

LAPPEENRANNAN-LAHDEN TEKNILLINEN YLIOPISTO  
School of Energy Systems  
Konetekniikka

*Tuomas Huuskonen*

**TULOSIGNAALIN MUOKKAUS –ALGORITMIT TRUKIN MASTON  
HUOJUNNAN HALLINTAAN**

31.8.2020

Tarkastajat: Professori Jussi Sopanen & TkT Eerik Sikanen

## **TIIVISTELMÄ**

Lappeenrannan-Lahden Teknillinen yliopisto  
School of Energy Systems  
Konetekniikka

Tuomas Huuskonen

### **TULOSIGNAALIN MUOKKAUS –ALGORITMIT TRUKIN MASTON HUOJUNNAN HALLINTAAN**

Diplomityö

2020

65 sivua, 28 kuvaa, 4 taulukkoa and 4 liitettä

Tarkastajat: Professori Jussi Sopenen & TkT Eerik Sikanen

Avainsanat: tulosignaalin muokkaus, värähtelyn vaimennus, MATLAB, Simulink, simulaatio, työntömastotrukki

Tämän työn tavoitteena on tutkia ja simuloida erilaisten tulosignaalin muokkausmenetelmien soveltuvuutta ja tehokkuutta automaattisen työntömastotrukin työntöliikkeeseen hallitsemaan värähtelyjä. Tarvetta edellä mainitulle teknologialle on, sillä korkeissa nostoissa joudutaan tällä hetkellä jopa odottamaan, että huojuminen vähenee, ennen kuin taakka voidaan laittaa hyllyyn tai ottaa sieltä pois.

Työssä tutkitaan kuutta eri matemaattista algoritmia Matlabin ja Simulinkin avulla simuloiden. Värähtelyn vaimennustehokkuus, algoritmin nopeus ja mallinnusvirheiden sietokyky ovat keskeisiä kriteerejä ja lopulta päädyttiin Zero Vibration –algoritmiin, joka jakaa liikekäsken kahteen impulssiin. Sopivasti ajoitettuna impulssit vaimentavat ohjausliikkeestä johtuvat jäännösvärähtelyt täysin. Tämän tekniikan soveltuvuutta myös manuaalikäyttöön pohditaan. Lopuksi simuloidaan kaksi identtistä kuormankäsittelysykliä useilla eri nostokorkeuksilla ja massoilla. Simulaatiot ajetaan signaalinmuokkausalgoritmi päällä ja pois päältä, jotta saadaan vertailukelpoisia tuloksia tarkempaa analyysiä varten.

Alle yhdeksän metrin nostokorkeuksissa lähes kaikki odottaminen saatiin simulaatioissa eliminoitua ja korkeampien nostojen värähtelyä vaimennettua jopa 69 %. Kuormankäsittelysykliä, joka on kuormankäsittelyyn kuluva ajamisesta pois oleva aika, saatiin lyhennettyä simuloidussa testitapauksessa lähes 30 %, kun työntöliikkeen signaali muokattiin korkeassa ja suuri massaisessa nostossa.

## **ABSTRACT**

LUT University  
School of Energy Systems  
Mechanical Engineering

Tuomas Huuskonen

## **INPUT SHAPING – ALGORITHMS FOR REACH TRUCK MAST SWAY REDUCTION**

Master's thesis

2020

65 pages, 28 figures, 4 tables and 4 appendices

Examiners: Professor Jussi Sopanen & D. Sc. (Tech.) Eerik Sikanen

Keywords: input shaping, sway, reduction, MATLAB, Simulink, simulation, reach truck

The aim of this thesis is to research and simulate different kinds of input shaping algorithms and their applicability and effectiveness for automatic reach truck mast sway control. This technology is definitely needed, since while doing high lifts a waiting period becomes necessary due to the inaccuracy of pallet placement if the mast is swaying.

This thesis evaluates six different mathematical algorithms by simulating with Matlab and Simulink. Residual vibration level, level of delay and robustness of the algorithms are evaluated against each other and Zero Vibration –shaper was chosen with the fore mentioned criteria. ZV-shaper divides the given impulse into two impulses latter of which is delayed. If this is done according to the correct natural frequency and damping ratios, the residual vibration is reduced ideally to zero. The applicability of this technique is also assessed for the manual reach truck. In the end, two identical load handling cycles are simulated at different heights and masses. One with shaping and one without shaping. The results of these simulations are analyzed and compared.

At lift heights below nine meters almost all waiting time was eliminated in the simulations. Even at the higher lifts, the waiting time was reduced by over 69 %. Load handling cycle, which is the time the load is handled between stopping and accelerating again, was shortened by almost 30 % in the high lift height and high mass - simulation case by using input shaping.

## ALKUSANAT

Aluksi haluan kiittää koko LUT-yliopiston henkilökuntaa koko yliopistoaikani jatkuneesta asiantuntevasta opetuksesta ja aivan mahtavasta opiskeluympäristöstä. Erityisesti haluan kiittää Professori Jussi Sopasta erittäin tehokkaasta diplomityön ohjauksesta, joka mahdollisti tämän työn valmistumisen todella nopealla aikataululla, vaikka COVID19-pandemia aiheutti sekasortoa samaan aikaan. Lisäksi haluan kiittää tutkijatohtori Eerik Sikasta hänen tarjoamistaan työkaluista ja yleisestä avusta tämän työn aihepiirissä.

Haluan kiittää Timo Heineä ja Mikael Jaakkolaa Mitsubishi Logisnext Oy:stä tämän mielenkiintoisen diplomityöaiheen tarjoamisesta ja ohjauksesta sen alusta loppuun. Erityisesti kiitos Timolle teknisestä tuesta, minulle alkuun vieraan teorian kanssa. Kiitos myös Mikaelille avusta ja tuesta työn tiimoilta.

*Tuomas Huuskonen*

Järvenpää 14.8.2020

## SISÄLLYSLUETTELO

<b>TIIVISTELMÄ .....</b>	<b>1</b>
<b>ABSTRACT.....</b>	<b>2</b>
<b>ALKUSANAT .....</b>	<b>3</b>
<b>SISÄLLYSLUETTELO .....</b>	<b>5</b>
<b>SYMBOLI- JA LYHENNELUETTELO.....</b>	<b>7</b>
<b>1 JOHDANTO.....</b>	<b>9</b>
1.1 Mitsubishi Logisnext Europe.....	10
1.2 Signaalin muokkaus ja värähtelyn vaimennus.....	11
1.3 Tarvekartoitus .....	12
1.4 Tavoitteet .....	12
1.4.1 Tutkimusongelmat ja –kysymykset.....	13
1.4.2 Hypoteesi ja rajaukset.....	13
<b>2 LASKENTA JA SIMULAATIOMENETELMÄT .....</b>	<b>15</b>
2.1 Algoritmit.....	15
2.1.1 Zero Vibration ja distributed delay.....	16
2.1.2 Zero Vibration and Derivative and Derivative .....	20
2.1.3 Extra Insensitive .....	22
2.1.4 Unity Magnitude Zero Vibration.....	23
2.2 Vertailu .....	24
2.2.1 Vasteaika .....	24
2.2.2 Virheensietokyky.....	26
2.2.3 Jäännösvärähtely.....	28
2.2.4 Vertailutulokset .....	29
2.3 Projektin tarvekartoitus.....	31
2.3.1 AGV.....	31
2.3.2 Manuaalikäyttö .....	32
2.4 Dynaamisen järjestelmän simulointi.....	33
2.4.1 Dynaamisen heilurijärjestelmän mallinnus .....	34
2.4.2 Ohjausjärjestelmä ja visualisointi.....	36
2.5 Pysähtymismatkaratkaisu.....	39

<b>3</b>	<b>TULOKSET .....</b>	<b>42</b>
3.1	Vertailuajokierros .....	45
3.2	Tapaus 1: ei signaalin muokkausta .....	46
3.3	Tapaus 2: ZV-menetelmä.....	47
3.4	Vertailu ja data.....	48
<b>4</b>	<b>POHDINTA.....</b>	<b>52</b>
4.1	Simulaatioanalyysi.....	52
4.2	Kontrollerin sykliajan vaikutus.....	53
4.3	Jarrun vaikutus huojumiseen .....	55
4.4	Herkkyysanalyysi.....	58
4.5	Jatkotutkimusmahdollisuudet .....	59
<b>5</b>	<b>JOHTOPÄÄTÖKSET.....</b>	<b>61</b>
	<b>LÄHTEET .....</b>	<b>64</b>

## **LIITTEET**

LIITE I: Lastin käsittelysyklin simulaatiotuloksia

LIITE II: Kuormankäsittelysykliä simulaatiomalli

LIITE III: Päätasen simulaatiomallin ajotiedosto

LIITE IV: Heilurimallin vaste askelsignaaliin

## SYMBOLI- JA LYHENNELUETTELO

### Symbolit

$\beta$	Jaetun viiveen yläraja
$\varepsilon$	Aika [s]
$\zeta$	Vaimennussuhde
$\Phi$	Yksinkertaistusmuuttuja
$\omega$	Taajuus [rad/s]
$\omega_n$	Ominaistaajuus [rad/s]
$\omega_d$	Vaimennettu taajuus [rad/s]
$A_i$	Muokkausmenetelmien osa-amplitudit
$c$	vaimennusvakio [N/(m/s)]
$c_c$	Kriittinen vaimennus [N/(m/s)]
$C(\zeta, \omega)$	Jäännösvärähtelyn rajoite
$D(s, \beta)$	Siirtofunktio jaetulle viiveelle
$F(t)$	Herätevoima [N]
$K$	Yksinkertaistusmuuttuja
$k$	Jousivakio [N/m]
$m_s$	Nostettava massa [kg]
$S(\zeta, \omega)$	Jäännösvärähtelyn rajoite
$t_i$	Muokkausmenetelmäkohtainen aikapaikka [s]
$T$	Jaksonaika [s]
$V$	Jäännösvärähtely [%]
$V_{tol}$	Jäännösvärähtelytoleranssi [%]
$w(\varepsilon)$	Jaetun viiveen kulmakerroin
$y(t)$	Jaettu viive
$z_s$	Siirtymä tasapainopisteestä
$z_r$	Herätepoikkeutus

### Lyhenteet

AGV	Automatic Guided Vehicle, Automaattitrucki
ART	Automated Reach Truck, Automaattityöntömastotrucki

EI	Extra Insensitive
DZV	Distributed Zero Vibration
LHC	Load Handling Cycle, Kuormankäsittelysykli
ML	Mitsubishi Logisnext
MLE	Mitsubishi Logisnext Europe
UMZV	Unity Magnitude Zero Vibration
ZV	Zero Vibration
ZVD	Zero Vibration Derivative



## 1 JOHDANTO

Trukit ovat kaupallisten tuotteiden logistiikkaketjujen avainkomponentteja. Niillä lastataan kuorma-autoja, laivoja ja lentokoneita, joilla tapahtuu pitkän matkan kuljetus. Nykypäivänä, kiihtyvän digitalisaation johdosta, on myös trukkimarkkina automatisoitunut ja on syntynyt automaattitrukkeja, tai englanniksi Automated Guided Vehicles (AGV). Ennen, kun tilaus tai noutava kuorma-auto saapui varastolle, hyppäsi trukkikuski trukkiinsa ja lähti etsimään tilaukseen kuuluvia hyödykkeitä. Automaattitrukeilla tilanne on hyvin erilainen. Trukit tietävät koko ajan missä tavarat ovat, sekä mikä on nopein reitti tavarain luokse. Kun tilaus saapuu, herää automaattikone valmiustilastaan ja suuntaa suoraan kohti haluttua lavaa, jonka se kuljettaa nopeinta mahdollista reittiä jatkokuljetukseen.



**Kuva 1.** Työntömastotrukki on saanut nimensä maston työntöliikkeestä. Kuvassa RoclA Automated Reach Truck (ART). (RoclA AGV. 2020.)

Trukkeja on todella monenlaisiin käyttötarkoituksiin. Tässä työssä keskitytään suurimmalta osalta työntömasto-tyyppisiin trukkeihin, jollaisen voi nähdä kuvassa 1. Työntömastotrukissa masto voi liikkua myös lineaarisesti eteen- ja taaksepäin, perinteisten ylös, alas ja kallistusliikkeiden lisäksi. Kun varastoiden läpi kulkee enemmän ja enemmän hyödykkeitä, on varastohyllyjen oltava todella korkeita. Tämä tarkoittaa tietysti, että myös trukkien mastojen on oltava todella korkeita, jopa yli 10-metrisiä. Mikä tahansa 10-metrinen ja ainoastaan alhaalta tuettu kappale, jossa on vielä massa huipulla, huojuu varmasti. Erityisesti, jos sitä pitää liikutella. Huojumisesta ei käytännössä voi päästä eroon, joten sitä on yritettävä kontrolloida.

Huojunta työntömastotrukin mastossa voi muodostua ongelmaksi siirtyessä kohti maksiminostokorkeuksia. Erityisesti kun liikkeitä ja pieniä korjauksia tehdään lastin ollessa nostettuna lähelle suurinta nostokorkeutta. Tämä diplomityö keskittyy ratkaisemaan tätä ongelmaa teorian tasolla matemaattisesti. Ratkaisua lähdetään hakemaan tulossignaalin muokkaus –nimisellä menetelmällä, joka kattaa monia erilaisia algoritmeja. Kaikissa algoritmeissa on samankaltainen periaate, mutta ne on toteutettu toisistaan poikkeavin tavoin.

Tässä kappaleessa selitetään signaalin muokkauksen peruseriaate ja esitetään taustatietoa aiheesta. Tarvetta tälle teknologialle kartoitetaan lyhyesti, jotta rajaukset, tutkimuskysymykset ja tavoitteet saadaan selitettyä. Kappaleessa 2 erilaisia algoritmeja vertaillaan ja parhaat valitaan jatkosimulointia varten. Lisää tarvekartoitusta ja projektin mahdollisia hyötyjä tutkitaan myös pidemmälle. Tulokset kappaleessa esitetään simulaatioiden tuloksia erilaisilla vaimennusmenetelmillä ja vertaillaan niitä vaimentamattomaan ja vapaasti huokuvaan tapahtumaan. Näitä tuloksia ja niiden luotettavuutta analysoidaan lisää kappaleessa 4, jossa käydään läpi myös jatkotutkimuskohteita sekä pohditaan aihealueen ympäriltä käytännönongelmia. Viidennessä kappaleessa kerätään tästä työstä tärkeimmät tulokset ja vedetään yhteen kaikki maininnan arvoiset tulokset tästä projektista.

