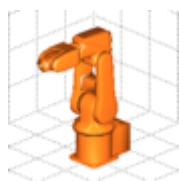


## Vizualizácia robotických ramien v MATLAB-e

Lešo Martin · MATLAB/Comsol

27.06.2016



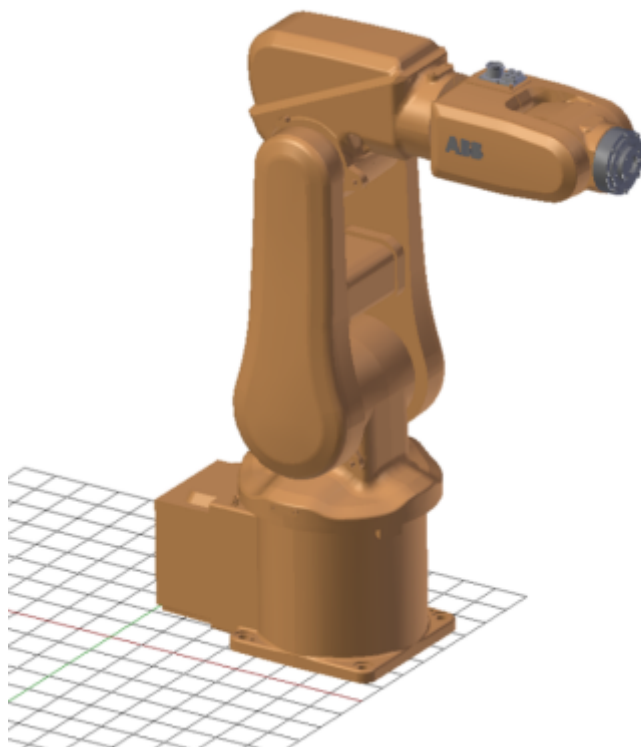
Článok sa zaoberá procesom vytvorenia detailnej 3D vizualizácie robotického ramena prostredníctvom patch objektov, venuje sa vytvoreniu vizualizácie robotického ramena v externom editore, importovaniu modelu robotického ramena do programu MATLAB a jeho následnému zobrazeniu. V príspevku sú opísané rôzne metódy pre docielenie čo najlepšej vizualizácie robotických ramien pomocou nastavenia osvetlenia a materiálov robotického ramena. Záver článku sa zaoberá nastavením kamery pre grafickú scénu a nastavením požadovanej veľkosti zobrazenia vizualizovanej scény.

### 1. Úvod

V oblasti robotiky existuje mnoho programov, ktoré umožňujú simuláciu modelov robotického ramena prepojeného s jeho vizualizáciou, pričom mnohé z nich neobsahujú možnosť zasahovať do ich matematického opisu, alebo sú tvorené len jednoduchou 2D vizualizáciou, resp. schematickou 3D vizualizáciou [1-5]. V rámci tohto príspevku je prezentovaná metóda pre vizualizáciu ľubovoľných modelov robotických ramien v programe MATLAB, ktorá je založená na polygónových modeloch vytvorených funkciou patch, pričom jednotlivé modely robotických ramien boli vyexportované z programu Blender [5] do OBJ typu súboru a následne importované do programu MATLAB za použitia funkcií read\_obj a draw\_objects, ktoré boli podrobne opísané v [6]. Tento príspevok nadväzuje na [6], pričom je vysvetľuje opis tvorby modelu robotického ramena, jeho potrebné úpravy pred exportom a jeho následné zobrazenie v programe MATLAB, v rámci ktorého je zahrnuté i nastavenie scény, ako aj kontrola nad jej osvetlením a ovládanie kamery v rámci vizualizovanej scény.

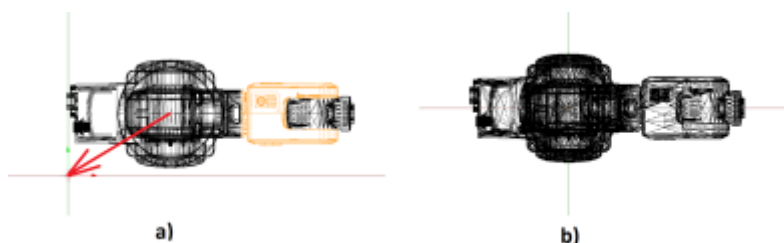
### 2. Tvorba modelu robotického ramena

Pre tvorbu vizuálneho modelu je možné použiť ľubovoľný editor, pričom podmienkou je možnosť exportu vizualizácie objektov vo forme OBJ typu súboru. V našom prípade bol ako primárny 3D editor použitý program Blender. Pre importovanie detailných 3D vizualizácií do programu MATLAB je potrebné vytvoriť 3D model robotického ramena. V rámci tohto článku uvažujeme vytvorenie vizualizácie robotického ramena ABB - IRB 120, ktorého model je dostupný [7] v rôznych CAD formátoch a jeho vizualizácia je zobrazená na obr. 1.



