

Vodenje haptičnega vmesnika zasnovano v okolju Matlab xPC Target

Luka Jensterle, Matjaž Mihelj, Marko Munih

Laboratorij za robotiko, Fakulteta za elektrotehniko,
Univerza v Ljubljani
E-pošta: luka_jensterle@yahoo.com

Haptic interface control system designed in Matlab xPC Target

Haptic interfaces are devices that connect user with a virtual environment. Their main purpose is to generate forces that user senses as a virtual environment. The haptic interface is a medium for energy transmission between the user and the virtual environment.

In this paper we present the haptic robot HapticMaster and its control system built in Matlab xPC Target framework for real time operation. The developed Matlab haptic library includes blocks for position and admittance control, direct and inverse kinematics, Jacobian matrix, haptic object blocks and blocks for collision detection.

1 Uvod

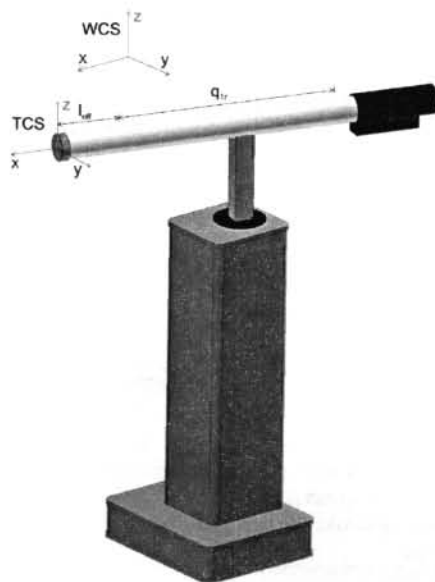
Haptični vmesniki so naprave, ki omogočajo človeku interakcijo z navideznim okoljem. Njihova glavna naloga je generiranje sile, ki jo uporabnik občuti kot navidezno okolje v katerega posega. Haptični vmesnik je tako medij za izmenjavo energije med uporabnikom in navideznim okoljem. S haptičnim vmesnikom težimo k čim bolj vernemu prikazovanju kinestetičnih in taktilnih informacij.

V programskem okolju Matlab xPC Target, smo ustvarili knjižnico blokov, ki predstavljajo celoten regulacijski sistem robota. Knjižnica vključuje bloke za pozicijsko in admitančno vodenje, direktno in inverzno kinematiko, blok za Jacobijevo matriko, bloke za haptične objekte in bloke za detekcijo trka. XPC Target je programsko okolje, s pomočjo katerega ustvarimo Simulink shemo za vodenje robota in jo prenesemo na krmilni računalnik, kjer vodenje teče v realnem času.

2 Haptični vmesnik HapticMaster

Robot HapticMaster je primarno zgrajen kot haptični vmesnik, ki omogoča uporabniku, da deluje na navidezno okolje in tudi občuti reakcijske sile le tega. Robot ima tri aktivne prostostne stopnje: vertikalna translacija, horizontalna rotacija in horizontalna translacija. Svetovni koordinatni sistem (WCS) je postavljen v središče delovnega prostora, osi pa imajo smeri določene, kot je prikazano na sliki 1. Ko je vrh robota v koordinatnem izhodišču, so vsi sklepi na polovici vrednosti delovnega območja. Koordinatni

sistem orodja (TCS) je pritrjen na vrh robotske roke, kjer je senzor sile (glej sliko 1).



Slika 1. Robotski manipulator HapticMaster

Na vrh robota lahko pritrdimo različne naprave, ki naredijo sistem primeren, na primer, za simulatorje letenja ali za rehabilitacijo zgornjih okončin.

