

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ
TECHNOLÓGIE



KATEDRA INFORMATIZÁCIE A RIADENIA
PROCESOV

RIADENIE VARÁKA REKTIFIKAČNEJ KOLÓNY AKO
SYSTÉMU S INTEGRAČNÝMI VLASTNOSTAMI

Diplomová práca

Dňa: 21.mája 2005
Miesto: Bratislava

Vedúci diplomovej práce: Ing. Magdaléna Ondrovičová
Vypracoval: Bc. Ondrej Palacka



SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta chemickej a potravinárskej technológie

Radlinského 9, 812 37 Bratislava

Katedra: **informatizácie a riadenia procesov**

Číslo: 11/KIRP/2005

Vec: **Zadanie diplomovej práce**

Meno a priezvisko študenta: **Bc. Ondrej Palacka**

Meno a priezvisko vedúcej diplomovej práce: **Ing. Magdaléna Ondrovičová**

Meno a priezvisko konzultanta diplomovej práce:

Názov diplomovej práce:

**Riadenie varáka rektifikačnej kolóny ako systému s integračnými
vlastnosťami**

Termín odovzdania diplomovej práce: **21. mája 2005**

Diplomová práca sa odovzdáva v 3 exemplároch vedúcemu katedry.

Bratislava **1. februára 2005**

Doc. Dr. Ing. Miroslav Fikar
vedúci katedry

Prof. Ing. Dušan Bakoš, DrSc.
dekan

Podakovanie:

Rád by som podakoval Ing. Magdaléne Ondrovičovej za odborné vedenie a rady pri realizácii diplomovej práce. Taktiež by som chcel podakovať rodine za morálnu i hmotnú podporu počas štúdia.

Abstrakt

Úlohou diplomovej práce bolo navrhnúť riadenie pre varák rektifikačnej kolóny s integračnými vlastnosťami. K hlavným programom využitým pri riadení patrí STEP7 a WinCC, ktoré sú pomocou pracovnej stanice SIMATIC S7 - 300 napojené na varák. Ako regulátor bol v STEP7 využitý PID regulátor, ktorý predstavuje blok FB41. Práca sa taktiež zaoberá využitím DDE rozhrania na prepojenie programov MATLAB, Microsoft Excel a WinCC. V MATLABe sa pomocou navrhnutého regulátora vypočíta akčná veličina a pomocou DDE rozhrania a spomenutých programov sa posiela akčný zásah na varák rektifikačnej kolóny. Pri riadení varáku boli využité viaceré typy regulátorov.

Abstract

The main aim of this work was design controllers for the boiler of rectifier convoy with integrating properties. General programs used to control boiler are STEP7 and WinCC, which are connected via workstation SIMATIC S7 - 300 to the boiler. The PID controller was used to control system in STEP7. This is represented by block FB41. The work also deals with DDE interface for connection of MATLAB, Microsoft Excel and WinCC programs. MATLAB is used to design operating value of the spiral power via before mentioned PID controller. This operating value is applied with help of DDE interface to boiler of rectifier convoy. To control of boiler were used various kind of controllers.

Obsah

<i>Úvod</i>	<i>1</i>
1 Riadiaci systém SIMATIC S7 – 300	2
1.1 Postup pri vytváraní riadiaceho projektu	3
2 Programový a konfiguračný software STEP7	4
2.1 Konfigurácia siete a I/O modulov	4
2.2 Programovanie logických blokov pomocou FBD	6
2.3 Spustenie programu a jeho diagnostika	8
3 DDE komunikácia	10
3.1 Vzťah klient – server	10
3.2 Definovanie DDE komunikácie	10
3.3 Možnosti DDE komunikácie	11
3.4 Možnosti MATLABu pri DDE komunikácii	11
3.4.1 Sprístupnenie MATLABu ako serveru pri DDE komunikácii	11
3.4.2 Sprístupnenie MATLABu ako klienta pri DDE komunikácii	12
3.5 Práca s časovačom v MATLABe	13
3.5.1 Parametre a funkcie objektu časovača	13
4 Program WinCC a jeho prepojenie so STEP7	15
4.1 Práca vo WinCC	15
4.1.1 Prepojenie WinCC a STEP7 pomocou tagov	16
4.1.2 Prepojenie WinCC a Microsoft Excel pomocou tagov	17
4.2 Vizualizácia projektu	20
4.2.1 Vytvorenie ovládacích prvkov	20
4.2.2 Tvorba trendov a archívov	22
4.3 Aktivácia WinCC	23
5 Návrh regulátorov, simulácie a priebehy riadenia varáku	24
5.1 Matematický model varáka	24
5.2 Syntéza regulátora metódou umiestnenia pólov pre matematický model varáka	27
5.2.1 Riadenie varáka P regulátorom s využitím STEP7	28
5.3 Padeho aproximácia	30
5.4 Syntéza LQ regulátora	31
5.4.1 Riadenie LQ regulátorom v MATLABe	33
5.4.2 Riadenie varáka LQ regulátorom pomocou DDE rozhrania	34
5.5 Návrh regulátora pomocou optimálnej teórie riadenia	36
5.5.1 Riadenie regulátorom navrhnutým optimálnou teóriou riadenia v MATLABe	38

5.5.2	Riadenie varáka regulátorom navrhnutým optimálnou teóriou riadenia pomocou DDE rozhrania	39
5.6	Návrh PD regulátora metódou umiestnenia pólov pre matematický model po Padeho aproximácii	41
5.7	Vyhodnotenie ukazovateľov kvality riadenia	43
<i>Záver</i>		<i>44</i>
<i>Literatúra</i>		<i>45</i>
<i>Príloha A</i>		<i>46</i>
<i>Príloha B</i>		<i>50</i>

Úvod

Varák ako jedna z hlavných súčastí rektifikačnej kolóny zodpovedná za ohrev kvapalných zmesí pri ich separácii. Z tohto hľadiska je nutné dostatočne ovládať ohrev varáka pre dosahovanie kvalitných výsledkov pri delení zmesi. V súčasnej dobe sa kladie dôraz na čo najväčšie zautomatizovanie systémov a ich následné kvalitné riadenie vzhľadom na požiadavky spotrebiteľa.

Cieľom tejto práce bolo navrhnuť rôzne typy regulátorov na riadenie varáka, sledovanie a vyhodnotenie riadenia pomocou navrhnutých regulátorov. K tomu sa využili produkty SIMATIC S7, Microsoft Corporation a MATLABu.

Diplomová práca sa skladá z piatich kapitol. V prvej kapitole je opísaný riadiaci systém SIMATIC S7 – 300, jeho súčasti a postup pri vytváraní riadiaceho projektu. Druhá kapitola opisuje programový a konfiguračný software STEP7, spôsoby programovania, diagnostiky a spúšťania programu. Tretia kapitola sa zaoberá DDE komunikáciou jej základnými princípmi a funkciami v jednotlivých programoch na ktorých prepojenie bola využitá. Štvrtá kapitola opisuje program WinCC a prácu s týmto programom. Taktiež sa zaoberá prepojením WinCC a STEP7 a prepojením WinCC a Microsoft Excel pomocou DDE rozhrania. Posledná piata kapitola opisuje zostavenie matematického modelu varáku a regulátorov, a simulácie riadenia navrhnutými regulátormi pomocou MATLABu. Navrhnuté regulátory sú tiež aplikované na varák rektifikačnej kolóny. Priebehy riadenia sú zobrazené v tejto kapitole.

1 Riadiaci systém SIMATIC S7 – 300

Na riadenie varáka rektifikačnej kolóny bol použitý riadiaci systém SIMATIC S7 – 300, ktorý pozostáva zo štyroch hlavných častí, ktoré spolu tvoria kompaktný celok.

- **Priemyselný Rack PC 830** patrí do strednej triedy priemyselných PC, postavených na báze technológie od Intelu, vhodný pre používanie v priemyselnom prostredí, ktoré znemožňuje použitie štandardných PC kvôli požiadavkám na odolnosť voči teplotám vibráciám a ďalším vplyvom.
- **Pracovná stanica SIMATIC S7 – 300** je univerzálny PLC schopný pôsobiť v širokom spektre aplikácií automatizačného inžinierstva s dôrazom na produkciu, ekonomické a praktické riešenia v rôznych odvetviach priemyslu. Pracovná stanica obsahuje:
 - *Napájací modul PS 307 5A*
 - *CPU 315 – 2 DP* je centrálna procesorová jednotka s prostredím pre veľkú programovú pamäť s MPI, PROFIBUS interfacom
 - *Modul rozhrania IM 360* umožňuje prenos procesných dát, operátorskú kontrolu a monitoring
 - *Zásuvné vstupno – výstupné I/O moduly* (maximálny počet I/O modulov na jednu pracovnú stanicu S7 – 300 je osem)
- **Prepojenie PC a pracovnej stanice SIMATIC** pomocou MPI (Multi Point Interface) kábla, Profibus, Industrial Ethernet, resp. PTP. Taktiež je možné kombinovať jednotlivé spôsoby a tým vytvoriť sieť, ktorá umožní prenos dát medzi jednotlivými pracovnými stanicami a PC.
- **STEP7 a WinCC** sú softwarové súčasti riadiaceho systému, ktoré slúžia k programovaniu PLC, sprístupňujú dáta užívateľovi. Sú vhodné na monitoring a riadenie reálnych procesov.

1.1 Postup pri vytváraní riadiaceho projektu

Pri vytváraní riadiaceho projektu, je dôležité prihliadať na jeho kompaktnosť, tak aby plnil požadované kritéria a ciele. Pri tvorbe riadiaceho projektu treba splniť nasledovné:

- Konfigurácia I/O modulov, ktorá umožňuje prepojenie medzi pracovnou stanicou a reálnym procesom.
- Konfigurácia siete, ktorá umožňuje prepojenie PC s pracovnou stanicou SIMATIC pomocou zvoleného rozhrania.
- Vytvorenie nového projektu v programe STEP7, v ktorom sú uložené všetky funkcie a dáta potrebné k riadeniu.
- Vytvorenie riadiaceho programu pomocou programovacieho jazyku STEP7, ktorý umožňuje riadiaci program napísať pomocou klasických príkazov, programovaním funkčných blokov alebo kombináciou obidvoch uvedených typov. Vytvorený program po nakopírovaní do RAM pamäte CPU možno kedykoľvek diagnostikovať a logicky testovať aj bez priebehu reálneho procesu.
- Vytvorenie vizualizačného prostredia umožňuje užívateľovi definovať si vlastné vizualizačné prostredie pre daný proces pomocou programu WinCC, ktorého súčasťou je grafický editor. WinCC umožňuje rozmiestnenie jednotlivých ovládacích prvkov, vstupno – výstupných polí a monitorovacích okien na obrazovke užívateľa tak, aby mal neustály prehľad nad priebehom procesu a rýchlymi zásahmi ho mohol aktívne ovplyvňovať.
- Prepojenie riadiaceho programu a vizualizačného prostredia. Vizualizačné prostredie a riadiaci program je potrebné prepojiť. Prepojenie sa uskutočňuje prostredníctvom tagov. Tag je virtuálny dátový kanál, cez ktorý prechádzajú dáta. Jeden „koniec“ tagu je napojený na určitú pamäťovú adresu (tá slúži ako zásobník dát) a druhý koniec tagu tieto dáta sprístupňuje užívateľovi.

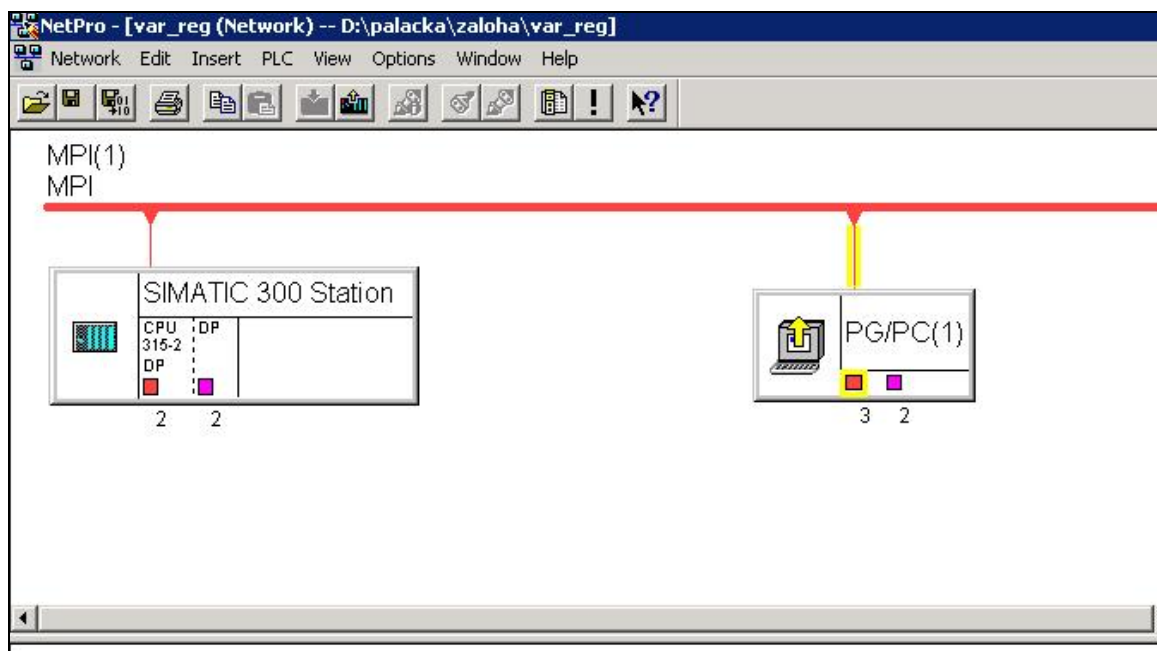
2 Programový a konfiguračný software STEP7

STEP7 sa používa na realizáciu relatívne rozsiahlych a komplexných aplikácií, kde sa vyžaduje pre programovanie, funkcie alebo komunikačné moduly použitie jazykov vyššej úrovne a jazykov grafických návrhov. STEP7 je zlučiteľný s doplnkovým balíkom Engineering tools.

Prvotnou úlohou je vytvorenie nového projektu za pomoci *Project Wizard*, ktorý automaticky vygeneruje jednotlivé podzložky projektu. Po tomto kroku je možné začať jednotlivé zložky modifikovať a dopĺňať o potrebné informácie.

2.1 Konfigurácia siete a I/O modulov

Pri konfigurácii siete sa vytvára komunikačná sieť medzi PC a príslušnými pracovnými stanicami. Základné spojenie pomocou MPI je možné vidieť na obr. 2.1.



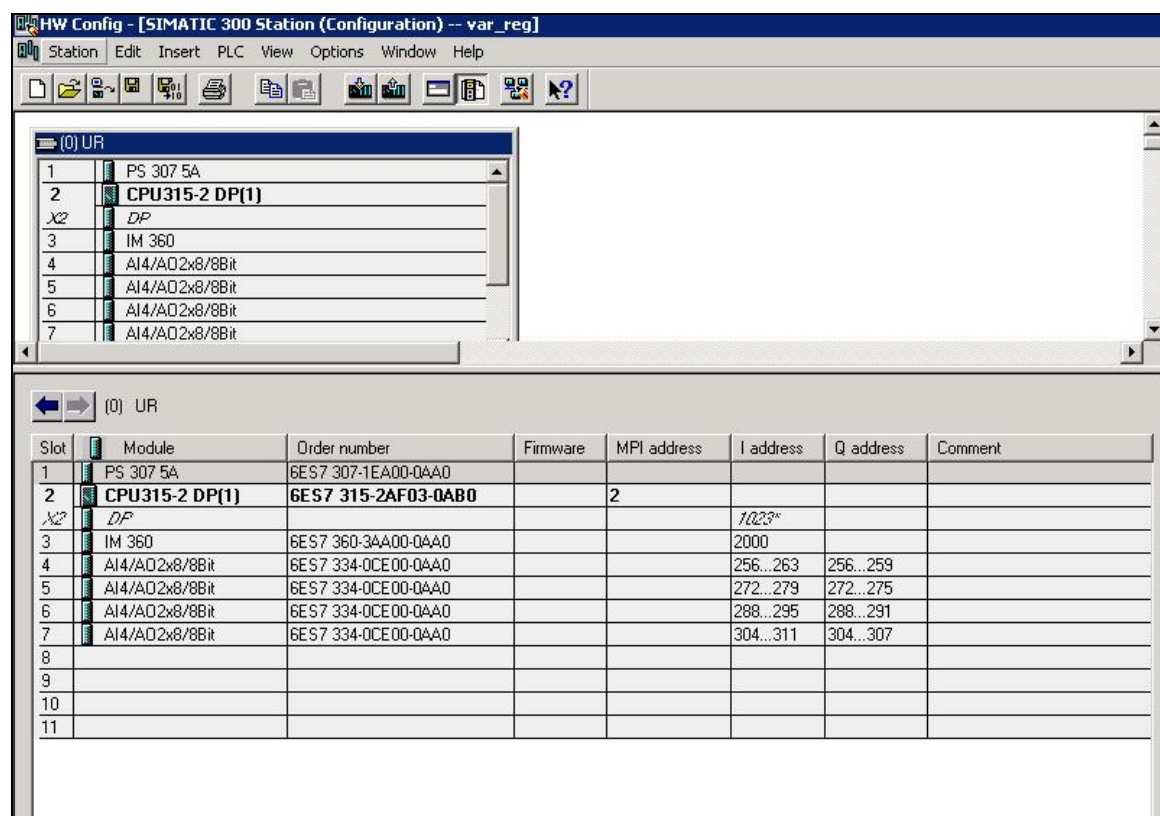
Obr. 2. 1 Zobrazenie nakonfigurovanej siete v STEP7

Pri konfigurácii siete musí mať každý objekt rozdielny NOD. CPU má rezervovaný NOD 2 a PG/PC má rezervovaný NOD 0.

Pomocou riadiacej stanice SIMATIC S7 – 300 je zabezpečené prepojenie snímačov a akčných členov s PC. Analógové vstupno – výstupné I/O moduly, ktoré spracovávajú

v mojom prípade prúdový signál, sú zapojené do riadiacej stanice. K I/O modulom sa pripájajú jednotlivé zariadenia, ktoré sa pomocou SIMATICU dajú riadiť monitorovať a ovládať.