Obr.1 Vizualizácia robotického ramena v programe Blender

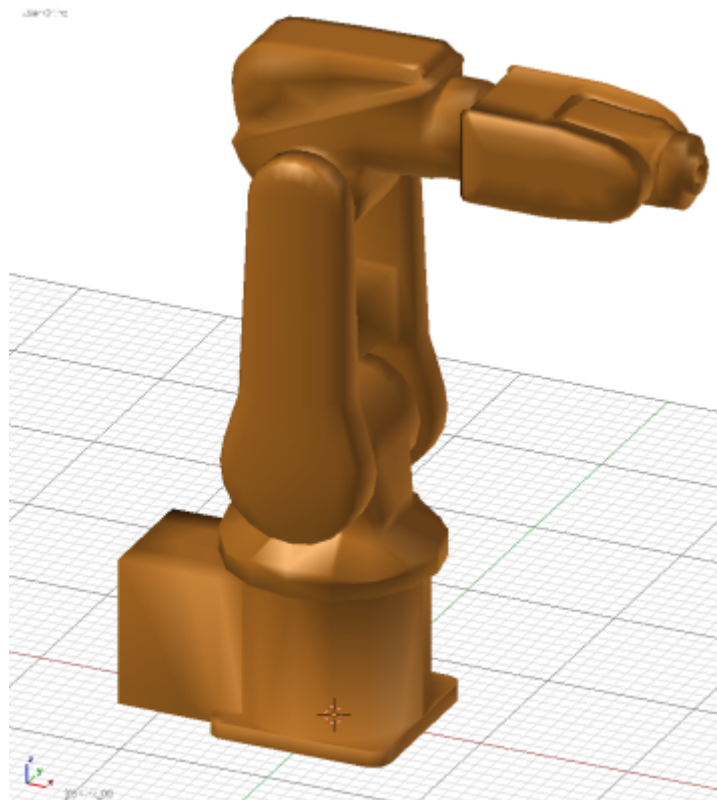
Pre importovanie vizualizácie do programu Blender bol stiahnutý model vytvorený v programe Google Sketchup, z ktorého bol model exportovaný do OBJ typu súboru, ktorý bol následne v editore Blender upravený tak, aby celkové členenie objektu tvorilo 7 častí (základňa a 6 článkov robotického ramena), ktoré boli pomenované ako Link00 až Link06. Takéto rozčlenenie modelu robotického ramena je vhodné z dôvodu prehľadnosti jednotlivých častí modelu robotického ramena, ako aj budúcej vizualizácie robotického ramena pre rôzne nastavenie natočenia servopohonov robotického ramena. V prípade uvažovania budúcej zmeny polohy robotického ramena v programe MATLAB je tiež vhodné posunúť model robotického ramena do nulového bodu (bolo realizované posunutím kurzoru do nulového bodu a presunutie pivotu modelu na kurzor klávesovou skratkou Shift+Alt+Ctrl+c). Polygóny tvoriace jednotlivé časti boli následne konvertované na trojuholníkový tvar klávesovou skratkou Ctrl+t v editačnom móde. Takto vytvorený model je po vyexportovaní do OBJ j formátu vhodný pre importovanie do MATLAB-u a jeho následnú vizualizáciu (obr. 2).



