

PRÍSPEVOK K TVORBE ROBOTIZOVANÉHO PRACOVISKA NA VIRTUÁLNEJ SCÉNE

CONTRIBUTION TO CREATING OF ROBOTIZED WORKPLACE IN VIRTUAL SCENE

Oliver MORAVČÍK, Pavol BOŽEK, Pavol BEZÁK,
Miriam IRINGOVÁ

*Autori: Prof. Dr. Ing. Oliver Moravčík, CSc., Ing. Pavol Božek, CSc., Ing. Pavol Bezák,
Ing. Miriam Iringová*

*Pracovisko: Katedra aplikovanej informatiky a automatizácie, Materiálovotechnologická
fakulta, STU*

Adresa: Hajdóczyho 1, 917 24 Trnava, SR

Tel.: 00421 33 544 77 34

*E-mail: moravcik@mtf.stuba.sk, bozek@mtf.stuba.sk, bezak@mtf.stuba.sk,
iringova@mtf.stuba.sk*

Abstract

Problematika robotov je v súčasnosti veľmi dynamicky sa rozvíjajúca oblasť strojárskych technológií. Je snaha o vývoj inteligentných robotov, ktorých vlastnosti a schopnosti sa zvyčajne overujú na modeli. Vzhľadom k dostatočnej výpočtovej kapacite súčasných bežne dostupných počítačov sa modelovanie a simulácia robotov dá elegantne uskutočniť vo virtuálnom prostredí. Ak chceme vytvoriť virtuálny model robota, je nutné si ujasniť voľbu geometrického modelu a grafickej knižnice.

Robot problematics at present is very dynamically developing area of machine engineering technologies. Intelligent robots are developed and their characteristics and capabilities are usually verified on model. Thanks to the sufficient computational capacity of actual normally accessible computers the process of robot modeling and simulation can be easily carried out in virtual environment. Before the process of virtual robot model creation it is necessary to make oneself clear the choice of geometric model and graphical library.

Key words

robot, robot virtuálny, prostredie virtuálne, model geometrický, model stenový

robot, virtual robot, virtual environment, geometric model, surface model

Úvod

Vzdelávací proces na univerzite technického zamerania si vyžaduje nadobudnutie určitých praktických skúseností, ktoré dopĺňajú teoretické poznatky a zjednodušujú pochopenie vyučovanej látky [2]. Vybavenie učebni však úplne nezodpovedá skutočným kapacitným požiadavkam. Takýmto prípadom je aj výučba programovania priemyselného robota. Možnosťou riešenia problémov s výučbou na finančne nedostupnom zariadení je vytvorenie zjednodušeného virtuálneho modelu robota, ktorý bude svojim rozhraním identický so skutočným priemyselným robotom a bude pracovať na rovnakom princípe. Z tohto dôvodu je venovaná pozornosť projektu vývoja virtuálneho priemyselného robota. Jeho jednotlivé algoritmy budú kopírovať chovanie skutočného priemyselného robota a budú poskytovať porovnateľné výstupy. Programovanie na tomto simulátore sa môže nahradiť programovaním na reálnom priemyselnom robote, resp. robotizovanom pracovisku. Nosnou myšlienkou projektu je vytvorenie virtuálneho modelu priemyselného robota aplikovaného v robotizovanej bunke na počítači. Priemyselný robot je pre svoju jednoduchosť operátorskej obsluhy ako aj jednoduchosť prístupu k jednotlivým užívateľským funkciám v jednotlivých režimoch výnimočne vhodný pre účely výučby novoakreditovaného predmetu „Roboty a manipulátory“, ale aj pre rôzne stupne vzdelávania.

Voľba geometrického modelu a grafickej knižnice

Z analýzy požiadaviek na virtuálny robot vyplýva, že najvhodnejším geometrickým modelom pre vyriešenie problému grafického znázornenia objektov vo virtuálnej scéne je použitie stenového modelu.

