

TEKNILLINEN KORKEAKOULU
Sähkö- ja tietoliikennetekniikan osasto

Olli Haavisto

Kävelevän robottimallin kehittäminen sekä sen datapohjainen mallitus ja säätö

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi
diplomi-insinöörin tutkintoa varten Espoossa 9.2.2004.

Työn valvoja

Professori Heikki Hyötyniemi

Tekijä: Olli Haavisto

Työn nimi: Kävelevän robottimallin kehittäminen sekä sen datapohjainen mallitus ja säätö

Päivämäärä: 9.2.2004

Sivumäärä: 72

Osasto: Sähkö- ja tietoliikennetekniikan osasto

Professuuri: AS-74 Systeemitekniikka

Työn valvoja: Prof. Heikki Hyötyniemi

Kävelevien robottien ohjaus on haastava monimuuttujamenetelmiä vaativa säätöongelma. Tässä työssä käsitellään kaksijalkaisen kävelevän robotin mallinnusta ja säätöä. Aluksi robottimallin dynamiikkayhtälöt johdetaan Lagrange-mekaniikalla ja kehitetään Matlab/Simulink-ympäristöön simulointityökalu mallin simuloimiseksi. Malli saadaan kävelemään ohjaamalla sitä erillisillä PD-säätimillä jatkuvasti päivitettävien referenssisignaalien mukaisesti.

PD-ohjattua kävelystä kerätyn systeemin syöte- ja vastedatan perusteella mallitetaan datapohjaisesti kävelyn käänteinen dynamiikka, eli kuvaus systeemin tiloilta ohjauksille. Käytettävänä mallirakenteena on klusteroitu regressio, jossa kokonaismalli koostuu lokaaleista pääkomponenttiregressiomalleista. Muodostettua mallia sovelletaan robottimallin ohjaamiseen siten, että kävelijän tilaa vastaava regressiomallin ohjausestimaatti kytketään suoraan kävelijän ohjaukseksi.

Työssä havaitaan, että klusteroidulla regressiomallilla pystytään toistamaan PD-ohjattu kävely lähes muuttumattomana. Toisaalta ohjaus osoittautuu melko herkäksi poikkeamille, jotka ajavat systeemiä pois opitun käyttäytymisen alueelta, eikä toiminnan datapohjaista optimointia voida toteuttaa.

Avainsanat: Kaksijalkainen, kävely, robotti, datapohjainen mallitus, klusteroitu regressio, pääkomponenttiregressio, Lagrange-mekaniikka

Author: Olli Haavisto**Name of the Thesis:** Development of a walking robot model and its data-based modeling and control**Date:** 9.2.2004**Number of pages:** 72**Department:** Department of Electrical and Communications Engineering**Professorship:** AS-74 Control Engineering**Supervisor:** Prof. Heikki Hyötyniemi

The control of walking robots is a challenging multivariable problem. This thesis concerns the modeling and control of a biped robot. The first aim of the work is to derive the chosen robot model dynamics using Lagrangian methods and develop a Matlab/Simulink tool for the simulation of the model. To make the model walk, separate PD controllers are used to control the biped according to continuously updated reference signals.

The second aim is to model the inverse dynamics, that is, the mapping from the system output to the input, of the biped gait using input-output-data collected from the PD controlled system. The model structure applied is the clustered regression, which combines several local principal component regression models. The model is then utilized to control the biped so that the control signal estimate corresponding to the current system state is directly used as an input for the system.

It is shown that the clustered regression model can repeat the PD controlled gait with quite good accuracy. The controlled system is, however, relatively sensitive to errors which try to drive it out of the learned state space regions and the optimization of the control is not possible.

Keywords: biped, walking, robot, data-based modeling, clustered regression, principal component regression, Lagrangian mechanics

Alkulause

Tämä työ on tehty Teknillisen korkeakoulun Systeemitekniikan laboratoriossa jatkona jo kesällä 2002 aloitettuun datapohjaisen mallintamisen ja ohjauksen tutkimukseen.

Kiitän työn valvojaa professori Heikki Hyötyniemeä hänen antamastaan kannustavasta ohjauksesta ja neuvoista sekä mahdollisuudesta diplomityön tekemiseen.

Laboratorion koko henkilökuntaa kiitän myönteisen ja innostavan työskentelyilmapiirin luomisesta. Lisäksi haluan kiittää vanhempiani ja veljeäni saamastani taustatuesta.

Espoossa 9.2.2004

Olli Haavisto

Sisältö

Symboliluettelo	3
1 Johdanto	5
2 Kaksijalkaisten kävelevien robottien mallintaminen ja ohjaaminen	7
2.1 Mallintaminen	7
2.2 Passiivinen dynaaminen kävely	7
2.3 Optimaaliset liikeradat	8
2.4 Neuroverkkopohjainen ohjaus	8
2.5 Geneettinen ohjelmointi	9
3 Kävelijän simulointi	11
3.1 Malli	11
3.2 Mallin yhtälöt	13
3.3 Simulink-toteutus	15
3.3.1 Kävelijän dynamiikka	15
3.3.2 Alustan tukivoimat	16
3.3.3 Polvikulmien rajoittimet	17
3.4 Simulaattorin käyttöliittymä	17
3.5 Kävelijän parametrit	18
4 PD-ohjaus	20
4.1 Säätimet	20
4.2 Referenssisignaalit	20
4.3 Parametrit	22
4.4 Kävelyliike	23
4.5 Simulink-toteutus	24
5 Paikallinen oppiminen	26
5.1 Taustaa	26
5.2 Paikallisesti painotettu regressio	27
5.3 Lokaalit mallit	27
5.4 Esimerkkejä ja sovelluksia	28
6 Klusteroitu regressiosäätö	30
6.1 Periaate	30
6.2 Lokaalien pääkomponenttiregressiomallien muodostus	31
6.3 Ohjauksen laskenta	31
6.4 Optimointi	32
6.4.1 Optimisäätö	32
6.4.2 Dynaaminen ohjelmointi	33
6.4.3 Klusteroidun regressiorakenteen optimointi	34