## 1.1 Mitsubishi Logisnext Europe

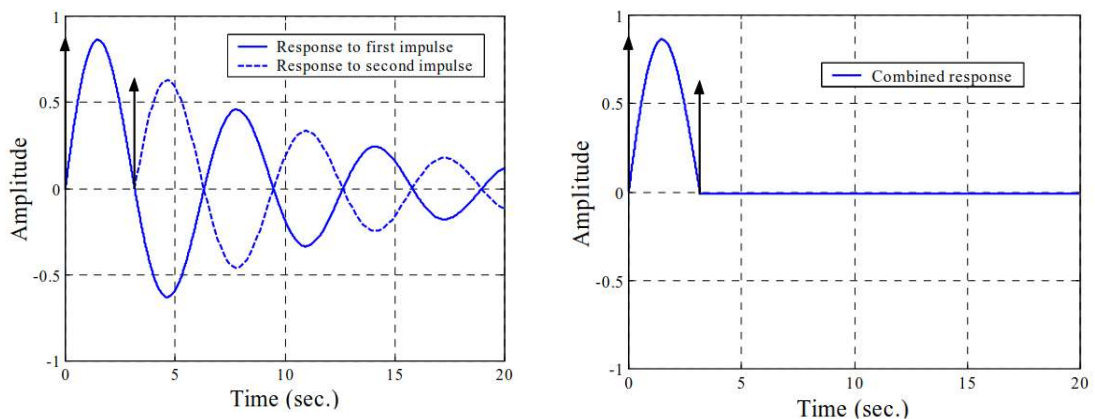
Mitsubishi Logisnext Europe Oy (MLE) on osa Mitsubishi Logisnext (ML) -yhtymää, joka perustettiin jo 1937. ML vaikuttaa Amerikassa, Aasiassa, Tyynen valtameren alueella, Euroopassa, Lähi-Idässä and Afrikassa. Johtavana toimittajana logistiikan ja materiaalin käsittelyratkaisuihin, ML toimittaa laitteisto- ja ohjelmistoratkaisuja logistiikkamarkkinoille ympäri maailman. (MLE 2020.)

1.4.2020 alkaen entinen Rocla Oy tunnetaan nyt nimellä Mitsubishi Logisnext Europe Oy. Tämä fuusio muiden ML:in omistamien Euroopan yksiköiden kanssa yhdisti tuotanto ja tuotekehitys –toiminnot saman hallinnon alle Euroopassa. Nimi Rocla on kuitenkin vielä brändinimenä muun muassa automaattitrukki ratkaisuihin.

## 1.2 Signaalin muokkaus ja värähtelyn vaimennus

Varhaisimmat signaalin muokkausmenetelmät kehitettiin vuosina 1950–1960. Tämän aallon peruuttamiskonseptin kehitti O.J.M Smith, mutta se ei levinnyt laaja-alaiseen käyttöön. Ongelmana tuolloin oli mikroprosessoriteknikan puute ja menetelmän herkkyys ominaistajuuden ja vaimennussuhteiden mallivirheille. Vuonna 1990 Singer ja Seering julkaisivat signaalin muokkaamisesta tutkimuksen, jossa he lisäsivät rajoiteyhtälöitä herkkyysongelman torjumiseksi. Tämä tutkimus uudisti kiinnostuksen referenssisyöttöjen esisuodatukseen ja aloitti laajemman leviämisen muokkaustapojen käytössä ja tutkimuksessa. Nykyään erilaisia signaalin muokkausmuotoja on monissa järjestelmissä, esimerkiksi kiintolevyissä, nostureissa ja teollisuusroboteissa. (Kang 2011.)

Tulosignaalin muokkaus on tekniikka värähtelyn vaimentamiseen avoimessa järjestelmässä. Peruseriaatteena on, että alkuperäinen tulosignaali siepataan, muokataan ja syötetään toimilaitteelle uudessa muodossaan. Koska tämä tapahtuu avoimessa järjestelmässä, takaisinkytkentää ei ole. Signaali lähetetään portaittain sopivalla viiveellä ja amplitudilla, jotka riippuvat rakenteen ominaistajuudesta ja vaimennussuhteesta. Signaalin amplitudi pysyy samana muotoilun jälkeen, mutta impulssit, joilla se on toteutettu, vaimentavat toistensa impulssien aiheuttaman värähtelyn. Kuva 2 havainnollistaa käskyn osiksi jakamista kun käytössä on yksinkertainen, ”Zero Vibration” (ZV) –muokkain. Oikean puoleisessa kuvassa on teoreettisella tasolla järjestelmän vaste vasemman puolen käskyillä.



**Kuva 2.** Yksittäinen impulssi jaettuna kahdeksi impulssiksi. Järjestelmän kokonaisvaste impulssihin näkyy oikealla. (Poty et al. 2003.)

### 1.3 Tarvekartoitus

Huojunnanvaimennus on välttämätön AGV-maailmassa. Päälimmäinen syy AGV-ratkaisun ostamiselle ja rakentamiselle johonkin varastoon, on saada hyödykkeille luotettavampi, nopeampi, johdonmukaisempi ja ennustettavampi siirtokyky. AGV-vaunun kuormankäsittelysykli (LHC) määritellään myöhemmin tarkemmin, mutta yksinkertaisesti sen voi ajatella seuraavasti: Aika, joka trukilta kuluu kuorman nostamisessa, hyllylle laittamisessa, maston laskemisessa ja uudelleen liikkeelle lähtemisessä. Tai päinvastoin. Käytettäessä yli 10 metriä korkeaa mastoa, heiluminen on suuri osa kuormankäsittelysykliä ja tällä hetkellä on välillä odotettava heilahtelun vaimenemista hyväksyttävälle tasolle ennen kuin jatkotoimenpiteitä voidaan suorittaa. Siksi mikä tahansa kuormankäsittelysykliä lyhentävä järjestelmä on tarpeen ja hyödyllinen. Tulosignaalin muokkaus ei edellytä avoimena järjestelmänä toteutettuna mitään ylimääräisiä laitteita. Siksi se on erittäin kustannustehokas ja koodin yksinkertaisuuden myötä, alkulaskelmien ja taulukointien jälkeen, erittäin kevyt ja helppo kontrollerille suorittaa.

Yrityksen suunnalta järjestelmän kehittäminen vaatii jonkin verran työtunteja, mutta todennäköisesti lyhentää odotusaikaa ja siten kuormankäsittelysykliä huomattavasti suhteellisen pienillä lisäkustannuksilla. Asiakas saa AGV-järjestelmäänsä tuottavuutta lisäävän tekniikan, joka on tietysti rahan arvoinen. Markkinoinnissa järjestelmästä on ainoastaan hyötyä, sillä parantaahan se suorituskykyä ja turvallisuutta, joka on aina hyvä asia. Suhteellisen pienillä muunnoksilla ja testaamisella järjestelmän voisi mahdollisesti myös jälkiasentaa vanhoihin järjestelmiin, sen yksinkertaisuuden ja helpon muokattavuuden takia.

### 1.4 Tavoitteet

Tämä opinnäytetyö keskittyy työntömastotrukin 11-metrisen maston värähtelyyn ja sen hallinnoimiseen matemaattisesta näkökulmasta. Tämä tarkoittaa käytännössä erilaisten muokkausalgoritmien tutkimista ja testaamista MATLAB Simulinkissä. Käytössä on myös Simscape Multibody -lisäosa, jolla saadaan tehtyä reaaliaikainen visualisointi simuloitaville tapahtumille. Ensisijainen tavoite on tutkia signaalinmuokkausta laajemmin ja syvällisemmin, jotta sen käyttömahdollisuuksia eri käyttötarkoituksissa saadaan kartoitettua.

Muutamiiin jo havaittuihin ongelmakohtiin keskitytään tarkemmin, sekä esimerkiksi käyttö automaattitrukissa on vielä tutkimatta kokonaisuudessaan.

#### 1.4.1 Tutkimusongelmat ja –kysymykset

Tässä kappaleessa listataan tutkimuskysymykset ja –ongelmat, joiden pohjalta aihetta lähdetään tutkimaan.

Tutkimusongelmat tässä työssä ovat seuraavat:

- Materiaalinkäsittelysykli ei ole niin tehokas kuin se voisi olla ja sykli sisältää odotusperiodin huojumisen vähenemiseksi.
- Turvallisuutta voidaan parantaa, jos suuren massan yli 10 metrin korkeudessa tapahtuvaa huojuntaa saadaan vähennettyä.
- Pysähtymismatkan ramppi vaikea todentaa manuaalikäytössä. Tämä saattaa johtaa törmäykseen radan päätyyn.
- Kiihtyvyydet ja liikenopeudet ovat mahdollisesti määriteltävä uudelleen signaalinmuokkauksen jälkeen.

Tutkimuskysymykset voidaan määritellä seuraavasti:

- Mikä algoritmi olisi paras tähän käyttötarkoitukseen?
- Kuinka materiaalinkäsittelysykli määritellään ja minkälaisiin osiin se saadaan jaettua?
- Mikä osuus materiaalin käsittelysyklistä antaa suurimman parannukset?
- Onko materiaalinkäsittelysykli nopeampi signaalin muokkauksen ja liikkeiden optimoinnin jälkeen?
- Kuinka toteuttaa pysähtymisrampin approksimointi manuaalikäyttöön?
- Kuinka pysähtyminen radan päätyyn tehdään mahdollisimman tehokkaasti manuaalikäytössä?

#### 1.4.2 Hypoteesi ja rajaukset

Mitä luultavimmin materiaalinkäsittelysykli nopeutuu ja värähtelyä saadaan vähennettyä. Alustavan tutkimuksen ja aiheeseen tutustumisen perusteella, signaalin muokkaus on erittäin

hyvä vaihtoehto matalataajuisten ja hidastiikkisten systeemien vaimentamiseen. Suurien nostokorkeuksien ja isojen nostettavien massojen takia, ominaistajuudet tulevat olemaan tässä työssä melko matalia. AGV-käyttöön ainoastaan syklin nopeuttaminen on mielekäs tulos. Kuitenkin niin, ettei nopeuksia lähdetä nostamaan turvallisuuden tai luotettavuuden kustannuksella. Tämän lisäksi manuaalikäytössä käyttäjätuntumalla on suuri vaikutus, mutta sitä ei lähdetä tämän työn puitteissa tutkimaan.

Tämä työ keskittyy pääasiassa matemaattisiin algoritmeihin ja vertailee niitä. Maston taajuudet ja vaimennusarvot on mitattu aiemmin, joten nämä arvot ovat käyttövalmiita, eikä lisämittauksia mastosta tarvita. Yhtä valittavaa algoritmia tutkitaan tarkemmin näillä mittarvoilla. Huomioon otetaan vain ensimmäisen asteen ominaistajuudet, sillä korkeammat moodit eivät ole aiheuttaneet ongelmia käytössä, eikä niitä ole lähitaajuusalueilla. Simulointia käytetään tärkeimpänä visualisoinnin ja todentamisen lähteenä, joten prototyyppiä ei tarvitse rakentaa.

## 2 LASKENTA JA SIMULAATIOMENETELMÄT

Parhaan algoritmin löytäminen on yhdistelmä riittävän hyvää jäännösvärähtelyn tasoa, riittävän nopea vasteajan kanssa. Tavoitteena on saada masto huojumaan niin vähän kuin mahdollista, pitäen samalla liikekäskeyn vasteaika järkevänä. Muuten odottamisongelma vain siirretään liikekäskeyn ramppiin: tällöin odottaisimme huojumisen vaimenemisen sijaan, että signaali nousee tai laskee. Näin onkin jo löytynyt kaksi kriteeriä vertailuun: vasteaika ja vaimennustaso verrattuna alkuperäisen signaalin aiheuttamaan jäännösvärähtelyyn.

Tässä luvussa esitellään myös simulointimenetelmät ja mallit. Algoritmi on rakennettu MATLAB R2019b -sovellukseen käyttämällä Simulinkia, joka on graafinen ympäristö, jolla voidaan mallintaa, analysoida ja simuloida dynaamisia järjestelmiä. Se on erittäin monipuolinen ja kykenevä järjestelmä, joka käyttää helposti ymmärrettäviä lohkokaaevioita. MATLAB toimii myös kokonaisuudessaan taustalla, joten jos lohkokaaevioilla ei voida tehdä jotain, se voidaan käynnistää MATLABin .m -koodista.

### 2.1 Algoritmit

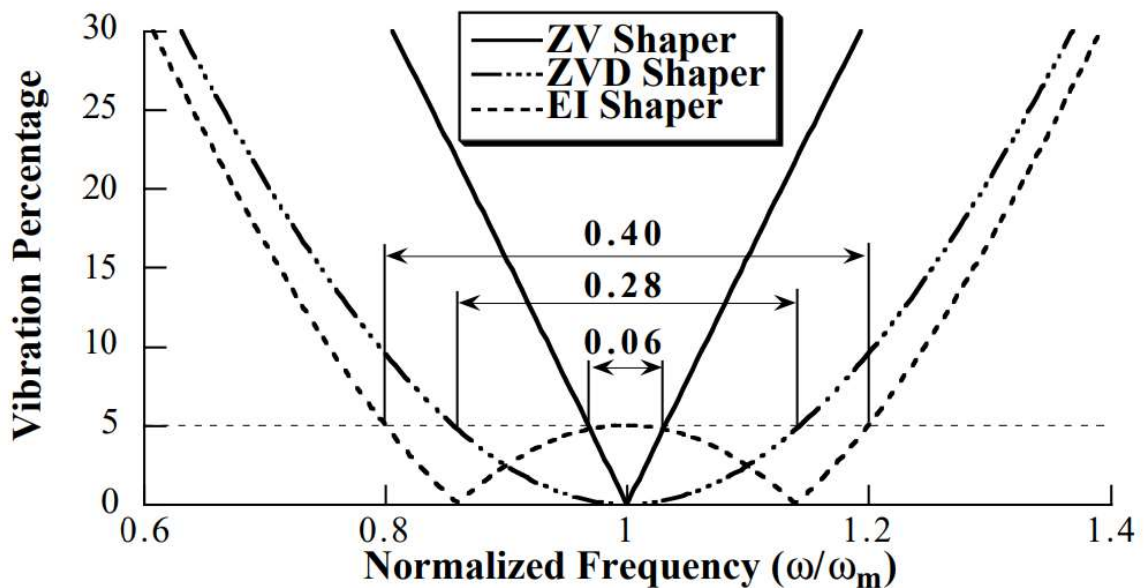
Eri algoritmeilla on erilaiset vahvuudet ja heikkoudet. Tässä luvussa algoritmeja arvioidaan ja verrataan ominaisuuksiltaan, joita pidetään tärkeimpinä tässä sovelluksessa. Yksi pääkriteereistä nousu- ja laskuajan, sekä jäännösvärähtelytason ohella on virheensietokyky. Jälkimmäinen tarkoittaa algoritmin kykyä vaimentaa värähtelyä, vaikka ominaistajuus eroaa mallille annetusta. Laskelmat enemmän derivoiduille signaalinmuokkaimille ovat erittäin pitkiä ja monimutkaisia; siksi niitä ei sisällytetä tähän työhön niiltä osin, joilta niitä ei tarvita vertailun kannalta.