Obr.2 Úprava modelu robotického ramena v programe Blender, a) pred úpravou b) po úprave

Druhou komplikovanejšou možnosťou prípravy modelu pre jeho export je vymodelovanie robotického ramena priamo v programe Blender, čím je možné presne definovať pozíciu jednotlivých polygónov a ich bodov. Nevýhodou takéhoto riešenia je

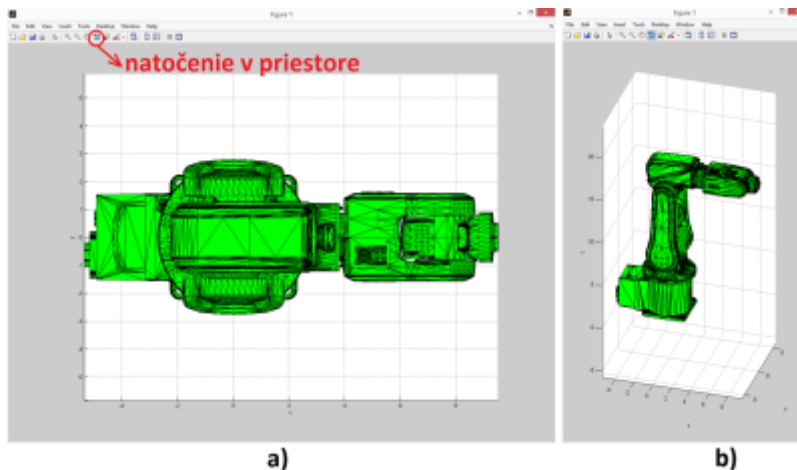
jeho náročnosť, pričom za výhodu je možné považovať vytvorenie modelu presne podľa určených požiadaviek a tým vypustiť niektoré detaily s cieľom redukcie použitých bodov a polygónov, čím je možné urýchliť budúci prepočet bodov pri animovaní robotických ramien. Vytvorené robotické rameno je zobrazené na obr. 3 a skladá sa z 1813 bodov, pričom importované robotické rameno z programu Google Sketchup je zložené z 31 150 bodov (obr. 1). Zjednodušením modelu sme dokázali približne 17 násobne urýchliť prepočet bodov pre rôzne polohy robotického ramena.



Obr.3 Zjednodušený model robotického ramena

### 3. Vizualizácia robotických ramien v programe MATLAB

Táto kapitola je zameraná na celkovú vizualizáciu importovaných robotických ramien v programe MATLAB, pričom kladie dôraz na dosiahnutie maximálnej kvality zobrazenia robotických ramien, ako aj na samotné nastavenie snímania kamery grafickej scény a jej osvetlenie. Po importovaní a zobrazení robotického ramena je možné dosiahnuť vizualizáciu zobrazenú na obr. 4, pričom pomocou nástroja rotate 3D vieme zmeniť pohľad kamery na scénu (vyznačený červenou farbou na obr. 4a).



Obr.4 Vizualizácia robotického ramena a) pohľad zhora b) pohľad z boku [6]

