Úloha priamej a inverznej kinematiky robotických systémov

Tutoriál k bakalárskej práci Modelovanie a simulácia robotických systémov v prostredí MATLAB/Simulink s využitím špecializovaných toolboxov

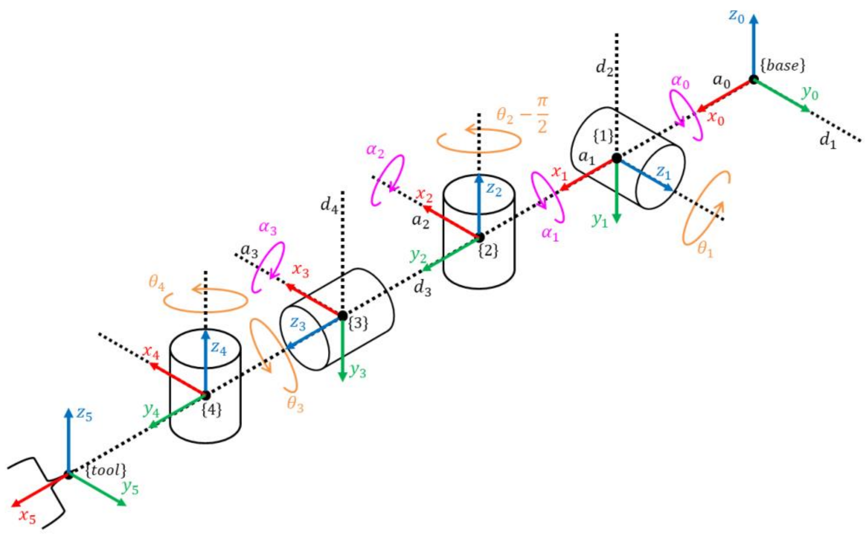
Úlohy:

1. **Postup pre odvodenie kinematického modelu vybraného robotického systému (subsystém robota NAO)**
   a. **Odvodenie matematického modelu**
   b. **Vytváranie simulačného modelu pomocou Robotics Toolboxu v prostredí MATLAB**
2. **Postup pre riešenie úlohy priamej kinematiky pre odvodený model**
3. **Postup pre riešenie úlohy inverznej kinematiky pre odvodený model**

Úloha č.1 : Postup pre odvodenie kinematického modelu vybraného robotického systému (subsystém robota NAO)

1.a. Odvodenie matematického modelu

Pre získanie modelu vybraného robotického systému, ktorým je ľavá ruka robota NAO, potrebujeme s ním uvažovať v zero angle konfigurácii ako môžeme vidieť na obrázku Obr. 1.



Obr. 1. Kinematický model ľavej ruky robota NAO

Model vyjadríme pomocou modifikovanej Denavit-Hartenbergovej konvecie a postupujeme nasledovne.

- Od hlavného základného rámca **O**, celého robota, ku rámcu kinematického reťazca ľavej ruky sa dostaneme pomocou translácii po osiach *y* a *z*.
- Pre získanie osi rotácie prvého kĺbu {1} otočíme náš základný rámec o $-\frac{\pi}{2}$ okolo osi x_0, teda parameter $\alpha_0 = -\frac{\pi}{2}$. Dodatočné translácie nie sú potrebné, a teda konštanty a_0 a d_1 sú nulové.
- Ďalšou rotáciou $\alpha_1 = \frac{\pi}{2}$, okolo osi x_1, získame os rotácie z_2 druhého kĺbu {2}, ktorého parametre a_2, d_3 sú opäť nulové.
- Na získanie osi z_3 by bolo potrebné vykonať rotáciu okolo osi y_2, čo ale DH konvencia neumožní, preto zavedieme parameter ofset $o_2 = -\frac{\pi}{2}$ pre druhý kĺb, ktorý nám umožní dosiahnuť zero angle konfiguráciu a získanie ďalších DH parametrov. Ofsetom súradnicového systému {2} získame os rotácie z_3 tretieho kĺbu, pomocou rotácie okolo osi x_2 o uhol $\alpha_2 = -\frac{\pi}{2}$. Po vykonaní tejto rotácie, počiatok súradnicového systému tretieho kĺbu získame posunom po osi z_2 o $d_3 = UpperArmLength$. Posledná konštanta a_3 pre tretí kĺb je opäť nulová.
- Os z_4 posledného kĺbu dostaneme po otočení súradnicového systému {3} okolo osi x_3 o uhol $\alpha_3 = -\frac{\pi}{2}$. Ostatné konštanty a_3, d_4 sú nulové.