6.5	Aikaisempia sovelluksia	35
7	Klusteroidun regressiosäätäjän soveltaminen kävelijän ohjaukseen	37
7.1	Säätäjän Simulink-toteutus	37
7.2	Opetusdata	39
7.3	Datan klusterointi	40
7.4	Toimintapisteiden ja piirremuuttujien lukumäärien valinta . . .	43
7.5	Toimintapisteiden sisäisten mallien opetus	45
7.6	Opittu kävely	45
7.7	Adaptiivinen opetus	50
7.8	Opetuksen toistaminen	51
8	Johtopäätökset	53
A	Lagrange-mekaniikka	58
A.1	Yleistetyt koordinaatit	58
A.2	Lagrangen yhtälöt	58
B	Dynamiikkamallin yhtälöt	61
C	Monimuuttujaregression menetelmiä	65
C.1	Pääkomponenttianalyysi	65
C.2	Pääkomponenttiregressio	67
C.3	Hebbin ja anti-Hebbin oppimiseen perustuva pääkomponentti- regressioalgoritmi (HAH)	67
C.3.1	Neuroverkon rakenne ja toiminta	68
C.3.2	Pääkomponenttianalyysi	71
C.3.3	Pääkomponenttiregressio	72

Symboliluettelo

$A(q)$	Kävelijän dynamiikkayhtälöiden inertiamatriisi
$b(q, \dot{q}, M, F)$	Kävelijän dynamiikkayhtälöiden oikea puoli
C_u^p	Toimintapisteen p paikka tila-avaruudessa
C_y^p	Toimintapisteen p paikka ohjausavaruudessa
D	Mielivaltainen $n \times n$ ortogonaalimatriisi
$f(\xi(k), v(k))$	Yleisen systeemin tilansiirtofunktio
F	Alustan tukivoimien vektori
F_{q_i}	Yleistettyyn koordinaattiin q_i liittyvä yleistetty voima
g	Maan putoamiskiihtyvyys
G	Regressiokuvaus
h	Diskretionnin näyteväli
H	Kustannuksen painomatriisi
I_n	$n \times n$ -kokoinen yksikkömatriisi
J	Kustannuskriteeri
k	Diskreetti aikaindeksi
K^p	p :nnen toimintapisteen ohjausestimaatin painokerroin
l	Säätimen ulostulovektorin dimensio
l_0, l_1, l_2	Kävelijän jäsenten pituudet
m	Säätimen syötevektorin dimensio
m_0, m_1, m_2	Kävelijän jäsenten massat
M	Momenttivektori
n	Pääkomponenttien lukumäärä
N	Opetusdatan näytteiden lukumäärä
N^p	Toimintapisteeseen p liittyvien opetusnäytteiden lukumäärä
N_{op}	Toimintapisteiden lukumäärä
p	Toimintapisteen indeksi
p^*	Parhaan toimintapisteen indeksi
P_{xx}^p	Piirredatan kovarianssin käänteismatriisi (toimintapiste p)
q	Yleistettyjen koordinaattien vektori
δq_r	Koordinaatin q_r suuntainen virtuaalinen poikkeama
r_0, r_1, r_2	Kävelijän massojen etäisyydet nivelistä
R_{xu}^p	Signaalien x ja u välinen ristikovarianssi (toimintapiste p)
$R(\cdot)$	Robotin dynamiikkamalli
s_L, s_R	Vasemman (L) ja oikean (R) jalan kosketussensorin arvo
T	Liike-energia
$u(k)$	Skaalattu ja nollakeskiarvoistettu säätimen syötevektori
$u_{\text{real}}(k)$	Systeemin tila eli säätimen syötevektori
$\hat{u}(k)$	Rekonstruoitu säätimen syötevektori
$\tilde{u}(k)$	Säätimen syötevektori, josta on poistettu sensoriarvot
U	Säätimen syötevektoridata

$v(k)$	Ohjaus
w_i	i :s pääkomponenttivektori
W	Pääkomponenttikanta
δW	Virtuaalinen työ
$x(k)$	Piirremuuttujavektori
(x_0, y_0)	Kävelijän ylävartalon massakeskipisteen koordinaatit
X	Piirremuuttujadata
$y(k)$	Skaalattu ja nollakeskiarvoistettu ohjaus
$y_{\text{real}}(k)$	Säätimen ulostulovektori eli systeemin ohjaus
Y	Systeemin ohjausdata
$z(k)$	Valkaistu piirremuuttujavektori
α	Kävelijän ylävartalon kulma
β_L, β_R	Kävelijän vasemman ja oikean jalan reiden kulma
$\Delta\beta$	Reisikulmien erotus $\beta_R - \beta_L$
γ_L, γ_R	Kävelijän vasemman ja oikean jalan polvikulma
θ	Robotin nivelen kulma
Θ	Ominaisvektoreitten muodostama matriisi
λ	Unohduskerroin
Λ	Ominaisarvomatriisi
μ_k, μ_s	Liike- ja lepokitkakertoimet
ϕ	Pääkomponenttikannan transpoosi
σ^2	Ohjausestimaattien painotusfunktion varianssi
σ_n^2	Naapuruusfunktion varianssi
τ	Jatkuva aikamuuttuja
$\xi(k)$	Yleisen systeemin tila
$\mathbf{0}_n$	$n \times n$ -kokoinen nollamatriisi