Akčná veličina je výkon špirál a výstupná veličina je teplota vo varáku. Vlastnosti varáku sú popísané v stati 5. Na riadenie špirál varáka slúži triak. Triak pracuje s prúdovým signálom v rozmedzí 4 – 20 mA. Vzhľadom k tomu som na pripojenie triakového meniča výkonu k pracovnej stanici S7 – 300 použil analógové I/O moduly SM 334, AI 4/AO 2 x 8/8 Bit, ktoré majú štyri vstupné a 2 výstupné kanále a rozlišovaciu schopnosť 8 bitov. Meraný a výstupný rozsah je od 0 – 20 mA. Na konfiguračnom okne hardware (obr. 2.2) možno vidieť obsadenie jednotlivých slotov daného railu.



Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 307 5A	6ES7 307-1EA00-0AA0					
2	CPU315-2 DP(1)	6ES7 315-2AF03-0AB0		2			
X2	DP				1023*		
3	IM 360	6ES7 360-3AA00-0AA0			2000		
4	AI4/AO2x8/8Bit	6ES7 334-0CE00-0AA0			256...263	256...259	
5	AI4/AO2x8/8Bit	6ES7 334-0CE00-0AA0			272...279	272...275	
6	AI4/AO2x8/8Bit	6ES7 334-0CE00-0AA0			288...295	288...291	
7	AI4/AO2x8/8Bit	6ES7 334-0CE00-0AA0			304...311	304...307	
8							
9							
10							
11							

Obr. 2. 2 Nakonfigurované okno hardwaru v STEP7

Rail predstavuje celú pracovnú stanicu a jeho riadky jednotlivé sloty, v ktorých sú obsadené jednotlivé moduly. Zariadenie sa na I/O modul pripojí tak, že sa mu priradí špecifická adresa.

2.2 Programovanie logických blokov pomocou FBD

Na programovanie v STEP 7 som využíval FBD (function block diagram), ktorý umožňuje programovanie pomocou blokovej schémy. Ďalšie možnosti programovania sú pomocou STL (statement list), kde sa programuje pomocou príkazov a LAD (ladder logic), ktorý pri programovaní využíva schematické zapojenie. V prípade potreby možno jednotlivé programovacie jazyky vzájomne kombinovať.

V programe STEP7 sa využívajú nasledovné bloky:

- **OB** – organizačné bloky. Udávajú štruktúru užívateľského programu. Ja som využíval blok OB1, ktorý reprezentuje hlavný program, ktorý pracuje v cyklickom režime tak, že si načíta jednotlivé funkčné bloky alebo funkcie.
- **FB** – funkčné bloky, sú samostatne programovateľné a ukladajú sa v nich vnútorné premenné. Logická operácia obsiahnutá v FB bloku sa vykonáva nezávisle na ostatných blokoch.
- **FC** – funkcie, obsahujú algoritmy pre najčastejšie používané funkcie.
- **DB** – dátové bloky, slúžia na uloženie užívateľských dát. Pre každý naprogramovaný FB blok je nutné vytvoriť asociovaný DB blok.
- **SFB, SFC** – systémové funkčné bloky a systémové funkcie. Sú integrované priamo v CPU a umožňujú vstup do niektorých dôležitých systémových funkcií.

Pomocou týchto blokov je možné naprogramovať všetky potrebné logické operácie ako aj riadenie daného reálneho procesu. Ako prvé boli vytvorené FB bloky. V bloku FB1 sa vykonáva prepočet hodnoty premennej z odporového teplomera Pt100 z dátového formátu INT na REAL a taktiež kalibrácia na stupne celzia (°C). Na meranie teploty bol použitý platinový odporový teplomer. Odporový signál prechádza z Pt100 cez prevodníky, ktorý konvertuje odporový signál na prúdový v rozsahu (4 – 20 mA) a cez I/O modul vstupuje do STEP7. Prevod snímaného signálu na stupne celzia je podľa rovnice [9]:

$$y = -74,9899 + 0,01356 x$$

V bloku FB2 dochádza k prevodu výkonu vyhrievacích špirál na výstupný signál podľa rovnice [9]:

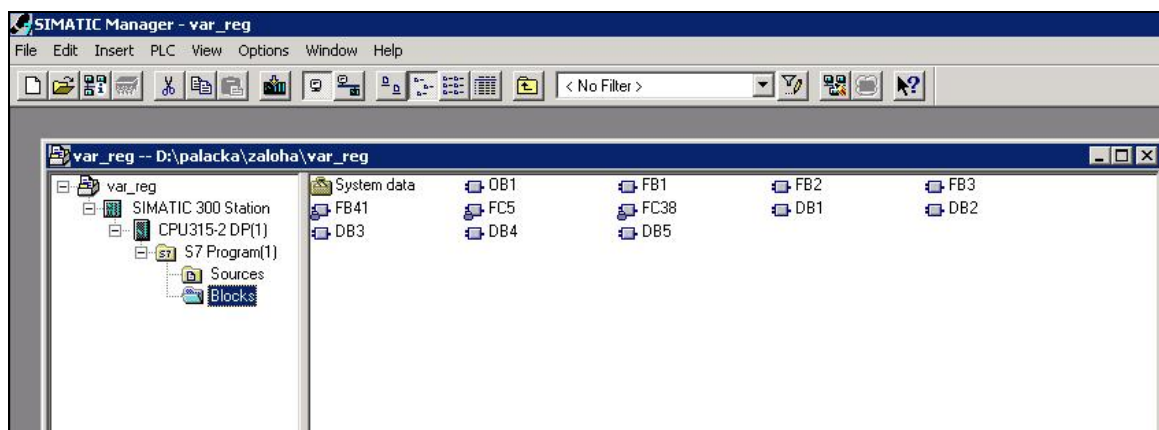
$$y = 5529 + 221,19 x$$

Výkon vyhrievacích špirál je riadený výstupným signálom v rozsahu (4 – 20 mA) privádzaným na triakový menič výkonu. V bloku FB3 sa vykonáva prevod signálu z dátového formátu REAL na dátový formát INT. Pre každý funkčný blok bol vytvorený asociovaný DB blok.

Ako už bolo spomenuté OB blok načíta jednotlivé FB bloky. Do bloku OB1 som začlenil vopred pripravené FB bloky. Po začlenení je treba priradiť bloku vstupné a výstupné signály. Taktiež je treba zdefinovať symbolickú premennú daného DB bloku, ktorý je asociovaný s príslušným FB blokom. Keďže navrhnutý program chcem využívať na riadenie varáka rektifikačnej kolóny je potrebné do bloku OB1 začleniť regulačný blok v mojom prípade FB41. Jedná sa o kontinuálny regulátor, ktorý na základe daných hodnôt zložiek PID regulátora vypočítava výkon ohrevu špirály. Blok FB41 umožňuje jednotlivu vypnutie alebo zapnutie každej zložky regulátora a taktiež prepnutie riadenia na manuál, kedy výkon ohrevu špirály zadáva užívateľ.

Pri programovaní logických blokov som využil spoločný dátový blok DB. Jedná sa o dátový blok do ktorého sa ukladajú premenné ktoré sa využívajú vo viacerých blokoch.

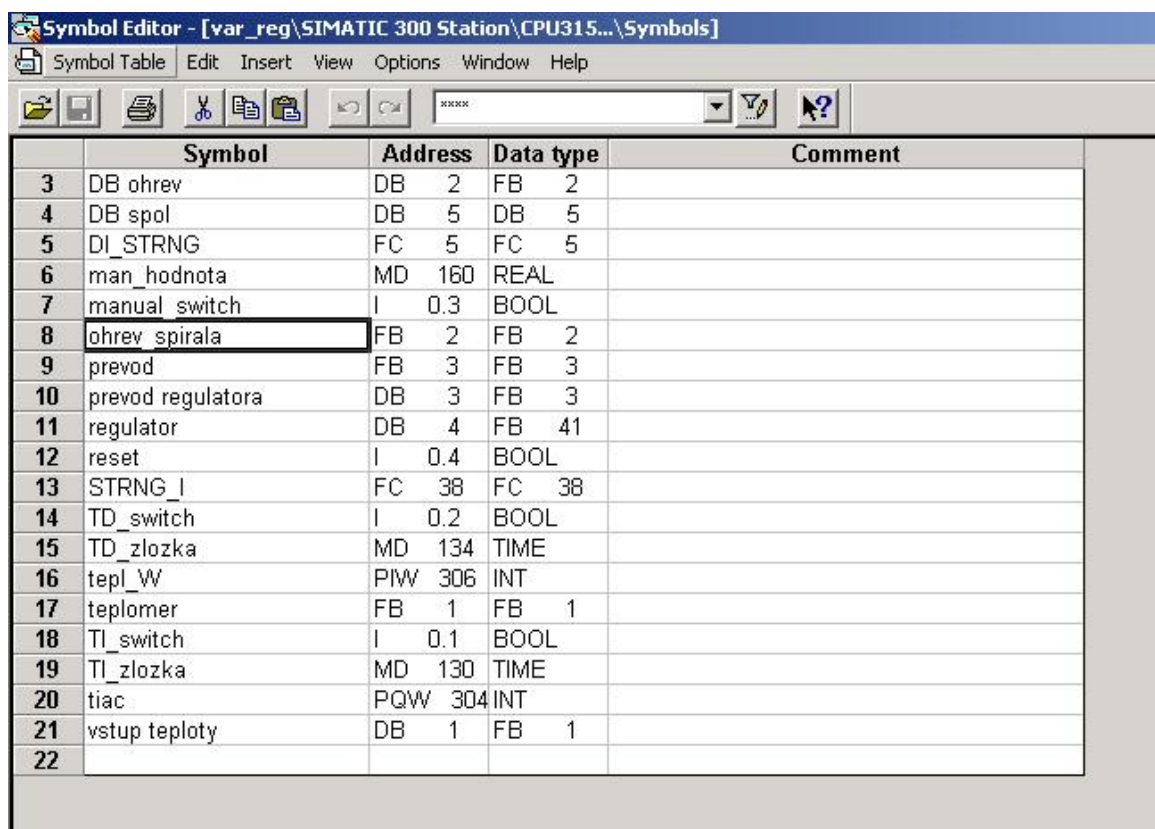
Obsah jednotlivých blokov možno vidieť v prílohe A. Pracovné okno projektu možno vidieť na obr. 2.3.



Obr. 2.3 Okno vytvorených blokov v STEP7

Pre ľahšiu orientáciu medzi použitými typmi blokov sa používa *Symbols editor*. V okne symbol editor-u je možné priradiť symbolickú premennú nielen jednotlivým I/O adresám, ale aj použitým blokom. V stĺpci *Address* je ako absolútna adresa uvedený typ a číslo použitého bloku. Rovnaké označenie je použité aj pre stĺpec *Data type*. Symbolickú tabuľku je možné dopĺňať aj počas programovania jednotlivých blokov. Nie

je potrebné pred programovaním zadať všetky symbolické premenné ktoré budú následne použité. Symbolickú tabuľku možno vidieť na obr. 2.4.



	Symbol	Address	Data type	Comment
3	DB ohrev	DB 2	FB 2	
4	DB spol	DB 5	DB 5	
5	DI_STRNG	FC 5	FC 5	
6	man_hodnota	MD 160	REAL	
7	manual switch	I 0.3	BOOL	
8	ohrev spirala	FB 2	FB 2	
9	prevod	FB 3	FB 3	
10	prevod regulatora	DB 3	FB 3	
11	regulator	DB 4	FB 41	
12	reset	I 0.4	BOOL	
13	STRNG_I	FC 38	FC 38	
14	TD_switch	I 0.2	BOOL	
15	TD_zlozka	MD 134	TIME	
16	tepl_W	PIW 306	INT	
17	teplomer	FB 1	FB 1	
18	TI_switch	I 0.1	BOOL	
19	TI_zlozka	MD 130	TIME	
20	tiac	PQW 304	INT	
21	vstup teploty	DB 1	FB 1	
22				

Obr. 2. 4 Symbolická tabuľka pre varák rektifikačnej kolóny

2.3 Spustenie programu a jeho diagnostika

Na spustenie vytvoreného projektu je potrebné vytvoriť on-line spojenie medzi pracovnou stanicou SIMATIC 300 a PC. Pracovná stanica má 4 pracovné módy:

- **RUN P** – umožňuje testovať program pomocou tabuľky premenných, monitorovaním a zmenou ich hodnôt za chodu
- **RUN** – pracovný režim
- **STOP** – slúži na zastavenie bežiaceho programu
- **M RES** – mód určený na resetovanie pamäte procesorovej jednotky

Pred spustením programu treba nastaviť kľúč CPU315-2DP pracovnej stanice do polohy STOP. Pred nahraním programu do pracovnej stanice treba skontrolovať kontrolné LED diódy. Ak je všetko v poriadku tak by mala svietiť zelená LED dióda zapnutého

zdroja, ďalej zelená LED (DC5V) a oranžová LED dióda pre nastavenie kľúča do polohy STOP.

Na vytvorenie on-line spojenia je potrebné nakopírovať bloky zo STEP7 do RAM pamäte procesora. To sa môže spraviť voľbou PLC Download. Po nahraní programu do pamäte procesora je treba nastaviť kľúč pracovnej stanice do polohy RUN alebo RUN P. Vytvorený program možno otestovať v RUN alebo RUN P móde. Testuje sa tak, že sa treba nastaviť do zvoleného bloku a pomocou ikony *Monitor (on/off)* je možné vidieť ako jednotlivé funkčné bloky fungujú. Ak logická operácia v bloku prebieha, je blok vysvietený zelenou farbou. V prípade že logická operácia neprebieha blok je vysvietený modrou prerušovanou čiarou. Ak je blok zle pripojený alebo je nesprávne definovaná určitá premenná k nemu prislúchajúca je vysvietený červenou farbou. V RUN P móde na rozdiel od RUN módu možno ovplyvňovať program počas jeho chodu. Do bloku *Variable Table* možno definovať premenné, ktoré treba monitorovať alebo modifikovať až na premenné typu PQW, ktoré nemôžu byť monitorované a premenné typu PIW modifikované.

3 DDE komunikácia

DDE (Dynamic Data Exchange) umožňuje aplikáciám zriadiť spojenie a používať ho na prenos dát. Jedná sa o vzájomnú komunikáciu aplikácií na báze operačného systému MS Windows. Aplikácie komunikujú formou „rozhovoru“ v reálnom čase. Jedna z aplikácií je označovaná ako server a druhá ako klient.

3.1 Vzťah klient – server

Server - poskytuje informácie a prijíma požiadavky od inej aplikácie zaujímajúcej sa o dáta. Server je pasívny v konverzácii a odpovedá iba žiadostiam o informácie od aktívneho partnera t. j. klienta.

Klient - posiela požiadavky a riadi proces komunikácie. Riadi komunikačný kanál, inicializuje jeho otvorenie, posiela požiadavky a opäť ho zatvára.

Každá aplikácia môže pôsobiť ako server pre viac klientov alebo ako klient pre viac serverov a zároveň môže byť serverom aj klientom. Úlohou servera je posielat' dáta klientovi, úlohou klienta je zahájenie rozhovoru, požiadanie serveru o dáta, požiadanie serveru o vykonanie príkazov (execute). Keď je rozhovor aktívny, môže klient požiadať server o dáta (request) alebo spustiť avizovaný cyklus. Klient môže požiadať server aby ho informoval o každej zmene určitej časti dát. Server doručí potom oznámenie o zmene pri každej zmene.

3.2 Definovanie DDE komunikácie

Konverzácia medzi aplikáciami pomocou DDE sa zabezpečuje pomocou troch hlavných parametrov. Tieto parametre sú *Názov služby*, *Názov témy* a *Názov prvku*, ktoré musí poslať klient. *Názov služby (service)* je názov aplikácie serveru DDE resp. názov spustiteľného súboru bez prípony alebo iný názov určený serverom. *Názov témy (topic)* definuje subjekt DDE konverzácie, t. j. predmet záujmu napr. dátový súbor, okno serveru. *Názov prvku (item)* je identifikátor špecifického dátového prvku, t. j. konkrétne položky o ktoré má klient záujem napr. bunka tabuľky, obsah poľa. Parametre *Názov služby* a *Názov témy* sa počas konverzácie medzi klientom a serverom nemôžu meniť. Server po prijatí požiadavky na konverzáciu potvrdí jej prijatie a vytvorí DDE spojenie. Spojenie možno ukončiť zavretím jednej z aplikácii klient alebo server.

3.3 Možnosti DDE komunikácie

Poznáme tri typy komunikačných liniek medzi serverom a klientom. Rozlišujú sa podľa spôsobu sprístupňovania dát na:

- **Hot link** (automatic) - spojenie v prípade, že sa dáta na serveri zmenia, zmena sa automaticky prenesie na klienta. To znamená, že server oznámi každú zmenu hodnoty klientovi. Zmeny sú posielané dovtedy, kým klient neukončí dátové spojenie so serverom.
- **Warm link** (notify) - pri zmene dát sprístupňovaných na serveri je klientovi zaslaná správa o zmene a ten má potom možnosť rozhodnúť sa, či zmenené dáta prijme a ak áno vyžiada si ich od serveru.
- **Cold link** (manual) - Jedná sa o jednorázový príjem dát zo serveru pri jednorázovej požiadavke klienta. Klient pošle žiadosť serveru pre prenos zmenených dát. Pri zmene dát na serveri klient nie je upozornený a je plne zodpovedný za aktualizáciu svojich dát. V tomto prípade komunikácie nie je záruka, že všetky zmeny dát budú zachytené.

3.4 Možnosti MATLABu pri DDE komunikácii

MATLAB umožňuje komunikovať pomocou DDE rozhrania s ďalšími aplikáciami buď ako server alebo klient.

3.4.1 Sprístupnenie MATLABu ako serveru pri DDE komunikácii

Ak chceme využiť MATLAB pri DDE komunikácii ako server musíme pre *Názvu služby* zadať „matlab“. MATLAB ako server umožňuje DDE komunikáciu pomocou dvoch tém a to systémovej „system“ a dátovej „engine“.