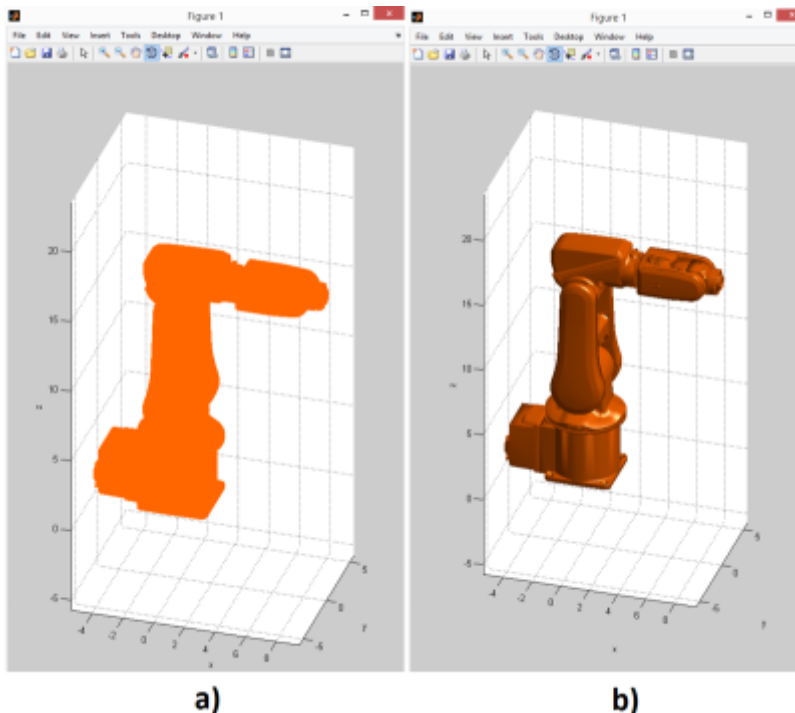
### 3.1 Nastavenie farby materiálu

Ako je možné vidieť na obr.4, rušivým elementom pri takejto vizualizácii sú čierne okraje polygónov. Pre odstránenie tohto zobrazenia je možné postupovať dvoma spôsobmi a to nastavením priehľadnej farby pre jednotlivé hrany polygónov (Edges) alebo nastavením totožnej farby polygónov a hrán. V nasledujúcom príklade je odstránenie čiernej farby hrán polygónov prevádzané spriesvitnením hrán. Takéto riešenie je výhodné z dôvodu, že pri požadovanej zmene farby polygónov robotického ramena nie je potrebné aplikovať zmenu farby aj na hrany polygónov.

Ďalšou často kladenou požiadavkou pri vizualizácii objektov je zmena ich farby. Štandardne sa presné nastavenie farby prevádza definovaním jej červenej, modrej a zelenej zložky v rámci jedného vektora, pričom tieto hodnoty sú definované v rozmedzí od 0 po 1. Vzhľadom na to, že väčšinou sú farby reprezentované trojicou 8 bitových čísel (rozmedzie 0 až 255 v desiatkovej sústave, resp. 0x00 až 0xFF v hexa sústave) dostaneme požadovanú farbu v rámci programu MATLAB po predelení týchto čísel číslom 255 pre desiatkovú sústavu. Import modelu, jeho vykreslenie, nastavenie priehľadnosti jednotlivých hrán a nastavenie farby polygónov je možné realizovať nasledovným spôsobom, pričom tieto nastavenia sú aplikované pre všetky individuálne objekty tvoriace vizualizáciu robotického ramena.

```
% Import vizualizácie
objects = read_obj();
% Zobrazenie vizualizácie
handles = draw_objects(objects);
for i = 1: numel(handles.obj_id)
    % Nastavenie priehľadnosti hrán
    set(handles.obj_id(i), 'Edgealpha', 0)
    % Nastavenie farby
    set(handles.obj_id(i), 'Facecolor', ...
        [255/255 102/255 0/255])
end
```

Po aplikovaní tohto kódu je vytvorená vizualizácia zobrazená na obr. 5 a)



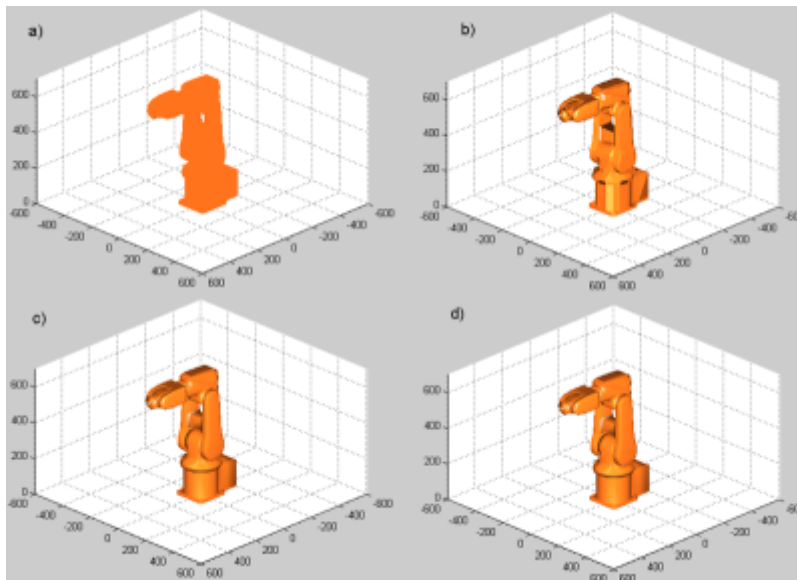
Obr.5 Vplyv osvetlenia na vizualizáciu a) bez osvetlenia b) s osvetlením

### 3.2 Nastavenie osvetlenia vizualizovanej scény

Ako je znázornené na obr. 5 a), odstránením hrán polygónov sa úplne stratila informácia o hĺbke obrazu. Pre zlepšenie výsledku je nutné vytvoriť osvetlenie scény. Najjednoduchší spôsob pridania osvetlenia je príkazom `camlight`, ktorý pridá osvetlenie smerom od kamery. Osvetlená vizualizácia robotického ramena je znázornená na obr. 5b), pričom bol zvolený typ algoritmu osvetlenia `phong`. Osvetlenie scény bolo vykonané nasledujúcim spôsobom:

```
camlight
lighting phong
```