3 Admitančno vodenje

V splošnem so lahko haptični vmesniki vodeni imedančno ali admitančno. V admitančnem modelu haptične interakcije haptični vmesnik generira premike kot odziv na izmerjeno silo. To pomeni, da merimo silo, s katero roka uporabnika deluje na vrh robota in izračunavamo hitrosti v sklepih, s katerimi se nato robot premika [4]. Za robot HapticMaster smo za vsako prostostno stopnjo določili neodvisen admitančni model vodenja. To nam omogoča kinematična struktura robota. Vrh robota se obnaša kot točkasta masa m na katero delujeta sila uporabnika F_{meas} in navidezna sila okolja F_{virt} . Merjeno silo F_{meas} izmerimo na senzorju sile na vrhu robota. To je sila s katero uporabnik deluje na vrh robota. Sila F_{virt} pa je povratna sila navideznega okolja nazaj na uporabnika. Za rotacijski sklep moramo izračunani pospešek deliti še s polmerom vrtenja robotske roke r , da dobimo kotni pospešek.

Admitančno vodenje za translacijski osi lahko zapišemo

$$\ddot{q} = (F_{meas} + F_{virt}) / m$$

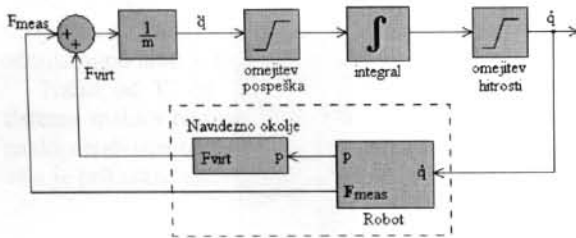
$$\dot{q} = \int \ddot{q} dt = \int (F_{meas} + F_{virt}) / m dt \quad (1)$$

admitančno vodenje za rotacijsko os pa

$$\ddot{q} = (F_{meas} + F_{virt}) / (m r)$$

$$\dot{q} = \int \ddot{q} dt = \int (F_{meas} + F_{virt}) / (m r) dt \quad (2)$$

Veličine v enačbah so skalarne vrednosti, saj nas zanimajo le posamezne osi. Navidezna masa vrha robota mora zaradi učinka vztrajnosti imeti čim manjšo vrednost, ker pa z njo delimo, njena vrednost ne sme biti nič. Če je masa premajhna, robot postane nestabilen (velik pospešek pri majhni sili), kar se odraža v nihanju vrha robota. Stabilno delovanje robota dosežemo, če nastavimo navidezno maso vrha robota na 3 kg.



Slika 2: Blokovna shema admitančnega modela za horizontalno oziroma vertikalno translacijsko prostostno stopnjo; pri rotacijski prostostni stopnji moramo upoštevati še radij vrtenja.

Z integracijo pospeška dobimo referenčno hitrost \dot{q} za vodenje posameznega sklepa. Admitančno vodenje je prikazano na sliki 2.

3.1 Stabilnost haptičnega vmesnika

Eden najpomembnejših kriterijev pri načrtovanju vodenja haptičnih vmesnikov je zagotavljanje stabilnosti delovanja. To pomeni, da pri nobenem pogoju upravljanja ali nastavitvi parametrov navideznega okolja, ne pride do neželenih nihanj ali drugih divergentnih obnašanj robota. Stabilnost lahko zagotovimo ob upoštevanju kriterija pasivnosti, ki določa da je sistem pasiven, če se energija v sistemu troši [3].

Če je sistem aktiven, moramo odvečno energijo nekje potrošiti. To izvedeno tako, da načrtamo observator in regulator pasivnosti. Z observatorjem pasivnosti opazujemo kdaj in koliko energije sistem generira in to odvečno energijo potrošimo na regulatorju pasivnosti [3].

