VIZUALIZÁCIA TECHNOLOGICKÉHO PROCESU PROSTREDNÍCTVOM DDE ROZHRANIA.

M.Foltin, T.Chvostek, M. Humaj

Katedra Automatizovaných Systémov Riadenia, Fakulta Elektrotechniky a Informatiky,

Slovenská Technická Univerzita, Ilkovičova 3, 812 19 Bratislava, Slovenská Republika

Abstrakt

V predkladanom článku sa zaoberáme vizualizáciou technologického procesu v reálnom čase. Systém pozostáva z plazmového rezacieho stroja prepojeného pomocou komunikačnej zbernice CAN s PC. Osobný počítač, ktorý je pripojený na CAN slúži na zber údajov a realizuje i samotnú vizualizáciu rezacieho stroja v prostredí MS Windows. Vizualizácia je realizovaná v spolupráci s MS Office + DDE a Matlab + VRML.

1 Úvod

V súčasnej dobe už väčšina priemyselných modulov, či zariadení využívaných v automatizácii poskytuje možnosť komunikácie po niektorej zo štandardizovaných priemyselných zberníc. Často sa využívajú zbernice ako sú napr. CAN, Profibus, Modbus a iné, ktoré poskytujú vysoký stupeň ochrany dát proti rušeniu, nízke čakacie odozvy, či umožňuje prenosy údajov na veľké vzdialenosti. Dáta pri využití niektorej z priemyselných zberníc musia byť však prenášané v presne definovanej forme, v závislosti od použitého aplikačného protokolu. Riadiaci softvér spracúvajúci údaje zo zbernice preto musí byť navrhovaný a implementovaný v závislosti od týchto pravidiel, aby dokázal využiť čo najväčší potenciál poskytovaný daným komunikačným kanálom. V prípade riadiaceho algoritmu treba spracovávať údaje rýchlo (napr. každých 5ms), s čo najmenšími prestojmi a častokrát si nemôžeme dovoliť nespracovať, či vynechať prijaté vzorky (napr. pri rezaní plechu plazmou).

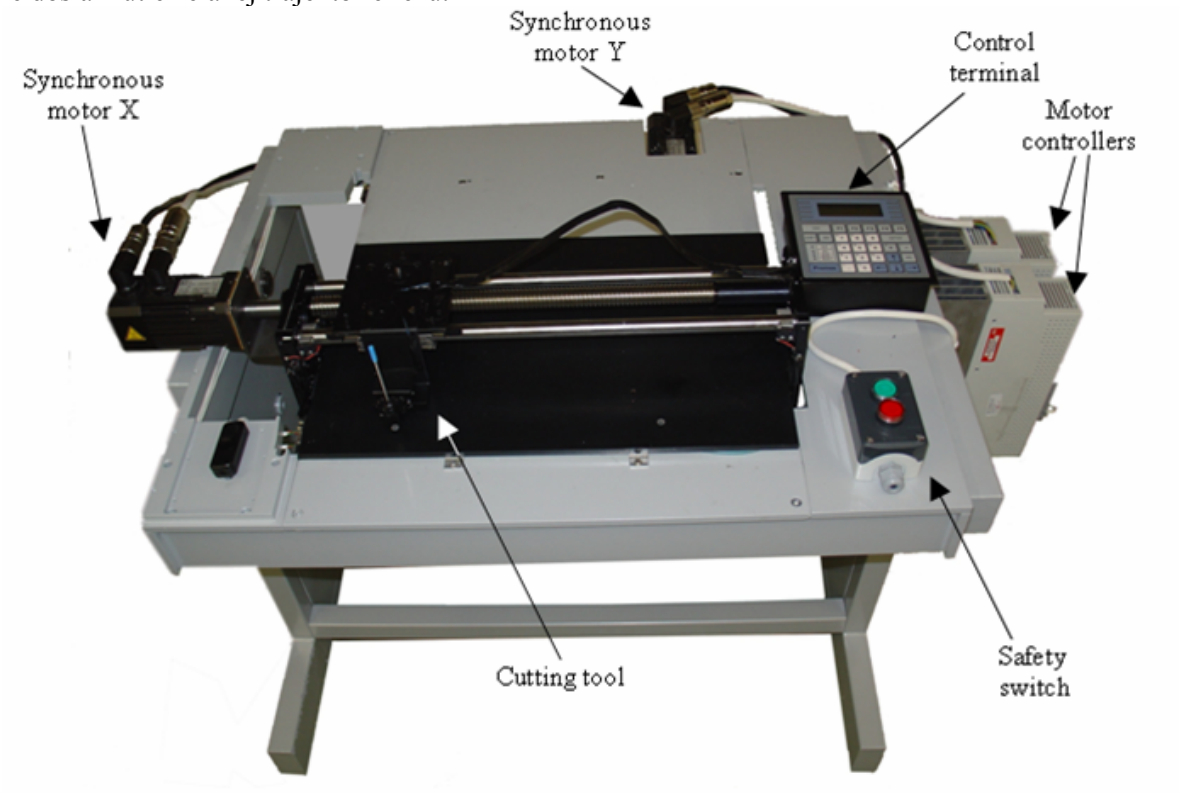
Pri použití vizualizačného softvéru, však nie sú podmienky až také prísne. Možeme pracovať s rádovo s 10 až 100násobne väčšou periódou vzorkovania, ako je perióda riadiaceho algoritmu (tj. je povolené aj niektoré vzorky vynechať), bez toho aby sa to výrazne prejavilo na výslednom grafickom zobrazení. Ako príklad môžeme uviesť to, že človeku stačí občerstviť údaje na obrazovke (napr. graf výstupu) každých 250ms a nie každú jednu milisekundu (pozn. v závislosti od dynamiky procesu), čo by kládlo zbytočne vysoké nároky na výkon hardvéru. Vizualizácia býva často realizovaná na počítačoch, či nadradených zariadeniach, ktorých výpadok zväčša nemá výrazný vplyv na celkové riadenie a slúžia len ako doplnok riadenia.