Kaiken perustana ohjearvon muokkaamisessa on ominaistajuus,  $\omega = \sqrt{k/m}$ , jossa  $k$  on jousivakio ja  $m$  on huojuva massa. Ominaisajuuden perusteella lasketaan, minkälaisilla pulsseilla jäännösvärähtely-yhtälöstä saadaan tulokseksi nolla. Kun myös vaimennussuhde on tiedossa, voimme jo yksinkertaisimmillaan ratkaista kahden pulssin sekvenssin, joilla jäännösvärähtely on nolla. Vaimennussuhde on kahden vaimennuskertoimen suhde:  $\zeta = \frac{c}{c_c}$ , jossa kriittinen vaimennus,  $c_c = 2\sqrt{km}$  ja  $c$  on rakenteesta laskemalla selviävä vaimennusarvo.

Useimmiten on kuitenkin helpointa mitata vaimennussuhde suoraan rakenteesta, jolloin saadaan varmasti luotettava ja systeemiä kuvaava arvo.

### 2.1.1 Zero Vibration ja distributed delay

Yksinkertaisin näistä menetelmistä on nimeltään Zero Vibration (ZV) –muokkain. Se toimii jakamalla alkuperäisen signaalin kahteen osaan ja syöttämällä toisen pulssin heti ja toisen puolen jaksonajan kuluttua, jolloin systeemi on juuri värähtämässä toiselle puolelle referenssiin nähden. Teoriassa tämä on yksinkertaista ja helppoa, mutta menetelmä on hyvin herkkä virheille ominaistajuuden ja vaimennussuhteiden mallinnuksessa. Lisäksi kaikki epälineaarisuudet järjestelmän käyttäytymisessä aiheuttavat jäännösvärähtelyjä, jos käytössä on ZV-muokkain, kuten kuva 3 osoittaa. (Singhose et al. 1997.)



**Kuva 3.** Herkkyyskäyrät ZV, ZVD (Zero Vibration Derivative) and EI (Extra Insensitive) –muokkaimille. Kuva ilmaisee kuinka suuri virhe oikean ja mallinnetun taajuuden välillä voi olla, jotta pysytään alle 5 % jäännösvärähtelyssä. (Singhose et al. 1996.)



Jos tiedämme järjestelmän ominaistajuuden ja vaimenemissuhteen, jäännösvärähtelyjä annetun käskyn jälkeen voidaan kuvata seuraavan yhtälön kuvaamalla tavalla (Singh 2002.):

$$V(\zeta, \omega) = e^{-\zeta \omega t} \sqrt{C(\zeta, \omega)^2 + S(\zeta, \omega)^2} \quad (1)$$

Jossa:

$$C(\zeta, \omega) = \sum_{i=1}^n A_i \cdot e_i^{\zeta \omega t_i} \cdot \cos(\omega_d \cdot t_i) \quad (2)$$

$$S(\zeta, \omega) = \sum_{i=1}^n A_i \cdot e_i^{\zeta \omega t_i} \cdot \sin(\omega_d \cdot t_i) \quad (3)$$

Yhtälöissä (2) ja (3)  $A_i$  ja  $t_i$  ovat amplitudi ja aikapaikka jo mainituille signaaleille,  $n$ :n ollessa impulssien määrä. Vaimennettu taajuus saadaan yhtälöstä:

$$\omega_d = \omega \sqrt{1 - \zeta^2} \quad (4)$$

Yhtälö (1) on jäännösvärähtelyprosentti, joka kertoo paljonko impulssisarja aiheuttaa järjestelmälle värähtelyä yhteen summattuun impulssiin verrattuna. Määrittämällä yhtälön (1) nollaan, voimme ratkaista amplitudit ja aikapaikat, jotka johtavat jäännösvärähtelyarvoon nolla. Muutama lisärajoite on vielä määriteltävä, jotta nämä yhtälöt saadaan ratkaistua. (Singh 2002.):

$$\sum A_i = 1 \quad (5)$$

$$A_i > 0, i = 1, 2, \dots, n \quad (6)$$

Nyt, kun yhtälön (1) asettaa nolleen, sekä yhtälöt (5) ja (6) pätevät, voimme ratkaista amplitudit  $A_1$ ,  $A_2$  ja aikapaikat  $t_1$ ,  $t_2$  jotka johtavat nollan jäännösvärähtelyyn. Totta kai,  $t_1$  voidaan asettaa nollassi, koska ensimmäinen sysäys annetaan loogisesti ajanhetkellä 0. Seuraava impulssi voidaan antaa puolen jaksonajan kuluttua, jolloin järjestelmä on värähtämässä aloitussuunnasta vastakkaiseen suuntaan kuten kuva 2 aiemmin osoitti, joten:

$$t_2 = \frac{\pi}{\omega_d} \quad (7)$$

Nyt kun aikapaikat ovat tiedossa, on enää laskettava amplitudit. Ne ratkeavat seuraavista yhtälöistä:

$$A_1 + A_2 = I \quad (8)$$

$$A_1 - (I - A_1) \cdot e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} = 0 \quad (9)$$

$A_1$  ratkaiseminen yhtälöstä (9) sijoittamalla  $A_2$ , joka ratkeaa yhtälöstä (8), antaa seuraavan tuloksen:

$$A_1 = \frac{e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}}{1 + e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}} \quad (10)$$

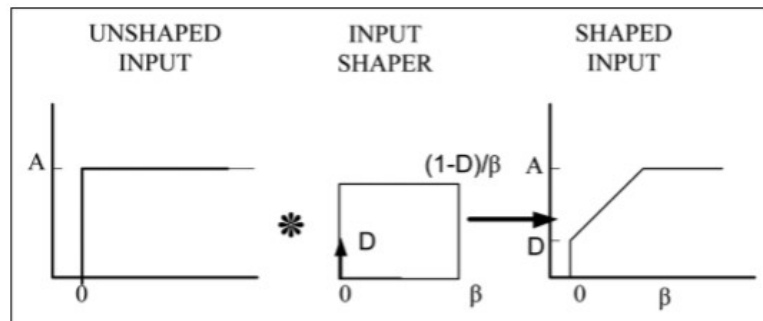
$A_2$  voidaan ratkaista samoilla yhtälöillä. Nämä laskut antavat seuraavat ratkaisut, jotka ovat nyt toimiva ZV-signaalinmuokkain.

$$[A_i \ t_i] = \begin{bmatrix} \frac{I}{I+K} & 0 \\ \frac{K}{I+K} & \frac{\pi}{\omega_d} \end{bmatrix} \quad (11)$$

Jossa:

$$K = e^{\frac{\zeta \pi}{\sqrt{1-\zeta^2}}} \quad (12)$$

ZV-muokkaimen yhtälössä (11) impulssi on jaettu kahteen osaan. Toinen vaihtoehto on tietysti jakaa signaali osittain nousevaksi käyräksi. Tällaista signaalinmuokkainta kutsutaan Distributed Zero Vibration (DZV) –muokkaimeksi, jonka perusidea on täysin sama, mutta toinen amplitudi annetaan nousevana käyränä yksittäisen impulssin sijaan. Kuvasta 4 nähdään, kuinka vasemmalla oleva askelsignaali muokataan DZV-menetelmällä. Järjestelmän värähtelyä saadaan vähennettyä huomattavasti, sekä saadaan nousukäyrästä hieman tasaisempi ja hallitummin nouseva.



**Kuva 4.** DZV-menetelmän perusidea graafisesti esitettyinä. (Maghsoudi et al. 2017.)

Viivettä voi molemmissa muokkaimissa kuvata seuraavalla yhtälöllä (Maghsoudi et al. 2017.):

$$y(t) = \int_0^{\beta} x(t - \varepsilon) dw(\varepsilon) \quad (13)$$

jossa  $x$  ja  $y$  ovat vastaavasti tulo ja lähtö. Viiveen jakautumista ajassa voidaan kuvata suurella  $w(\varepsilon)$  loppuajasta  $\beta$  riippuen. (Maghsoudi et al. 2017.) Täten, voidaan kirjoittaa:

$$w(\varepsilon) = \begin{cases} 0 & \varepsilon < 0 \\ \frac{1}{\beta}\varepsilon & \varepsilon \in [0, \beta] \\ 1 & \varepsilon > \beta \end{cases} \quad (14)$$

Koska  $dw(\varepsilon) = \frac{1}{\beta}d\varepsilon$  kun,  $\varepsilon \in [0, \beta]$ , voimme kirjoittaa yhtälön (13) seuraavasti (Maghsoudi et al. 2017.):

$$y(t) = \frac{1}{\beta} \cdot \int_0^{\beta} x(t-\varepsilon) d\varepsilon \quad (15)$$

Siirtofunktio DZV-muokkaimelle voidaan kirjoittaa Laplace-muunnoksella seuraavasti (Maghsoudi et al. 2017.):

$$D(s, \beta) = \frac{1 - e^{-s\beta}}{s \cdot \beta} \quad (16)$$

### 2.1.2 Zero Vibration and Derivative and Derivative

Ero ZV and Zero Vibration and Derivative (ZVD) –muokkaimen välillä on yksi lisätty rajoiteyhtälö. Tämä hidastaa vasteaikaa yhteen kokonaiseen jaksonaikaan, mutta kuten kuvasta 3 selviää, siihen vaikuttaa virhe paljon vähemmän ominaistajuuden tai vaimennussuhteen mallinnuksessa. Rajoiteyhtälöitä voidaan lisätä useita kertoja, muodostaen ZVD, ZVDD, ZVDDD ja niin edelleen signaalinmuokkaimia. Jokainen derivointikerros hidastaa muokkaustapahtumaa puolella jaksonajalla ja lisää virheensietokykyä yhdellä lisäimpulssilla. Näin ollen sopiva kompromissi vasteajan ja virheensietokyvyn välillä on tehtävä. (Poty et al. 2003; Singhose et al. 1997)

$$\sum_{i=1}^3 A_i \cdot t_i \cdot e_i^{\zeta \omega_n t_i} \cdot \cos(\omega_d \cdot t_i) = 0 \quad (17)$$

$$\sum_{i=1}^3 A_i \cdot t_i \cdot e_i^{\zeta \omega_n t_i} \cdot \sin(\omega_d \cdot t_i) = 0 \quad (18)$$

Yllä olevat yhtälöt (17) ja (18) tunnetaan derivaattarajoiteyhtälöinä (engl. derivative constraints). Paras ratkaisu näiden yhtälöiden täyttämiseksi on ZVD-muokkain. Sen laskeminen ja derivoiminen on pitkä prosessi, joten ne jätetään kirjoittamatta tässä vaiheessa ja pelkät vertailun kannalta tarpeelliset ratkaisut näytetään. ZVD-muokkain saa ratkaistuna seuraavat arvot:

$$[A_i \quad t_i] = \begin{bmatrix} \frac{1}{K^2+2\cdot K+1} & 0 \\ \frac{2\cdot K}{K^2+2\cdot K+1} & \frac{\pi}{\omega_d} \\ \frac{K^2}{K^2+2\cdot K+1} & \frac{2\cdot\pi}{\omega_d} \end{bmatrix} \quad (19)$$

Lisäämällä toinen derivaattarajoiteyhtälö, saamme arvot ZVDD-muokkaimelle:

$$[A_i \quad t_i] = \begin{bmatrix} \frac{1}{K^3+3\cdot K^2+3\cdot K+1} & 0 \\ \frac{3K}{K^3+3\cdot K^2+3\cdot K+1} & \frac{\pi}{\omega_d} \\ \frac{3\cdot K^2}{K^3+3\cdot K^2+3\cdot K+1} & \frac{2\cdot\pi}{\omega_d} \\ \frac{K^3}{K^3+3\cdot K^2+3\cdot K+1} & \frac{3\cdot\pi}{\omega_d} \end{bmatrix} \quad (20)$$

Tässä tapauksessa ZVD ja ZVDD sisällytetään vertailuun, mutta yhtään pidemmälle derivoidut signaalinmuokkaimet olisivat liian hitaita vasteajaltaan tähän käyttötarkoitukseen. Jaksonaika on melko pitkä käyttötarkoituksessamme, sillä ominaistajuudet ovat niin matalia. Täten, jopa kaksi kertaa derivoitu ZVDD-muokkain saattaa olla jo liian hidas vasteajaltaan.