4 Haptični objekti

Interaktivno navidezno okolje lahko zgradimo s pomočjo haptičnih objektov. Haptični objekt je tridimenzionalno (3D) telo ali ploskev v 3D prostoru, ki

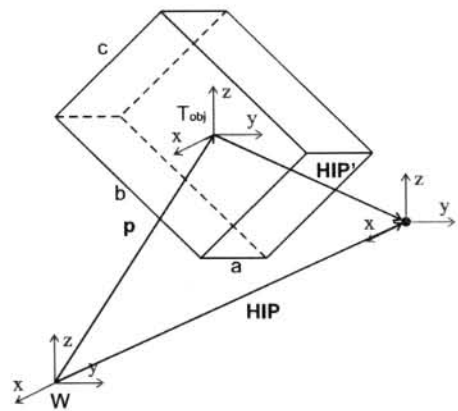
se ga z vrhom robota navidezno dotikamo [1]. Osnovni haptični objekti so stena, krogla, kvader, valj, vektorsko polje sil in drugi.

4.1 Haptični dotik

Haptični dotik običajno nastopa med vrhom haptičnega vmesnika, ki je modeliran točkovno kot točka haptične interakcije (**HIP** – haptic interaction point) in navideznim okoljem, na primer, haptičnim objektom [4]. Ta dotik moramo najprej zaznati, nato pa izračunati silo, s katero objekt deluje nazaj na uporabnika, oziroma uporabnik preko haptičnega vmesnika na objekt.

Vzemimo, da imamo objekt kvader, ki leži v legi določeni s transformacijsko matriko T_{obj} glede na svetovni koordinatni sistem. Dotik med **HIP** in objektom najlažje določimo, če položaj **HIP** iz svetovnega koordinatnega sistema preslikamo v koordinatni sistem objekta glede na sliko 3.

$$\mathbf{HIP}' = T_{obj}^{-1} \mathbf{HIP} \quad (3)$$



Slika 3: Objekt in vrh robota v navideznem okolju

S tem smo dosegli, da podajamo pozicijo vrha robota glede na koordinatni sistem objekta, ne glede na to, kje se objekt nahaja (glej sliko 3). Na ta način lahko enostavneje preverimo, ali se **HIP'** dotika objekta ali ne. V primeru kvadra moramo ugotoviti le, ali je absolutna vrednost koordinat točke **HIP'** manjša od polovice vrednosti vseh treh dimenzij kvadra. Če to drži, pomeni, da se nahajamo znotraj objekta. Podobno enostavno lahko dotik objekta preverimo tudi v primeru krogle, oziroma valja.

4.2 Izračunavanje sile dotika

V nadaljevanju bomo razložili osnovni pristop za izračun sile, s katero haptični objekt preko haptičnega vmesnika interaktivno deluje na uporabnika ter na katere pogoje moramo biti pri tem pozorni. V funkciji za haptični objekt najprej izračunamo silo, s katero **HIP** deluje na objekt. Sila navideznega okolja na uporabnika pa je tej sili nasprotna.

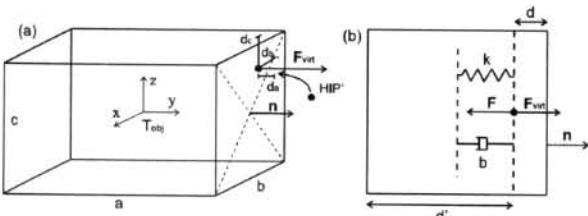
Haptični objekt je omejitven v prostoru. Iz tega sledi, da ko smo s **HIP** izven objekta, je sila navideznega okolja enaka nič, vse dokler ne pridemo v stik z objektom.

Stena haptičnega objekta je definirana kot vzmet in usmerjeni dušilnik [4]. Povratno silo začnemo izračunavati ob dotiku objekta. Sila je sorazmerna globini prodora v objekt d preko konstante togosti k in relativni hitrosti med **HIP** in točko dotika na objektu preko konstante dušilnika b .

