

# Master-Slave riadenie robotických ramien

## Master-Slave control of robotic arms

Peter Girovský<sup>1</sup>, Radovan Sivý<sup>2</sup>

<sup>1</sup>[peter.girovsky@tuke.sk](mailto:peter.girovsky@tuke.sk), <sup>2</sup>[radovan.sivy@tuke.sk](mailto:radovan.sivy@tuke.sk)

Katedra elektrotechniky a mechatroniky, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Letná 9, 04200 Košice, Slovenská Republika

**Abstrakt**— V tomto príspevku cheme predstaviť realizáciu MASTER-SLAVE riadenia robotických ramien. Robotické ramená boli postavené pomocou servopohonov z rady Dynamixel AX-12A, ktoré sú v tomto článku stručne popísané. V rámci realizácie bolo vytvorené aj grafické užívateľské prostredie, ktoré je predstavené v závere tohto článku. Riadnie tohto robotického systému bolo riešené v programe MATLAB. V článku je taktiež popísaný spôsob komunikácie servopohonov s riadiacim programom cez sériový port počítača.

**Kľúčové slová** - riadenie, robotické rameno, servomotor, Dynamixel, GUI - grafické užívateľské prostredie

**Abstract**— In this paper we would like to present the master-slave control of the robotic arms. Robotic arms were constructed by means of actuators of a number of Dynamixel AX-12A, which in this article described briefly. During realization of this control, was also created the graphical user interface, which is presented at the end of this article. The control of the robotic system was designed in Matlab. The article also describes a method of communication with the actuator and control program through the serial port.

**Keywords** – control, robotic arm, servodrive, Dynamixel, GUI - graphical user interface

### I. ÚVOD

Robotika je dnes bežnou súčasťou nielen priemyselnej výroby, ale stáva sa aj bežnou súčasťou každodenného života. Čo sa týka využitia robotických systémov v oblasti výučby je vhodným prvkom stavebnica BIOLOID. Táto stavebnica predstavuje modulárny kit pozostávajúci z riadiacej jednotky, servopohonov Dynamixel a snímačov okolia. Z jednotlivých prvkov tejto stavebnice, je možné poskladať rôzne či už jednoduchšie alebo zložitejšie robotické systémy a vyskúšať si na nich základy riadenia takýchto systémov.

Tento príspevok sa zaoberá možnosťami využitia servopohonov Dynamixel stavebnice BIOLOID, z ktorých sú zostrojené dve robotické ramená. Riadiaci program pre takýto systém môže byť vytvorený v rôznych prostrediach a programovacích jazykoch. Pre toto riešenie sme zvolili program MATLAB, ktorý obsahuje potrebné nástroje pre vytvorenie komunikácie a grafického rozhrania [1], [2].

Samotné riadenie robotických ramien bolo realizované ako Master-Slave systém. Takéto riadenie sa zakladá na vytvorení primárnej ruky (Master), ktorou pohybuje operátor, a sekundárnej ruky (Slave), ktorá opakuje pohyby primárnej ruky v reálnom čase. Pre jednoduchšie ovládanie bolo potrebné taktiež vytvoriť grafické užívateľské prostredie, pomocou

ktorého sa dané robotické ramená budú ovládať a zároveň bude slúžiť pre zobrazenie jednotlivých stavov systému.

Hlavnou úlohou tejto práce teda bolo vytvoriť riadiaci program s užívateľským grafickým prostredím zobrazujúcim koncovú polohu bodu ramena pre riadenie robotického systému Master-Slave.