### 2.1.3 Extra Insensitive

Extra insensitive (EI) –muokkain antaa mallintajan valita värähtelytoleranssin, joka sallitaan jäännösvärähtelyksi. Suurempi toleranssi mahdollistaa teoriassa nopeamman vasteajan. EI-muokkaimet ovat ZV(D<sup>x</sup>) –muokkaimien kanssa samanlaisia siinä mielessä, että ne hidastuvat puolella jaksonajalla per lisätty rajoiteyhtälö, mutta toisaalta jäännösvärähtelyn määrä vähenee jokaisella rajoiteyhtälöllä. Kuten sanottu, EI-muokkain perustuu nolla-värähtelyvaatimuksen osittaiseen vapauttamiseen. Tämä onnistuu toleranssin ( $V_{tol}$ ) kasvattamisella jäännösvärähtelyn yhtälölle ( $V$ ), joka aiemmin oli 0.

$$V(\zeta, \omega) = e^{-\zeta\omega t} \sqrt{C(\zeta, \omega)^2 + S(\zeta, \omega)^2} = V_{tol} \quad (21)$$

Mikäli vaimennussuhde ( $\zeta$ ) on nolla, EI-muokkaimelle on mahdollista laskea analyttinen ratkaisu. (Singhose et al. 1997; Kang 2014.) Ratkaisu nähtävillä alla olevassa yhtälössä:

$$[A_i \quad t_i] = \begin{bmatrix} \frac{(1+V_{tol})}{4} & 0 \\ \frac{(1-V_{tol})}{2} & \frac{\pi}{\omega_d} \\ \frac{(1+V_{tol})}{4} & \frac{2\cdot\pi}{\omega_d} \end{bmatrix} \quad (22)$$

Kuitenkin, mikäli  $\zeta \neq 0$ , ratkaisu joudutaan numeerisesti approksimoimaan oletetuilla arvoilla  $V_{tol}$  ja  $\zeta$ . Tällaista ratkaisua kutsutaan osittain analyttiseksi ratkaisuksi (engl. partially analytical). Laskelman yksityiskohtainen esitys on laaja ja löytyy kokonaisuudessaan esitettynä Kangin tekstistä. Täten, yhtälöissä (23)-(26) on esitetty vain ratkaisut osittain analyttiselle signaalinmuokkaimelle. (Kang 2014.)

$$A_1 = \frac{K^2 \cdot \Phi^2 \cdot (1 + K \cdot V_{tol})}{(1 + K \cdot \Phi^2) \cdot (1 + K)} \quad (23)$$

$$A_2 = \frac{K^2 \cdot (1 - V_{tol}) + K \cdot (1 - K^2 \cdot V_{tol}) \cdot \Phi^2}{(1 + K \cdot \Phi^2) \cdot (1 + K)} \quad (24)$$

$$A_3 = \frac{I + K \cdot V_{tol}}{(I + K \cdot \Phi^2) \cdot (I + K)} \quad (25)$$

$$[A_i \quad t_i] = \begin{bmatrix} A_1 & 0 \\ A_2 & \frac{\pi}{\omega} \\ A_3 & \frac{2 \cdot \pi}{\omega} \end{bmatrix} \quad (26)$$

Joissa:

$$\Phi = e^{\frac{\zeta}{\sqrt{1 - \zeta^2}} \phi} \quad (27)$$

#### 2.1.4 Unity Magnitude Zero Vibration

Unity Magnitude Zero Vibration (UMZV) on edellisistä hieman poikkeava tapa signaalin muokkaamiseen. Sillä on potentiaalia olla nopeampi kuin mikään muu tässä työssä esitelty muokkain, mutta sillä on omat kääntöpuolensa. UMZV-muokkaimen periaatteena on antaa myös negatiivinen impulssi järjestelmään. Tämä johtaa siihen että, voidaan käyttää suurempia amplitudeja, jotka kuitenkin aiheuttavat hieman erilaisia ongelmia. Yksi kääntöpuoli muun muassa on se, että negatiiviset impulssit herättävät korkeamman moodin värähtelyjä. (Shome et al. 2013)

Tämän tyyppiset signaalinmuokkaimet eroavat muista menetelmistä myös siltä kantilta, että ne mahdollistavat lopullisen arvon ylittämisen hetkellisesti käytännön sovellutuksissa. Vain positiivisia impulsseja käyttävät signaalinmuokkaimet eivät voi käyttää suurempaa impulssia kuin alkuperäinen signaali. Negatiivisen pulssin käyttäminen mahdollistaa näin systeemin vasteen kompensoimisen halutulle tasolle, vaikka ensimmäinen käsky olisi alkuperäistä signaalia suurempi. (Shome et al. 2013) UMZV-muokkain voidaan esittää seuraavalla yhtälöllä:

$$[A_i \quad t_i] = \begin{bmatrix} 1 & 0 \\ -1 & T/6 \\ 1 & T/3 \end{bmatrix} \quad (28)$$

Ensimmäisen impulssin on tietenkin oltava oikeaan suuntaan, jotta järjestelmä lähtee kiihtymään haluttuun suuntaan. Kun negatiivinen impulssi annetaan järjestelmään, tulee kolmas impulssi pakolliseksi, jotta järjestelmän vaste pysyy suunnaltaan oikeana. Suunniteltuun käyttötarkoitukseen tämän metodin käyttäminen voi olla melko haastavaa, sillä hydrauliiikkaa ohjaavan järjestelmän tarkkuus ja sykli aika näin nopeita ja tarkkuutta vaativia impulsseja käyttäessä voi olla liian matalalla tasolla.

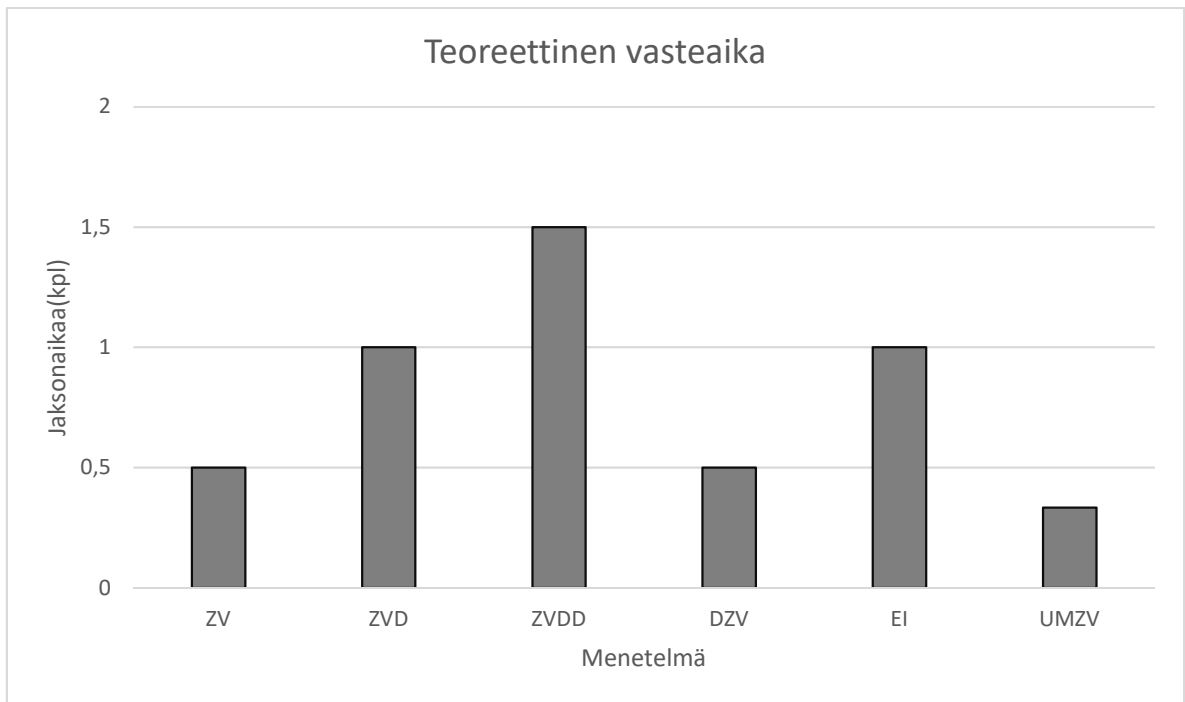
## 2.2 Vertailu

Tässä kappaleessa selitetään erilaisia ominaisuuksia ja vertaillaan niitä muokkainkohtaisesti. Koska osa on huomattavastikin erilaisin periaattein toteutettu, on niiden keskinäinen vertailu tarpeen. Se auttaa valinnan tekemisessä, sekä toimii ohjeena tulevaisuutta varten, jotta vastaavaa tutkimusta ei tarvitse lähteä toistamaan. Pisteytys tapahtuu antamalla kaikille signaalinmuokkaimille pisteet yhdestä kuuteen, kuutosen ollessa paras ja ykkösen huonoin. Nämä arvot taulukoidaan ja näin nähdään kategoria kerrallaan, mikä menetelmä on paras missäkin osa-alueessa. Tämä mahdollistaa sopivan muokkaimen valitsemisen, kun tiedetään mikä on tärkeää tietyn järjestelmän kohdalla.

### 2.2.1 Vasteaika

Vasteaika on todella tärkeä tässä käyttötarkoituksessa, sillä ensisijaisena tavoitteena ei ole eliminoida huojuntaa, vaan vähentää ja hallita sitä lastin käsittelyajan nopeuttamiseksi. Näin ollen, signaalinmuokkaimen vasteajalla on suuri vaikutus. Käytännössä, signaalinmuokkain jättää sitä enemmän jäännösvärähtelyä, mitä nopeampi sen vasteaika on. Eli yleensä on valittava vasteajan tai tehokkaan vaimennuksen välillä.





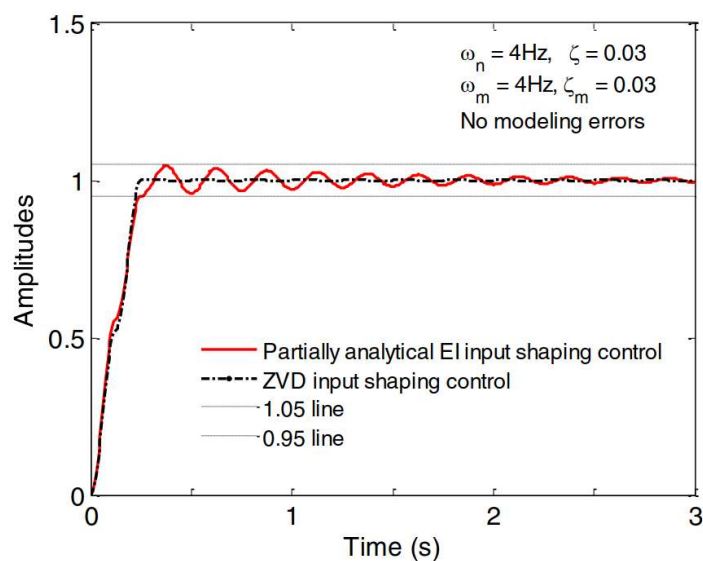
**Kuva 5.** Eri menetelmien viiveet.

Yllä oleva kuva 5 näyttää eri muokkausmenetelmien vasteaikaerot teoreettisella tasolla. Kuten kuvasta näkee, toimivat nopeat ZV ja DZV –menetelmät puolen jaksonajan viiveellä. Ne toimivat hyvin virheettömässä mallissa, joka vastaa täysin fyysistä systeemiä. Paremmiin mallinnusvirhettä kestävä ZVD ja EI –menetelmät toimivat kahdella impulssilla, eli kokonaisen jaksonajan viiveellä. Tämä on jo melko pitkä aika, ominaistajuuksien ollessa niin pienet tässä käyttötarkoituksessa. Simulaatioista näkee parhaiten, kuinka systeemit käyttäytyvät näillä menetelmillä muokattuina, joten lopullinen päätös tehdään vasta simulointivaiheessa. UMZV-muokkain on vahvoilla tässä kategoriassa, sillä sen 1/3 jaksonajan vaste on kaikista nopein.

Pienimpään jäännösvärähtelyyn päästään ZVDD-menetelmällä. Näin pienellä ominaistajuuksella varustetulla systeemillä ei ole juurikaan ylitystä tai jäännösvärähtelyä. Kuten jo aiemmin mainittu, tällä on vaikutuksensa vasteaikaan, joka on 1.5 jaksonaika. Se on pitkä aika, mutta käyttökelpoinen ainakin vertailun merkeissä. Simulaatioista näkee tarkemmin oikeat vasteajan, mutta mitä luultavimmin ne ovat suhteessa samat.

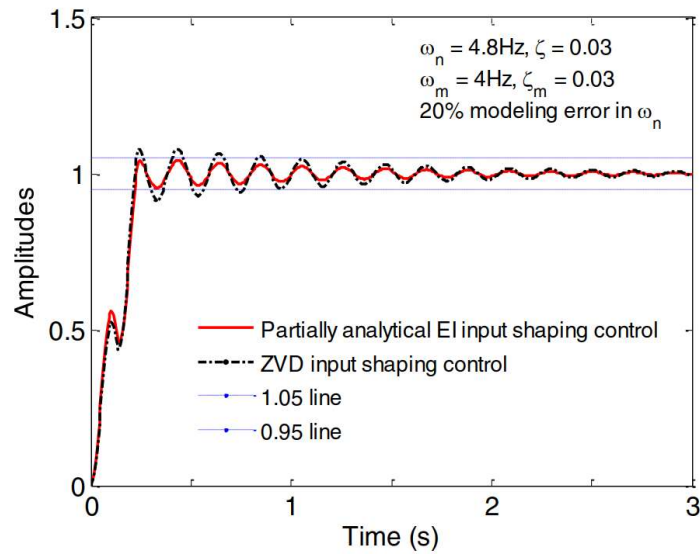
### 2.2.2 Virheensietokyky

Jäännösvärähtely ( $V$ ) on nolla ZV( $D^x$ )-menetelmillä, jos vaimennussuhteessa tai ominaistaajuudessa ei ole virhettä. (Kang 2014.). Koska EI-muokkaimet toimivat sallimalla vaimennussuhteeseen virhettä, voittaa ZVD-menetelmä ideaalitapauksessa, kuten kuvasta 6 nähdään. Tämä oli odotettavissa, sillä kun  $V = 0$ , ZVD-menetelmä toimii ideaalisesti. Tämä kuitenkin muuttuu kun järjestelmään mallinnetaan 20 % virhe ominaistaajuuteen, kuten kuvassa 7.



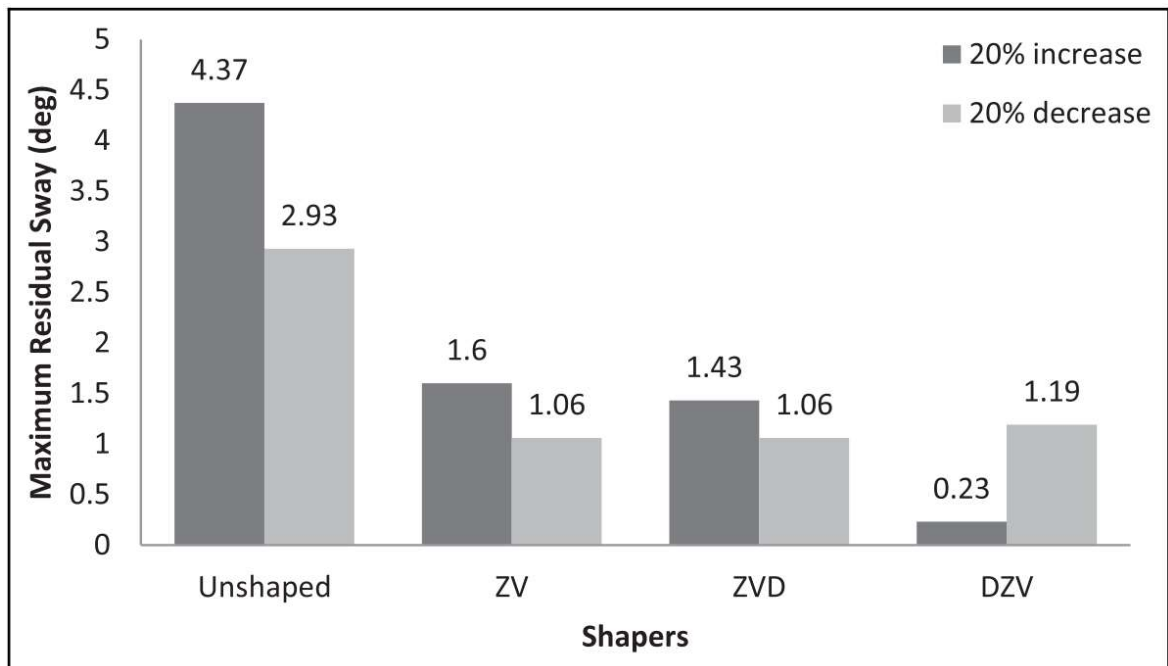
**Kuva 6.** ZVD vs. EI –menetelmät kun mallinnusvirhettä ei ole. (Kang 2014.)