Stenový model neobsahuje informácie, pomocou ktorých by bolo možné implicitne identifikovať, či je bod v priestore súčasťou objemu alebo nie. Stenový objem je v klasickom ponímaní definovaný pomocou matice stien, matice hrán a matice vrcholov. Pri použití modelu opísaného pomocou jazyka VRML a pri úvahe, že topologické informácie o príslušnosti vrcholu k hrane nie sú pre jednoznačnú identifikovateľnosť a samotné vykreslenie potrebné, je možné túto úroveň abstrakcie v hierarchickej údajovej štruktúre vynechať. Pri samotnom vykresľovaní objektu na scénu je vhodné použiť vysokovýkonnú knižnicu OpenGL. Grafické algoritmy OpenGL sú optimalizované na maximálny výkon a pri použití hardvérovej podpory antialiasingu je možné získať vysokú úroveň podobnosti s reálnymi objektmi. Časť kódu na obr. 1 implementovaného v jazyku C++, ktorý používa príkazy knižnice OpenGL, vykreslí plochu – trojuholník ohraničený bodmi so súradnicami $[X_1, Y_1, Z_1]$, $[X_2, Y_2, Z_2]$, $[X_3, Y_3, Z_3]$.

```
glBegin(GL_TRIANGLES);  
    glVertex3f(X1, Y1, Z1);  
    glVertex3f(X2, Y2, Z2);  
    glVertex3f(X3, Y3, Z3);  
glEnd();
```

Obr. 1. Časť kódu v C++ vykresľujúca trojuholník

Rovnako je možné pomocou OpenGL vykresľovať aj iné plošné objekty, napr. pás trojuholníkov, obdĺžnik či mnohoúholník.

Výber formátu pre import objektov na scénu

Keďže objekty by mali byť znázorňované vo virtuálnej scéne, je vhodné použiť ako vstupný formát WRL, ktorý bol vytvorený pre potreby virtuálnej reality. Formát WRL obsahuje dáta zapísané pomocou VRML (Virtual reality markup language), čo je opisný jazyk pre prostredie virtuálnej reality. V opise jazyka je možné zadefinovať všetky atribúty potrebné pre realistické zobrazenie telies, od polohy a natočenia cez definovanie farby a mapy až po efekty, napr. priehľadnosť. Telesá na scéne je možné opísať pomocou skladania telies z primitívnych telies, ako sú napr. guľa, kváder, cylinder a podobne. Komplexné telesá sú opísané pomocou plôch vytvorených aproximáciou. V opise sú využité dva zoznamy. Jeden reprezentuje geometrické vlastnosti, čiže súradnice vrcholov v priestore. Druhý zoznam predstavuje topologické informácie a identifikuje, ktoré vrcholy tvoria jednotlivé steny. Keďže sa používajú indexy bodov, čo sú celé čísla, je tento spôsob práce s dátami výhodný kvôli nízkym nárokom na pamäť a rýchlemu spracovaniu, pretože počítače využívajú celočíselné dátové typy ako natívne.

Jednotlivé steny možno reprezentovať pomocou trojuholníkov, štvorcov alebo mnohouholníkov [1]. Nasledujúci fragment z modelu scény reprezentuje priestorový objekt kocky opísanej pomocou WRML na obr. 2 takto:

```
#VRML V2.0 utf8
# Produced by 3D Studio MAX VRML97 exporter, Version 5,
Revision 0,93
# Date: Tue Jan 10 16:24:52 2006
DEF Box01 Transform {
  translation 2.894 0 7.717
  children [
    Shape {
      appearance Appearance {
        material Material {
          diffuseColor 0.6039 0.7255 0.898
        }
      }
      geometry DEF Box01-FACES IndexedFaceSet {
        ccw TRUE
        solid TRUE
        convex TRUE
        coord DEF Box01-COORD Coordinate { point [
          -50 0 50, 50 0 50, -50 0 -50, 50 0 -50, -50 100 50,
          50 100 50, -50 100 -50, 50 100 -50]
        }
        coordIndex [
          0, 2, 3, 1, -1, 4, 5, 7, 6, -1, 0, 1, 5, 4, -1, 1, 3,
          7, 5, -1, 3, 2, 6, 7, -1, 2, 0, 4, 6, -1]
        ]
      }
    ]
  }
}
```