1 Johdanto

Kävelevien robottien ohjaamisessa joudutaan tyypillisesti ratkaisemaan hyvin haastavia ongelmia. Robottien monimutkainen mekaaninen rakenne johtaa mallinnuksessa hankaliin epälineaarisiin dynamiikkayhtälöihin, jolloin systeemien matemaattinen käsittely vaikeutuu. Säättöongelmana kävelyn ohjaaminen vaatii monimuuttujamenetelmien käyttöä, sillä toimilaitteita on yleensä paljon ja ristikkäisvaikutukset eri säätösuureiden välillä ovat voimakkaita.

Pelkästään kaksijalkaisiakin kävelijöitä on tutkittu runsaasti ja erilaisia ohjausmenetelmiä kehitetty paljon. Perinteisempien, valmiiksi laskettuihin liikeratoihin perustuvien säätöalgoritmien rinnalle ovat nousseet esimerkiksi neuroverkkoja käyttävät datapohjaiset lähestymistavat.

Datapohjaisessa mallinnuksessa systeemin toimintaa pyritään mallintamaan pelkästään ohjaus- ja vastedatan perusteella, jolloin laitteen sisäistä rakennetta ei tarvitse tuntea. Mallirakenteen valinta vaikuttaa voimakkaasti mallinnuksen onnistumiseen, mutta myös käytetyn datan pitää sisältää riittävästi informaatiota, jotta mallinnus olisi mahdollista.

Monimutkaisten systeemien kuvaamisessa ongelma kannattaa yleensä jakaa pienempiin osiin, joiden rakenne pysyy yksinkertaisempana. Koko systeemiä kuvaava malli voidaan siis muodostaa useiden, helposti analysoitavissa olevien mallien yhdistelmänä. Näin mallitus on *skaalautuvaa*: Samalla menetelmällä pystytään käsittelemään yhtä hyvin yksinkertaisia kuin monimutkaisempiakin systeemeitä.

Tässä työssä sovelletaan kahta eri mallitusmenetelmää. Ensimmäisenä tavoitteena on muodostaa kaksijalkaisen kävelijän dynaaminen simulointimalli lasquemalla systeemin tarkat dynamiikkayhtälöt Lagrange-mekaniikalla. Mallin simulointi toteutetaan Matlab/Simulink-ympäristössä.

Simulointimallin ja siihen liitetyn yksinkertaisen säätäjän avulla simuloidaan kävelyä ja kerätään systeemistä dataa. Toisena tavoitteena työssä onkin tutkia paloittain lineaarisen regressiorakenteen eli klusteroidun regression käyttöä kävelijän dynamiikan datapohjaiseen mallintamiseen ja näin muodostetun mallin soveltuvuutta systeemin ohjaukseen. Lisäksi selvitetään mahdollisuuksia ohjauksen optimointiin mallia päivittämällä.

Klusteroitua regressiota on tarkoitus käyttää systeemin käänteisen dynamiikkamallin muodostamiseen. Systeemin tilan ja tarvittavien ohjausten välinen yhteys tietyllä liikeradalla tallennetaan suoraan mallirakenteeseen, jolloin mitattua tilaa vastaavaa ohjausestimaattia voidaan sellaisenaan käyttää ohjaukseen. Ohjausmenetelmän yksinkertaisen rakenteen ja datapohjaisen oppimisen perusteella klusteroitua regressiosäätöä voidaan verrata biologisten systeemien toimintaan: muistiin opitusta mallista saadaan pelkän mittaussignaalin perus-

teella tilanteeseen sopiva ohjaus.

Vastaavaa lähestymistapaa on aikaisemmin käytetty yksinkertaisen robottikäsivarren ohjaamiseen [1], ja tulokset osoittivat menetelmän toimivaksi. Kävelijän ohjaaminen on kuitenkin huomattavasti haastavampi ongelma, koska systeemin dynamiikka on monimutkaisempi ja vaihtelee voimakkaasti kävelysyklin aikana.

Työ jakautuu osiin siten, että luvussa 2 esitellään kirjallisuuteen perustuen erilaisia menetelmiä, joita on käytetty kaksijalkaisten kävelevien robottien mallintamiseen ja ohjaukseen. Luvut 3 ja 4 kuvaavat työssä kehitetyn kävelijän mallin ja PD-ohjauksen. Datapohjaisen mallinnuksen teoriaa ja soveltamista kävelijän mallin ohjaamiseen käydään läpi luvuissa 5–7 ja koko työn tulokset kootaan yhteen luvun 8 yhteenvedossa.

2 Kaksijalkaisten kävelevien robottien mallintaminen ja ohjaaminen

Kaksijalkaisten kävelevien robottien mallintamiseen ja ohjaamiseen on sovellettu useita eri lähestymistapoja. Yksinkertaisimpia laitteita ovat passiiviset kävelijät, joiden toiminta perustuu systeemin sopivaan mekaaniseen rakenteeseen eikä ulkoisia ohjaussignaaleita tarvita. Toista äärilaitaa edustavat esimerkiksi Honda-yhtymän kehittämät monimutkaiset ja voimakkaita ohjauksia vaativat humanoidirobotit, joiden liikkeet koettavat matkia ihmisen käyttäytymistä mahdollisimman tarkasti.