$$\mathbf{F}'_{virt} = (k d + b(\mathbf{v}_O - \mathbf{v}_{HIP}) \cdot \mathbf{n}) \mathbf{n}$$

$$d = \min(\mathbf{HIP}' - \mathbf{dim}_{obj} / 2)$$

kjer je \mathbf{F}'_{virt} sila na vrh robota v lokalnem koordinatnem sistemu objekta, d je globina prodora **HIP** v objekt, \mathbf{v}_O je vektor hitrosti objekta, \mathbf{v}_{HIP} je vektor hitrosti vrha robota in \mathbf{n} je vektor normale na ploskev dotika. Enačba (4) govori, da globlje kot prodremo v steno in z večjo hitrostjo kot se ji približujemo, večja je sila. Pri hitrosti upoštevamo tako razliko translacijske hitrosti, kot tudi razliko hitrosti med **HIP** in med točko dotika na steni objekta, zaradi rotiranja samega objekta. Smer sile je vedno v smeri normale \mathbf{n} na ploskev dotika, ki kaže ven iz objekta. Pri odmikanju od stene moramo upoštevati le globino prodora v steno, hitrosti pa ne (usmerjeni dušilnik), saj bi nam v nasprotnem primeru negativna hitrost preprečevala odmik od objekta. Globino prodora izračunamo kot razliko med pozicijo vrha robota **HIP'** in polovico dimenzije objekta \mathbf{dim}_{obj} (glej sliko 4). Paziti moramo, da vedno vzamemo razdaljo do stene, ki je **HIP** najbližja, saj le ta stena prispeva silo, ostale pa ne.



Slika 4: (a) Prodor **HIP** v objekt in pripadajoča sila in (b) sila modelirana z vzmetjo in dušilko.

Problem nastopi blizu robov objekta, saj lahko naletimo na pogoj, da je kljub dotiku ene stene, **HIP** bližje drugi steni, kar povzroči neželene smeri sil ali nezveznosti v silah. To odpravimo tako, da dokler se nahajamo s **HIP** znotraj objekta, ohranjamo smer sile, izračunavamo le njeno velikost, ki kaže v smeri vektorja normale na

ploskev dotika. Izračunana vrednost sile, s katero delujemo na objekt je sedaj izražena v koordinatnem sistemu objekta, zato jo s premultiplikacijo z rotacijsko matriko \mathbf{R}_{obj} objekta preslikamo v svetovni koordinatni sistem.

$$\mathbf{F}_{virt} = \mathbf{R}_{obj} \mathbf{F}'_{virt}$$

Nasprotna sila deluje na uporabnika. Iz vsote vseh sil, ki delujejo na telo, nato izračunamo translacijski pospešek objekta (glej sliko 5).

$$\mathbf{F}_{aux} + \mathbf{F}_{virt} - \mathbf{F}_b + \mathbf{F}_g = m \ddot{\mathbf{p}}$$

$$\ddot{\mathbf{p}} = m^{-1}(\mathbf{F}_{aux} + \mathbf{F}_{virt} - b_{trans} \dot{\mathbf{p}} + m \mathbf{g})$$

kjer je \mathbf{F}_{aux} sila navideznega okolja na objekt, \mathbf{F}_b je sila dušenja okolja in b_{trans} je translacijsko dušenje. Če enačbo (6) integriramo, dobimo translacijsko hitrost $\dot{\mathbf{p}}$ in položaj objekta \mathbf{p} . Iz vsote vseh navorov, ki delujejo na telo, izračunamo kotno hitrost objekta.