Obr. 2. Reprezentácia kocky vytvorenej pomocou triangulácie v jazyku VRML

Problém použitia triangulácie a mapovania

Vytvorenie stenového modelu pomocou trojuholníkov je veľmi výhodná metóda. Výhodou je hlavne rýchlosť spracovania, pretože grafické karty aj tak rozkladajú mnohouholníky na trojuholníky. Problém nastáva pri použití textúr. Textúry sa aplikujú na jednotlivé plochy. V prípade vytvorenia štvorca pomocou dvoch geometricky podobných trojuholníkov by sme museli textúru rozdeliť na dve časti a aplikovať ju na oba trojuholníky. Toto riešenie by znamenalo riziko nekonzistentného prechodu či prelínania sa textúr na ploche, ktorá má tvoriť celok. Je preto výhodnejšie vytvoriť jednotlivé steny ako mnohouholníky.

Použitie mnohouholníkov odstraňuje redundáciu dát a zvolený model je menej náročnejší na pamäť.

Tvorba stromu reprezentujúceho scénu

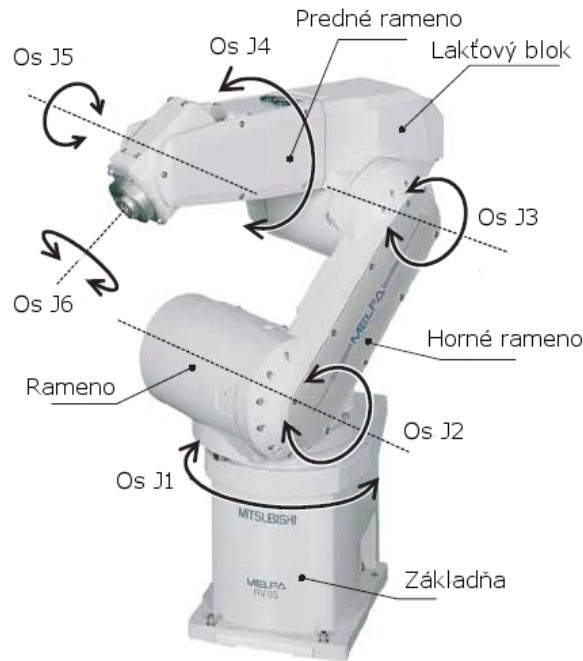
Pri reprezentácii scény je vhodným riešením stromová reprezentácia scény. Scénu tvoria stroje, ktoré sú zložené z objektov. Tvar objektov je aproximovaný mnohouholníkmi. Presnosť aproximácie závisí od množstva použitých mnohouholníkov. Počet strojov umiestnených na scéne, objektov, ktoré tvoria jednotlivé stromy a mnohouholníkov vytvárajúcich objekty je variabilný. V jazyku C++ je dynamický zoznam reprezentovaný pomocou štruktúr s dátovou časťou a smerníkom na nasledujúci prvok.

Identifikácia kinematiky angulárneho robota na virtuálnej scéne

Cieľom riadenia na virtuálnej scéne je plne funkčný virtuálny výrobný systém s priemyselným robotom riadený tak ako v praxi. Virtuálny automatizovaný systém technologického pracoviska je možné využiť v laboratórnych podmienkach pre zaškolenie a záchvat operátorov a programátorov nepretržitých prevádzok. Okrem toho automatizované výrobné systémy vnímané ako periféria priemyselného robota bude možné jednoduchým spôsobom včleniť do simulácie virtuálneho robotizovaného komplexu.