### II. SYSTÉM RIADENIA

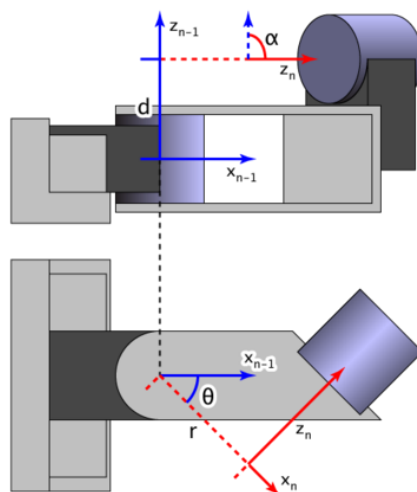
Hlavnou myšlienkou bolo vytvorenie robotického systému Master-Slave na báze servopohonov Dynamixel AX-12A. Jednou z funkcií grafického užívateľského prostredia vytvoreného pomocou GUI programu Matlab je aj zobrazovanie polohy koncového bodu ramena v súradnicovom systéme  $x, y, z$ . Výpočet polohy koncového bodu ramena je robený na základe Denavit-Hartenbergovej rotácie [3]. Podľa definície Denavit-Hartenbergovej rotácie je každý kĺb reprezentovaný svojím súradnicovým systémom a 4 parametrami, ktoré určujú akou transformáciou prejdeme k najbližšiemu segmentu tak, aby nová os  $z$  bola v osi otáčania nasledujúceho kĺbu. Pri Denavit-Hartenbergovej rotácii sú dôležité štyri základné parametre (obr. 1):

$d$  – vzdialenosť kĺbov po osi  $z$

$\Theta$  – uhol natočenia okolo osi  $z$

$r$  – dĺžka kolmice po sekundárnej osi

$\alpha$  – uhol natočenia posunutého súradnicového systému, aby os bola v osi otáčania nového kĺbu



Obr. 1. Zobrazenie parametrov Denavit-Hartenbergovej rotácie

Vynásobením čiastkových transformácií dostaneme výslednú transformačnú maticu. Čiastkové transformácie

predstavujú homogénne matice transformácie. Homogénna matica má tvar:

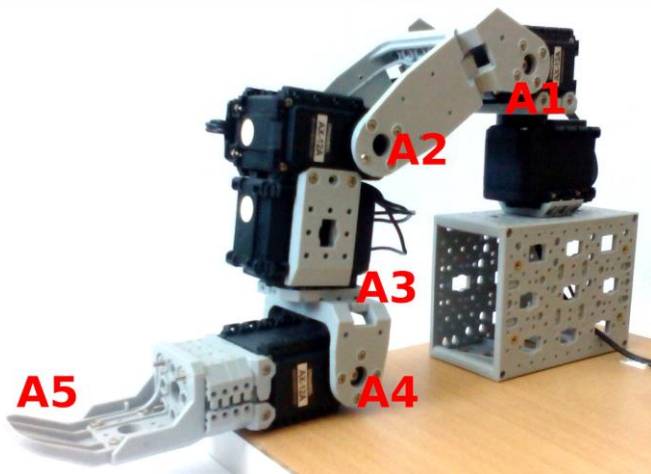
$$P = \begin{bmatrix} R & r \\ 0^T & 1 \end{bmatrix} \quad (1)$$

$R$  – je matica transformácie rotácie  
 $r$  – je matica transformácie posunutia  
 $0^T$  – je nulový vector

Podľa osi otáčania sú 3 matice transformácie rotácie:

$$\begin{aligned} Trz &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ Try &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\ Trx &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \end{aligned} \quad (2)$$

Navrhnuté rameno v našom prípade malo 5 stupňov voľnosti podľa obr. 2, na základe ktorých boli zostavené transformácie pre jednotlivé kĺby ramena podľa Denavit-Hartenbergovej rotácie.



Obr. 2. Štruktúra robotického ramena

$$\begin{aligned} A_1 &= Trz(\theta_1) \times Tpz(70) \times Tpx(25) \times Trx(-90) \\ A_2 &= Trz(\theta_2) \times Tpy(15) \times Tpx(77) \times Trx(0) \\ A_3 &= Trz(\theta_3) \times Tpz(0) \times Tpx(50) \times Try(90) \\ A_4 &= Trz(\theta_4) \times Tpz(25) \times Tpx(0) \times Try(-90) \\ A_5 &= Trz(\theta_5) \times Tpz(0) \times Tpx(110) \times Trx(90) \end{aligned} \quad (3)$$

Výslednú maticu transformácie dostaneme vynásobením jednotlivých čiastkových transformácií.

$$P = A_1 \times A_2 \times A_3 \times A_4 \times A_5 \quad (4)$$

Na základe výslednej transformácie dostávame polohu koncového bodu robotického ramena v priestore [3].

