dosazovat do jednotlivých členů Taylorova polynomu).

```
// polynom aproximujici funkci sin x je ve tvaru:
// sin x = x - (x^3/3!) + (x^5/5!) - (x^7/7!)
// pripravime si jednotlive cleny
// clen (x^3/3!)
LD     SM0.0                        // log.1
MOVR   w_d:VD1024, AC0              // priprava w do AC0
*R     6.28, AC0                    // prevod na radiany (-3.14 az +3.14)
MOVR   AC0, AC1                     // x do AC1
*R     AC0, AC0                     // x na druhou
*R     AC0, AC1                     // x na treti do AC1
/R     6.0, AC1                     // x na treti deleno 3 faktorial do AC1
```

Obr. 7.5 Výpočet prvního členu Taylorova polynomu

Na obrázku 7.5 je ukázka výpočtu prvního členu Taylorova polynomu. Na třetím řádku kódu je výše zmiňovaný převod na rozsah v radiánech. Následuje výpočet prvního (respektive druhého) členu Taylorova polynomu. Analogicky se dopočítají i zbylé členy polynomu a podle jejich pořadí a znamének ze vztahu 7.2 se tyto členy navzájem sečtou ,respektive odečtou.

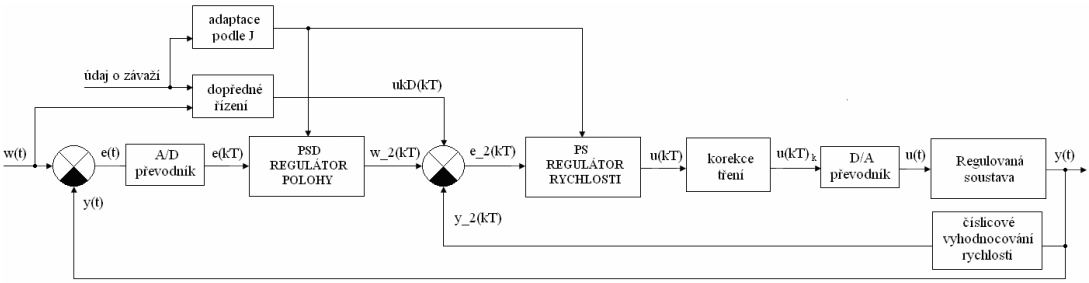
```
LDW=   znaminko:VW1016, 1           // jestlize znaminko je 1 (zaporna w na vstupu)
*R     -1.0, sinX:VD1012            // vynasobime hodnotu sinX -1

LD     SM0.0                        // log.1
// hodnotu ukD vynasobime hodnotou sinX a vlozime do promenne ukD
*R     sinX:VD1012, ukD:VD1018      //  sinX krat ukD
TRUNC  ukD:VD1018, ukD:VD1018       // skratime real na D_integer
//nyni je v ukD hodnota dopredneho akcniho zasahu
```

Obr. 7.6 Určení znaménka pro dopředný akční zásah

Nyní již známe hodnotu dopředného akčního zásahu. Na konec podprogramu je nutné provést kontrolu, zdali výsledná hodnota nemá být záporná. Pokud je v proměnné „znamínko“ hodnota rovna jedné, byla žádaná hodnota na začátku podprogramu záporná a je třeba celý dopředný akční zásah vynásobit hodnotou -1.

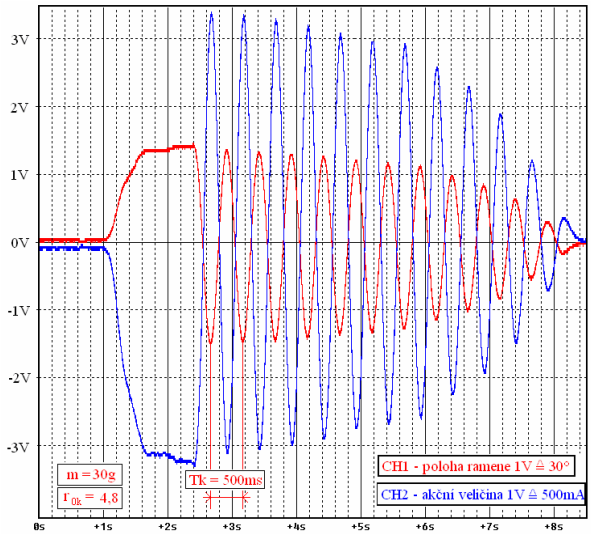
8. Podprogram adaptaci podle momentu setrvačnosti

Jelikož, jak již bylo uvedeno v kapitole 7, budeme mít využitím dotykového displeje k dispozici údaj o hodnotě závaží na rameni, byla by škoda jej nevyužít k adaptaci (přizpůsobení) jednotlivých konstant pro již vytvořené regulátory. Jako vstup do bloku adaptivní regulace nám bude sloužit právě tato hodnota závaží na rameni. Výstupem z něj bude zásah do parametrů použitých regulátorů, které změní svoje parametry (adaptují se) podle režie adaptivního bloku.

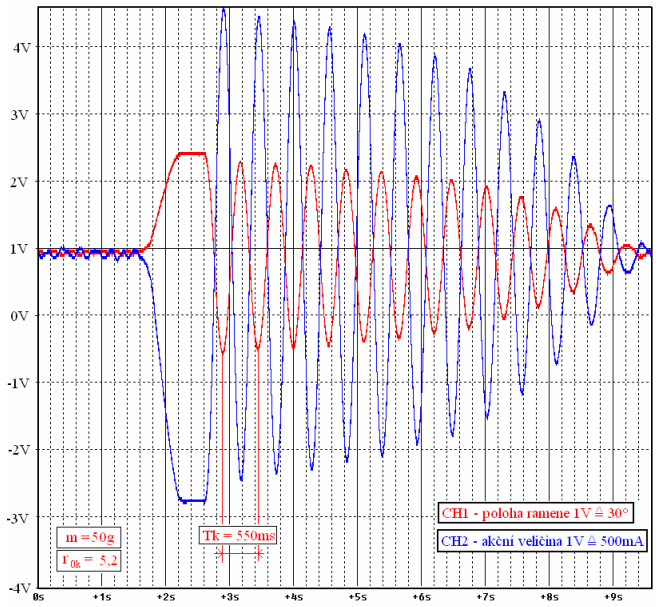


Obr. 8.1 Blokové schéma regulace s adaptací podle momentu setrvačnosti ramene

Jelikož máme k dispozici dva programy, u kterých se parametry regulátorů navzájem liší, bude nutné pro každý z nich vytvořit podprogram na „adaptaci“ zvlášť. Začneme jednodušší variantou a to programem s jedním regulátorem pro PSD regulaci polohy ramene. Abychom mohli měnit parametry tohoto regulátoru, bylo nutné pro všechny možné kombinace závaží na rameni provést experimentální Ziegler-Nicholsonovu metodu. Z těchto měření nám vyšly parametry blízké optimálním pro jednotlivá závaží. Na obrázku 4.10 v kapitole 4 je vidět průběh polohy ramene a akční veličiny na hranici soustavy bez závaží. Zde na obrázku 8.2,respektive 8.3, uvádím průběhy na hranici stability se závažím o hmotnosti 30g, respektive 50g.



0br. 8.2 Průběh polohy ramene a akční veličiny na hranici stability soustavy se závažím m=30g



Obr. 8.3 Průběh polohy ramene a akční veličiny na hranici stability soustavy se závažím m=50g

Z odečtených hodnot kritické periody a z údaje kritického zesílení jsem dopočítal parametry regulátorů pro jednotlivá zatížení ramene podle tabulky 4.1 [Švarc 2007]. Průběhy pro zatížení 10g, 20g a 40g uvedeny nejsou. Jejich hodnoty kritického zesílení a kritické doby kmitu tedy spolu s ostatními hodnotami uvádím v tabulce 8.1.

hmotnost závaží na rameni [g]	kritické zesílení [-]	kritická dobu kmitu [ms]	Kp[-]	Ti [s]	Td [s]
0	3,6	400	2,16	0,185	0,044
10	4,1	430	2,46	0,215	0,052
20	4,5	470	2,7	0,235	0,056
30	4,8	500	2,88	0,25	0,06
40	5	525	3	0,263	0,063
50	5,2	550	3,12	0,275	0,066

Tab 8.1 Naměřené a vypočtené hodnoty Ziegler-Nicholsonovou metodou pro PSD regulátor

Tyto hodnoty použijeme při vytváření prvního podprogramu „Adaptace_podle_J_PSD“. Tento podprogram v případě změny údaje o závaží bude muset být volaný před algoritmem výpočtu akční veličiny, neboť jeho zásah do parametrů regulátoru bude mít významný vliv na výsledek výpočtu. Jak již bylo řečeno, pomocí dotykového displeje budeme měnit hodnoty v proměnné na adrese VW1010 v závislosti na aktuálním závaží na rameni. Proto tedy první část podprogramu pro adaptaci bude vyhodnocovat údaj na této adrese.

```
Urceni parametru Kp, Ti a Td podle J (respektive podle udaje o zavazi na rameni) pro PSD regulator polohy:
LDW=    zavazi:VW1010, 0            // pokud je udaj o zavazi 0 (rameno bez zavazi)
MOVR    2.16, Kp:VD0                // proporcionalni konstanta
MOVR    0.2, Ti:VD4                 // casova integracni konstanta
MOVR    0.0444, Td:VD8              // casova derivacni konstanata

LDW=    zavazi:VW1010, 10           // pokud je udaj 10 (rameno se zavazim 10g)
MOVR    2.46, Kp:VD0                // proporcionalni konstanta
MOVR    0.215, Ti:VD4               // casova integracni konstanta
MOVR    0.0516, Td:VD8              // casova derivacni konstanata

LDW=    zavazi:VW1010, 20           // pokud je udaj 20 (rameno se zavazim 20g)
MOVR    2.7, Kp:VD0                 // proporcionalni konstanta
MOVR    0.235, Ti:VD4               // casova integracni konstanta
MOVR    0.0564, Td:VD8              // casova derivacni konstanata


LDW=    zavazi:VW1010, 30           // pokud je udaj 30 (rameno se zavazim 30g)
MOVR    2.88, Kp:VD0                // proporcionalni konstanta
MOVR    0.25, Ti:VD4                // casova integracni konstanta
MOVR    0.06, Td:VD8                // casova derivacni konstanata

LDW=    zavazi:VW1010, 40           // pokud je udaj 40 (rameno se zavazim 40g)
MOVR    3.0, Kp:VD0                 // proporcionalni konstanta
MOVR    0.2625, Ti:VD4              // casova integracni konstanta
MOVR    0.063, Td:VD8               // casova derivacni konstanata


LDW=    zavazi:VW1010, 50           // pokud je udaj 50 (rameno se zavazim 50g)
MOVR    3.12, Kp:VD0                // proporcionalni konstanta
MOVR    0.275, Ti:VD4               // casova integracni konstanta
MOVR    0.066, Td:VD8               // casova derivacni konstanata
```

