



IKER
GAZTE
NAZIOARTEKO
IKERKETA EUSKARAZ

II. IKERGAZTE

NAZIOARTEKO IKERKETA EUSKARAZ

2017ko maiatzaren 10, 11 eta 12
Iruñea, Euskal Herria

ANTOLATZAILEA:
Udako Euskal Unibertsitatea (UEU)

INGENIARITZA ETA ARKITEKTURA

**Gorputz-anitzeko sistemen
modelizazioa operadore
errekurtsibo eta espresioen
atomizazioaren bidez**

*Aitor Plaza, Javier Ros, Xabier Iriarte
eta Jokin Aginaga*

103-110 or.
<https://dx.doi.org/10.26876/ikergazte.ii.03.15>

ANTOLATZAILEA:



ELKARLANEAN:



LAGUNTZAILEAK:



Gorputz-anitzeko sistemen modelizazioa operadore errekurtsibo eta espresioen atomizazioaren bidez.

Aitor Plaza Puértolas eta Javier Ros Ganuza

Xabier Iriarte Goi eta Jokin Aginaga Garcia¹

Nafarroako Unibertsitate Publikoa. Arrosadia Kanpusa z/g, 31006 Iruñea

Laburpena

Artikulu honetan laburtzen da artikulu honen egile baten tesiaren zati bat. Tesia gorputz-anitzeko sistemen dinamikaren arloan kokatzen da, hain zuzen ere modelizazio sinbolikoan eta bertan deskribatzen dira kontuan hartu beharreko teknika batzuk nahi bada “kalitatezko” modeloak lortzea. Azkenen helburua da modelo hauek denbora-erleko aplikazioetan erabiltzea. Artikulu honetan deskribatzen dira modu eskematiko batean zeintzuk diren teknika horiek eta zein den haien garrantzia modeloaren kalitatean.

Hitz gakoak: gorpuz-anitzeko sistemak, modelizazio sinbolikoa, denbora-erreal

Abstract

In this paper one of the author summarizes a part of his PhD. thesis. This thesis is focused on the Multibody Systems Dynamics field, specially on symbolic modeling task. In the thesis are described some techniques and algorithms that should be used if quality multibody models are desired. The main target is use these models in real-time applications. In this paper these techniques and their influence in good symbolic models will be described in a schematic way.

Keywords: multibody systems, symbolic modeling, real-time

1. Sarrera eta motibazioa

Denbora-erreal kontzeptua arlo ezberdinetan oso erabilia izan da, batez ere arlo teknikoetan. Batez ere, gorputz-anitzeko sistemen dinamikaren (GSD) arloan erraten da kalkulu bat denbora-errelean egin ahal dela sistemaren estatua momentu batean aitzineko beste momentu batetik kalkulatzeko bada bi momentu horien denbora diferentzia baino azkarrago.

Egun, GSDren arloan denbora-errealeko modeloak ezinbesteko tresna bat bilakatu dira hainbat aplikazioetan: simulazioa, kontrol prediktiboa, “Hardware in the Loop”, estimazio errekurtsiboa (Kalman iragazketa) eta abar. Denbora-errealeko funtzionamendua nahi bada, besteak beste hiru osagai hauek kontuan hartu behar dira (De Jalon eta Bayo, 2012) konputazio-sistema, integrazio metodoak eta modeloa deskribatzen duten funtzioen “kalitatea”. Alde batetik, modeloek konputazio-sistema batean funtzionatu dute eta hauek izan behar dira ahalik eta azkarrenak. Beste aldetik, integrazio metodo ezberdinak (inplizitua, esplizitua, pausu-anitzekoa, ...) egonkorak eta karga konputazional gutxikoak izan behar dira, kalkuluak ere ahalik eta azkarren egiteko. Azkenik, modeloen “kalitatea” ona izan behar da, hau da, modeloa deskribatzen duten funtzioen (konputazio-sisteman erabiltzeko kodea) ebaluazio numerikoak¹ kalkulu kopuru optimo (minimo) bat izan behar du.

GSDren simulazio numerikoetan agertzen diren funtzio tipikoak (Jacobtararak, masa matrizea, indar orokortuen bektorea eta abar...) modelizazio sinboliko zein numerikoaren bidez lortzen ahal dira. Hainbat autorek (Fisette eta Samin, 2003; Schmitke eta Goossens, 2011; Gil, 2005) erakutsi duten moduan, metodo sinbolikoak² oso eraginkorra den kodea sortzeko balio du eta, gehienetan, bere eraginkortasunak kode ez-sinbolikoarena (numerikoa) gainditzen du. Honi gehitu behar zaio modelo numerikoetan ez bezala (simulazio pausu bakoitzean modeloa sortu behar da) modelo sinbolikoak bakarrik behin sortu behar direla eta honek simulazioari arintasuna ematen dio.