Do systémovej skupiny patria prvky [4]:

- SysItems – poskytuje zoznam prvkov podporovaných témou „system“
- Format – poskytuje zoznam všetkých formátov, ktoré MATLAB podporuje ako server
- Topics – poskytuje zoznam tém podporovaných serverom

Do dátovej skupiny patria prvky:

- EngEvalString – príkaz na zaslanie dát z MATLAB-u po vyžiadaní
- EngStringResult – výsledok po zaslaní dát z MATLAB-u

- EngFigureResult – grafické zobrazenie po zaslaní dát z MATLAB-u
- <matrix name> - názov premennej pri posielaní a prijímaní dát MATLABom



Obr. 3. 1 Komunikácia MATLABu ako serveru

3.4.2 Sprístupnenie MATLABu ako klienta pri DDE komunikácii

MATLAB je možné využívať pri DDE komunikácii ako klienta. Pri tejto komunikácii sa využívajú príkazy, ktoré umožňujú pripojenie k zvolenému DDE serveru.

Medzi hlavné využívané funkcie patria [4]:

- ddeadv – nastavenie upozorňujúcej slučky so serverom
- ddeexec – vykonanie príkazu serverom
- ddeinit – inicializácia komunikácie medzi MATLABom a DDE serverom
- ddepoke – poslanie dát z MATLABu serveru
- ddereq – žiadosť o dáta zo serveru
- ddeterm – ukončenie DDE komunikácie medzi MATLABom a serverom
- ddeunadv – ukončenie upozorňujúcej slučky zo serverom



Obr. 3. 2 Komunikácia MATLAB-u ako klienta

Tento typ komunikácie som využil v mojom prípade. Program možno vidieť v prílohe B.

3.5 Práca s časovačom v MATLABe

Pri DDE spojení využívam v MATLABe objekt časovača, ktorý umožňuje plánovať zvolené úlohy podľa požadovaného poradia. Pri spustení objektu časovača dôjde k zaradeniu zvolenej úlohy do poradia na jej vykonanie. V mojom prípade som využil objekt časovača na spúšťanie výpočtu akčného zásahu pomocou zvoleného regulátora v požadovaných intervaloch.

Objekt časovača sa definuje pomocou funkcie *timer*, ktorá je potrebná pre jeho vytvorenie. Ďalej sa definujú vlastnosti časovača, podľa požiadaviek užívateľa. V mojom prípade išlo o periodické spúšťanie výpočtu. Prácu s časovačom ukončíme pomocou príkazu *stop* a pre definitívne ukončenie práce je potrebné časovač vymazať z pamäti príkazom *delete*.

3.5.1 Parametre a funkcie objektu časovača

Pri vytváraní objektu časovača pracujeme s príkazmi, ktoré určujú správanie sa časovača. Môžeme ich rozdeliť do dvoch hlavných skupín na funkcie a parametre [4].

Funkcie určujú pri akej udalosti sa majú vykonať a sem patria:

- *ErrorFcn* – funkcia sa vykonáva v prípade chyby
- *StartFcn* – funkcia sa vykonáva pri štarte časovača
- *StopFcn* – funkcia sa vykonáva pri zastavení časovača
- *TimerFcn* – obslužná funkcia sa vykonáva časovačom v každom kroku

Parametre definujú nastavenia a momentálny stav časovača sem môžem zaradiť:

- *AveragePeriod* – určuje priemernú periódu vykonávania obslužnej funkcie
- *InstantPeriod* – čas medzi poslednými dvomi vykonaniami obslužnej funkcie
- *Running* – určuje aktuálny stav časovača
- *TaskExecuted* – určuje počet vykonaní obslužnej funkcie od spustenia
- *TaskToExecute* – určuje koľko krát by mal časovač vykonať funkciu definovanú v *TimerFcn*
- *ExecutionMode* – určuje spôsob zoradovania obslužnej funkcie
- *BusyMode* – určuje spôsob ďalšieho spracovania obslužnej funkcie, ak posledné spracovanie nebolo ukončené
- *Period* – určuje periódu medzi dvoma vykonaniami obslužnej funkcie

- *StartDelay* – určuje časový interval pred prvým vykonaním funkcie v sekundách
- *Name* – textový reťazec, ktorý určuje užívateľ
- *Tag* – popis, ktorý určuje užívateľ
- *UserData* – dáta, ktoré určuje užívateľ

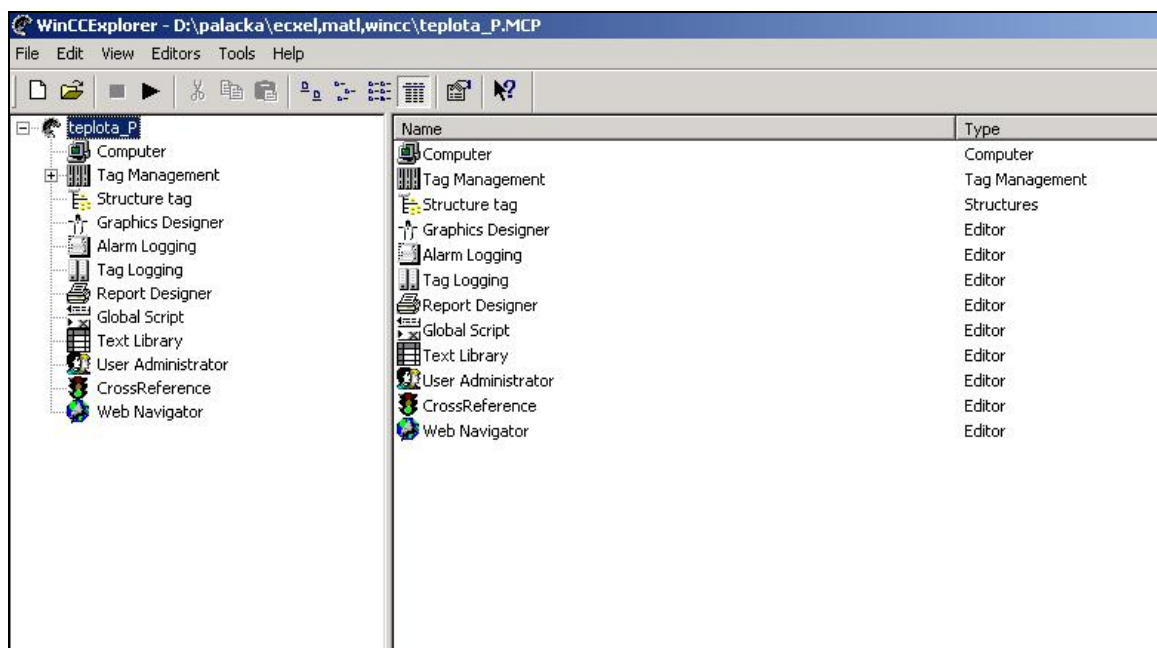
Zvolil som si periódu posielania akčného zásahu každú sekundu. Program časovaču možno vidieť v Prílohe B.

4 Program WinCC a jeho prepojenie so STEP7

Vizualizačný software slúži na vizualizáciu programu STEP7. Umožňuje vytvorenie grafického prostredia v ktorom možno sledovať a ovládať prebiehajúce procesy pričom jednotlivé dáta čerpá alebo posiela prostredníctvom STEP7, s ktorým je prepojený pomocou tagov.

4.1 Práca vo WinCC

Skôr ako začneme pracovať v programe musíme nadefinovať rozhranie, pomocou ktorého bude WinCC komunikovať s PLC. Vytvorí sa nový projekt *File – New – Single-User Projekt* tu treba zvoliť meno nového projektu. Na Tag Management-e sa zvolí položka *Add New Driver* a z následnej ponuky sa zvolí súbor *SIMATIC S7 Protocol Suite.CHN*. V ponuke Tag Management-e sa objaví nový podstrom *SIMATIC S7 PROTOCOL SUITE* [1]. WinCC automaticky vygeneruje okno s jednotlivými modulmi, ktoré tvoria jeho podzložky (obr. 4.1).



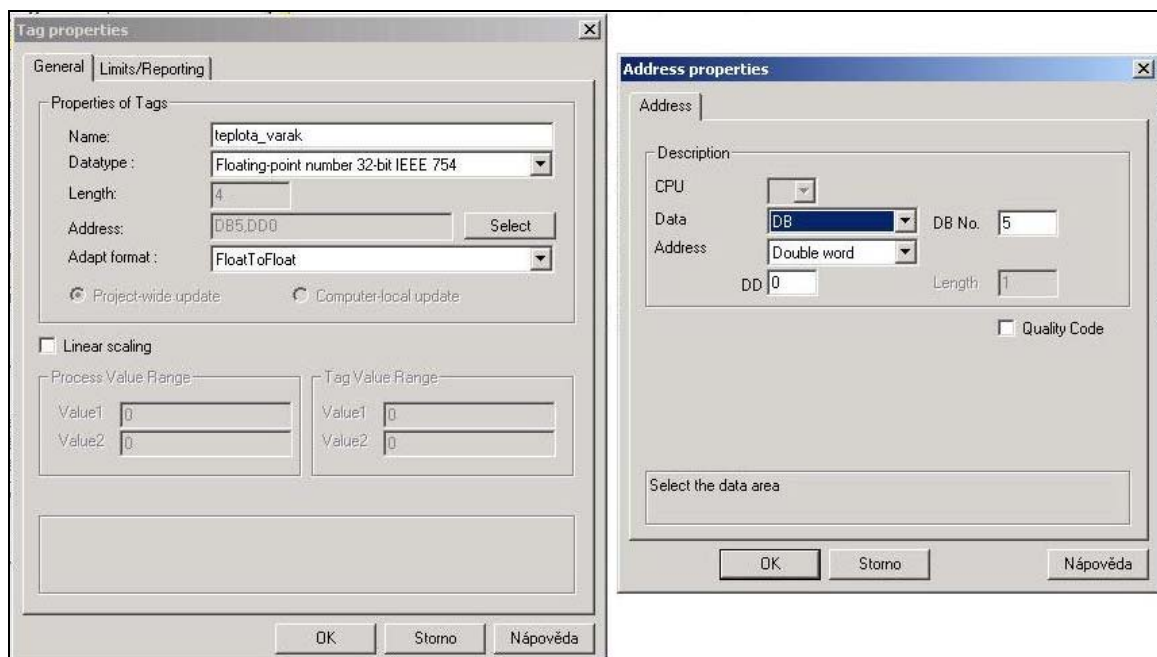
Obr. 4. 1 Zobrazenie hlavných položiek vo WinCC

WinCC pozostáva z viacerých modulov, v ktorých sa podľa potreby definujú jednotlivé požiadavky. V mojom prípade medzi najčastejšie využívané patria:

- *Computer* – umožňuje nastavovať spúšťanie programu podľa požiadaviek užívateľa
- *Tag Management* – slúži na definovanie rozhrania a tagov pomocou ktorých je WinCC spojené s ďalšími programami.
- *Graphics Designer* – jedná sa o grafický editor, v ktorom možno zobrazovať dané zariadenie a taktiež sledovať proces pomocou grafov alebo polí ktoré ukazujú nami žiadané hodnoty.
- *Tag Logging* – je určený na vytváranie archívov, pomocou tagov zadefinovaných v Tag Management-e. Do archívov sa ukladajú sledované veličiny v čase, ktorého cyklus volíme podľa potreby.
- *Global Script* – umožňuje písanie programov v programovacom jazyku C

4.1.1 Prepojenie WinCC a STEP7 pomocou tagov

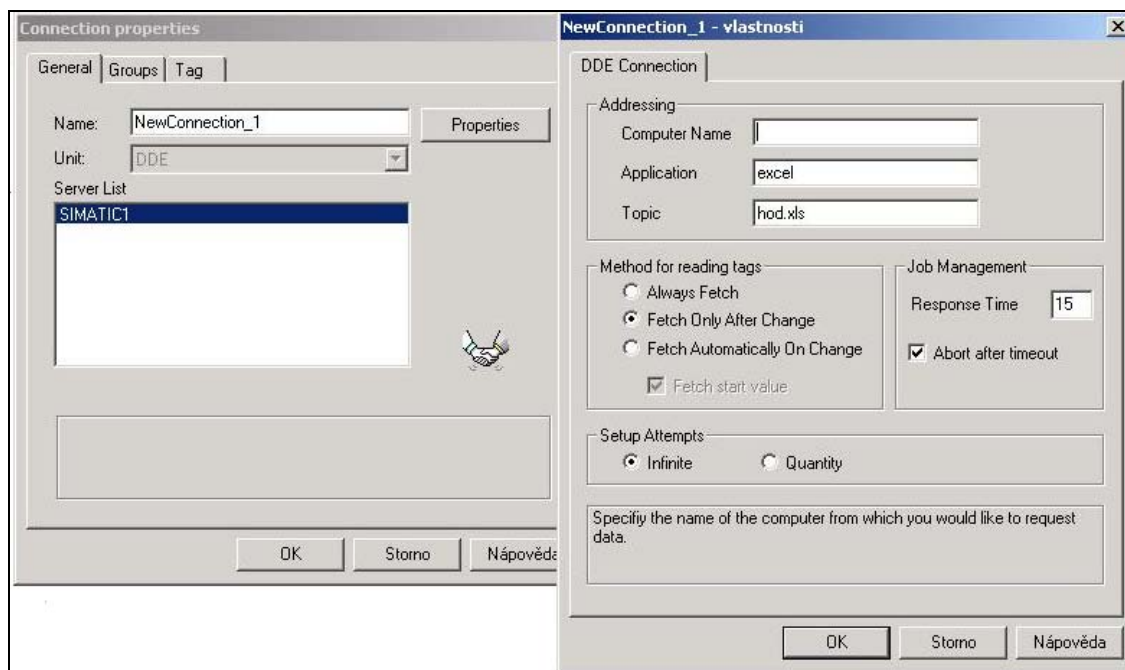
Prepojenie WinCC s riadiacim programom STEP7 sa robí pomocou tagov. Tag je dátový kanál, cez ktorý prechádzajú dáta.. Ako bolo spomenuté tagy sa vytvárajú v Tag Managment-e po pridaní komunikačného driver-u *SIMATIC S7 PROTOCOL SUITE*. Pri vytváraní tagu sa volí jeho názov a definuje typ premennej, ktorú tag využíva a adresa pomocou ktorej bude daný tag spojený so STEP7. Pre riadenie varáku rektifikačnej kolóny boli vytvorené tagy pre snímanie teploty, výkonu špirály, žiadanej teploty, výpočet rozdielu medzi žiadanou a meranou teplotou, zložky regulátora PID s možnosťou zapínania a vypínania jednotlivých zložiek a tag pomocou ktorého možno manuálne ovládať výkon špirály. Okná tvorby tagu možno vidieť na obr. 4.2.



Obr. 4. 2 Okná tvorby tagu pre snímanie teploty vo varáku

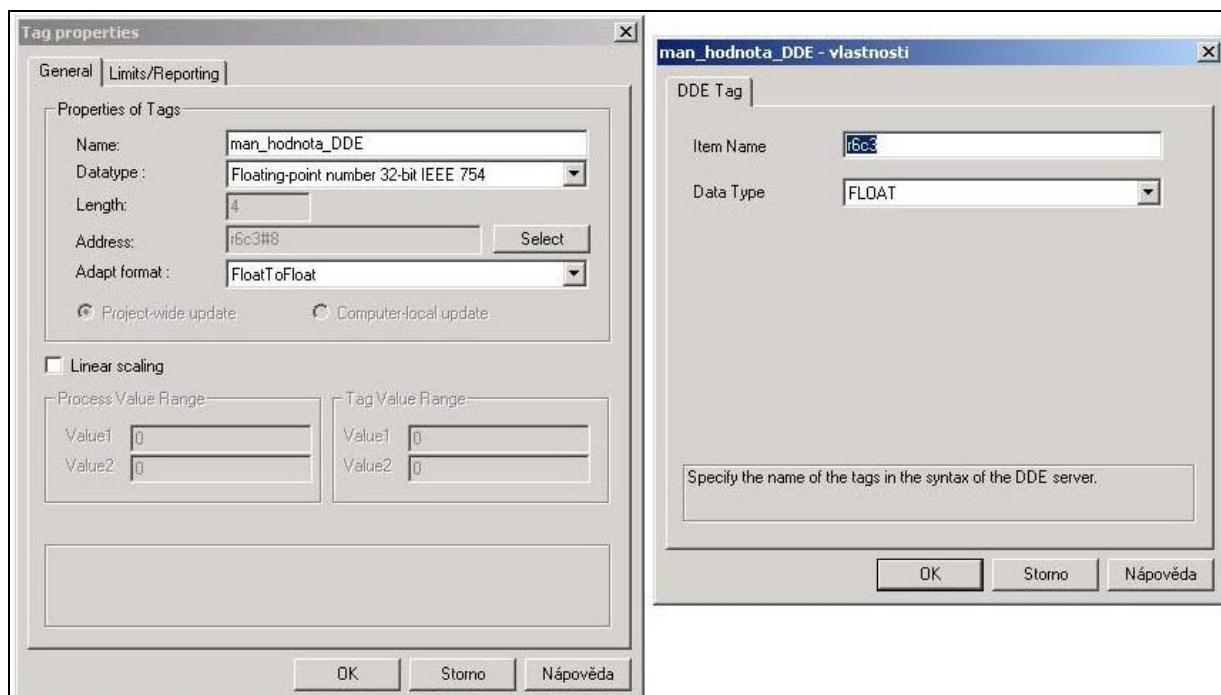
4.1.2 Prepojenie WinCC a Microsoft Excel pomocou tagov

Keďže pri práci využívam DDE komunikáciu je potrebné vytvoriť vo WinCC špecifické tagy umožňujúce tento typ komunikácie. Tieto tagy sa vytvárajú v Tag Management-e, kde sa pridá komunikačný driver *Windows DDE.CHN* a vytvorí sa rozhranie vo WinCC pre DDE komunikáciu. Tento typ tagov nazývame externé tagy. Pre DDE komunikáciu treba zadať v *Connection Properties* aplikáciu. Keďže v mojom prípade využívam WinCC z pozície *Klienta*, do názvu služby zadávam program, ktorý bude *Server*. Potom do *Názvu služby* zadávam „excel“ a *Názov témy* bude názov súboru v programe excel v ktorom majú byť zobrazované požadované údaje (obr. 4.3).



Obr. 4. 3 Zadefinovanie DDE spojenia pre program WinCC ako Klient

Tagy sa vytvárajú podobne ako je to popísané v stati 4.1.1, ale rozdiel je pri definovaní adresy, kde sa adresuje do položky *Item Name* konkrétnu bunku v Exceli v ktorej majú byť údaje zobrazované (obr.4.4). Premenná je typu FLOAT, čiže pre zobrazovanie desatinných miest.