Obr. 8.4 První část podprogramu pro adaptaci parametrů regulátoru

Pomocí instrukce pro porovnání zjistíme aktuální hodnotu závaží a na jejím základě změníme hodnoty globálních proměnných, které ovlivňují výpočet akčního zásahu (parametry regulátoru). Po změně těchto hodnot je nutné ještě provést přepočet konstant Ci a Cd, které se počítají v našem první podprogramu „Start“, který se provádí pouze v prvním cyklu, neboť ve vztazích 4.2 a 4.3, které slouží k určení těchto konstant figurují konstanty Kp, Ti a Td., které jsme právě změnili. Na obrázku 8.5 je zobrazen výpočet nové derivační konstanty.

```
Vypocet nove derivacni konstanty pro PSD regulator  Cd = Kp * Td/Tvz (podle udaje o zavazi):
LD      SM0.0                       // log.1
MOVR    0.0, Cd:VD128               // vynulujeme promennou Cd
MOVR    Td:VD8, Cd:VD128            // Td do Cd
/R      Tvz:VD13, Cd:VD128          // Td/Tvz
*R      Kp:VD0, Cd:VD128            // Cd = Kp * Td/Tvz
```

Obr. 8.5 Výpočet nové derivační konstanty

Výpočet nové integrační konstanty se provádí analogicky. Při výpočtu se ovšem držíme vztahu 4.3.

Tímto je první podprogram pro adaptaci PSD regulátoru polohy ramene hotový. Pro druhý program se dvěma regulátory je tvorba podprogramu podobná. Ten nazveme intuitivně „Adaptace_podle_J_PSD_PS“. Pro jeho tvorbu bylo však nejprve nutné určit parametry pro PS regulátor, kde PSD regulátor bude vyřazen z funkce. Po jejich určení přijde na řadu PSD regulátor, při jehož určování parametrů bude PS regulátor v chodu.

Ziegler-Nicholsonovu motodou byly určeny hodnoty, které by měly být blízké optimálním. Jak již ale z kapitoly 5. víme, při nastavování parametrů pro smyčku rychlosti

jsou tyto hodnoty pouze orientační a vhodnou úpravou je možné najít tyto konstanty vhodnější. V tabulce 8.2 jsou hodnoty určené metodou Ziegler-Nicholsonovou.

PS regulátor rychlosti

hmotnost závaží na rameni [g]	kritické zesílení [-]	kritická dobu kmitu [ms]	Kp[-]	Ti [s]
0	0,9	100	0,54	0,05
10	1,6	105	0,96	0,053
20	2,4	110	1,44	0,055
30	3,3	115	1,98	0,058
40	4,3	120	2,58	0,06
50	5,4	125	3,24	0,063

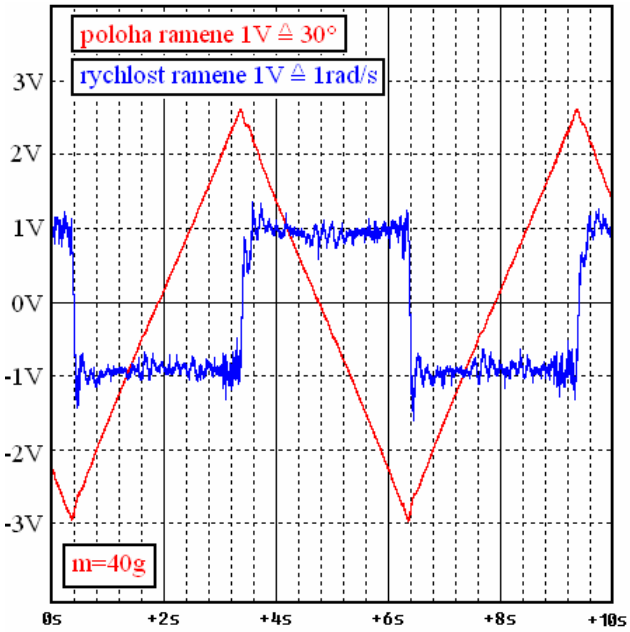
Tab. 8.2 Naměřené a vypočtené hodnoty Ziegler-Nicholsonovou metodou pro PS regulátor

Při těchto parametrech měla soustava, stejně jako tomu bylo v kapitole 5, nestabilní chování. Bylo nutné jednotlivé hodnoty pro jednotlivá závaží upravit tak, aby se chování soustavy zlepšilo. Na obrázku 8.6 je vidět část podprogramu pro adaptaci, kde je možné odečíst jednotlivé korigované hodnoty. Pro tyto hodnoty měla soustava lepší chování než s nastavením podle metody Ziegler-Nicholsonovi.

```
Urceni parametru Kp2, Ti2 podle J (respektive podle udaje o zavazi na rameni) pro PS regulator rychlosti:
LDW=    zavazi:VW1010, 0          // pokud je udaj o zavazi 0 (rameno bez zavazi)
MOVR    0.6, Kp2:VD300            // proporcionalni konstanta 0,6
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1

LDW=    zavazi:VW1010, 10         // pokud je udaj 10 (rameno se zavazim 10g)
MOVR    0.95, Kp2:VD300           // proporcionalni konstanta 0,95
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1

LDW=    zavazi:VW1010, 20         // pokud je udaj 20 (rameno se zavazim 20g)
MOVR    1.3, Kp2:VD300            // proporcionalni konstanta 1,3
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1

LDW=    zavazi:VW1010, 30         // pokud je udaj 30 (rameno se zavazim 30g)
MOVR    1.65, Kp2:VD300           // proporcionalni konstanta 1,65
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1
LDW=    zavazi:VW1010, 40         // pokud je udaj 40 (rameno se zavazim 40g)
MOVR    2.0, Kp2:VD300            // proporcionalni konstanta 2,0
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1

LDW=    zavazi:VW1010, 50         // pokud je udaj 50 (rameno se zavazim 50g)
MOVR    2.35, Kp2:VD300           // proporcionalni konstanta 2,35
MOVR    0.1, Ti2:VD304            // casova integracni konstanta 0,1
```

Obr. 8.6 Část podprogramu s korigovanými parametry PS regulátoru

Na obrázku 8.7 je pro názornost uveden průběh polohy a rychlosti ramene, které bylo zatíženo 40g závažím a jehož žádaná rychlost byla opět generována podprogramem z obrázku 5.6. Rameno bylo řízeno PS regulátorem rychlosti, který měl již korigovány parametry pro odpovídající závaží.



Obr. 8.7 Průběh polohy a rychlosti ramene se závažím m=40g

Dále bylo nutné nastavit PSD regulátor polohy. Opět jsme pomocí Ziegler-Nicholsonovi metody orientačně určili parametry regulátoru, které by se měly blížit hodnotám optimálním. Tyto hodnoty byly opět určeny pro kombinaci všech závaží na rameni. Změřené a dopočítané parametry jsou v tabulce 8.3.

PSD regulátor polohy

hmotnost závaží na rameni [g]	kritické zesílení [-]	kritická dobu kmitu [ms]	Kp[-]	Ti [s]	Td [s]
0	13	150	7,8	0,075	0,018
10	13	155	7,8	0,1775	0,0186
20	13	160	7,8	0,18	0,0192
30	13	165	7,8	0,1825	0,0198
40	13	170	7,8	0,185	0,0204
50	51	540	30,6	0,27	0,065

Tab 8.3 Naměřené a vypočtené hodnoty Ziegler-Nicholsonovou metodou pro PSD regulátor

Hodnoty v tabulce 8.3 jsou opět pouze orientační. Pro každé závaží na rameni byla zvlášť provedena úprava těchto hodnot pro zlepšení regulačních vlastností. Na obrázku 8.8 je vidět další část podprogramu, ze které je možno odečíst jednotlivé hodnoty pro jednotlivá závaží, která se při zkouškách jevila jako velmi blízká optimálním.

```
Urceni parametru Kp, Ti a Td podle J (respektive podle udaje o zavazi na rameni) pro PSD regulator polohy:
LDW=    zavazi:VW1010, 0            // pokud je udaj o zavazi 0 (rameno bez zavazi)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    2.5, Ti:VD4                 // casova integracni konstanta bude 2,5
MOVR    0.01, Td:VD8                // casova derivacni konstanata bude 0,01

LDW=    zavazi:VW1010, 10           // pokud je udaj 10 (rameno se zavazim 10g)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    2.25, Ti:VD4                // casova integracni konstanta bude 2,25
MOVR    0.0125, Td:VD8              // casova derivacni konstanata bude 0,0125

LDW=    zavazi:VW1010, 20           // pokud je udaj 20 (rameno se zavazim 20g)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    2.0, Ti:VD4                 // casova integracni konstanta bude 2,0
MOVR    0.015, Td:VD8               // casova derivacni konstanata bude 0,015

LDW=    zavazi:VW1010, 30           // pokud je udaj 30 (rameno se zavazim 30g)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    1.75, Ti:VD4                // casova integracni konstanta bude 1,75
MOVR    0.0175, Td:VD8              // casova derivacni konstanata bude 0,0175

LDW=    zavazi:VW1010, 40           // pokud je udaj 40 (rameno se zavazim 40g)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    1.5, Ti:VD4                 // casova integracni konstanta bude 1,5
MOVR    0.02, Td:VD8                // casova derivacni konstanata bude 0,02

LDW=    zavazi:VW1010, 50           // pokud je udaj 50 (rameno se zavazim 50g)
MOVR    5.0, Kp:VD0                 // proporcionalni konstanta bude 5.0
MOVR    1.25, Ti:VD4                // casova integracni konstanta bude 1,25
MOVR    0.0225, Td:VD8              // casova derivacni konstanata bude 0,0225
```

Obr. 8.8 Část podprogramu s korigovanými parametry PSD regulátoru

Nakonec tohoto druhého podprogramu je nutné provést přepočet konstant Ci a Cd a také Ci2, neboť v rovnicích určujících jejich hodnoty vystupují adaptací měněné konstanty. Přepočet se provádí tak, jak tomu bylo u prvního podprogramu pro adaptaci pomocí vztahů 4.2 a 4.3. Na obrázku 8.9 vidíme přepočet nové integrační konstanty pro PS regulátor.

```
Vypocet integracni konstanty 2 pro PS regulator  Ci2 = Kp2 * Tvz/Ti2:
LD      SM0.0                       // log.1
MOVR    0.0, Ci2:VD424              // nastavime Ci na nulu
MOVR    Tvz:VD13, Ci2:VD424         // Tvz do Ci2
/R      Ti2:VD304, Ci2:VD424        // Tvz/Ti2
*R      Kp2:VD300, Ci2:VD424        // Ci2 = Kp2 * Tvz/Ti2
```

Obr. 8.9 Výpočet integrační konstanty

Tímto jsou podprogramy pro adaptaci hotové. V následující kapitole využijeme k řešení zadaného problému dotykový displej a vytvoříme pro něj příslušnou aplikaci, která je nutná k funkčnosti podprogramů vytvořených v kapitolách 7 a 8, tedy k adaptaci regulátorů podle závaží na rameni a pro dopředné řízení.