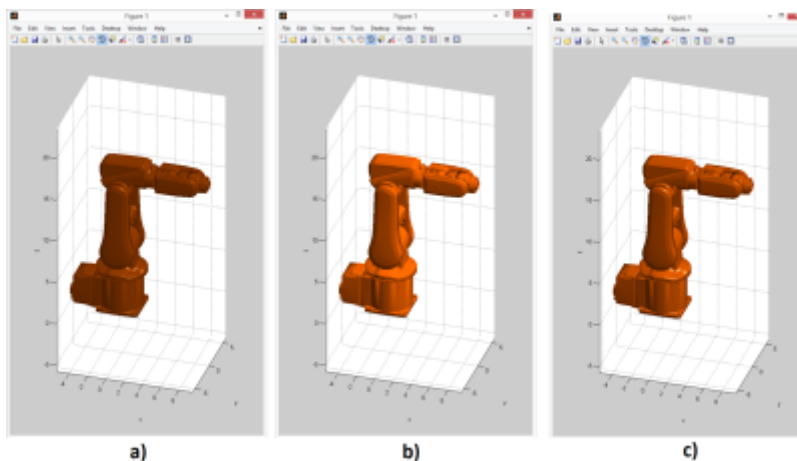
Program MATLAB má definované tri základné možnosti typu osvetlenia, ktoré sa nastavujú funkciou `lighting`. Osvetlenie typu `flat` vytvára jednotné osvetlenie pre každý polygón objektu. Osvetlenie typu `goraud` vypočítava normály bodov a interpoluje ich lineárne pre každý polygón objektu. Osvetlenie typu `phong` interpoluje normály bodov pre všetky polygóny objektu a počíta odlesk pre všetky pixely obrazu. Tento typ osvetlenia vo všeobecnosti vytvára najlepšie výsledky, ale zároveň patrí k najpomalšiemu typu osvetlenia. V prípade vypnutia osvetlenia `lighting('none')` sa objekt vykreslí jednofarebne. Porovnanie jednotlivých typov osvetlení je zobrazené na obr. 6. [11]



Obr.6 Rôzne typy osvetlenia a) none, b) flat, c) gouraud, d) phong

### 3.3 Nastavenie typu materiálu robotického ramena

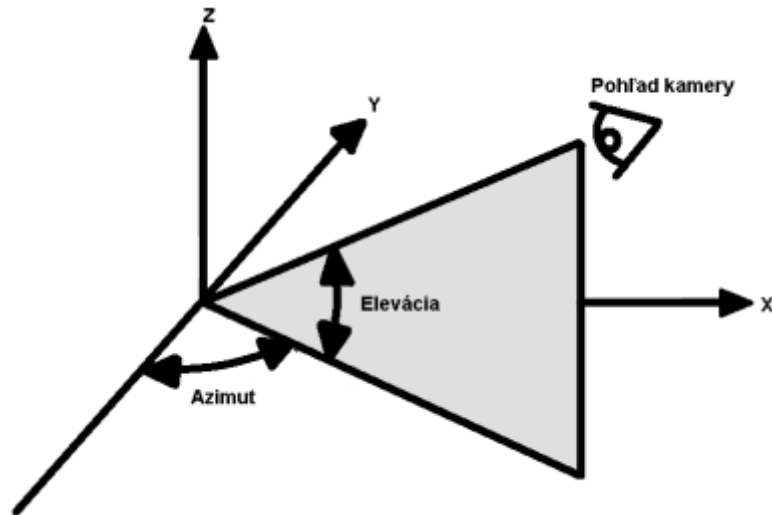
Priamo s nastavením osvetlenia súvisí aj nastavenie materiálu objektu. V programe MATLAB sú definované tri všeobecné typy materiálov, a to shiny - lesklý, metal - kovový, dull - matný materiál. Tieto predvolené typy materiálov združujú tri argumenty funkcie material, prvý argument určuje mieru vplyvu farby materiálu z okolia, druhý určuje mieru vplyvu samotnej farby polygónov objektu, posledný tretí argument určuje mieru odlesku materiálu [9]. Zobrazenie robotického ramena pri nastavení troch typov materiálov je uvedené na obr. 7.