- Na záver nám zostáva napraviť orientáciu súradnicového systému {4}, aby zodpovedalo orientácie systému koncového efektora, pomocou rotácie okolo osi z_4 o uhol $\frac{\pi}{2}$. Poslednou transláciou po novovzniknutej osi x_5 získame finálnu polohu súradnicového systému {tool}, ktorý reprezentuje koncový efektor kinematického modelu ľavej ruky robota NAO.

Získané DH parametre zapisujeme do tabuľky Tab. 1.

Tab. 1 Denavit-Hartenbergove parametre kinematického modelu ľavej ruky robota NAO

Číslo kĺbu	**Natočenie ramena**	**Dĺžka ramena**	**Vysunutie kĺbu**	**Uhol v kĺbe**	**Ofset**
i	α_{i-1}	a_{i-1}	d_i	θ_i	o_i
1.	$-\frac{\pi}{2}$	0	0	θ_1	0
2.	$\frac{\pi}{2}$	0	0	θ_2	$-\frac{\pi}{2}$
3.	$-\frac{\pi}{2}$	0	UpperArmLength	θ_3	0
4.	$\frac{\pi}{2}$	0	0	θ_4	0
$T_{base}^{O} = Trans_y(ShoulderOffsetY + ElbowOffsetY) \cdot Trans_z(ShoulderOffsetZ)$					
$T_{tool}^{4} = Rot_z\left(\frac{\pi}{2}\right) \cdot Trans_x(HandOffsetX + LowerArmLength)$					

Jednotlivé hodnoty konštánt sa nachádzajú v tabuľke Tab. A. v prílohe

1.b. Vytváranie simulačného modelu pomocou Robotics Toolboxu v prostredí MATLAB

Pomocou hodnôt z tabuľky Tab. 1. vytvoríme simulačný model pre ľavú ruku robota NAO v prostredí MATLAB s využitím Robotics Toolboxu, ako to je vo funkcii **leftarm_mld()**

```
function leftarm=leftarm_mdl()
params;
leftarm = SerialLink( [
    Revolute('d', 0, 'alpha', -pi/2, 'a', 0, 'modified')
    Revolute('d', 0, 'alpha', pi/2,  'a', 0, 'offset', -pi/2, 'modified')
    Revolute('d', UpperArmLength, 'alpha', -pi/2, 'a', 0, 'modified')
    Revolute('d', 0, 'alpha', pi/2,  'a', 0, 'modified')
    ], ...
    'base', transl(0, ShoulderOffsetY+ElbowOffsetY, ShoulderOffsetZ), ...
    'tool', trotz(pi/2)*transl(HandOffsetX+LowerArmLength, 0, 0), ...
    'name', 'left arm');
end
```

Volaním tejto funkcie vygenerujeme SerialLink objekt pre kinematický model ruky, na ktorom následne vieme volať funkcie toolboxu na výpočet priamej či inverznej kinematiky. Tento objekt je v prostredí MATLABU reprezentované nasledovne

```
Command Window
>> leftarm = leftarm_mdl()

leftarm =

left arm:: 4 axis, RRRR, modDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          0|          0|    -1.5708|          0|
|  2|         q2|          0|          0|     1.5708|    -1.5708|
|  3|         q3|        105|          0|    -1.5708|          0|
|  4|         q4|          0|          0|     1.5708|          0|
+---+-----------+-----------+-----------+-----------+-----------+
base:    t = (0, 113, 100), RPY/xyz = (0, 0, 0) deg
tool:    t = (0, 114, 0), RPY/xyz = (90, 0, 0) deg
```