9. Využití dotykového displeje

Jak již bylo naznačeno v kapitolách 7 a 8, budeme k podprogramům v nich vytvořených, potřebovat mimo jiné údaj o hodnotě závaží na rameni. Jelikož hodnotu závaží určuje obsluha ramene, nabízí se možnost v řešení pomocí dotykového displeje, který by komunikoval s automatem. Pomocí něho by obsluha mimo jiné mohla údaj o hodnotě závaží zadat přes dotykové rozhraní, které by jistými kroky tuto hodnotu poslalo přes komunikační port do automatu.

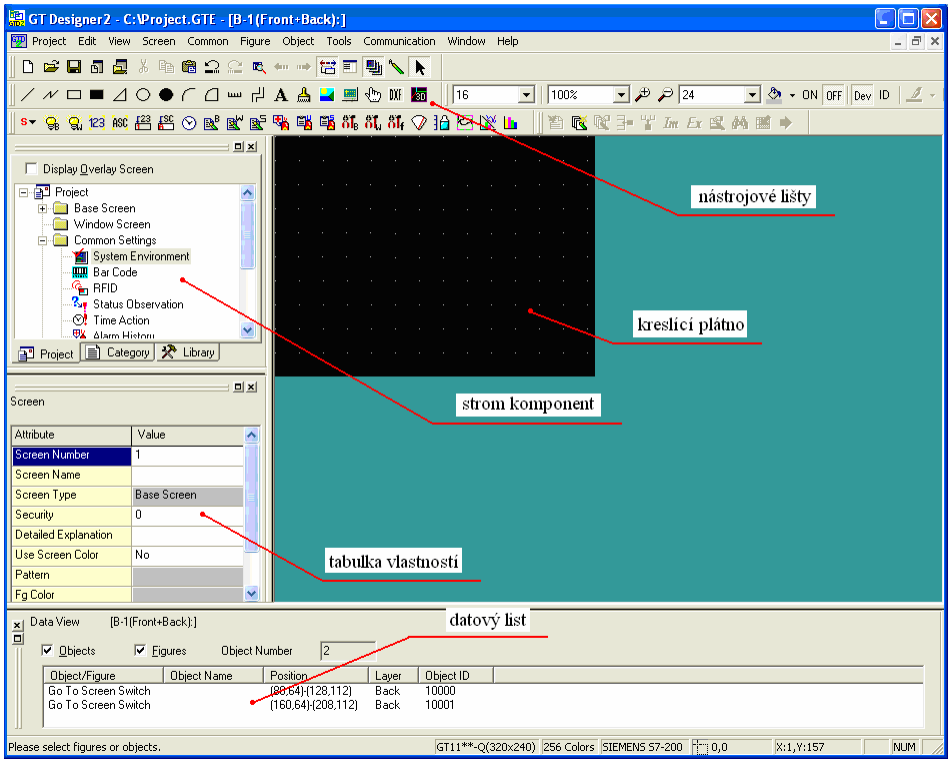
9.1 dotykový display MITSUBISHI GOT1000

K realizaci uživatelského rozhraní bylo využito kompatibility automatu SIMATIC S7-200 s dostupným displejem MITSUBISHI GOT1000 model GT1155 QSBD. Jedná se o barevný display s rozlišením 320 x 240 zobrazovacích bodů. Jeho napájecí napětí je stejně jako u automatu a rozšiřujícího modulu 24V. Využijeme tedy k napájení tohoto zařízení společného zdroje LOGO! Power.

Obr. 9.1 MITSUBISHI GOT1000 model GT1155 QSBD

9.2 GT Designer2

K vytváření aplikací pro tento display poskytuje firma Mitsubishi prostředí GT Designer2. Při spuštění tohoto prostředí se otevře průvodce, který nám pomůže ve výběru typu panelu, nastavení barev a také volby PLC, s kterým bude displej komunikovat. Dále se zde volí mimo jiné například komunikační rozhraní s možností detailního nastavení komunikačních parametrů. Po nastavení potřebných dat v průvodci se otevře pracovní plocha prostředí.



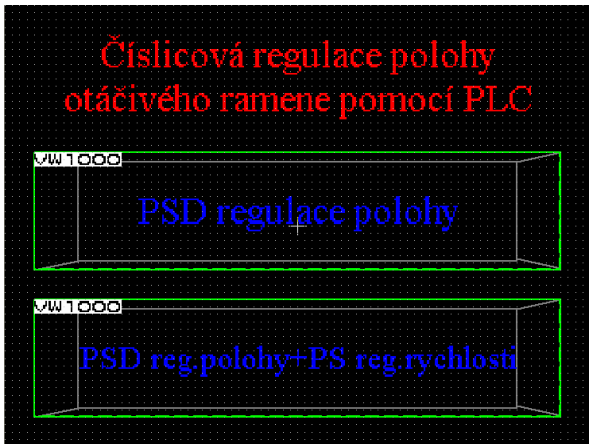
Obr. 9.2 Prostředí GT Designer2

Plocha obsahuje nástrojové lišty, z nichž lze vybírat potřebnou komponentu nebo volit potřebnou funkci pomocí jednotlivých ikon. Dále kreslící plátno, které reprezentuje displej, na které lze umisťovat jednotlivé komponenty tak, jak si představujeme jejich zobrazení na panelu. Tabulka vlastností slouží k prohlížení nebo případné editaci jednotlivých objektů v projektu. Pomocí datového listu můžeme k těmto objektům otvírat jednotlivé formuláře, ve kterých se nastavují příslušné vlastnosti, parametry nebo jejich funkce. Ve stromu komponent můžeme přepínat mezi jednotlivými objekty aplikace nebo mezi již vytvořenými okny. Práce v tomto prostředí je velice intuitivní a přehledná.

9.3 Vytvoření aplikace pro dotykový display

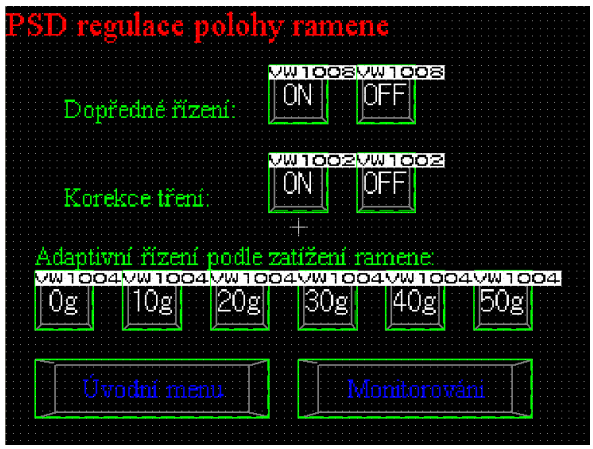
Jak již bylo naznačeno dříve, dotykový display nám bude sloužit pro zadávání údaje o hodnotě závaží na rameni, kterou budeme dále využívat pro dopředné řízení a pro adaptaci parametrů regulátoru podle momentu setrvačnosti, který je funkcí právě této hodnoty. Jelikož v zadání této práce je tvorba několika alternativ regulace, bude výhodné, když pomocí tohoto displeje budeme moci volit, který z již vytvořených dvou programů bude rameno regulovat (zda-li program s jedním PSD regulátorem polohy nebo řešení obsahující dvě smyčky a to pro PSD regulátor polohy a PS regulátor rychlosti natočení ramene). A dále zda-li v jednotlivých programech volat či nevolat jednotlivé podprogramy pro korekci tření, adaptaci nebo dopředné řízení. Tento komplexní program sestavíme z již vytvořených částí v příští kapitole. Nyní se budeme zabývat návrhem aplikace pro dotykový displej v GT Designer2, která bude sloužit k ovládání tohoto programu.

První věc, kterou je třeba navrhnout, je úvodní menu displeje. Jelikož máme vytvořeny dva programy, které mohou každý samostatně regulovat polohu ramene, bude vhodné na úvodní menu dát možnost k výběru jednoho z těchto programů. Ve skutečnosti půjde o výběr jednoho ze dvou různých přerušení, přičemž jedno bude pracovat jako regulační smyčka s jedním regulátorem a druhé jako zapojení se dvěma regulačními smyčkami a s dvěma regulátory. Na obrázku 9.3 je pohled na návrh tohoto úvodního menu.



Obr. 9.3 Úvodní menu

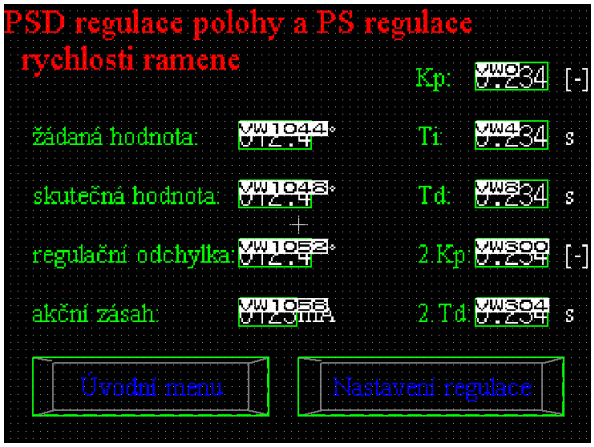
V levém horním rohu obou tlačítek je uvedena adresa bytu VW1000. Jedná se o režijní bajt, který nám bude sloužit k určení, které ze dvou přerušení si uživatel vybral. Při stisku tlačítka pro PSD regulaci se na tuto adresu vloží hodnota 0. Při stisku tlačítka druhého se na VW1000 vloží hodnota 1. Toho budeme později využívat při sestavování komplexního programu. Obě tlačítka jsou typu multi action switch a jsou nastaveny také na přepnutí na další screen. Je tedy nutné vytvořit pro obě možnosti regulace screen další. Na každém z nich bude možnost výběru jednotlivých podprogramů a také pro možnost zadání hodnoty závaží na rameni tak, jak je tomu vidět na obrázku 9.4.



