

TEPELNO-OPTICKÁ SÚSTAVA uDAQ28/LT

TECHNICKÝ A UŽÍVATEĽSKÝ MANUÁL

Matlab[®] a Simulink[®] sú ochrannými známkami produktov spoločnosti
The MathWorks, Inc.

© Copyright 2006 Mikuláš Huba, Peter Kurčík, Martin Kamenský

Obsah

1 TECHNICKÉ ÚDAJE	4
A) MERACÍ A KOMUNIKAČNÝ SYSTÉM UDAQ28/LT.....	4
B) GRAFICKÉ CHARAKTERISTIKY PROCESOV.....	5
C) KOMUNIKAČNÝ INTERFACE V MATLABE.....	6
D) SÚPIS DODÁVANÉHO HARDVÉRU	6
E) SÚPIS DODÁVANÉHO SOFTVÉRU.....	6
2 INŠTALÁCIA A UPDATE.....	7
2.1 INŠTALÁCIA OVLÁDAČA ZARIADENIA	7
2.2 INŠTALÁCIA OVLÁDAČA A SOFTVÉROVÉHO BALÍKU V MATLABE	11
2.3 UPDATE FIRMWARU MIKROPROCESORA	11
3 POPIS KOMUNIKÁČNÉHO ROZHRANIA UDAQ28/LT PRE MATLAB	12
3.1 REÁLNY ČAS A OS WINDOWS.....	12
3.2 PRÁCA V MATLABE POČAS VYKONÁVANIA EXPERIMENTU V REÁLNO M ČASE.....	12
3.3 KOMUNIKAČNÉ ROZHRANIE TEPELNO-OPTICKEJ SÚSTAVY	13
<i>Serial port</i>	13
<i>Sample time (sec)</i>	14
<i>Read timeout</i>	15
<i>Matlab priority</i>	15
<i>Sampling delayed (%)</i>	15
<i>Warning if delayed (error if unchecked)</i>	15
<i>Detailed timing and timeout printout</i>	15
<i>Automatické nastavenie parametrov simulácie</i>	16
3.4 UKLADANIE DÁT EXPERIMENTU.....	17
3.5 ZÁKLADNÁ VSTUPNO-VÝSTUPNÁ SIMULAČNÁ SCHÉMA	18
4 PROBLÉMY A ICH RIEŠENIE.....	19
5 BALÍK SIMULAČNÝCH SCHÉM A RIADIACICH ALGORITMOV	21
5.1 PRÍKAZY <i>EXLIST</i> A <i>EXNUM</i>	21
5.2 PRÍKAZY <i>MFILES</i> , <i>MDL</i> , <i>MAT</i> , <i>MOPEN</i> A <i>COPEN</i>	21
5.3 ZOZNAM EXPERIMENTOV	22

1 Technické údaje

A) Merací a komunikačný systém uDAQ28/LT

Vstupy:

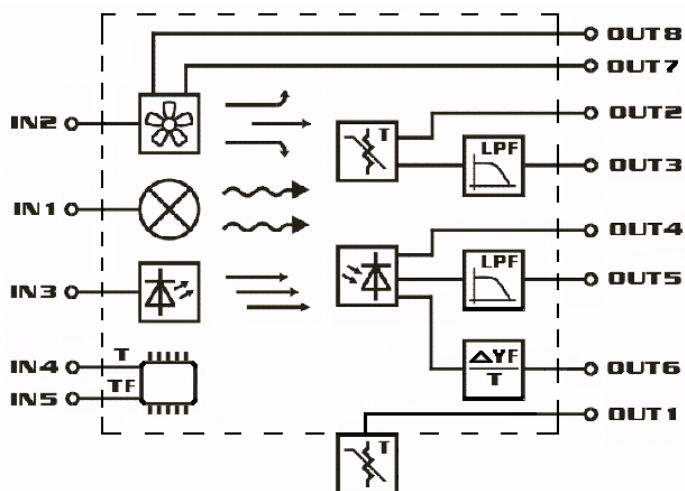
Žiarovka	0-5V na 0-20W svetelného výkonu
Ventilátor	0-5V na 0-6000 ot/min ventilátora
LED	0-5V na 0-100% svetelného výkonu LED
T, D	údaje pre mikroprocesor za účelom výpočtu filtrovanej derivácie svetelného výstupu (perióda vzorkovania svetelného výstupu v ms - minimálna hodnota je 1 - a koeficient aktuálnej vzorky diskrétného filtra 1. rádu s presnosťou na 3 desatinné miesta)

Výstupy:

Okolitá teplota	senzor PT100
Teplota	rozsah 0-100°C
	presnosť: lepšia ako 99%
Filtrovaná teplota	(filter 1. rádu, časová konštanta cca 20s)
Svetelná intenzita	
Filtrovaná svetelná intenzita	(filter 1. rádu, časová konštanta cca 20s)
Filtrovaná derivácia priameho kanála svetelnej intenzity	
Prúd odoberaný ventilátorom (0-50 mA)	
Otáčky ventilátora (0-6000 ot/min)	

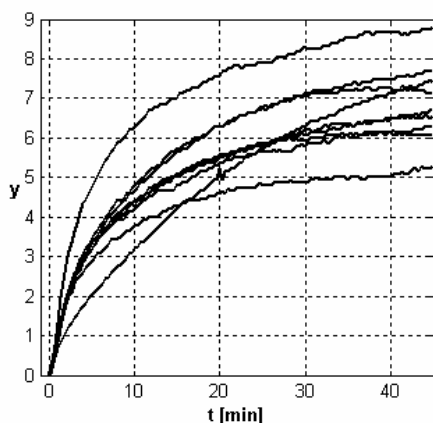
Rozsah pracovných teplôt: 0-70°C
 Napájanie: 12V/2A DC z externého adaptéra

Komunikačné rozhranie: USB – virtuálny sériový port
 Rýchlosť prenosu dát 250 kbit/s

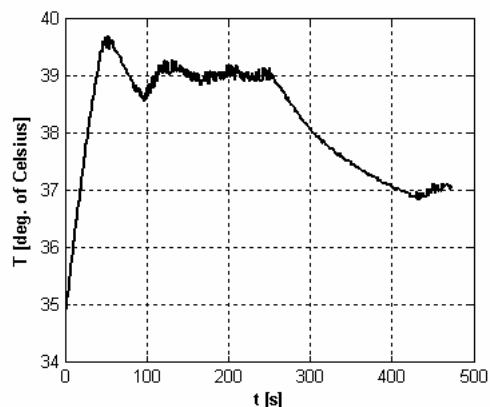


Obr. 1.1 Základná elektrická schéma tepelno-optickej sústavy uDAQ28/LT

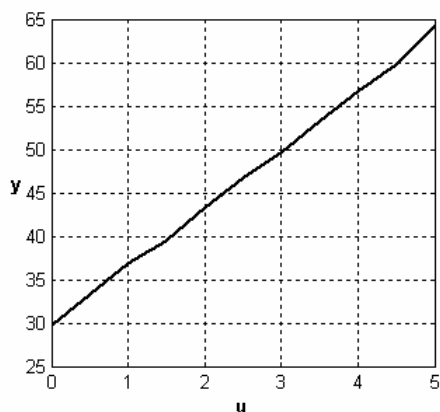
B) Grafické charakteristiky procesov



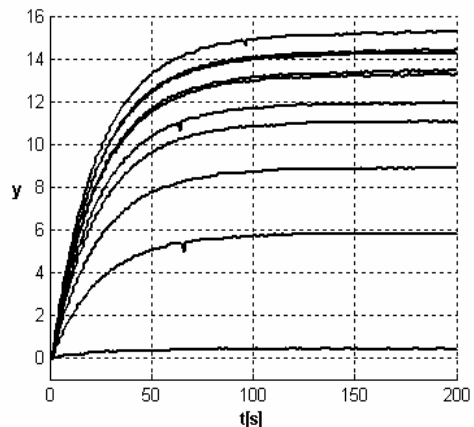
Obr. 1.2 Rodina prechodových charakteristík filtrovaného tepelného kanála