Obr.2. SerialLink objekt kinematického modelu ľavej ruky robota NAO v MATLABe

Úloha č.2. : Postup pre riešenie úlohy priamej kinematiky pre odvodený model

Na základe odvodených DH parametrov kinematického modelu, uvedených v tabuľke Tab.1., zostavíme homogénne transformačné matice, zodpovedajúce transformáciám medzi jednotlivými kĺbmi kinematického reťazca ruky. V nasledujúcich maticiach zavedieme skratky pre goniometrické funkcie $\sin(\theta_i) = s_{\theta_i}$ a $\cos(\theta_i) = c_{\theta_i}$.

$$T_1^{base} = \begin{pmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_{\theta_1} & -c_{\theta_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T_2^1 = \begin{pmatrix} c_{o_2+\theta_2} & -s_{o_2+\theta_2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{o_2+\theta_2} & c_{o_2+\theta_2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$T_3^2 = \begin{pmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ -s_{\theta_3} & -c_{\theta_3} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T_4^3 = \begin{pmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{\theta_4} & c_{\theta_4} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rovnakým spôsobom získame dodatočné transformačné matice slúžiace na popis základného rámca a koncového efektora, v ktorých skratky sú iniciálovými abreviáciami konštánt z tabuľky Tab.1.

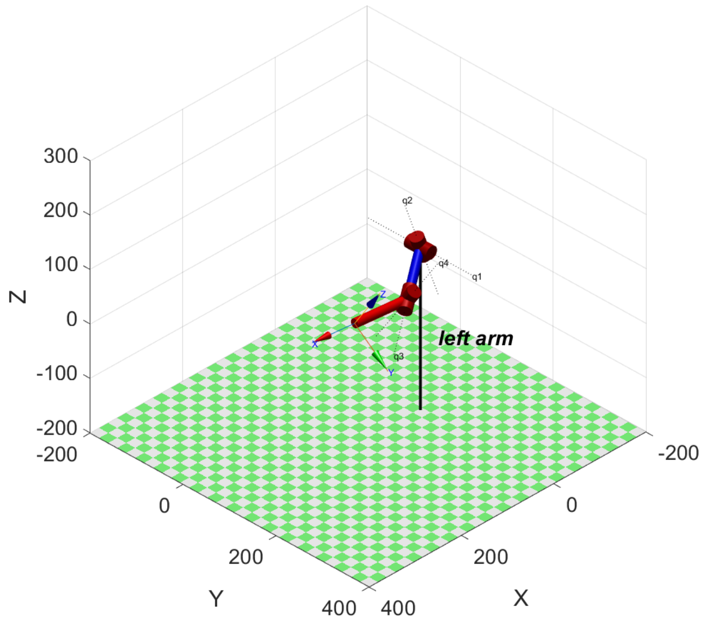
$$T_{base}^{O} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -EOY-SOY \\ 0 & 0 & 1 & SOZ \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T_{tool}^{4} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -HOX-LAL \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Vynásobením týchto matíc v správnom poradí a dosadením požadovaných hodnôt kĺbových premenných θ_i získame riešenie úlohy priamej kinematiky pre model ľavej ruky robota.

Algoritmické riešenie spočíva opäť v zostavení uvedených matíc a ich vynásobení ako môžeme vidieť vo fuinkcii **nao_leftarm_myfkine(q)**, kde vstupný parameter **q** zodpovedá vektoru hodnôt kĺbových premenných v stupňoch.

```
function T = nao_leftarm_myfkine(q)
params;
q = q.*deg;
T_base_0 = transl(0,ShoulderOffsetY+ElbowOffsetY,ShoulderOffsetZ);
T_1_base = my_HT_matrix(-pi/2,0,0,q(1),0);
T_2_1 = my_HT_matrix(pi/2,0,0,q(2),-pi/2);
T_3_2 = my_HT_matrix(-pi/2,0,UpperArmLength,q(3),0);
T_4_3 = my_HT_matrix(pi/2,0,0,q(4),0);
T_tool_4 = trotz(pi/2)*transl(HandOffsetX+LowerArmLength,0,0);
T = T_base_0*T_1_base*T_2_1*T_3_2*T_4_3*T_tool_4;
end
```

Funkcia vráti homogénnu transformačnú maticu **T**, ktorá je riešením úlohy priamej kinematiky pre tento model. Na obrázku Obr.3. je zobrazený model ľavej ruky robota NAO s využitím funkcie **plot** Robotics Toolboxu, pri hodnotách kĺbových premenných $\boldsymbol{q} = (20°, 32°, -40°, -40°)$ ako ilustrácia riešenia priamej kinematiky pre tieto hodnoty.