Špecifikum aplikačných protokolov priemyselných zberníc, komplikuje spracovanie údajov v už existujúcich vizualizačných nástrojoch. Údaje zo zbernice je treba najprv upraviť do formy, ktorej rozumie daná grafická aplikácia a až následne dáta vizualizovať. Na tento účel sa využívaju rozhrania ako sú napr. DDE, COM, DCOM, OPC, atď. V našej práci sme sa zaoberali prvým spomenutým rozhraním – DDE, ktorým sme vytvorili dátový most medzi reálnym zariadením a prostredím Matlab. DDE sme zvolili ako osvedčené, široko podporované rozhranie dostupnými vizualizačnými aplikáciami v oblasti riadenia (napr. InTouch) a prenos údajov cez DDE sa dá ľahko programovo implementovať. Na druhej strane nám prostredie Matlab ponúka široké spektrum možnosti grafického spracovania snímaných údajov.

Cieľom tohto projektu bolo jednoducho vytvoriť grafickú reprezentáciu reálneho zariadenia v trojrozmernom prevedení. Pre vytvorenie 3D modelu zariadenia sme použili jazyk VRML, ktorý bol pôvodne navrhnutý na prezentáciu 3D objektov na internete. Reálne zariadenie predstavoval plazmový rezací stroj komunikujúci po CAN. Na získavanie dát zo zariadenia komunikujúceho cez zbernicu CAN, bol vytvorený komunikačný modul (v jazyku *Visual Basic*) zabezpečujúci prijatie dát zo zbernice (prostredníctvom PCI CAN karty v PC) a ich predspracovanie. Následne pomocou *Virtual Reality toolboxu* (*VR toolboxu*) zahrnutého v *Simulinku* sme vytvorili previazanie 3D modelu s aktuálnymi dátami zo stroja, čím vznikla animácia kopírujúca stav reálneho zariadenia.

2 Opis technologického procesu

Technologický proces, ktorý sme použili v našom projekte tvorí zmenšený model reálneho plazmového rezacieho stroja. Tvorí ho rezací stôl o rozmeroch rezacej platne 320x240mm nad ktorou sa hýbe rezacia hlavica (Obr. 1). Zariadenie realizuje pohyb hlavice v dvoch osiach nad pracovným stolom. Pohyb je zabezpečený dvoma synchrónnymi motormi s permanentným magnetom, pripojenými k meničom od firmy Lenze. Prevod rotačného pohybu na lineárny je na báze posuvných prepravníkov so šnekovým prevodom. Hlavica obsahuje elektromagnetické prítlačné pierko, ktoré slúži ako náhrada za drahé plazmové médium, ktoré zatiaľ nie je k dispozícii. Elektrickú ochranu prekročenia pracovného priestoru zabezpečujú koncové spínače pre každú os pohybu. Na určenie nulovej pozície slúži dvojica tzv. *home* snímačov v pravom dolnom rohu pracovného priestoru. Zariadenie je (prostredníctvom meničov) prepojené s riadiacim počítačom cez zbernicu CAN. Riadiace PC, vybavené obslužnými softvérom bežiacom v prostredí *Realtime Linux.* PC generuje a posiela na základe rezacieho plánu meničom v periodickom slede (2ms) zodpovedajúce CAN správy pre dosiahnutie želanej trajektórie rezu.