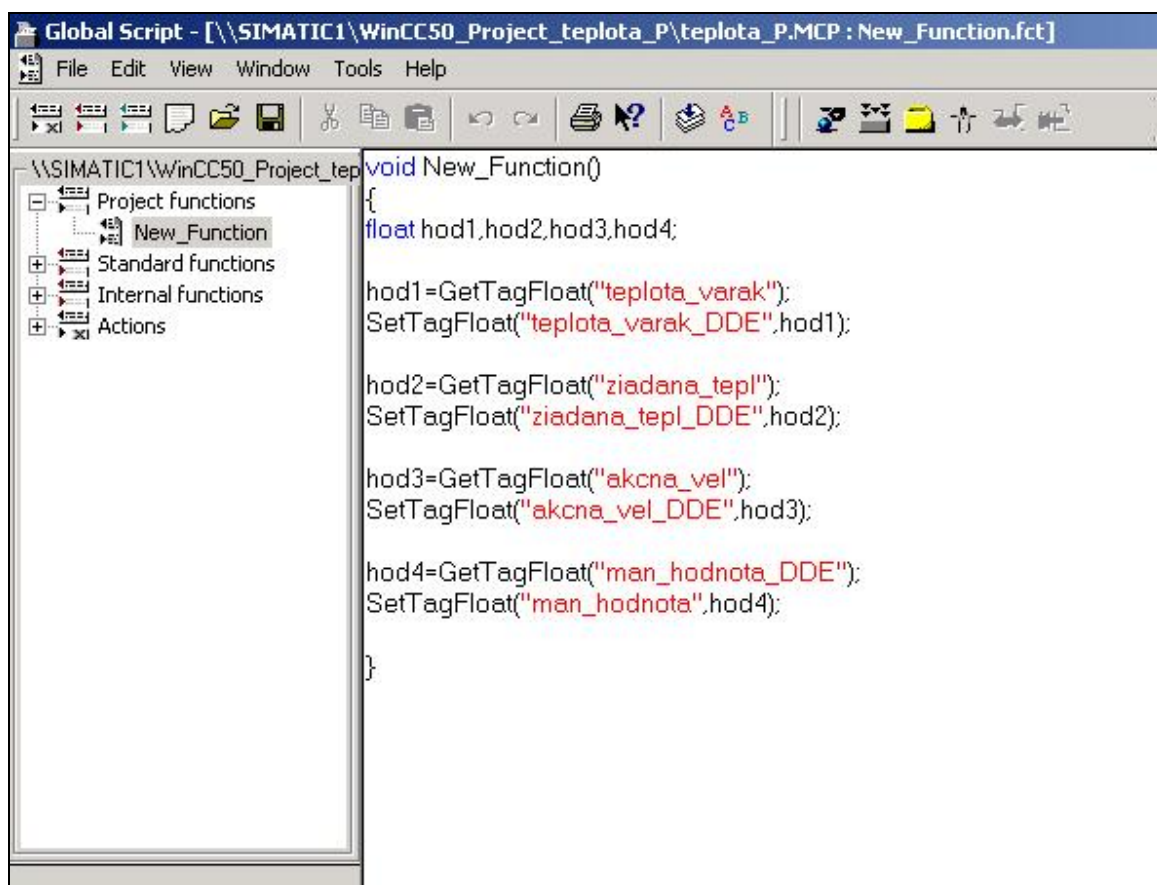


Obr. 4. 4 Okná tvorby tagu pre DDE komunikáciu medzi WinCC a Excelom.

Pre úspešné prepojenie WinCC a Excelu pomocou tagov je potrebné vytvoriť program v module *Global Script-e*, pomocou ktorého bude možné kopírovanie hodnôt medzi tagmi, ktoré spájajú WinCC - STEP7 a WinCC – Excel. Pri písaní programu sú využité dva hlavné príkazy:

- *GetTagFloat* – načíta hodnotu tagu do danej premennej
- *SetTagFloat* – nastaví hodnotu tagu na hodnotu danej premennej

V mojom prípade využívam na DDE komunikáciu štyri tagy („teplota_varak, ziadana_templ, akcna_vel a man_hodnota“). Ako možno vidieť na obr. 4.5 prvé tri spomenuté tagy posielajú hodnotu do Excelu a tag „man_hodnota“ načíta hodnoty z Excelu.



Obr. 4. 5 Program napísaný v Global Script pre komunikáciu tagov

4.2 Vizualizácia projektu

Vizualizácia projektu v *Graphics Designer*, umožňuje sledovanie a riadenie zvolených veličín a parametrov. WinCC umožňuje vytvorenie grafických objektov podľa požiadaviek užívateľa, ale taktiež má širokú paletu grafických objektov, ktoré sa nachádzajú v knižnici programu.

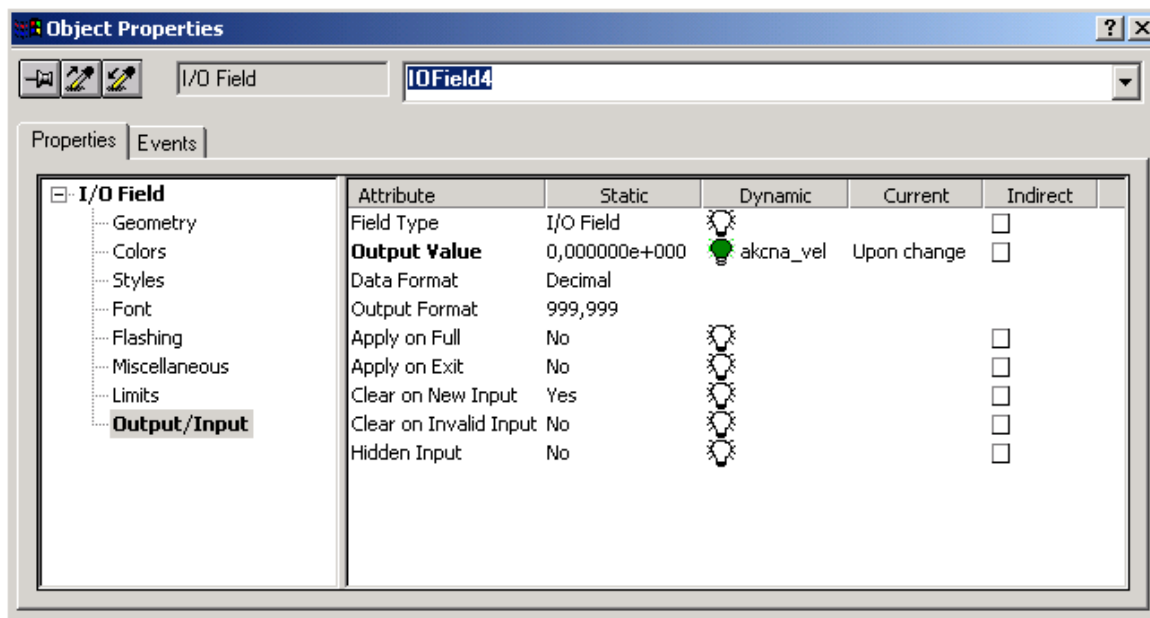
4.2.1 Vytvorenie ovládacích prvkov

Ďalšou dôležitou časťou programu sú ovládacie prvky, ktoré umožňujú ovplyvňovať alebo sú ovplyvňované vznikajúcimi udalosťami v procese. Zaznamenávajú zmenu, ktorá vznikla v procese alebo zmenu, ktorú vykoná užívateľ pri zásahu do procesu. Ovládacie prvky možno prirovnať k dynamickým objektom. Nachádzajú sa v *Object Palette* v *Smart Objects*.

Medzi najpoužívanejší ovládací vstupno-výstupný prvok patrí *I/O Field*. Je to objekt vo forme textového poľa, ktorého hodnota závisí od hodnoty na pripojenom tagu. Na príklade I/O Field možno vysvetliť jeho aktiváciu vo WinCC. Po umiestnení I/O Field na pracovnú plochu sa otvorí okno, ktoré umožňuje definovať *tag* ktorého hodnoty bude objekt zobrazovať a tiež treba zvoliť periódu v akej budú hodnoty obnovované. V mojom prípade som nastavil hodnotu obnovovania údajov na *Upon Change*. Ďalšie nastavenie vlastností objektu sa robí po vyvolaní okna na objekte a voľbou *Properties*. Medzi hlavné nastavenia vstupno-výstupného poľa patrí:

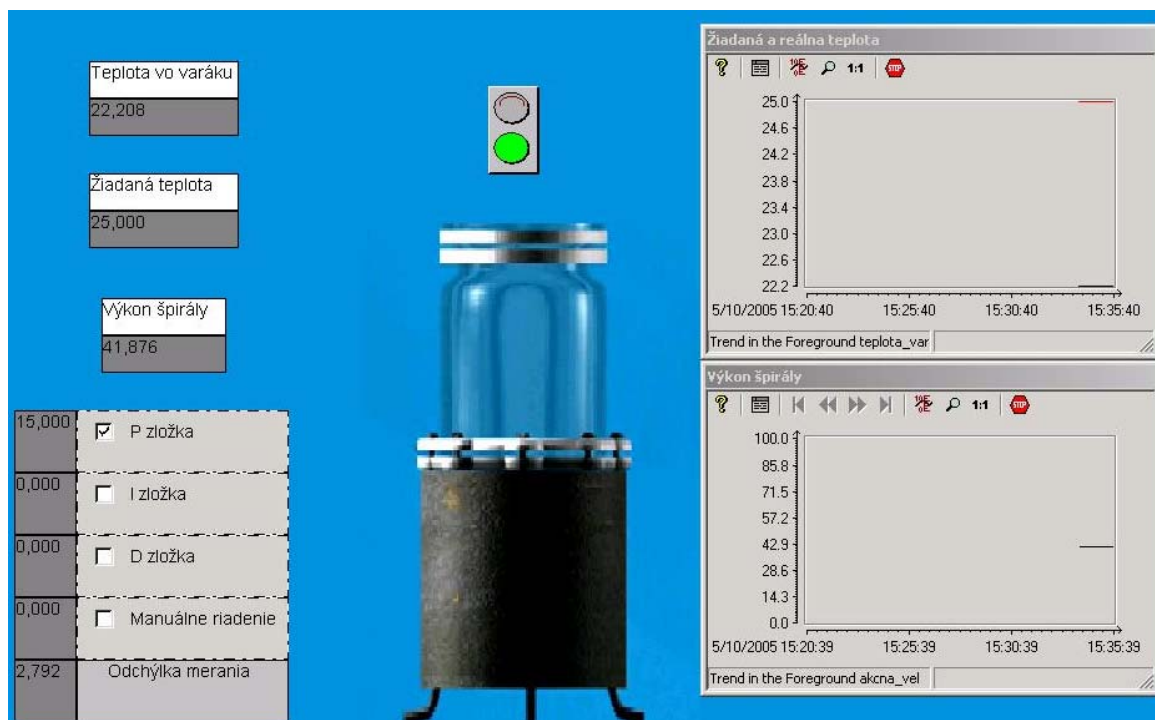
- **Field Type** – umožňuje nastaviť typ poľa na Input, Output, Both, kde
Input(vstupné) – možno do poľa zapisovať hodnoty a tým riadiť proces napr. hodnotu žiadanej teploty
Output(výstupné) – zobrazuje aktuálne hodnoty namerané v systéme napr. zobrazenie teploty vo varáku
Both(kombinované) – umožňuje zapisovanie a zároveň aj sledovanie aktuálnej hodnoty procesu
- **Data format** – možnosť voľby zobrazovaného formátu (Binary, Decimal, Hexidecimal, String)
- **Output Format** – umožňuje nastaviť koľko ciferné čísla má pole zobrazovať

Ako možno vidieť na obr. 4.6 vlastnosti poľa sú rozdelené na statické a dynamické. Statické vlastnosti sa nastavujú priamo v *Graphics Designer-i*, ale dynamické vlastnosti predstavuje hodnota tagu. Dynamickú vlastnosť je daná v prípade zelenej žiarovky pri príslušnej vlastnosti. V ľavej časti (obr. 4.6) sa nachádzajú kategórie vlastností.



Obr. 4. 6 Vlastnosti objektu I/O Field

Ďalším ovládací prvok, ktorý som využil pri práci vo WinCC je indikátor kopírovania hodnôt *On_Off_3*. Slúži na indikáciu komunikácie medzi tagmi. Po kliknutí na ikonu *Display Library* môžeme umiestniť *On_Off_3* z *Global Library/Operation/Toggle Buttons* na pracovnú plochu. V položke Properties cez Tag Assignment vyberiem pre dynamickú vlastnosť položku *C-Action* v ktorej definujem názov funkcie napísaný v *Global Script-e*. Názov funkcie je *NewFunction* (obr.4.5). Pokiaľ indikátor zobrazuje zelenú farbu, tak prebieha funkcia napísaná v *Global Script-e*. V opačnom prípade je v programe chyba a kopírovanie medzi tagmi nie je správne. Výsledné zobrazenie vizualizačného okna možno vidieť na obr. 4.7.



Obr. 4. 7 Zobrazenie aktívneho okna vo WinCC

4.2.2 Tvorba trendov a archívov

K vizualizácii projektu tiež patrí zobrazenie meraných priebehov pomocou trendov čiže grafov. Trendové údaje môžeme sledovať *online*, bez možnosti ukladať namerané údaje na ďalšie spracovanie, alebo pomocou *archív-ov*, kedy sa namerané údaje ukladajú do súborov s možnosťou ďalšieho spracovania. Ak chceme namerané hodnoty zaznamenávať je vhodné najskôr vytvoriť archív na ktorý sa pripájajú trendové zobrazenia v *Graphics Designer-i*.

Archív sa vytvára v module *Tag Logging*. Nový archív sa vytváram tak, že vyvolám voľbu *Wizard*, kde zvolím názov archívu a taktiež tagy, ktorých priebehy chcem zaznamenávať. Vytvorené súbory, ktoré ukladajú merané hodnoty majú rovnaký názov ako je názov daného tagu. Pre zvolený tag sa môže nastavovať interval zapisovania v mojom prípade je to 1s. Taktiež je dôležité správne nastaviť veľkosť archívneho súboru. Pri nedostatočnej kapacite súboru dôjde prepisovaniu nameraných hodnôt. Archívne súbory sú vo formáte *.DBF, ktoré možno ďalej spracovávať prostredníctvom programov Excel, Origin a ďalších. Ich aktiváciu treba potvrdiť v *Project Properties*. Základné parametre archívu sú týmto nastavené.

Ako už bolo spomenuté ako jeden z vizualizačných prvkov v prostredí *Graphics Designer* je *WinCC Trend Control*. Nachádza sa v *Object Palette* v záložke *Controls*. Po umiestnení *Trend-u* na pracovnú plochu sa otvorí okno, v ktorom definujeme tag, ktorého

hodnoty majú byť zobrazované. Tu je možnosť rozhodnúť sa či využijeme archiváciu údajov. *Trend* taktiež umožňuje nastavovanie vlastností zobrazovania v jednotlivých položkách podľa požiadaviek užívateľa.

4.3 Aktivácia WinCC

Po vytvorení projektu vo WinCC je potrebné zabezpečiť spúšťanie jednotlivých častí programu pre správnu komunikáciu so STEP7 a PLC. Toto sa zabezpečuje vo vlastnostiach vetvy *Computer* (obr. 4.1) v hlavnom okne *WinCCExplorer*, kde sa nachádza záložka *Startup*, sa nastavujú moduly, ktoré sa majú spustiť pri štarte projektu. V mojom prípade som aktivoval *Graphics Runtime*, ktorý zabezpečuje obsluhu grafického rozhrania, ďalej *Tag Logging Runtime*, ktorý slúži na správu trendových archívov a *Global Script Runtime* pre spúšťanie funkcie napísanej v *Global Script-e*

Po správnom nastavení modulu možno projekt úspešne aktivovať. Kompletné spustenie STEP7 a WinCC možno vykonať v nasledujúcich krokoch:

1. Zapnúť PLC stanicu a nastaviť kľúč do polohy STOP.
2. Aktivovať riadiaci program v STEP7 jeho nakopírovaním do RAM pamäte CPU.
3. Nastaviť kľúč do polohy RUN a aktivovať program vo WinCC stlačením tlačidla PLAY v *Graphics Designer-i*.

5 Návrh regulátorov, simulácie a priebehy riadenia varáku

Varák je časť zariadenia kolóny, ktoré zabezpečuje primárne vyhrievanie zmesi kvapalín určených na separáciu. Je vyrobený z nerezovej ocele a sú v ňom zabudované vyhrievacie telesá. Vyhrievacie telesá pozostávajú zo štyroch elektricky vyhrievaných špirál. Z bezpečnostných dôvodov a z dôvodu udržania konštantnej plochy prestupu tepla je potrebné, aby boli výhrevné telesá úplne ponorené v náplni kolóny. Tri špirály majú výkon 1,5 kW a jedna, ktorej výkon je 2,0 kW. Varák je navrhnutý tak, že je možná optická kontrola výška hladiny. Pod úrovňou varáku je umiestnený vypúšťací ventil na odber destilačného zvyšku. Akčná veličina je výkon špirál a výstupná veličina je teplota vo varáku. Vlastnosti varáku sú popísané v stati 5. Na riadenie špirál varáka slúži triak.