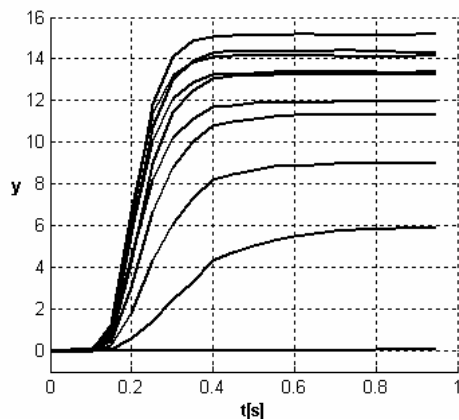
Obr. 1.3 Príklad riadenia tepelného procesu



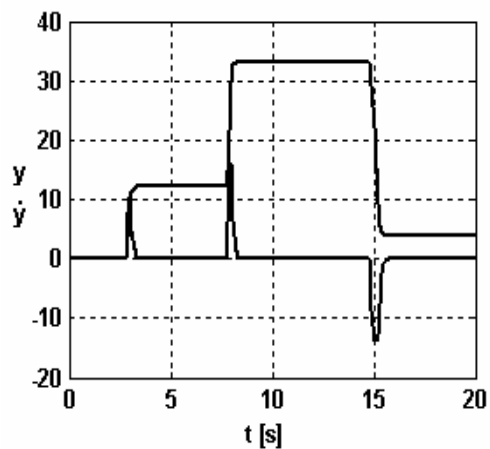
Obr. 1.4 Prevodová charakteristika tepelného kanála



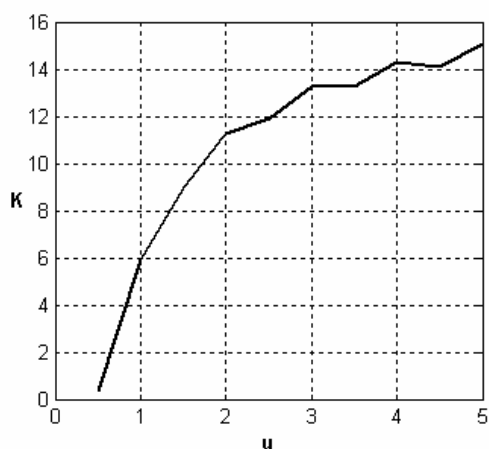
Obr. 1.5 Rodina prechodových charakteristík filtrovaného optického kanála



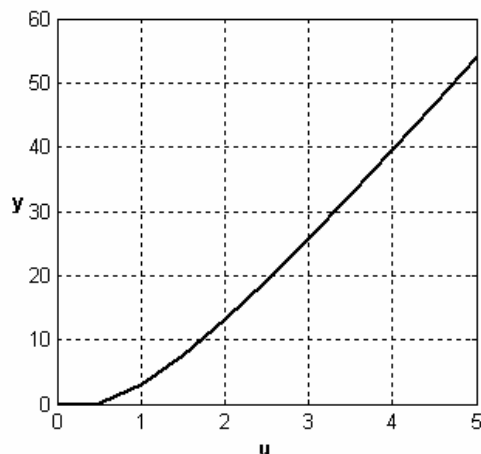
Obr. 1.6 Rodina prechodových charakteristík optického kanála



Obr. 1.7 Meranie derivácie výstupu svetelného kanála



Obr. 1.8 Zosilnenie otvoreného obvodu optického kanála



Obr.1.9 Prevodová charakteristika optického kanála

C) Komunikačný interface v Matlabe

- perióda vzorkovania cca 40-50 ms
(postačujúca perióda vzorkovania pre kvázispojité riadenie je vzhľadom na dynamiku systému cca 1 sekunda)
- spodná hranica timeoutu pre čítanie dát z portu cca 25 ms
- možnosť nastaviť vyššiu prioritu pre Matlab ako je priorita bežnej aplikácie bežiackej pod OS Windows
- nie je potrebné buildovať simulačnú schému, z čoho vyplýva možnosť použiť simulinkovský blok *Matlab Fcn*, ktorý môže vykonávať algoritmus napísaný v m-súbore
- nastavenie všetkých parametrov komunikácie v 1 okne grafického užívateľského rozhrania.

D) Súpis dodávaného hardvéru

Merací a komunikačný systém uDAQ28/LT	1 ks
Stabilizovaný napájací zdroj	1 ks
Žiarovka	1 ks
Demontovateľný teplovzdušný tunel s otáčacou hlavicou	1 ks
Usb kábel	1 ks

E) Súpis dodávaného softvéru

Ovládač zariadenia pre OS Windows XP (prípadne inú verziu OS Windows)
Dve verzie ovládačov zariadenia pre danú verziu Matlabu
Balík simulačných schém implementovaných v Simulinku, balík riadiacich algoritmov a podporných príkazov vo forme matlabovských súborov