### III. MASTER-SLAVE ROBOTICKÝ SYSTÉM

Návrh Master-Slave robotického systému bol robený pre robotické ramená zostrojené zo servopohonov Dynamixel AX-

12A. Štruktúra ramena bola zvolená tak, aby čo najviac odpovedala štruktúre ľudskej ruky.

#### A. Dynamixel AX-12A

Dynamixel od firmy Robotics je inteligentný servopohon určený pre robotické aplikácie (Fig.3). Samotný modul je zložený z prevodovky, jednosmerného motora a riadiacich obvodov. Komunikácia je realizovaná rozhraním MultiDrop. Používa sa protokol TTL Half Duplex UART. MultiDrop umožňuje pripojenie viacerých modulov na jednu dátovú linku. Čo má zásadný vplyv pri tvorení komunikácie s inými platformami. Komunikácia totižto prebieha vyskladaním digitálneho paketu, ktorý obsahuje informáciu odosielateľa, prijímateľa a jednotlivé hodnoty premenných. V jednom digitálnom pakete sa môže prenášať viac informácií (poloha, rýchlosť, obmedzenie momentu atď.)

To umožňuje efektívne posielat veľké balíky dát jednotlivým modulom naraz. Riadiace obvody, okrem komunikácie, sledujú a vyhodnocujú fyzikálne veličiny [4].



Obr. 3. Dynamixel AX-12A

Pri prekročení min. a max. hodnôt veličín vynútenia vypnutie pohonu. Samotný servopohon možno riadiť s pomerne dobrou presnosťou, rozlíšenie pre rýchlosť a polohu sa definuje na 1024 krokov. Napriek svojej kompaktnej veľkosti dokáže vyvinúť krútiaci moment 1,62Nm, pri napájaní 10V. Na základe týchto vlastností bol zvolený tento typ servopohonu pre konštrukciu Master-Slave robotického system [4].

#### B. Riadenie systému

Riadiť servopohony Dynamixel AX-12A stavebnice BIOLOID pomocou PC je možné rôznymi spôsobmi. Buď sa použije pre riadenie program napísaný v prostredí, ktoré je dodávané spolu so stavebnicou, alebo sa použije program napísaný v jazyku C, prípadne C-mex pre Matlab. V našom prípade boli riadené servopohony z prostredia Matlab.

Na komunikáciu modulov Dynamixel s Matlabom slúži USB2Dynamixel prevodník od firmy ROBOTIS. Jedná sa o prevodník z USB na jeden s troch možných spôsobov komunikácie:

- TTL logika pre servomotory s 3-pinovým konektorom
- RS485 pre servomotory so 4-pinovým konektorom
- RS232 pre komunikáciu cez sériový kábel s riadiacou jednotkou, prípadne pre bezdrôtovú komunikáciu

V našom prípade sa jednalo o Half-duplexnú asynchrónnu sériovú komunikáciu medzi Matlabom a servopohonmi cez TTL linku. Samotná komunikácia riadenia so servopohonmi je realizovaná pomocou paketov, pričom Inštrukčný paket zobrazený na obr.4 je vysielaný z riadenia do modulu Dynamixel.

0xFF 0xFF ID dĺžka inštrukcia Parameter 1 ... Parameter n Kontrolný súčet

Obr. 4. Inštrukčný paket

Stavový paket zobrazený na obr.5, je odozvou na Inštrukčný paket a odosiela sa z modulu Dynamixel do riadenia.