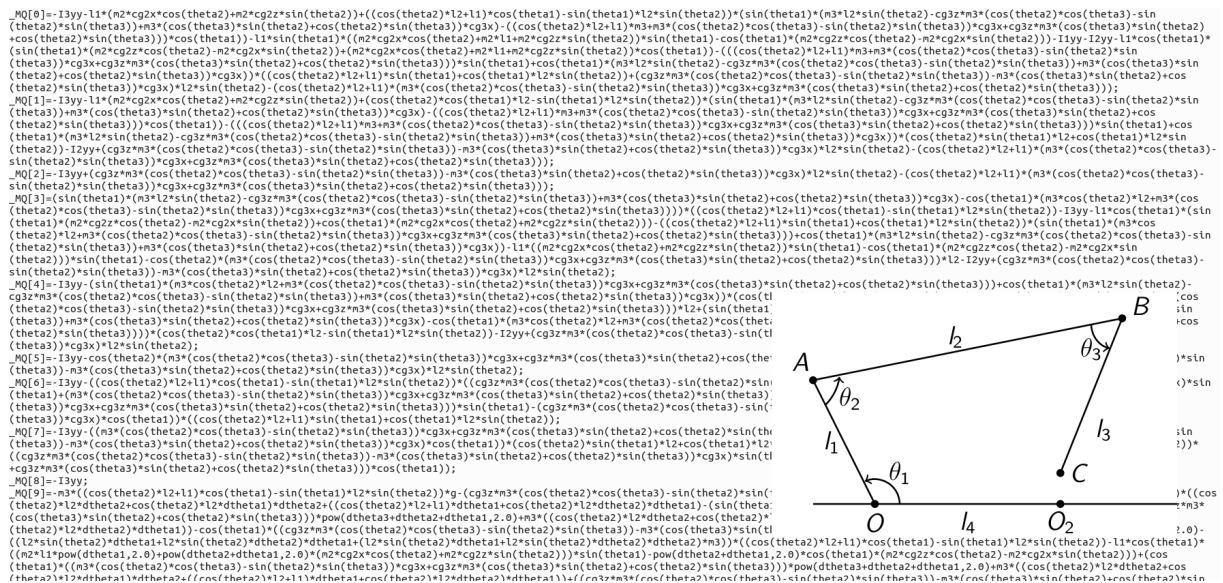
¹Ebaluazio numerikoa deitzen zaio funtzioetan kodearen aldagaiak bere zenbakizko balioekin ordezkatzeari eta datu horiek operatzeari.

²Metodo sinbolikoek kalkuluak egiten dituzte sinboloen bidez, hau da, letrak eta zenbakien bidez. Adibidez: $a + 2a = 3a$

Hala eta guztiz ere, sistemaren funtzioak modu sinboliko batez sortzea konplexutasun handiko lan bat izaten ahal da, batez ere, sistemak konplexuak (eta ez hain konplexu) direnean. Hau da, zalantzarik gabe, modelizazio sinbolikoaren botila-lepoa. Lan zama handi honek bi arrazoi nagusi ditu. Alde batetik, existitzen da ongi ezaguna den “espresio sinbolikoen ezteandaren” arazoa non espresio sinbolikoaren tamaina erraldoia bilakatzen den ezinezkoa bilakatuz haiekin lan egitea. Eta beste aldetik, hainbat espresio sinboliko ezberdinetan agertzen ahal direlako errepikatzen diren azpi-espresioak³. Ez da komenigarria espresio bera bi aldiz (edo gehiago) ebaluatzea beraz espresio hauek bilatu eta identifikatu egin behar dira erraldoia izaten ahal den espresio multzo batean eta honek ere izugarritzko karga konputazional eskatzen du.

1 irudiak erakusten du aski ezaguna den “lauki artikulatua” mekanismoa eta bere masa matrizea gehi indar orokortua bektorea. Matrize (funtzio) honen tamaina 3×4 da eta ikusten ahal denez, bere ebaluazio numerikoa gauzatzeko operazio aunitz egin behar dira, 2326 operazio hain zuzen ere.

1 Irudia: Lauki artikulatu baten masa matrizea eta indar orokortuen bektorea.



Irudi honek laburbiltzen du lan honen motibazioa zein izen den, hau da, modelo simple baten kodeak horrelako itxura baldin badu, nolako izanen da modelo askoz handiago batena?