5.1 Matematický model varáka

Pre simulačné potreby bolo treba zostaviť matematický model varáka rektifikačnej kolóny naplnený destilovanou vodou. Keďže varák je valcovitého tvaru z nameraných hodnôt priemeru a výšky som získal objem:

- priemer $d = 0,3 \text{ m}$
- výška $h = 0,3 \text{ m}$

$$V = \frac{\pi d^2}{4} h = \frac{\pi \cdot 0,3^2}{4} 0,3 = 0,02121 \text{ m}^3 = 21,21 \text{ dm}^3 \quad (5.1)$$

Matematické modely sa zostavujú na základe materiálových a energetických bilancií. Keďže v mojom prípade predpokladám že objem vody bude konštantný, tak na zostavenie matematického modelu použijem energetickú bilanciu. Do úvahy beriem ďalšie zjednodušujúce predpoklady ako je zanedbanie strát tepla do okolia, konštantné hodnoty hustoty, tepelnej kapacity a tlaku:

- hustota $\rho = 998 \text{ kg/m}^3$
- tepelná kapacita $c_p = 4,18 \text{ kJ/kg.K}$

Potom matematický model ma tvar:

$$Q(t) = V \cdot \rho \cdot c_p \frac{dv(t)}{dt} \quad (5.2)$$

Q – teplo dodané do varáku [kJ]

- v- teplota vo varáku [K]

a po zjednodušení $Q(t) = k \frac{dv(t)}{dt}$ (5.3)

kde $k = 88,48 \text{ kJ/K}$

Definujem vstupnú odchýlkovú veličinu:

$$U(t) = Q(t) - Q^s \quad (5.4)$$

- Q^s – výkon špirály v ustálenom stave

a stavovú odchýlkovú veličinu:

$$x(t) = v(t) - v^s \quad (5.5)$$

- v^s – teplota vo varáku v ustálenom stave

Po dosadení odchýlkových veličín do matematického opisu dostávam:

$$\bar{U}(t) = k \cdot \frac{d\bar{x}(t)}{dt} \quad (5.6)$$

Stavový opis bude mať tvar:

$$\frac{d\bar{x}(t)}{dt} = \frac{\bar{U}(t)}{k} \quad (5.7)$$

na základe stavového opisu môžem vyjadriť skalárne vektory, vyjadrujúce stavové,

vstupné a výstupné veličiny: $\bar{A} = 0$; $\bar{B} = \frac{1}{k}$; $\bar{C} = 1$; $\bar{D} = 0$;

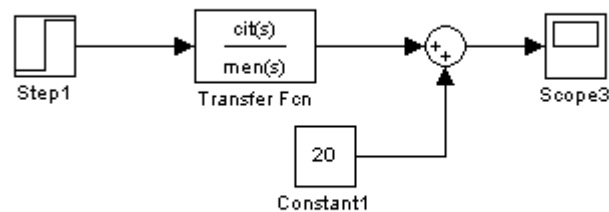
V programe MATLAB pomocou príkazu ss2tf (*State-space to transfer function conversion*) získam prenos, ktorý je v tvare:

$$G_P(s) = \frac{1}{88,48s} \quad (5.8)$$

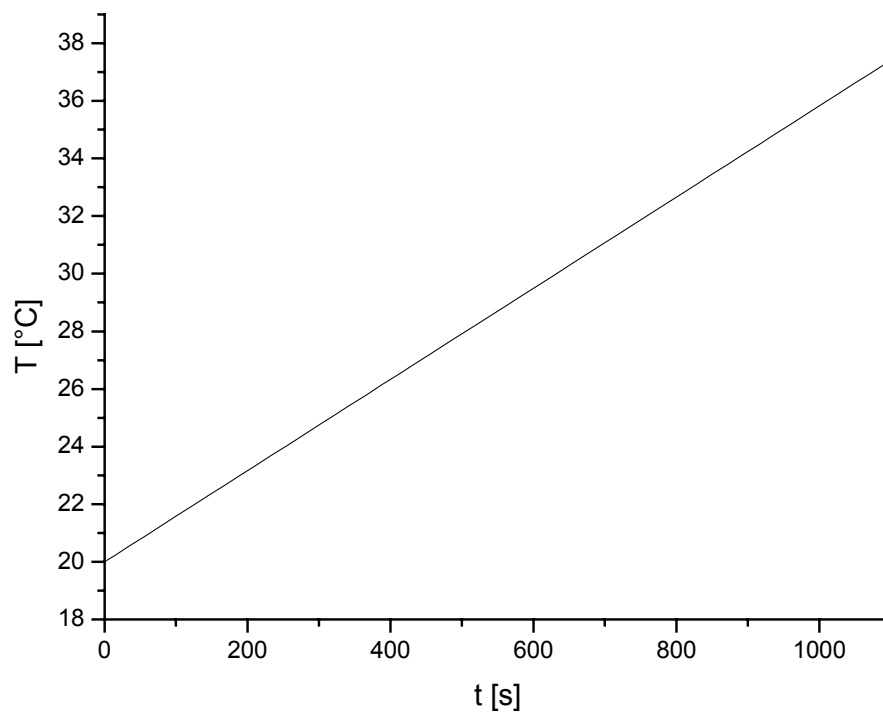
resp.

$$G_P(s) = \frac{0,0113}{s} \quad (5.9)$$

Pomocou MATLABu a simulačného programu Simulink som získal priebeh prechodovej charakteristiky varáka, ktorý je opísaný prenosom (5.9) Počiatočná teplota bola 20°C pri nulovom výkone varáku a skoková zmena bola sto percent výkonu špirály. Priebeh prechodovej charakteristiky je na obr. 5.2.



Obr. 5. 1 Simulačná schéma v programe MATLAB



Obr. 5. 2 Zobrazenie priebehu prechodovej charakteristiky v programe MATLAB pre prenos (5.9)

5.2 Syntéza regulátora metódou umiestnenia pólov pre matematický model varáka

Metóda umiestnenia pólov má rozšírenú aplikáciu v praxi na nastavenie parametrov PID regulátora. Hlavnou myšlienkou metódy umiestnenia pólov je určiť menovateľ prenosu uzavretého regulačného obvodu (URO), ktorý determinuje stabilitu uzavretého regulačného obvodu. Pre prenos (5.9), ktorý vykazuje integračné vlastnosti môžeme navrhnúť regulátor bez integračného člena v tvare rovnice:

$$G_R(s) = Z_R(1 + T_D \cdot s) \quad (5.10)$$

Pri navrhovaní regulátora metódou umiestnenia pólov vychádzam z charakteristickej rovnice uzavretého regulačného obvodu pre ktorú platí:

$$1 + G_P \cdot G_R = 0 \quad (5.11)$$

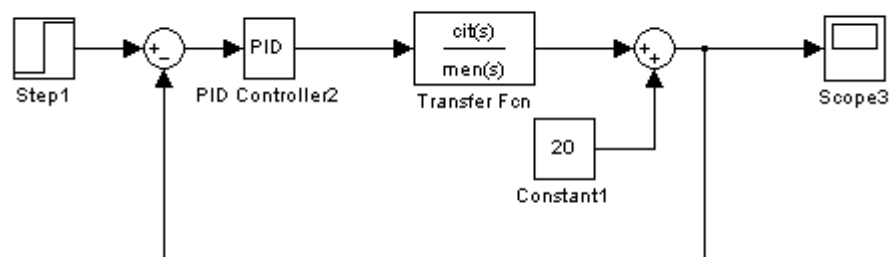
Po dosadení do rovnice (5.11) z (5.9) a (5.10) dostávam:

$$\frac{s(1 + 0,0113Z_R T_D) + 0,0113Z_R}{s} = 0 \quad (5.12)$$

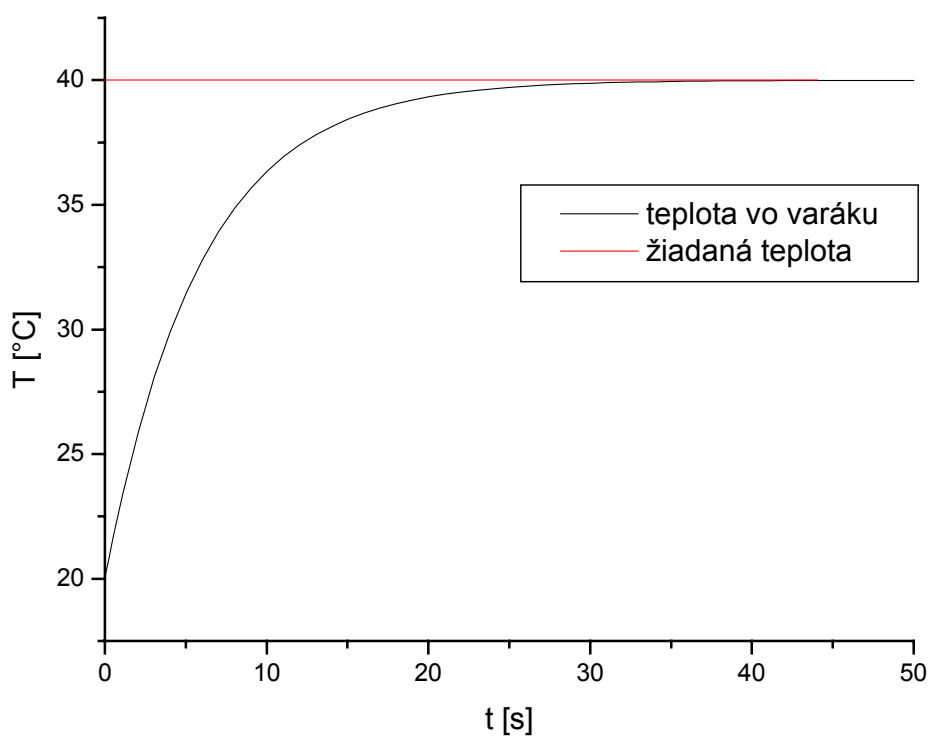
Ako možno vidieť charakteristická rovnica (čitateľ) je prvého rádu a je možné voliť jeden pól. Keďže PD regulátor má dva voliteľné parametre, jeden z nich je nadbytočný. V ďalšom budem uvažovať nulovú D zložku, čiže P regulátor. Charakteristická rovnica sa zjednoduší na tvar:

$$s + 0,0113Z_R = 0 \quad (5.13)$$

Keďže máme riadený systém prvého rádu, potom pre nastavenie pólov použijem charakteristickú rovnicu: $s + \omega_0 = 0$. Porovnaním koeficientov pri rovnakých mocninách a zvolením $\omega_0 = 0,1695$ dostávam hodnotu P regulátora $P = 15$. Výsledok riadenia modelu varáka pomocou navrhnutého regulátora v MATLABe možno vidieť na obr. 5.4.



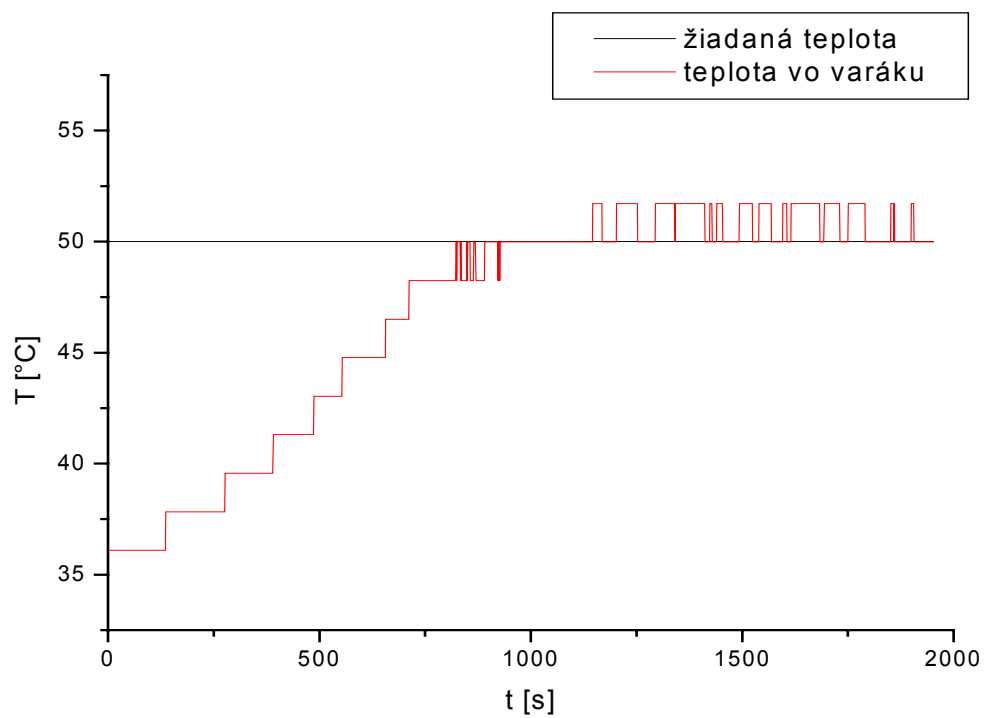
Obr. 5. 3 Simulačná schéma URO s P regulátorom v programe MATLAB



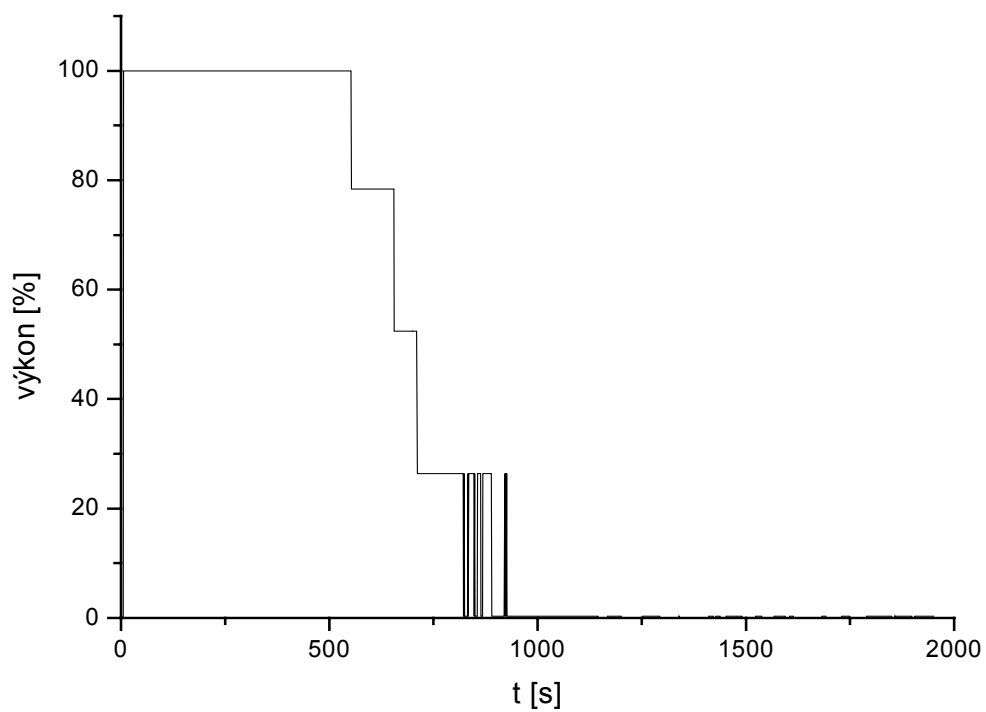
Obr. 5. 4 Zobrazenie priebehu riadenia varáka (5.9) pomocou P regulátora.

5.2.1 Riadenie varáka P regulátorom s využitím STEP7

Riadenie pomocou navrhnutého P regulátora som taktiež aplikoval na varák. Využil som programy STEP7 (obr. 2.3) a WinCC, kde pomocou bloku FB41 (Príloha A) som zadal vypočítanú hodnotu P zložky do riadiaceho programu. Priebehy riadenia možno vidieť na obr. 5.5 a priebeh výkonu špirály na obr. 5.6.



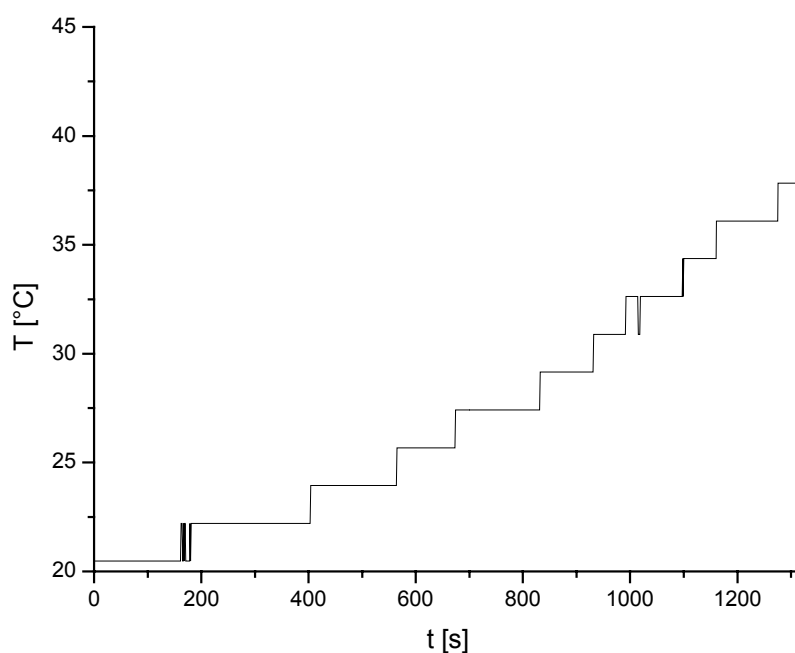
Obr. 5. 5 Zobrazenie priebehu teploty pri riadení pomocou P regulátora



Obr. 5. 6 Priebeh výkonu špirály pri riadení pomocou P regulátora

5.3 Padeho aproximácia

Meraním prechodovej charakteristiky varáka pomocou STEP7 a WinCC, som zistil rozdiel medzi nameranou prechodovou charakteristikou (obr. 5.7) a prechodovou charakteristikou získanou pomocou matematického modelu (obr. 5.2).



Obr. 5. 7 Prechodová charakteristika varáka nameraná pomocou STEP7 a WinCC

Z obr. 5.7 som určil dopravné oneskorenie $D = 160s$. Padeho aproximácia sa môže používať pre zahrnutie dopravného oneskorenia systému priamo do prenosu, bez toho aby sme ho vyjadrili pomocou exponenciálneho tvaru.

