

RIADENIE ANGULÁRNEHO ROBOTA VO VIRTUÁLNOHOM PROSTREDÍ

CONTROL OF ANGULAR ROBOT IN A VIRTUAL ENVIRONMENT

Oliver MORAVČÍK, Pavol BOŽEK, Miriam IRINGOVÁ, Pavol BEZÁK,
Elena VIDOVÁ

Autori: Prof. Dr. Oliver Moravčík, Dr. Ing. Pavol Božek, Ing. Miriam Iringová, Ing. Pavol Bezák, Dr. Ing. Elena Vidová

Pracovisko: Katedra aplikovanej informatiky a automatizácie, Materiálovotechnologická fakulta STU

Adresa: Paulínska 16, 917 24 Trnava

Tel.: 00421 33 54 47 34, e-mail: moravcik@mf.stuba.sk

Abstract

Projekt sa zaoberá návrhom a riadením robota vo virtuálnom prostredí. Popisuje spôsob výstavby virtuálnej scény, použité dátové štruktúry a algoritmy. Zaoberá sa voľbou a výberom vstupného formátu a grafickou reprezentáciou objektov na scéne. Ďalej je zrealizovaný výpočet doprednej kinematiky, výpočet časti inverznej kinematiky a výpočet vektora efektora a vektora rotácie prenášaného telesa. K jednotlivým častiam výpočtu je priložený stručný komentár dokumentujúci použité riešenie.

This contribution is dedicated to design and control of an angular robot in a virtual environment, a technique of building a virtual scene, used data structures and algorithms. It covers a selection of input format and graphic representation of objects. Then the contribution illustrates calculation of forward kinematics, calculation of part inverse kinematics and calculation of effector vector and rotation vector of carried object. Each part of calculation is described by short comment.

Key words

angulárny robot, dopredná kinematika, inverzná kinematika, virtuálna scéna, angular robot, forward kinematics, inverse kinematics, virtual scene, effector

Úvod

V procese výučby študentov alebo v praxi pracovnej obsluhy robotizovaných technologických pracovísk s robotom sa čoraz viac využíva výpočtová technika. Úlohou projektu je navrhnúť vhodnú technológiu na implementáciu počítačového modelu virtuálneho technologického pracoviska. Podmienkou je, aby bola technológia čo najviac nezávislá od

platformy na akej bude prezentovaná a aby využívala v najväčšej miere najnovšie normy a štandardy oblasti počítačových technológií [1].

Výsledkom je na takomto modeli cvičiť a skúšať manipulačné postupy riadenia. Virtuálny model pracoviska simuluje jednoduchú logiku odvodenú od reálneho robotizovaného pracoviska.

Cieľom projektu je vytvoriť plne funkčné virtuálne robotizované laboratórium s priemyselným robotom ovládaným tak ako v praxi, počítačom PC. Virtuálny priemyselný robot je pre svoju jednoduchosť operátorskej obsluhy ako aj jednoduchosť prístupu k jednotlivým funkciám v používateľskom režime výnimočne vhodný pre účely výučby riadenia a programovania priemyselných robotov aj na rôznych stupňoch vzdelávania.

Ďalším dôležitým pozitívom je vytvorenie takého programového prostredia ovládania robotizovaného laboratória, ktoré spĺňa všetky ekologické, ergonomické a funkčné podmienky.

Jedným z ďalších cieľov je vytvorenie rozhrania medzi aplikáciou virtuálnej simulácie a medzi softvérovým rozhraním, ktoré bude priamo riadiť prostredníctvom hardvérového interfejsu priemyselného robota.

Konkrétny návrh postupu na dosiahnutie cieľov projektu

V prvej fáze sa vypracuje koncepcia tvorby projektu. Bude definovaný celý rozsah práce, čo všetko a akým spôsobom sa má simulovať. V druhej fáze návrhu bude namodelované robotizované pracovisko v konkrétnej mierke. V tejto fáze sa vypracuje analýza súčasného stavu režimov manipulačného zariadenia a na základe modelu budú navrhnuté nové praktické funkcie, ktoré by mali nahradiť niektoré staršie nepraktické ovládania robota. Záver bude venovaný testovaniu. Následne po otestovaní aplikácie bude virtuálna aplikácia uvedená do výučby.