2 Arloko egoera eta ikerketaren helburuak

Modelizazio sinbolikoa beti errobotikaren munduarekin oso lotuta izan da (Kircanski *et al.*, 1993) eta esparru honetan 80ko hamarkadan garatu ziren lehendabiziko formulazio errekursiboak (Featherstone, 1983) zeinek optimoa zen kodea sortzea zuten helburu. Garai hartako konputazio-sistemak oso mugatuak zirez garrantzi handia eman zion kode optimoa lortzeari. Geroztik, konputazio sistemen potentzia haztearekin batera, modelizazio sinbolikoa erori zen modelizazio numerikoaren alde.

Hala ere, azken urte hauetan goraka bat izan du batez ere errobotikaren esparruaren kanpo, denbora-erreal kontzeptua beste arlo batzuetan sartu izan delako, batez ere aplikazio aunitzetan ingeniaritzaren esparruan. Abiabidez, makinak ingurune birtual batean kontroladore erreal batekin simulatzeko (Ros *et al.*, 2013), trenetan gupila eta karrilaren arteko kontaktu puntua aurkitzeko (Falomi *et al.*, 2011) edota gidari baten portaera simulazio batean testatzeko (Shiiba, 2007).

Gaur egun existitzen dira GSDren simulazioetarako kodea sortzeko software pakete sinboliko ezberdin

³ Adibidez hurrengo espresioak, $a = l \sin(\theta)k + 2(m \cos(\theta) + l \sin(\theta))^2$, errepikatzen den azpi-espresio bat, $l \sin(\theta)$, du.

batzuk (Robotran⁴, MapleSim⁵, Neweul-M²⁶, Wolfram SystemModeler⁷ besteak beste). Bakoitzak aurreko atalean aipatutako bi arazo horiek modu ezberdin batean tratatzen ditu, arrakasta maila ezberdinak lortuz. Hala ere, programa hauek badituzte indarguneak zein desabantailak.

Adibidez, programa batzuek lortzen dute ezagunak diren formulazio errekurtsiboak eta koordinatu erlatiboak erabiliz. Horrela saihesten dute errepikatzen diren azpi-espresioen bilaketa, karga konputazionala txikiagotuz (hau da Robotran-en indargune bat). Honek, nolabait, konprometitzen du software-n orokortasuna, erabiltzailea formulazio zehatz bat erabiltzera behartuta baitago, beste formulazioen abantailak alde batera utziz. Pentsa dezagun, abidinez, Lagrange-n formulazioa batean koordinatu absolutuekin.

Arrazoi honengatik, gorputz-anitzeko sistemen modelizazioa gauzatzeko kode sinbolikoaren pakete orokorrak (MAPLE⁸, MATLAB Sybolic Toolbox⁹ edota Sympy¹⁰) erabili izan dira. Adibidez, programa batzuek aurreko pakete sinbolikoetan oinarrituta daude, MapleSim (MAPLE) eta Neweul-M² (MATLAB Sybolic Toolbox). Baina, pakete sinboliko hauek orokorrak izateagatik, ez dira gai gorputz-anitzeko sistemen optimizazio guztiak gauzatzeko, optimoa ez den kodea sortuz.

Lan honek helburu ezberdin batzuk ditu eta haiei aurre egiteko oinarritzko teknika eta algoritmo ezberdin batzuk deskribatzen dira. Hauen xedea, alde batetik, GSDren ekuazioak (eta beharreko kode funtzioak) kalitatezkoak sortzekoa da, denbora-errealeko aplikazioetan erabil daitezzen. Beste aldetik, modelizazioa modu eraginkor batean egin beharko da, modelizazio sinbolikoa aipatutako botila-lepoa izan ez dadin. Azkenik, hauek baliagarriak izan behar dira edozein formulazio eta edozein koordinatu mota erabili nahi bada, erabiltzaileak erosoago sentiarazten duen teknika erabil dezan. Hau da, gai izan behar dira gorputz-anitzeko sistemen duen topologiatik abantaila guztia ateratzeko, formulazio eta koordinatu menpekoak izan gabe.

Teknika eta algoritmo hauek LIB3D-MEC-GiNaC (J. Ros, 2007) GSDrako programa-liburutegia sinbolikoan programatuak izan dira.

3 Ikerketaren muina

Sekzio honetan deskribatzen dira ikerketan honen ondorio diren teknika eta algoritmo sinbolikoak. Hauek tan sakondu nahi bada, artikulua honen tesian (Plaza, 2016) aurkitzen ahal dira.

3.1 Atomizazioa

Arlo honetan “Atomizazioa” deitzen diogu espresio sinboliko multzo batean errepikatzen diren azpi-espresioak kudeatzen dituen teknikari. Atomizazioak azpi-espresio hauek bilatzen ditu eta aldagai balio-kide batengatik ordezkatzen ditu. Kontzeptu hau lan honetan oinarritzkoa izan denez, hurrengo adibideak lagunduko du hobeki ulertzeko.