Obr. 9.4 Interface pro PSD regulaci polohy ramene

Je zde možnost výběru, zda vypnout či zapnout blok korekce tření a dopředného řízení. Dále je zde možnost zadat hodnotu závaží, na které se regulátor následně adaptuje. Jsou zde také tlačítka pro návrat do úvodního menu a možnost přejít na další okno, kde bude zobrazeno monitorování vybraných konstant a proměnných. Tyto tlačítka jsou intuitivně popsány jako „Úvodní menu“ a „Monitorování“. U každého tlačítka je v levém horním rohu adresa, stejně jako tomu bylo u tlačítek úvodního menu. Tlačítka popsaná jako ON a OFF při stisku mění barvu, aby bylo na první pohled jasné, jak je regulace momentálně nastavena. Režijní bajty pro komplexní program jsou VW1008 a VW1002. Na adrese VW1010, která není na obrázku vidět, bude vždy hodnota odpovídající zvolené hmotnosti v gramech. Tato adresa se edituje při stisku jednoho z tlačítek s popisem 0,10,20,30,40 nebo 50. Všechny tyto adresy jsou pro nás důležité a při sestavování jednotného programu bude nutné podle hodnot na těchto adresách provádět patřičné kroky v programu.

Při volbě druhého tlačítka na úvodním menu se otevře okno pro přerušení s dvěma regulátory. Toto okno je velmi podobné svým vzhledem a plní stejnou funkci jako okno, které je zde popsáno a proto ho již nebudu uvádět. Poslední dvě okna, která je třeba vytvořit jsou ta, která se objeví při stisku tlačítka „Monitorování“. V tomto okně budeme mít možnost „nahlédnout“, jaké má regulátor momentálně nastavené parametry Kp, Ti a Td a také bude možné zde přímo odečítat hodnotu skutečnou a žádanou, které budou v automatu pro toto použití přepočítány na stupně. Ve stupních budeme moci sledovat i regulační odchylku a akční zásah, který bude uváděn v miliampérech.



Obr. 9.5 Monitorování proměnných a konstant při regulaci

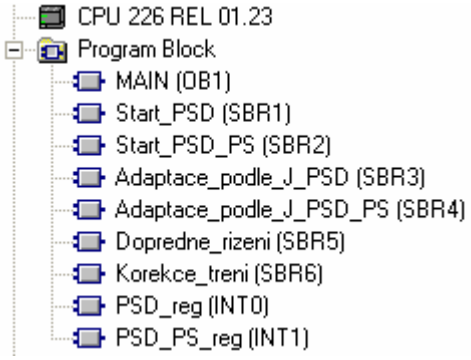
Na obrázku 9.5 je vidět návrh tohoto monitorovacího okna pro variantu se dvěma regulátory. Obsahuje sedm funkcí, které slouží ke čtení hodnoty ze zadané adresy. Proto bude nutné dodržet, aby na námi zvolených adresách byly odpovídající hodnoty proměnných, které jsou u příslušných zobrazení popsány. Okno obsahuje také dvě tlačítka. Jedno pro návrat k úvodnímu menu a druhé pro návrat k oknu, kde se regulace nastavuje.

Analogicky vypadá okno pro monitorování vybraných konstant a proměnných pro variantu s jedním regulátorem. Pravý sloupec hodnot je stejný, v tom levém je o dvě konstanty (Kp2 a Ti2) méně. Samozřejmě obsahuje také tlačítka pro návrat k úvodnímu menu a k nastavení regulace.

.

10. Sestavení komplexního programu

V této kapitole využijeme jednotlivé programy z kapitol 4 a 5, a podprogramy z kapitol 6, 7 a 8 k sestavení komplexního programu. Chod tohoto programu bude možno ovlivňovat pomocí dotykového displeje a aplikace pro něj vytvořené v kapitole 9.



Obr. 10.1 Část instrukčního stromu – program block

Na obrázku 10.1 je část instrukčního stromu, ve které jsou vidět jednotlivé části konečného programu. Obsahuje dvě přerušení, mezi nimiž bude moci uživatel volit hned na úvodním menu dotykového displeje. Dále podprogramy pro korekci tření, adaptaci a pro dopředné řízení. Obsahuje také nové dva podprogramy „Start_PSD“ a „Start_PSD_PS“. Tyto podprogramy budou sloužit, jak již název napovídá, k počátečním úkonům před spuštěním příslušných přerušení. Komplexní program obsahuje také tři tabulky symbolických názvů proměnných. Jejich názvy jsou vidět na obrázku 10.2.

Obr. 10.2 Část instrukčního stromu – symbol table

První dvě tabulky již známe, jsou to tabulky, které jsou na obrázcích 4.2 a 5.2 z předchozích kapitol. Pro komplexní program bylo třeba doplnit část proměnných, které v programu využíváme převážně k režii jeho chodu. Jejich tabulka symbolických názvů je tedy pojmenován jako „production“. Tato tabulka je na obrázku 10.3.

			Symbol	Address	Comment
1			reg	VW1000	pomocna promenna pro urceni zda volat PSD_reg ci PSD_PS_reg
2			treni	VW1002	pomocna promenna pro urceni zda volat Korekci treni ci nikoliv
3			adaptace_PSD	VW1004	pomocna promenna pro urceni zda volat Adaptaci podle J_PSD ci nikoliv
4			adaptace_PSD_PS	VW1006	pomocna promenna pro urceni zda volat Adaptaci podle J_PSD_PS ci nikoliv
5			dopredne	VW1008	pomocna promenna pro urceni zda volat Dopredne rizeni ci nikoliv
6			zavazi	VW1010	udaj o hodnote zavazi na rameni v gramech
7			sinX	VD1012	sinus natoceni ramene v radianech (pro dopredne zizeni)
8			znaminko	VW1016	pomocna promenna pro uchovani znamenka w
9			ukD	VD1018	slozka akcniho zasahu od dopredneho rizeni
10			w_d	VD1024	pomocna promenna pro uchovani hodnoty w v podprogramu dopredne rizeni
11			panel_w_PSD	VD1028	zadana hodnota ve stupnich
12			panel_y_PSD	VD1032	skutecna hodnota ve stupnich
13			panel_e_PSD	VD1036	regulacni odchylka ve stupnich
14			panel_u_PSD	VD1040	akcni zasah v miliamperech
15			panel_w_PSD_PS	VD1044	zadana hodnota ve stupnich pro PSD_PS
16			panel_y_PSD_PS	VD1048	skutecna hodnota ve stupnich pro PSD_PS
17			panel_e_PSD_PS	VD1052	regulacni odchylka pro PSD_PS
18			panel_u_PSD_PS	VD1056	akcni zasah v miliamperech pro PSD_PS

Obr. 10.3 Tabulka proměnných pro „režii“ programu

Hodnoty proměnných na řádcích 1 až 5 jsou editovány pomocí dotykového panelu. Nabývají vždy hodnot 1 nebo 0. Pro hodnotu jedna je příslušný podprogram uživatelem žádán. Výjimku tvoří pouze proměnná „reg“, kde při hodnotě nula je třeba volat přerušení s jedním regulátorem a pro hodnotu jedna s regulátory dvěma. Další nové proměnné na řádcích 11 až 18 slouží k uchovávání hodnot pro panel, respektive jejich hodnoty dotykový displej zobrazuje.

Nyní je třeba v příslušných částech programu zajistit případné volání námi vytvořených podprogramů. Začneme podprogramem „Adaptace_podle_J_PSD“ a „Adaptace_podle_J_PSD_PS“. První z podprogramů je třeba dát k dispozici pro první přerušení, druhý pro přerušení druhé. Jak bylo popsáno v kapitole 4, při přerušení nejprve načteme vstupy (žádanou a skutečnou hodnotu natočení ramene), z těchto vypočítáme regulační odchylku a následuje výpočet akční veličiny pomocí rovnice 4.1 . Před tímto výpočtem je nutné zjistit, jestli obsluha panelu nežádá podprogram pro adaptaci, neboť následující výpočet je závislý na parametrech regulátoru. Informaci o změně hodnoty v proměnné „zavazi“ nám dávají proměnné „adaptace_PSD“ a „adaptace_PSD_PS“, kde každá z nich slouží k informaci pro svoje příslušné přerušení. Je-li hodnota rovna nule, aktuální parametry regulátoru respektive regulátorů odpovídají zadanému závaží. Pokud uživatel provede změnu hodnoty závaží na displeji, proměnná „adaptace_PSD“, respektive „adaptace_PSD_PS“ změní hodnotu na jedničku a je třeba zavolat příslušný podprogram. Na obrázku vidíme část pro přerušení PSD_reg. První instrukce vyjadřuje podmínka, která je-li splněna, voláme již zmiňovaný podprogram. Po jeho skončení se provede další řádek, kde nastavíme hodnotu proměnné „adaptace_PSD“ na nulu, neboť adaptace proběhla a aktuální nastavení odpovídá údaji z displeje. Dokud tedy uživatel hodnotu závaží nezmění, podprogram zůstane nevolán.

```
LDW<>  adaptace_PSD:VW1004, 0      // pokud je promenna adaptace_PSD ruzna od nuly
CALL   Adaptace_podle_J_PSD:SBR3   // zavolame podprogram Adaptace_podle_J
// v tomto podprogramu upravime hodnoty Kp,Ti a Td podle zavazi na rameni
// z techto upravenych hodnot vypocitame take nove hodnoty Ci a Cd (v podprogramu)
MOVW   0, adaptace_PSD:VW1004      // nakonec vlozime 0 do promenne adaptace
// protoze hodnoty jsou upraveny a neni treba podprogram volat do dalsi zmeny zavazi
```

Obr. 10.4 Volání podprogramu pro adaptaci PSD regulátoru

Analogickým způsobem se testuje, zda volat či nevolat podprogram pro dopředné řízení. Tento se ovšem volá, respektive nevolá, až za výpočtem akčního zásahu. Pokud je na displeji pro dopředné řízení stisknuto tlačítko „ON“, volá se tento podprogram v každém cyklu, kde vypočítává potřebný dopředný akční zásah, který je následně přičítán k akčnímu zásahu z rovnice 4.1.Posledním podprogramem k volání je „Korekce_treni“. Ten je k dispozici až na závěr cyklu, kde před výpisem hodnoty na výstup provedeme nebo neprovedeme korekci tření v závislosti na požadavku obsluhy.

Po výpisu hodnoty na výstup je ještě nutné zjistit, zda příští cyklus bude probíhat opět tímto přerušením nebo zda uživatel přepnul na panelu na přerušení druhé. K tomu nám slouží poslední network, který porovnává hodnotu proměnné „reg“ s nulou.