Obr.3. Kinematický model ľavej ruky robota NAO v Robotics Toolboxe

Úloha č.3. : Postup pre riešenie úlohy inverznej kinematiky pre odvodený model

Pri odvodení analytickej metódy riešenia inverznej kinematiky využijeme inverzné goniometrické funkcie a postupné oddelenie kĺbov od celého reťazca. Tento postup spočíva v analýze transformačných matíc a na základe nich v zostavení rovníc pre výpočet hodnôt kĺbových premenných.

Prvým krokom je odstránenie známych a konštantných transformácii z reťazca, ktorými sú matice pre základný rámec a koncový efektor

$$T_4^{base} = (T_{base}^{O})^{-1} \cdot T_{tool}^{O} \cdot (T_{tool}^{4})^{-1}$$

A tak dostávame maticu opisujúcu časť ramena čisto s kĺbmi

$$T_4^{base} = \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} \\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$t_{1,1} = s_{(o_2-\theta_2)} c_{\theta_1} s_{\theta_4} - c_{\theta_4}(s_{\theta_1} s_{\theta_3} - c_{(o_2-\theta_2)} c_{\theta_1} c_{\theta_3})$$
$$t_{1,2} = s_{\theta_4}(s_{\theta_1} s_{\theta_3} - c_{(o_2-\theta_2)} c_{\theta_1} c_{\theta_3}) + s_{(o_2-\theta_2)} c_{\theta_1} c_{\theta_4}$$
$$t_{1,3} = c_{\theta_3} s_{\theta_1} + c_{(o_2-\theta_2)} c_{\theta_1} s_{\theta_3}$$
$$t_{1,4} = d_3 s_{(o_2-\theta_2)} c_{\theta_1}$$
$$t_{2,1} = c_{(o_2-\theta_2)} s_{\theta_4} - s_{(o_2-\theta_2)} c_{\theta_3} c_{\theta_4}$$
$$t_{2,2} = c_{(o_2-\theta_2)} c_{\theta_4} + s_{(o_2-\theta_2)} c_{\theta_3} s_{\theta_4}$$
$$t_{2,3} = -s_{(o_2-\theta_2)} s_{\theta_3}$$
$$t_{2,4} = d_3 c_{(o_2-\theta_2)}$$
$$t_{3,1} = -c_{\theta_4}(c_{\theta_1} s_{\theta_3} + c_{(o_2-\theta_2)} c_{\theta_3} s_{\theta_1}) - s_{(o_2-\theta_2)} s_{\theta_1} s_{\theta_4}$$
$$t_{3,2} = s_{\theta_4}(c_{\theta_1} s_{\theta_3} + c_{(o_2-\theta_2)} c_{\theta_3} s_{\theta_1}) - s_{(o_2-\theta_2)} c_{\theta_4} s_{\theta_1}$$
$$t_{3,3} = c_{\theta_1} c_{\theta_3} - c_{(o_2-\theta_2)} s_{\theta_1} s_{\theta_3}$$
$$t_{3,4} = -d_3 s_{(o_2-\theta_2)} s_{\theta_1}$$

Skúmaním prvkov matice vidíme, že pomocou $t_{1,4}$ a $t_{3,4}$ vieme napísať vzťah na výpočet prvej kĺbovej premennej θ_1 , s využitím inverznej trigonometrickej funkcie tan^{-1}

$$\theta_1 = \tan^{-1}\left(\frac{-t_{3,4}}{t_{1,4}}\right) = \tan^{-1}\left(\frac{-(-d_3 s_{(o_2-\theta_2)} s_{\theta_1})}{d_3 s_{(o_2-\theta_2)} c_{\theta_1}}\right)$$

