

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**RIADIACI SYSTÉM S INVERZNOU KINEMATIKOU PRE  
MOBILNÉ ROBOTICKÉ RAMENO**

24038ac9-cead-47c8-b9af-a7aa69d643d0

2012

Peter Pukančík

**UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY**

**RIADIACI SYSTÉM S INVERZNOU KINEMATIKOU PRE  
MOBILNÉ ROBOTICKÉ RAMENO**

Bakalárska práca

Študijný program: Aplikovaná informatika  
Študijný odbor: 9.2.9 Aplikovaná informatika  
Školiace pracovisko: Katedra aplikovanej informatiky  
Školiteľ: Mgr. Pavel Petrovič PhD.

**Bratislava 2012**

**Peter Pukančík**



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Peter Pukančík  
**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)  
**Študijný odbor:** 9.2.9. aplikovaná informatika  
**Typ záverečnej práce:** bakalárska  
**Jazyk záverečnej práce:** slovenský

**Názov:** Riadiaci systém s inverznou kinematikou pre mobilné robotické rameno  
**Cieľ:** Cieľom je zostrojiť mobilného robota s ramenom Lynxmotion a vytvoriť pre neho kompletný riadiaci systém. Implementácia algoritmu inverznej kinematiky, v prípade možnosti využitia videnia alebo stereovidenia, prípadne iných senzorov. Demonštrácia funkčnosti hotového systému na jednoduchom príklade.  
**Literatúra:** Eduardo Bayro Corrochano: Geometric Computing for Perception Action Systems, Springer 2001.


**Kľúčové slová:** robotika, inverzná kinematika, mobilný robot


**Vedúci:** Mgr. Pavel Petrovič, PhD.

**Dátum zadania:** 17.10.2011

**Dátum schválenia:** 02.11.2011

  
doc. RNDr. Mária Markošová, PhD.  
garant študijného programu

  
.....  
študent

  
.....  
vedúci

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím uvedených zdrojov a literatúry

Bratislava 1.6.2012

.....  
Peter Pukančík

### **Pod'akovanie**

Veľká vďaka patrí vedúcemu mojej bakalárskej práce Mgr. Pavlovi Petrovičovi PhD., za aktívny prístup, hodiny strávené konzultáciami a pomocou pri procese tvorby tejto bakalárskej práce

# Abstrakt

Pukančík, Peter. Riadiaci systém s inverznou kinematikou pre mobilné robotické rameno. Bakalárska práca. Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky. Katedra aplikovanej informatiky. Školiteľ: Mgr. Pavel Petrovič, PhD.

Táto práca mapuje proces tvorby mobilného robotického systému od návrhu, cez konštrukciu až po implementáciu obslužnej funkcionality a kinematiky. Najskôr predstaví úvod do problematiky robotických systémov a prístupov k riešeniu problémov inverznej kinematiky. Potom prejde od špecifikácie až po návrh robotického systému a nakoniec ukáže praktické implementovanie doprednej a inverznej kinematiky pre robotické rameno.

Práca predstavuje ľahké nazretie do čoraz populárnejšej témy, robotiky.

## **Kľúčové slová:**

robotické rameno, inverzná kinematika, mobilný robot, kinematika, robotika, manipulátor, DOF, Denavit-Hartenberg

# Abstract

Pukančík, Peter. Control system with inverse kinematics for mobile robotic arm manipulator. Bachelor thesis. Comenius University in Bratislava. Faculty of mathematics, physics and informatics. Department of applied informatics. Bachelor thesis supervisor: Mgr. Pavel Petrovič, PhD.

This bachelor thesis maps the process of creating mobile robotic system from the designing phase, through constructing phase to implementation of control functions and kinematics. It starts with introduction to robotics systems issues and possible ways to solve inverse kinematics. Then it deals with specifications and design of robotic system and it shows practical implementation of forward and inverse kinematics for robotic arm at last.

This thesis presents simple introduction to more and more popular topic, robotics.

## **Kľúčové slová:**

robotic arm, inverse kinematics, mobile robot, kinematics, robotics, manipulator, DOF, Denavit-Hartenberg

# Obsah



# Úvod

Robotika je jedna z prudko sa rozvíjajúcich technologických oblastí. To čo ešte pred pár rokmi bolo súčasťou sci-fi filmov sa teraz objavuje čoraz častejšie vo vývojových laboratóriách a častokrát to z nich aj rýchlo vylezie. Hovoriaca robotická hlava skoro na nerozoznanie od ľudskej, dvojnohý robot s takmer dokonalou stabilitou pri chôdzi či android kompletne kopírujúci mimiku a pohyby ľudského tela, to sú všetko fascinujúce pohľady do prítomnosti a nie budúcnosti.

A keďže človek je od prírody zvedavý tvor, nedal som si ujsť možnosť trochu nazrieť pod pokrievku tejto úžasnej vednej disciplíny. Motiváciou do témy mi bola paradoxne práve skôr nulová znalosť problematiky robotiky aj kinematiky, ale aj úprimný záujem uchopiť nové poznatky z tohto odboru. Človek sa však najrýchlejšie naučí na praktických príkladoch a preto som si vybral túto tému ako bakalársku prácu.

Problematika tejto práce môže byť pre ľudí znalých odboru príliš jednoduchá, nakoľko však opisuje ešte len prvé kroky takého robotického dieťaťa na tomto svete, nemá zmysel zatiaľ riešiť rôzne zložité teoretické problémy. Práca s reálnymi mechanizmami a súčiastkami je o niečom inom ako teoretické či virtuálne systémy. Aj tento robot má pred sebou veľké plány do budúcnosti, ale na všetko treba ísť postupne.

Aj keď cieľom tejto práce nieje prednáška o elektrotechnike, ale o praktickej aplikácii, pokúsim sa aspoň z praktického hľadiska vysvetliť niektoré podstatnejšie témy dotýkajúce sa oboru robotických systémov či elektroniky. Následne si rozoberieme problém kinematiky a rôzne prístupy k výpočtu inverznej kinematiky. S teoretickými vedomosťami sa pustíme do cieľa tejto práce. Prednesieme špecifikácie požiadaviek na náš nový robotický systém, navrhne ich implementáciu a pokúsime sa zostrojiť tento robotický systém obohatený o inverznú kinematiku robotického ramena.

Práca je štruktúrovaná do 4 častí. V prvej spravíme prehľad východísk k riešeniu nášho cieľa práce. V druhej časti prednesieme požiadavky na systém, vzhľadom na ciele práce. V tretej časti spravíme návrh na základe špecifikácií a v poslednej štvrtej časti opíšeme výsledky našej práce a úspešnosť dosiahnutia cieľov.

# Kapitola 1

## VÝCHODISKÁ

### 1.1 Ciele práce

Cieľom práce je navrhnuť a skonštruovať mobilnú robotickú platformu s robotickým ramenom. Pre obe samostatné časti robota naprogramovať ovládacie prvky a funkcie a navrhnuť komunikačné rozhranie medzi robotom a riadiacim systémom. Nakonfigurovať riadiaci systém a v ňom implementovať funkcionalitu doprednej a inverznej kinematiky robotického ramena s použitím štandardov pre robotiku. Následne na jednoduchých príkladoch demonštrovať funkčnosť ovládania robota a využitia jeho autonómnych procesov, konkrétne výpočet a použitie inverznej kinematiky.

### 1.2 Robotika

Robot je elektromechanické alebo biomechanické zariadenie, ktoré vykonáva autonómne alebo predprogramované úlohy, prípadne slúži ako manipulačný nástroj plne ovládaný človekom.

Robotika je vedecký a technický odbor zaoberajúci sa dizajnom, štrukturálnou kompozíciou, konštrukciou, výrobou, funkcionalitou a využitím robotov. Taktiež sa zaoberá aj zariadeniami a systémami, ktoré robota ovládajú a manipulujú, informáciami zo spätnej väzby senzorov a ich spracovaním.

Pôvod slova *robot* sa datuje k roku 1920 a prvý krát ho použil spisovateľ Karel Čapek vo svojej hre “R.U.R“, kde týmto slovom označoval pracovníkov, syntetických umelých ľudí, vykonávajúcich robotu namiesto ľudí. Pojem *robotika* použil ako prvý sci-fi spisovateľ Isaac Asimov v diele “Runaround“ v roku 1941 a spolu s ním kodifikoval aj známe základne tri zákony robotiky. Už od čias antického Grécka bolo skonštruovaných mnoho mechanických strojov významovo podobným pojmu robot, no éra prvých moderných robotov, tak ako ich chápeme dnes, sa začína v 50tych rokoch 20. storočia, zostrojením robota “Unimate“

vynálezcom Georgom Devolom.

Potreba robotov a ich prudký rozvoj je spôsobený ich úlohou v spoločnosti, kde nachádzajú využitie hlavne v oblastiach, ktoré sú buď nebezpečné pre zdravie človeka, alebo slúžia ako lacná nevyčerpatel'ná pracovná sila.

V robotike sa spájajú mnohé vedné disciplíny, ktoré zahŕňajú rôzne oblasti súvisiace s robotmi:

- **Mechanické inžinierstvo** – napr. štruktúra robota, konkrétne pri mobilných robotoch pohybový mechanizmus, ...
- **Informatika** – napr. reprezentácia dát, spracovanie dát, plánovanie, výpočty, vykonávanie algoritmov, ...
- **Elektrotechnické inžinierstvo** – napr. obvody, integrácia jednotlivých podsystémov, senzory, komunikácia, ...
- **Kognitívna psychológia a neuroveda** – napr. kopírovanie správania sa živých organizmov pri riešení žiadaných podobných úkonov, ...

Existuje veľké množstvo atribútov, podľa ktorých by sme mohli deliť robotov, pozrime sa na tie, ktoré sú podstatné pre nášho konkrétneho robota.