Návrh rozhrania aplikácie

Vo fáze návrhu rozhrania aplikácie je potrebné definovať rozhranie medzi aplikáciou a používateľom aplikácie. Ďalšou dôležitou podmienkou prehľadnosti ovládania aplikácie je to, aby používateľ nebol „zavalený“ množstvom ovládacích prvkov. Musí ich byť čo najmenej, aj funkcia musí byť na prvý pohľad jasná. V aplikácii simulácie virtuálneho robotizovaného pracoviska síce bude mnoho ovládacích prvkov, ale budú usporiadané a včlenené do prostredia tak, aby ich použiteľnosť bola nenáročná, prehľadná a spĺňala všetky požiadavky zadané používateľom [1].

Komunikačné rozhrania

V jednotlivých častiach projektu budú použité tieto štandardy:

- VRML 97 – pre popis virtuálnej scény
- COBRA 2.0 pre zabezpečenie komunikácie medzi serverom a klientom
- JAVA pre programovanie platformovo nezávislej aplikácie.

Parameter modelu je nutné navrhovať tak, aby bolo možné tento model rozširovať jednoduchým dopĺňaním parametrov. Celá virtuálna scéna je reprezentovaná pomocou údajového stromu, kde vrcholom stromu je samotná scéna a nachádza sa na najvyššej bode v hierarchii. Samotná scéna nie je reprezentovaná žiadnou dátovou štruktúrou, pretože je iba jedna. Nastavenie scény je opísané pomocou premenných OpenGL ako napr. farba pozadia, natočenie scény, jej posunutie, rozmery mriežky, parametre nasvietenia a pod. Príslušnosť objektov k scéne sa

vd'aka jej jednoduchosti prekladá automaticky a taktiež nie je reprezentovaná pomocou žiadneho relačného vz'ahu.

Zvolené riešenie tak reprezentuje stenový model telesa. Stenový model neobsahuje informácie, pomocou ktorých by bolo možné implicitne identifikovať, či je bod v priestore súčasťou objemu alebo nie. Stenový objem v klasickom ponímaní je 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.

Dátová štruktúra reprezentujúca teleso tak obsahuje dva smerníky na tieto zoznamy. Ďalej obsahuje informácie o farbe telesa a jeho polohe a natočení voči súradnicovému systému stroja, ku ktorému patrí a o príznakoch, ktoré špecificky definujú úlohy telesa v mechanizme stroja. Každému telesu je priradené číslo, na základe ktorého sa telesu umožňuje niektorý z pohybov. Teleso ďalej nesie informáciu o tom, či je chápadlom alebo nie je. V prípade, že je chápadlom tak o poradovom čísle chápadla pre prípad, že je vymodelovaných viacero chápadiel súčasne a o tom či reprezentuje model otvoreného alebo zatvoreného chápadla. Pretože telesá tvoria prvky dynamického zoznamu obsahujú i smerníky na predchádzajúce a nasledujúce prvky v zozname.

Stroje sú v hierarchii dátových štruktúr nad telesami a sú uložené v zozname strojov a odkazujú na seba pomocou smerníkov. Stroje sa odkazujú smerníkom na zoznam telies, z ktorých sú príslušné stroje vytvorené, resp. na prvý prvok tohto zoznamu. Dátová štruktúra stroja obsahuje podobne ako dátová štruktúra telesa premenné pre uloženie informácií o polohe a natočení, tento krát sú však poloha stroja a jeho otočenie vz'ahované voči svetovému súradnicovému systému.

Do údajových štruktúr bolo taktiež potrebné zakomponovať informácie o kinematike strojov [3]. Pri analýze počtu a umiestnenia týchto bodov vyplýva záver, že počet charakteristických bodov robota resp. je zhodný s počtom osí, v ktorých vykonáva rotačné pohyby. Tento poznatok možno zovšeobecniť i na ostatné typy mechanizmov.

Jazyk VRML a grafická knižnica OpenGL

Všeobecná charakteristika VRML

VRML (Virtual Reality Modeling Language) je opisný jazyk vytvorený pre popis interaktívnych 3D objektov a svetov. Je to taktiež univerzálny výmenný formát pre 3D grafiku a multimédiá. Využitie VRML je rôznorodé a zahŕňa použitie pre technickú a vedeckú vizualizáciu, multimediálne prezentácie, zábavu, podporu vzdelávania, www stránky a virtuálne svety. Je to normalizovaný súborový formát, ktorý je definovaný v norme ISO/IEC 14772.