Tässä luvussa esitellään aluksi kaksijalkaisten kävelijöitten mallinnusta ja luodaan sitten katsaus erilaisiin menetelmiin, joita on käytetty kävelijöitten ohjaamiseen.

2.1 Mallintaminen

Mallinnuksessa tarkoituksenmukaista on keskittyä robotin oleelliseen rakenteeseen ja toimintoihin. Melko usein kaksijalkaisten kävelijöitten toiminta rajoitetaan teoreettisen käsittelyn helpottamiseksi kaksiulotteiseen tilanteeseen, jossa kävelyä tarkastellaan robotin etenemissuuntaan nähden sivulta päin.

Kävelyliike voidaan jakaa karkeasti kahteen eri osaan: *Kaksoistukivaiheessa* kävelijän molemmat jalat koskettavat maahan painon siirtyessä taaemmalta jalalta etummaiselle. *Heilahdusvaiheessa* vain tukijalka koskettaa maahan ja taaempi jalka heilahtaa eteen. Näitä vaiheita toistamalla voidaan kuvata koko kävely kummankin jalan toimiessa vuorotellen tukijalkana. Jos halutaan mallintaa myös juoksua, täytyy lisäksi ottaa huomioon tilanne, jossa kumpikaan jalka ei ole kiinni maassa.

Kävelijöitten ohjaukseen käytetään yleensä robotin niveliin kytkettäviä momentteja. Säädön toteuttaminen ei kuitenkaan ole suoraviivaista, koska lähes aina systeemillä on enemmän vapausasteita kuin säädettävissä olevia suureita. Lisäksi tilannetta vaikeuttaa systeemin dynamiikan voimakas muuttuminen kävelijän siirtyessä yhdestä kävelyn vaiheesta toiseen.

2.2 Passiivinen dynaaminen kävely

Passiivinen dynaaminen kävely perustuu pelkästään kävelijän dynaamisen rakenteen hyödyntämiseen. Kävelyyn ei tarvita ulkopuolista energiaa, mutta passiiviset kävelijät pystyvät säilyttämään stabiilin, toistuvan kävelyliikkeen vain loivasti alaspäin viettävillä alustoilla. Näiden laitteiden kävelyssä toinen jalka

heilahtaa omalla painollaan vapaasti eteenpäin toisen jalan tukiessa systeemiä maahan. Heilahduksen lopuksi paino siirtyy jalalta toiselle, jolloin toinen jalka vuorostaan heilahtaa eteen.

McGeer [2, 3] osoitti, että sopivasti rakennettu kaksijalkainen laite pystyy kävelemään loivaa alamäkeä ilman aktiivista ohjausta. Tämän jälkeen erilaisia passiivisia kävelijöitä on rakennettu ja aihetta tutkittu runsaasti (esim. [4, 5]). Yksinkertaistenkin laitteiden kävely on hyvin ”luonnollisen” näköistä — varsinakin, jos kävelijällä on myös polvinivelet. Esimerkiksi ihminen käyttää kävelyssään suurelta osin hyödyksi vartalon ja jalkojen dynaamista rakennetta, jolloin kävely vaatii mahdollisimman vähän energiaa.

Puhtaasti passiivisten kävelijöitten suurin rajoitus on niiden kyky kävellä vain alamäkeä. Lisäämällä systeemiin heikkotehoinen ohjaus voidaan saada aikaan kävelijä, joka säilyttää stabiilin kävelyliikkeen myös tasaisella tai loivassa ylämäessä [6, 7], mutta jonka energiankulutus on lähellä minimiä.

2.3 Optimaaliset liikeradat

Etukäteen laskettuja optimaalisia liikeratoja käytetään yleisesti kävelijöiden ohjaamiseen. Tässä lähestymistavassa systeemiä pyritään ohjaamaan siten, että sen osien liikkeitä noudattavat valmiiksi talletettuja referenssiliikeratoja. Ongelmana on kuitenkin tasapainon säilyttäminen esimerkiksi epätasaisella alustalla tai muuten epäideaalisissa olosuhteissa. Menetelmät tasapainon säilyttämiseen voidaan jakaa kahteen ryhmään: *staattisiin* ja *dynaamisiin kävelijöihin*. Staattisten kävelijöitten pystyssäpysyminen perustuu siihen, että laitteen painopiste pidetään jatkuvalla ohjauksella kohtisuorasti tukijalan tai tukijalkojen peittämän alustan osan päällä. Menetelmä vaatii voimakkaita ohjauksia ja johtaa usein hitaaseen ja kömpelöön kävelyyn. Toisaalta kävelijä on aina stabiilissa tilassa, eli pysyy pystyssä vaikka liike pysäytettäisiin.

Dynaamisten kävelijöitten tasapainon säilyttämisessä puolestaan tarkastellaan sen alustan pisteen sijaintia, jonka kautta systeemin kokonaistukivoiman pystysuora komponentti kulkee (Zero Moment Point, ZMP). Pisteen tulee sijaita aina mahdollisimman keskellä tukipintaa, ja jos piste siirtyy tukipinnan reunalle, alkaa laite kaatua. Esimerkiksi Honda-roboteissa [8] tasapainon säilyttäminen perustuu tämän pisteen halutun ja todellisen sijainnin jatkuvaan säätämiseen.