Kde parameter d_3 je *UpperArmLenght* z tabuľky Tab.1. ako to bolo v prípade priamej kinematiky pre tento model. Po získaní hodnoty θ_1 daný kĺb odstránime z reťazca

$$T_4^1 = (T_1^{base})^{-1} \cdot T_4^{base}$$

a dostávame transformačnú maticu pre zostávajúce kĺby.

$$T_4^1 = \begin{pmatrix} s_{(o_2-\theta_2)}s_{\theta_4} + c_{(o_2-\theta_2)}c_{\theta_3}c_{\theta_4} & s_{(o_2-\theta_2)}c_{\theta_4} - c_{(o_2-\theta_2)}c_{\theta_3}s_{\theta_4} & c_{(o_2-\theta_2)}s_{\theta_3} & d_3 s_{(o_2-\theta_2)} \\ c_{\theta_4}s_{\theta_3} & -s_{\theta_3}s_{\theta_4} & -c_{\theta_3} & 0 \\ c_{(o_2-\theta_2)}s_{\theta_4} - s_{(o_2-\theta_2)}c_{\theta_3}c_{\theta_4} & c_{(o_2-\theta_2)}c_{\theta_4} + s_{(o_2-\theta_2)}c_{\theta_3}s_{\theta_4} & -s_{(o_2-\theta_2)}s_{\theta_3} & d_3 c_{(o_2-\theta_2)} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

V tomto vzťahu vidíme, že hodnotu druhej kĺbovej premennej θ_2 vieme určiť pomocou prvkov matice $t_{1,4}$, $t_{3,4}$, teda

$$\theta_2 = \tan^{-1}\left(\frac{t_{1,4}}{t_{3,4}}\right) = \tan^{-1}\left(\frac{d_3 s_{(o_2-\theta_2)}}{d_3 c_{(o_2-\theta_2)}}\right)$$

Zavádzaním parametra ofset o_2, pri riešení problému priamej kinematiky však prináša dve riešenia pre hodnoty kĺbovej premennej θ_2, z ktorých práve jedno je správne. Vzhľadom na to, že hodnoty predchádzajúcich kĺbových premenných ovplyvnia tie nasledujúce, pre nájdenie tých správnych hodnôt je potrebná validácia s využitím priamej kinematiky na získaných kĺbových premenných. Po získaní správnej hodnoty θ_2 tento kĺb opäť odstránime z reťazca

$$T_4^2 = (T_2^1)^{-1} \cdot T_4^1 = \begin{pmatrix} c_{\theta_3}c_{\theta_4} & -c_{\theta_3}s_{\theta_4} & s_{\theta_3} & 0 \\ s_{\theta_4} & c_{\theta_4} & 0 & d_3 \\ -c_{\theta_4}s_{\theta_3} & s_{\theta_3}s_{\theta_4} & c_{\theta_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Pomocou prvkov $t_{1,3}$ a $t_{3,2}$ určíme hodnotu tretej kĺbovej premennej podľa vzťahu

$$\theta_3 = \tan^{-1}\left(\frac{t_{1,3}}{t_{3,3}}\right) = \tan^{-1}\left(\frac{s_{\theta_3}}{c_{\theta_3}}\right)$$

Na záver odstránime z reťazca aj tretí kĺb

$$T_4^3 = (T_3^2)^{-1} \cdot T_4^2 = \begin{pmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{\theta_4} & c_{\theta_4} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

a zo zostávajúceho vzťahu vypočítame hodnotu poslednej kĺbovej premennej

$$\theta_4 = \tan^{-1}\left(\frac{t_{3,1}}{t_{1,1}}\right) = \tan^{-1}\left(\frac{s_{\theta_4}}{c_{\theta_4}}\right)$$