Test zda preruseni neukoncit (zda neni v promenne reg jina hodnota nez 0):

```
LDW<>  reg:VW1000, 0               // jestlize je promenna reg ruzna od nuly
// volame podprgr pro pocatecni nastaveni a pocatecni vypocty pro PSD_PS_reg
CALL   Start_PSD_PS:SBR2
CRETI                              // ukoncime toto preruseni
```

Symbol	Address	Comment
reg	VW1000	pomocna promenna pro urceni zda volat PSD_reg ci PSD_PS_reg
Start_PSD_PS	SBR2	Start_PSD_PS - podprogram volany pri zahajeni preruseni (pouze v prvnim ...

Obr. 10.5 Přepnutí přerušení na druhé a ukončení aktuálního

Pokud tato hodnota není rovna nule, došlo ke změně ze strany uživatele a je nutné přepnout na přerušení druhé (v tomto případě na přerušení s dvěma regulátory). To se provede zavoláním podprogramu Start_PSD_PS, respektive Start_PSD podle toho, v jakém přerušení se právě nacházíme. V tomto podprogramu se, jak již bylo zmíněno výše, přepočítají konstanty pro výpočet akčního zásahu a také připojí druhé přerušení, jak je vidět na obrázku 10.5. Samozřejmě je nutné nastavit jeho interval, který nám reprezentuje proměnná „Tvzms“.

Nastaveni intervalu preruseni, volani a povoleni preruseni PSD_reg (INT0):

```
LD     SM0.0                       // log.1
MOVB   Tvzms:VB12, SMB34           // nastaveni intervalu preruseni na 20ms
ATCH   PSD_reg:INT0, 10            // volame PSD_reg
ENI                                // povoleni preruseni
```

Symbol	Address	Comment
PSD_reg	INT0	Podprogram preruseni pro algoritmus PSD regulace pohohy ramene (s moz...
Tvzms	VB12	vzorkovaci perioda v ms

Obr. 10.6 Připojení přerušení a nastavení jeho intervalu

Po proběhnutí tohoto podprogramu je ještě nutné původní přerušení ukončit. Instrukce pro ukončení přerušení je CRETI. Po této instrukci je toto přerušení zrušeno a čeká se na uplynutí intervalu, aby mohlo začít přerušení další.

Ovládání programu a jeho jednotlivých částí z panelu již máme tedy hotové. Zbývá tedy už jenom upravit, respektive přepočítat, jednotlivé proměnné, které jsou na panelu v okně „Monitorování“ k dispozici v jednotkách, se kterými automat nepracuje. Jedná se o žádanou hodnotu, skutečnou hodnotu a regulační odchylku, které je třeba zobrazovat ve stupních a také akční zásah, který má na panelu násobnou jednotku miliampér.

```
Prepocet velicin na jednotky zobrazovane na panelu:
LD      SM0.0                       // log.1
MOVR    w:VD100, AC0                // w do AC0
*R      360.0, AC0                  // vynasobeni hodnotou 360
MOVR    AC0, panel_w_PSD:VD1028     // hodnotu vlozime do promenne panel_w_PSD

LD      SM0.0                       // log.1
MOVR    y:VD104, AC0                // y do AC0
*R      360.0, AC0                  // vynasobeni hodnotou 360
MOVR    AC0, panel_y_PSD:VD1032     // hodnotu vlozime do promenne panel_y_PSD

LD      SM0.0                       // log.1
MOVR    ek:VD108, AC0               // ek do AC0
*R      360.0, AC0                  // vynasobeni hodnotou 360
MOVR    AC0, panel_e_PSD:VD1036     // hodnotu vlozime d opromenne panel_e_PSD
```

Obr. 10.7 Přepočet veličin na jednotky zobrazované na panelu

Na obrázku 10.7 vidíme část přerušení pro PSD regulaci polohy (analogická část je i u přerušení s dvěma regulátory), ve které přepočítáváme hodnoty na jednotky zobrazované na panelu. Žádanou hodnotu, skutečnou hodnotu a regulační odchylku je třeba na panel zobrazovat ve stupních. Příslušné proměnné w, y a ek mají rozsahy od -0,5 do 0,5 podle aktuální polohy a tento rozsah odpovídá rozsahu ve stupních od -180° do 180°. Pokud tedy jednotlivé hodnoty vynásobíme konstantou 360, výsledná hodnota bude odpovídat poloze ve stupních.

Akční zásah je na panelu třeba zobrazovat v miliampérech. Výstupní rozsah má hodnoty od -16000 do 16000. Hodnota 16000 jednotek odpovídá proudu 2,5A. Podělíme-li tedy aktuální hodnotu akčního zásahu konstantou 6,4, dostaneme hodnotu, která odpovídá akční veličině v miliampérech.

```
LD      SM0.0                       // log.1
MOVW    AC0, AC1                    // akcni zasah z AC0 do AC1
ITD     AC1, AC1                    // konverze integer na double
DTR     AC1, AC1                    // konverze doule na real
/R      6.4, AC1                    // deleni konstantou (prevod na mA)
MOVR    AC1, panel_u_PSD_PS:VD1056  // hodnota v mA pro panel
```

Obr. 10.8 Převod akční veličiny na hodnotu v miliampérech

Na závěr provedeme úpravu editoru datového bloku, kde nadefinujeme pomocí režijních proměnných stav, který budeme požadovat od programu při přechodu automatu z režimu STOP do režimu RUN. Toto nastavení je teoreticky libovolné. Zvolíme tedy

například nastavení, kde budou všechny podprogramy vypnuté a aktivní přerušení bude PSD_reg, tedy poběží regulace s jedním regulátorem polohy.

```
//-----------------------------------------------------------------------------
// Nastaveni parametru pro rezii rizeni:

// pro volani preruseni s PSD reg polohy pro reg = 0
// pro volani preruseni s PSD reg polohy a PS reg rychlosti pro reg = 1
reg:VW1000 0

// pro volani podprogramu Korekce_treni: 0-nevolame podprgr. 1-volame podprgr.
treni:VW1002 0

// pro volani podprogramu Adaptace_podle_J pro PSD regulator polohy
// 0-nevolame podprgr. 1-volame podprgr.
adaptace_PSD:VW1004 0

// pro volani podprogramu Adaptace_podle_J pro PSD reg. polohy a PS reg. rychlosti
// 0-nevolame podprgr. 1-volame podprgr.
adaptace_PSD_PS:VW1006 0

// uhavana hodnota zavazi na rameni
zavazi:VW1010 0

// pro volani podprogramu Dopredne_rizeni: 0-nevolame podprgr. 1-volame podprgr.
dopredne:VW1008 0
```

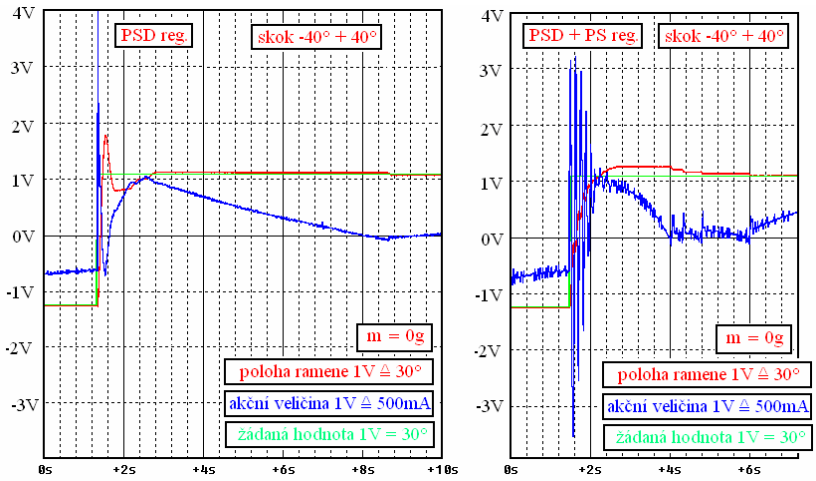
Obr. 10.9 Část editoru datového bloku pro počáteční nastavení režie regulace

Pro libovolnou změnu tohoto nastavení lze v tomto editoru změnit hodnotu patřičné proměnné.

11. Měření a porovnání regulačních vlastností jednotlivých variant regulace

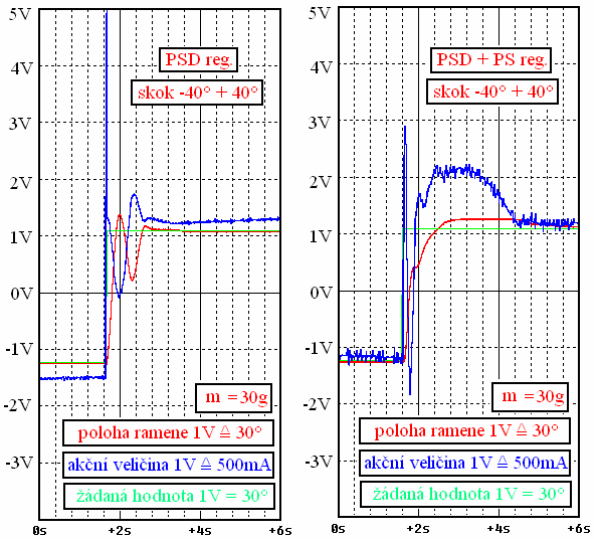
11.1 Zhodnocení kvality regulace PSD reg. polohy a PS reg. rychlosti

V zadaní této práce je návrh několika variant regulace, lišících se především dosaženou kvalitou regulace. V této kapitole tedy zhodnotíme regulační vlastnosti navržených variant regulace. Jako první budeme detailněji zkoumat rozdíl mezi řízením s jedním regulátorem PSD polohy ramene a řízením se dvěma regulačními smyčkami pro polohu a rychlost. Obě tyto varianty budou bez dalších pomocných podprogramů. Tedy nebude možná ani adaptace podle závaží a regulátory budou mít nastaveny parametry, které odpovídají přibližnému optimu s ramenem bez závaží i pro měření se závažím.