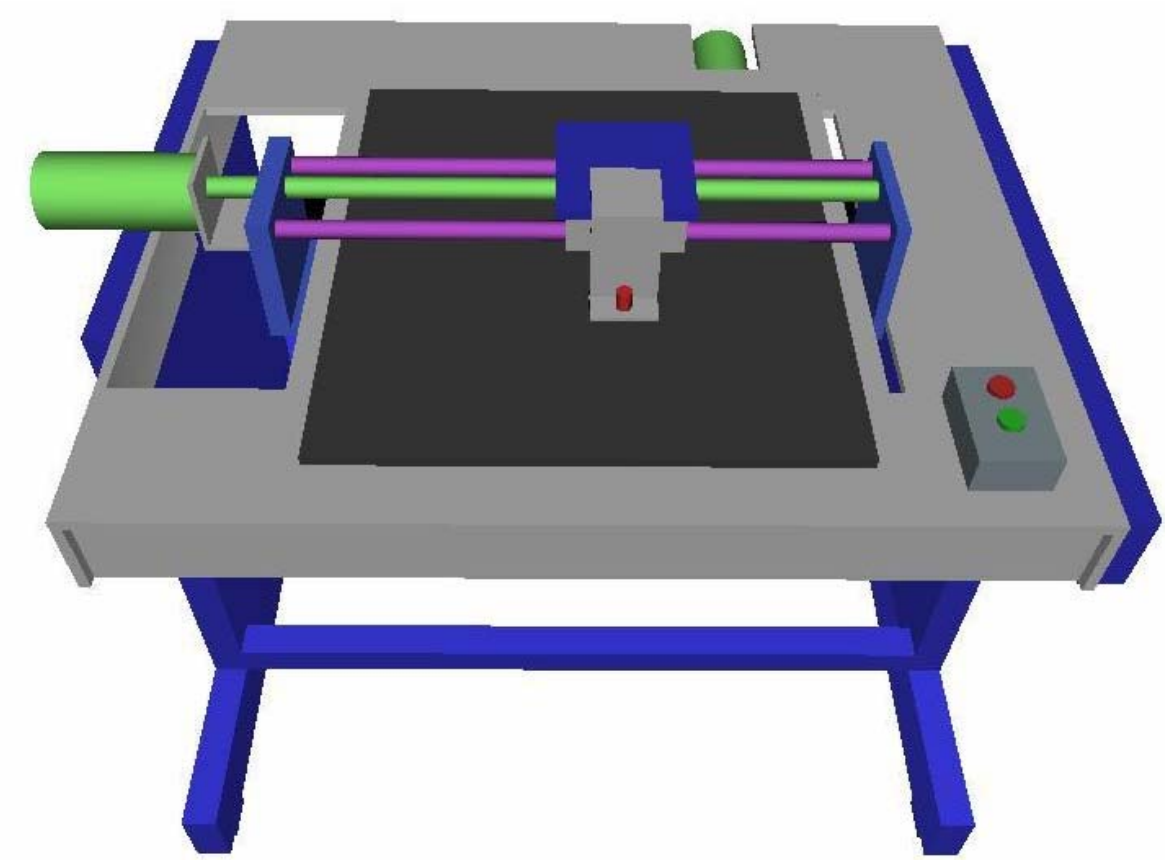
Obr. 1 Plazmový rezací stroj

3 Grafické modelovanie plazmového rezacieho stroja

Trojrozmerný model rezacieho stroja bol vytvorený za účelom verne reprezentovať správanie sa zariadenia v prostredí virtuálnej reality. Model je vytvorený v jazyku VRML, ktorý je možné prepojiť s prostredím Matlab (cez *VR toolbox*), či prezentovať na Internete. Jazyk VRML umožňuje modelovať statické objekty a zároveň ich aj animovať. Najzaujímavejšou črtou VRML je možnosť vytvárania dynamických svetov, prehrávať v nich zvuky a filmy, umožniť používateľovi interaktívne zasahovať do sveta a rozšíriť svety pomocou skriptov, či malých programov. Na vytváranie 3D modelov je možné použiť niektoré dostupné VRML editory (napr. 3D Studio Max). Nevýhodou použitia VRML editorov je, že modely po uložení obsahujú zbytočne veľa údajov (bodov) a tým kladú vyššie nároky na grafický hardvér, či prenos údajov smerom k užívateľovi. Nami navrhovaný 3D model bol vytvorený s dôrazom na potlačenie týchto negatív (Obr. 2).

Pre zobrazovanie virtuálnych svetov vo VRML je potrebné mať špeciálny prehliadač určený pre virtuálnu realitu, alebo zásuvný modul (plug-in) k bežnému prehliadaču. Pre Mozilla kompatibilné

prehliadače sa používa napríklad Cosmo player a pod Internet Explorerom napr. Cortona, či Blaxxsun. VRML súbory používajú koncovku *.wrl* (čo v angličtine znamená world, čiže svet).



Obr. 2 3D model plazmového rezacieho stroja

4 Matlab Virtual Reality Toolbox

Virtual Reality Toolbox umožňuje obojsmernú interakciu prostredia Matlab/Simulink s prostredím virtuálnej reality. Modely dynamických systémov vytvorené v Simulinku je možné realisticky vizualizovať v trojrozmernom prostredí a získať tak lepšiu priestorovú predstavu o tom ako fungujú. Behom simulácie sa užívateľ môže pohybovať vo virtuálnej scéne, pozorovať simulovaný systém z rôznych pohľadov, byť jeho súčasťou a dokonca ho z prostredia virtuálnej reality ovládať. Pri konštrukcii zariadenia je možné na virtuálnom modely overiť akým spôsobom reagujú ich súčasti (Virtual Prototyping). Virtual Reality Toolbox má i objektovo orientované programovacie rozhranie, pomocou ktorého je možné pracovať s virtuálnymi svetmi v prostredí Matlabu. Tento spôsob práce je vhodný pre vizualizáciu zložitých geometrických tvarov, simulácie fyzikálnych dejov (MKP výpočty), morphing a podobne. Virtual Reality Toolbox je jedným zo štandardných grafických výstupov pre vizualizáciu mechanických sústav modelovaných pomocou produktu SimMechanics.

Virtual Reality Toolbox obsahuje tiež bloky pre pripojenie špeciálnych periférii pre interakciu modelov v Simulinku s prostredím virtuálnej reality - Joystick, SpaceMouse a pod. Otvorený štandard VRML 97 (Virtual Reality Modelling Langauge), umožňuje jednoduchú tvorbu virtuálnych svetov a publikovanie výsledkov v sieti WWW.