EI-menetelmä on edelleen pyydettyssä alle 5 % värähtelyssä kuten sen kuuluukin olla. ZVD-menetelmä ei kuitenkaan toimi enää ideaalisesti ja sen näkee suuremmasta ylityksestä sekä suuremmasta jäännösvärähtelystä. Tämä tekee EI-muokkaimesta paremman vaihtoehdon käytännön ratkaisuihin, jotka eivät ole täysin lineaarisia tai jotka sisältävät vaikeasti mallinnettavaa käyttäytymistä. Virhettä syntyy helposti esimerkiksi jo kontrollerien syklijajasta.



**Kuva 7.** ZVD vs. EI –menetelmät kun mallinnusvirhettä ominaistajuudessa 20 %. (Kang 2014.)

ZV, ZVD and DZV –menetelmien virheensietovertailu antaa odottamattomia tuloksia. Maghsoudin tutkimusryhmä huomasi kokeissaan, että ZV and ZVD –muokkaimet reagoivat melko symmetrisesti sekä negatiiviseen, että positiiviseen virheeseen ominaistajuudessa. DZV-menetelmä on kuitenkin melko epäsymmetrinen reaktioltaan. Kun virhe on negatiivista, on reaktio suurempi kuin ZV ja ZVD –muokkaimilla. Kuitenkin jos mallinnusvirhe on +20 %, ei DZV reagoi juurikaan kuten kuva 8 osoittaa. (Maghsoudi et al. 2017.) Käytännössä signaalinmuokkainten arvot laskettiin ominaistajuudella X, jonka jälkeen kokeiltiin, kuinka paljon tulokset muuttuvat, kun ainoastaan järjestelmän ominaistajuutta muutetaan.



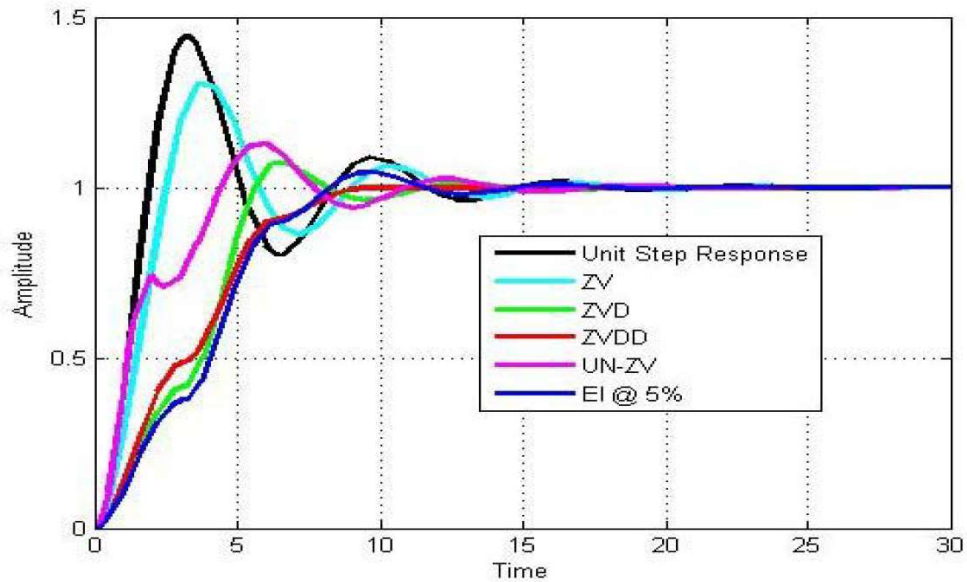
**Kuva 8.** Ominaistaajuuden muuttamisen vaikutus jäännösvärähtelyyn ZV, ZVD ja DZV -menetelmillä  $\pm 20\%$  muutoksella. (Maghsoudi et al. 2017.)

### 2.2.3 Jäännösvärähtely

Koska kaikki kriteerit ovat toisistaan riippuvaisia, ei jäännösvärähtelyn vertaileminen ole täysin suoraviivaista. Käyttötarkoitus johon tulosiinuaalinmuokkainta rakennetaan, on hyvin taulukoitu ominaistaajuuden arvoiltaan, mutta järjestelmä ei käyttäydy täysin lineaarisesti taajuuden noustessa. Täten yksinkertainen ZV-muokkain saattaisi olla riittävä, kun suurta virhettä simulaatiomallin ja käytännön välillä ei pääse syntymään. Tämä johtaa myös siihen, että muokkaimen virheensietokyvyllä ei tässä vertailussa olisi juurikaan väliä.

Valitettavan useassa tapauksessa on niin, että mitä pienempi jäännösvärähtely saavutetaan, sitä pidempi vasteaika kyseisellä menetelmällä on. Kuten kuvasta 9 havaitaan, ZVDD voittaa pienimmällä jäännösvärähtelyllään. Kuitenkin, sen vasteaika on todella hidas verrattuna muihin muokkaimiin. Seuraavaksi, vain vähän ZVD-muokkaimen edellä on EI-muokkain. Ideaalitulanteessa, jossa mallinnusvirhettä fyysiseen maailmaan verrattuna ei ole, olisi ZVD-muokkain EI:tä parempi. EI-muokkain taas kestää mallinnusvirhettä paremmin. Seuraavaksi sijoittuvat UMZV, DZV ja ZV edellä mainitussa järjestyksessä. DZV:tä ei ole kuvassa 9, mutta se on vain hieman ZV-muokkainta parempi pehmeämmän noususignaalin johdosta.

On kuitenkin otettava huomioon, että kuvan tulokset ovat hieman tätä työtä eroavasta järjestelmästä, sillä siinä on entistä pienempi ominaistaajuus sekä huomattava määrä vaimennusta.



**Kuva 9.** Jäännösvärähtelyn määrä ja eri menetelmien vasteet päällekkäin. (Shome et al. 2013)

#### 2.2.4 Vertailutulokset

Alla oleva taulukko 1 on pistejärjestelmä tämän vertailun tuloksille. Kuten tuloksista näkyy, on tarjolla selkeä paremmuusjärjestys jokaisen kriteerin perusteella. Näin voidaan suoraan vertailla menetelmien eri ominaisuuksia. Tosiasia on kuitenkin se, että kriteerit eivät ole tasa-arvoisia useimmissa käytännön tapauksissa. Erilaisiin käyttökohteisiin on selkeästi toisistaan poikkeavia tarpeita ja vaatimuksia, joten on mietittävä jokaisen projektin kannalta, mikä kriteeri on missäkin tapauksessa painoarvoltaan suurin.

*Taulukko 1. Algoritmien vertailutaulukko. Paras kussakin kategoriassa saa kuusi pistettä.*

	ZV	ZVD	ZVDD	DZV	EI	UMZV
Virheensieto	1	3	6	4	5	2
Vasteaika	6	3	1	5	2	4
Jäännösvärähtely	1	4	6	2	5	3

DZV ja ZV –muokkaimet ovat nopeimmat vertailuryhmästä. DZV:llä on selkeästi paremmat pisteet virheensietokategoriassa kuten kuva 8 osoittaa. DZV-menetelmän reaaliaikainen laskeminen kuitenkin pudottaa näkymättömästi menetelmän pisteitä, sillä matematiikka muokkaimen taustalla on melko monimutkaista ja approksimoinnin takia ei täysin tarkkaa. Lisäksi, ZV-muokkaimen saa käyttäytymään samantyyllisellä tavalla, kun sillä muokkaa ramppi muodossa olevaa signaalia. ZV-muokkain on erittäin yksinkertainen ja hyvä vaihtoehto, mikäli simulaatiot osoittavat sen olevan tarpeeksi tehokas jäännösvärähtelyn osalta.

Kuten taulukko 1 näyttää, ZVDD on paras kaikissa muissa kriteereissä, paitsi vasteajassa. Vasteaika kuitenkin on koko joukon hitain, kuten jo aiemmin todettu. Näin matalataajuisessa projektissa ZVDD-menetelmä on liian hidaskäyttöinen siitä saatavaan hyötyyn nähden.

EI-muokkain on jo selkeästi parempi käytettävyydeltään kuin ZVDD-muokkain, nopeammasta vasteajasta johtuen. Matalien taajuuksien aiheuttama kahden jaksonajan mittainen viive on myös EI-menetelmällä pitkä. EI-menetelmässä on myös DZV:n kaltainen ongelma, eli monimutkaiset laskelmat, joiden käytännön implementointi on vähintäänkin hankalaa ja teettää kohtuuttomasti töitä saatuihin tuloksiin nähden. Vaihtoehtona on myös olettaa vaimennus nolllaksi ja käyttää ideaalia EI-muokkainta, joka jo aiemmin kappaleessa 2.1.3 esiteltiin.

ZVD-muokkain on melko hyvä kaikessa, muttei erityisen hyvä missään. Simulaatioiden tuloksien perusteella tehdään lopullinen päätös tämän menetelmän suhteen, mutta teorian perusteella se on vertailun keskikastissa puolinäkymättömänä yksilönä. UMZV-menetelmä on lähes samassa tilanteessa. Sen suorituskyky on hyvällä tasolla kaikissa kategorioissa, mutta erityisen hyvällä tasolla vasteajassa. Negatiivisten impulssien tarve kuitenkin pakottaa

siirtämään tämän menetelmän tutkimista, sillä sen toimimaan saattaminen vaatisi tutkimusta ja mahdollisesti kehitystä myös hydraulisen sekä ohjausjärjestelmän saralla. Tämä menetelmä on siis ehdottomasti jatkotutkimuksen arvoinen, mutta tämän työn rajauksiin sen jatkojalostaminen ei mahdu.

Paras ratkaisu käytännön toteutukseen saattaisi olla ZV, ZVD tai jopa DZV-muokkain, jos mallinnus- ja laskelmointihaasteet saadaan ratkaistua. Lopullista valintaa on vaikea tehdä pelkän teorian perusteella, mutta muutamia menetelmiä voidaan jo pois sulkea. UMZV-menetelmä on mielenkiintoinen ja jatkotutkimuksen arvoinen, mutta käytännön rajoitteet siirtävät sen tutkimisen toiseen tutkimukseen. ZVDD-muokkain on niin hidaskäyttöinen verrattuna nopeampiin algoritmeihin, että sen antamista hyödyistä ei ole suurta apua, kun ne saavutetaan niin hitaasti. EI-muokkaimen kohdalla valinnan tekeminen on hieman hankalaa, sillä se ei ole paljoa parempi kuin ZVD-muokkain, mutta on huomattavasti vaikeampi mallintaa ja rakentaa. Kuten sanottu, on myös mahdollista kokeilla vaimentamattoman järjestelmän EI-muokkainta, joka ei ota huomioon vaimennussuhdetta täysin, mutta on selkeästi yksinkertaisempi mallintaa. Tämä menetelmä kuvattuna yhtälössä (22).

### 2.3 Projektin tarvekartoitus

Tässä kappaleessa projektille tehdään tarvekartoitusta kahdesta eri näkökulmasta – AGV ja manuaalikäyttöön. Molemmassa käyttötarkoituksissa on omat vaatimuksensa, jotka eroavat niin paljon, että ne kannattaa käsitellä erikseen. Tähän mennessä on jo todettu, että tulosignaalin muokkaus on erittäin kustannustehokas tapa värähtelyn ja sitä aiheuttavien signaalien hallitsemiseen.

#### 2.3.1 AGV

Automaattitruckien maailmassa kuljetuskapasiteetti on kaikki kaikessa. Asiakas ei tahdo yhtään truckia, vaan mieluiten ratkaisun, jossa hyödykkeet siirretään paikasta A paikkaan B ilman minkäänlaista kalustoa. Nykypäivän teknisistä rajoitteista johtuen, asiakkaan on vielä ostettava ratkaisuja, jotka sisältävät trukkeja. Edellä mainittu periaate kuitenkin toimii, sillä useimmiten vähiten trukkeja sisältävä tarjous voittaa, mikäli kuljetuskapasiteetti on sama. Näin on vähemmän huoltoa vaativia koneita, eikä niin montaa hajoamiskohdetta. Jo nämä faktat tukevat tätä projektia, sillä jos suhteellisen pienellä rahallisella panostuksella saadaan

nopeutettua lastin käsittelyaikaa, voi se olla juuri se kapasiteetin lisäys, jolla seuraava tarjouskilpailu voitetaan

Rocla AGV-koneissa on lavan etsintäjärjestelmä (engl. Pallet finder), joka skannaa hyllyssä olevia lavoja ja lavataskuja virheiden estämiseksi. Järjestelmä perustuu 3D-skannaukseen, joten on sanomattakin selvää, etteivät mallit ole tarkkoja, mikäli nostokelkkaan kiinnitetty skanneri huojuu huomattavasti. Esimerkiksi, jos painava lava pitää nostaa 10-metrin korkeuteen. Kone osaa ajaa suurin piirtein oikeaan kohtaan ja nostaa maston lähelle oikeaa korkeutta. Tässä vaiheessa tapahtuu 3D-skannaus, kun maston huojuminen on vaimentunut tarpeeksi ajoliikkeen pysähtymisen ja maston noston jäljiltä. 10-metrin korkeudessa huojuminen voi olla satoja millimetrejä ja suuren massan takia, värähtely taajuus on niin pieni, että odottamalla värähtelyn vaimenemista voi joutua olemaan paikallaan jopa kymmeniä sekunteja. Jos tätä tapahtumaa voidaan lyhentää esimerkiksi viidellä sekunnilla, on sen vaikutukset koko AGV-ratkaisun tasolla huomattavat. Hypoteettisessa fleetissä on nyt 15 vaunua. Yksi vaunu voi tehdä keskimäärin noin 10 lastinkäsittelysykliä tunnissa. Näillä arvoilla muutamat sekunnit kertautuvat jo minuuteiksi koko fleetin tasolla. Tämä laskutoimitus oli oikeaan dataan perustuva, mutta silti erittäin hypoteettinen. Se ei kuitenkaan ole epäselvää, etteikö syklin nopeutuksella olisi parantava vaikutus kuormankäsittelykapasiteettiin.