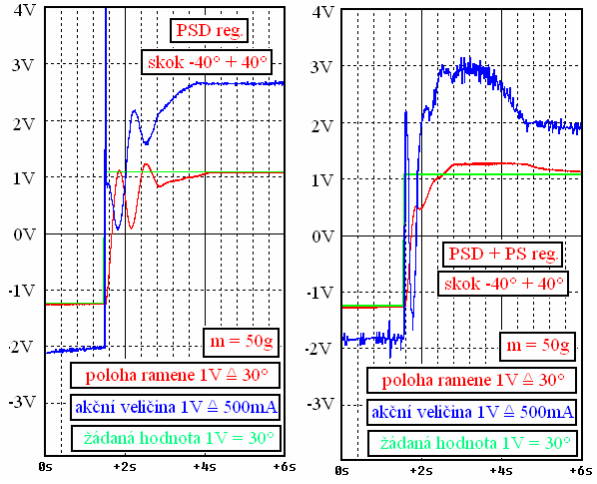
Obr. 11.1 Přechodové charakteristiky pro m=0g

První přechodové charakteristiky na obrázku 11.1 byly měřeny při rameni bez závaží a při skoku žádané hodnoty od -40° do 40°. Jelikož bylo rameno v obou případech bez závaží, vliv tření, který doprovázel celé toto měření, byl velice velký. Přechodový děj pro regulační smyčku s jedním regulátorem (smyčku s regulátorem proudu, který je integrován v panelu [Šabot2002], uvažujeme již jako regulovanou soustavu, tedy tuto smyčku již neuvádím, i když se regulace zúčastňuje), trval 7,35s. V druhém případě (pro regulační smyčku rychlostní i polohovou) trval již jen 3,35s. Znatelně lepší výsledek tedy vykazuje varianta se dvěma regulátory. V přechodové charakteristice s jedním PSD regulátorem je i znatelný překmit, který v případě druhém není.



Obr. 11.2 Přechodové charakteristiky pro m=30g

Na obrázku 11.2 jsou přechodové charakteristiky pro stejné podmínky, jako tomu bylo na obrázku předchozím, s výjimkou přidání závaží na rameno o hmotnosti 30g. V tomto případě je přechodový děj kratší pro variantu s jedním regulátorem. Pro tuto trval 1,35s namísto 3,6s, které trval přechodový děj pro dva regulátory. Z tohoto výsledku lze usoudit, že na změnu podmínek v regulované soustavě bez změny parametrů regulátoru reagovala lépe regulační smyčka s jedním regulátorem. Průběh jeho akční veličiny ukazuje při skoku velkou reakci, která je zapříčiněna derivační složkou.

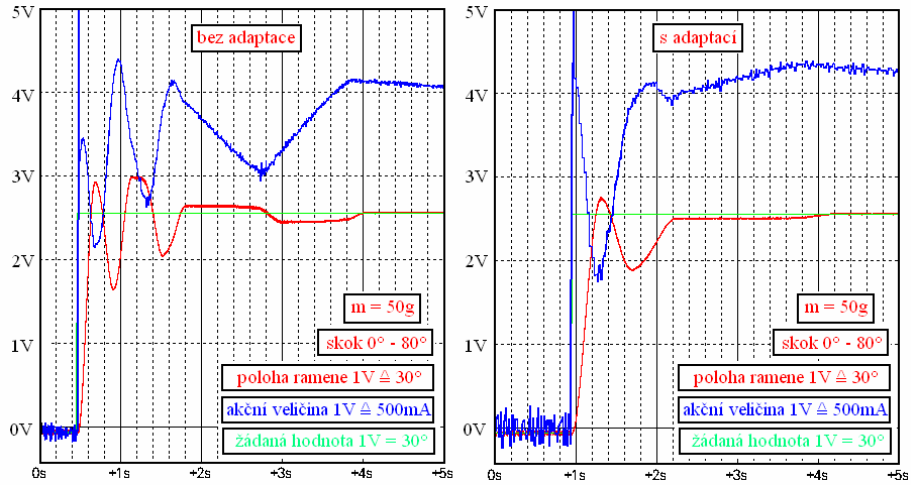


Obr. 11.3 Přechodové charakteristiky pro m=50g

Obrázek 11.3 zobrazuje přechodové charakteristiky se závažím na rameni o hodnotě 50g. Přechodové děje pro obě varianty mají dobu trvání 2,5s pro jeden regulátor a 4,0s pro dva regulátory. Zkouškami prokázaly, že při optimálním nastavením má regulační vlastnosti lepší varianta se dvěma regulátory, tedy polohy a rychlosti. Při zatížení soustavy a tím pádem změně jejich vlastností, se ukázala být lepší varianta s jedním regulátorem polohy.

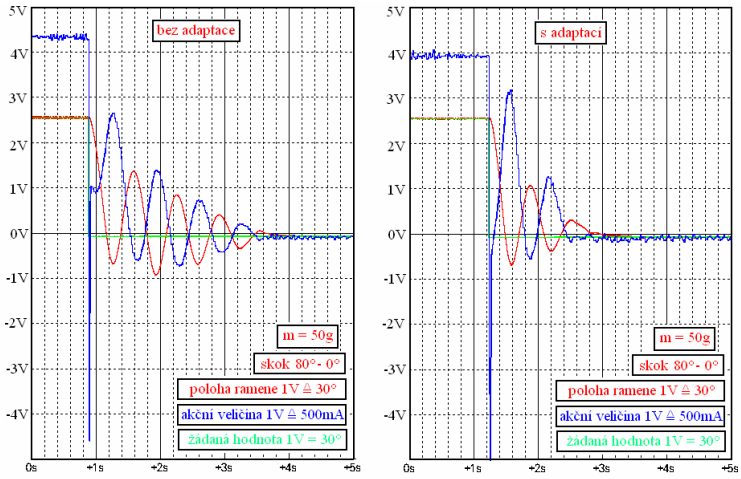
11.2 Zhodnocení vlivu podprogramu pro adaptaci podle momentu setrvačnosti

V kapitole 11.1 byly průběhy, které odpovídají regulaci, bez adaptace parametrů regulátorů. Další přechodové charakteristiky byly prováděny za účelem důkazu vlivu jejich adaptace parametrů podle závaží na rameni.



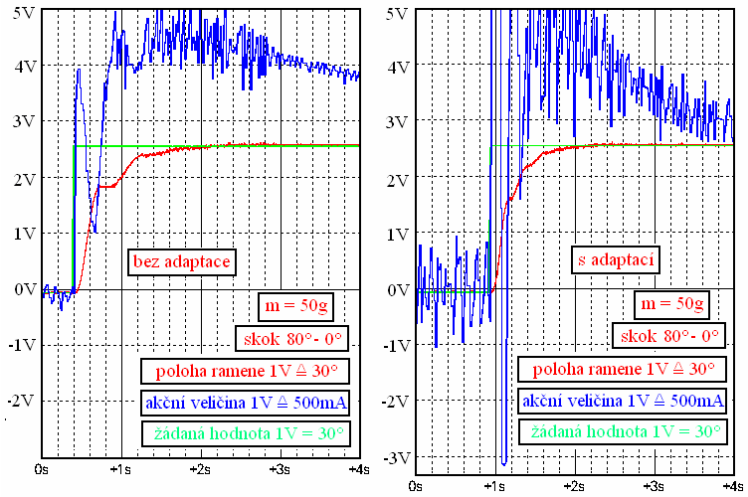
Obr. 11.4 Přechodové charakteristiky pro PSD reg., 0° - 80°

Na obrázku 11.4 jsou přechodové charakteristiky, kde první je pro PSD regulátor bez adaptace a druhá pro tentýž s adaptací. Skoková změna byla z 0° na 80°. Při měření bylo na rameni závaží o hmotnosti 50g. Pro variantu bez adaptace trval přechodný děj 7s. S adaptací na uvedené závaží se doba zkrátila na 2,8s, což dokazuje velký vliv této varianty na zlepšení kvality regulace.



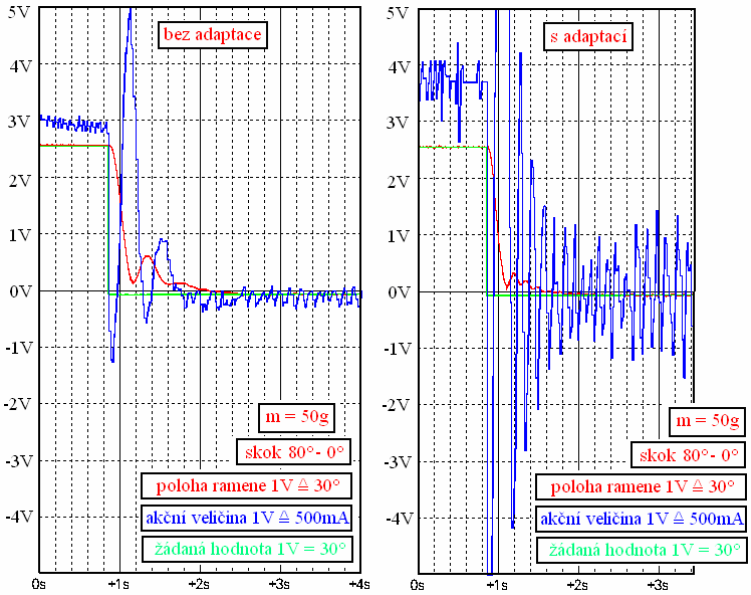
Obr. 11.5 Přechodové charakteristiky pro PSD reg., skok 80° - 0°

Na obrázku 11.5 jsou přechodové charakteristiky pro skok opačné, tedy z 80° na 0°. Závaží i regulační smyčka zůstaly stejné. Přechodové děje mají doby trvání 1,7s a 3s, kde hodnota kratší patří variantě s adaptovanými parametry regulátoru. Opět se potvrzuje kladný vliv adaptace na regulaci s PSD regulátorem polohy.



Obr. 11.6 Přechodové charakteristiky pro PSD a PS regulátory, skok 0° - 80°

Na obrázku 11.6 jsou opět přechodové charakteristiky k určení vlivu adaptace. Hodnoty skoku a hmotnosti závaží jsou patrné z jednotlivých charakteristik. Na rozdíl od předchozích dvou, zde se jedná již o regulaci se dvěma regulátory. I zde je vidět kladný vliv adaptace. Doba přechodového děje ze s adaptací zmenšila z 1,5s na 1s.



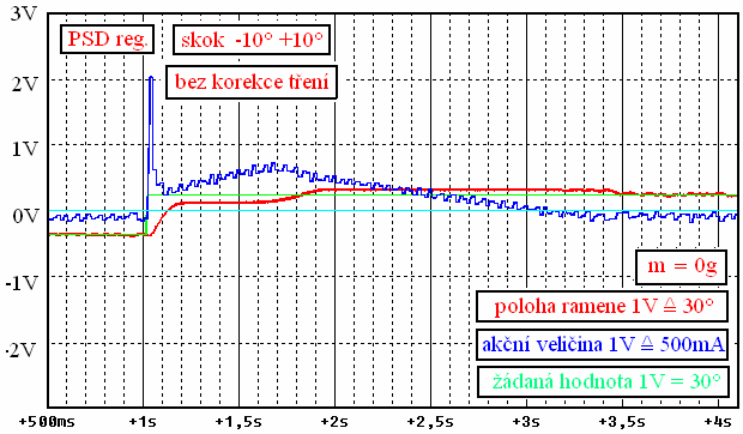
Obr. 11.7 Přechodové charakteristiky pro PSD a PS regulátory, skok 80° - 0°

Tak jako tomu bylo u PSD regulátoru i zde uvádím na obrázku 11.7 i skok zpět, tedy z 80° na 0°. Opět se potvrzuje kladný vliv adaptace. Doba přechodového děje se v tomto případě zkrátila na polovinu. Konkrétně z 1,2s na 600ms.