Izan dadila hurrengo bektore sinbolikoa, bi espresio sinbolikoek osatuta:

$$\delta = \begin{bmatrix} -F_b + \sin(-\theta) l_p m_p \dot{\theta}^2 c_{vis} \dot{x} \\ M_p + (\theta - \theta_0) k - \sin(-\theta) l_p g m_p \end{bmatrix} \quad (1)$$

Multzo honetan badago errepikatzen ari den azpi-espresio bat: $\sin(-\theta) l_p m_p$, hain zuzen ere.

Beraz, hau aldagai laguntzaileen, α_1 y α_2 , bidez izaten ahal da ordezkatuta¹¹:

⁴Web orria: www.robotran.be/

⁵Web orria: www.maplesoft.com/products/maplesim/index.aspx

⁶Web orria: www.itm.uni-stuttgart.de/research/neweul/neweul_en.php

⁷Web orria: www.wolfram.com/system-modeler/

⁸Web orria: www.maplesoft.com

⁹Web orria: mathworks.com/products/symbolic.html

¹⁰Web orria: www.sympy.org/

¹¹Ikusi hau ez dela aukera bakarra

$$\begin{aligned}
\alpha_1 &= l_p \, m_p \\
\alpha_2 &= \sin(-\theta) \, \alpha_1 \\
\delta &= \left[\begin{array}{c} -F_b + \alpha_2 \, \dot{\theta}^2 \, c_{vis} \, \dot{x} \\ M_p + (\theta - \theta_0) \, k - \alpha_2 \, g \end{array} \right]
\end{aligned} \tag{2}$$

δ bektorea ebaluazio numerikoa gauzatzean, azpi-espresio ber-ebaluazioa saihesten da, $\sin(-\theta) \, l_p \, m_p$ espresioa bakarrik behin ebaluatu behar baita α_2 balio numerikoa lortuz. Gero $\alpha_2 \, \delta$ bektorean agertzen den tokian ordezkutzen da. Kasu sinple honetan, honek lau operazio (hiru biderketa eta sinu bat) aurretzen ditu eta gainera espresioaren errepresentazioa txikiagoa da 1 ekuazioan baino.

Arlo honetan aldagai baliokide horiei “atomoak” deitzen zaizkie. Garrantzitsua ikustea atomo bakoitzak jadanik partzialki atomizatuta izan den azpi-espresio bat adierazten ahal duela, adibidean gertatzen den bezala. Atomizazioaren kontrako prozedurari des-atomizazioa deitu zaio, hau da, atomizatua den espresio batetik jatorrizko espresioa lortzeari.

3.1.1 Sortu ahaleko atomizazioa

Atomizazioak erraldoia izaten ahal den espresio multzo batean errepikatzen diren azpi-espresio guztiak bilatzea eta ea errepikatzen diren ikustea eskatzen du. Lan hau, konputazionalki oso pisutsua da konbinazioa kopurua ere erraldoia izaten ahal baita. Horregatik atomizazio prozesua arintzeko lan honetan “sortu ahaleko atomizazioa” asmatua izan zen.

Idea ea da konplikatua, operazio aljebraiko bat (batuketa/kenketa eta biderketa/zatiketa) edo funtzio matematiko (sin edo cos) bat egiten den aldi guztietan atomo bat sortzen da.

Atomizazio teknika aplikatzen bada 1 irudiak erakusten duen sisteman funtzio bera optimizatua lortzen da. Kasu honetan bakarrik ebaluazio numerikoak 285 operazio behar ditu. Adibide gisa, hurrengo kode zerrendak erakusten du zein den funtzio honen itxura.