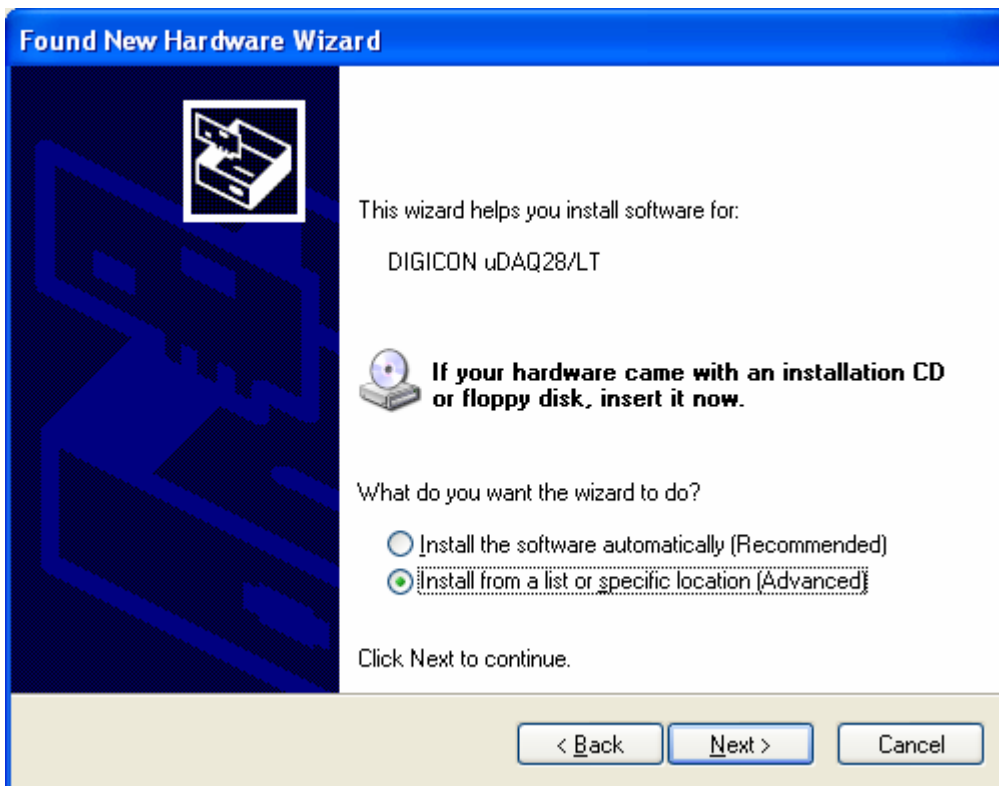
2 Inštalácia a update

2.1 Inštalácia ovládača zariadenia

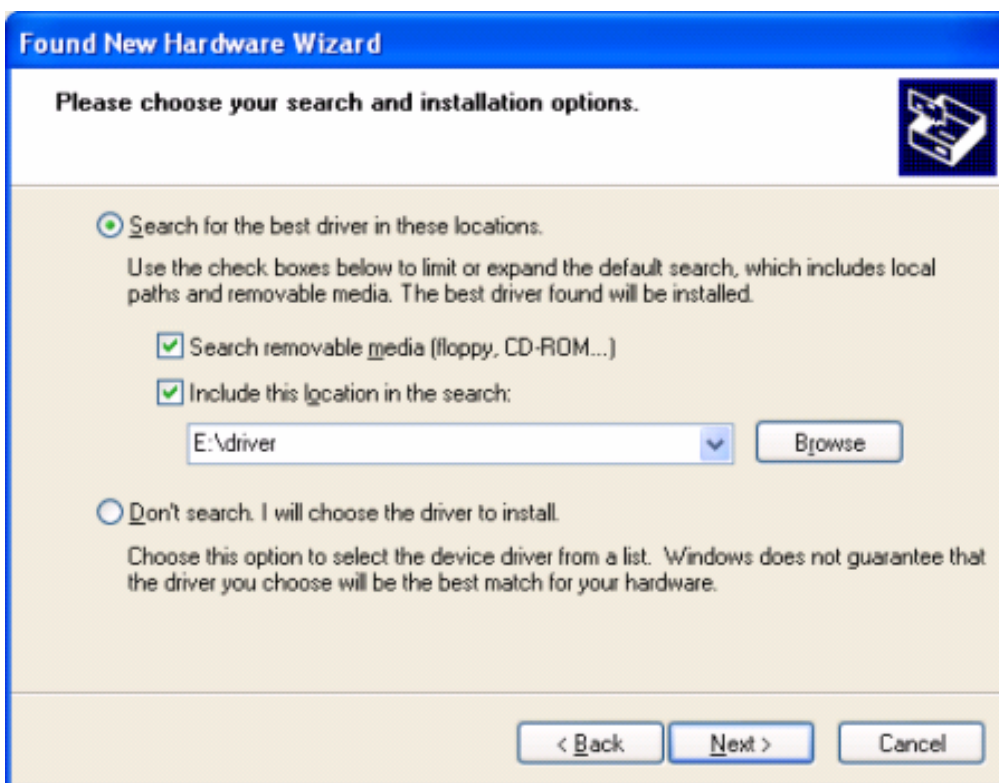
Po pripojení zariadenia do elektrickej siete napájacím káblom a pripojení k PC operačný systém automaticky detekuje nový hardvér a spustí sprievodcu inštaláciou nového hardvéru (obr. 2.1). Ak sa tak nestane, treba skúsiť USB kábel pripojiť k inému USB portu počítača alebo ho od aktuálneho USB portu PC odpojiť a opäť pripojiť. Pridanie nového hardvéru je možné spustiť v OS Windows aj cez Start -> Control Panel -> Add new hardware. Ďalej vyberieme inštaláciu ovládača z konkrétneho umiestnenia (obr. 2.2).



Obr. 2.1 Sprievodca inštaláciou nového hardvéru

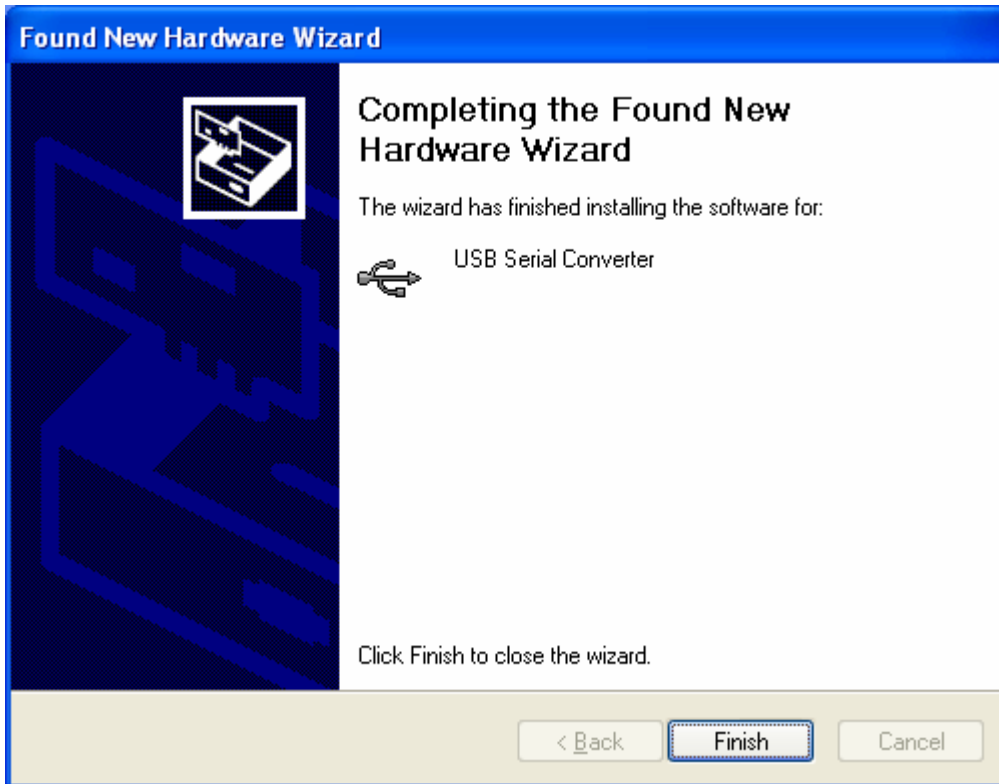


obr. 2.2 Voľba inštalácie zo známej lokalizácie



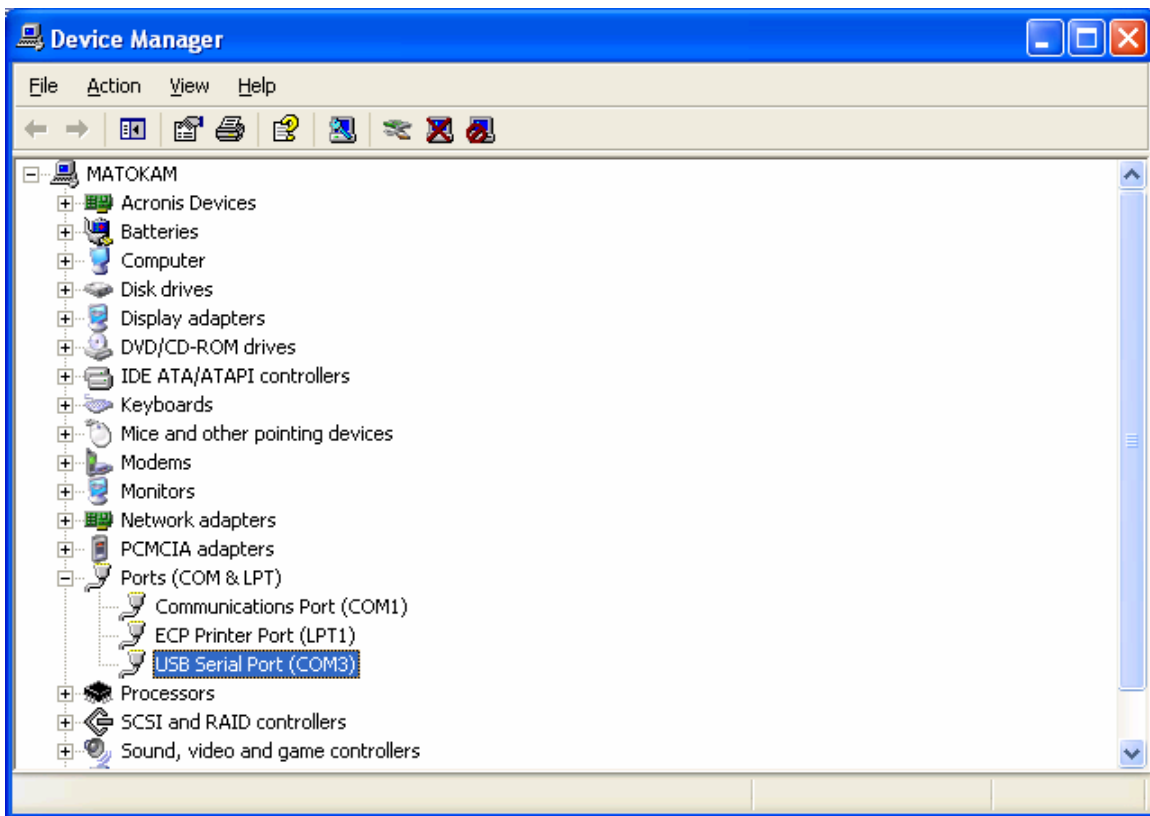
obr. 2.3 Výber umiestnenia ovládača zariadenia

Umiestnenie ovládača zariadenia môžeme vybrať priamo v adresári *driver* na CD-ROM jednotke (obr. 2.3). Kompletná inštalácia pozostáva z 2 krokov pridávania nového hardvéru (do operačného systému sa inštaluje *USB Serial Converter* a *USB Serial Port*, sprievodca pridávaním hardvéru sa spustí automaticky aj druhýkrát). O úspešnom dokončení inštalácie ovládača nás informuje dialógové okno (obr. 2.4).