V této kapitole se potvrdil kladný vliv adaptace parametrů regulátorů na hodnotu momentu setrvačnosti, respektive na hodnotu závaží na rameni.

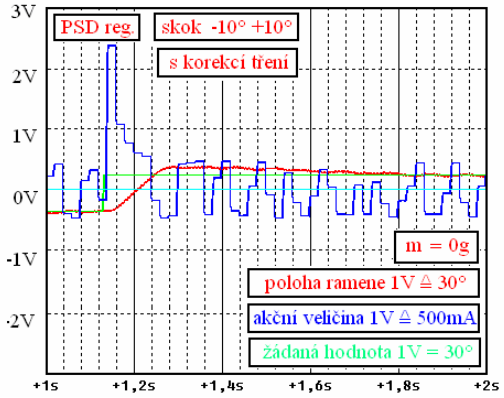
11.3 Zhodnocení vlivu podprogramu pro korekci tření

Zde se podíváme, jak ovlivňuje kvalitu regulace podprogram pro korekci tření. Nejnázornější pro měření vlivu korekce tření bude přechodová charakteristika pro rameno bez závaží a s malým skokem polohy ramene. Na obrázku 11.8 je přechodová charakteristika pro skok -10° +10°. Tato charakteristika je pro regulaci bez podprogramu pro korekci tření. Přechodný děj trval 2,5s.



Obr. 11.8 Přechodová charakteristika bez korekce tření

Na obrázku 11.9 je přechodová charakteristika obdobná, jako je tomu na obrázku předchozím. Výjimku tvoří zapnutý podprogram pro korekci tření při regulaci. Při této korekci doba přechodového děje se zkrátila na bez 500ms (pozn. je nutné si všimnout jiného měřítka časových os). Tedy v tomto případě podprogram pro korekci výrazně ovlivnil rychlost přechodného děje, který urychlil 5krát. Bohužel takový výsledek je patrný pouze při malých změnách polohy a pro rameno bez závaží, kde je vliv tření nejmarkantnější. Při jiných zkouškách vlivu korekce tření se zlepšení regulace nepotvrdilo.



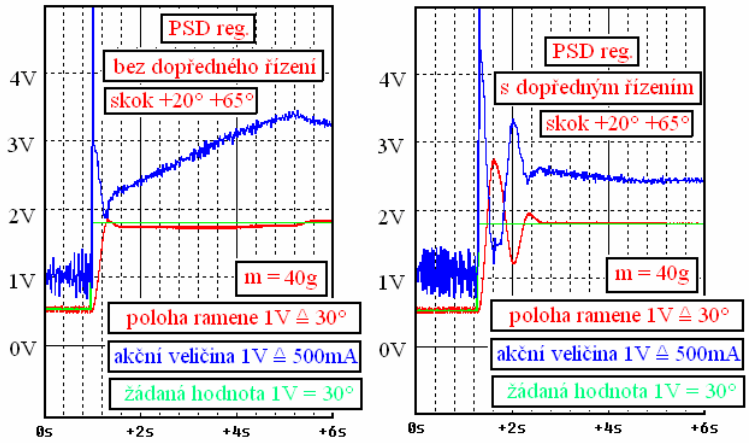
Obr. 11.9 Přechodová charakteristika s korekcí tření

Při bližším zkoumání tření soustavy se ukázalo, že jeho velikost i poloha, kde se vyskytuje, je proměnná. Při jednotlivých měřeních bylo zjištěno, že rameno bez závaží je schopné vlivem tohoto tření dokonce zůstat například ve stacionární poloze 90° při nulovém proudu. Při dalším pohybu ramene a následném poklesu proudu na nulu

rameno zůstává ve stacionární poloze opět jiné a v původní oblasti vysokého výskytu tření rameno volně padá do polohy nulové výchylky. Velikost a vliv tření se tedy ukázal velice dominantní a navíc místo, kde se vyskytuje, je proměnné. Podprogram pro jeho korekci upravuje akční zásah podle průběhu na obrázku 6.4. Jelikož se ovšem tření vyskytuje náhodně a jeho hodnota se výrazně mění není možné jej úspěšně korigovat. Původ tření může být ve spojce, která spojuje snímač polohy s hřídelí motoru. Může mít také původ v nesouososti hřídele a snímače v místě spojky. Další možné místo vzniku tření komutátor.

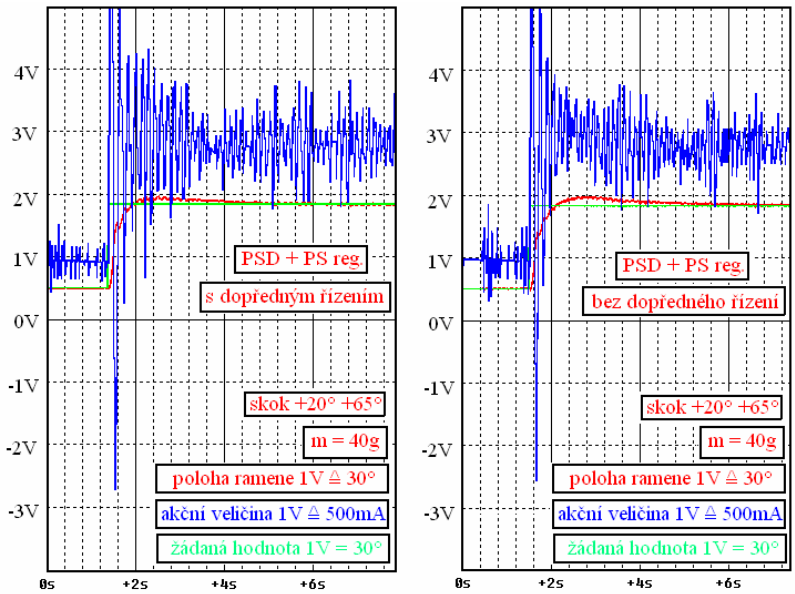
11.4 Zhodnocení vlivu podprogramu dopředného řízení

Jak je popsáno v předchozí kapitole, je tření při rotaci ramene velice intenzivní a jeho velikost je proměnná. Proměnná jsou i místa, respektive polohy, kde tření momentálně působí. Díky tomuto vlivu bude funkčnost i tohoto podprogramu silně ovlivněna.



Obr. 11.10 Přechodové charakteristiky pro dopředné řízení m=40g

Na obrázku 11.10 jsou zobrazeny přechodové charakteristiky pro regulaci s dopředným řízením a bez něj. Charakteristiky jsou pro regulaci s jedním regulátorem. V případě využití dopředného řízení trval přechodný děj 1,3s a bez jeho využití 4,4s. Lze tedy konstatovat, že v tomto případě dopředné řízení výrazně zlepšilo regulaci při tomto skoku. Bohužel při jiných pokusech podprogram již takto příznivě regulaci neovlivňoval, zvláště v případech, kdy byl skok realizován o více jak 60°. Tento jev lze připsat opět výraznému tření, které se chová nevypočitatelně a jeho velikost se mění. V podprogramu pro dopředné řízení se počítá dopředný zásah z naměřených hodnot potřebných k trvalému vychýlení na hodnotu 90°. Proměnlivé tření ovšem tuto hodnotu neustále ovlivňuje a navržený algoritmus nemá možnost se aktuální změně přizpůsobit. Proto dopředný zásah v závislosti na aktuálním tření neodpovídá aktuální potřebné hodnotě.



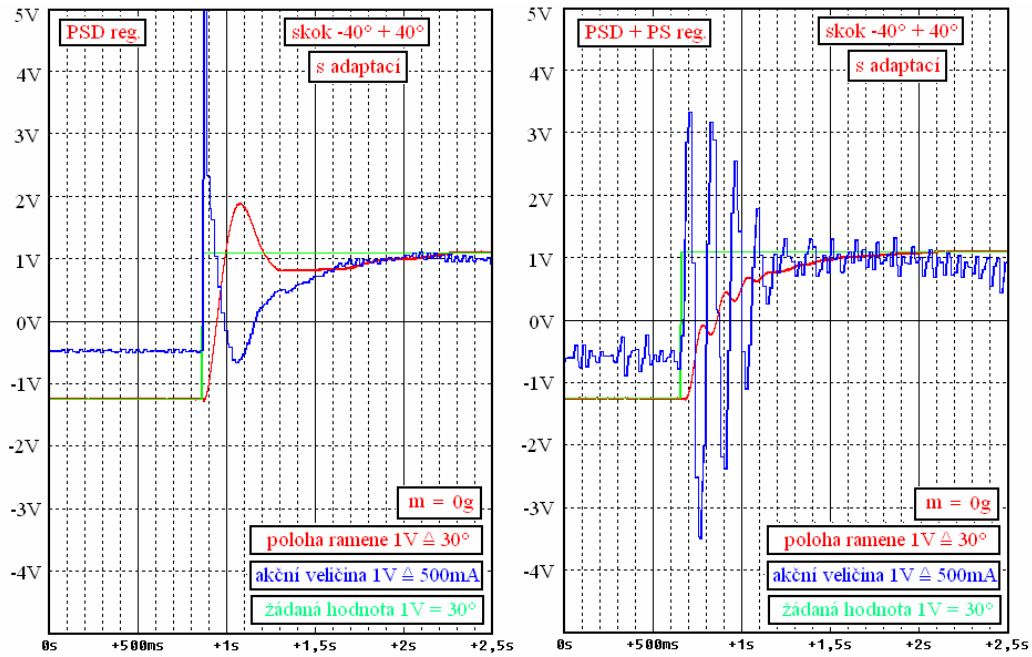
Obr. 11.11 Přechodové charakteristiky pro dopředné řízení m=40g

Na obrázku 11.11 jsou charakteristiky pro regulaci s a bez dopředného řízení pro dvě regulační smyčky. I v tomto případě uvádím příklad, kde tento podprogram přispívá k urychlení přechodného děje. Ovšem stejně jako tomu bylo i u regulační smyčky s jedním regulátorem , i v tomto případě pro větší skoky žádané hodnoty tento podprogram výrazněji nepřispívá ke zlepšení regulačních vlastností díky výraznému vlivu tření.