Zerrenda 1: Lauki artikulatu baten masa matrizea eta indar orokortuen bektorea. Atomizatua

```

atom15 = sin(theta1);
atom101 = -l2*atom27;
atom26 = cos(theta2);
atom100 = atom26*12;
atom14 = cos(theta1);
atom102 = l1+atom100;
atom193 = -atom15*atom101;
atom1 = sin(theta3);
atom0 = cos(theta3);
atom127 = -atom1*atom27+atom26*atom0;
atom43 = cg3z*m3;
atom42 = cg3x*m3;
atom130 = atom1*atom26+atom27*atom0;
atom137 = -atom42*atom130+atom127*atom43;
atom514 = -atom101*m3-atom137;
atom134 = atom43*atom130+atom127*atom42;
atom517 = atom134+atom102*m3;
atom521 = -atom514*atom15;
atom195 = atom193-atom14*atom102;
atom523 = atom521+atom14*atom517;
atom190 = atom14*atom101;
atom515 = atom514*atom14;
atom192 = atom190-atom15*atom102;
atom519 = atom15*atom517+atom515;
atom24 = m2*cg2x;
atom25 = cg2z*m2;
atom75 = atom24*atom26+atom27*atom25;
atom501 = atom75+l1*m2;
atom78 = -atom24*atom27+atom26*atom25;
atom499 = -atom14*atom78;
atom186 = -l1*atom15;
atom505 = atom78*atom15;
atom187 = -l1*atom14;
atom511 = -l2yy-l1*atom75;
atom526 = atom101*atom137;
atom529 = -atom526-atom134*atom102-l3yy;
atom540 = atom134+atom100*m3;
atom545 = atom14*atom540+atom521;
atom542 = atom540*atom15+atom515;
atom550 = -atom526-atom100*atom134-l3yy;
atom554 = -atom14*atom137+atom134*atom15;
atom558 = atom15*atom137+atom14*atom134;
atom99 = l2*dtheta2;
atom106 = -atom26*atom99;
atom585 = dtheta1*(atom106-dtheta1*atom102)
          +(atom106-atom100*dtheta1)*dtheta2;
atom107 = -atom27*atom99+dtheta1*atom101;

atom124 = dtheta3+dtheta2+dtheta1;
atom591 = -atom589*m3+(atom124*atom124)*atom137;
atom587 = -atom585*m3+(atom124*atom124)*atom134;
atom593 = atom15*atom591+atom14*atom587;
atom597 = -atom15*atom587+atom14*atom591;
atom71 = dtheta2+dtheta1;
atom70 = -l1*(dtheta1*dtheta1);
atom574 = (atom71*atom71)*atom75-atom70*m2;
atom82 = (atom71*atom71)*atom78;
atom57 = -g*m3;
atom30 = -atom27*atom15+atom26*atom14;
atom33 = atom14*atom27+atom26*atom15;
atom46 = atom30*atom0-atom33*atom1;
atom49 = atom1*atom30+atom33*atom0;
atom40 = g*(atom24*atom30+atom33*atom25);
atom56 = g*(atom46*atom42+atom49*atom43);
atom92 = atom70*atom78;
atom602 = -atom589*atom134+atom585*atom137;
atom253 = atom46*Fz3-atom49*Fz3;
atom250 = Fx3*atom46+atom49*Fz3;
atom200 = atom193-atom100*atom14;
atom198 = atom190-atom100*atom15;

MQ[0] = -l1yy+atom511+(atom501*atom15+atom499)
        *atom186+atom195*atom523+atom187*(atom501
        *atom14+atom505)+atom529+atom192*atom519;
MQ[1] = atom523*atom200+atom511+atom198*atom519+atom529;
MQ[2] = atom529;
MQ[3] = (atom75*atom15+atom499)*atom186+atom192*
        atom542+atom550-l2yy+atom195*atom545+atom187
        *(atom505+atom14*atom75);
MQ[4] = atom545*atom200+atom198*atom542+atom550-l2yy;
MQ[5] = atom550;
MQ[6] = atom554*atom192+atom195*atom558-l3yy;
MQ[7] = atom200*atom558+atom198*atom554-l3yy;
MQ[8] = -l3yy*(atom574*atom15-atom82*atom14)*atom187-
        -atom192*atom593+atom92-atom195*atom597+
        (atom574*atom15-atom82*atom14)*atom187-
        g*(m1*atom15*cg1z+cg1x*atom14*m1)-My2-
        atom40+atom602-atom195*atom57-(Fx2*atom30+
        atom33*Fz2)*atom186+g*m2*atom187-atom250*atom192-
        My3+(Fx2*atom33-Fz2*atom30)*atom187-(atom82*
        atom15+atom14*atom574)*atom186-atom195*atom253
        -atom56;
MQ[10] = -atom250*atom198+atom92-atom198*atom593-My2-
        atom40-atom253*atom200+atom602-My3-atom597*atom200
        -atom56-atom57*atom200;
MQ[11] = atom602-My3-atom56;

```

3.2 Operazio sinbolikoak

Espresio sinbolikoekin lan egitean ohikoa da, batez ere GSDren arloan, deribazio eta ordezkatzeko operazioak ere gauzatu behar izatea. Lan honetan operazio hauek optimizatuak izan dira ez dadin derrigorrezkoa izan espresioa des-atomizatzea operazio bera gauzatzeko. Hortaz aparte, emaitza (espresio bat atomizatua) modu optimoan atomizatua lortzen da.

Horrelako operadore bereziekin lan egiteak nagusiki bi abantaila ditu, alde batetik, ez da denbora galtzen des-atomizazio→deribazio/ordezkatze→atomizazio prozesuan eta, beste aldetik, jatorrizko atomizazioa mantentzen da (azpi-espresio berdinak berdin atomizatuta daude).