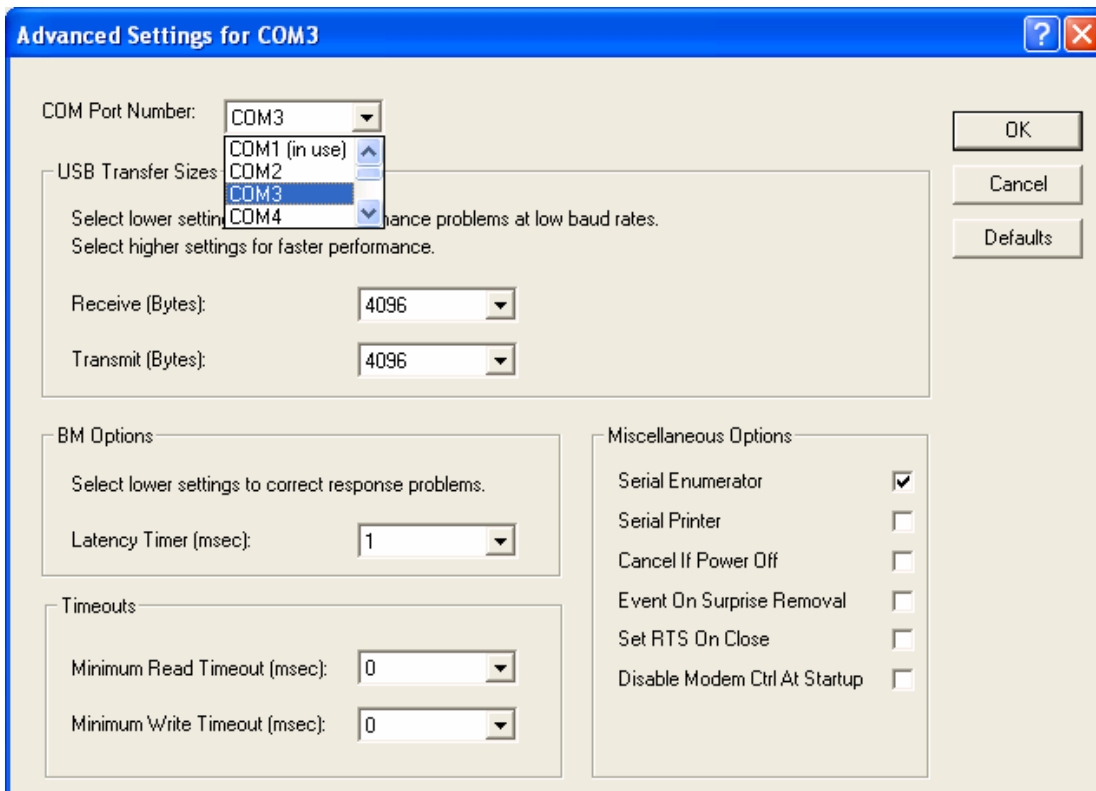


obr. 2.4 Úspešné dokončenie inštalácie

Po nainštalovaní zariadenia je v systéme - zozname hardvérových zariadení - vytvorený (virtuálny) sériový port, ktorý je označený ako *USB Serial Port*. Dostaneme sa k nemu cez *Device Managera* (*Start* → *Settings* → *Control Panel* → *System* → *Hardware* → *Device Manager*) výberom položky *Ports (COM & LPT)* → *USB Serial Port*. (obr. 2.5). Systém mu automaticky priradí jedno z neobsadených čísel com portov. Ak je priradené číslo väčšie ako 4, je ho potrebné na karte portu zmeniť na jedno zo zatiaľ neobsadených čísel v rozsahu 1 až 4. Urobíme to výberom záložky *Port Settings* → *Advanced* → *COM Port Number* (obr. 2.6). Na tej istej záložke nastavíme parameter *Latency Timer (msec)* na hodnotu 1, čím minimalizujeme percento straty dát pri ich prenose (obr. 2.6).



obr. 2.5 USB Serial Port



obr. 2.6 Priradenie čísla COM portu USB Serial Portu, nastavenie Latency Timera

2.2 Inštalácia ovládača a softvérového balíka v Matlabe

Inštaláciu ovládača a softvérového balíku v Matlabe vykonáme nasledovne:

- 1) Pred inštaláciou je vhodné skontrolovať, či v pamäti „nevisí“ neukončený proces *matlab.exe*; ak áno, treba ho ukončiť (položka *Processes* v *Task Manageri*, pomocou kláves CTRL+ALT+DEL)
- 2) Spustíme Matlab
- 3) Pracovný adresár zmeníme na lokalizáciu, kde sa nachádza CD-ROM jednotka, resp. inštalačný súbor ***udaq_setup.p*** a adresár *files* (napr. príkazom `cd e:`)
- 4) Vykonáme súbor ***udaq_setup.p*** – príkaz `udaq_setup`

Po úspešnom zbehnutí inštalácie je na lokálnom disku PC vytvorený adresár *matlabroot\udaq\udaq28LT* s príslušnými podadresármi a súbormi.

(*matlabroot* reprezentuje meno adresára, ktorého meno získame po zadaní príkazu *matlabroot* v príkazovom okne Matlabu)

Po dokončení inštalácie sa otvorí vzorová mdl schéma ***udaq28LT_iov2.mdl***, ktorá sa nachádza v *matlabroot\udaq\udaq28LT\examples* a zároveň knižnica 2 blokov (driverov), ktorú reprezentuje mdl súbor ***matlabroot\udaq\udaq28LT\blks\udaq28LT_lib.mdl***. Vlastnú simulačnú schému komunikujúcu s tepelno-optickou sústavou si užívateľ môže vytvoriť skopírovaním bloku drivera z knižnice (alebo iného funkčného mdl súboru) do vlastného mdl súboru.

Pri prípadnom zlyhaní inštalácie treba urobiť tri kroky:

1. Vytvoriť adresár *matlabroot\udaq\udaq28LT*
2. Nakopírovať obsah adresára *files* nachádzajúci sa na CD do adresára *matlabroot\udaq\udaq28LT*
3. Pridať do zoznamu matlabovských ciest adresár *matlabroot\udaq\udaq28LT\blks* (príkaz `pathtool` alebo cez menu: File -> Set Path ...) a uložiť aktualizovaný zoznam aj pre nasledujúce Matlab sessions.

2.3 Update firmwaru mikroprocesora

Na update firmwaru mikroprocesora slúži utilita ***update.exe***, ktorá hľadá pripojené zariadenie a následne má užívateľ možnosť vybrať súbor, ktorým bude firmware prepísaný. Jedná sa o zásah, ktorý je potrebný len v odôvodnených prípadoch, prípadne pri updatovaní verzie firmwaru výrobcom. V prípade potreby bude užívateľovi e-mailom poslaný potrebný súbor.

3 Popis komunikačného rozhrania uDAQ28/LT pre Matlab

3.1 Reálny čas a OS Windows

Operačný systém Windows nie je navrhnutý ako OS reálneho času, preto sa pri snahe pracovať s reálnym časom stretáme s dvoma problémami.

Prvým je, že časovače (timers) vo Windowse pracujú len s určitou presnosťou, t.j. na niekoľko milisekúnd. Takýmto spôsobom pracuje blok ***udaq28LT_v1R13*** z udaq-knižnice. Ak použijeme blok ***udaq28LT_v2R13*** (ktorý nepoužíva priamo windows časovač), dosiahneme vyššiu presnosť časovania, ale za cenu vysokého vyťaženia procesora.