Mobilnosť:

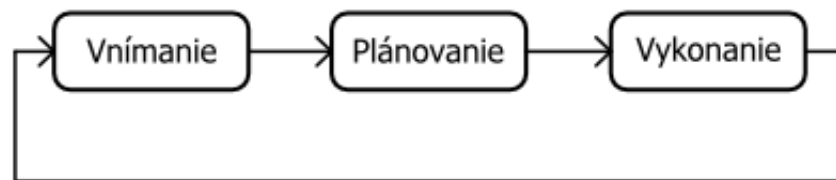
- **mobilné roboty** – môžu sa hýbať v priestore, ich priestorový operačný modus závisí od konštrukcie pohybového mechanizmu
- **stacionárne roboty** – majú fixnú nehybnú základňu, ktorá nemení svoju polohu v priestore, jednotlivé časti robota môžu byť mobilné, napr. robotické rameno

Autonómnosť:

- **priame riadenie** – resp. tele-operatívny, operátor má neustále kontrolu nad činnosťou robota, pracuje s príkazmi na najnižších úrovniach, robot má minimálnu alebo žiadnu možnosť ovplyvniť výsledný efekt
- **polo-autonómny** – resp. tele-robotický, prítomnosť operátora je stále nutná po celý čas, ale s robotom komunikuje na vyššej úrovni, robot vie zadané údaje interpretovať,

vyhodnotiť na základe predprogramovaných modelov a prípadne upraviť napr. na základe údajov zo senzorov

- **plne-autonómny** – prítomnosť operátora nieje potrebná po celý čas, robot má prípadne potrebné údaje na chod systému zo senzorov, avšak pracuje len na základe predprogramovaných vzorcov správania sa, prípadne učiacich sa metód
- **umelá inteligencia** – zatiaľ len teoretický model, robot má plne samostatné možnosti sa svojvoľne učiť a následne vykonávať činnosť



Obr. 1.1 Riadiaci cyklus robota

Riadiaci cyklus sa preto dá pre robota nášho typu abstrahovať do cyklu troch krokov:

- **Vnímanie** – získanie informácií z externého zdroja, senzorov, vrátane komunikácie s operátorom
- **Plánovanie** – plánovanie úloh, interpretovanie a spracovanie informácií, učenie, výpočet, plánovanie akcie, napr. pohyb motorov
- **Vykonanie** – vykonanie naplánovaných akcií, výstup informácií, napr. spätná komunikácia s operátorom

A nakoniec ešte rozdelenie robota na jednotlivé komponenty:

- **Zdroj (power source)** – batérie, ...
- **Mechanické časti** – konštrukcia, fyzické elementy štruktúry robota, napr. pre manipulátor (rameno) sú to kĺby (joints), spoje (links),..., alebo pre mobilného robota karoséria(chassis),...
- **Pohonné zariadenia (actuators)** – motory, servomotory, ...
- **Efektor (end effector, gripper)** – komponent, ktorý sa nachádza na konci manipulátoru, slúži ako interaktívny člen robota s okolím

- **Senzory (sensory system)** – získavajú informácie o prostredí v ktorom sa nachádza robot a aj interné informácie o stave jednotlivých komponentov robota
- **Radiče a modulátory (controllers, encoders)** – ovládajú a spravujú pohyb pohonných zariadení
- **Riadiace prvky** – zabezpečujú časti plánovanie a vykonávanie z riadiaceho cyklu, komunikáciu, obsluhu robota

### 1.2.1 Mobilné roboty

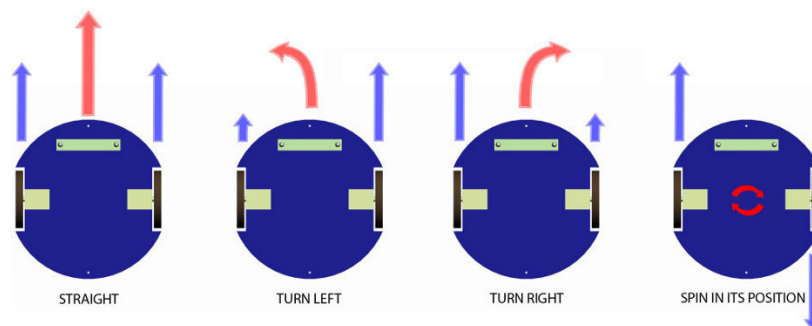
Mobilné roboty sú roboty schopné pohybu v určenom prostredí, čiže nemajú fixnú základňu ako robotické rameno.

Podľa určeného priestoru mobility poznáme:

- **pozemné (terrestrial)** – roboty pohybujúce sa po pevnom povrchu, potrebná je aj gravitácia
- **vodné (aquatic)** – operujúce pod aj nad hladinou vody
- **lietajúce (aerial)** – lietajúce stroje, v našej atmosfére rátajúce s gravitáciou, odporom vzduchu, ...
- **vesmírne (space)** – operujúce v prostredí bez gravitácie, môžu byť lietajúce alebo je ich pohyb odkázaný na dotyk či uchopenie s pevnou plochou

Pri pozemných robotoch záleží od želaného rádiusu využitia aký zvolíme mechanizmus pohybovej sústavy. Môže ísť o nohy, kolečká, pásy,... Každé má svoje pre a proti vzhľadom na pracovné prostredie robota.

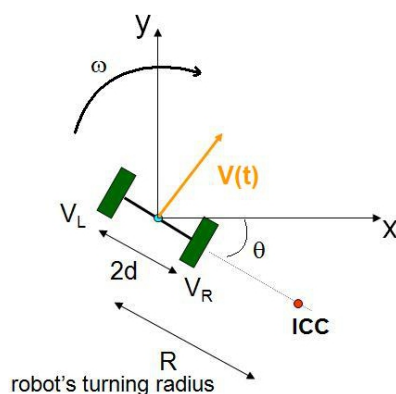
Zamerajme sa na kolečkové roboty. Spôsob uchytienia kolečiek na konštrukciu determinuje spôsob pohybu robota. Ak sú kolečká schopné natočenia voči konštrukcii robota, využíva sa tento princíp na zatáčanie, napr. v dnešných autách nájdeme implementované Ackermannovo riadenie, kde sa predné kolesá auta natáčajú nezávisle na sebe voči osi kolmej k rovine podvozku. Ak sú však kolečká pevne orientované voči konštrukcii, robot sa otáča tzv. šmykom. V takom prípade hovoríme o diferenčnom riadení.



**Obr. 1.2** Ukážka zatáčania robota s diferenčným riadením [15]

Pri diferenčnom riadení sa už nezaobráame iba kinematikou, čiže geometrickým vyjadrením pohybu, ale aj dynamikou, čiže silami pôsobiacimi pri pohybe. V tomto prípade rýchlosť otáčania kolies či zrýchlenie.

Diferenčné riadenie je založené na nezávislom otáčaní kolečiek (resp. celých náprav) na oboch stranách robota, ukážka je znázornená na obrázku 1.2. Pri rozdielnych rýchlostiach otáčania sa kolečiek na protiľahlých stranách robota sa mení bod otáčania ICC (Instantaneous Center of Curvature) a tým sa mení aj polomer  $R$  otáčania sa celého robota, čo je vzdialenosť medzi referenčným bodom robota (stred medzi kolečkami, resp. nápravami) a ICC. Pozri obrázok 1.3.



**Obr. 1.3** Geometrické znázornenie princípu diferenčného riadenia [15]

Pre diferenčnom riadení sa dá rôznymi prístupmi [1][10] vypočítať aktuálna pozícia robota doprednou kinematikou, alebo naplánovať cesta inverznou kinematikou. Nevýhodou diferenčného riadenia je, že aj malá nepresnosť veľkostí rýchlosti rezultuje vo významnú odchýlku v trajektórii robota. Navyše mimo laboratórneho prostredia je, vzhľadom na

nepresnosti v teréne, takmer nemožné presne určiť pozíciu len vďaka odometrii a výpočtom doprednej kinematiky. Preto je externý zdroj pozície robota (GPS, akcelerometer,...) potrebný na priebežné korekcie výpočtov.

## 1.2.2 Robotické rameno

Najbežnejším typom manipulátora je robotické rameno. Pozostáva zo súboru pevných častí (rigid links) spojených kĺbmi (joints), ktoré spolu tvoria tzv. kinematické zreťazenie, väčšinou za sebou jednoducho sériovo zapojené. Na jednom konci sa nachádza efektor (end effector), ktorý slúži na manipuláciu s objektami a na druhom konci je pevne spojené s telom nosného robota, alebo na fixnú základňu. Robotické rameno je typ stacionárneho, nemobilného robota.



Obr 1.4 Znárodné DOF na konkrétnych príkladoch robotického ramena a ľudskej ruky [16]

Kĺby robotického ramena ponúkajú, konkrétne na základe danej skonštruovanej štruktúry a mechanizmu spojenia, tzv. **stupne voľnosti – DOF (Degrees Of Freedom)**. V 3-D priestore poznáme 6 stupňov voľnosti, 3 posuvné a 3 rotačné okolo bázových osí priestoru. Pre jednoduchosť priradíme každému kĺbu robota iba jeden DOF, pričom zložitejšie kĺby simulujeme kombináciou viacerých jednorozmerných DOF kĺbov.

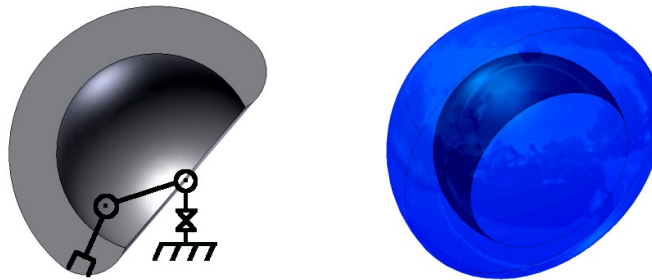
Stupne voľnosti pre jednoduchých robotov teda delíme na dva základné:

- **rotačný (rotational)** – umožňuje rotáciu medzi dvoma pevnými spojeniami v kĺbe
- **posuvný (prismatic)** – spôsobuje posun pevných spojení v kĺbe

Na obrázku 1.4 sú znázornené dva príklady ako nájsť stupne voľnosti, na modeli robota či ľudskej ruky. Ide o každé miesto kde dochádza k jednej zo 6 možných zmien, posunu, ohnutiu či otočeniu. Na robotovi sa dajú ľahko identifikovať aj vyhľadáním jednotlivých pohonných

zariadení. V reálnom svete majú jednotlivé DOF ešte aj vlastné obmedzenia (constraints), napr. servomotory zväčša nepresahujú rozsah otočenia 200 stupňov a pod.

Odporúča sa udržať počet DOF na robotickom ramene v čo najmenšom možnom počte, nakoľko každý ďalší DOF komplikuje následné výpočty kinematiky (1.2.4). Každý DOF je počiatočným bodom nového súradnicového systému, pričom vzťahy medzi jednotlivými súradnicovými systémami sú v robotike štandardizované Denavit-Hartenberg konvenciou (1.2.4.1.1).