Obr.7 Nastavenie materiálu robotického ramena a) metal b) shiny c) dull

### 3.4 Nastavenie kamery a veľkosti vizualizovanej scény

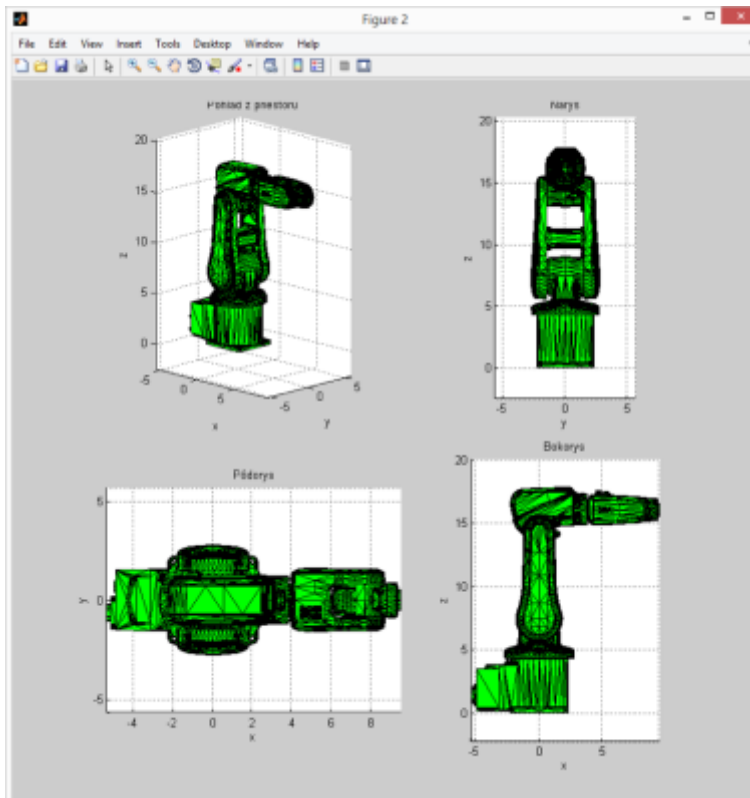
Pre zobrazenie grafov a 3D objektov je v rámci každého grafického okna definovaná kamera. Kamera je všeobecne definovaná jej polohou a smerovým vektorom. Funkcia campos(pozícia) slúži na určenie polohy kamery. Na zmenu orientácie kamery je možné použiť jednu z trojice funkcií. Funkcia camup(vektor) nastavuje priamo smerový vektor snímania kamery. Príkazom camtarget(pozícia) sa nastavuje bod, na ktorý je kamera natočená. Poslednou funkciou pre nastavenie orientácie kamery je funkcia view(azimut, elevácia), ktorá nastavuje vektor snímania kamery podľa obr. 8. Kompletný výpis funkcií pre prácu s kamerou je uvedený v [10].



Obr.8 Funkcia view

Medzi najčastejšie zobrazované pohľady v rámci technických prác patrí pôdorys (pohľad zhora), bokorys (pohľad z boku) a nárys (pohľad spredu). Pre vytvorenie týchto rôznych pohľadov môžeme použiť nasledovný kód, pričom funkcia subplot slúži na rozčlenenie grafického okna na viac častí:

```
rob = read_obj();
subplot(2,2,1)
draw_objects(rob, gca);
view(45,15)
title('Pohľad z priestoru')
subplot(2,2,2)
draw_objects(rob, gca);
view(90,0)
title('Nárys')
subplot(2,2,3)
draw_objects(rob, gca);
view(0,90)
title('Pôdorys')
subplot(2,2,4)
draw_objects(rob, gca);
view(0,0)
title('Bokorys')
```