0xFF	0xFF	ID dĺžka	chyba	Parameter 1	...	Parameter n	Kontrolný súčet
------	------	----------	-------	-------------	-----	-------------	-----------------

Obr. 5. Stavový paket

### C. Komunikácia servopohonu s Matlabom

Komunikácia so servopohonmi je realizovaná pomocou inštrukcií pre prácu so súborami ako napr. fopen, fclose, read, write a podobne. V našom prípade bola komunikácia realizovaná pomocou sériového portu, pričom bolo potrebné na začiatku definovať globálnu premennú, pomocou ktorej sme identifikovali sériový port určený na komunikáciu programu MATLAB s robotickými ramenami. Touto premennou bola premenná handles.s a nastavenie komunikačného portu je v programe definované nasledovne:

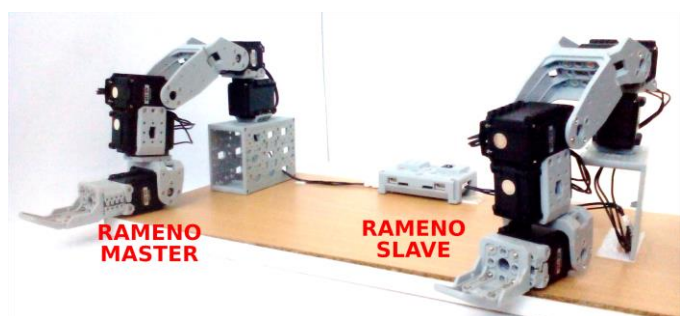
```
handles.s = serial(COM1);
set(handles.s,'BaudRate', 200000,
    'Parity', 'None', 'DataBits', 8 ,
    'StopBits', 1);
```

Takto definovaná premenná je využívaná v riadiacom programe pre komunikáciu s definovaným sériovým portom.

Funkcie fwrite a fread nám umožňujú odosielanie resp. prijímanie dát cez sériový port, pričom pod pojmom "údaje" máme na mysli inštrukčný a stavový paket pre servopohon Dynamixel. Vzhľadom k výpočtu kontrolného súčtu pre paket musí byť pôvodný 16-bitový paket prevedený na 10-bitový. Pomocou inštrukcie fwrite vyšleme inštrukčný paket cez komunikačný port. Druhá inštrukcia, fread slúži na čítanie stavového paketu. Po ukončení komunikácie sa odporúča uzavrieť otvorený komunikačný kanál, na čo slúži inštrukcia fclose. Pri viacnásobnom spustení grafického užívateľského rozhrania môže dôjsť k duplicitě sériových portov využívaných v programe MATLAB, aby sa zabránilo kolízii pri komunikácii, je vhodné na začiatku programu použiť inštrukciu fclose(instrfind). Pomocou tejto inštrukcie zatvoríme všetky, už skôr otvorené sériové porty.

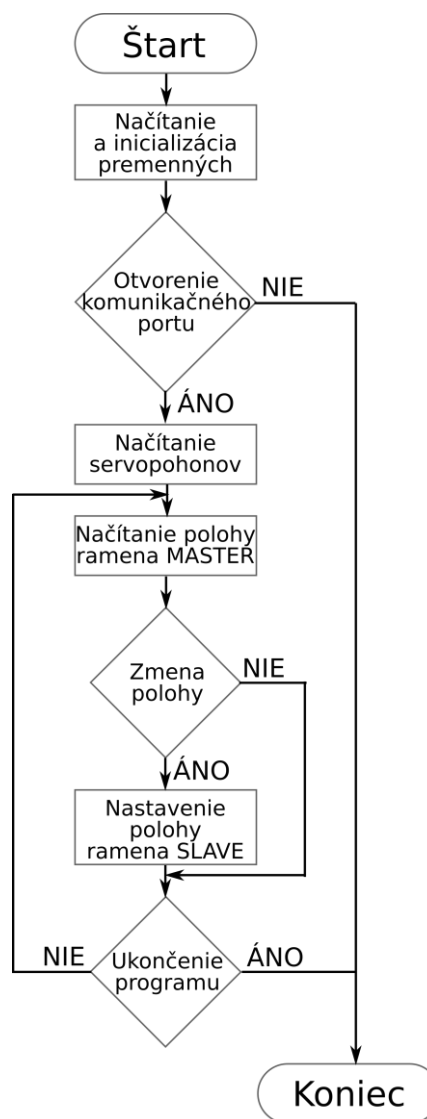
### D. Master-Slave riadenie servopohonov

Základ fungovania princípu Master-Slave riadenia je sledovanie polohy jednotlivých servomotorov ramena Master a poslať ich na rameno Slave (obr.6). Tento princíp bol realizovaný vytvorenou funkciou Real-Time, ktorá prevádzkala základnú funkciu Master-Slave riadenia a to kopírovanie v reálnom čase. Po spustení programu sa načítajú dáta z grafického prostredia a otvorí sa port pre komunikáciu medzi PC a servopohonmi. V prípade, že sa port otvoriť nepodarí vypíše sa chybové hlásenie „Chyba pri otváraní USB2Dynamixel“, uvoľní sa port a ukončí sa program.