(Je na užívateľovi, pre ktorý blok sa rozhodne; vzhľadom na čas. konštanty sústavy nie je nutné používať blok ***udaq28LT_v2R13***, pri voľbe malej periódy vzorkovania (cca pod 0,3 s) alebo nízkej hodnoty timeoutu pre čítanie dát z portu je to však lepšia voľba. Ak chce užívateľ vždy dosiahnuť maximálnu presnosť časovania, ktorú driver poskytuje, defaultne použije blok ***udaq28LT_v2R13***.)

Druhým problémom je multitasking, pridelenie procesora a systémových prostriedkov striedavo aktuálne bežiacim procesom. Tento sa dá značne eliminovať, ak priradíme Matlabu prioritu vyššiu ako je priorita bežnej Windows aplikácie. Na to slúži parameter *Matlab priority* v grafickom komunikačnom rozhraní tepelno-optickej sústavy popísanom v časti 3.2.

3.2 Práca v Matlabe počas vykonávania experimentu v reálnom čase (dodržanie periódy vzorkovania)

Nasledovné upozornenie súvisí so spôsobom vykonávania Matlabu samotného. Nastavenie vyššej priority Matlabu síce zabezpečí, aby bola perióda vzorkovania dodržaná aj pri iných spustených aplikáciách okrem Matlabu, týka sa však Matlabu ako celku. Preto rôzne iné akcie vykonávané v Matlabe počas experimentu môžu viesť k časovým oneskoreniam v komunikácii so sústavou (či pozastaveniu komunikácie až do ukončenia inej činnosti v Matlabe).

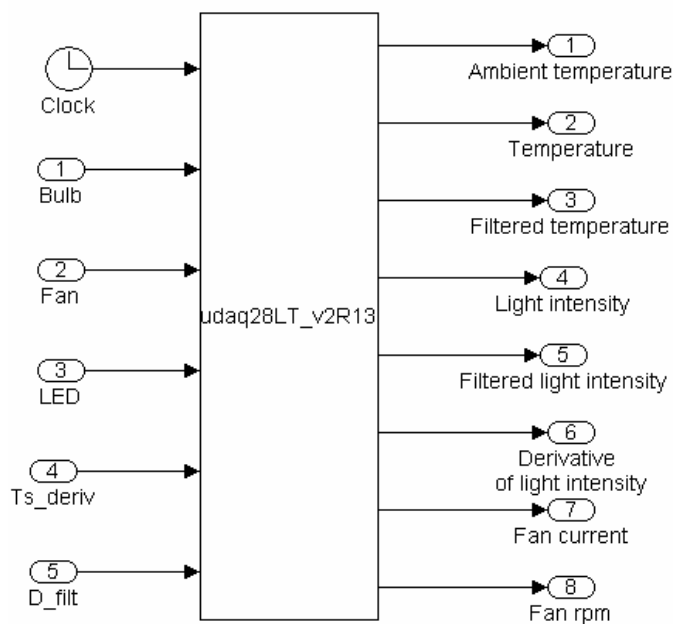
Jedná sa najmä o činnosti súvisiace s grafikou, preto sa ich *neodporúča* vykonávať počas simulácie, najmä pri malých hodnotách periódy vzorkovania (resp. pri malých hodnotách rozdielu medzi nastavenou periódou vzorkovania a nastaveným časom pre timeout čítania z portu, cca do 0,3 s)). Sú to napr. otváranie okien, zmena mierky v okne, výber položiek (popup) menu, posúvanie slidera v bloku slider-gain, drag and draw operácie, ale tiež napríklad samotné spustenie a zastavenie aplikácie je z tohto hľadiska vhodnejšie pomocou rýchlych kláves CTRL+T a CTRL+C ako pomocou príslušných tlačítk.

Približný čas, na ktorý jednotlivé operácie zamestnajú procesor, je možné zisťovať pri hraničnom nastavení pre periódu vzorkovania a zaškrtnutými voľbami *Warning if delayed (error if unchecked)* a *Detailed timing and timeout printout* v okne komunikačného rozhrania.

Nedá sa stopercentne vyhnúť výskytu *timeout* pri čítaní dát zo zariadenia prípadne občasnému nedodržaniu malej periódy vzorkovania, každú takúto udalosť však komunikačné rozhranie **detekuje** a podľa nastavení volieb užívateľa v okne komunikačného rozhrania vykoná **príslušnú akciu** (pozri časť 3.3) !

3.3 Komunikačné rozhranie tepelno-optickej sústavy

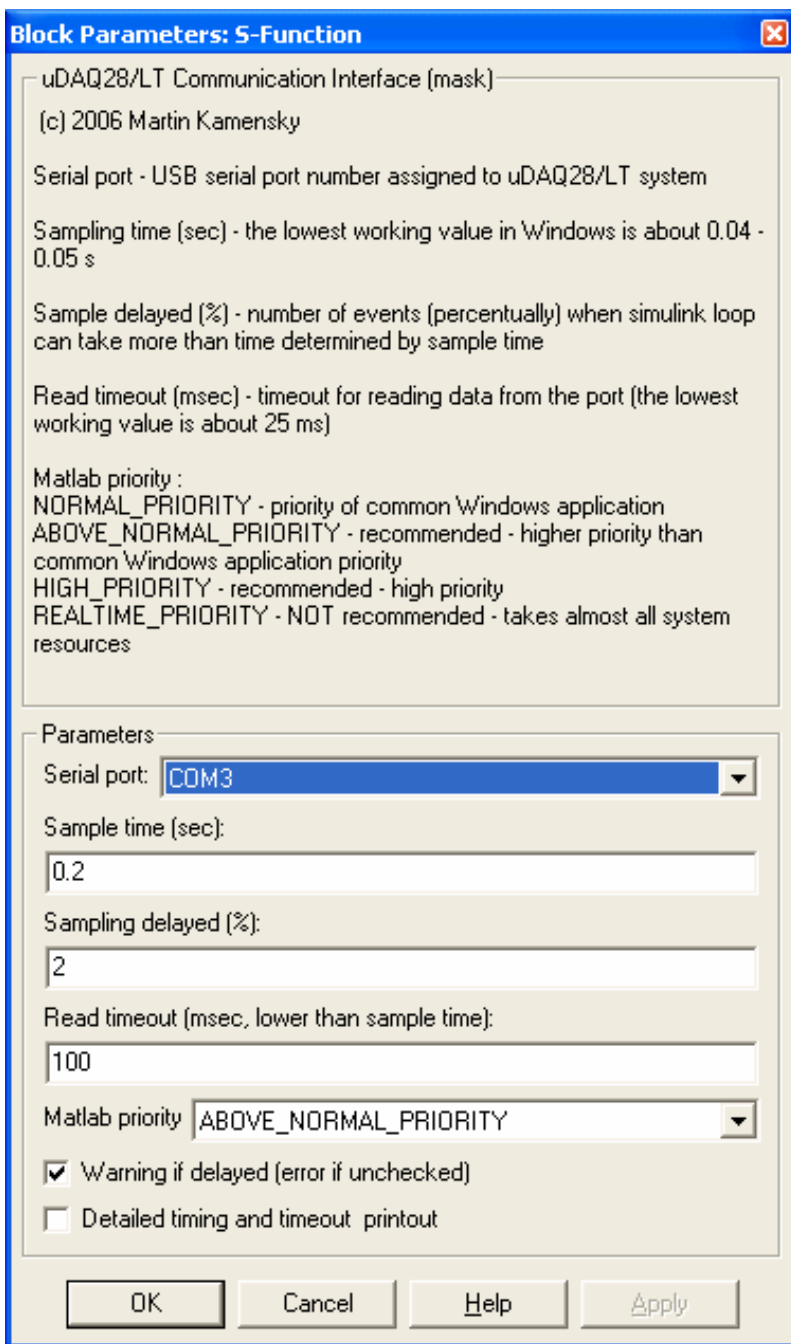
je v Matlabe reprezentované 1 z blokov **udaq28LT_v1R13** (obr. 3.1) alebo **udaq28LT_v2R13**, ktoré sa nachádzajú v knižnici **matlabroot\udaq\udaq28LT\blks\udaq28LT_lib.mdl**. Ich jednotná grafická podoba je znázornená na obr. 3.2, význam zadávaných parametrov je nasledovný:



obr. 3.1 Simulinkovský blok komunikačného rozhrania

Serial port

Číslo sériového portu, musí byť zhodné s číslom sériového portu priradeným zariadeniu (pozri kapitolu 2.1 *Inštalácia ovládača zariadenia*)



obr. 3.2 Komunikačné rozhranie tepelno-optickej sústavy

Sample time (sec)

Periódá vzorkovania simulačnej schémy (komunikácie s procesom) zadávaná v sekundách. Spodná hranica periódy vzorkovania, s ktorou systém dokáže komunikovať pod OS Windows je približne 0.04 – 0.05 sec (je závislá od výkonnosti počítača a rýchlosti prístupu k portu).