2.4 Neuroverkkopohjainen ohjaus

Neuroverkkojen vahvuutena on niiden kyky mallintaa monimutkaisiakin epälineaarisia funktioita. Niitä on sovellettu kaksijalkaisten kävelijöitten ohjaukseen

useissa tutkimuksissa, yleensä suorittamaan jotakin tiettyä laskentaa säätäjän sisällä. Tällainen tehtävä voi olla esimerkiksi käänteisen kinematiikan laskenta [9]. Värähtelevää neuroverkkoa tai neuroniryhmää (neural oscillator) voidaan käyttää kävelijän jaksollisten malliliikeratojen muodostamiseen [10] tai suoraan nivelten momenttien laskentaan [11].

Neuroverkkoja käytetään myös adaptiivisissa systeemeissä. Tyypillisesti verkolle opetetaan systeemin epälineaarinen dynamiikka, jota sitten hyödynnetään säätäjän toiminnassa [12].

2.5 Geneettinen ohjelmointi

Geneettisten algoritmien periaate matkii luonnossa tapahtuvaa evoluutiota paremman ratkaisun muodostamiseksi tiettyyn ongelmaan. Ratkaisu, esimerkiksi joukko parametrien arvoja, pitää aluksi koodata merkkijonoksi. Näitä ratkaisuehdotuksia (*yksilöitä*) muodostetaan satunnainen joukko eli *populaatio*, joka toimii algoritmin alkutilana. Kunkin ratkaisun hyvyys arvioidaan *hyvyysfunktion* avulla, joka antaa ratkaisuun liittyvän *hyvyysarvon*. Mitä suurempi hyvyysarvo, sitä parempi ratkaisu on.

Populaation parhaat yksilöt tuottavat seuraavan sukupolven kolmella eri menetelmällä: Kopioituvat sellaisenaan uuteen populaatioon, kopioituvat hieman muunnettuna tai tuottavat parina kaksi jälkeläistä, joiden rakenne on sekoitus kummastakin alkuperäisestä yksilöstä. Uuden populaation syntyyn vaikuttavat yksilöt valitaan hyvyysarvojen perusteella, mutta valinnassa on myös satunnaisuutta. Tällä pyritään välttämään hyvyysfunktion lokaaleihin minimeihin ajautuminen.

Geneettinen ohjelmointi [13] yhdistää automaattisen ohjelmoinnin ja geneettiset algoritmit. Kehittyvän yksilöjoukon eli populaation muodostavat erilaiset tietokoneohjelmat, joiden hyvyys määräytyy niiden ajettaessa antamista tuloksista. Kahden ohjelman risteytyksessä satunnaisia osia ohjelmista vaihdetaan keskenään, jolloin syntyy ongelman mahdollisesti paremmin ratkaisevia ohjelmia.

Koska vähitellen kehittyvät ohjausalgoritmit saattavat tuottaa myös robotin rakenteelle vaarallisia ohjauksia, käytetään ohjelmien hyvyysarvojen laskentaan yleensä aluksi simulaattoria. Esimerkiksi Sigel-simulaattori [14] mahdollistaa erilaisten robottirakenteiden simuloinnin ja geneettisen ohjelmoinnin kokeilemisen. Käyttämällä simuloinneissa todellisten robottien simulointimalleja voidaan saadut ohjausalgoritmit siirtää todellisiin systeemeihin, jos simulaattori pystyy kuvaamaan robotin riittävän tarkasti [15].

Ohjausalgoritmien kehittyminen vaatii todella paljon simulointikertoja, koska kaikkien ohjausyritysten hyvyys pitää arvioida simuloimalla. Todellisten fysi-

kaalisten systeemien simulointi on yleensä hidasta, joten hyvän ohjauksen löytäminen voi kestää kauankin. Lisäksi ongelmia aiheuttavat simulaatiomallin ja todellisen systeemin eroavuudet.

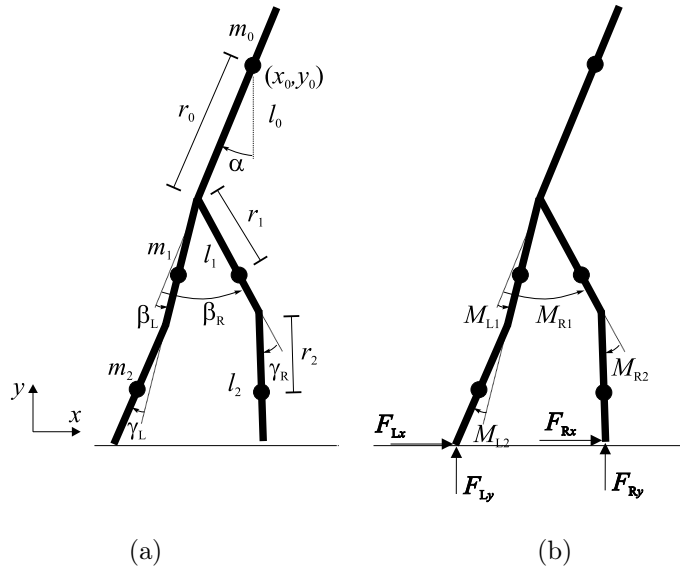
3 Kävelijän simulointi

Tässä työssä ensimmäisenä tavoitteena oli kehittää kävelevä robottimalli, jota simuloimalla voidaan kokeilla erilaisia ohjausmenetelmiä ja kerätä kävelevästä systeemistä dataa. Malliksi valittiin kaksijalkainen kävelijä, jonka dynamiikkayhtälöt johdettiin Lagrange-mekaniikalla. Varsinainen simulointi suoritettiin Matlabin Simulink-ympäristössä, ja simulointitulosten tarkasteluun kehitettiin graafinen käyttöliittymä.