5 DDE rozhranie

DDE je skratka pre *Dynamic Data Exchange*, čiže dynamickú výmenu dát. Rozhranie DDE je aplikačné programové rozhranie, ktoré umožňuje programom v systéme MS Windows navzájom komunikovať a vymieňať si navzájom informácie. DDE je založené na architektúre klient - server. Server poskytuje svoje dáta klientom, ktorý možu dáta čítať i meniť. Bežná komunikácia prebieha tak, že klient nadviaže spojenie a požiada server o poskytované údaje. DDE rozhranie sa dá dokonca

využívať i prostredníctvom lokálnej siete, kde klienti dokážu spracovávať údaje zo vzdialeného počítača, na to je však nutné použiť, konfiguračné nástroje (dodávane s MS Windows), ktoré tieto údaje zviditeľnia. Nástroje ktoré dokážu využívať rozhranie DDE sú napr. Visual C, Visual Basic, Delphi, MS Office, Matlab, InTouch, atď.

DDE komunikácia sa delí na tri typy:

- horúce spojenie - hot link (automatic) - v prípade, že sa dáta na serveri zmenia, zmena sa automaticky prenesie na klienta
- teplé spojenie - warm link (notify) - pri zmene dát sprístupňovaných na serveri je klientovi zaslaná správa o zmene a ten má možnosť rozhodnúť sa, či zmenené dáta príjme (ak áno, vyžiada si ich od servera),
- studené spojenie - cold link (manual) - 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)

Klient pri otváraní spojenia zo serverom musí zadať tri dátové položky:

- *server* - názov servera, s ktorým chce klient komunikovať, väčšinou je to názov programu (napr. Excel, alebo View), ale táto a ostatné dátové položky sú závislé na konkrétnom serveri. V prípade komunikácií cez sieť je názov servera rozšírený o názov sieťového uzla, napr. \\uzol1\Excel
- *topic* - názov predmetu komunikácie; do tejto položky sa zadáva napr. názov súboru v Exceli, ktorý obsahuje požadované dáta, ale táto hodnota je taktiež závislá na serveri, napr. pre prístup ku I/O premenným InTouchu je potrebné zadať tagname
- *item* - názov konkrétnej sprístupňovanej položky, napr. názov I/O premennej v InTouchi, alebo bunky v zošite Excelu.

Klient má ďalej možnosť odoslať zmenené dáta na server (opačný smer komunikácie - poke), či požiadať server o vykonanie nejakého makra. Toto makro je definované serverom, napr. presun kurzora v Exceli.

6 Popis realizácie DDE komunikácie medzi prostredím Matlab a MS Excel

Na zabezpečenie dynamickej výmeny dát nám poskytuje balík Matlab sadu DDE funkcií, takže priamo v Matlabe môžeme nadviazať spojenie s inou MS Windows DDE aplikáciou. V našom prípade sme potrebovali zabezpečiť výmenu dát medzi Matlabom a MS Excelom

Spojenie je realizované typu klient - server pričom Matlab sa tvári ako klient. Klient nadviaže spojenie a server - MS Excel, prijíma požiadavky od klienta. Spojenie je typu cold-link, čiže Matlab v určitej perióde žiada klienta o dáta. Pokúšali sme sa vytvoriť aj spojenie hot-link, ale vyskytli sa komplikácie s ich aktualizáciu v bežiacej simulácii v simulinku.

6.1 MS Excel ako DDE server

Jedným z hlavných bodov práce bolo vyriešiť komunikáciu medzi reálnym zariadením a jeho 3D modelom. Prenos dát medzi riadiacim počítačom a zariadením je realizovaný cez CAN zbernicu. Výhodou CAN je, že údaje zo zbernice je možne prijímať všetkými zariadeniami (po nastavení správnej masky prijímania správ), bez ovplyvnenia samotného riadenia. Počítač, na ktorom bežala vizualizácia sme pripojili na zbernicu zásuvnou kartou PCI 7841.

Samotný komunikačný modul sme riešili v prostredí jazyka *Visual Basic for Application* (VBA) podporovaného aplikáciou MS Excel. V kancelárskom balíku MS Office a aj v ďalších aplikáciach sa tento jazyk používa pre tvorbu makier. DDE server by bolo možné riešiť aj v iných vývojových prostrediach, ale zvolili sme MS Excel z dôvodu veľmi jednoduchej implementácie a grafickej prehľadnosti. Pri programovaní sme využili dodávanú knižnicu funkcií (DLL) ku CAN karte, ktorú sme importovali do prostredia VBA.