Periódá vzorkovania musí mať väčšiu hodnotu (dostatočnú časovú rezervu) ako je hodnota timeoutu (parameter *Read timeout*).

Read timeout

Timeout v milisekundách pre načítanie dát z portu. Spodná hranica timeoutu pre OS Windows je približne 25 ms (je závislá od výkonnosti počítača a rýchlosti prístupu k portu). Perióda vzorkovania musí mať väčšiu hodnotu (dostatočnú časovú rezervu) ako je hodnota timeoutu.

Matlab priority

Priorita, s ktorou bude pod OS Windows vykonávaný program Matlab (proces matlab.exe). Odporúčané hodnoty sú ABOVE_NORMAL_PRIORITY alebo HIGH_PRIORITY. Pri týchto nastaveniach má Matlab pridelené vysoké percento procesorového času a systémových prostriedkov.

(Tým sa značne zvýši úspešnosť dodržania nastavených parametrov komunikácie v reálnom čase aj pri bežiacich iných aplikáciách súčasne s Matlabom.)

Sampling delayed (%)

Udáva maximálnu dovolenú početnosť (v percentách) výskytu chýb v komunikácii, pričom sleduje tieto 2 typy chyby:

- Timeout čítania dát z portu (načítanie dát neprebehne za dobu nastavenú parametrom *Timeout*)
- Predchádzajúci krok simulácie nebol ukončený za čas určený pre 1 periódu vzorkovania (parametrom *Sampling time*)

V kombinácii s nastavením hodnoty parametra

Warning if delayed (error if unchecked)

sa vykonávanie simulácie správa podľa tabuľky 3.1.

Detailed timing and timeout printout

Pri zaškrtnutí tohto parametra sa navyše výskyt chyby vypisuje aj počas simulácie do Matlab command window, napr.:

```
Timeout occurred ... getting last read values  
Sampling delayed and read timeout 1 of 103
```

```
Delay = 90 ms  
Sampling delayed and read timeout occurrence 0.42 % : 2 in 473 samplings
```

Pozn.: pri nastavení technicky neuskutočniteľných parametrov simulácie môže byť početnosť chýb vyššia ako 100% (max. 200%) - napr. keby sa obidva typy chyby vyskytli viac ako v polovici vzoriek.

Sampling delayed (%)	Warning if delayed (...)	Reakcia na výskyt chyby počas vykonávania simulácie	Akcia po ukončení simulácie
0	False	okamžité ukončenie simulácie	výpis početnosti chýb
viac ako 0	False	ukončenie simulácie, ak je početnosť chýb väčšia ako hodnota parametra <i>Sampling delayed</i> *	výpis početnosti chýb
0	true	žiadna	výpis početnosti chýb; výpis WARNINGU, ak je početnosť chýb väčšia ako hodnota parametra <i>Sampling delayed</i>
viac ako 0	True	žiadna	výpis početnosti chýb; výpis WARNINGU, ak je početnosť chýb väčšia ako hodnota parametra <i>Sampling delayed</i>

Tab. 3.1 Reakcia na výskyt chyby v komunikácii podľa užívateľských volieb

* najskôr však po prvej chybe, ktorá sa vyskytne buď po 10 sekundách simulácie alebo po 100 periódach vzorkovania (podľa toho, čo nastane skôr); dôvodom je vysoká percentuálna početnosť chýb pri výskyte aj 1 chyby v malom počte vzoriek

Automatické nastavenie parametrov simulácie

Pred samotnou komunikáciou Matlabu s tepelno-optickým procesom (t.j. pred spustením vykonávania simulačnej schémy) sa automaticky nastaví nasledovné parametre simulácie:

Parameter *Fixed step size* v menu *Simulation->Simulation parameters* (na prvej záložke *Solver*) sa nastaví na rovnakú hodnotu, ako bola zadaná hodnota pre periódu vzorkovania v komunikačnom rozhraní pre sústavu.

Na tej istej záložke sa nastaví aj hodnota parametra *Solver* na metódu *ode1*.

(Automatické nastavenie týchto parametrov sa vykoná aj kliknutím na tlačítko *OK* alebo *Apply* na karte komunikačného rozhrania sústavy.)



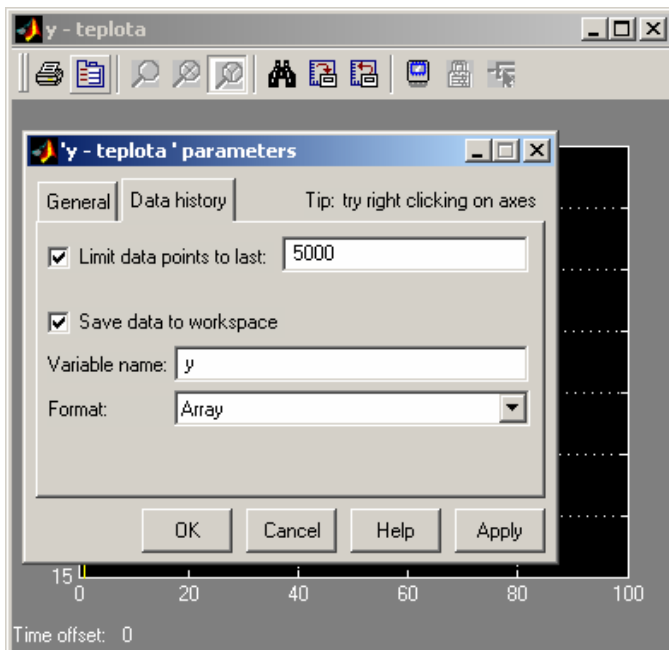
Upozornenie: Zadanie metódy parametra *Solver* vyššieho stupňa ako 1 by viedlo k nekorektnej komunikácii so zariadením !

Parameter *SolverMode* (tiež v menu *Simulation->Simulation parameters*, záložka *Solver*) sa nastaví na *SingleTasking*, čo umožňuje vykonávanie schémy s použitím viacerých periód vzorkovania s rôznymi hodnotami.

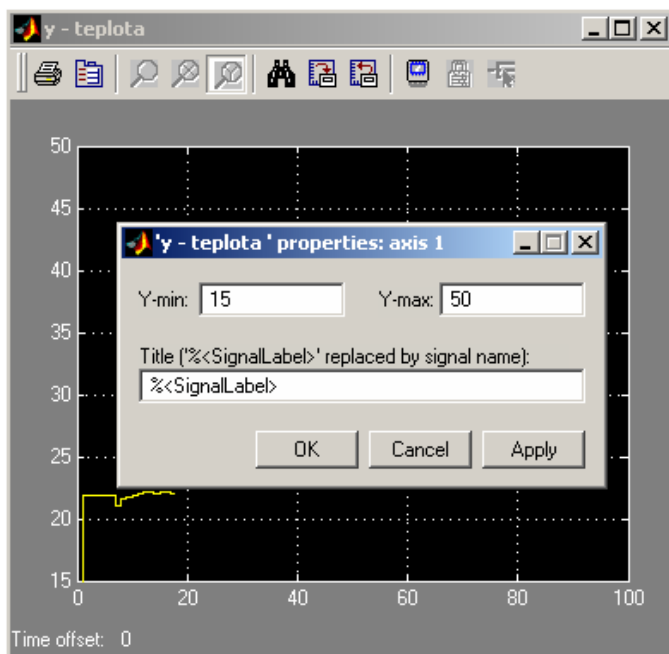
Posledným príkazom, ktorý sa automaticky vykoná pred spustením simulácie, je vypnutie pozastavenia výpisov po nastavenom počte riadkov v príkazovom okne Matlabu.