Seuraavassa kuvataan aluksi systeemin tarkka malli sekä dynamiikkayhtälöiden muodostaminen. Tämän jälkeen käydään läpi mallin Simulink-toteutus ja simuloinneissa käytetyt parametriarvot. Tarkempi kuvaus Simulink-mallista sekä graafisesta käyttöliittymästä sisältyy erilliseen dokumentaatioon [16].

3.1 Malli

Työssä käytetty systeemimalli kuvaa voimakkaasti yksinkertaistettua kaksijalkaista kävelijää. Simuloinnin nopeuttamiseksi ja laskennan helpottamiseksi malli toteutettiin kaksikulotteisena, jolloin kulkusuuntaan nähden sivuttainen liike voitiin jättää huomiotta. Mallin identtiset jalat koostuvat jäykistä sääri- ja reisiosista, jotka on yhdistetty polvinivelillä. Ylävartalon muodostaa yksi jäykkä kappale, joka kiinnittyy jalkoihin lonkkanivelillä. Kuvassa 1(a) on esitetty kävelijän rakenne sekä systeemin tilan kuvaamisessa käytetyt muuttujat.



Kuva 1: Systeemin muuttujat ja vakiot (a) sekä ulkoiset voimat ja momentit (b).

Tarkasteltavan systeemin asennon ja paikan kuvaamiseksi kaksiulotteisessa koordinaatistossa tarvitaan vähintään seitsemän muuttujaa, eli systeemillä on seitsemän vapausastetta. Koordinaattipari (x_0, y_0) määrittää ylävartalon massakeskipisteen paikan ja kulma α poikkeaman y-akselin suunnasta. Vasemman (L) ja oikean (R) jalan asennot ylävartaloon nähden kuvataan lonkka- ja polvinivelten kulmilla $(\beta_L, \beta_R, \gamma_L, \gamma_R)$.

Kävelijän ylävartalon sekä reisien ja säärien pituudet määräytyvät kuvan 1(a) mukaisesti parametreista l_0, l_1 ja l_2 . Ylävartalon massakeskipisteen (massa m_0) etäisyys lantiosta on r_0 . Kummankin reiden massakeskipisteen (massa m_1) oletetaan sijaitsevan lonkan ja polven kautta kulkevalla suoralla etäisyydellä r_1 lonkkanivelestä. Vastaavasti säärien massakeskipisteet (massa m_2) sijaitsevat polven ja jalan kärjen kautta kulkevalla suoralla etäisyydellä r_2 polvesta.

Kävelyalustan mallintamiseksi molempien jalkojen päihin on mahdollista vaikuttaa vaaka- ja pystysuuntaisilla ulkoisilla voimilla $(F_{Lx}, F_{Ly}, F_{Rx}, F_{Ry})$, jolloin erilaisten alustamateriaalien ja epätasaisten alustojen käyttö simuloinnissa on helppoa (kuva 1(b)). Ulkoiset voimat muodostetaan säätäjällä, joka kytkeytyy päälle jalan osuessa alustaan. Varsinaisina ohjaussignaaleina mallissa ovat ylävartalon ja reisien väliset momentit (M_{L1}, M_{R1}) sekä polvinivelten momentit (M_{L2}, M_{R2}) .

Jalkojen vuorovaikutus alustan kanssa toteutetaan siis erillisillä ulkoisilla voimilla, jolloin sama dynamiikkamalli kuvaa kävelijää kaikissa tilanteissa. Tämä sallii simulointimallin käytön mielivaltaisten liikkeiden simuloinnissa. Lisäksi dynamiikkamalli on *holonominen*, eli mikään systeemin osa ei voi kohdata liikkuessaan ulkopuolista mekaanista rajoitusta. Toinen mahdollinen lähestymistapa olisi käyttää erillisiä malleja riippuen maata koskettavien jalkojen lukumäärästä ja olettaa, että maahan koskettava jalka ei pääse liukumaan. Näin tarvittavat mallit olisivat yksinkertaisempia, mutta siirtymät mallien välillä pitäisi laskea erikseen.

Työssä käytettyä robottimallia vastaavia kaksijalkaisia kävelijöitä on tutkittu runsaasti. Rakenteeltaan täysin vastaava on esimerkiksi RABBIT-robotti [17], jolle on myös johdettu simulointimalli. Tämän robotin mallinnuksessa on kuitenkin oletettu, että kävelyssä heilahtavan jalan osuessa maahan toinen jalka nousee välittömästi ilmaan. Näin yhdellä mallilla on voitu kuvata kaikki kävelyn vaiheet, mutta tukijalan vaihtuminen aiheuttaa aina erikseen laskettavan askelmaisen muutoksen systeemin tilassa. Samoin simulointimallin toiminta on rajoitettu toistamaan askeleita vuorotellen, eikä esimerkiksi molempien jalkojen yhtäaikaista kosketusta maahan ole simuloitavissa.