Komunikačný modul mal 4 základné funkcie, ktoré sa dali vyvolať v MS Excel:

- Inicializácia komunikácie s CAN
- Prijímanie a spracovanie dát
- Zastavenie prijímania dát
- Ukončenie komunikácie s CAN

Inicializácia nastaví komunikačne rýchlosti (500kbs), parametre prenosu, masku správ (0x7FF) pre oba porty CAN karty a tiež inicializovala samotný prenos údajov. Prijímanie a spracovanie správ prebiehalo cyklicky v nekonečnej slučke. Prijaté správy sa filtrujú a spracúvajú sa iba tie správy, ktoré nesú informáciu o rýchlosti a polohe hlavice v smere osi x a y. Dáta prijímame ako osmicu bajtov, z ktorých vyberáme požadované údaje (polohu) a po ich úprave ich zapisujeme do buniek v MS Excel. Informácie o prijatých dátach a ovládanie základných funkcii komunikačného modulu sú v prehľadne zobrazené v zošite programu MS Excel, na Obr. 3. Zapisované údaje v bunkách MS Excel sú automaticky dostupné pre DDE klienta pomocou reťazca rXcY kde X je číslo riadku a Y číslo stĺpca z ktorého čítame údaje. Pre ukončenie vizualizácie zvolíme zastavenie prijímania dát a ukončenie komunikácie s CAN.

	A	B	C	D	E
1		**Komunikacia so zbernicou CAN**			
2					
3		Init PCI-7841	Start Receiving Data		**Receive status**
4					**0**
5					
6		Done PCI-7841	Stop Receiving Data		
7					
8					
9					
10		**Prijate data**			
11					
12	Rychlost			Poloha	
13	X	Y		X [0 - 320 mm]	Y [0 - 240 mm]
14	3516.54	0.00		0.00	0.00
15	3841.00	0.00			
16					
17					
18		**Data z CAN**	**pre x**	**pre y**	
19		**Data0**	101	101	
20		**Data1**	6	6	
21		**Data2**	1	0	
22		**Data3**	15	0	
23		**Data4**	1	252	
24		**Data5**	0	255	
25		**Data6**	0	255	
26		**Data7**	0	255	
27					
28		**Pocet sprav**	-1		
29					

Obr. 3 Zošit súboru *can_dde1.4.xls* v MS Exceli

6.2 Matlab ako DDE klient

Matlab nám poskytuje sadu niekoľkých jednoduchých funkcií na komunikáciu s naším DDE serverom. Je možné použiť spojenie typu *hot-link* i *cold-link*. Prijaté údaje možeme ukladať do premenných, či využívať v simulinku (knižnica ddelib).

Použité funkcie pre DDE komunikáciu v Matlabe:

channel = ddeinit('*Excel*','can_dde1.4.xls')

Funkcia inicializujúca spojenie medzi matlabom a vybranou aplikáciou v našom prípade to je *Excel*. Ako druhý parameter sme zadali názov súboru na ktorý sme sa napojili (can_dde1.4.xls). Podotýkame, že pre úspešné napojenie je potrebné mať otvorený daný súbor v MS Exceli. Funkcia nám vráti číslo komunikačného kanálu.

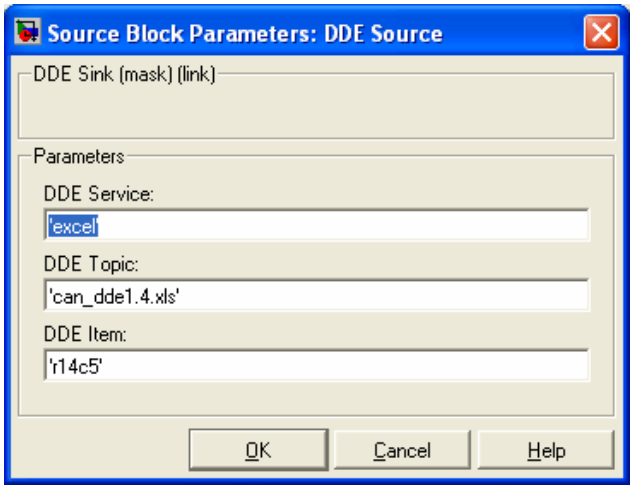
data = ddereq(channel,'r14c4')

Funkcia zabezpečujúca poslanie žiadosti o dáta cez vytvorený kanál. Druhý parameter definuje názov konkrétnej bunky z ktorej žiadame údaje (riadok 14, stĺpec 4). Zoznam parametrov, ktoré dokáže spracovať MS Excel, je možné nájsť na internete.

rc = ddeterm(channel)

Zabezpečí zrušenie nadviazaného spojenia s daným číslom kanálu a vráti nám informáciu o úspešnosti danej operácie.

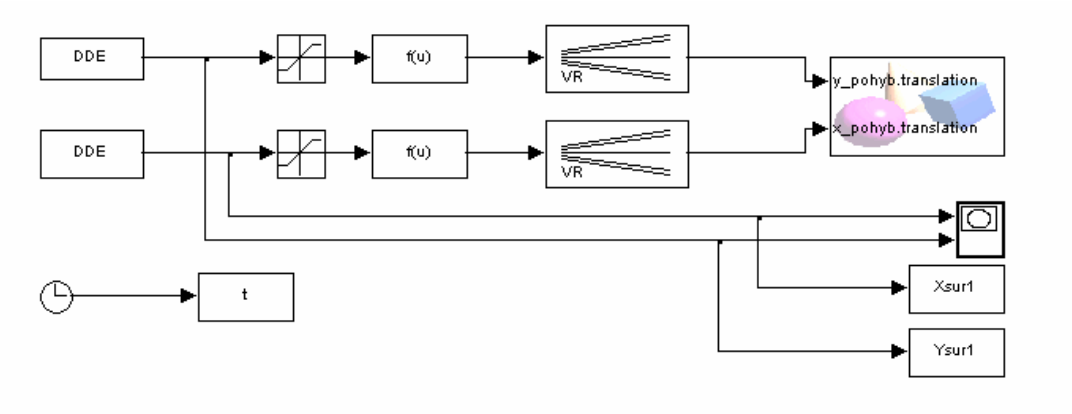
K zjednodušeniu spúšťania celej simulácie sme stiahli z internetu voľne dostupnú DDE knižnicu do Simulinku (ddelib, dostupná na stránke www.mathworks.com), ktorá bez znalosti programovania DDE pridá do Simulinku blok s názvom DDE. Po nakonfigurovaní bloku DDE (Obr. 4) je zabezpečená komunikáciu typu cold-link prostredníctvom vyššie uvedených funkcií s DDE serverom. Pri spustení simulácie sa vytvorí DDE kanál a v každom cykle pošle žiadosť o dáta. Po zastavení sa spojenie zruší.