Tarkempi analyysi ja laskelmat ovat esitetty kappaleessa 3. Tarve huojunnan vaimennukselle on kuitenkin selkeä. Tehokkuuden lisäksi myös turvallisuus nousee, mikä on tietenkin ainoastaan positiivinen asia. Tämän kaltaisilla algoritmeilla on siis suuri potentiaali ja jatkotutkimus on erittäin aiheellista.

### 2.3.2 Manuaalikäyttö

Manuaalikäyttöön tilanne ei ole niin selkeä, kuin AGV-käytössä. Manuaalikäyttö ei ole ollenkaan luotettavaa, johdonmukaista tai ennustettavaa, koska trukin logiikka ei pysty lukemaan ihmisen ajatuksia. Koneesta voidaan siis tehdä mahdollisimman nopea, tehokas ja helppokäyttöinen, jotta kuljettajalla on parhaat mahdollisuudet onnistua mahdollisimman hyvin tuloksin.



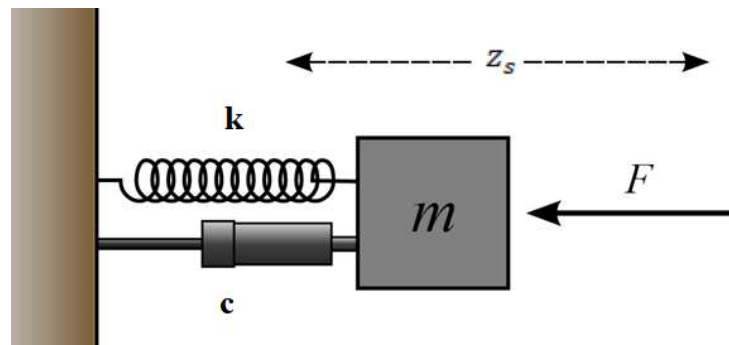
Todella suuri osuus manuaalikäytön signaalin muokkaamisesta liittyy käyttäjätuntuman mukavaksi säätämisestä. Tämä pakottaa valittavan algoritmin vasteajan olemaan mahdollisimman nopea, ettei käyttäjä tunne koneen elävän omaa elämäänsä. Jos käyttäjä pysäyttää työntöliikkeen ja kone pysähtyy 0.5-2 sekunnin viiveellä riippuen nostokorkeudesta ja massasta, voi se olla todella epämiellyttävää käyttäjän mielestä. Tätä asiaa ei kuitenkaan aleta valtavan työmäärän takia tässä työssä käytännössä testaamaan.

Manuaalikäyttöön suurimmat hyödyt saadaan työntönopeuden ja -kiihtyvyyden nostamisella, joka voi onnistua signaalin muokkauksen jälkeen. Tietenkin myös huojumisen väheneminen nopeuttaa työskentelyä. Yhtä suuriin säästöihin tuskin päästään, kuin AGV-puolella, sillä kone ei tiedä liikkeen alkaessa mihin liike tulee pysäyttää. Tämä johtaa siihen, että työntöradan loppuun asti ei voida ajaa täyttä vauhtia, sillä on pidettävä huoli siitä, että liike ehditään vielä pysäyttämään signaalinmuokkaimen aiheuttaman viiveen aikana, ilman että masto törmää ratansa päätyyn. Tämän ongelman ratkominen tapahtuu seuraavassa kappaleessa, mutta selvää on, että liikkeiden optimoinnilla signaalinmuokkaimelle, on saavutettavissa ajansäästöä.

#### 2.4 Dynaamisen järjestelmän simulointi

SIMULINK mallintaminen mahdollisti eri algoritmien erilaisten ominaisuuksien huomaamisen, tutkimisen ja vertailemisen. Kuten simulointi aina, vaati tämänkin tapauksen mallintaminen ja simuloiminen yksinkertaistuksia ja oletuksia. Onneksi tässä käyttötapauksessa käytännön toteutus on todella suoraviivainen yksinkertaistaa, eikä yksinkertaistaminen vääristä tuloksia merkittävästi.

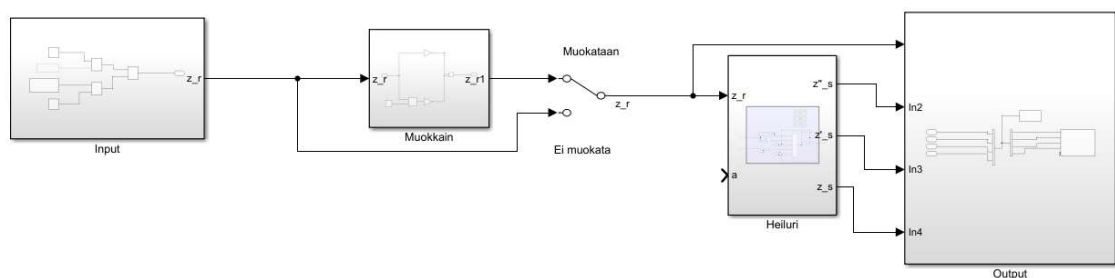
Kuvassa 10 kuvattuna millaiseksi masto on yksinkertaistettu ensimmäisessä SIMULINK mallissa. Heiluria poikkeutetaan paikkasignaalilla, joka aiheuttaa kiihtyvyyden, jonka maston liikkeelle laittaminen tai pysäyttäminen aiheuttaisi. Näin saadaan selville massan huojumiskäyttäytyminen, sekä vaimenemisen nopeus.



**Kuva 10.** SIMULINK-simulaatioissa käytettävä yksinkertaistettu malli maston värähtelyliikkeestä. (Karonen. 2005)

#### 2.4.1 Dynaamisen heilurijärjestelmän mallinnus

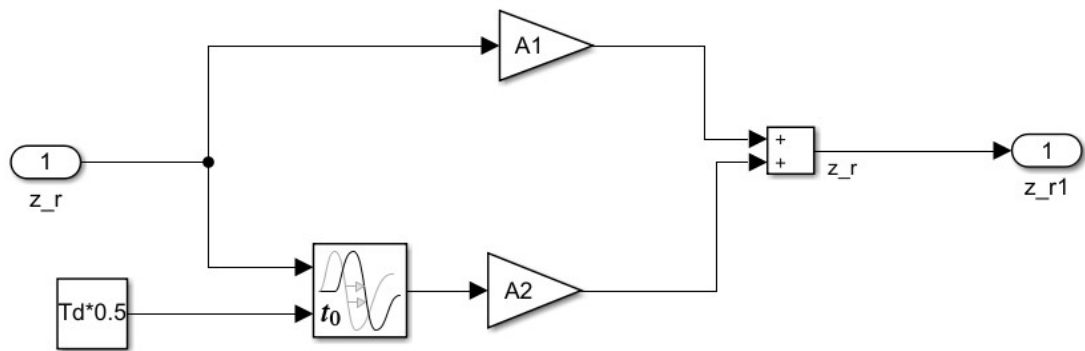
Tässä osuudessa esitellään mallinnuksen tuotos ja käydään hieman läpi, miten tähän on päädytty. Kuten jo kuvasta 10 nähdään, lähdettiin tutkimusongelmaa lähestymään yksinkertaisen jousimassasysteemin kautta, sillä se kuvaa todella hyvin tätä käyttötarkoitusta. Käytännössähän masto on jousi, jolla on vaimennusominaisuuksia, sekä päässä on vielä massa.



**Kuva 11.** SIMULINK-malli, jolla testattiin erilaisia muokkausalgoritmeja, kokonaisuudessaan.

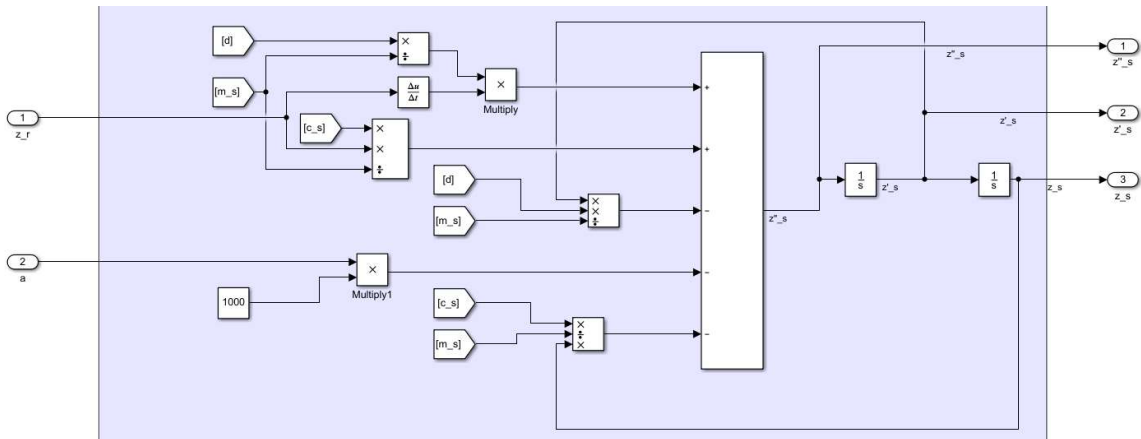
SIMULINK-mallissa heiluriin voidaan syöttää joko muokattu tai muokkaamaton signaali. Näin saadaan helposti tutkittua järjestelmän vastetta eri tapauksissa. Kuvan 11 mukaisesti Input-alijärjestelmästä valittava signaali syötetään, joko muokattuna tai muokkaamattomana heiluriin, josta sitten otetaan tulokset Output-lohkolla jatkokäsittelyyn. Signaalinmuokkain –alijärjestelmä on totta kai erilainen jokaiselle erilaiselle menetelmälle. Kuvassa 12 on ZV-

muokkaimen yhtälö kahdella impulssillaan ( $A1$  ja  $A2$ ). ZVD-menetelmän mallissa olisi vastaavasti kolme impulssia, joista kahdessa jälkimmäisessä on viivettä.



**Kuva 12.** ZV-muokkain. Tulosignaali siis jaetaan kahteen impulssiin, jossa toinen annetaan viiveellä.

Itse heiluri on matemaattisesti ratkaistu yhtälö, joka on ratkaisun jälkeen mallinnettu Simulinkiin simulaatioiden mahdollistamiseksi. Simulink-malli esitetty kuvassa 13. Kiihtyvyydestä saa loogisesti kerran tai kaksi integroimalla, joko nopeuden tai paikkatiedon. Massa, vaimennussuhde, jousivakio ynnä muut tarpeelliset tiedot saadaan MATLAB:n puolelta .m-tiedostosta, johon on taulukoitu ominaistajuudet, alkuarvot ja laskettavien arvojen yhtälöt.



**Kuva 13.** Heilurin malli Simulinkissä.

Kuvassa 13 mallinnettuna yhtälö johon Simulink-malli pohjautuu. Alla esitetty yhtälö on heilurin perusmuotoinen yhtälö, jossa voidaan poikkeuttaa heiluria joko massaan kohdistuvalla voimalla tai paikan muutoksella  $z_r$ .

$$m_s \cdot z_s \frac{d^2 z_s}{dt^2} + c \cdot z_s \frac{dz_s}{dt} + k \cdot z_s = d \cdot z_r \frac{dz_s}{dt} + k \cdot z_r - F(t) \quad (29)$$

Mallista otetaan ulos paikka, nopeus ja kiihtyvyys, joilla pystytään analysoimaan värähtelyä. Myös  $z_r$  tai voima, jolla heiluria poikkeutetaan, otetaan Output-laatikkoon vertailua varten.

Yhtälö (29) ratkaistaan  $z_s \frac{d^2 z_s}{dt^2}$  suhteen, jotta yhtälö voidaan mallintaa kuvan 13 tavalla:

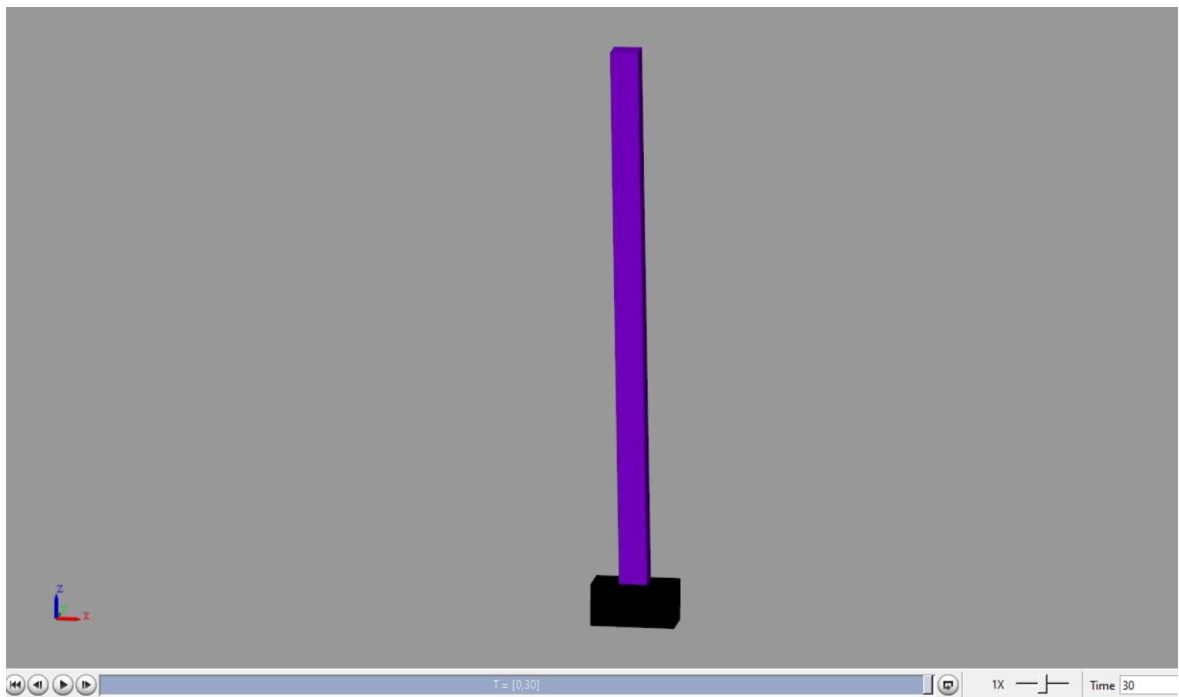
$$z_s \frac{d^2 z_s}{dt^2} = \frac{1}{m_s} (c \cdot z_r \frac{dz_s}{dt} + k \cdot z_r - c \cdot z_s \frac{dz_s}{dt} - k \cdot z_s - F(t)) \quad (30)$$

Tästä yhtälöstä muodostetulla, aiemmin kuvatulla Simulink-mallilla on simuloitu kappaleen 3 vastekuvaajat keskinäistä vertailua varten.