Vliv podprogramu dopředného řízení nelze tedy opět, jako tomu bylo u korekce tření, jednoznačně určit. Tření soustavy se mění a podle této změny se mění i vliv a přispění tohoto podprogramu ke zlepšení regulačních vlastností.

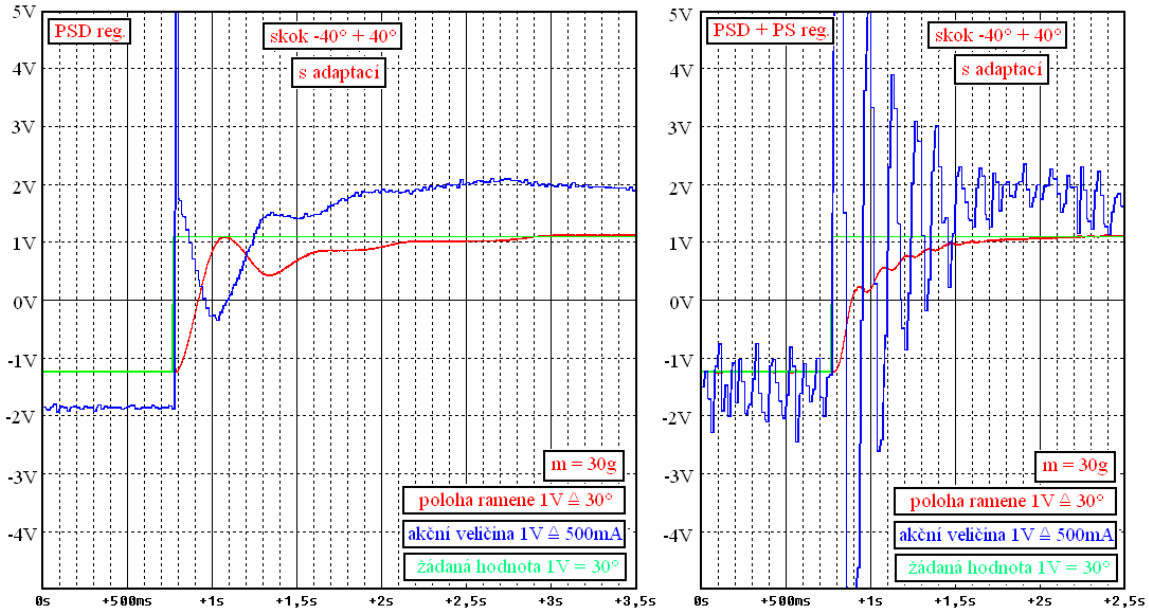
11.5 Porovnání vybraných variant regulace

Díky vlivu tření mají tedy nejstabilnější výsledky kvality regulace varianty s jedním a dvěma regulátory v kombinaci s podprogramem pro adaptaci jejich parametrů. V této kapitole si uvedeme srovnání kvality regulace pro případ s jednou programovou regulační smyčkou pro regulaci polohy a dvěma smyčkami, kde je regulace obohacena o regulátor rychlosti. Vyhodnocení bude opět pomocí přechodových charakteristik pro skoky od -40° do +40° pro hmotnosti závaží 0, 30 a 50 gramů.



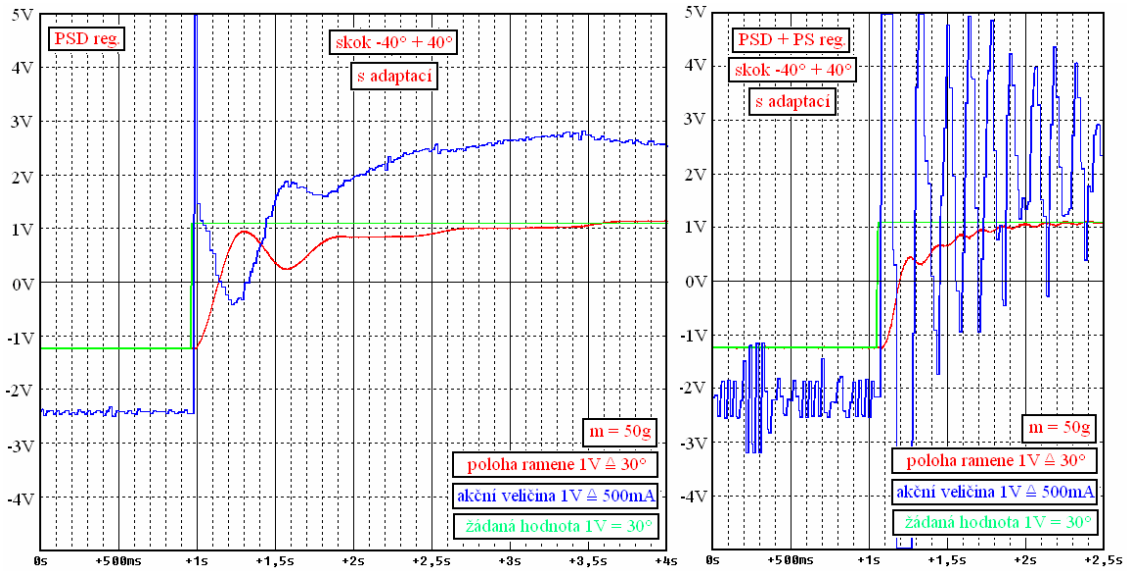
Obr. 11.12 Přechodové charakteristiky pro m=0g

Na obrázku 11.12 jsou přechodové charakteristiky pro rameno bez závaží. Zde je patrný lepší výsledek pro samotný regulátor polohy, kde doba přechodného děje je přibližně 1,35s oproti 1,7s u druhé varianty. Ovšem nevýhodou je značný překmit, který je na škodu zejména při dlouhodobém provozu, kde více zatěžuje akční členy.



Obr. 11.13 Přechodové charakteristiky pro m=30g

Na obrázku 11.13 vidíme srovnání se závažím na rameni o hmotnosti 30g. Zde již přechodný děj je u varianty se dvěma regulátory kratší (1s) oproti druhé variantě (2,1s). Překmit, který byl znatelný u předchozí charakteristiky, zde již patrný není.



Obr. 11.14 Přechodové charakteristiky pro m=50g

Na obrázku 11.14 jsou uvedeny poslední přechodové charakteristiky pro zatížení ramene závažím o hmotnosti 50g. Zde je rozdíl doby přechodných dějů nejmarkantnější. V případě samotné PSD regulace polohy je 2,6s. Pro regulaci obohacenou o regulátor rychlosti je pouze 1s. Charakteristiky jsou opět bez překmitu.

Závěrem této kapitoly lze tedy říci, že varianta s jedním regulátorem polohy nedosahuje takových výsledků jako varianta s regulátory dvěma. Výrazným zlepšením v kvalitě regulace je adaptace podle momentu setrvačnosti. Další podprogramy pro korekci tření a pro dopředné řízení nelze jednoznačně zhodnotit. Na tyto varianty má velký vliv tření, jehož chování, jak je již uvedeno výše, je velmi proměnlivé. Pro různé zkoušky vykazovaly tyto podprogramy různých výsledků, kde v některých ovlivnily regulaci v kladném smyslu, ale v některých i ve smyslu záporném.

12. Závěr

Cílem této diplomové práce byl návrh a realizace číslicové regulace polohy ramene. Bylo nutné analyzovat vlastnosti dané soustavy se svislým ramenem na hřídeli stejnosměrného motoru, navrhnout několik variant řešení, aplikovat vybraná řešení a ověřit kvalitu regulace. Prostředkem řízení měl být programovatelný automat.

V úvodu této diplomové práce jsem popsal zadanou regulovanou soustavu a provedl počáteční nutná měření jejich vybraných vlastností. K číslicovému řešení regulace této soustavy byl vybrán programovatelný automat firmy Siemens, který byl doplněn rozšiřujícím modulem pro dosažení dostatečného počtu možných vstupů a výstupů automatu. Pro tento automat byly vytvořeny dva samostatně pracující programy, z jichž jeden obsahoval jednu číslicovou regulační smyčku s PSD regulátorem polohy ramene a druhý byl rozšířen o další regulační smyčku s PS regulátorem rychlosti ramene. Dále byly vytvořeny podprogramy pro korekci tření, dopředné řízení a pro adaptaci podle momentu setrvačnosti ramene. Dále bylo využito dotykového displeje firmy Mitsubishi, pro který bylo vytvořeno uživatelské rozhraní. Pro toto rozhraní byl vytvořen komplexní program, který se skládá z úvodních dvou programů, navíc doplněných o vytvořené podprogramy.

V závěru této práce je uvedena řada záznamů z měření regulačních vlastností různých kombinací možností regulace spolu s jejich zhodnocením.

SEZNAM POUŽITÉ LITERATURY:

[1] ŠARBORT , Jakub. *Regulační obvod polohy otáčivého ramene*. Brno, 2002. 61 s. VUT v Brně, FSI. Vedoucí diplomové práce doc. Ing. Zdeněk Němec, CSc.

[2] KAVÁN, Martin. *Číslicová regulace polohy otáčivého ramene*. Brno, 2003. 71 s. VUT v Brně, FSI. Vedoucí diplomové práce doc. Ing. Zdeněk Němec, CSc.

[3] ŠVARC, Ivan, ŠEDA, Miloš, VÍTEČKOVÁ, Miluše. *Automatické řízení*. 2007. první vyd. Brno : AKADEMICKÉ NAKLADATELSTVÍ CERM, 2007. 324 s. ISBN 978-80-214-3491-1.

[4] MARTINÁSKOVÁ, Marie, ŠMEJKAL, Ladislav. Řízení programovatelnými automaty. Praha : Vydavatelství ČVUT, 1998. 160s. ISBN 9788001029251.

[5] ROUBÍČEK, Ota. Elektrické motory a pohony. 1. vyd. Praha : BEN - technická literatura, 2004. 195 s. ISBN 80-7300-092-X.

[6] Programovatelný automat S7-200, Systémový manuál. 2004. 515 s.

[7] MITSUBISHI GT Designer2 version 2, Screen Design Manual, (for GOT 1000 series 2003). 2046 s.

[8] SIEMENS [online]. Siemens, 2009 [cit. 2009-05-29]. Dostupný z WWW: <http://www.siemens.cz/cz/ >.

[9] MITSUBISHI [online]. 2009 , 04.05.2009 [cit. 2009-05-29]. Dostupný z WWW: <http://www.mitsubishi-automation-cz.com/>.

[10] AUTOMATIZACE [online]. 2004 [cit. 2009-05-29]. Dostupný z WWW: <http://www.automatizace.cz>.