Hau dela eta, lan honetan espresio baten deribazio eta ordezkatzeko operazioak gauzatzeko bi algoritmo errekurtsibo garatu dira. Hauek funtzionatzen dute espresio atomizatuak, partzialki atomizatuak eta ez atomizatuak modu berdinean. Algoritmo hauek lan honen autorearen tesian aurkitzen ahal dira.

3.3 Kodearen sorkuntza

Erran den bezala, azken helburua GSDren simulazioak gauzatzeko kode optimoa sortzearena da. Hau da, hainbat funtzio¹² lortu behar dira.

Helburu honekin hurrengo prozedura segitzea gomendatzen da lan honetan:

1. Sinbolikoki funtzioa lortu GSDren printzipioak eta sortu ahalako atomizazioa erabiliz. Funtzio bakoitza espresio sinboliko multzo batek osatzen du.
2. Funtzio hori atomo multzo baten menpekota izan da. Atomo hauek identifikatu behar dira. Kontuan izan behar da atomo bat beste atomoen menpekota izaten ahal dela, beraz prozedura errekurtsibo bat behar da.
3. Funtzioa osatzen duten espresioetan atomo errepikatuak egoten ahal dira. Hauek identifikatu behar dira eta eliminatu.
4. Azkenik, kodea optimizatu asmoz, bakarrik behin agertzen diren atomoak¹³ bere espresioengatik ordezkatzeko. gerta daiteke atomo batzuk bakarrik behin agertzea,

Autorearen tesian prozedura hau segitzeko algoritmo multzo bat proposatzen da.

3.4 Gorputz-anitzeko sistemen topologia: Operadore errekurtsiboak

Orain arte deskribatu diren teknika eta algoritmo guztiak orokorrak izan dira, hau da, GSDren arloan zentratuta badaude ere, beste arlo batean aplikagarriak dira, konputazio sinbolikoa egin behar denean.

Aurkeztutako teknikak (eta batez ere atomizazioa) oraindik osatzen ahal dira GSDren munduak dituen ezaugarriekin. Hain zuzen ere, sistemen topologiari erreparatzen bada ikusten da bere baitan dagoen “errekurtsibitatea” sortu ahalako atomizazioarekin batera kode optimoa lortzeko laguntzen duela.

Helburu honekin proposatzen da ohiko gorpuz-anitzeko sistemen zuhaitz-estruktura (gorputzez edota erreferentziaz osatua. Irudia 2a) bitan zatitzea. Hau dela eta, bi zuhaitz-estruktura definitzen dira, bata puntuz osatua (Irudia 2c) eta bestea basez(orientazioz. Irudia 2b) osatua.

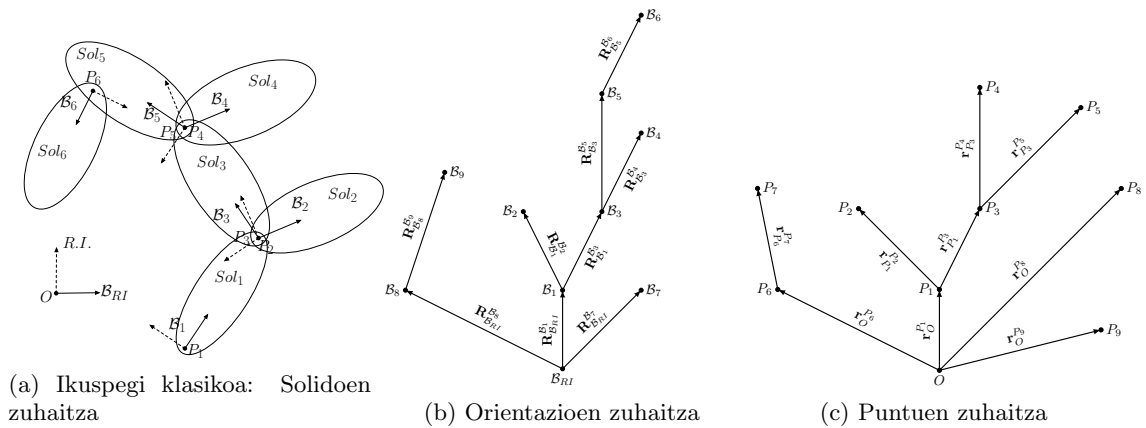
GSDren agertzen diren operazio tipikoak (posizio bektorea, errotazio-matrizeak, abiatura bektoreak eta azelerazio bektoreak) konputatzeko bi estruktura berri hauek erabiltzen diren operadore errekurtsibo multzo bat definitu da. Operadore hauek gai dira sistemak duen “errekurtsibitate” ateratzeko optimizazio guztiak, kontuan hartu gabe zein den erabiltzen ari den formulazioa.