Obr. 4 Parametre bloku DDE v simulinku

6.3 Simulačná schéma VR toolboxu

Na overenie správnej funkčnosti 3D modelu, čiže korektnosti pohybu navrhnutých grafických blokov modelu, sme vytvorili pomocnú schému v simulinku. S využitím bloku *joystick input* sme mohli 3D model ovládať pomocou zariadenia *Gamepad*, čo nám potvrdilo jeho správny návrh. Po doladení celého modelu sme vstupy z ovládača *Gamepad* nahradili blokmi DDE (Obr. 5). Jeden blok slúži pre os X, druhý, pre os Y. Údaje z týchto blokov sa ešte transformujú (prepočítajú na milimetre) na požadované údaje, ktoré nakoniec vstúpia do VR bloku. VR blok má ako jeden zo svojich parametrov názov súboru v ktorom sa nachádza VRML model. Na základe prijatých údajov animuje vo webovom prehliadači zvolený 3D VRML model rezacieho stroja.

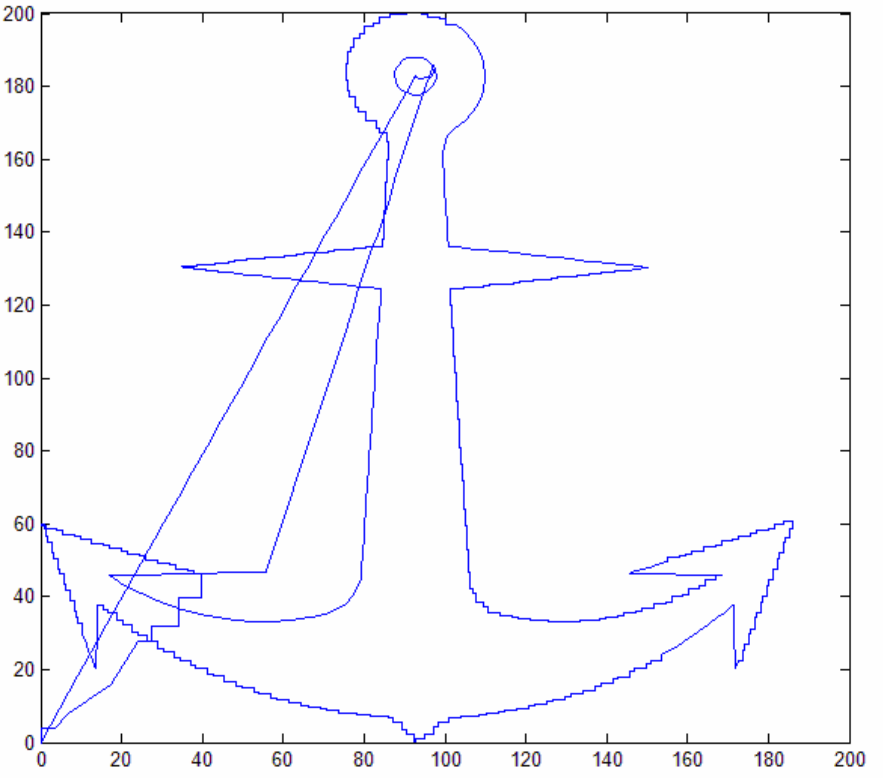


Obr. 5 Bloková schéma modelu v Simulinku

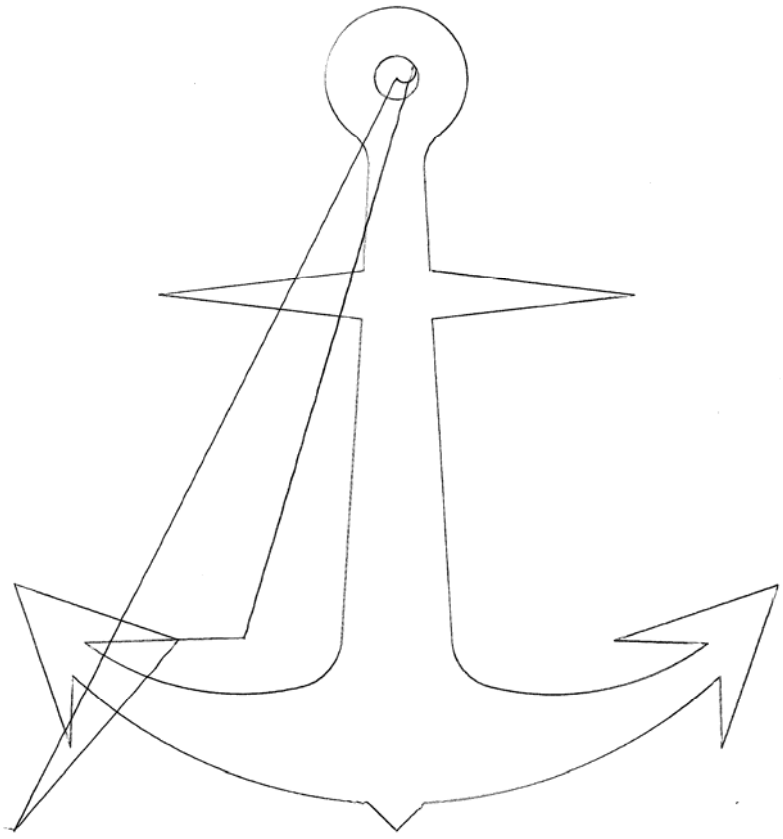
7 Dosiahnuté výsledky

Na overenie zvoleného riešenia sme priebežne realizovali niekoľko testov. Po vyriešení komunikácie medzi simulinkom a zbernicou CAN (cez DDE) sme pristúpili k porovnaniu výstupov reálneho zariadenia a nášho 3Dmodelu. Obslužnému softvéru k rezaciemu stroju sme zadali vytvoriť niekoľko výrobkov podľa zvolených rezacích plánov. Keďže v hlavici stroja nie je plazmový rezací nástroj, ale zapisovacie pierko výstup bol realizovaný ako kresba na papier. Počas "rezania" plánu sme mohli sledovať činnosť