#### 2.4.2 Ohjausjärjestelmä ja visualisointi

Jotta voi rakentaa ohjausjärjestelmän, on ensin rakennettava toimiva järjestelmä Simulinkiin. Sen todettiin olevan mielekkäin Simscapella, sillä sen kautta saadaan helposti

animaatio visualisoimaan värähtelyä. Näin havainnollistaminen on selkeää ja yksinkertaista, sekä asiaan perehtymätönkin kykenee tapauksen hahmottamaan. Simscape on työkalulaatikko Simulinkiin, joka mahdollistaa fyysisten järjestelmien mallintamisen ja testaamisen helposti. Tässä tapauksessa fyysinen toiminnallisuus rakennetaan Simscapeella ja sitä ohjataan erilaisilla inpuuteilla. Simulaatio saadaan visualisoitua 3D-mallein, jotka generoituvat automaattisesti, kun niille antaa sopivat alkuarvot. Tämä on hyvin paljon vähätöisempää, kuin kaiken mallintaminen itse. Näin työaikaa jää oikeaan tutkimukseen, eikä sitä kulu mekaaniseen mallinnustyöhön.



**Kuva 14.** Matlabin Mechanics Explorerin näkymä Simscape Multibody -mallista.

Malli koostuu yksinkertaisesti kelkasta, jossa sitten on korkea jäykkä palkki pystyssä. Tässä kohdassa on pieni yksinkertaistus, sillä oikea masto on joustava, mutta koska taajuudet on taulukoitu ja liike on suhteessa niin pientä, ei tällä ole merkittävää vaikutusta tuloksiin. Kelkkaa liikutetaan maksimissaan 80 mm/s nopeudella, sekä maksimissaan 200 mm/s<sup>2</sup> kiihtyvyydellä. Malli on pitkälle kehitetty alkuperäisestä, mutta pohjautuu osittain Mathworksin esimerkkimalliin (Mathworks. 2018).

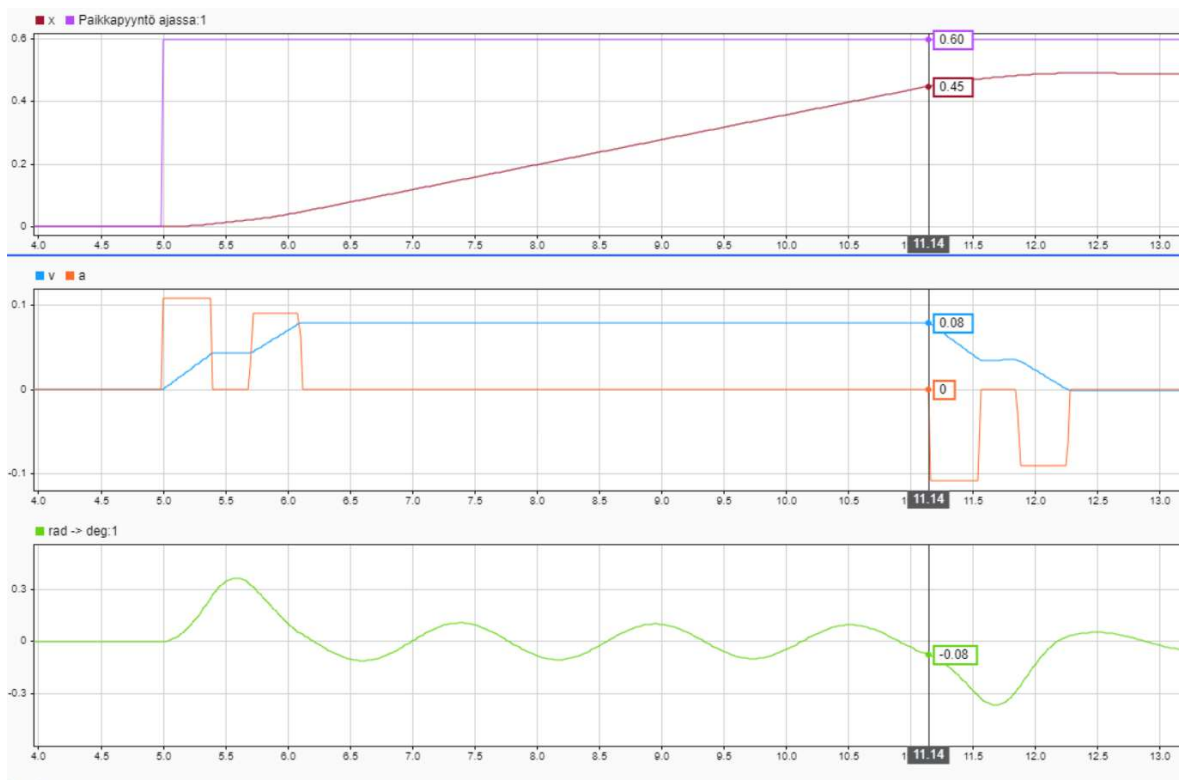


takaisinkytkennästä kuitenkin olisi hyötyä, sillä ympäristön muuttajat voivat vaikuttaa siihen, liikkuuko masto kuten oletetaan. Kaikissa koneissa ei sopivaa takaisinkytkentää ole, tai se on yksinkertainen induktiivinen anturi liikkeen päässä. Näin ollen paikoituslogiikka pitää räätälöidä jokaiseen käyttötarkoitukseen sopivaksi, kun tiedetään millainen anturointi missäkin koneessa on. Tämän työn simulaatioissa käytetään aluksi takaisinkytkentänä tarkkaa tietoa vaunun paikasta. Samainen algoritmi on kuitenkin mahdollista yksinkertaistaa toimimaan pelkillä rajakytkimillä, mikäli sille on tarvetta.

## 2.5 Pysähtymismatkaratkaisu

Erityisesti manuaalikäytössä on otettava huomioon, kuinka mahdollinen törmäystilanne estetään. Päin työntöradan mekaanista rajaa ajaminen aiheuttaa suuren heilahduksen ja voi taten olla vaarallinen. Automaattikäytössä tämä ongelma on helppo kiertää, sillä koodissa määritellään joka hetki, missä ja mihin työntöliike on menossa. Näin voidaan määritellä, ettei maston paikkatieto saa koskaan arvoja jotka ovat yli liikeradan pituuden.

Manuaalikäytössä on kuitenkin otettava huomioon, että kuski voi ja mitä luultavimmin ajaa päin rajaa. Eli on varmistettava, että pysähtyminen tapahtuu hallitusti ja että signaalin muokkaus tulee tehtyä ennen törmäystä. Käytännössä on monitoroitava ja laskettava aikaa sekä etäisyyttä, joka törmäykseen on nykyisellä nopeudella. Koska määrittelemme maksimikiihtyvyyden, tai tässä tapauksessa hidastuvuuden, sekä signaalin muokkauksen aiheuttaman viiveen, tiedämme mikä on pysähtymisaika ja –matka. Yksinkertaisimmillaan, voidaan määritellä, että kun liike etenee tähän kriittiseen rajaan asti, pysähdytään signaalin muokkauksen mukaisella kahdella impulssilla ja irrotetaan täysin kuljettajan antama liikkumisohje, kunnes liike on pysähtynyt. Näin saadaan inhimillisyyks poistettua pysähtymistapahtumasta, joka on aina yksinkertaistava asia. Kuvan 16 simulaatio on tehty juuri tällä periaatteella.



**Kuva 16.** Simulaatio pysäytysalgoritmista, kun ollaan ajamassa päin rajaa. Kuski yrittää ajaa pisteeseen 0.6 m ja rajana on 0.5 m. Koodi ottaa hallinnan kuljettajalta datakursorin kohdalla ja pysäyttää liikkeen hallitusti kuskista riippumatta.

Kuvassa 16 on simuloitu tilannetta, jossa kuljettaja on ajamassa päin päätyrajaa. Liikkeellelähtö tapahtuu normaalisti ja kuljettaja ohjaa työntöliikettä. Kun lähestytään päätyrajaa, laskee kone koko ajan, mikä on pysähtymisaika kuluvalle ajanhetkellä. Kun se menee alle kriittisen ajan, pysäytetään liike kuskin tekemisistä riippumatta. Näin kuski on 1.2 sekuntia ilman työntöliikkeen hallintaa, kuvan mukaisessa tilanteessa. Kuvan tilanteessa on 700 kg massa yhdeksän metrin korkeudessa. Korkeammalla tai suuremmalla massalla kyseinen aika on tietysti pidempi ja matalammalla tai kevyemmällä massalla lyhyempi.

Käytännöntoteutus tästä on vain IF-lauseita toistensa sisällä. Kuvassa 17 on osa tästä koodista. Törmäysturva aktivoituu kuvan tapauksessa, kun ollaan 50 mm päässä päädyistä ja kelkka on liikkeessä positiiviseen suuntaan. Mikäli kelkka on paikassa joka täsmää paikkapyyntöön, liikekäsky asetetaan nollaan. Mikäli liikettä ei ole ja liikekäskykin on nolla, asetetaan käsky odottamaan signaalia käyttäjältä. Mikäli pysähtymisaika menee alle



kriittisen ajan, annetaan ennalta laskettu pysähtymiskäske, joka on kulkenut signaalin muokkaimen läpi. Näin tässä tapauksessa -12 000 N voimasta vain  $A1$ :n verran on ensimmäinen pulssi, jonka jälkeen liikekäske asetetaan nolnaan, kunnes  $MaxV \cdot A2$  täsmää. Tämän jälkeen annetaan toinen pulssi, joka on  $A2$ :n kokoinen ja pysäyttää liikkeen.

```

% Loppupään törmäyksen estokoodi
if v > 0.001 & x > 0.45
    f=F
    if x > 0.45 & F~=0
        f=0
        if F < 0 & v <= 0
            f=F
        end
    elseif x==S
        f=0
    elseif t < sqrt(complex(0.56-x)/maxA) & F==0
        f=-12000
        if v <= (maxV*A2)
            f=0
        end
    else
        f=F
    end
end
end

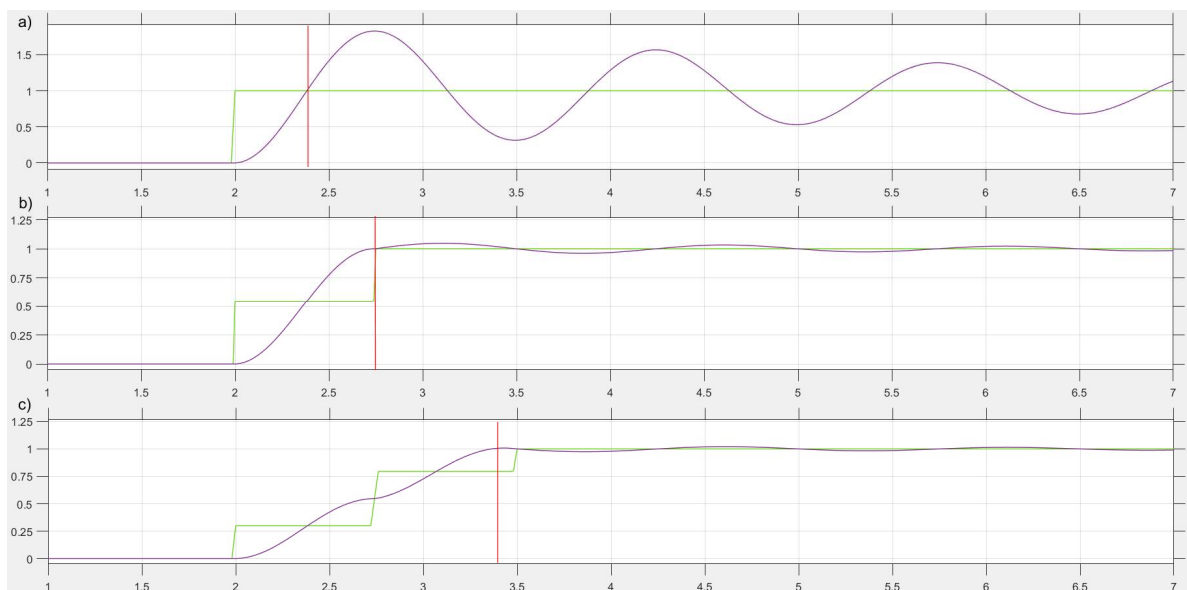
```

**Kuva 17.** Osa törmäyksenestokoodista. Käytössä on joukko sisäkkäisiä ehtoja. Näin saadaan tehtyä signaalin muokkaus ennen pysähtymistä, eikä törmätä päätyrajaan.

Lisäksi voidaan asettaa loppupisteen lähialueille pienempi maksiminopeus, joka kytkeytyy, kun alueelle ajetaan. Näin kuljettajan manuaalista ohjauskäskeä voidaan pitää päällä pidempään pysähtymismatkan ollessa lyhyempi. Tämä tietysti hidastaa päästä päähän ajamista, jossain määrin. Kuljettajasta voi tuntua epämiellyttävältä, että kone ei tee hetkeen mitä käsketään, mutta tähän tottuminen ei ihmisavoilta veisi kauaa. Manuaalikäytössä viive tulee olemaan käyttäjämukavuuteen hyvin paljon vaikuttava aspekti. Tätä voidaan myös vähentää lyhentämällä ZV-menetelmän kahden impulssin välistä viivettä, mutta tällöin jäännösvärähtely kasvaa, koska vaimennuksesta tulee epätäydellistä.