**Obr 1.5** Pracovný priestor robotického ramena, vľavo v rovine, vpravo v priestore [16]

Dĺžky jednotlivých spojení, typ kĺbov a obmedzenia pre jednotlivé pohonné zariadenia, DOF, určujú dosah a manipulačný pracovný priestor robotického ramena. Grafický príklad pracovného priestoru robotického ramena je na obrázku 1.5.

Schopnosť manipulácie a rozsah interakcie s prostredím sú determinované aj typom efektora:

- **chytač (gripper)** – najjednoduchší typ efektora, ide zväčša o 2 prsty, ktoré vedú zmenou rozsahu zovretia uchopiť objekt
- **nástroj (tool)** – využíva sa hlavne v priemysle a medicíne, na pozícii efektora má daný typ nástroja, napr. šraubovák, striekacie zariadenia na farby či chirurgické nástroje
- **samostatný robot** – pokiaľ je efektor zložitým zariadením, napr. kopíruje komplexnú ľudskú ruku či má niekoľko DOF, tak sa efektor považuje za samostatného robota

V prípade prvých dvoch možností sa efektor nepovažuje za DOF na robotickom ramene, v prípade zložitého zariadenia sa vyhodnocuje samostatne a tiež nepatrí medzi DOF ramena.



### 1.2.3 Kinematika robotického ramena

Zatiaľ sme sa pozreli na robotické rameno len z tej mechanickej časti. Vieme z akých mechanických častí pozostáva a vo veľmi primitívnej forme by sme s ním dokonca mohli aj mechanicky pracovať priamym riadením. Na to, aby robot dostal aspoň základný rozmer autonómnosti, musíme ho vedieť fyzicky uchopiť a reprezentovať v priestore a v návaznosti na jeho okolie. Robotické rameno je popísané ako kinematické zreťazenie mechanických súčastí. Vystupuje teda v dvoch rovinách. Každý kĺb (joint resp. DOF) je samostatná časť, preto mu môžeme priradiť vlastný súradnicový systém v priestore a zároveň si ho určíme ako počiatočný bod tohto nového súradnicového systému. Zároveň je to ale zreťazenie, preto je potrebné poznať v akom vzťahu sú tieto jednotlivé časti a ich súradnicové systémy. Toto nám ešte zabezpečuje lineárna algebra a geometria.

Kinematika je odbor klasickej mechaniky a využijeme ju v ďalšom postupnom kroku v rozšírení funkcionality robotickej ruky a to pohybom. **Kinematika** popisuje z geometrického hľadiska pohyb bodov, telies či celých systémov, študuje jednotlivé trajektórie pohybov. Nerieši sily, ktoré vyvolali tento pohyb, to študuje dynamika.

Pohyb jednotlivých častí robota popíšeme ako vzájomný pohyb ich tzv. **referenčných rámcov** (3.2.4.1), čo sú tie priradené vlastné **súradnicové systémy**. Ich orientácia v priestore by nemala byť v robotike ľubovoľná, ale je určená štandardom (1.2.4.1.1).

#### 1.2.3.1 Súradnicové systémy, Transformácie

Zmeny medzi jednotlivými súradnicovými systémami budeme popisovať transformačnými maticami. Aby sme mohli používať uniformné matice aj pre rotáciu (lineárne) ale aj pre posun (afínne), prejdeme na **homogénnu reprezentáciu** súradníc. Pridáme jeden rozmer, čiže teraz budeme mať bod  $P = (x,y,z)$  v 3-D priestore popísaný ako  $P = (x, y, z, 1)^T$  a matice budú rozmeru 4x4. [9]

V 3-D priestore potrebujeme na jednoznačnú reprezentáciu objektu, teda aj časti robota (rigid link), až 6 veličín, 3 rotačné a 3 pozičné. Teda máme 6 transformačných matíc [9]. Matice môžeme násobiť, čo odpovedá skladaniu (lineárnych) zobrazení. Násobenie nieje komutatívne a násobí sa postupne sprava.

### 1.2.3.1.1 Denavit-Hartenberg parametre

**Konvencia Denavit-Hartenberg** je štandard pridelovania referenčných rámcov pre kinematické zreťazenie robotickej ruky v priestore. Výhodou je, že redukuje počet parametrov zo šiestych na 4. [6] Algoritmom [6][7][8] sa pridelí počiatočný bod a orientácia referenčných rámcov pre jednotlivé časti a ako výsledok pre každú vzájomnú susednú transformáciu získame 4 parametre, ktoré popisujú transformáciu návazných referenčných rámcov okolo 2 osí,  $z$  a  $x$ .

Tieto 4 **parametre** popisujú vzťah medzi  $z_{i-1}$  a  $z_i$  a  $x_{i-1}$  a  $x_i$ , kde referenčný rámec  $i$  nadväzuje vporadí na rámec  $i-1$ . Pre prehľadnosť sa uvádzajú v tabuľke.

- $a_i$  – vzdialenosť  $z_{i-1}$  k  $z_i$  na osi  $x_{i-1}$
- $\alpha_i$  – uhol medzi  $z_{i-1}$  a  $z_i$  okolo osi  $x_{i-1}$
- $d_i$  – vzdialenosť od  $x_{i-1}$  k  $x_i$  na osi  $z_i$
- $\theta_i$  – uhol medzi  $x_{i-1}$  a  $x_i$  okolo osi  $z_i$

Pozn. Parametre sa môžu mýliť, prvý a tretí sú latinské písmená **a** a **d**, druhý je grécke písmeno alfa

V referenčnom ráme, vytvorenom podľa D-H algoritmu, označuje os  $z$  smer kĺbu (DOF). Posuvný kĺb teda ovplyvňuje parameter **d** a otočný kĺb mení parameter **θ** (theta). Parametre **a** a **α** (alfa) sú konštanty.

$$\begin{aligned}
 A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} & (3.10) \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

**Obr 1.6** Denavit-Hartenberg transformácia referenčných rámcov [6]

Transformácia medzi 2 referenčnými rámcami prebieha v 4 krokoch (obr. 1.6), to znamená 4

transformačné matice, kde po ich vynásobení získavame jednu transformačnú maticu  $A_i$ . Táto matica vyjadruje transformáciu medzi referenčným rámcom  $i-1$ , na ktorý nadväzuje rámec  $i$ , preto sa často označuje  $A^{i-1}_i$ , resp.  ${}^{i-1}A_i$ , čo doslova znamená “vyjadrenie súradnicového priestoru, rámca,  $i$  v priestore  $i-1$ ”.

Využitie je jednoduché. Na vyjadrenie pozície bodu  $P = (x,y,z)$  zo súradnicového systému  $i$ , označenie  $P^i$ , v súradnicovom systéme  $i-1$ , označenie  $P^{i-1} = (x',y',z')$ , použijeme násobenie matic. Konkrétne:  $(x',y',z',1)^T = {}^{i-1}A_i \times (x,y,z,1)^T$

### 1.2.3.2 Dopredná kinematika

**Dopredná kinematika** robotického ramena je problematika zaoberajúca sa nájdením súradníc efektora pri danej štruktúre a známom nastavení kĺbov ramena.

Vďaka transformáciám medzi referenčnými rámcami pomocou H-D transformačných matic a danou štruktúrou ramena, kde referenčný rámec základne spĺňa  $i = 0$  a referenčný rámec efektora  $i = n$ , vieme vyjadriť celkovú transformáciu, z referenčného rámca efektora do rámca základne,  ${}^0T_n$  ako výsledok zret'azenia transformácií  ${}^0A_1 \times \dots \times {}^{i-1}A_i \times \dots \times {}^{n-1}A_n$

Pozíciu efektora vyjadrenú v súradnicovom systéme základne,  $P^0 = (x^0,y^0,z^0)$ , potom nájdeme ako výsledok rovnice:  $(x^0,y^0,z^0,1)^T = {}^0T_n \times (0,0,0,1)^T$  pričom bod  $(0,0,0)$  vyjadruje počiatkový bod v referenčnom rámci efektora.

### 1.2.3.3 Inverzná kinematika

**Inverzná kinematika** robotického ramena sa zaoberá opačným problémom ako dopredná kinematika. Hľadá vhodnú konfiguráciu variabilných parametrov ramena k tomu, aby sa efektor nachádzal na zadanom bode a so zadanou orientáciou. Tento problém je omnoho komplexnejší a nieje tak priamočiary ako dopredná kinematika. Navyše, pri doprednej kinematike je bod efektora unikátny, jednoznačne určený konfiguráciou ramena, pri IK v 3-D priestore existuje niekoľko možných riešení problému s ohľadom na orientáciu ramena.

Existuje mnoho prístupov k riešeniu IK a stále sa objavujú nové. Môžeme ich rozdeliť na:

- analytické
- iteratívne
  - numerické
  - heuristické

#### **1.2.3.3.1 Analytický prístup**

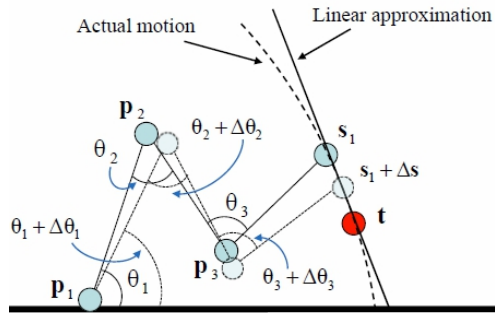
Analytickým prístupom k riešeniu problému inverznej kinematiky robotického ramena si rozdelíme celý systém kinematického zreťazenia na menšie jednoduchšie podsystémy a riešime ich samostatne. Tento prístup má význam len u menej zložitých kinematických zreťazení s malým počtom stupňov voľnosti DOF. V optimálnom prípade sa dá problém zjednodušiť až na súbor čisto grafických riešení a výsledná konfigurácia sa vyráta priamo, bez potreby iteratívnych aproximácií konfigurácie.

#### **1.2.3.3.2 Numerický prístup**

##### **Jacobian [3][4][5]**

Najčastejšie sa pri riešení IK numerickým prístupom využíva Jacobian matica, čo je matica parciálnych derivácií vektorovej funkcie vzhľadom na iný vektor. Podstata Jacobianu spočíva v tom, že reprezentuje najlepšiu lineárnu aproximáciu k diferencovateľnej funkcii blízko daného bodu. V prípade IK riešení sa používa Jacobian na lineárnu aproximáciu pohybu efektora vzhľadom na meniacu sa konfiguráciu parametrov ramena. Lineárnu aproximáciu je nutné použiť, nakoľko inverzná funkcia k funkcii, získanej z rovníc doprednej kinematiky, z ktorých IK vychádza, je nelineárna a zobrazenie nieje jednoznačné. [4]

Existuje viacero prístupov a korekcií, ktoré vylepšujú pôvodnú “Jacobian inverse” metódu: Jacobian pseudo-inverse, Jacobian Transpose, DLS Damped Least Square, atď.