$$\mathbf{L} = \int (\mathbf{T}_{aux} + \mathbf{T}_{virt} - \mathbf{T}_b) dt$$

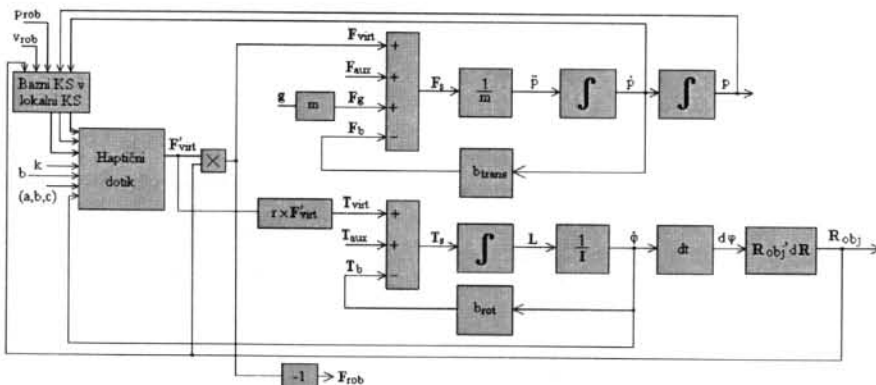
$$\mathbf{L} = \mathbf{I} \dot{\phi}$$

$$\dot{\phi} = \mathbf{I}^{-1} \int (\mathbf{T}_{aux} + r \times \mathbf{F}_{virt} - b_{rot} \dot{\phi}) dt$$

kjer je \mathbf{L} vrtilna količina objekta, \mathbf{T}_{aux} navor navideznega okolja na objekt, \mathbf{T}_b navor zaradi dušenja okolja, b_{rot} je rotacijsko dušenje, $\dot{\phi}$ kotna hitrost in r je ročica na katero deluje sila \mathbf{F}_{virt} . Iz kotne hitrosti izračunamo spremembo orientacije v enem računskem koraku $d\mathbf{R}$ nato pa novo orientacijo s produktom $\mathbf{R}_{obj} d\mathbf{R}$ kot je razvidno iz slike 5. Izhodne vrednosti funkcije za haptični objekt so pozicija in orientacija objekta ter sila navideznega okolja na uporabnika.

5 Detekcija trka med objekti

Detekcija trka med objekti je pomemben del načrtovanja navideznega okolja, saj določa trke in prenos energije med samimi haptičnimi objekti. Posledica trkov je sprememba gibalne količine objektov.



Slika 5: Bločna shema za izračun sil haptičnega objekta.

Detekcija trka je realizirana podobno kot dotik točke haptične interakcije s haptičnim objektom, le da je tokrat namesto HIP drugi objekt. Ker objekt ni točka kot HIP, je detekcija trkov med objekti zahtevnejša naloga. Trk vedno obravnavamo le med dvema objektoma. Če je v navideznem okolju prisotnih več objektov, moramo določiti trke vsakega objekta z vsakim.

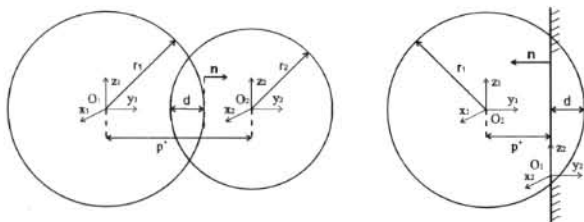
5.1 Trk dveh krogel

Obravnavali bomo trk med dvema kroglama ter med kroglo in steno. Krogla je v smislu obravnave trka najenostavnejši objekt. Ker imamo opravka le s krogli in stenami, lahko poenostavimo, da je trk vedno le točkoven. To pa ne velja vedno za druga telesa, ki lahko trčijo eno ali več točkovno, linijsko ali celo ploskovno, kar zelo oteži izračunavanje trka in sile. Ostalih teles zato na tem mestu ne bomo obravnavali.

Podobno kot v poglavju 4.1 določimo trk med objektoma tako, da določimo en objekt kot referenčni, z referenčnim koordinatnim sistemom, koordinatni sistem drugega pa transformiramo glede na prvega. S tem dosežemo, da vedno preračunavamo le relativno lego drugega objekta glede na prvega. Kot poenostavitev določitve trka, lahko tudi ostale oblike objektov ovijemo v navidezno kroglo (ang. bounding sphere) [2], ki objekt v celoti obdaja. Polmer navidezne krogle pa je enak razdalji od koordinatnega sistema objekta do najbolj oddaljene točke na objektu. Za prvi približek trka lahko vedno opazujemo trk navideznih krogel, šele nato, če je potrebno, izračunamo trk med samima objektoma.