VRML je schopný reprezentovať statické a dynamické (animované) 3D objekty, multimediálne objekty s hyperlinkovými odkazmi na jednotlivé komponenty multimédií ako text, zvuky, obrázky, animácie a filmy.

Norma ISO/IEC 14772 nepredpokladá potrebu definície fyzických zariadení ani iných koncepcií závislých na implementácii napr. rozlíšenie obrazovky a vstupných zariadení a pod. Naopak, je to norma určená pre široké spektrum zariadení a implementácií a nezahŕňa ani také zariadenia ako monitor (displej) alebo myš.

Každý VRML súbor:

- nepriamo definuje svetový súradnicový systém pre všetky objekty definované v rámci súboru,
- priamo definuje a vytvára množinu 3D a multimediálnych objektov,
- môže špecifikovať hyperlinky k iným súborom a aplikáciám,
- môže definovať správanie sa objektov.

Hierarchická štruktúra VRML súborov umožňuje modelovať neobmedzene veľké dynamické svety. Ako príklad možno uviesť, že ak súbor svet.wrl špecifikuje svet (Zem) ako guľu, môže zároveň obsahovať hyperlinkové odkazy na kontinenty napr. (európa.wrl, amerika.wrl atď.), ktoré následne môžu obsahovať ďalší stupeň odkazov na jednotlivé mestá (paríž.wrl, londýn.wrl) nachádzajúce sa v Európe.

Medzi základné definície VRML popísané v norme ISO/IEC 14772 medzi inými patria:

Author (autor) – osoba alebo zástupca (pôvodca), ktorý vytvoril VRML súbor.

Avatar – abstraktná reprezentácia používateľa vo VRML svete.

Child (potomok) – inštancia detského uzlu (Children node).

Children node (detský uzol) – jeden z množiny uzlov, ktorého inštancia môže byť sústredená v skupine zdieľajúcej rovnaké vlastnosti v závislosti od typu skupinového uzla.

Event (udalosť) - správa zaslaná z jedného uzla do iného. Udalosť pozostáva z časovej známky a hodnoty poľa.

Geometry node (geometrický uzol) – uzol obsahujúci matematický popis trojrozmerného bodu, čiary, plochy, textového reťazca alebo masívu. Sú to uzly typu Box (kváder), Cone (ihlan), Cylinder (valec), Sphere (guľa), text atď.

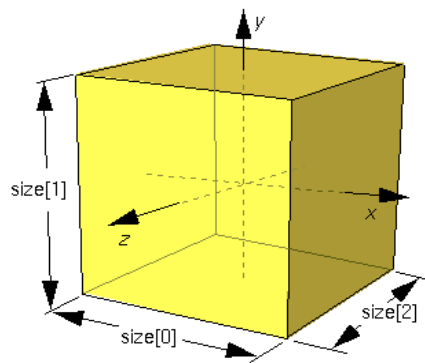
Object (objekt) – súbor dát a procedúr, reprezentovaný pomocou pravidiel a syntaxe popísanej v norme ISO/IEC 14772. Je obvykle synonymom pojmu uzol.

World (svet) – súbor jedného alebo viacerých VRML súborov a iného multimediálneho obsahu, ktorý pri interpretácii VRML prehliadačom, reprezentuje interaktívny „zážitok“ podľa zámeru autora.

Primitíva VRML

Spoločným znakom primitív, teda elementárnych objektov vo VRML je, že ich počiatok ich súradnicového systému sa nachádza v ich strede.

Kváder je jedným zo základných primitívnych objektov definovaných v špecifikácii VRML. Jeho charakteristickými parametrami sú výška, šírka a hĺbka, ktoré môžu nadobúdať hodnoty od 0 do $+\infty$. Prednastavené rozmery kvádra sú 2 2 2, čo pri umiestnení súradnicového systému do stredu kvádra (v tomto prípade kocky) znamená, že objekt siaha od -1 do $+1$ v každej osi (obr. 1).



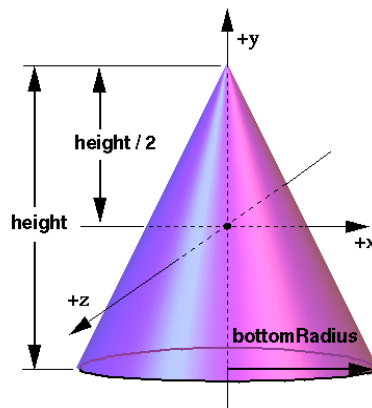
Obr. 1. Primitívum VRML - kocka

VRML kód reprezentujúci kocku ako primitívum má takúto podobu:

```
Box {
    field SFVec3f size 2 2 2    # (0, □)
}
```

Aplikácia textúr na kváder je realizovaná po jednotlivých stenách. Geometria kvádra je definovaná iba pri pohľade z vonka. Pohľad z vnútra kvádra nie je definovaný.