Bi zuhaitzen zatiketaren esker eta aurreko operadoreak erabiliz operazio sinboliko guztiak modu berdinean egiten direla ziurtatzen da. Honek ondorio garrantzitsu bat du, atomoak modu berdinean sortu direla eta, beraz, atomoen “birziklapena” maximizatuko dela. Azken honek lotura zuzena du sortutako kodearen kalitatearekin.

¹² Arlo honetan tipikoak dira hurrengo funtzioak: $\mathbf{M}(\mathbf{q}), \delta(\mathbf{q}, \dot{\mathbf{q}}), \Phi(\mathbf{q}), \dot{\Phi}(\mathbf{q}), \beta(\mathbf{q}), \gamma(\mathbf{q}, \dot{\mathbf{q}})$

¹³ Sortu ahalako atomizazioak atomo aunitz sortzen ditu. Gerta daiteke batuk espresio sinbolikoetan bakarrik behin agertzea.

2 Irudia: Gorputz anitzeko sistemak

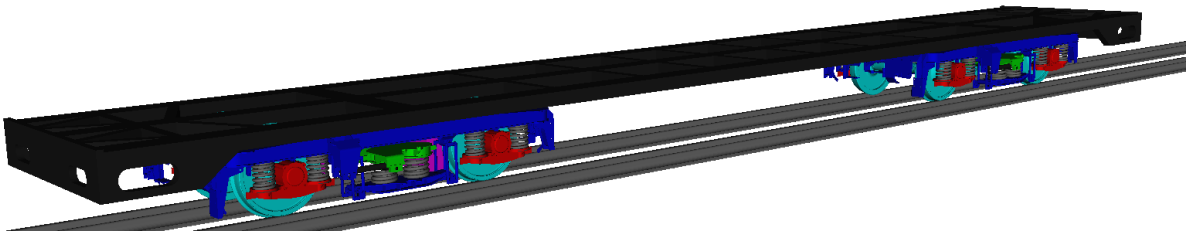


Kontuan hartzen bada sistemaren topologia, 1 irudiak erakusten duen funtzioaren ebaluazio numerikoak orain 264 operazio behar izanen luke. Hasierako modeloarekin konparatuta, kasu hotenean operazio kopurua gutxi gorabehera %11-ra jautsi da.

3.5 Beste adibide bat. Lokomotora bat

Sekzio honetan, beste modelo konplexuago bat aztertzeen da, hain zuzen ere 3 irudiak erakusten duen lokomotora (J. Ros eta Iriarte, 2016). Hau 21 gorputzez osatuta dago 76 koordinatu erlatiboen bidez.

3 Irudia: Lokomotora baten gorputz-anitzeko modeloa.



Kasu honetan, hasierako modeloaren (aurreko teknikak aplikatu gabe) masa matrizea gehi indar orokortuen bektorearen ebaluazioak 19899139 operazio behar ditu. Atomizazioa erabiliz operazio kopurua 636695 operaziotara pasatzen da. Eta gainera, sistemak duen topologia kontuan hartzen bada operazio kopurua jausten da 36695 operaziotara, hau da, %0.2 txikiagoa hasierakoarekin konparatuta. Honek lotura zuzena du konputazio-sistemak ebaluazio numerikoa gauzatzeko beharko duen denborarekin. Haiz zuenen ere, ebaluazioak beharko du hasierako modeloarendako beharko lukeenaren, gutxi gorabehera, denboraren %0.2. Modu berdinean, kodearen fitxategiaren tamaina askoz txikiago da. Hasierako motelaren fitxategiak gutxi gorabehera 250MB okupatzen ditu eta optimizatuak 656kB. Aipagarria da konpiladoreak¹⁴ 250MB-tako fitxategia ezin izan du kapitulatu, honek ere justifikatzen du lan honetan aurkeztutako teknikak.

4 Ondorioak

Lan honen ondorioak aunitz izan badira ere, nagusiak hauek dira:

- Atomizazio prozesua aplikatu beharreko teknika da nahi bada kode optimoa lortzea. Kode honek, optimoa izateagatik, operazio matematiko minimo bat beharko du ebaluazio numerikoa gauza-

¹⁴gcc 4.9.2 konpilatzailea

tzean. Ikusi denez, honek lotura du ebaluazio gauzatzeko beharreko denborarekin eta beraz, denbora-errealean funtzionatzearekin.