stroja prostredníctvom animácie 3D VRML modelu vo webovom prehliadači na vzdialenom počítači. Pre porovnanie s reálne nakresleným plánom sme prijaté dáta o polohe hlavice archivovali. Porovnanie reálneho výstupu a nasnímaného cez MS Excel je na obrázkoch č. 6 až 9. Na výstupoch sa nachádzajú všetky pohyby hlavice aj s jej presunom zo základnej polohy.

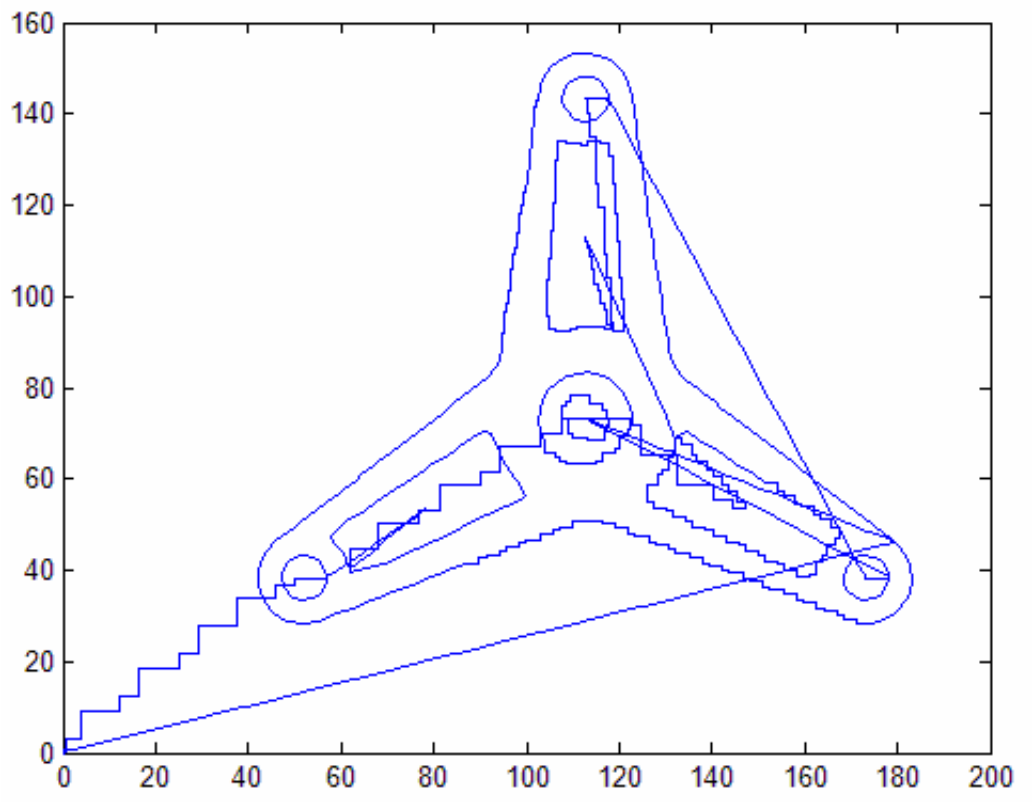
Z porovnania obrázkov (Obr. 6 až 9) je zrejmé, že naše riešenie je prijateľné a postačujúce pre sledovanie správania sa stroja počas rezania plánu. Odchýlky na výslednom obrazci oproti skutočnému výrobku, sú spôsobené vynechávaním niektorých vzoriek, čím vznikajú tzv. „zuby" na výslednom obrazci. Vo výraznej miere to spôsobujú to obmedzenia použitého návrhu v rýchlosti spracovania prijatých dát v prostredí VBA a MS Windows, ktoré sa správa nedeterministicky a má nepredvídateľné oneskorenia. Väčšie rozdiely sa vyskytujú hlavne v tých častiach procesu kde sa hlavica pohybuje veľkou rýchlosťou a tým dochádza aj k väčšej relatívnej zmene súradníc. Na sledovanie technologického procesu sú však dosiahnuté výsledky prijateľné. Nedostatky návrhu sa dajú potlačiť prísnejšou implementáciou DDE servera, napr. v prostredí Delphi, kde sa využijú presnejšie časovače a rýchlejšie sa spracujú prijímané dáta. Na úrovni MS Windows sa však stále budú vyskytovať odchýlky od pôvodného rezacieho plánu, vďaka *soft-realtime* správaniu sa operačného systému.