Ihlan je definovaný svojou výškou a polomerom základne. Os symetrie je zhodný s osou Y jeho súradnicového systému (obr. 2). Prednastavená hodnota pre polomer je 1.0 a pre výšku 2.0. Obe hodnoty nesmú byť menšie ani rovné 0.



Obr. 2. Primitívum VRML - ihlan

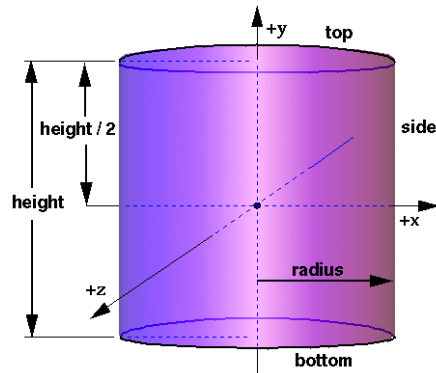
Zápis VRML kódu reprezentujúci ihlan ako primitívum je:

```
Cone {
    field SFFloat bottomRadius 1    # (0, □)
    field SFFloat height 2    # (0, □)
    field SFBool side TRUE
    field SFBool bottom TRUE
}
```

Textúry aplikované na plášť valca sú orientované proti smeru hodinových ručičiek.

VRML kód popisujúci ihlan obsahuje dve logické premenné `side` a `bottom`, ktoré definujú či táto časť objektu bude zobrazená alebo nie (obr. 2).

Podobne ako pri ihlane i os symetrie valca je totožná s osou Y súradnicového systému. Parametre výška a polomer sú štandardne prednastavené na 1 a musia byť väčšie ako 0. Počiatok súradnicového systému resp. ťažisko objektu sa nachádza v strede valca. (obr. 3). Logické premenné `top`, `bottom` a `side`, ktoré sú štandardne nastavené na `TRUE` určujú viditeľnosť vrchnej podstavy, spodnej podstavy a plášťa.

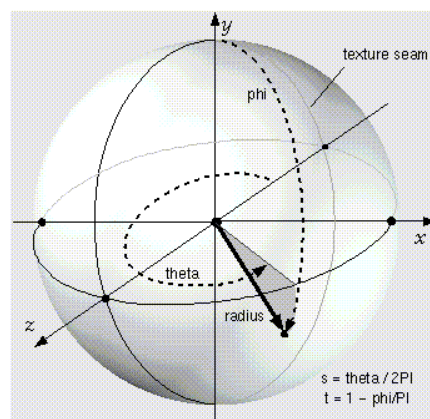


Obr. 3. Primitívum VRML – valec

Zápis VRML kódu reprezentujúci valec ako primitívum je:

```
Cylinder {
  field SFFloat bottom TRUE
  field SFFloat height 2 # (0,∞)
  field SFFloat radius 1 # (0,∞)
  field SFFloat side TRUE
  field SFFloat top TRUE
}
```

Jediným parametrom gule je jej polomer, ktorý musí nadobúdať hodnoty väčšie ako 0.



Obr. 4. Primitívum VRML – guľa

Jednoduchý zápis VRML kódu reprezentujúci guľu ako primitívum je:

```
Sphere {
  field SFFloat radius 1 # (0,□)
}
```

Reprezentácia telies vo VRML pomocou polygónov

Komplexne telesá ako tvarovo zložité telesá alebo telesá s dierami nemožno reprezentovať primitívnymi telesami. VRML preto poskytuje možnosť definovať tvar telies opisne pomocou jednotlivých bodov a ich príslušnosti k stenám telesa.

```
#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]
        }
      }
    ]
  }
}
```