Obr. 6. Master-Slave robotický systém

Pri úspešnom otvorení sa nastaví rýchlosť pohybu servomotorov na požadovanú hodnotu a spustí sa cyklus, ktorým sa opakuje kopírovanie polohy ramena Master, až pokiaľ nedôjde k zastaveniu funkcie. V tele tohto cyklu je aktuálna poloha načítaná do matice „PresentPos“ pomocou cyklu, kde pomocou funkcie NUM\_SERVO sa identifikuje počet servopohonov a následne sa ID čísla servopohonov ramena Master uložia do matice „idm“. Obdobne sa identifikujú servopohony ramena Slave, ktoré sú uložené v matici „ids“. Následne je vytvorený a odoslaný Syncwrite paket na adresy v matici „ids“. V prípade predchádzajúceho pohybu Master ramena sa prvým spustením programu toto rameno odblokuje. Pre zníženie zaťaženia komunikačnej linky sa kontroluje práve načítaná matica polohy ramena s predchádzajúcou, a ak sú rovnaké, paket sa neodošle. Principiálny algoritmus riešenia Master-Slave riadenia robotického ramena je zobrazený na obr.7



Obr. 7. Algoritmus Master-Slave riadenia

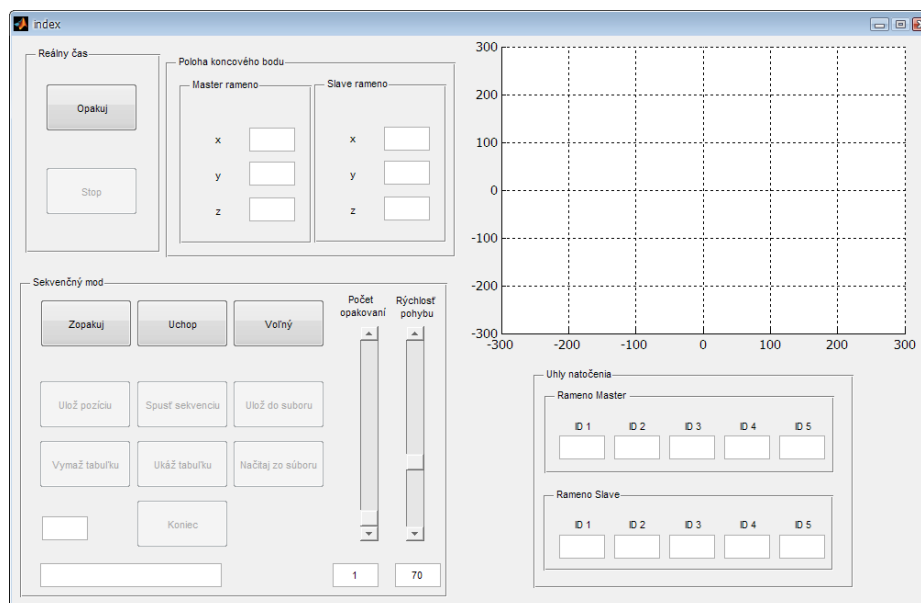
### E. GUI rozhranie pre ovládanie Master-Slave robotických ramien

Na overenie správnosti návrhu štruktúry robotického systému typu Master-Slave bolo vytvorené Grafické užívateľské rozhranie, ktorého základná obrazovka je zobrazená na obr. 8. Jeho návrh vyplýval z možnosti Real-Time riadenia,

sekvenčného riadenia a zobrazenia bodu pomocou Denavit-Hartenbergovej rotácie.