Všetky ramená angulárneho robota vykonávajú rotačný pohyb so šiestimi stupňami voľnosti. Kinematika takéhoto robota má otvorený kinematický reťazec. Základné prvky angulárneho robota tvoria tieto časti:

1. základňa (base) – je časť robota, ktorá je pevne spojená so zemou,
2. ramená (link) – sú to pevné časti robota,
3. kĺby (joint) – sú časti robota, ktoré umožňujú voľný alebo riadený pohyb dvoch ramien, ktoré kĺb spája,
4. chápadlo (end effector) – koncová časť robota, ktorá slúži k uchopeniu, manipulácii predmetu alebo na namontovanie ďalších nástrojov napr. striekacích či zvaracích hlavíc,
5. kinematická dvojica (kinematic pair) – je to dvojica ramien spojená kĺbom,
6. kinematický reťazec (kinematic chain) – je to množina ramien spojených kĺbmi.



Obr. 3. Osi rotácie angulárneho robota

Kinematický reťazec je možné reprezentovať grafom, kde kĺby tvoria uzly a ramená vytvárajú hrany grafu. Charakteristické body v kinematike angulárneho robota sú body, ktorými prechádzajú osi rotácie. Keďže os rotácie nie je jednoznačne určená okolo jedného bodu, je nutné určiť os súradnicového systému, ktorá je rovnobežná s touto osou rotácie. Vnímaním súradnicových systémov optikou nazerania OpenGL ide vlastne o posunutie súradnicového systému do bodu identifikovaného ako charakteristický. Následne je vykonaná rotácia okolo tohto bodu.

Pre identifikáciu charakteristických bodov je potrebné dosiahnuť nezávislosť od CAD systému, v ktorom budú objekty exportované do formátu WRL modelované. Je preto nutné nájsť vhodné médium prenosu informácie, ktorá bude nezávislá na zvolenom konverznom formáte. Po analýze počtu a umiestnenia týchto bodov je zrejmé, že počet charakteristických bodov je zhodný s počtom osí, v ktorých angulárny robot vykonáva rotačné pohyby podľa obr. 3. V tomto prípade ide teda o 6 význačných bodov.

Pri realizácii pohybov v OpenGL ide o skladanie pohybov a viacnásobný posun súradnicového systému. Takáto modifikácia súradnicového systému ovplyvní polohy všetkých následne vykreslených objektov. Je potrebné zistiť, kde táto zmena nastáva a akými predchádzajúcimi pohybmi môže byť v prípade ich uskutočnenia poloha objektu na scéne ovplyvnená. Systém číselného kódovania je navrhnutý tak, že umožní jednoznačne identifikovať pohyby pre každú os rotácie. Uvedené kódy sú realizované ako bitová mapa, čo zabezpečí, že ľubovoľná kombinácia pohybov bude nezameniteľná a jednoznačná. Nasledovná tabuľka určuje návrh kódov pre všeobecnú kinematiku angulárneho robota.

BITOVÁ MAPA POHYBOV PRE ANGULÁRNY ROBOT

Tabuľka 1

Typ priemyselného robota	Identifikácia pohybu	Kód (dekadický binárny)
Angulárny robot	rotačný pohyb okolo osi 1	1 1
	rotačný pohyb okolo osi 2	2 10
	rotačný pohyb okolo osi 3	4 100
	rotačný pohyb okolo osi 4	8 1000
	rotačný pohyb okolo osi 5	16 10000
	rotačný pohyb okolo osi 6	32 100000

Zachovaním princípu jednoznačnosti je možné kódovú mapu rozšíriť o translačné pohyby, a tak ju použiť pre kinematiku iných priemyselných robotov.