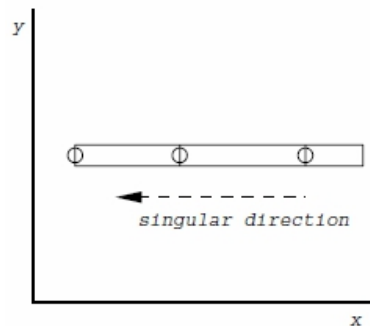
**Obr. 1.7** Lineárna aproximácia skutočného kinematického pohybu pri riešení pomocou Jacobian matice [3]

Výhody:

- výsledné konfigurácie pôsobia vizuálne vcelku prirodzene (napr. porovnateľne výsledkom CCD), hlavne ak je hľadaný bod vo väčšej vzdialenosti
- dobre sa implementuje a vyhodnocuje problém viacerých koncových efektorov

Nevýhody:

- veľká výpočtová náročnosť jednotlivých iterácií (v každej sa pracuje s maticami) a tým pádom dlhší čas výpočtu robí z Jacobiana menej vhodnú metódu k real-time aplikáciám.
- Metódy, ktoré využívajú inverznú Jacobian maticu, majú problémy so singularitami vo výpočtoch. Transponovaný Jacobian a DLS sa vyhýbajú priamym singularitám, ale zvyknú mať aj tak problémy v okolí možných singularít.
- Obmedzenia (samotných kĺbov alebo priestorové) sa síce dajú implementovať do výpočtov, ale nieje to zrejme a priamočiara úloha



**Obr. 1.8** Manipulátor v singularitnej konfigurácii [4]

## Singularity

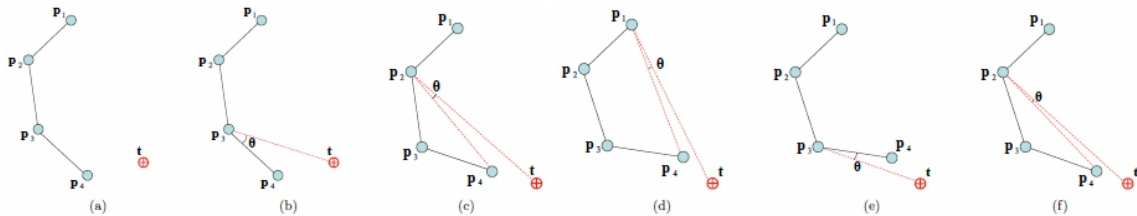
Pri Jacobianoch sa dostáva manipulátor do singularnej konfigurácie, ak je Jacobian matica singularna, t.j. jej determinant sa nerovná nule, jej riadky sú lineárne nezávislé.

Singularity sú body, pre ktoré buď nie sú funkcie definované, alebo v nich veličiny získavajú nepríjemné vlastnosti, tu konkrétne má manipulátor nekonečné zrýchlenie. Keď sa vyskytnú singularity, žiadnou zmenou uhlov manipulátor nedokáže dosiahnuť požadovaný bod.

### 1.2.3.3 Heuristický prístup

#### CCD [3][4]

**Cyclic Coordinate Descent** je jednoduchá a populárna metóda, ktorá sa často využíva v počítačových grafických systémoch a v robotike.



Obr. 1.9 Ukážka CCD algoritmu [3]

Algoritmus metódy CCD je veľmi jednoduchý. Cyklus prechádza od poslednej časti zretžazenia manipulátora, efektoru, až po začiatok a pokračuje znova od efektoru. V každej iterácii sa snaží zmenou príslušného kĺbu priblížiť efektor čo najbližšie k cieľu (otočiť kĺbom aktuálnej iterácie tak, aby sa efektor presunul na pozíciu na pomyselné spojnice medzi aktuálnym kĺbom a cieľom). Cyklus sa skončí ak je efektor v prípustnom okolí cieľa, alebo ak sa dosiahne určené maximum iterácií.

Výhody:

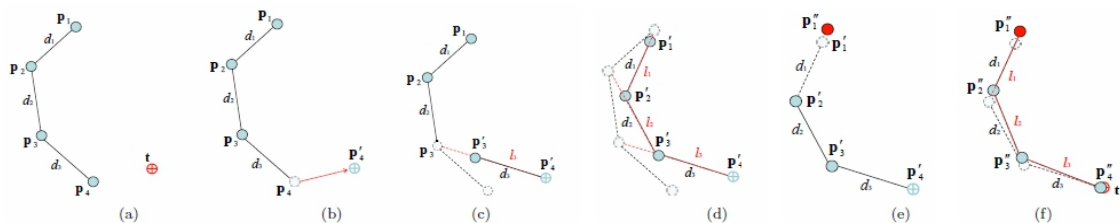
- CCD má nízku výpočtovú zložitosť a je omnoho rýchlejšia než Jacobian metódy (nemusia pracovať s maticami), čo ju robí populárnou v real-time aplikáciách
- nemá problém so singularitami [4]

Nevýhody:

- najväčšou nevýhodou metódy CCD sú neprirodzený pohyb a aj možná výsledná konfigurácia, nakoľko sa často počas výpočtu zvykne manipulátor zvinúť sám do seba. Na realistickejší pohyb je nutné určiť (kĺbové) obmedzenia.
- CCD je dizajnovaná na prácu s jednoduchým sériovým zreťazením, musí byť teda modifikovaná na prácu s viacerými koncovými efektormi.
- Preceňuje význam pohybu s kĺbmi blízko efektora, čo má za následok veľa zbytočných krokov, obzvlášť pri finálnom priblížení
- Pridať globálne (priestorové) obmedzenia je zložité

### FABRIK [3]

FABRIK (Forward and Backward Reaching Inverse Kinematics) je pomerne nová heuristická metóda na riešenie IK. Má malú výpočtovú zložitosť, je rýchlejšia než CCD, funguje aj pre viacero koncových efektorov a obmedzenia sa dajú pomerne ľahko implementovať. Navyše nemá problém ani s neprirodzeným pohybom a singularitami.



Obr. 1.10 Algoritmus metódy FABRIK [3]

Algoritmus je o niečo zložitejší než CCD, je rozdelený na 2 časti. V prvej časti prejde iteratívne od posledného bodu zreťazenia, efektora, po počiatočný bod zreťazenia. Bod efektora inicializuje na pozíciu cieľa a potom iteratívne mení pozíciu kĺbu, ktorý predchádza iteratívne aktuálnemu kĺbu v zreťazení, tak, že tento kĺb presunie na pomyselnéj spojnici medzi nimi do takej vzdialenosti, aby spĺňal dĺžku pôvodnej vzdialenosti medzi týmito kĺbmi. V druhej časti algoritmu iteratívne prejde od počiatočného bodu zreťazenia k poslednému bodu zreťazenia, efektora. Najskôr inicializuje, teraz už zmenený, počiatočný bod zreťazenia

na jeho pôvodnú pozíciu a opakuje postup z prvej časti, len teraz ovplyvňuje od iteratívne aktuálneho kĺbu jeho nasledujúci kĺb v zret'azení.

### Porovnanie rýchlostí a výpočtovej náročnosti

Príklad reálneho experimentu na ukázanie výpočtovej náročnosti numerických metód oproti heuristickým metódam. Na obrázku (1.11) je tabuľka priemerných výsledkov z 20 pokusov na kinematickom zret'azení s 10 kĺbmi.

**Table 3.1.:** Average results (over 20 runs) for a single kinematic chain with 10 joints.

	Reachable Target				Unreachable Target	
	Number of Iterations	Matlab exe. time (sec)	Time per iteration (in msec)	Iterations per second	Number of Iterations	Matlab exe. time (sec)
FABRIK	15.461	0.01328	0.8	1164	67.564	0.06207
CCD	26.308	0.12356	8.8	213	390.135	3.92869
Jacobian Transpose	1311.190	12.98947	9.9	101	6549.000	33.90473
Jacobian DLS	998.648	10.48051	10.5	95	2881.667	14.87918
Jacobian SVD-DLS	808.797	9.29652	11.5	87	2808.452	15.97591

**Obr. 1.11** Porovnanie efektívnosti algoritmov IK, test 1 [3]

Na obrázku (1.12) je znázornení rozdiel vo výsledkoch, keď sa implementovali (kĺbové) obmedzenia pre metódy CCD a FABRIK.

**Table 3.3.:** Average results when joint constraints are incorporated.

	Number of Iterations	Matlab exe. time (sec)	Frames per second
FABRIK	15.461	0.01328	75.301
CCD	26.308	0.12359	8.091
FABRIK Constrained	17.142	0.03110	32.154
CCD Constrained	26.857	0.29281	3.415

**Obr. 1.12** Porovnanie efektívnosti algoritmov IK, test 2 [3]



## 1.3 Prehľad a analýza technológií

### 1.3.1 AVR mikroradiče a technológia

AVR je označenie radu 8-bitových RISC programovateľných jednočipových mikroradičov od firmy Atmel. Je založená na modifikovanej Harvardskej architektúre, čo znamená, že má dátovú pamäť a pamäť programu fyzicky oddelenú a tiež aj samostatné zbernice od nich k procesoru, čo výrazne urýchľuje vykonávanie inštrukcií. Flash, EEPROM a SRAM sú všetky integrované na jednom čipe.

- **Flash** – nevolatilná pamäť, čiže uchováva informácie aj bez zdroja napájania, slúži ako pamäť programu, je možné ju elektricky preprogramovať či vymazať, až 10000 krát
- **EEPROM** – nevolatilná, je pomalšia a menšia ako Flash pamäť, slúži na uchovávanie konfiguračných informácií, možné ju prepísať elektronicky až 100000 krát
- **SRAM** – volatilná, uchováva dáta, registre za behu programu

Medzi dôležité vlastnosti a súčasti AVR mikroradičov, ktoré sa často využívajú patria:

- **ISP In-System Programming** – umožňuje elektronicky prepísať pamäť programu mikroradiča bez potreby fyzicky vyberať čip zo systému, používa sa na to ISP programátor [11], ktorý komunikuje cez zbernicu SPI
- **8-bit a 16-bitové časovače** – časovače s možnosťou vyvolávať **prerušená** (interrupts) a vytvárať **PWM signál** (1.3.3.1)
- **analog-digitálne prevodníky**
- **prerušená (interrupts)** – externé aj interné (ADC, časovače, USART, ...)
- **4 input-output porty** – 8-bitové, PORTA až PORTD
- sériové rozhrania
  - **SPI** – využíva sa pri ISP
  - **UART/USART** – (1.3.2)
  - **I<sup>2</sup>C**

## AVR Studio – IDE

[http://www.atmel.com/microsite/avr\\_studio\\_5/default.aspx](http://www.atmel.com/microsite/avr_studio_5/default.aspx)

Pôvodne sa mikroradiče programovali len v assembleri, neskôr sa pridala podpora vyšších programovacích jazykov ako napr. C. AVR má vlastné IDE (integrované vývojové prostredie) pre programovanie svojich mikročipov, v ktorom má už integrované knižnice na prácu s avr registrami, prerušeniami či nastavením kódu pre konkrétny AVR čip.