3.4 Ukladanie dát experimentu

Ukladanie zaznamenávaných dát do premenných vo workspace Matlabu zabezpečíme v blokoch *Scope* zaškrtnutím políčka *Save data to workspace* (príslušné okno sa otvoríme cez ikonu *Parameters*). Tiež tu nastavujeme meno premennej, jej dátový typ a počet ukladaných vzoriek (obr. 3.3). Pri voľbe dátového typu *Array* budú v prvom stĺpci poľa uložené hodnoty časového vektora.



obr. 3.3 Nastavovanie vlastností zaznamenávania I/O signálov



obr. 3.4 Nastavenie zobrazovaného rozsahu signálu

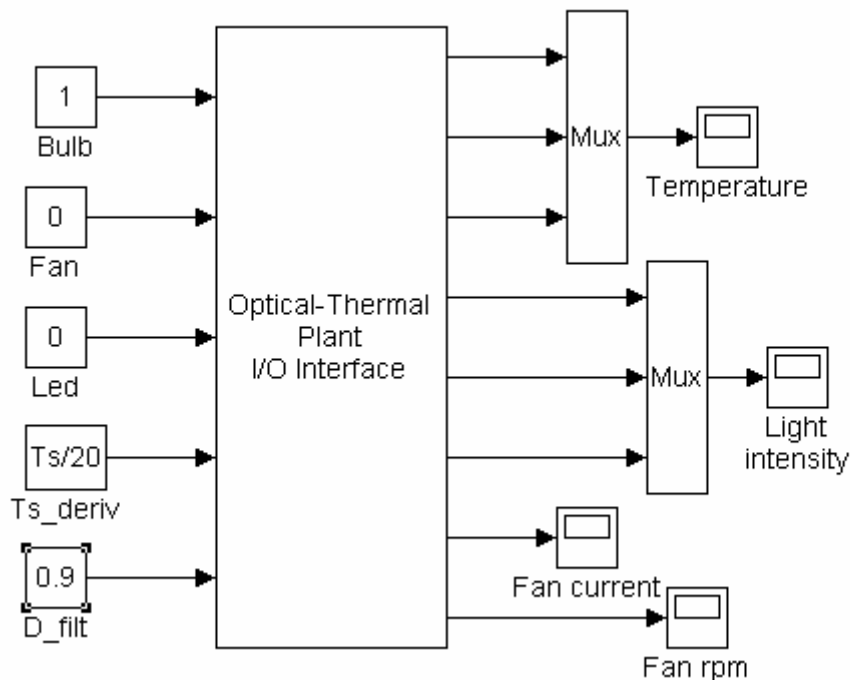
Na záložke *General* toho istého okna zadávame hodnotu periódy vzorkovania (môže ostať nulová) a volíme si rozsah x-ovej (časovej) osi. Nastavenie y-ovej osi (rozsah meraného signálu) možno zmeniť po kliku na pravé tlačítko myši v priestore otvoreného bloku *Scope* a výbere *Axes properties* (obr. 3.4).

3.5 Základná vstupno-výstupná simulačná schéma

Na obrázku 3.5 je znázornená základná vstupno-výstupná simulačná schéma s označenými vstupmi a výstupmi zo sústavy. Subsystem označený “Thermo-Optical Plant I/O Interface” obsahuje blok s odkazom na spomínané komunikačné rozhranie (obr. 3.1)

- ❗ **Upozornenie :** Prvé 3 vstupy do sústavy majú zmysel len v rozsahu 0 až 5 Voltov !
- Hodnoty mimo tohto rozsahu pretečú a transformujú sa do rozsahu 0-5 V.
V regulačných schémach je potrebné rozsah týchto vstupov obmedziť, napr. blokom *saturation*.

Štvrtým vstupom je hodnota periódy vzorkovania, s ktorou mikroprocesor vzorkuje svetelný kanál za účelom výpočtu derivácie. Minimálna hodnota je 1 ms, pričom menšia hodnota je vždy transformovaná na jednotku. Posledný vstup reprezentuje koeficient aktuálnej vzorky diskrétného filtra 1. rádu s presnosťou na 3 desatinné miesta.



obr. 3.3 Základná vstupno-výstupná simulačná schéma

4 Problémy a ich riešenie

1) *Môže sa stať, že po pripojení meracieho systému USB káblom k PC operačný systém nedetekuje sústavu ako nový hardvér*

Riešenie:

- treba skúsiť iný USB port PC;
- prípadne odpojiť a zapojiť USB kábel znovu
- spustiť sprievodcu pridávaním nového hardvéru a dať vyhľadať nový pripojený hardvér (to, že je systém Windowsom rozpoznávaný, indikuje Windows zvukovým znamením, resp. v hardvérových zariadeniach sa vytvorí nové zariadenie *USB Serial Port*)

2) *Chybové hlásenia Matlabu týkajúce sa zvoleného portu*

-sú to napr. *'Error getting com State ...', 'Error setting com state ...', 'Serial port n does not exist ...', ...*

Riešenie:

a) skontrolovať, či operačný systém detekuje pripojenú sústavu a či je priradené správne číslo com portu v bloku komunikačného rozhrania. Vybraté číslo portu (v oboch blokoch) sa musí zhodovať s číslom com portu, ktoré je priradené *USB Serial Portu*. Toto zistíme (resp. zmeníme) v OS Windows cez *Device Managera* (*Start* → *Settings* → *Control Panel* → *System* → *Hardware* → *Device Manager*) výberom položky *Ports (COM & LPT)* → *USB Serial Port* a ďalej výberom záložky *Port Settings* → *Advanced* → *COM Port Number*

b) ak je na karte komunikačného rozhrania zvolený správny port a Matlab hlási takéto chyby, treba zvoliť nesprávny port, potvrdiť, vybrať opätovne správny port a potvrdiť (Matlab nebol korektne pripojený k portu)

c) ak zlyhá riešenie podľa b), treba odpojiť a znovu zapojiť usb kábel z/do PC; ak sa problém nevyrieši, tak odpojte a znovu zapojte zariadenie, a to okrem usb kábla aj napájací kábel (z a do elektrickej siete - spôsobí reset zariadenia)

d) niekedy v systéme Windows bežia 2 procesy *matlab.exe*, napriek len jednej spustenej aplikácii Matlabu; v takomto prípade zvykne pomôcť ukončenie jedného z nich (v *Správcovi úloh*, angl. *Task Manageri*), spravidla procesu obsadzujúceho menej pamäte

3) *Chybové hlásenia Matlabu 'Too fast for this system ...', 'Too fast sample time for this system ...'*

Riešenie:

- skontrolovať, či je dostatočný rozdiel medzi nastavenou periódou vzorkovania a timeoutom pre čítanie dát z portu (závisí od výkonnosti PC)
- zväčšiť periódou vzorkovania (parameter *Sampling time*)

4) Chybové hlásenie Matlabu 'Error: read timeout occurred ...'

Riešenie:

- treba zvýšiť čas nastavený pre timeout čítania z portu

(v prípade neúspešného čítania dát z portu sa zoberú posledné úspešne načítané data; význam zvyšovať čas nastavený ako timeout pre čítanie dát z portu má zmysel v prípade, keď je výskyt tejto chyby pomerne častý, kedy je spôsobený nízkou hodnotou timeoutu; ojedinelý vznik tejto chyby nesúvisí priamo s hodnotou parametra *Read timeout*)

5) Snímače ukazuje vysoké hodnoty (napr. o 20 – 50 stupňov navyše, alebo úplne nezmyselné)

Riešenie:

a) rovnako ako v bode 2)

b) kontakt snímača PT100 vo vnútri teplovzdušného tunelu bol (mechanicky) poškodený, vznikol zlý spoj, treba ho dobre priletovať (pri správnom zaobchádzaní so sústavou sa nestane)

6) Iné

Posledné účinné riešenie pri eventuálnych problémoch s komunikáciou (skôr ako skúsiť reštartovať PC) spočíva v 2 krokoch:

- na strane OS Windows – skontrolovať, či v systéme Windows nebežia 2 procesy *matlab.exe*, aj napriek len jednej spustenej aplikácii Matlabu; v takomto prípade zvykne pomôcť ukončenie jedného z nich (v *Správcovi úloh*, angl. *Task Manageri*), spravidla procesu obsadzujúceho menej pamäte

- na strane zariadenia - odpojenie a opätovné zapojenie zariadenia, a to buď len usb káblom od PC alebo aj odpojením a zapojením do elektrickej siete (spôsobí reset zariadenia)

5 Balík simulačných schém a riadiacich algoritmov

5.1 Príkazy *exlist* a *exnum*

Zoznam a popis simulačných schém získa užívateľ zadaním príkazu *exlist*.
(S parametrom príkazu *sk* získa popis schém v slovenskom jazyku.)