Obrázok 5. Kocka vytvorená pomocou polygónov (štvorcov) v jazyku VRML

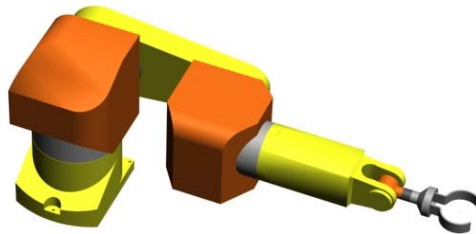
Ako je vidno z obrázku 5, opis kocky pomocou polygónov je zrealizovaný pomocou dvoch zoznamov. Jeden zo zoznamov je zoznam bodov, ktorý určuje súradnice jednotlivých bodov a druhým je zoznam plôch tvoriacich objekt, pričom jednotlivé zložky pre plochu obsahujú referenciu vo forme poradového čísla bodu v zozname bodov. Geometrické informácie sú tak oddelené od topologických. Takéto riešenie charakteristické zníženými pamäťovými nárokmi a nárokmi na výpočtový výkon, pretože jednotlivé súradnice bodov sú uložené iba raz a referencie zoznamu plôch na pozíciu v zozname bodov sú celočíselné hodnoty. Uvedeným spôsobom je možné graficky popísať ľubovoľné teleso.

Výsledok projektu je aplikácia, ktorá slúži na simuláciu pohybov angulárneho robota v rámci výrobnjej bunky so základnou schopnosťou manipulácie s objektmi. Umožňuje modelovať výrobnú bunku z modelov reálnych robotov, technologických strojov a manipulačných zariadení.

Výpočet doprednej kinematiky

Je teda dobré navrhnuť systém číselného kódovania, ktorý umožní jednoznačne identifikovať pohyby pre každú os. Uvedené kódy je výhodné realizovať ako bitovú mapu, čo zabezpečí, že ľubovoľná kombinácia pohybov bude nezameniteľná a jednoznačná. Ďalej je potrebné, aby bol pohyb telies v rámci mechanizmu stroja regulovaný. To je možné zabezpečiť stanovením rozsahu pohybu. Vytvorenie kódovej mapy a dátových štruktúr umožňuje zobrazenie angulárnych robotov vo virtuálnej scéne, pričom rozmery robotov môžu byť rôzne, avšak rovnaká štruktúra kinematického reťazca a popísaný algoritmus sú schopné zabezpečiť korektné zobrazenie v súlade s vytvoreným modelom i pohyb v jednotlivých kĺboch robota

Kinematika robota alebo mechanizmu stanovuje polohu koncového bodu efektora v priestore na základe parametrov pootočenia alebo vysunutia jednotlivých kĺbov a ramien. Prvým krokom pri výpočte doprednej kinematiky je identifikácia jednotlivých translačných a rotačných pohybov, ktoré sú potrebné pre pohyb jedného snímku do druhého. Identifikácia je aplikovaná na modeli robota, ktorý bol umiestnený v počiatku súradnicového systému a všetky uhly natočenia sa rovnali 0, ako ukazuje obr. 6.



Obr. 6. Angulárny robot v základnom stave

Záver

Výsledok projektu je aplikácia, ktorá slúži na simuláciu pohybov angulárneho robota v rámci výrobnjej bunky so základnou schopnosťou manipulácie s objektmi. Umožňuje modelovať výrobnú bunku z modelov reálnych robotov, technologických strojov a manipulačných zariadení. Používateľ tak dokáže v priebehu okamihu a bez námahy premiestňovať niekoľkotonové zariadenia, aby odhalil projekčné nedostatky alebo zvýšil efektivitu pracoviska [2]. Vývoj aplikácie bude pokračovať i naďalej s využitím na KAIA MTF STU v novoakreditovanom predmete a s plnou podporou projektu KEGA MŠ SR č. 3/3111/05. Výsledky projektu budú priebežne publikované.

Zoznam bibliografických odkazov:

- [1] ORFALI, R., HARKEY, D. *Client/server Programming with Java and Cobra*. Wiley CNew York: computer Publishing, 1998
- [2] MIHOK, J., ŠEBO, D. Životný cyklus výrobku a jeho environmentálne aspekty. In *Manažment priemyselných podnikov, vedecký časopis Univerzitetnej vedeckotechnickej spoločnosti*, 2005, roč. 2, č. 1, s. 34-38. ISSN 1336-5592
- [3] LACKO, F.: Some possibilities for induction of the required pre-tension of bolted joints. In *Proceedings of Contributions 13 th International Scientific Conference CO.MAT-TECH 2005*, Bratislava: Vydavateľstvo STU, s. 691-697. ISBN 80-227-2286-3