Uvažujme všeobecný tvar s integračnými vlastnosťami [6]:

$$G(s) = \frac{K}{s} e^{-Ds} \quad (5.14)$$

kde podľa Pade-ho aproximácie platí:

$$e^{-Ds} = \frac{2 - Ds}{2 + Ds} \quad (5.15)$$

Prenos po aproximácii má tvar:

$$G_{ps} = \frac{K(2 - Ds)}{s(2 + Ds)} = \frac{b_0 - b_1s}{s^2 + a_1s} \quad (5.16)$$

kde $b_0 = \frac{2K}{D}$, $b_1=K$, $a_1 = \frac{2}{D}$

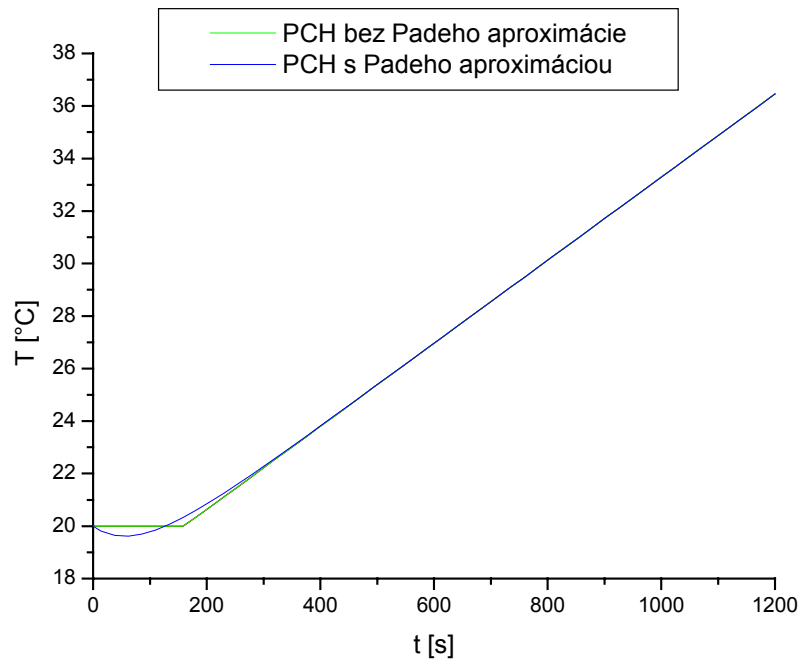
Tvar prenosu (5.9) s vyjadreným dopravným oneskorením v exponenciálnom tvare je:

$$G(s) = \frac{0,0113}{s} e^{-160s} \quad (5.17)$$

Pre prenos (5.17) s Padeho aproximáciou platí:

$$G_p(s) = \frac{-0,0113s + 1,4125 \cdot 10^{-4}}{s^2 + 1,25 \cdot 10^{-2}s} \quad (5.18)$$

Porovnanie prechodových charakteristík modelu varáka vyjadreného pomocou rovníc (5.17) a (5.18)



Obr. 5. Porovnanie prechodových charakteristík v MATLABe

5.4 Syntéza LQ regulátora

LQ regulátor je navrhnutý na prenos po Padeho aproximácii (5.16). Pri návrhu LQ regulátora vychádzam z diofantickej rovnice:

$$AP + BQ = M \quad (5.19)$$

$$\text{kde} \quad M = N.G \quad (5.20)$$

N je stabilná časť polynómu rovnice spektrálnej faktorizácie.

$$k = N.N^* = a.a^* \quad (5.21)$$

Polynóm k som zvolil tak, aby bola zabezpečená možnosť riadenia, kde $k = s^2 - 1$ a G je stabilná časť polynómu rovnice spektrálnej faktorizácie:

$$h = G.G^* = \varphi.a^*.a + \psi.b^*.b \quad (5.22)$$

$$\text{kde} \quad a = s^2 + 0,0125s$$

$$a^* = -0,00125s + s^2$$

$$b = -0,0113s + 0,00014$$

$$b^* = 0,0113s + 0,00014$$

$$\varphi = 5$$

$$\psi = 1$$

Po dosadení dostávam pre h

$$h = s^4 - 0,0038s^2 + 1.10^{-7} \quad (5.23)$$

Polynómy G a N vypočítam pomocou MATLABu a polynomického toolboxu príkazom *spf* (Polynomial spectral factorization) z polynómov (5.21 a 5.22) a dostanem tvary polynómov:

$$G = 2,2s^2 + 0,039s + 0,00014 \quad (5.24)$$

$$N = s + 1 \quad (5.25)$$

Hodnotu polynómu M získam dosadením rovníc (5.24 a 5.25) do rovnice (5.18):

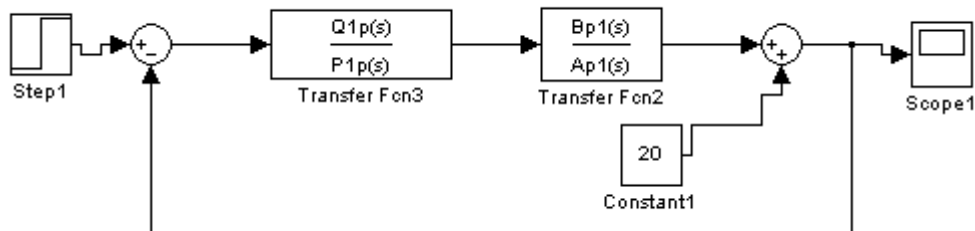
$$M = 2,2s^3 + 2,3s^2 + 0,039s + 0,00014 \quad (5.26)$$

V MATLABe pomocou polynomického toolboxu a príkazu *axbys* získam polynomický tvar LQ regulátora, ktorý má tvar:

$$G_R(s) = \frac{Q(s)}{P(s)} = \frac{q_0}{p_0} = \frac{80s + 1}{2,2361s + 3,1514} \quad (5.27)$$

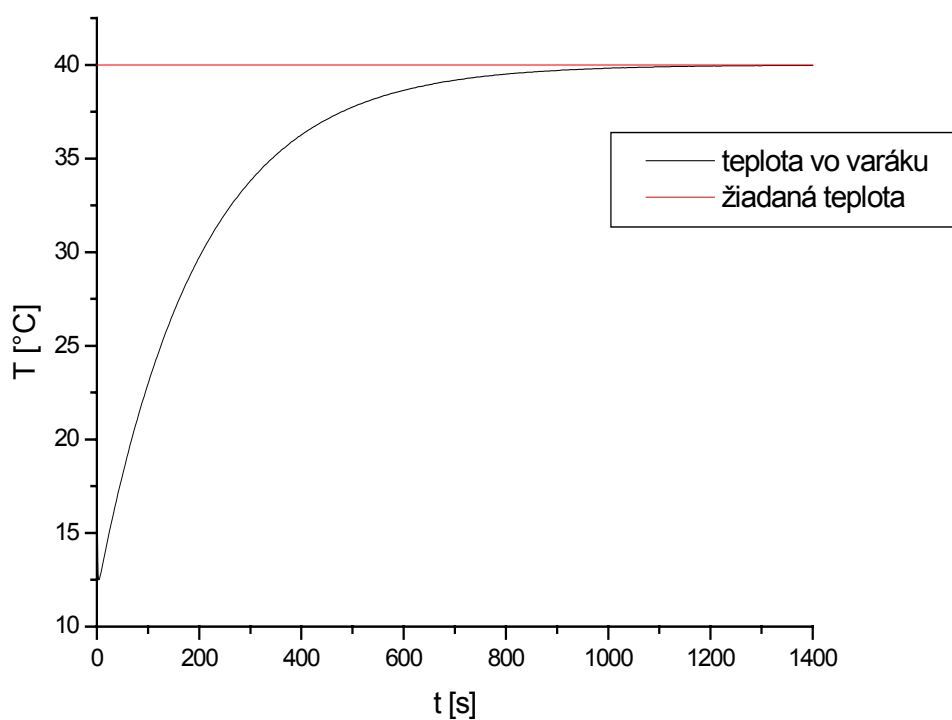
5.4.1 Riadenie LQ regulátorom v MATLABe

Pomocou navrhnutého regulátora som v MATLABe simuloval priebeh riadenia podľa schémy na obr. 5.9.



Obr. 5. 8 Simulačná schéma URO s LQ regulátorom v programe MATLAB

Simuláciu riadenia modelu varáka (5.18) navrhnutým LQ regulátorom v MATLABe možno vidieť na obr. 5.10.



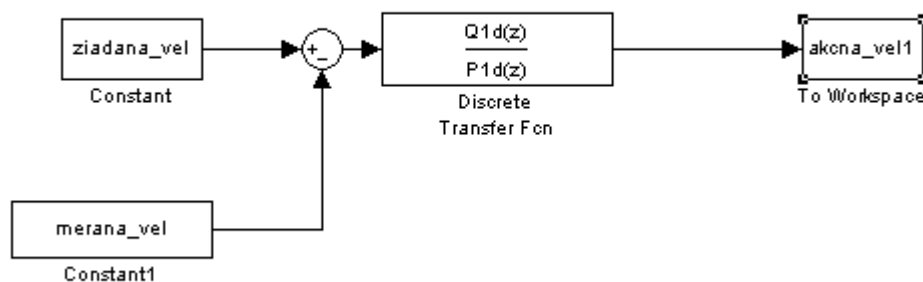
Obr. 5. 9 Zobrazenie priebehu riadenia varáka (5.18) pomocou LQ regulátora.

5.4.2 Riadenie varáka LQ regulátorom pomocou DDE rozhrania

Takto navrhnutý LQ regulátorom bol použitý pri riadení varáka pomocou DDE spojenia programov MATLAB, Excel, WinCC a STEP7. Hodnoty žiadanej teploty a aktuálnej teploty boli načítané do MATLABu, kde pomocou LQ regulátora bol vypočítaný akčný zásah, ktorý bol späť posielaný do STEP7 a na základe ktorého sa menil výkon špirály. Diskrétny tvar prenosu (5.28) bol získaný pomocou príkazu *c2d* v MATLABe a jeho tvar je:

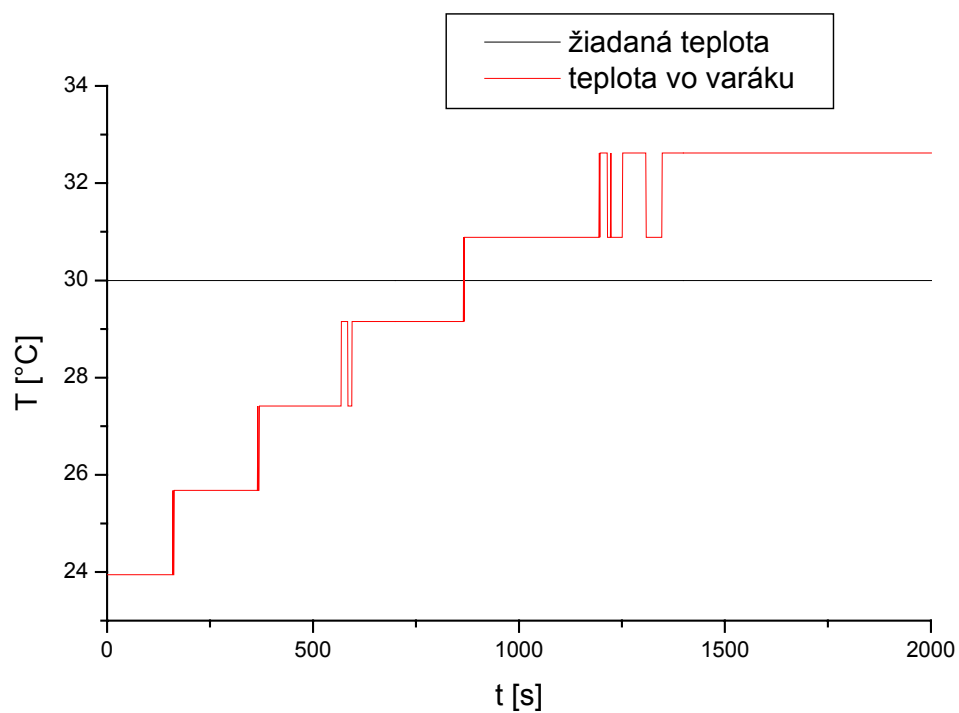
$$G_R(z) = \frac{35,7771z - 35,7354}{z - 0,8685} \quad (5.28)$$

Výpočet akčného zásahu bol robený podľa schémy na obr. 5.11.

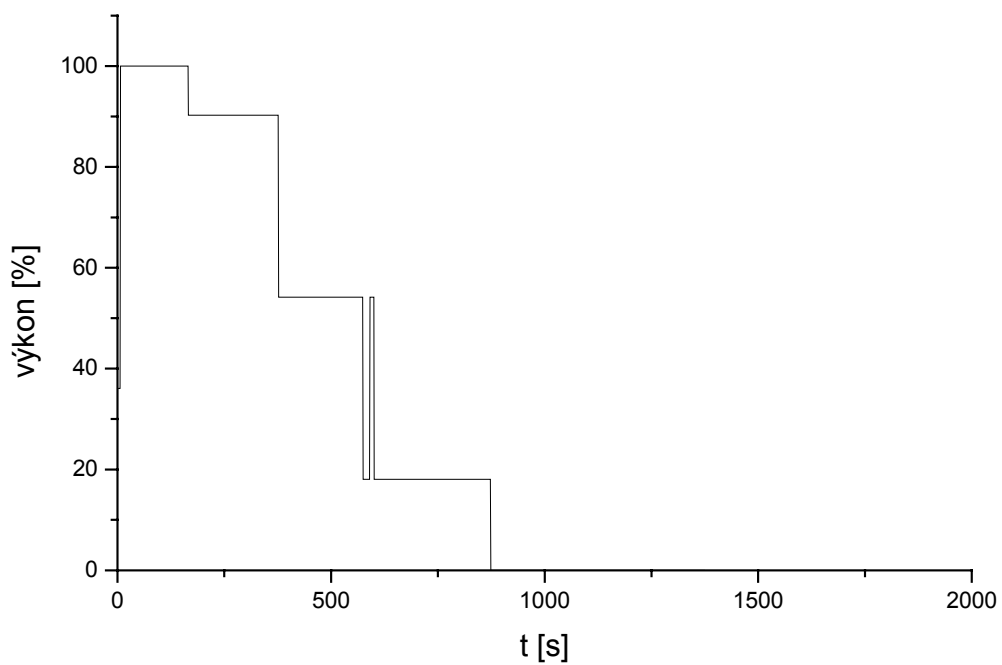


Obr. 5. 10 Schéma v MATLABe na výpočet akčného zásahu pre LQ regulátor

Nameraný priebeh teploty je znázornený na obr. 5.12 a priebeh výkonu špirály možno vidieť na obr. 5.13.



Obr. 5. 11 Zobrazenie priebehov teploty pri riadení pomocou LQ regulátora



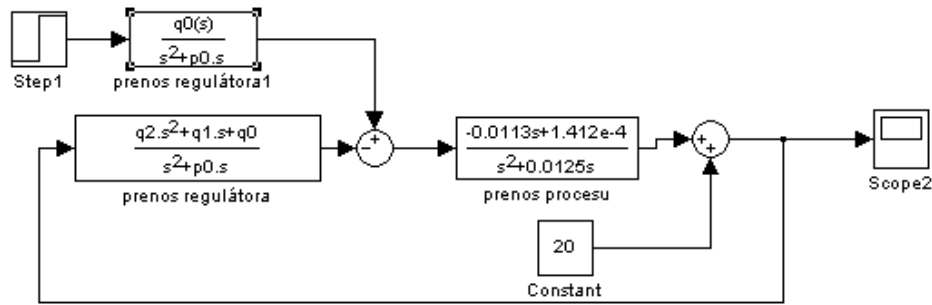
Obr. 5. 12 Priebeh výkonu špirály pri riadení pomocou LQ regulátora

5.5 Návrh regulátora pomocou optimálnej teórie riadenia

Pri návrhu regulátora vychádzam z predpokladu, že pre prenos po Padeho aproximácii (5.18) platí nerovnosť [7]:

$$\deg b \leq \deg a \quad (5.29)$$

Pre riadenie bola použitá schéma 2DOF regulátora (two-degree-of-freedom) s dvoma stupňami voľnosti. Schéma riadenia pre MATLAB je zobrazená na obr. 5.14.



Obr. 5. 13 Schéma riadenia systému pomocou navrhnutých regulátorov

Ako možno vidieť na obr. 5.14 regulačný obvod obsahuje spätnú väzbu a priamu väzbu.