3.2 Mallin yhtälöt

Kävelijän dynamiikan mallinnus toteutettiin Lagrange-tekniikalla (liite A). Systemin tila määräytyy yleistetyistä koordinaateista

$$q = [x_0, y_0, \alpha, \beta_L, \beta_R, \gamma_L, \gamma_R]^T \quad (1)$$

ja niiden aikaderivaatoista. Jokaiseen koordinaattiin liittyy vastaava yleistetty voima:

$$F_q = [F_{x_0}, F_{y_0}, F_\alpha, F_{\beta_L}, F_{\beta_R}, F_{\gamma_L}, F_{\gamma_R}]^T. \quad (2)$$

Merkitään reisien massakeskipisteiden paikkoja karteesisissa koordinaateissa (x_{L1}, y_{L1}) ja (x_{R1}, y_{R1}) . Säärien massakeskipisteitten sijainnit ovat (x_{L2}, y_{L2}) ja (x_{R2}, y_{R2}) ja jalkojen päitten koordinaatit (x_{LG}, y_{LG}) ja (x_{RG}, y_{RG}) . Vasemman jalkaan liittyvät koordinaatit voidaan lausua yleistettyjen koordinaattien avulla seuraavasti:

$$\begin{cases} x_{L1} &= x_0 - r_0 \sin \alpha - r_1 \sin(\alpha - \beta_L) \\ y_{L1} &= y_0 - r_0 \cos \alpha - r_1 \cos(\alpha - \beta_L) \\ x_{L2} &= x_0 - r_0 \sin \alpha - l_1 \sin(\alpha - \beta_L) - r_2 \sin(\alpha - \beta_L + \gamma_L) \\ y_{L2} &= y_0 - r_0 \cos \alpha - l_1 \cos(\alpha - \beta_L) - r_2 \cos(\alpha - \beta_L + \gamma_L) \\ x_{LG} &= x_0 - r_0 \sin \alpha - l_1 \sin(\alpha - \beta_L) - l_2 \sin(\alpha - \beta_L + \gamma_L) \\ y_{LG} &= y_0 - r_0 \cos \alpha - l_1 \cos(\alpha - \beta_L) - l_2 \cos(\alpha - \beta_L + \gamma_L). \end{cases} \quad (3)$$

Oikean jalan vastaavat koordinaatit saadaan korvaamalla yhtälöissä (3) vasemman jalan kulmat β_L ja γ_L oikean jalan kulmilla β_R ja γ_R .

Systemin liike-energia voidaan helposti lausua karteesisissa koordinaateissa eri massapisteiden liike-energioiden summana:

$$T = \frac{1}{2} (m_0(\dot{x}_0^2 + \dot{y}_0^2) + m_1(\dot{x}_{L1}^2 + \dot{y}_{L1}^2 + \dot{x}_{R1}^2 + \dot{y}_{R1}^2) + m_2(\dot{x}_{L2}^2 + \dot{y}_{L2}^2 + \dot{x}_{R2}^2 + \dot{y}_{R2}^2)). \quad (4)$$

Kutakin yleistettyä koordinaattia q_r vastaavan yleistetyn voiman lauseke F_{q_r} johdetaan kasvattamalla koordinaatin arvoa *virtuaalisen poikkeaman* δq_r verran ja pitämällä muut yleistetyt koordinaatit vakioina. Kaikkien systemiin vaikuttavien voimien muutoksessa tekemä *virtuaalinen työ* δW_{q_r} riippuu seuraavan yhtälön mukaisesti halutusta voimasta:

$$\delta W_{q_r} = F_{q_r} \delta q_r. \quad (5)$$

Yleistettyjen voimien yhtälöiksi yleistettyjen koordinaattien suhteen saadaan

näin

$$\begin{cases} F_{x_0} &= F_{Lx} + F_{Rx} \\ F_{y_0} &= -(m_0 + 2m_1 + 2m_2)g + F_{Ly} + F_{Ry} \\ F_{\alpha} &= -\left(\frac{\partial y_{L1}}{\partial \alpha} m_1 + \frac{\partial y_{L2}}{\partial \alpha} m_2 + \frac{\partial y_{R1}}{\partial \alpha} m_1 + \frac{\partial y_{R2}}{\partial \alpha} m_2\right)g + \frac{\partial y_{LG}}{\partial \alpha} F_{Ly} \\ &\quad + \frac{\partial y_{RG}}{\partial \alpha} F_{Ry} + \frac{\partial x_{LG}}{\partial \alpha} F_{Lx} + \frac{\partial x_{RG}}{\partial \alpha} F_{Rx} \\ F_{\beta_L} &= -\left(\frac{\partial y_{L1}}{\partial \beta_L} m_1 + \frac{\partial y_{L2}}{\partial \beta_L} m_2\right)g + \frac{\partial y_{LG}}{\partial \beta_L} F_{Ly} + \frac{\partial x_{LG}}{\partial \beta_L} F_{Lx} + M_{L1} \\ F_{\gamma_L} &= -\frac{\partial y_{L2}}{\partial \gamma_L} m_2 g + \frac{\partial y_{LG}}{\partial \gamma_L} F_{Ly} + \frac{\partial x_{LG}}{\partial \gamma_L} F_{Lx} + M_{L2}. \end{cases} \quad (6)$$

Oikeaan jalkaan liittyvät voimat saadaan korvaamalla vasemman jalan voimien lausekkeissa suuret oikean jalan vastaavilla suureilla.