Tieto vzťahy na analytický výpočet hodnôt kĺbových premenných môžeme programovo implementovať v prostredí MATLAB, ako to je aj vo funkcii **nao_leftarm_analytikine(T)**, ktorá ako vstupný parameter akceptuje homogénnu transformačnú maticu vyjadrujúcu požadovanú polohu a orientáciu koncového efektora ľavej ruky.

```
function [q,e] = nao_leftarm_analytikine(T)
q = [0,0,0,0];
e = "Joint values from analytical solver";
params;

%calculating known transformations for base and tool
T_base_0 = transl(0,ShoulderOffsetY+ElbowOffsetY,ShoulderOffsetZ);
T_tool_4 = trotz(pi/2)*transl(HandOffsetX+LowerArmLength,0,0);

%removing known transformations
T_4_base = inv(T_base_0)*T*inv(T_tool_4);

% calculating q1 and checking for joint limits
q(1) = atan(-T_4_base(3,4)/T_4_base(1,4));
if (q(1) < LShoulderPitch(1)) || (q(1) > LShoulderPitch(2))
    e = "Joint values exceed joint limits, try another target T";
    return;
end

%removing first joint from the chain (q1 is known)
T_1_base = my_HT_matrix(-pi/2,0,0,q(1),0);
T_4_1 = inv(T_1_base)*T_4_base;

%because of offset, need to make some extra calculations with q2
q2temp = atan(T_4_1(1,4)/T_4_1(3,4));

% when q2temp is > 0
if q2temp > 0
    q2temp = -q2temp;
    q(2) = q2temp + pi/2;
else
    %when q2temp is < 0
    q2temp = q2temp + pi/2;
    q(2) = -q2temp;
end

%checking q2 for joint limits
if (q(2) < LShoulderRoll(1)) || (q(2) > LShoulderRoll(2))
    e = "Joint values exceed joint limits, try another target T";
    return;
end


%removing second joint from chain (q2 is known)
T_2_1 = my_HT_matrix(pi/2,0,0,q(2),-pi/2);
T_4_2 = inv(T_2_1)*T_4_1;

% calculating q3 and checking for joint limits
q(3) = atan(T_4_2(1,3)/T_4_2(3,3));
if (q(3) < LElbowYaw(1)) || (q(3) > LElbowYaw(2))
    e = "Joint values exceed joint limits, try another target T";
    return;
end

%removing third joint from the chain (q3 is known)
T_3_2 = my_HT_matrix(-pi/2,0,UpperArmLength,q(3),0);
T_4_3 = inv(T_3_2)*T_4_2;

% calculating q4 and checking for joint limits
q(4) = atan(T_4_3(3,1)/T_4_3(1,1));
if (q(4) < LElbowRoll(1)) || (q(4) > LElbowRoll(2))
    e = "Joint values exceed joint limits, try another target T";
    return;
end

end
```

Funkcia vracia vektor **q**, ktorý zodpovedá hodnotám kĺbových premenných ako riešenie úlohy inverznej kinematiky pre model ľavej ruky pri zadanej požadovanej polohe a orientácie koncového efektora. Taktiež vráti textový reťazec, ktorý obsahuje dodatočné informácie, či získané riešenie je validné alebo nebolo možné určiť hodnoty kĺbových premenných pre požadovaný vstup.

Podobným postupom dokážeme odvodiť kinematické modely pre ďalšie končatiny robota NAO, resp. pre analogické robotické systémy. Robotics Toolbox nám umožní zadefinovať nielen robotické manipulátory ale aj mobilné roboty získať pre nich riešenia úlohy priamej a inverznej kinematiky a získané výsledky vhodným spôsobom ilustrovať.

Príloha

Tab. A. Dĺžky ramien a ofsety pre robot NAO v mm

Názov	**Dĺžka v mm**	**Názov**	**Dĺžka v mm**
NeckOffsetZ	126.50	HipOffsetZ	85.00
ShoulderOffsetY	98.00	HipOffsetY	50.00
ElbowOffsetY	15.00	ThighLength	100.00
UpperArmLength	105.00	TibiaLength	102.90
LowerArmLength	55.95	FootHeight	45.19
ShoulderOffsetZ	100.00	HandOffsetZ	12,31
HandOffsetX	57.75		