Prenosy regulátorov sú v tvare :

$$\tilde{Q}(s) = Q(s)F(s) = \frac{q(s)}{sp(s)} \quad (5.30)$$

$$\tilde{R}(s) = R(s)F(s) = \frac{r(s)}{sp(s)} \quad (5.31)$$

kde $F(s)$ predstavuje integrátor a q , r , p sú polynómy, pričom p polynóm musí byť stabilný. Pri návrhu regulátorov sa vychádza z riešenia pre spätnú väzbu z rovnice:

$$a(s)s p(s) + b(s)q(s) = d(s) \quad (5.32)$$

a pre priamu väzbu

$$t(s)s + b(s)r(s) = d(s) \quad (5.33)$$

kde d je stabilný polynóm na pravej strane rovnice. Riadiaci systém uspokojivo spĺňa podmienky vnútornej stability. Stupeň polynómov v prenose zahŕňajúcich integrátor vyjadruje nerovnosť

$$\deg q \leq \deg p+1, \deg r \leq \deg p+1 \quad (5.34)$$

Ak sa zoberie do úvahy rovnica (5.34) a riešiteľnosť rovníc (5.32) a (5.33), stupeň polynómov q , p , a r môže byť odvodený ako:

$$\deg q = \deg a, \deg p \geq \deg a - 1, \deg r = 0 \quad (5.35)$$

Ďalej získame rovnosť:

$$r_0 = q_0 \quad (5.36)$$

Na získanie stabilných polynómov na pravej strane rovníc (5.32) a (5.33) sa využíva LQ metóda. Je známe, že LQ regulátor využíva stabilný polynóm g ako výsledok spektrálnej faktorizácie, podľa rovnice:

$$(sa(s))^* \varphi(sa(s)) + b^*(s)\mu b(s) = g^*(s)g(s) \quad (5.37)$$

kde $\mu \geq 0$ a $\varphi > 0$ sú váhové koeficienty. Ja som zvolil $\mu = 0,01$ a $\varphi = 500$. Použitím polynómu (5.36) dostávame pre polynómy p , q a r riešenie rovníc (5.32) a (5.33) s pravou stranou pre:

$$d(s) = g(s)n(s) \quad (5.38)$$

kde $n(s)$ je stabilný polynóm zabezpečujúci vlastnosti regulátora tak, že platí:

$$\deg n \geq \deg a - 1 \quad (5.39)$$

Prenosy regulátorov môžeme napísať v tvare:

$$\tilde{Q}(s) = \frac{q(s)}{sp(s)} = \frac{q_2 s^2 + q_1 s + q_0}{s(s + p_0)} \quad (5.40)$$

$$\tilde{R}(s) = \frac{r(s)}{sp(s)} = \frac{r_0}{s(s + p_0)} \quad (5.41)$$

parametre polynómu q a p sú vypočítané na základe rovníc:

$$q_0 = \frac{1}{K} g_0, \quad q_1 = \frac{1}{K} (g_1 + \tau g_0), \quad q_2 = \frac{\tau}{4K} (2g_1 + \tau g_0) \quad (5.42)$$

kde pre koeficienty g platí:

$$g_0 = \frac{2K}{\tau} \sqrt{\frac{\mu}{\varphi}}, \quad g_1 = \sqrt{\frac{4K}{\tau} \sqrt{\frac{\mu}{\varphi}} g_2 + K^2 \frac{\mu}{\varphi}}, \quad g_2 = \sqrt{\frac{2}{\sqrt{\varphi}} g_1 + \frac{4}{\tau^2}} \quad (5.43)$$

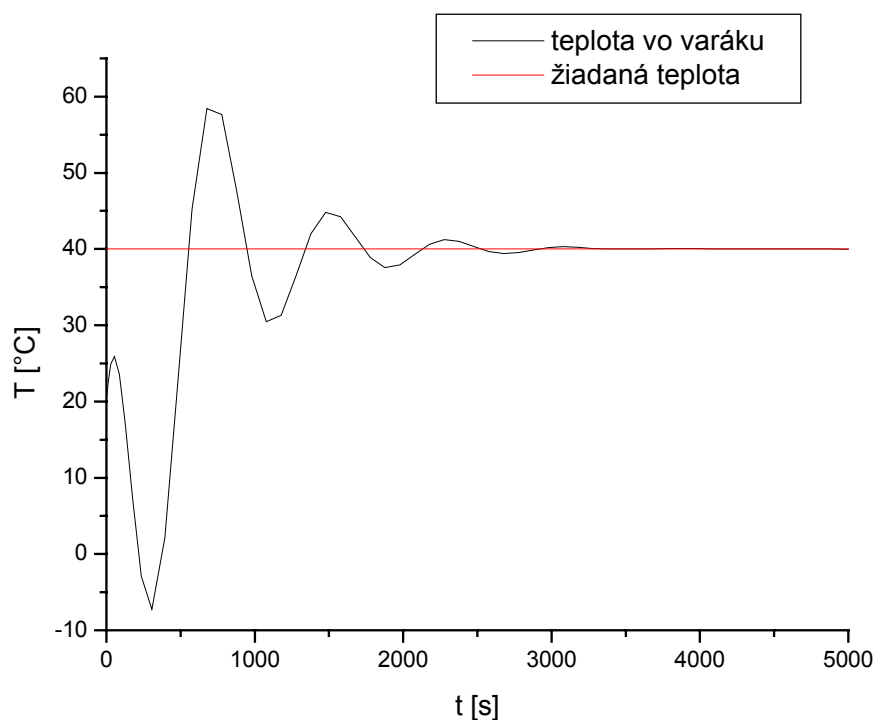
5.5.1 Riadenie regulátorom navrhnutým optimálnou teóriou riadenia v MATLABe

Koeficienty g_1 , a g_2 sú vypočítané v MATLABe pomocou funkcie *fsolve*. Výsledný prenos regulátorov je:

$$\tilde{Q}(s) = \frac{q(s)}{sp(s)} = \frac{1,0657s^2 + 0,0178s + 5,59 \cdot 10^{-5}}{s(s + 0,025)} \quad (5.44)$$

$$\tilde{R}(s) = \frac{r(s)}{sp(s)} = \frac{5,59 \cdot 10^{-5}}{s(s + 0,025)} \quad (5.45)$$

Riadenie modelu varáka pomocou prenosu (5.18) navrhnutými regulátormi v MATLABe možno vidieť na obr. 5.15.



Obr. 5. 14 Zobrazenie priebehu riadenia varáka (5.18) navrhnutého pomocou optimálnej teórie riadenia.

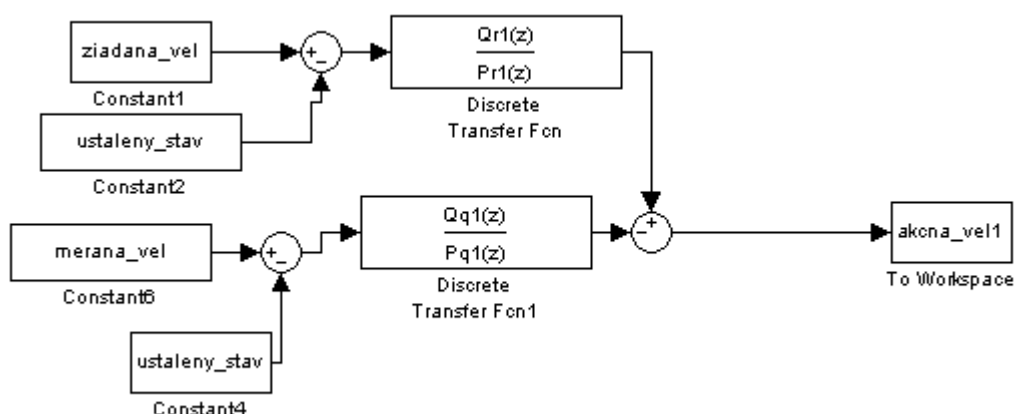
5.5.2 Riadenie varáka regulátorom navrhnutým optimálnou teóriou riadenia pomocou DDE rozhrania

Riadenie varáka navrhnutým regulátorom bolo vykonané pomocou DDE spojenia a schémy podľa obr. 5.16. Diskrétné tvary prenosov regulátorov boli získané v MATLABe príkazom *c2d*:

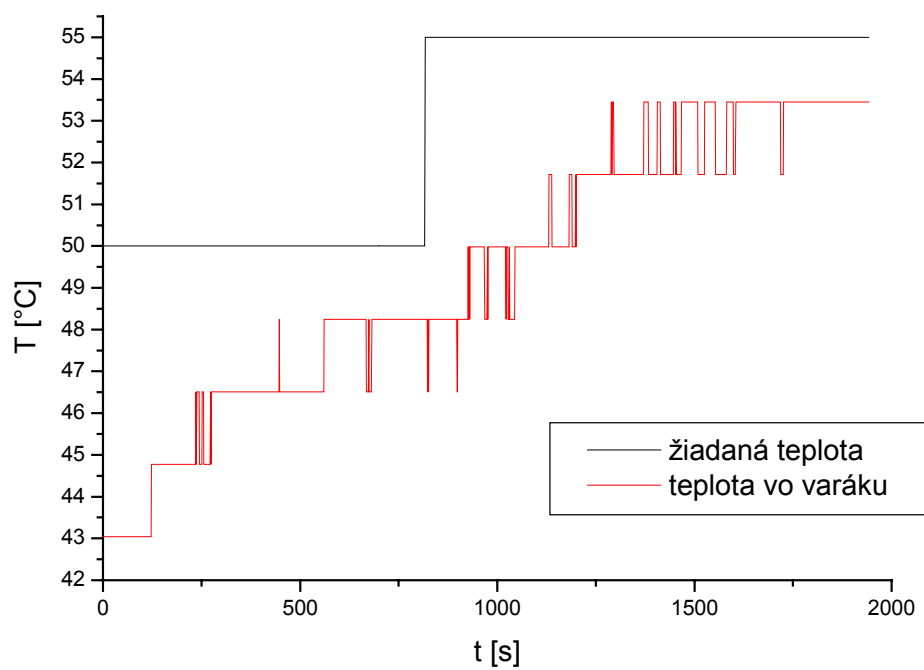
$$\tilde{Q}(z) = \frac{1,0657z^2 - 2,1297z + 1,064}{z^2 - 1,9975z + 0,9975} \quad (5.46)$$

$$\tilde{R}(z) = \frac{0,2793 \cdot 10^{-6}z + 0,279 \cdot 10^{-5}}{z^2 - 1,9975z + 0,9975} \quad (5.47)$$

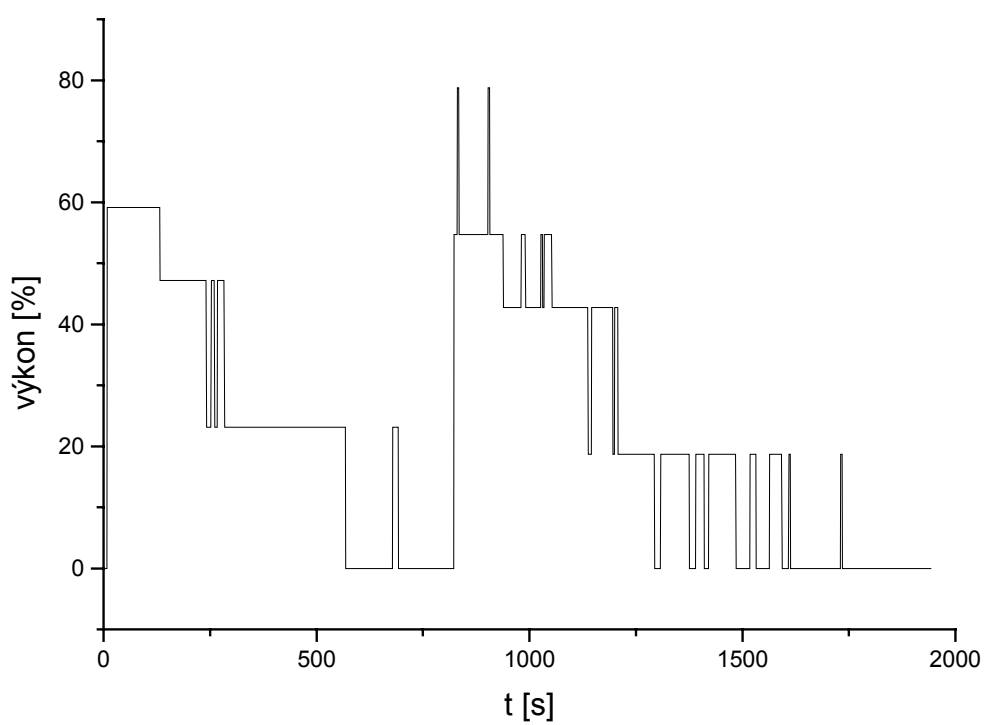
Nameraný priebeh teploty je znázornený na obr. 5.17 a priebeh výkonu špirály možno vidieť na obr. 5.18.



Obr. 5. 15 Schéma v MATLABe na výpočet akčného zásahu pre regulátor navrhnutý pomocou optimálnej teórie riadenia



Obr. 5. 16 Priebehy teplôt pri riadení pomocou optimálnej teórie riadenia regulátora



Obr. 5. 17 Priebeh výkonu špirály pri riadení pomocou optimálnej teórie riadenia regulátora

5.6 Návrh PD regulátora metódou umiestnenia pólov pre matematický model po Padeho aproximácii

V stati 5.2 bol navrhovaný PID regulátor metódou umiestnenia pólov pre matematický model varáka (5.9). Výsledný tvar regulátora bol regulátor s P zložkou. V tejto stati sa pokúsim navrhnúť PD regulátor pre prenos (5.18), ktorý sa využíva pre procesy s integračnými vlastnosťami alebo veľkými časovými konštantami. Pri návrhu regulátora vychádzam s prenosu (5.10) a tvaru prenosu (5.18). Podobne ako v predchádzajúcom prípade vychádzam z charakteristickej rovnice uzavretého regulačného obvodu (5.11). Po dosadení prenosov (5.10) a (5.18) do rovnice (5.11) dostávam:

$$1 + \frac{(Z_R + Z_R T_D s)(-0,0113s + 1,412 \cdot 10^{-4})}{s^2 + 0,0125s} = 0 \quad (5.48)$$

Po úprave dostávam charakteristickú rovnicu v tvare:

$$s^2 + \frac{s(0,0125 - 0,0113Z_R + 1,412 \cdot 10^{-4} Z_R T_D)}{1 - 0,0113Z_R T_D} + \frac{1,412 \cdot 10^{-4} Z_R}{1 - 0,0113Z_R T_D} = 0 \quad (5.49)$$

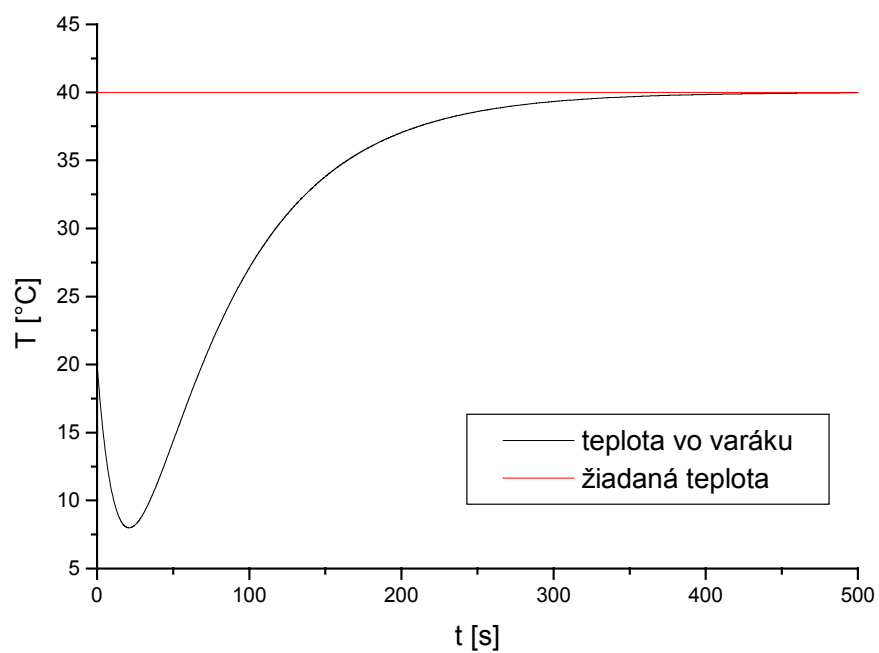
Charakteristická rovnica je druhého rádu a preto pre nastavenie pólov použijem rovnicu v tvare:

$$s^2 + 2\omega_0 s + \omega_0^2 = 0 \quad (5.50)$$

Porovnaním koeficientov pri rovnakých mocninách a zvolením $\omega_0 = 0,024$ dostávam hodnoty zložiek regulátora. Hodnota ω_0 bola volená tak, aby bola splnená podmienka stability charakteristickej rovnice podľa Nyquist-ovho kritéria. Výsledný tvar prenosu regulátora (5.10) je:

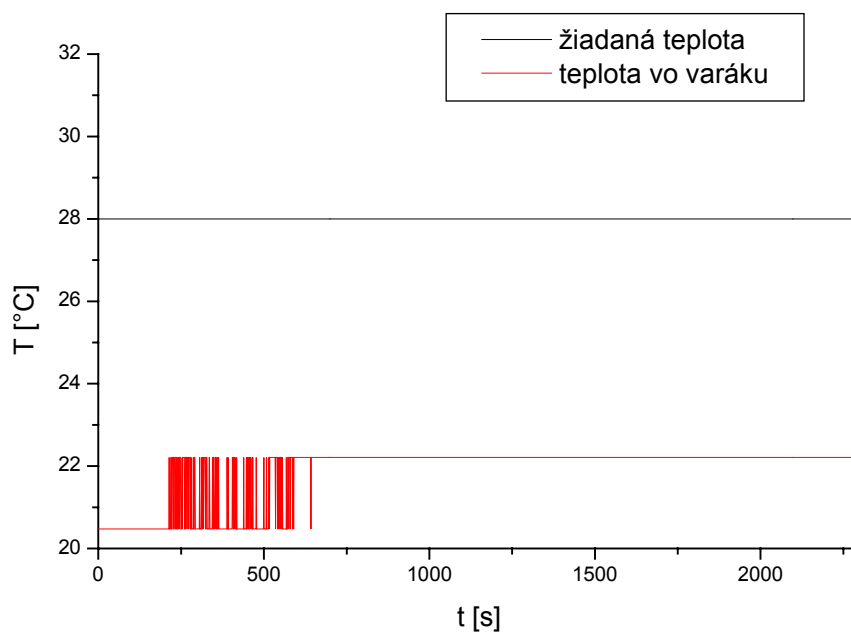
$$G_R(s) = 67,73s + 0,9566 \quad (5.51)$$

Simulácia riadenia bola uskutočnená pomocou simulačnej schémy na obr. 5.3 a priebeh simulácie je znázornený na obr. 5.19.

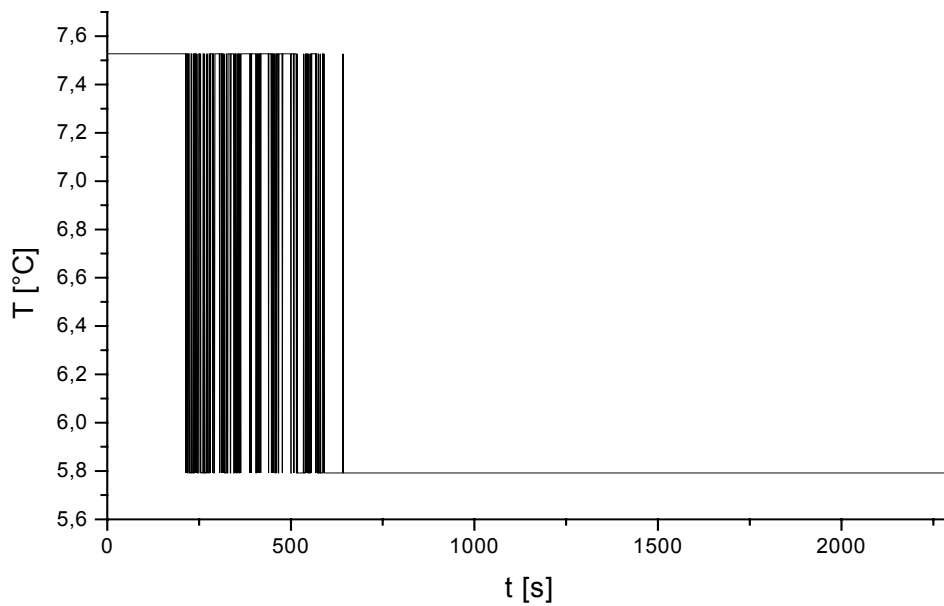


Obr. 5. 18 Zobrazenie priebehu riadenia modelu varáka (5.18) v MATLABe pomocou PD regulátora

Priebehy riadenia namerané pomocou STEP7 a WinCC sú znázornené pre teplotu na obr. 5.20 pre výkon špirály na obr. 5.21.