Systeemin dynamiikkayhtälöitten ratkaisemiseksi lausutaan kineettinen energia (4) yleistettyjen koordinaattien avulla käyttäen muunnoskaavoja (3). Saatu lauseke sekä yleistettyjen voimien lausekkeet (6) sijoitetaan Lagrangen yhtälöihin, jotka ovat muotoa

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_r} \right) - \frac{\partial T}{\partial q_r} = F_{q_r}. \quad (7)$$

Saatu seitsemän yhtälön toisen asteen differentiaaliyhtälöryhmä voidaan kirjoittaa matriisimuotoon

$$A(q)\ddot{q} = b(q, \dot{q}, M, F), \quad (8)$$

missä

$$M = [M_{L1}, M_{R1}, M_{L2}, M_{R2}]^T \quad (9)$$

ovat malliin vaikuttavat ulkoiset momentit ja

$$F = [F_{Lx}, F_{Ly}, F_{Rx}, F_{Ry}]^T \quad (10)$$

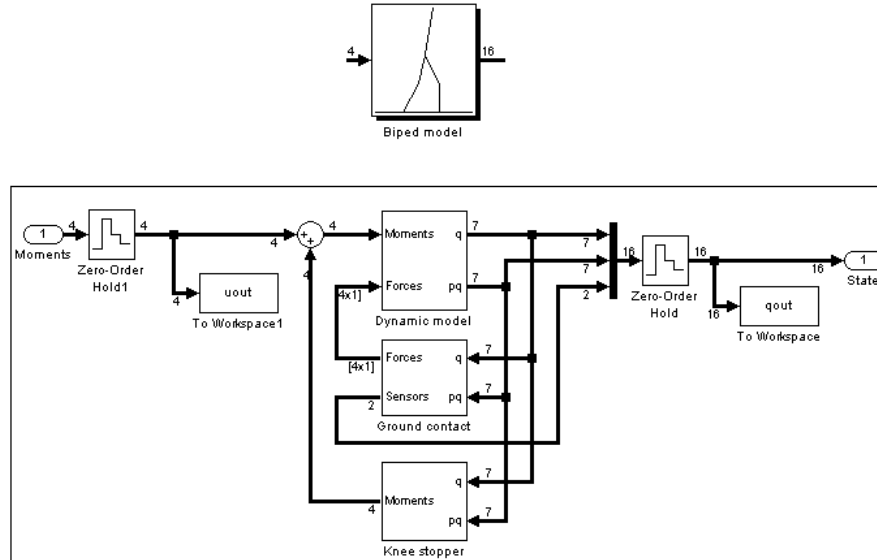
alustasta aiheutuvat tukivoimat kuvan 1(b) mukaisesti. Pystyvektori $b(q, \dot{q}, M, F)$ sisältää korkeintaan ensimmäisen asteen aikaderivaattoja yleistetyistä koordinaateista. Inertiamatriisi $A(q)$ ei sisällä yleistettyjen koordinaattien aikaderivaattoja.

Lagrangen yhtälöitten (7) mekaaninen muodostaminen ja kirjoittaminen lopulliseen muotoon (8) suoritettiin Mathematica-ohjelmistolla. Saadut lausekkeet $A(q)$ -matriisin alkioille sekä vektorille $b(q, \dot{q}, M, F)$ muunnettiin tämän jälkeen Matlab-muotoon, jotta niitä voitiin soveltaa simuloinnissa.

Inertiamatriisin $A(q)$ ja vektorin $b(q, \dot{q}, M, F)$ alkioitten lausekkeet ovat liitteenä B.

3.3 Simulink-toteutus

Kävelijän dynamiikkamallin simulointiin käytettiin Matlabin Simulink-ympäristöä, jossa mallin tilan ja tarvittavien tukivoimien laskenta voitiin koota yhdeksi lohkoksi (kuva 2).



Kuva 2: Kävelijän Simulink-malli muodostuu dynamiikkayhtälöiden, alustan tukivoimien ja polvikulmien rajoitinmomenttien laskentalohkoista.

Biped model -lohkon syötesignaalina on pystyvektori, joka sisältää mallin ohjaukseen käytettävät momentit (9). Ulostulosignaaliin on koottu yleistetyt koordinaatit (1), niiden ensimmäinen aikaderivaatta sekä kummankin jalan kosketusanturin arvo: $[q^T, \dot{q}^T, s_L, s_R]^T$. Jos jalka koskettaa maahan, nousee vastaava kosketussignaali (s_L, s_R) ykköseen. Jalan ollessa ilmassa signaalin arvo on nolla.

Kävelevää systeemiä sekä sen kosketusta alustaan simuloidaan jatkuvassa tilassa. Systeemin säätöön sovelletaan kuitenkin diskreettejä säätäjiä, joten ohjaus- ja ulostulosignaalit diskretoidaan nollannen kertaluvun pidolla. Lohko myös tallettaa diskreetit ulostulosignaalin ja syötesignaalin Matlabin työtilaan. Simuloinnissa käytettävät parametrit ja kävelijän alkutila syötetään lohkon maskin valintaikkunaan.

3.3.1 Kävelijän dynamiikka

Dynamiikkayhtälöitten (8) simulointia suorittava lohko näkyy kuvassa 3. Koska kiihtyvyysvektorin $\ddot{q}$ ratkaiseminen suljetussa muodossa ei käytännössä on-

[illegible]

3.3.2 Alustan tukivoimat

Tangentiaalisuunnassa otetaan huomioon alustan kitkaominaisuudet. Kun jalka osuu maahan, säädetään sen poikkeamaa osumiskohdasta PD-säätimellä, jonka ulostulona on tangentiaalinen voima F_t . Jos tarvittava voima kuitenkin ylittää suurimman mahdollisen kitkavoiman

missä μ_s on alustan lepokitkakerroin, alkaa jalka liukua. Tällöin tangentiaalinen tukivoima määräytyy liikekitkasta

missä μ_k on alustan liikekitkakerroin.

16