Do riešenia bolo pridané tzv. Sekvenčné riadenie. To funguje na princípe zaznamenávania sekvencie pohybov a ich

následnom opakovaní. Tým je možné tvoriť opakujúce sa pohybové cykly ramien. Pridaním sekvenčného typu riadenia robotických ramien sme chceli rozšíriť možnosti celého Master-Slave systému.



Obr. 8. Celkové GUI robotického systému

Obr. 9.

#### IV. ZÁVER

V tomto príspevku bol prezentovaný návrh Master-Slave riadenia robotických ramien zložených zo servopohonov Dynamixel AX-12A. Riadenie a grafické prostredie pre ovládanie týchto robotických ramien bolo realizované v prostredí programu Matlab. Uživatelské grafické prostredie bolo navrhnuté pre ovládanie Master-Slave riadenia v reálnom čase, ako aj na sledovanie polohy ramien, pri ktorom sa zobrazovali aktuálne uhly natočenia servomotorov a poloha koncového bodu ramena vypočítaná pomocou Denavit-Hartenbergovej notácie. Riadiaci systém, ako aj uživatelské grafické prostredie, boli rozšírené o tzv. sekvenčné riadenie z dôvodu zvýšenia možností uplatnenia robotického systému z hľadiska learningových procesov.

#### POĎAKOVANIE

Táto práca vznikla s podporou grantu VEGA 1/0464/15.

„Tento príspevok vznikol vďaka podpore v rámci operačného programu Výskum a vývoj, pre projekt: Univerzitný vedecký park TECHNICOM pre inovačné aplikácie s podporou znalostných technológií, kód ITMS: 26220220182, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.“



„Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ“

#### POUŽITÁ LITERATÚRA (POĎAKOVANIE)

- [1] SCOTT, Smith.: Matlab Advanced GUI Development.
- [2] KARBAN, Pavel.: Matlab a Simulink: Calculation and Simulation in the Programs. Brno: Computer Press vyd.2006. ISBN 80-251-1301-9 (in Czech).
- [3] The Denavit Hartenberg Convention, Robotic Institute Carnegie Mellon University. Available at: <http://pds5.egloos.com/pds/200707/30/01/denavithartenberg.pdf>
- [4] Dynamixel AX-12 Manual. Available at: [http://www.robotis.com/xe/bioloid\\_en](http://www.robotis.com/xe/bioloid_en)
- [5] SIVÝ, Radovan.: Control of Servodrives for the Robot BIOLOID, FEEaI TU Košice, 2012 (in Slovak).
- [6] Developing S-Functions. Available at: [http://www.mathworks.com/help/pdf\\_doc/simulink/sfunctions.pdf](http://www.mathworks.com/help/pdf_doc/simulink/sfunctions.pdf)
- [7] BAŠISTA, Ján.: Control of BIOLOID servodrives in MATLAB environment. Diploma thesis, FEEaI TU Košice, 2013 (in Slovak).
- [8] Bioloid. Available at: [http://www.robotis.com/xe/bioloid\\_en](http://www.robotis.com/xe/bioloid_en)
- [9] LUO, Zongwei.: Robotics, Automation, and Control in Industrial and Service Settings, IGI Global, 2015, ISBN 978-1-4666-8693-9

**P. Girovský** sa narodil v roku 1979 vo Vranove nad Topľou. V roku 2003 získal titul Ing. na Fakulte elektrotechniky a informatiky TU v Košiciach v odbore Silnoprúdové inžinierstvo. V roku 2009 získal titul PhD. na Fakulte elektrotechniky a informatiky TU v Košiciach v odbore Silnoprúdové inžinierstvo. V súčasnosti pracuje ako odborný asistent na Katedre elektrotechniky a mechatroniky FEI TU v Košiciach.

**R. Sivý** sa narodil v roku 1988 v Humennom. V roku 2012 získal titul Ing. na Fakulte elektrotechniky a informatiky TU v Košiciach. V roku 2016 získal titul PhD. na Fakulte elektrotechniky a informatiky TU v Košiciach. Oblasťami jeho záujmu sú robotika a autonómne vozidlá.