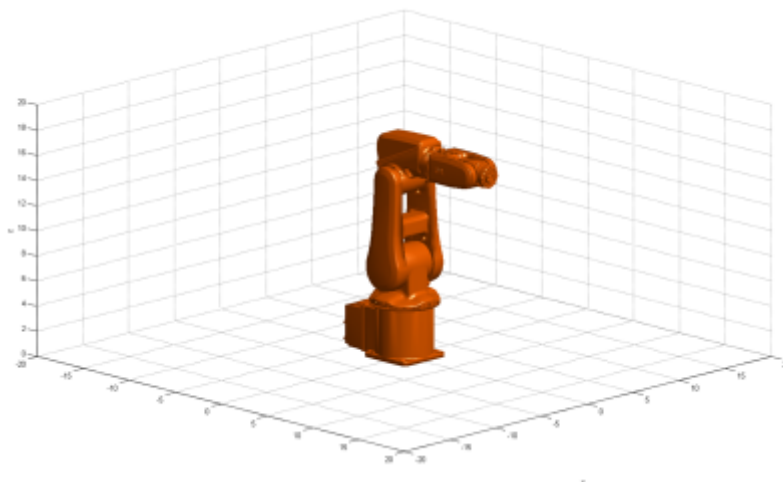


Obr.9 Zmena pohľadu kamery funkciou view

Takto vytvorená vizualizácia je uvedená na obr. 9. Okrem nastavenia samotnej vizualizácie objektov scény je častokrát potrebné nastavenie veľkosti scény, ktoré sa prevádza funkciami xlim, ylim a zlim podobne ako pri nastavení 2D grafov. Vytvorenie výslednej vizualizácie robotického ramena je možné vykonať nasledujúcim spôsobom:

```
rob = read_obj();
handles = draw_objects(rob)
view(45,15)
xlim([-20, 20])
ylim([-20, 20])
zlim([0, 20])
camlight;
lighting phong
for i = 1: numel(handles.obj_id)
    set(handles.obj_id(i), 'Edgealpha', 0)
    set(handles.obj_id(i), 'Facecolor', ...
        [255/255 102/255 0/255])
end
```





Obr.10 Výsledná vizualizácia robotického ramena

#### 4. Záver

Článok sa zaoberá 3D vizualizáciou robotického ramena ABB-IRB120 v programe MATLAB, pričom v prvej časti príspevku boli odvodené dva postupy prípravy modelu robotického ramena pre importovanie modelov do programu MATLAB. Druhá časť príspevku sa zaoberala samotnou vizualizáciou robotického ramena v programe MATLAB, pričom boli uvedené spôsoby a možnosti zmeny farby robotického ramena ako aj nastavenie typu materiálu. V záverečnej časti príspevku boli opísané možnosti a spôsoby nastavenia typu osvetlenia a nastavenia kamery pre snímanie obrazu, ktoré sú potrebné pre dotvorenie výslednej vizualizácie robotického ramena, ktorá je v závere príspevku uvedená.

#### Zoznam použitej literatúry

1. Gonzalez F., Zalewski J.: A robotic arm simulator software tool for use in introductory robotics courses," 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 861-866.
2. Nayar H. D.: Robotect: serial-link manipulator design software for modeling, visualization and performance analysis, Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on, 2002, pp. 1359-1364 vol.3.
3. Mohammed Abu Qassem, Abuhadrous I., Elaydi H.: Modeling and Simulation of 5 DOF educational robot arm, Advanced Computer Control (ICACC), 2010 2nd International Conference on, Shenyang, 2010, pp. 569-574.
4. Corke P.: Robotics Toolbox for MATLAB, ONLINE:  
[http://petercorke.com/Robotics\\_Toolbox.html](http://petercorke.com/Robotics_Toolbox.html)
5. Blender program. Online:  
<https://www.blender.org/>
6. Lešo M., Žilková J.: Pokročilá 3D vizualizácia v MATLAB-e, Posterus, 2016, Online:  
<http://www.posterus.sk/?p=18583>
7. ABB. Online:  
<http://new.abb.com/products/robotics/industrial-robots/irb-120/irb-120-cad>
8. Mathworks documentation. Online:  
<http://www.mathworks.com/help/matlab/ref/lighting.html>
9. Mathworks documentation. Online:  
<http://www.mathworks.com/help/matlab/ref/material.html>

- 
10. Mathworks documentation. Online:  
<http://www.mathworks.com/help/matlab/views.html>
  11. Lešo, M.: Vizualizácia v robotike, Bakalárska práca, FEI TU v Košiciach, 2012

---

Spoluautorom článku je Jaroslava Žilková, Katedra elektrotechniky a mechatroniky, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Slovenská republika

---