5.2 Izračun sile

Silo na telo izračunamo iz globine prodora enega objekta v drugi objekt, kot je razvidno iz slike 6.



Slika 6: (a) Trk dveh krogel in (b) trk med kroglo in steno.

$$\mathbf{F}'_1 = -(k_s d - b_s \mathbf{v}_d \cdot \mathbf{n}) \mathbf{n}$$

$$d = r_1 + r_2 - p'$$

$$\mathbf{v}_d = \mathbf{v}_1 - \mathbf{v}_2 \quad , \quad (8)$$

$$k_s = (1/k_1 + 1/k_2)^{-1}$$

$$b_s = (b_1 + b_2)$$

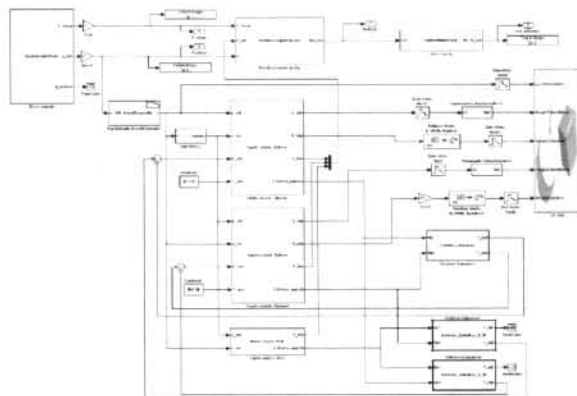
kjer je F'_1 sila na prvi objekt v lokalnem koordinatnem sistemu, p' je razdalja med središčema krogel, d je globina prodora ene krogle v drugo, k_s je togost interakcije, b_s dušenje interakcije, \mathbf{v}_d je razlika hitrosti med objektoma, \mathbf{n} pa vektor normale na ploskev dotika. Hitrosti objektov upoštevamo le, ko telesi prodirata eno v drugo, ne pa pri odmikanju. Silo je potrebno še

transformirati iz lokalnega v svetovni koordinatni sistem. Sila na drugo telo je nasprotna sili na prvo telo.

Izračunavanje trka med kroglo in steno je podobno, saj koordinatni sistem krogle transformiramo glede na steno, ki je togo vpeta v prostor.

6 Zaključek

Kot rezultat lahko s pomočjo blokov v Simulink okolju zgradimo navidezno okolje, ki vključuje blok za admitančno vodenje, haptične objekte kot so krogle, kvadri in stene ter blok za detekcijo trka med objekti, kot je prikazano na sliki 7.



Slika 7: Simulink shema naloge.

Ko se z vrhom robota dotaknemo objekta občutimo reakcijsko silo okolja. Za še bolj realno predstavitev pa lahko vključimo tudi 3D vizualizacijo, kjer se uporabnik nato nahaja sredi navideznega okolja, ki ga vidi in občuti.

Področje haptičnega vodenja je perspektivno tako na akademsko-izobraževalnem področju, v medicini za rehabilitacijo pacientov, kot tudi za zabavo v obliki naprednih računalniških iger.

Literatura

- [1] D. Baraff, "Non-penetrating rigid body simulation," *Eurographics '93 State of the Art Reports*, chapter 2, pp. 2-5, 1993.
- [2] D. Baraff, "Physically based modeling - rigid body simulation," *In Siggraph 2001 Course Notes*, pp. G34-G39, 2001.
- [3] B. Hannaford, in J.-H. Ryu, "Time domain passivity control of haptic interfaces," *IEEE Transactions on robotics and automatics and automation*, vol. 18, No.1, pp. 1-2, 2002.
- [4] Mihelj, M. Haptični roboti, Založba FE in FRI, 2007.