Obr. 5. 19 Zobrazenie priebehov teploty pri riadení pomocou PD regulátora



Obr. 5. 20 Priebeh výkonu špirály pri riadení pomocou PD regulátora

5.7 Vyhodnotenie ukazovateľov kvality riadenia

Pre prehľadnejšie zobrazenie dosiahnutých výsledkov pomocou navrhnutých regulátorov pri riadení varáka som zostavil tab. 5.1, v ktorej sú znázornené niektoré ukazovatele kvality riadenia v časovej oblasti:

- Maximálna dosiahnutá teplota y_{\max} [°C]
- Žiadaná teplota $w(\infty)$ [°C]
- Dosiahnutá teplota $y(\infty)$ [°C]
- Trvalá regulačná odchýlka (TRO) [°C] v

$$e(\infty) = |w(\infty) - y(\infty)| \quad (5.52)$$

Typ regulátora a prenosu	y_{\max} [°C]	$y(\infty)$ [°C]	$w(\infty)$ [°C]	$e(\infty)$ [°C]
P (5.9)	50	50	50	0
LQ (5.18)	32,6	32,6	30	2,6
2DOF (5.18)	53,5	53,5	55	1,5
PD (5.18)	22,2	22,2	28	5,8

Tab. 5. 1 Zobrazenie ukazovateľov kvality pre navrhnuté regulátory

Záver

Hlavnou úlohou práce bolo navrhnuť riadenie pre varák rektifikačnej kolóny ako systému s integračnými vlastnosťami. Pri riadení bolo využité prepojenie snímačov a akčných členov s počítačom pomocou pracovnej stanice SIMATIC S7 - 300. Ďalším krokom bolo vytvorenie projektu v programe STEP7 a jeho pripojenie k pracovnej stanici. Po otestovaní manuálneho ovládania v programe STEP7, som pristúpil k tvorbe vizualizačného prostredia v programe WinCC. Na základe fyzických parametrov varáka bol zostavený matematický model varáka, a jeho prechodová charakteristika bola simuláciou v MATLABe porovnaná s reálne nameranou prechodovou charakteristikou. Pre matematický model bol metódou umiestnenia pólov navrhnutý P regulátor, ktorý bol použitý pri riadení pomocou bloku FB41 v programe STEP7. Z priebehu riadenia možno povedať, že vypočítaný P regulátor dosahuje žiadajú veličinu.

Ďalej som vytvoril spojenie programov MATLAB, Microsoft Excel a WinCC pomocou DDE rozhrania. To bolo využité pri posielaní údajov z WinCC prostredníctvom Excelu ako serveru do MATLABu, kde pomocou navrhnutých schém a regulátorov bol posielaný akčný zásah späť do WinCC prostredníctvom Excelu. Pre riadenie pomocou DDE rozhrania bol navrhnutý LQ regulátor a regulátor pomocou optimálnej teórie riadenia pre matematický model varáka po Padeho rozvoji, ktorý zahŕňa do prenosu dopravné oneskorenie. Pri riadení pomocou LQ regulátora došlo k miernemu preregulovaniu žiadanej hodnoty čo mohlo byť spôsobené vlastnosťami LQ regulátora a zotrvačných vlastností špirály varáka. Pri riadení regulátorom navrhnutým pomocou optimálnej teórie riadenia, sa nepodarilo dosiahnuť žiadajú hodnotu a riadenie vykazuje trvalú regulačnú odchýlku. Výsledok riadenia je ovplyvniteľný voľbou váhových koeficientov, ktorými možno nastaviť veľkosť preregulovania a dobu regulácie. Na prenos matematického modelu po Padeho rozvoji bol taktiež navrhnutý PD regulátor metódou umiestnenia pólov. Riadenie sa uskutočňovalo pomocou bloku FB41 v programe STEP7.

Z výsledkov merania možno povedať, najmenšiu regulačnú odchýlku vykazuje P regulátor, ktorý bol navrhnutý pre matematický model (5.9) a použitý pri riadení v STEP7. Výsledky ostatných regulátor sú v Tab. 5.1. Výsledky meraní za podmienok spomenutých v práci sú reprodukovateľné.

Literatúra

- [1] Kožka, Š., Kvasnica, M.,: Programovanie PLC SIMATIC 300 (Základná príručka), KIRP, Bratislava 2001
- [2] SIMATIC, Manual, Standard Software for S7-300 and S7-400 PID Control V5.1
- [3] www4.ad.siemens.de
- [4] www.mathworks.com
- [5] Mészáros, A., Danko, J., Mikleš, J., Bakošová, M.,: Základy automatizácie, STU, Bratislava 1997
- [6] Dostál, P., Meszáros, A., Mikleš, J., (1994). A modified LQ tracking problem, Journal of EE, 45, 4, 129 - 133
- [7] Dostál, P., Bobál, V., Prokop, R.,: The simple controller design for unstable and integrating time delay systems using optimal control theory, 13th Int. Conference on Proces Control (2001)
- [8] Mikleš, J., Fikar, M.,: Modelovanie, identifikácia a riadenie procesov II, STU, Bratislava Verzia 5. 10. 2004
- [9] Ondrovičová, M., Bakošová, M., Čáran, M., Karšaiová, M.: Riadenie rektifikačnej kolóny riadiacim systémom SIMATIC S7-300. *ATP Journal*, č. 1, roč. **XI**, str. 44-45, 2004.

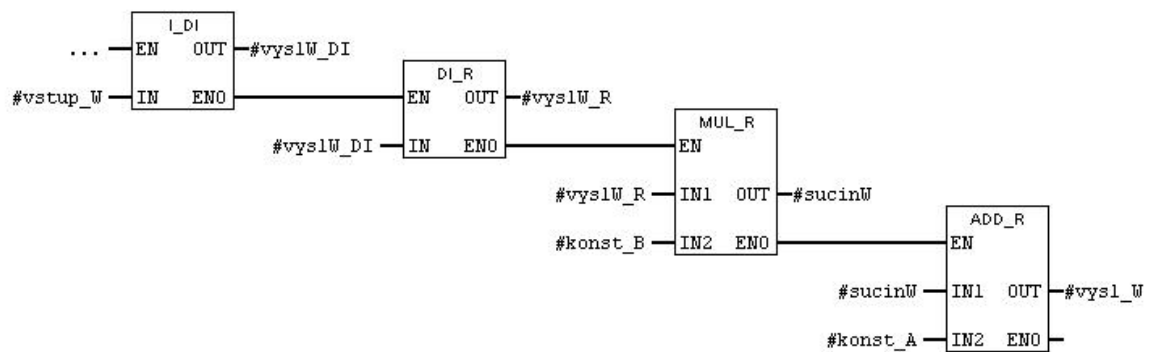
Príloha A

Výpis riadiaceho programu zo STEP7

FB1 : Title:

Comment:

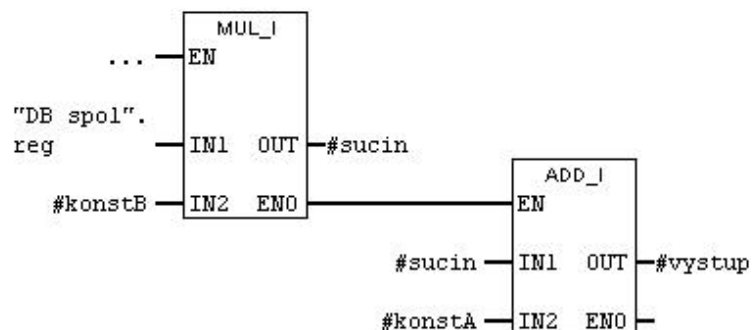
Network 1: Signal z Pt100 pre zvyšok



FB2 : Title:

Comment:

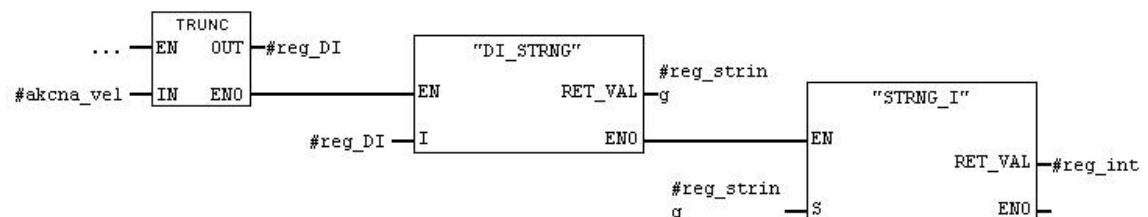
Network 1: Konverzia signalu



FB3 : Title:

Comment:

Network 1: Title:

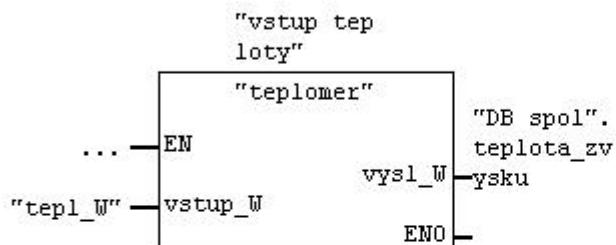


OBI :

Comment:

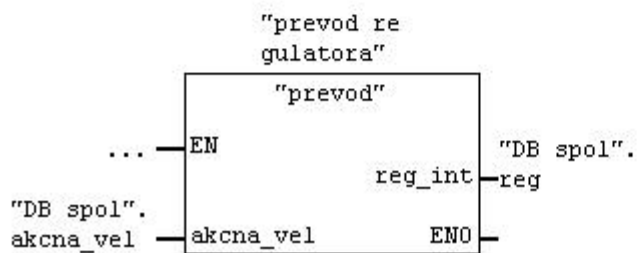
Network 1: Konverzia teploty zvyšku W

Konverzia a nasledna kalibracia, prevod signalu integer (INT) na C



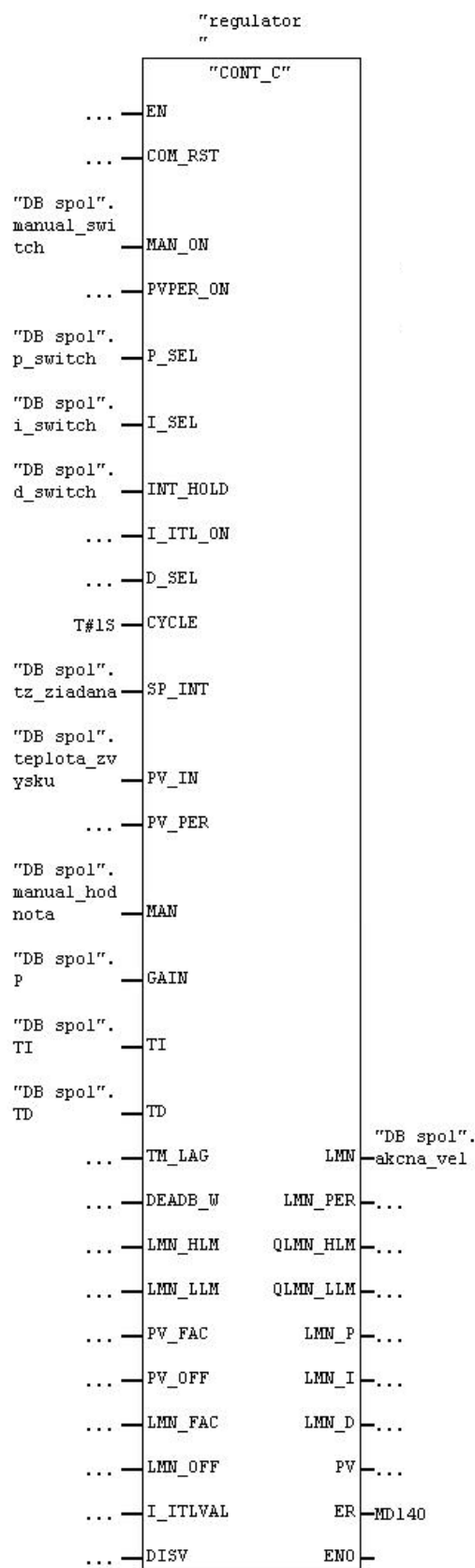
Network 2: Title:

Comment:



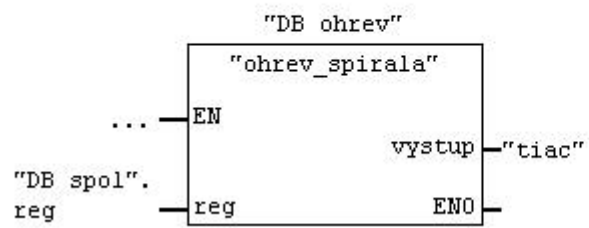
Network 3 : Regulatur FB41

Comment:



Network 4 : Title:

Comment:



Príloha B

Výpis m-file súboru *spojenie2.m* pre spojenie medzi programom Excel a MATLABom

```
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 - chan=ddeinit('excel','List1');
2 - ustaleny_stav=22;
3 - ziadana_vel=ddereq(chan,'r4c3');
4 - merana_vel=ddereq(chan,'r3c3');
5 - akcna_vel=ddereq(chan,'r5c3');
6 - merana_vel=merana_vel(1)
7 - ziadana_vel=ziadana_vel(1)
8 - sim('regulaciapap');
9 - ziadana_vyp=[akcna_vel(1)+akcna_vel(length(akcna_vel))]/2
10 - rc=ddepoke(chan,'r6c3',ziadana_vyp)
11
```

Výpis m-file súboru *casovac.m*, pre spúšťanie výpočtu akčnej veličiny v požadovanom intervale.

```
File Edit Text Window Help
[Icons]
1 cas=timer('TimerFcn','spojenie2','Period',1);
2 set(cas,'ExecutionMode','FixedRate');
3 start(cas)
```

Zobrazenie okna Microsoft Excel *hod.xls* pre riadenie varáka pomocou DDE rozhrania.

Soubor Úpravy Zobrazit Vložit Formát Nástroje Data Okno Nápověda									
[Icons]									
Arial CE 10 B I U [Icons] % 000 +,00 +,00 [Icons]									
M3 =									
	A	B	C	D	E	F	G	H	I
1									
2									
3		teplota vo varáku (°C)	49.9791						
4		žiadaná teplota (°C)	52						
5		výkon špirály (%)	7.15436						
6		akčný zásah (%)	8.154361						
7									
8									
9									

Výpis m-file súboru *LQvyp.m*, pre výpočet prenosu regulátora vid' rovnica (5.27).

```

File Edit View Text Debug Breakpoints Web Window Help
[Icons] [Stack: B]

1  %padeho rozvoj a vyp. LQ reg.
2  g1=tf([0.0113],[1 0])
3  [citp,menp]=pade(160,1);
4  g2=tf([citp],[menp]);
5  G=g1*g2;
6  [cit2,men2]=tfdata(G,'v')
7
8  Bp = mat2pol(cit2);
9  Ap = mat2pol(men2);
10 Apl=pol2mat(Ap);
11 Bpl=pol2mat(Bp);
12 %vahove koeficienty
13 fip=5;
14 psip=1;
15 %faktorizacia
16 Ahp=Ap';
17 Bhp=Bp';
18
19 Y1lp=Ap*Ahp
20 Y1lp=-s^2+1
21 [Dfp,jp]=spf(Y1lp)
22 %%%
23 Y2lp=fip*Ahp*Ap+psip*Bhp*Bp;
24 [Dcp,jp]=spf(Y2lp);
25 M=Dcp*Dfp
26 %%%
27 [Pap,Qp]=axbyc(Ap,Bp,Dcp*Dfp)
28 Plp=pol2mat(Pap)
29 Qlp=pol2mat(Qp)
30 %diskretny prenos padeho rozvoja
31 [arp,brp,crp,drp] = TF2SS(Qlp,Plp);
32 srpl=ss(arp,brp,crp,drp);
33 Tvp=0.1;
34 srp2=c2d(srpl,Tvp);
35 [adrp,bdrp,cdrp,ddrp]=ssdata(srp2);
36 [Qldp,Pldp]=ss2tf(adrp,bdrp,cdrp,ddrp)

```

Výpis m-file súboru *funpapier.m*, pre výpočet parametrov g_1 a g_2 vid' rovnicu (5.42)

```
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 function F=funpapier(x)
2 K=0.0113;tau=160;ni=0.01;fi=500;
3 F=[sqrt(4*K/tau*(sqrt(ni/fi))*x(2)+K^2*ni/fi)-x(1);sqrt(2/sqrt(fi)*x(1)+4/tau^2)-x(2)]
4
5 %spustanie v MATLABe fsolve('funpapier',[1 1]), kde [1 1] su pociatocne hodnoty vypoctu
6
```

Výpis m-file súboru *koefpapg.m*, pre výpočet parametrov g_0 , q_2 , q_1 , q_0 , p_0 vid' rovnice (5.41 a 5.42)

```
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 K=0.0113;tau=160;ni=0.01;fi=500;
2 g0=2*K/tau*sqrt(ni/fi)
3 p0=g2+(tau/4)*(2*g1+tau*g0)
4 q0=g0/K
5 q1=1/K*(g1+tau*g0)
6 q2=(tau/(4*K))*(2*g1+tau*g0)
7 %prenos Q
8 citq=[q2 q1 q0]
9 menq=[1 p0 0]
10 %prenos R
11 citr=[q0]
12 menr=[1 p0 0]
13
14 %diskretny tvar pre Q prenos
15 [aq,bq,cq,dq] = TF2SS(citq,menq);
16 sql=ss(aq,bq,cq,dq);
17 Tvq=0.1;
18 sq2=c2d(sql,Tvq);
19 [aq1,bq1,cq1,dq1]=ssdata(sq2);
20 [Qq1,Pq1]=ss2tf(aq1,bq1,cq1,dq1)
21
22 %diskretny tvar pre R prenos
23 [arz,brz,crz,drz] = TF2SS(citr,menr);
24 srz1=ss(arz,brz,crz,drz);
25 Tvr=0.1;
26 srz2=c2d(srz1,Tvr);
27 [arz1,brz1,crz1,drz1]=ssdata(srz2);
28 [Qr1,Pr1]=ss2tf(arz1,brz1,crz1,drz1)
29
```