Automatické vykonávanie príslušného experimentu je možné spustiť zadaním príkazu *exnum*, po ktorom nasleduje výzva na zadanie čísla experimentu (zhodného s číslom uvedeným vo výpise príkazu *exlist*).

Príkaz *exnum* otvorí mdl súbor reprezentujúci daný experiment, vykoná inicializáciu potrebných parametrov v inicializačnom m-súbore a spustí simuláciu. Meniť inicializačné parametre je teda možné editovaním inicializačného m-súboru, ktorý je pomenovaný v tvare *init_meno_mdl_suboru.m*

Príklad: súbor experimentu: *P11_control.mdl*
inicializačný súbor: *init_P11_control.m*

Súbory simulačných experimentov sú uložené v adresári *matlabroot\udaq\udaq28LT\myschemes* a jeho podadresároch. Súbory experimentov (mdl súbory) sa po otvorení (vyvolaným napr. príkazom *exnum*) nachádzajú v prezentačnej forme, napr. hneď s otvorenými oknami priebehov veličín atď. (je potrebný minimálny počet zásahov zo strany užívateľa počas simulácie). Prezentačnú formu súborov je možné obnoviť, napríklad po neželaných zmenách v nich a nasledovnom uložení, nakoľko obsah adresára *matlabroot\udaq\udaq28LT\myschemes* je kópiou obsahu adresára *matlabroot\udaq\udaq28LT\examples*.

Pridanie svojho experimentu do zoznamu súborov spustiteľných pomocou príkazu *exnum* je možné editáciou súboru *matlabroot\udaq\udaq28LT\blks\exlist.txt* podľa syntaxe uvedenej na jeho začiatku. Podmienkou je, aby sa inicializačný m-file experimentu nachádzal v adresári *matlabroot\udaq\udaq28LT\myschemes* (alebo v jeho podadresároch).

5.2 Príkazy *mfiles*, *mdl*, *mat*, *mopen* a *copen*

Táto sada príkazov zjednodušuje a urýchľuje prácu s matlabovskými súbormi. Užívateľ nemusí zadávať meno konkrétneho súboru, len ho vyberie podľa priradenému poradového čísla.

príkaz *mfiles* - listuje m-súbory v aktuálnom adresári, po zadaní čísla súboru ho vykoná
príkaz *mdl* - listuje mdl-súbory v aktuálnom adresári, po zadaní čísla súboru ho otvorí
príkaz *mat* - listuje mat- súbory v aktuálnom adresári, po zadaní čísla súboru načíta
z neho premenné do workspaceu
príkaz *mopen*, *copen* - listuje m-súbory, resp. c a cpp-súbory v aktuálnom adresári, po
zadaní čísla súboru ho otvorí

5.3 Zoznam experimentov

Uvádzame príklad zoznamu experimentov (výstup príkazu *exlist sk*), jeho obsah závisí od konkrétnej verzie balíka:

List of experiments located in C:\MATLAB6p5\udaq\udaq28LT\myschemes

EXPERIMENT 1

Zakladna schema na manipuláciu so sustavou -
ovladanie vstupov, sledovanie vystupov

myschemes\init_udaq28LT_iov2.m

EXPERIMENT 2

Meranie prevodovej charakteristiky optickej vetvy (v 11 bodoch).

Doba simulacie: 33 s
myschemes\light\init_io_char_up_olooop.m

EXPERIMENT 3

Spojity I0 regulator - priamy kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_I0_cont.m

EXPERIMENT 4

Diskretny I0 regulator - priamy kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_I0_disc2.m

EXPERIMENT 5

Spojity PI0 regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PI0_windupless.m

EXPERIMENT 6

Diskretny PI0 regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PI0_disc_windupless.m

EXPERIMENT 7

Meranie prechodovej charakteristiky filtrovaného kanála optickej vetvy.
(Najprv sa sustava privedie do ustaleneho stavu PI regulatorom,
potom sa vstupny signal prepne na skokovy.)

Doba simulacie: 180 s
myschemes\light\init_step_respl.m

EXPERIMENT 8

Experimentálne určovanie nahradneho prenosu optickej vetvy (filtrovaneho kanala).

Postup je zalozeny na odsimulovani prechodovej charakteristiky uzivatelom vybrateho nahradneho prenosu a jej porovnanim s vykreslenym priebehom prechodovej charalteristiky ziskanej v experimente cis. 7.
Bez dat ziskanych odsimulovanim experimentu cis. 7 nefunguje !

Doba simulacie: potrebny strojovy cas PC
myschemes\light\init_ident1_aprox_sim.m

EXPERIMENT 9

Spojity PI1 regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PI1_windupless.m

EXPERIMENT 10

Diskretny PI1 regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PI1_disc_windupless.m

EXPERIMENT 11

(Kvazi) Spojity linearny PID regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PID.m

EXPERIMENT 12

Diskretny linearny PID regulator - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PID_disc.m

EXPERIMENT 13

Diskretny linearny PID regulator doplneny ARW (antiwindup) strukturou - filtrovany kanal optickej vetvy.

Doba simulacie:
myschemes\light\init_PID_arw_disc.m

EXPERIMENT 14

Meranie prevodovej charakteristiky - priamy kanal tepelnej vetvy (v 11 bodoch).

(Na dosiahnutie ustalenyx stavov je pouzity PI1_windupless regulator.)

Ak je sustava zahriata, pred experimentom je potrebne najprv ju ochladit priblizne na teplotu okolia s pouzitim vstupu ventilatora !

Doba simulacie:

myschemes\temperature\init_io_char_up_cloop.m

EXPERIMENT 15

Meranie prechodovej charakteristiky priameho kanala tepelnej vetvy.

(Najprv sa sustava privedie do ustaleneho stavu PI1_windupless regulatorom, potom sa vstupny signal prepne na skokovy.)

Ak je sustava zahriata, pred experimentom je potrebne najprv ju ochladit priblizne na teplotu okolia s pouzitim vstupu ventilatora !

Doba simulacie:

myschemes\temperature\init_step_resp2.m

EXPERIMENT 16

Experimentalne urcovanie nahradneho prenosu tepelnej vetvy (priameho kanala).

Postup je zalozeny na odsimulovani prechodovej charakteristiky uzivatelom vybrateho nahradneho prenosu a jej porovnanim s vykreslenym priebehom prechodovej charalteristiky ziskanej v experimente cis. 15. Bez dat ziskanych odsimulovanim experimentu cis. 15 nefunguje !

Doba simulacie: potrebny strojovy cas PC

myschemes\temperature\init_ident2_aprox_sim.m

EXPERIMENT 17

Spojity PI1 regulator - priamy kanal tepelnej vetvy.

Doba simulacie:

myschemes\temperature\init_PI1_windupless.m

EXPERIMENT 18

Diskretny PI1 regulator - priamy kanal tepelnej vetvy.

Doba simulacie:

myschemes\temperature\init_PI1_disc_windupless.m

EXPERIMENT 19

(Kvazi) Spojity linearny PID regulator - priamy kanal tepelnej vetvy.

Doba simulacie:

myschemes\temperature\init_PID.m

EXPERIMENT 20

Diskretny linearny PID regulator - priamy kanal tepelnej vetvy.

Doba simulacie:
myschemes\temperature\init_PID_disc.m

EXPERIMENT 21

Diskretný lineárny PID regulátor doplnený ARW (antiwindup) štruktúrou
- priamy kanál tepelnej vetvy.

Doba simulacie:
myschemes\temperature\init_PID_arw_disc.m