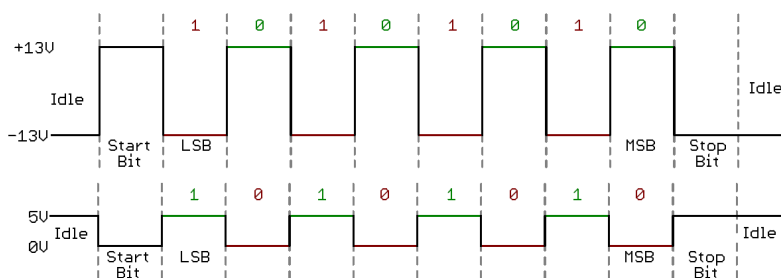
## AVRDUDE – Downloader/UploaDEr

<http://savannah.nongnu.org/projects/avrdude>

Ide o open-source program, ktorý sa používa pri ISP programovaní AVR mikroradičov.

### 1.3.2 Sériová komunikácia, USART

USART znamená Universal Synchronous/Asynchronous Receiver/Transmitter, čiže Synchronne/Asynchronne sériové komunikačné rozhranie, ktoré sa bežne vyskytuje na mikroradičoch a slúži na sériovú komunikáciu.



Obr 1.13 TTL (dolu) a RS-232 , obe posielajú informáciu 0b01010101 [17]

Sériová komunikácia je proces prenášania informácie po jednotlivých bitoch, narozdiel od paralelnej, ktorá posielá viac bitov naraz. USART vnútorne prekladá medzi paralelnou a sériovou reprezentáciou dát. Existuje viacero štandardov pre sériový prenos dát, najznámejším a v minulosti najpoužívanejším je RS232. Ten však pri komunikácii reprezentuje hodnoty 0 a 1 až napätím medzi  $\pm 3$  a  $\pm 25$ , zväčša  $\pm 12$  až  $\pm 15$ V a súčasné obvody mikroradičov sú napojené na zdroj  $V_{cc}$  5V resp. 3.3V resp. ešte menej a založené na logike TTL či CMOS, ktoré hodnoty 0 a 1 reprezentujú v rozmedzí 0 až +5 V.

Pre AVR mikroradiče môžeme zjednodušiť, že USART komunikuje na TTL logike. Dá sa rozdeliť do troch blokov:

- **Rx prijímač (receiver)** – má buffer do ktorého sa zapisujú prijaté bajty
- **Tx vysielateľ (transmitter)** – má buffer z ktorého sa vysielajú bajty
- **generátor rýchlosti (clock generator)** – riadi rýchlosť komunikácie

Fyzické komunikačné zariadenia sa prepoja tak, aby Tx jednej strany bol zapojený na Rx druhej strany a zem (GND) prepojená so zemou (GND). Treba dať pozor aby sa nezapojil Tx na Tx, lebo bez dostatočne veľkého ochranného rezistora pri takomto zapojení zhorí mikročip.

Následne ešte treba nastaviť:

- **baud rate** – prenosová rýchlosť, počet symbolov / impulzov za sekundu
- **data bits** - počet bitov v jednom byte / bloku, defaultne 8
- **parity** – nastaviť paritu, defaultne no-parity
- **stop bit** – počet stop bitov, defaultne 1
- **flow control** – kontrola pretečenia, defaultne no-flow control

USART umožňuje nastaviť medzi 2 spôsobmi komunikácie. Nastavenie na synchronnú komunikáciu funguje ako full-duplex, zároveň vysielala aj prijíma. Nastavenie na asynchrónnu komunikáciu funguje ako half-duplex a v jednom momente môže buď len vysielateľ alebo len prijímať.

### 1.3.3 Servomotory a DC motory

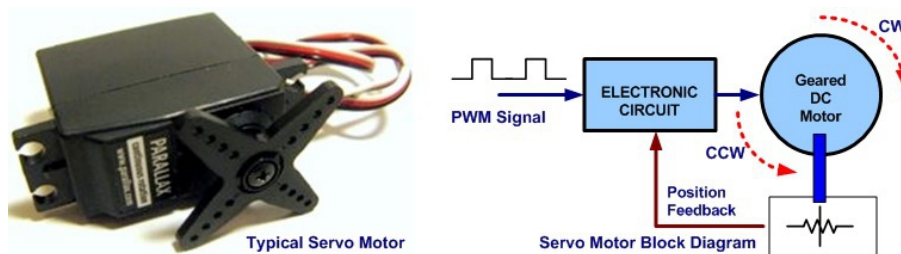
Elektrický **motor** na jednosmerný prúd **DC** (direct current) je zariadenie meniace jednosmerný elektrický prúd na mechanickú silu. Obojsmerný motor sa ovláda cez tri vstupy - signály: 2 input a 1 enable, kde inputy nastavujú smer otáčania a enable nastavuje rýchlosť otáčania, napr cez PWM (1.3.3.1). Motor môže byť v jednom zo 4 módov:

- točí sa v smere hodinových ručičiek
- točí sa proti smeru hodinových ručičiek
- brzdí
- voľnobeh

Keďže mikroradiče fungujú na menších hodnotách napätia ako motory, treba medzi mikroradič a motor vložiť medzičlánok, inak by sme s motormi nepohli a ešte by sme aj mohli spáliť mikroradič. Radiče motorov (motor driver, controller, H-bridge) zosilnia signály na potrebnú hodnotu.

Pod pojmom **servomotor** sa označuje servomechanizmus, čo je zariadenie, ktoré na základe rozdielu spätnej väzby zo senzorov a referenčného vstupu upravuje výkon svojho mechanizmu. V robotike sa využívajú modelárske RC jednosmerné servomotory, ktoré väčšinou tvoria:

- **elektrický DC motorček** – samotný mechanizmus servomotora
- **prevodové kolečká** – prenášajú krútiaci moment z motorčeka
- **potenciometer** – zariadenie, ktoré slúži ako senzor aktuálnej polohy mechanizmu a umožní spätnú väzbu
- **kontrolný obvod** – elektronický obvod, ktorý vyhodnocuje rozdiel medzi referenčnou hodnotou – vstupom a medzi aktuálnym nastavením mechanizmu a posiela informáciu motorčeku



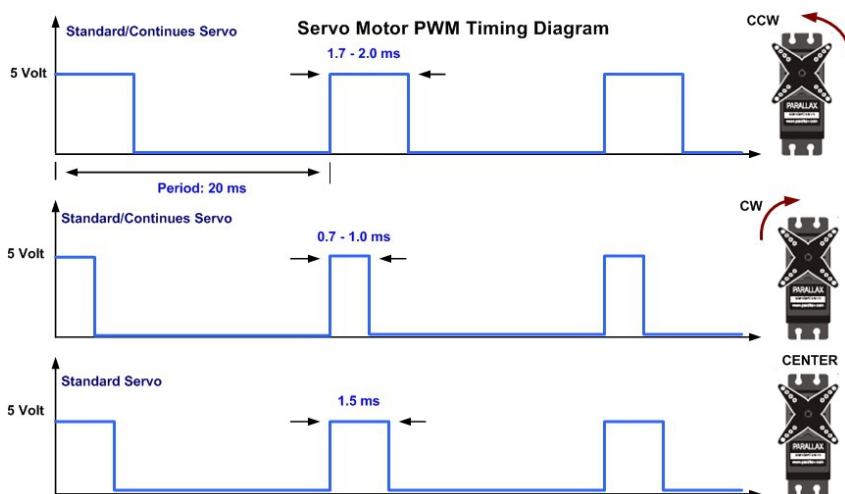
**Obr. 1.14** Ukážka typického RC servomotora a diagram pracovného cyklu [18]

RC servomotory sa pripájajú cez 3 konektory, jeden pre vstupné napätie  $V_{cc}$  4.8V až 6V, jeden pre uzemnenie (GND) a jeden pre signál - vstup referenčnej hodnoty cez PWM signál (1.3.3.1).

### 1.3.3.1 PWM signál

PWM (Pulse Width Modulation) Impulzová šírková modulácia je technika posielania informácie zakódovanej do šírky signálu v určenom rozsahu. Využíva sa či už na komunikáciu, prevod analógovej informácie do digitálnej alebo na moduláciu elektrického výkonu, dodávaného do zariadenia.

Funguje to na princípe zmeny dĺžky signálu posielaného v rozsahu danej periódy. Označuje sa to strieda (duty cycle) a je to dĺžka aktívneho stavu – signálu, voči dĺžke periódy a vyjadruje sa v percentách, 100% znamená, že celú periódu je signál 'zapnutý' a 0% znamená signál celú periódu 'vypnutý'. Napr. pri modulácii elektrického výkonu zariadenia určujeme šírkou signálu percentuálny pomer posielaného el. napätia.



Obr. 1.15 Diagram popisujúci nastavenia RC servomotora PWM signálom [18]

### Servomotory

RC servomotory majú pevne určenú frekvenciu 50Hz, teda šírka periódy je 20 ms a dĺžku signálu väčšinou v rozmedzí 0.5 – 2.5ms. Signál o dĺžke 1.5 ms teda nastaví servomotor do strednej polohy, signál 0.5 – 1.5ms bude točiť servomotor jedným smerom k hraničnej hodnote natočenia, 1.5 – 2.5ms druhým smerom. Smer otočenia závisí od konkrétneho servomotora. Mikroradič teda zakóduje požadovaný stav servomotora do signálu o rozsahu do 2 ms, pošle ho cez signálový konektor servomotora periodicky každých

20ms. Kontrolný obvod servomotora signál preloží (dekóduje), porovná s aktuálnym nastavením motorčeka a vykoná príslušný pohyb.