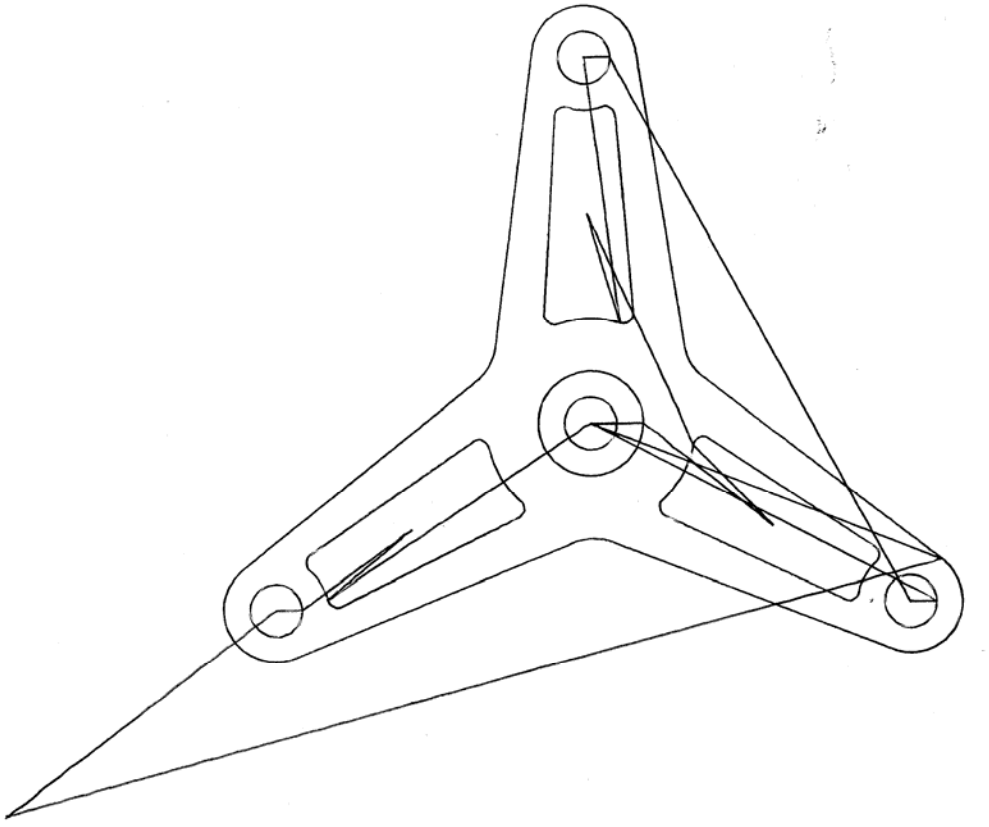
Obr. 6 Vstup 3D VRML modelu – Výrobok 1



Obr. 7 Výstup rezacieho stroja - Výrobok 1



Obr. 8 Vstup 3D VRML modelu – Výrobok 2



Obr. 9 Výstup rezacieho stroja - Výrobok 2

Literatúra

[1] D. E. Newland. *An Introduction to Random Vibrations, Spectral and Wavelet Analysis*. Longman Scientific & Technical, Essex, U.K., third edition, 1994.
[2] R. Balogh, I. Bélai, J. Dorner, P. Drahoš, Priemyselné komunikácie, STU 2001
[3] NuCom, PCI-7841 manuál, ADLINK Technology Inc., Jún 2001, www.adlink.com.tw
[4] Lenze, PDF manuály, www.lenze.com
[5] CAN Application Layer (CAL) for Industrial Applications, CiA/DS201 ... CiA/DS202, CiA/DS207, 1.5.1995
[6] CAL-based Communication Profile for Industrial Systems, Cia Draft Standart CiA/DS301, 22.9.1995
[7] CAN in Automation, CANopen/CAN dokumenty, www.can-cia.de
[8] Michael Barabanov - FSM Labs, Inc., Open RTLinux Installation Instructions, 2001 Zdroj: http://www.fsmlabs.com
[9] Matt Sherer, Writing Applications with RTLinux, Finite State Machine Labs (FSM), June 2001
Zdroj: http://www.fsmlabs.com

Martin Foltin
martin.foltin@stuba.sk

Tomáš Chvostek
tomas.chvostek@stuba.sk