- Atomizazioa optimoa izaten ahal da, baldin eta sortu ahalako atomizazioa egiten bada. Baina honek, kodea optimoa sortzeko, behar ditu prozedura bereziak errepikatzen diren atomoak eliminatzeko.
- Atomizatu edo partzialki atomizatuta dauden espresioekin lan egiteko prozedura bereziak behar dira ohikoak diren operazioak (deribazioa eta ordezkatzeta) gauzatzeko, batez ere nahi bada bere abantaila guztiekin lan egitea.
- Sistemaren topologiak garrantzia du, honetan baitago gorputz-anitzeko sistemaren errekurtsibitatea. Bakarrik hau kontuan hartuz lortuko da arlo honetan kode optimizatuta.
- Hemen aurkitzen diren teknika guztiak formulazio eta koordenatu motaren independenteak dira. Beraz, erabiltzaile batek nahi dituen (ezagutzen dituen edota erosoago dagoen) formulazio eta koordenatua erabiltzen ahal ditu eta, hala eta guztiz ere, sistemaren mami guztia aterako du kode optimoa sortuz.

5 Etorkizunerako planteatzen den norabidea

Teknika hauek guztiak probatuak izan dira LIB3D-MEC-GiNaC GSDrako programa-liburutegia sinbolikoan eta espero bezala funtzionatzen dute. Hala ere ikusi izan dira egiten ahal diren hobekuntza batzuk:

- GSDan espresioek trigonometrikoki sinplifikatuak¹⁵ izateko aukera dute. Autoreak bere tesian hau ere analizatu egin du eta metodo bat proposatzen du sinplifikazio trigonometriko posible hauek ez agertzeko. Lan hau momentuz teorikoa da eta garatuta badagoen ere, ez da programa-liburutegia inplementatu.
- Autoreak berak ere ikusi du formulazio batzuekin lortzen den kodea pixka bat optimoagoa dela beste batzuekin baino. Etorkizunean ikertu nahi du horren zergatia eta posible bada soluzio orokor bat bilatzea.

Erreferentziak

- DE JALON, JAVIER GARCIA, eta EDUARDO BAYO. 2012. *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer Science & Business Media.
- FALOMI, STEFANO, MONICA MALVEZZI, eta ENRICO MELI. 2011. Multibody modeling of railway vehicles: Innovative algorithms for the detection of wheelrail contact points. *Wear* 271.453 – 461. Proceedings of the 8th International Conference on Contact Mechanics and Wear of Rail / Wheel Systems, Florence, 2009.
- FEATHERSTONE, ROY. 1983. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research* 2.13–30.
- FISSETTE, PAUL, eta JC SAMIN. 2003. *Symbolic modeling of multibody system*. Kluwer Acad. Pub.
- GIL, J. 2005. *PhD Thesis. Symbolic algebra based preprocessor for multibody dynamics simulation*. Nafarroako Unibertsitate Publikoa.
- J. ROS, A. PLAZA, eta J. P. X. IRIARTE. 2016. Real-time simulation of a locomotive using symbolic multibody methods. *The Third International Conference on Railway Technology: Research, Development and Maintenance. Railways 2016, CIVIL-COMP, CIVIL-COMP, 5-8 April 2016* 2016. .
- J. ROS, L. ARRONDO, J. GIL X. IRIARTE. 2007. Lib3d_mec_ginac, a library for symbolic multibody dynamics. ECCOMAS Multibody Dynamics, Milano.
- KIRĆANSKI, NENAD, MIOMIR VUKOBRATOVIĆ, ALEKSANDAR TIMČENKO, eta MANJA KIRĆANSKI. 1993. Symbolic modeling in robotics: Genesis, application, and future prospects. *Journal of Intelligent & Robotic Systems* 8.1–19.
- PLAZA, A. 2016. *PhD Thesis. Symbolic Multibody Modeling for Real-Time*. Nafarroako Unibertsitate Publikoa.

¹⁵Orokorrean agertzen dira holako azpi-espresioak: $\sin^2(\theta) + \cos^2(\theta)$ eta hau 1-era sinplifikatzen ahal dira. Gauza da ez direla agertzen hain modu esplizituan, baizik eta atomoen artean galduak

- ROS, JAVIER, ROBERTO YOLDI, AITOR PLAZA, eta XABIER IRIARTE. 2013. Real-time hardware-in-the-loop simulation of a hexaglide type parallel manipulator on a real machine controller. In *New Trends in Mechanism and Machine Science*, volume 7 of *Mechanisms and Machine Science*, 587–597. Springer Netherlands.
- SCHMITKE, CHAD, eta PAUL GOOSSENS. 2011. Symbolic computation techniques for multibody model development and code generation. volume 2011.
- SHIIBA, TAICHIAND SUDA, YOSHIHIRO. 2007. Evaluation of drivers behavior with multibody-based driving simulator. *Multibody System Dynamics* 17.195–208.

6 Oharrak

Artikulu honetan laburtzen da autorearen tesiaren zati bat: “Symbolic Multibody Modelling for Real-Time” (Plaza, 2016).