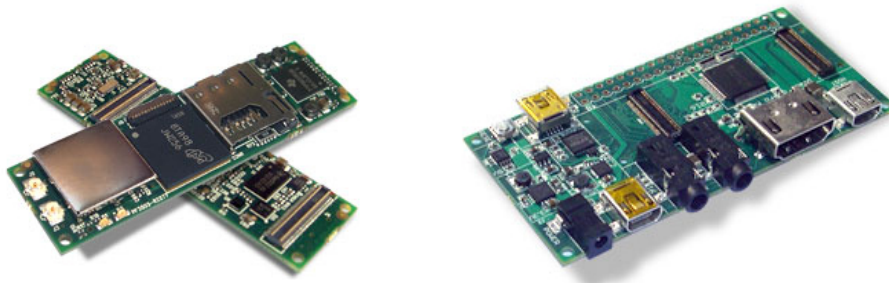
### DC motory

DC motorom sa PWM signálom posiela zakódovaná informácia o rýchlosti otáčania. Jednoducho, čím väčšia šírka signálu v perióde, tým viac elektrického napätia preosiela radič motora a tým rýchlejšie sa motor bude točiť.

Narozdiel od servomotorov, DC motory nemajú predpísanú frekvenciu pre PWM signál a preto treba experimentálne nájsť správnu hodnotu frekvencie a teda aj šírky periódy.

### 1.3.4 Gumstix

Gumstix je jedna z firiem ponúkajúcich tzv. SBD single-board computers, čo je typ embedovaného (uzavretého, používa sa výhradne ako celok) počítačového systému. Jedná sa o kompletne počítače (procesor, pamäť, vstupno-výstupné zariadenia,...) zabudované na jednej doske (plošnom spoji). Vďaka svojim malým rozmerom, nízkej spotrebe energie a zároveň veľkej časti funkcionality štandardného počítačového systému nachádzajú uplatnenie aj v robotike.



**Obr 1.16** Gumstix Overo Air COM a Gumstix expansion board Summit

V prípade Gumstixu je koncept SBD rozdelený na dve časti, na tzv. COM computer-on-module a EB expansion board (rozširujúca doska). COM takmer spĺňa požiadavky na SBD, má na jednej doske procesor aj pamäť, ale od vstupno-výstupných zariadení má len ovládače, samotné konektory k nim sa nachádzajú na rozširujúcej doske. Rozdelenie na moduly má za cieľ ponúknuť personalizovanejšie zariadenia, kde si užívateľ môže zvoliť rôzne zostavy a ľahšiu modernizáciu menej komplexných častí.

Operačný systém môže užívateľ nahrat' do flash pamäte na COM, alebo mať prenosný, uložený na SD karte, ktorá sa dá tiež zasunúť do COM. Gumstix ponúka svoj embedovaný systém linux, ale k dispozícií sú nástroje na vytvorenie vlastnej verzie linuxu či WindowsCE.

## Kapitola 2

# Špecifikácia práce

Máme vybrať konštrukčnú zostavu robota, ktorá sa skladá z mobilnej platformy *Rover A4WD1 v2* od firmy *Lynxmotion* a robotického ramena *ALB5* tiež od firmy *Lynxmotion*.

Úloha spočíva vo vytvorení robotického integrovaného systému, t.j.

- skonštruovanie štruktúry robota zo zadaných mechanických častí
- navrhnuť a vyrobiť integrovaný elektronický systém s ovládacími prvkami pre oba samostatné robotické podsystémy a riadiacimi prvkami pre obsluhu daných podsystémov
- pri návrhu elektronického systému treba prihliadať na plánovanú budúcu integráciu senzorických systémov videnia a ďalších

Požiadavky na ovládacie prvky elektronického systému sú:

- fyzické a elektronické prepojenie na jednotlivé mechanické časti robotického systému
- elektronické rozhranie a funkcionálna na ovládanie mechanických častí robotického systému
- implementácia komunikačných rozhraní s riadiacimi prvkami elektronického systému

Požiadavky na riadiace prvky elektronického systému sú:

- implementovať obslužnú a riadiacu funkcionálnu pre ovládanie všetkých obslužných podsystémov
- implementovať komunikačné rozhrania (fyzické rozhranie, komunikačné protokoly) so všetkými obslužnými podsystémami
- implementovať funkcionálnu doprednej a inverznej kinematiky pre systém robotického ramena
- schopnosť komunikácie s operátorom (napr. PC Host) cez štandardné komunikačné



rozhrania, napr. káblom(usb, ...) alebo a aj bezdrôtovo (wifi, bluetooth,... )

- pri návrhu prihliadať na plánované budúce rozšírenie robotického systému (konkrétne stereovidenia *Surveyor Stereo Vision System* od firmy *Surveyor Corporation*, a ďalšie...).

# Kapitola 3

## Návrh práce

### 3.1 Analýza špecifikácie práce

Pre prostredie práce si zvolíme robotické laboratórium na fakulte FEI STU, blok D, 7.poschodie. Nachádza sa tam vybavenie pre manuálnu prácu a prácu s elektronikou.

#### **Mechanické časti**

Robotické rameno *ALB5 Lynxmotion* obsahuje 5 servomotorov (3.3.3). K ich riadeniu bude teda treba 5 PWM signálov (3.3.3.1). Ako zdroj použijeme 6V batériu.

Mobilnú platformu *Rover A4WD1 v2* poháňajú 4 DC motory. K ich riadeniu bude teda treba 4 PWM signály (enable) a 8 smerových signálov (inputy) (3.3.3). Ako zdroj vyžadujú až 12V, pridáme teda ďalšiu 6V batériu.

#### **Elektronické časti**

Pre zachovanie samostatnosti podsystemov a prehľadnosti štruktúry, ale v konečnom dôsledku aj obmedzeniam technológie bežných mikroradičov (počet komunikačných portov na signály, fyzických pinov, ...) bude každý podsystem riadený vlastným mikroradičom AVR (3.3.1) od firmy *Atmel* [12][13]. Vstupné napätie majú v rozsahu 2.7 – 5.5V, ako zdroj teda použijeme 5 AA batérií s napojením na regulátor s výstupom 5V.

AVR mikroradiče fungujú na TTL 5V logike. Z toho vyplýva, že riadiace signály servomotorom môžu posielat' priamo, no na ovládanie DC motorov bude potrebovat' zosilnit' signál cez radiče motorov. (3.3.3)

Ako riadiaci system použijeme single-board computer (3.3.4) od firmy *Gumstix*. Nakoľko je

požiadavka aj na bezdrôtovú komunikáciu, vyberieme si COM (3.3.4) s touto technológiou, konkrétne *Gumstix Overo Air*. Komunikáciu káblom nám zabezpečí expansion board (3.3.4), konkrétne *Gumstix Summit*. *Gumstix Summit* vyžaduje vstupné napätie v rozsahu 3.5 – 5V, ako zdroj teda môžeme použiť už existujúci výstup z regulátora 5V od 5AA batérií.

### **Komunikácia**

Komunikáciu medzi mikroradičmi a riadiacou jednotkou *Gumstix* bude zabezpečovať sériové komunikačné rozhranie. Na strane mikroradičov cez USART(1.3.1 a 1.3.2), na strane *Gumstixe* cez USB host. Tu bude treba konverziu, načo využijeme USB-TTL konvertér [14]. Keďže by sme radi komunikovali na čo najrýchlejšej štandardne používanej rýchlosti prenosu 115200 bps, treba zabezpečiť čo najmenšiu bezstratovosť a bezchybnosť prenosu správne nastavenou frekvenciou mikroradičov. [23, napr. strana 197]

Komunikácia medzi *Gumstixom* a operátorom (reprezentovaný ako štandardný PC host) bude zabezpečovať buď USB rozhranie, alebo bezdrôtové rozhranie (*Gumstix Overo Air* má funkcionálnosť Bluetooth aj Wireless).

Plánované stereovidenie *Surveyor* bude môcť využívať práve Wireless rozhranie.

### **Funkcionalita**

Jeden z dôvodov výberu AVR mikroradičov je ich programovateľnosť, podpora výrobcu kvalitným IDE a výber mnohých návodov a manuálov z dôvodu ich rozšírenosti. Obslužné funkcie zabezpečované mikroradičmi budú naprogramované v jazyku C a pomocou ISP programovania(1.3.1) AVR mikroradičov budú implementované do príslušného mikroradiča.

Na riadiacom zariadení, *Gumstixe*, bude bežať operačný systém Linux a vďaka podpore rôznych kompilátorov máme na výber medzi programovacími jazykmi. Použijeme jazyk C++ a kompilátor g++.

## 3.2 Dekompozícia robotického systému

Robotický systém si môžeme rozdeliť na základe analýzy špecifikácií podľa viacerých kritérií.