Dopredná kinematika

Vyriešenie doprednej kinematiky je základným problémom pri riešení kinematiky angulárneho robota. Ide vlastne o určenie polohy efektora v priestore. Správne vyriešenie doprednej kinematiky je dôležitým krokom pre určenie inverznej kinematiky. Pri výpočte polohy efektora je jedinou neznámou iba poloha tohto bodu [3]. Parametre reprezentujúce natočenie v jednotlivých osiach sú známe. Základom výpočtu je nasledovná transformačná matica:

$$\left[\begin{array}{c|c} \text{Matica} & \text{Vektor} \\ \text{rotácie} & \text{pozície} \\ \hline \text{Perspektíva} & \text{Mierka} \end{array} \right]$$

Použitím Denavit Hartenbergovej metódy výpočtu a dodržaním konvencií platí:

1. rotácia okolo X_{i-1} osi - označíme ako θ_i ,
2. transláciu po osi X_{i-1} - označíme ako u_i ,
3. rotácia okolo novej osi Z - označíme ako ψ_i ,
4. transláciu po novej osi Z - označíme ako w_i ,
5. a súčet $\cos(\psi_1+\psi_2+\psi_3)$ - označíme ako c_{123} .

Dostávame nasledovnú maticu, pomocou ktorej sa dá opísať ľubovoľná výsledná geometrická transformácia skladajúca sa z rotácií a translácií:

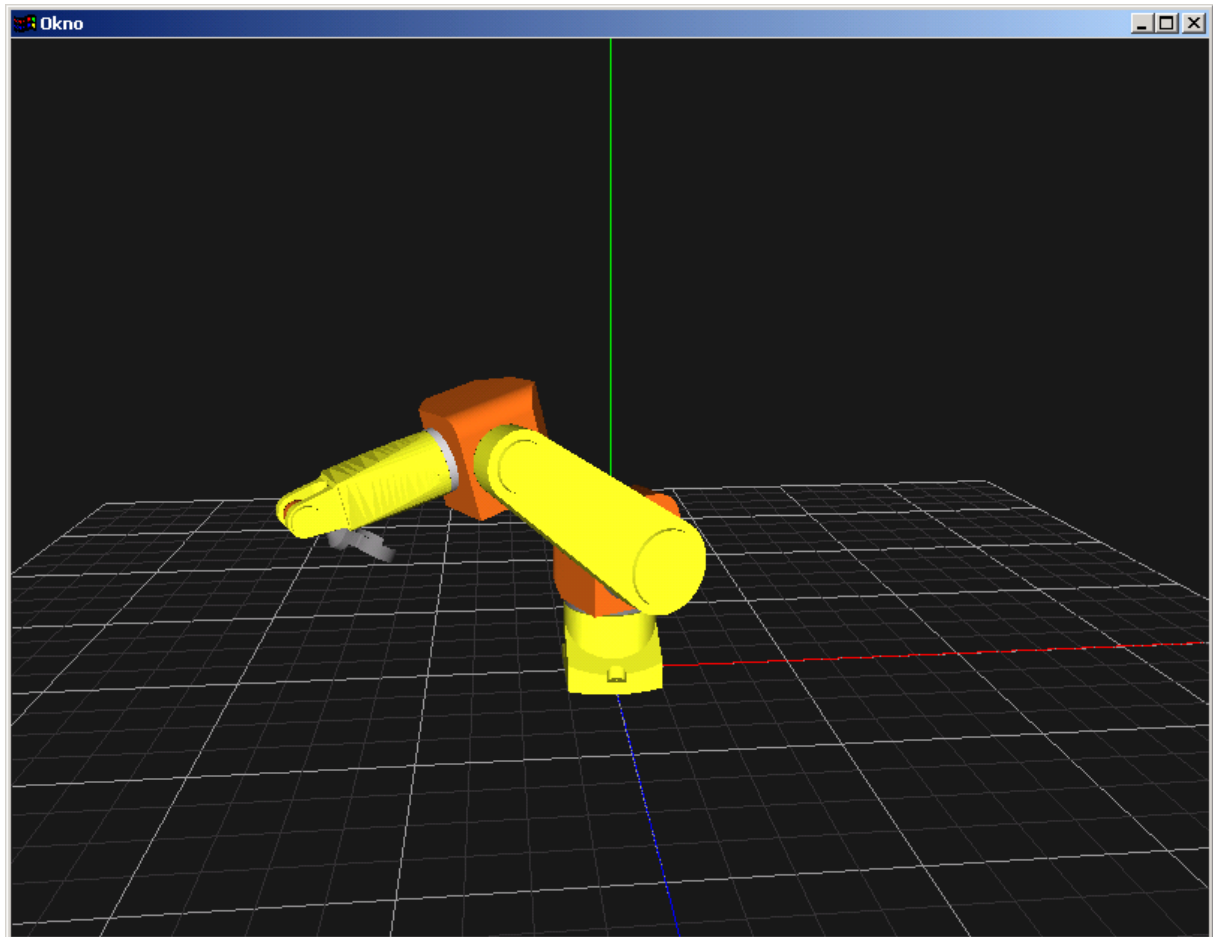
$$T_i^{i-1} = A_i = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\theta_i s\psi_i & c\theta_i c\psi_i & -s\theta_i & -s\theta_i w_i \\ s\theta_i s\psi_i & s\theta_i c\psi_i & c\theta_i & c\theta_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Výslednú polohu efektora je potom možné určiť vzťahom:

$$T_0^n = \prod_0^n A_i .$$

Inverzná kinematika

Na rozdiel od doprednej kinematiky je pri výpočte inverznej kinematiky úlohou vypočítať parametre uhlov natočenia v jednotlivých osiach na základe polohy koncového bodu efektora. Riešenie v tomto prípade nie je jednoznačné, treba riešiť problém singularít, čiže viacnásobných riešení.



Obr. 4. *Obráz robotu na virtuálnej scéne*

Výsledná aplikácia na obr. 4 obsahuje virtuálnu scénu s robotom v mierke 1:10 a umožňuje jeho riadenie pomocou režimov: učenie (TIN), automatický chod, krok po kroku a editovanie. Tieto režimy poskytujú plné ovládanie celej jeho kinematiky. Vytvorené riadiace programy je možné zálohovať do textového súboru. Všetky programy môžu byť editované.

Záver

Tendenciou dynamického rozvoja v oblasti robotizácie strojárskych technológií je vývoj inteligentných robotov, ktorých vlastnosti a schopnosti sa zvyčajne overujú na modeli. Vzhľadom k dostatočným možnostiam súčasných bežne dostupných počítačov sa modelovanie a simulácia robotov dá uskutočniť vo virtuálnom prostredí. Tvorbu virtuálneho modelu podmieňuje správna voľba geometrického modelu a grafickej knižnice [4]. Pri dostatočne určenej prenosovej rýchlosti je možné, aby si študent či pracovná obsluha precvičovali postup manipulácií priamo na vzdialenom pracovisku. Z toho vyplýva finančná

úspora, keďže nie je potrebné budovať niekoľko fyzických modelov automatizovaných výrobných systémov, ale stačí ich iba vymodelovať na virtuálnej scéne a napojiť na softvérové simulátory.

Uvedený príspevok je súčasťou riešenia projektu KEGA MŠ SR č. 3/3111/05.

Zoznam bibliografických odkazov:

- [1] ALLONZO, K. *Essential Kinematics for Autonomous Vehicles*. Pittsburg: Carnegie Mellon University, 1994, s. 50.
- [2] CSONGRÁDY, T. , PIVARČIOVÁ, E. ToolBookII Instructor a tvorba výučbových programov. In *Zborník referátov z MK: UNINFOS*. Žilina, 2002, s. 17–19.
- [3] CANNY, J. *The Complexity of Robot Motion Planning*. MIT Press. Davidor, 1989.
- [4] LOZANO-PÉREZ, T. A Simple Motion-Planning Algorithm for General Robot Manipulators. In *IEEE Transactions on Robotics and Automation*, 1996, 3, 22s4-238.