### **Mechanika:**

- robotický systém
  - mobilná platforma
  - robotické rameno

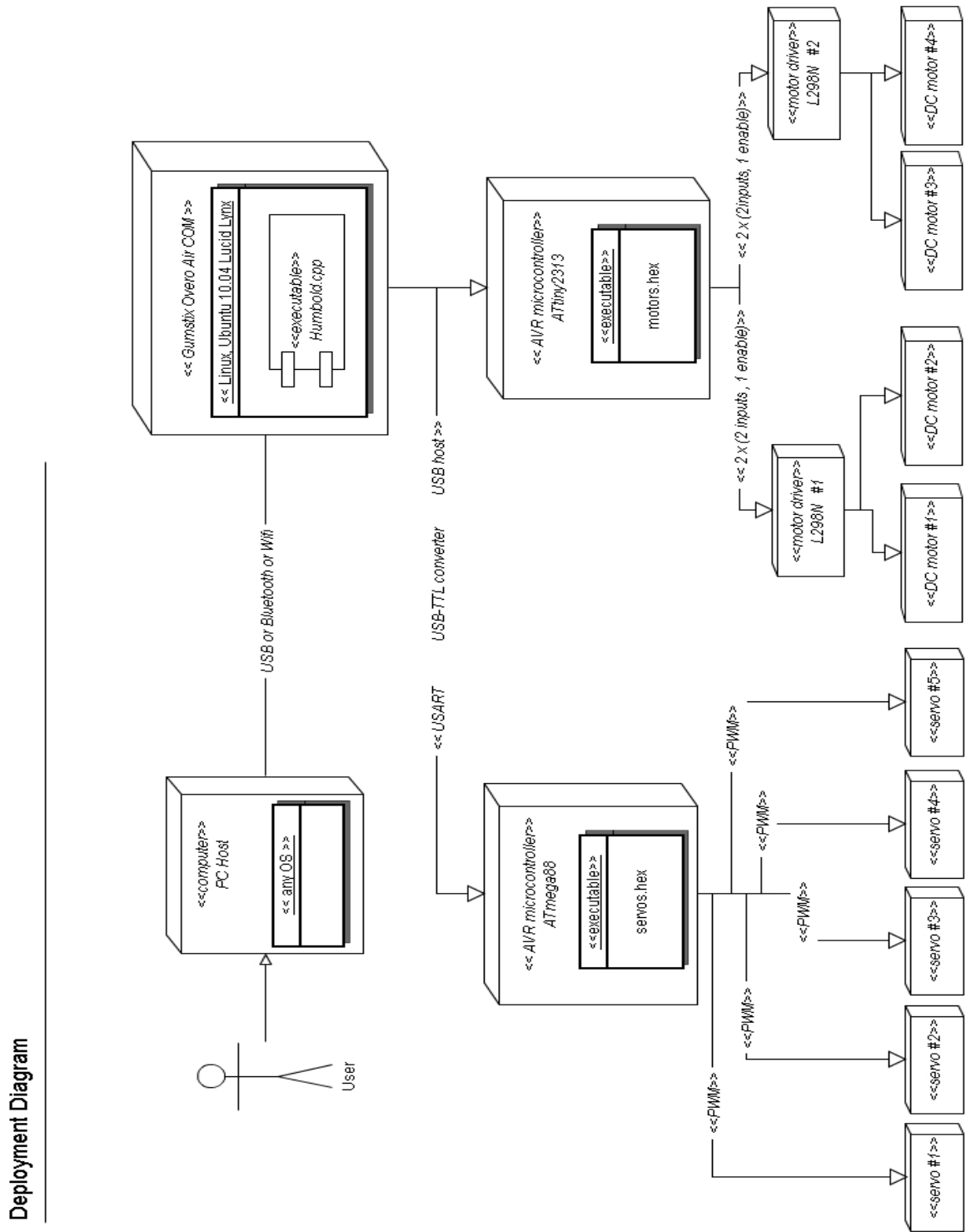
### **Funkcionalita:**

- Gumstix
  - výpočet kinematiky pre pohyb robotického ramena
  - riadiace funkcie na prácu s mikroradičmi
  - komunikácia s mikroradičmi
  - komunikácia s operátorom
- AVR ATmega88
  - ovláda servomotory ramena
- AVR Attiny2313
  - ovláda DC motory mobilnej platformy
- PC host (operátor)
  - zadáva príkazy

### **Riadenie:**

- Operátor
  - Gumstix
    - AVR ATmega88
      - motory mobilnej platformy
    - AVR Attiny2313
      - servomotory ramena

Grafické zobrazenie dekompozície:



Obr. 2.1 Deployment diagram robota

## **3.2.1 HW elektronika**

Nastavenia HW a ich význam.

### **3.2.1.1. ATmega88**

Vnútoraná frekvencia AVR mikročipu ATmega88 sa dá interne nastaviť na frekvencie, zväčša 4 či 8 MHz, [12, od strany 27], ktoré nám ale nevyhovujú pre štandardnú komunikačnú rýchlosť prenosu 115200 bps, pretože majú pri tejto rýchlosti veľkú chybovosť [12, od strany 195], preto pridáme externý kryštálový oscilátor, ktorý zmení frekvenciu mikroradiča na 11.0592MHz, čo je optimálna frekvencia pre danú rýchlosť prenosu [12, strana 197].

Ďalej nastavíme správne hodnoty registrov pri nastavení časovačov, aby posielali PWM signál s frekvenciou 50 Hz (1.3.3.1). Tento mikroradič má 3 časovače, každý vie vytvoriť výstupný signál na 2 pinoch a rameno má len 5 servomotorov takže nám to bude stačiť.

Treba implementovať obslužné funkcie na riadenie servomotorov.

Ešte potrebujeme povoliť prerušenia pri prijatí znaku na sériovom komunikačnom rozhraní USART.

Nakoniec implementujeme funkcie na spracovanie prijatej informácie podľa zadaného komunikačného protokolu. (3.2.4)

### **3.2.1.2. ATtiny2313**

Podobne ako u predošlého mikroradiča, vnútoraná frekvencia AVR mikročipu ATtiny2313 sa dá interne nastaviť na frekvencie, zväčša 4 či 8 MHz, [13, od strany 23], ktoré nám ale nevyhovujú pre štandardnú komunikačnú rýchlosť prenosu 115200 bps, pretože majú pri tejto rýchlosti veľkú chybovosť [12, napr. od strany 134], preto pridáme externý kryštálový oscilátor, ktorý zmení frekvenciu mikroradiča na 11.0592MHz, čo je optimálna frekvencia pre danú rýchlosť prenosu [12, napr. strana 136].

Pri DC motoroch nemáme predpísanú povinnú frekvenciu PWM signálu (1.3.3.1) ako u servomotorov, preto nastavíme frekvenciu PWM signálu posielaného časovačmi experimentálne.

Tento mikroradič má síce len 2 časovače, ale každý vie vytvoriť výstupný signál na 2 pinoch a máme len 4 DC motory takže nám to bude stačiť.

Ešte potrebujeme povoliť prerušenia pri prijatí znaku na sériovom komunikačnom rozhraní USART.

Nakoniec implementujeme funkcie na spracovanie prijatej informácie podľa zadaného komunikačného protokolu. (3.2.4)

### **3.2.2 Gumstix**

Vyberieme si ktorú distribúciu linuxu budeme chcieť používať ako operačný systém na *Gumstixe*. Keďže ide o embedovaný systém a výrobca ponúka vlastné už skompilované distribúcie, bude to prvá voľba.

V prípade, že nebudeme spokojní s danou distribúciou linuxu, výrobca ponúka aj nástroje na kompiláciu iných ľubovoľných distribúcií.

Následne implementujeme systém na riadenie obslužných funkcií mikroradičov spolu s komunikačným protokolom pre komunikáciu s mikroradičmi (3.2.4).

*Gumstix* bude na komunikáciu s mikroradičmi cez sériové komunikačné rozhranie potrebovať USB-TTL konvertér. [14]

### **3.2.3 PC host**

PC host bude slúžiť na spojenie sa operátora s *Gumstixom* na získanie prístupu k riadiacim funkciám pre ovládanie motorov a servomotorov.

Komunikovať budú cez štandardné rozhrania. V lepšom prípade bezdrôtovo, cez Bluetooth, v horšom prípade cez USB kábel, čo by obmedzovalo mobilitu robota.

### 3.2.4 Komunikačný protokol

Mikroradiče budú na komunikačnom kanáli zapojené sériovo za sebou. To znamená dva dôsledky:

- oba mikroradiče budú naraz prijímať rovnakú vstupnú informáciu
- oba mikroradiče môžu v rovnakom momente vyselať informáciu po tej istej linke

Ten druhý problém je neprijemnejší, nakoľko ak začnú oba naraz vyselať na tej istej linke, tak si vzájomne vyrušia vysielanú informáciu a na prijímač v *Gumstix* nepríde ani jedna v správnom tvare. Tento problém vyriešime jednoducho tak, že vyselať bude len ten, kto bude vyzvaný z *Gumstixu*.

Problém s nejednoznačnosťou adresáta vyriešime striktným určením komu je paket adresovaný hneď na jeho začiatku.

#### Komunikačný protokol

- Začiatok nového paketu neguje ľubovoľnú neukončenú informáciu z predchádzajúceho paketu. Označuje sa bytom označujúcim znak '!'
- Druhý byte určuje adresáta. Znak 'm' je pre motory, znak 's' je pre servomotory
- Nasleduje index zariadenia, ktorému sa má zmeniť hodnota. Servomotory 1-5, motory 1-4
- Pri pakete pre motory ešte nasleduje číslo označujúce stav v akom má byť motor (dopredu, dozadu, brzdiť, voľnobeh)
- Za tým nasledujú číslice, ktorých postupným skladaním sa získa nová hodnota motora
- Paket sa ukončuje znakom ';', kedy sa vyhodnotí prijatá informácia a uzavrie sa ďalšia komunikácia až kým nepríde nový štartovací znak
- Špeciálny znak 'v' môže byť vyhodnotený kedykoľvek počas spracúvania informácie, či pre motory, či pre servomotory. Ako výsledok pošle späť aktuálnu konfiguráciu pre



daný stav motorov, alebo servomotorov

- Znak 'b' môže byť vyhodnotení kedykoľvek počas spracúvania informácie pre servomotory. Ako výsledok pošle späť aktuálnu stav batérií.

**Tab. 2.1** Ukážka šablóny paketu pre servomotory, čiže ATmega88

Start Char	Adresát	Index serva	Hodnota pre nastavenia signálu, 1 časť	...	Stop Char
.	s	{0-9 v}	{0-9 v}	{0-9 v}	;

**Tab. 2.2** Ukážka šablóny paketu pre motory čiže ATtiny2313

Start Char	Adresát	Index motora	Hodnota pre nastavenie smeru motora	Hodnota pre nastavenia signálu, 1 časť	...	Stop Char
.	m	{0-9 v}	{0-9 v}	{0-9 v}	{0-9 v}	;

### 3.3 Štruktúra robotického ramena

Určené robotické rameno sa skladá z otočnej základne, 3 rotačných kĺbov a jedného efektora. Spolu má teda 4 stupne voľnosti DOF + efektor, teda 5 servomotorov.

Po umiestnení ramena do priestoru a určenia referenčných rámcov podľa štandardu H-D (1.2.3.1.1) dostaneme nasledujúcu tabuľku:

**Tab. 2.3** Ukážka šablóny paketu pre motory čiže Attiny2313

Referenčný rámec	$a_i$	$\alpha_i$ (alfa)	$d_i$	$\theta_i$
1	75	0	0	$q_2$
2	125	0	0	$q_3$
3	118	0	0	$-90 + q_4$
4	0	-90	22	$q_5$
5	0	0	45	90

Pozn. Narozdiel od H-D štandardu sú rámce označené klesajúcim indexom, t.j. rámec 5 je rámec otočnej základne a rámec 1 je rámec efektora

Kde hodnoty  $q_2$  až  $q_5$  sú stupne otočenia podľa aktuálneho nastavenia servomotorov v daných kĺboch.

- $q_2$  odpovedá natočeniu kĺbu zápästia. Na základe reprezentácie kĺbu v danom referenčnom rámci a meraní mechanického rozsahu kĺbu môže dosahovať hodnoty  $\langle 0, 125 \rangle$
- $q_3$  odpovedá natočeniu kĺbu lakťa. Na základe reprezentácie kĺbu v danom referenčnom rámci a meraní mechanického rozsahu kĺbu môže dosahovať hodnoty  $\langle 0, 147 \rangle$
- $q_4$  odpovedá natočeniu kĺbu ramena. Na základe reprezentácie kĺbu v danom referenčnom rámci a meraní mechanického rozsahu kĺbu môže dosahovať hodnoty  $\langle 0, 90 \rangle$
- $q_5$  odpovedá natočeniu kĺbu základne. Na základe reprezentácie kĺbu v danom referenčnom rámci a meraní mechanického rozsahu kĺbu môže dosahovať hodnoty  $\langle -180, 0 \rangle$

### 3.4 Funkcionalita

Robotické rameno bude mať okrem obslužných funkcií aj kinematickú funkcionality.

#### 3.4.1 Dopredná kinematika daného robotického ramena

Vďaka priamočiarej povahe tohto problému ide o jednoduchý krok dosadenia nájdených parametrov daného robotického ramena (3.3) do matic transformácií referenčných rámcov (1.2.3.1.1), ktorých vynásobením dostaneme jednu transformačnú maticu.

#### 3.4.2 Inverzná kinematika daného robotického ramena

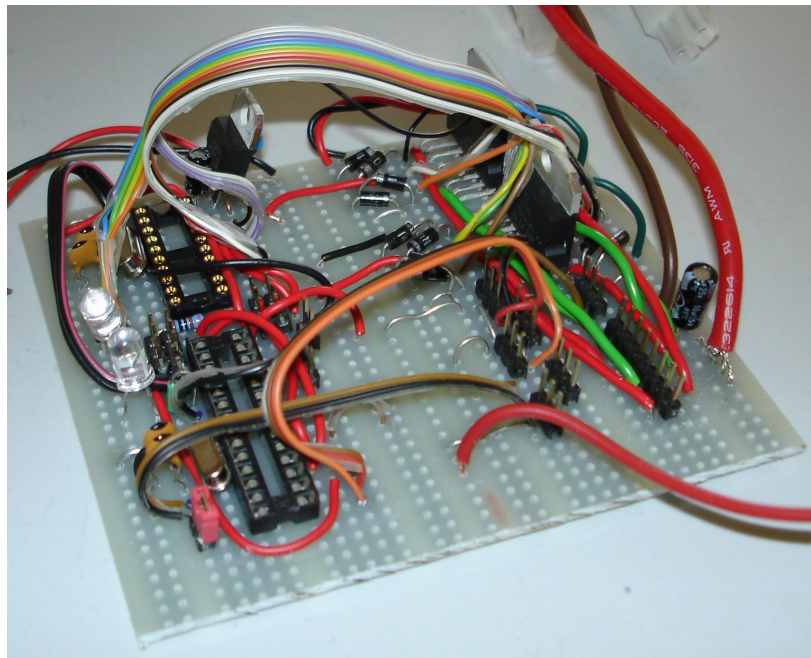
Problém inverznej kinematiky nieje tak priamočiary, ale nakoľko je štruktúra daného robotického ramena jednoduchá, dá sa analytickým prístupom zjednodušiť na grafické riešenie. Z povahy problému inverznej kinematiky ale vyplýva, že existuje viac možných konfigurácií na dosiahnutie danej pozície a tak je nutné zadať uhol zápästia, čiže uhol pod akým sa bude efektor približovať k hľadanej pozícii.

## Kapitola 4

# Realizácia, implementácia, výsledný systém

Robotický systém sa podarilo zdarne mechanicky skonštruovať podľa špecifikácie. Jediný rozdiel je, že servomotory 1,2 a 3 nie sú priamo ovládané PWM signálom z časovača, ale našou vlastnou implementáciou podstaty tohto signálu za pomoci prerušení.

Taktiež sa podarilo implementovať všetky riadiace a obslužné funkcie pre AVR mikroradiče. Z dôvodu opakovaných obštrukcií zo strany embedovaného systému *Gumstix* sme ale tieto funkcie testovali priamo spojením s PC hostom. Všetky časti robotického systému sa nám podarilo úspešne otestovať pri opakovaných pokusoch.



**Obr. 4.1** Doska s elektronikou na obsluhu robota v asi 2/3 procesu výroby

Implementácia doprednej a inverznej kinematiky sa taktiež podarila, ale už zo zmieneneho dôvodu zlej spolupráce s *Gumstixom* bola kinematika testovaná len virtuálne. Dosahovala rýchle a správne výsledky, po ošetrení aj v krajných pozíciách.

Robotický systém je plne pripravený na postupné rozširovanie o senzorické systémy a ďalšiu funkcionality spojenú s kinematikou mobilnej platformy.

Jediný, ale zo svojej podstaty príliš dôležitý, problém v implementácii návrhu je nedostačujúca funkcionality riadiaceho systému robota *Gumstix*.

# Záver

Napriek aktuálnemu neúspechu integrovať celý systém do jedného funkčného celku, nemôžem projekt považovať za neúspech. Z vytýčených cieľov sa podarilo splniť väčšinu, vrátane aktuálne hlavného cieľa, inverznej kinematiky. Navyše, až na bezdrôtovú komunikáciu, fungujú všetky jednotlivé časti na bilaterálnej úrovni v poriadku.

Robot je mechanicky skonštruovaný, z väčšej časti kopíruje zadané špecifikácie, má funkčné ovládanie pohonných jednotiek motorov aj robotického ramena. Komunikácia medzi jednotlivými časťami po fyzickej linke funguje správne, treba ešte doladiť bezdrôtovú komunikáciu.

Hlavne som spokojný s úspešným aplikovaním štandardu Denavit-Hartenberg, implementovaním doprednej kinematiky a nájdením rýchleho a spoľahlivého riešenia inverznej kinematiky.

Po doladení menších problémov s riadiacim systémom cez *Gumstix* sa môže plynulo pristúpiť k plánovanému rozšíreniu o systém senzorov. Pripravený je už systém stereovidenia a ďalšie nové zaujímavé funkcie vrátane kinematiky aj pre mobilnú platformu.

## Použitá literatura

- [1] Dudek, G. - Jenkins, M, 2000. *Computational Principles Of Mobile Robotics*. Cambridge University Press, 2000. ISBN 0-521-56021-7
- [2] Corrochano, E. B., 2001. *Geometric Computing for Perception Action Systems : Concepts, Algorithms and Scientific Applications*. Springer-Verlag New York , Inc, 2001. ISBN 0-387-95191-1
- [3] Aristidou, A – Hall, H, 2010. *Tracking and Modelling Motion For Biomechanical Analysis*. [ver. 5.10.2010] [online] Dostupné na internete 7.4.2012:  
<http://www.dspace.cam.ac.uk/bitstream/1810/237554/1/Tracking%20and%20Modelling%20Motion%20for%20Biomechanical%20Analysis.pdf>
- [4] Wells, Ch.,1993. *Inverse Kinematics and geometric constraints for articulated figure manipulation*. [online] Dostupné na internete 5.4.2012:  
[http://graphics.ucsd.edu/courses/cse169\\_w04/welman.pdf](http://graphics.ucsd.edu/courses/cse169_w04/welman.pdf)
- [5] Buss, S.R., 2009. *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods*. [online] Dostupné na internete 7.4.2012: <http://www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>
- [6] Forward Kinematics: the Denavit-Hartenberg convention.  
[online] Dostupné na internete 19.5.2012:  
<http://gmarani.org/wp/wp-content/uploads/2009/02/dh-parameters.pdf>
- [7] Williams II, R.L., 2011. *Robot Mechanics: Mechanics and Control of Robotic Manipulators*. [online] Dostupné na internete 19.5.2012:  
<http://www.scribd.com/doc/57606919/robo>
- [8] Choset, H. *Introduction to Robotics*. [online] Dostupné na internete 19.5.2012:

<http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/s06/lecture/lec16b.html>

- [9] Transformations in Homogeneous Coordinates, 2010.  
[online] Dostupné na internete 18.5.2012:  
<http://www.cs.iastate.edu/~cs577/handouts/homogeneous-transform.pdf>
- [10] Šolc, F – Nužil, T – Hrabec, J – Šembera, J, 2008. *Kinematický model kolového, smykem řízeného robota*. AT&P journal PLUS1, 2008.  
[online] Dostupné na internete 18.5.2012:  
[http://www.atpjournals.sk/buxus/docs/casopisy/atp\\_plus/plus\\_2008\\_1/plus01\\_05.pdf](http://www.atpjournals.sk/buxus/docs/casopisy/atp_plus/plus_2008_1/plus01_05.pdf)
- [11] Atmel Corporation, 2008. *AVR910: In-System Programming*.  
[online] Dostupné na internete 18.5.2012: <http://www.atmel.com/Images/doc0943.pdf>
- [12] Atmel Corporation, 2011. *8-bit Atmel Microcontroller. ATmega88*.  
[online] Dostupné na internete 15.4.2012: <http://www.atmel.com/Images/doc2545.pdf>
- [13] Atmel Corporation, 2010. *8-bit Atmel Microcontroller. ATtiny2313*.  
[online] Dostupné na internete 15.4.2012: <http://www.atmel.com/Images/doc2543.pdf>
- [14] FTDI Ltd, 2009. *TTL to USB Serial Converter Generic Cables. Datasheet*.  
[online] Dostupné na internete 23.4.2012:  
[http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_TTL-232RG\\_CABLES.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232RG_CABLES.pdf)
- [15] Sakshat Virtual Labs. [online] Dostupné na internete 18.5.2012:  
<http://iiith.vlab.co.in/index.php?sub=21&brch=72&sim=511&cnt=1> ,
- [16] Society of Robots. *Robot Arm Tutorial* [online] Dostupné na internete 19.5.2012:  
[http://www.societyofrobots.com/robot\\_arm\\_tutorial.shtml](http://www.societyofrobots.com/robot_arm_tutorial.shtml)

- [17] Sparkfun Electronics. [online] Dostupné na internete 19.5.2012:  
<http://www.sparkfun.com/tutorials/215>
- [18] Willem, B.R., 2009. Ermicroblog [online] Dostupné na internete 20.5.2012:  
<http://www.ermicro.com/blog/?p=771>



## Prílohy

### 1) CD

- zdrojové kódy AVR mikroradičov
- triedami robotického systému
- inverzná kinematika robotického ramena
- video z testu mechanickej funkcionality