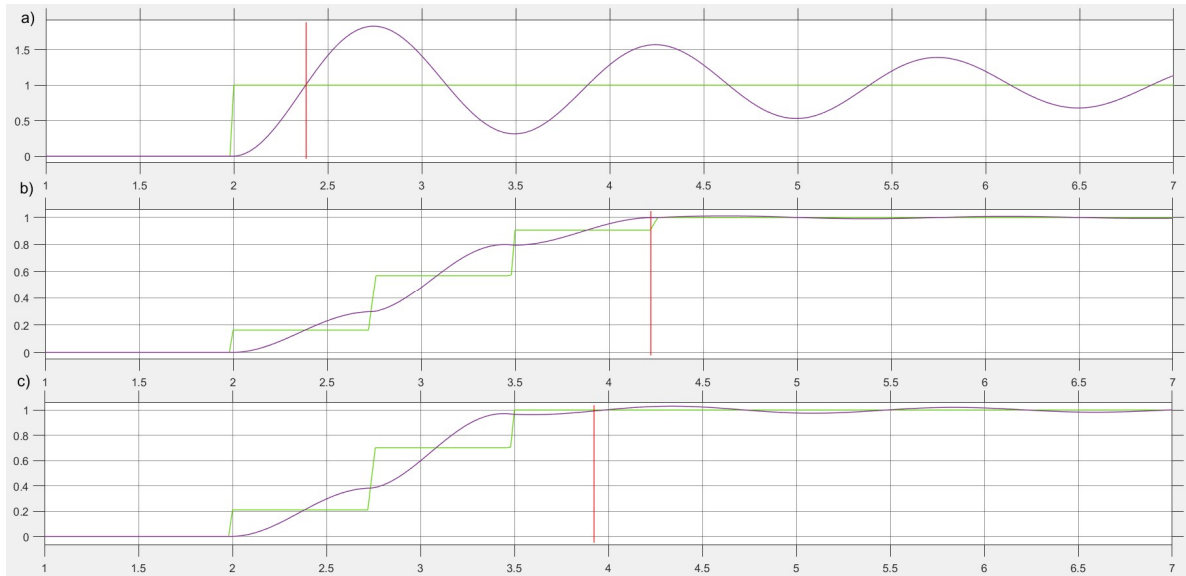
### 3 TULOKSET

Seuraavat MATLAB Simulink -simulaatioiden tulokset ovat laskettu oikeilla ominaistajuus- ja värähtelyarvoilla, mutta kiihtyvyy- ja nopeusrajoja, eikä maston pituutta otettu numeerisesti huomioon. Näin ollen numeeriset arvot, lukuun ottamatta aikaa, ovat vain vertailua varten ja prosentuaalisia. Malli on jo kuvassa 10 esitelty yksinkertainen heiluri, jolla on ennalta taulukoidut ominaistajuudet eri massoilla ja nostokorkeuksilla. Näitä ei kuitenkaan tähän työhön salassapitosyistä voinut sisällyttää. Heräte, joka järjestelmälle annetaan, on paikkatiedon askelsignaali, jossa massan paikkaa poikkeutetaan yhden yksikön verran. Seuraavissa kuvissa esitetyt vasteet on simuloitu 700 kg massalla yhdeksän metrin nostokorkeudessa. Kuvat 18, 19 ja 20 vertailevat eri muokkausmenetelmiä ylimpänä kussakin kuvassa olevaan muokkaamattomaan tapaukseen. Tässä vertailussa nousuaika lasketaan yksinkertaisesti sen mukaan, kuinka nopeasti systeemin vaste nousee käskettyyn referenssiarvoon. Kuvissa tämä tarkoittaa aikaa alkaen kahden sekunnin kohdalta, jolloin pulssi annetaan ja päättyen punaiseen pystyjanaan. Näin vasteaika saadaan eristettyä värähtelystä ja sitä voidaan käsitellä erikseen. Kuvat 18, 19 ja 20 on myös yhdistetty allekkain liitteeseen IV.



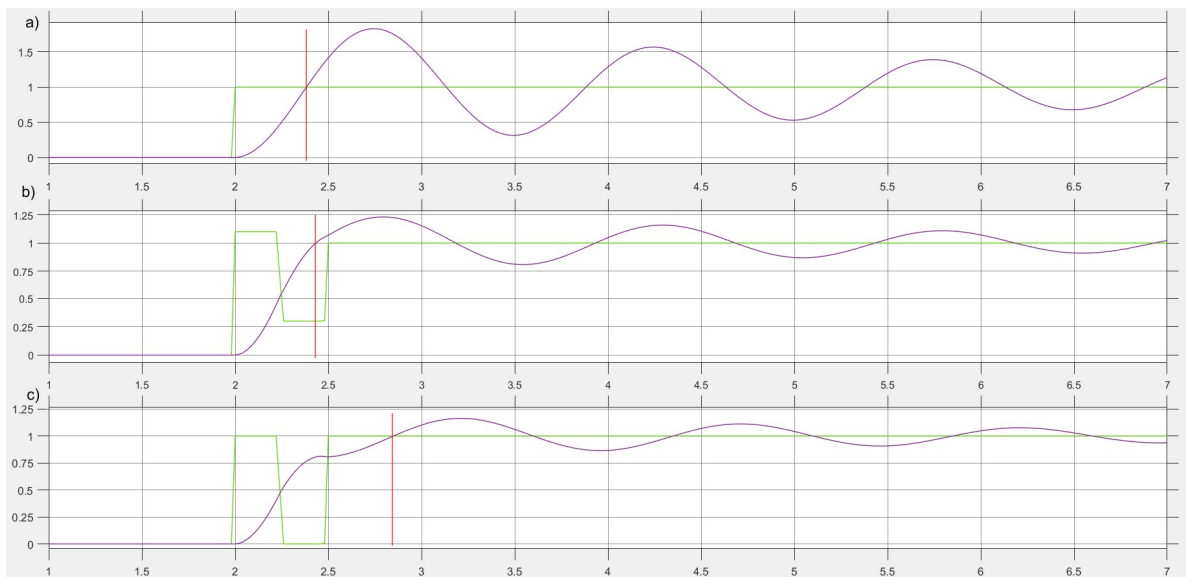
**Kuva 18.** Impulssit vihreällä ja systeemin vaste violetilla. a) muokkaamaton signaali, b) ZV-muokkain, c) ZVD-muokkain.

Kuten kuvasta 18 voi nähdä, muokkaamaton järjestelmä nousee jo alle puolessa sekunnissa haluttuun arvoon. Ylitystä tulee lähes 100 % ja jäännösvärähtelyn amplitudi jää erittäin suureksi. ZV-menetelmällä nousuaika kasvaa hieman alle sekuntiin ja ylitys, eli maksimijäännösvärähtelyn amplitudi, jää noin 5 prosenttiin. ZVD-menetelmä taas pidentää vastetta lähes 1.5 sekuntiin, mutta ylitys pysyy alle kolmessa prosentissa.



**Kuva 19.** Impulssit vihreällä ja systeemin vaste violetilla a) muokkaamaton signaali b) ZVDD -muokkain c) EI -muokkain.

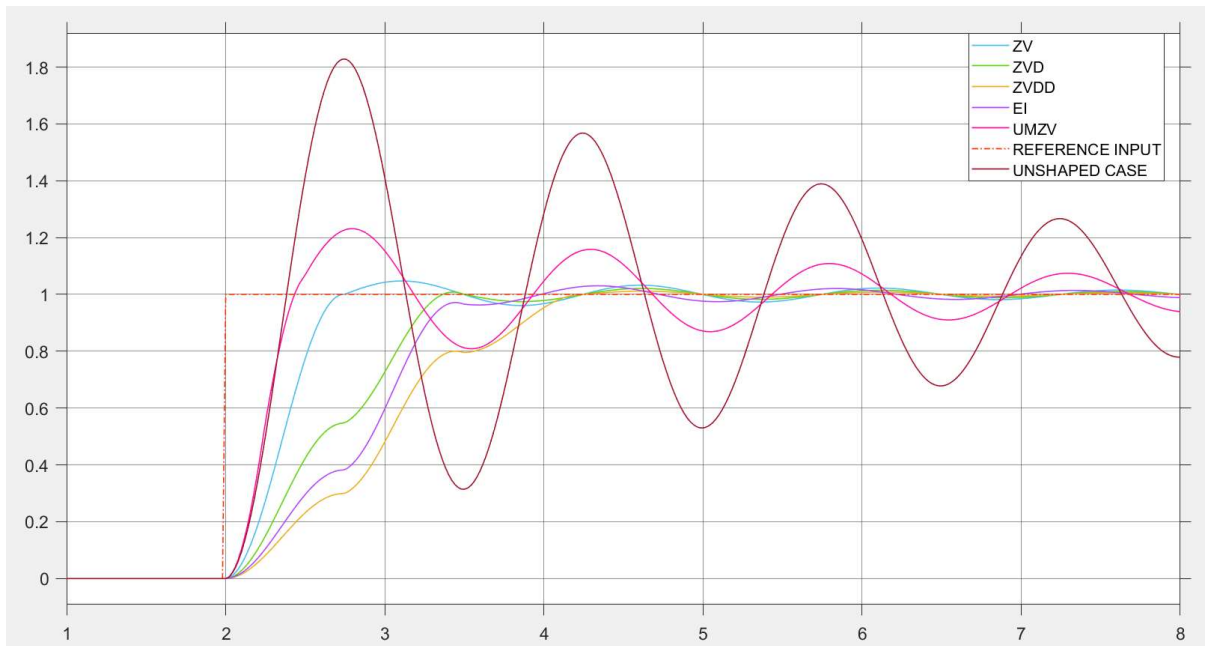
Kuvasta 19 nähdään kuinka EI-muokkain on yllättäen edellisen kuvan ZVD-menetelmää lähes puoli sekuntia hitaampi. Tämä on yllättävää, sillä teorian mukaan EI-menetelmän pitäisi olla nopeampi kuin ZVD:n, lähes samalle tasolle jäävän jäännösvärähtelyn kanssa. Edes vaimentamattomien järjestelmien EI-muokkaimella ei päästy juurikaan kuvan 19b käyrästä poikkeaviin arvoihin. Kohdassa 19c ZVDD-menetelmällä taas on kokonaisuudessaan yli kahden sekunnin vasteaika, joka on pisin kaikista, kuten teoriakin jo kertoi, mutta kun signaali nousee, ei ylitystä tule juuri ollenkaan.



**Kuva 20.** Impulssit vihreällä ja systeemin vaste violetilla. a) muokkaamaton signaali b) UMZV-muokkain optimoitu nopeaksi c) UMZV-muokkain vakioamplitudeilla.

Kuvassa 20 nähdään UMZV-menetelmän kaksi variaatiota. Kohdassa 20b järjestelmä on lähes yhtä nopea, kuin muokkaamaton järjestelmä, mutta sen ylitys on huomattavasti pienempi. Tässä tapauksessa on käytetty erikokoisia impulsseja, kuitenkin niin että niiden summa on lopulta 1. Kuvassa 20c on käytetty vain saman kokoisia impulsseja ja näin saadaan hieman parempia tuloksia amplitudin kannalta, mutta nopeus kärsii. Tuloksiensa pohjalta UMZV-menetelmä olisi paras, sillä se on nopea ja melko tehokas vaimentamaan värähtelyjä. Käytännössä sen ohjaus on luultavimmin hankalaa jo edellä mainittujen hydraulikkapiirien vasteen ja kontrollerien sykli-aikojen pituudesta riippuen. Simuloidessa tämä on helppo toteuttaa, mutta käytännössä muuttujia on paljon.

Kuva 21 yhdistää kaikkien muokausmenetelmien aiheuttamat järjestelmävasteet yhteen kuvaan. Näin saadaan yksinkertaisesti vertailtua vasteaikoja ja jäännösvärähtelyä. Kuten kuvaajastakin nähdään, ZV-muokkain on paljon useampi portaisia muokkaimia nopeampi. UMZV-muokkain on vielä nopeampi, mutta kuten jo todettiin, sille ei käytännön rajoitteiden takia ole käyttöä. Myös jäännösvärähtely on melko pientä verrattuna muokkaamattoman signaalin aiheuttamaan vasteeseen. Jo näillä kriteereillä voimme valita ZV-muokkaimen jatkojalostettavaksi tässä opinnäytetyössä. Kuitenkin UMZV-muokkain on ehdottomasti jatkotutkimuksen arvoinen.



**Kuva 21.** Eri menetelmien vasteet askelsignaaliin samassa kuvaajassa.

### 3.1 Vertailuajokierros

Tämä vertailu suoritetaan käyttämällä oikean automaattitrukin mustalaatikkodataa, josta on oikeisiin arvoihin peilaten laskettu esimerkkilastinkäsittelysykli. Näiden perusteella simuloidaan vaimentamaton malli tapaukseen 1, jotta saadaan vertailukelpoiset lähtöarvot. Tapaus 2 tulee olemaan muuten samanlainen, mutta työntöliikkeen kiihdytys- tai pysähtymisramppi muokataan. Kun tulokset molemmista saadaan laskettua, vertaillaan niitä suoraan toisiinsa. Tämän jälkeen tutkitaan ja analysoidaan koko fletin tasolla, kuinka paljon aikaa on säästetty. Kuvat

Arvoina käytetään samaa 700 kg massaa yhdeksän metrin korkeudessa. Työntörata on 500 mm pitkä ja se ajetaan päätyyn asti. Tyhjää nostokelkkaa voidaan nostaa vielä trukin liikkeessa, sillä sen huojumisamplitudi on melko pieni, sekä se vaimenee melko nopeasti pois. Näin ollen suurimmat aikasäästöt tulevat luultavimmin, kun lava pitäisi skannata Pallet Finderilla ennen sen kyytiin ottamista. Asetetaan 5 % värähtelytoleranssi muokkaamattoman tapauksen maksimiamplitudista ja käytetään sitä sallittavana värähtelynä. Kun kuorma on otettu pois hyllystä, lasketaan nostokelkka alas, joten tässä välissä ei jouduta odottamaan pitkää aikaa, sillä huojuvaakin mastoa voi laskea alaspäin. Kuormankäsittelysyklin

simulaatiossa ei oteta huomioon trukin liikkumisesta aiheutuvia värähtelyjä, koska oletuksena on, että trukki on paikallaan työntöliikettä tehdessä. Nostokelkan liikkeet ovat mukana simulaatioissa, jotta saadaan kokonaiskuva koko tapahtumaan kuluva ajasta. Seuraavissa tapauksissa ajanotto lopetetaan, kun nostokelkka on laskeutunut lattian tasoon, jotta saadaan kiinteä referenssipiste syklin loppumiselle. Arvot ovat osin mustalaatikkodatasta sovellettuja. Kuormankäsittelysykli ei ole täysin realistinen, mutta työntöliikkeen vaikutus kokonaisuuteen on täysin verrattavissa kahden samoin simuloitun tapauksen välillä. Ainoa ero ajojen välillä on signaalin muokkaus.

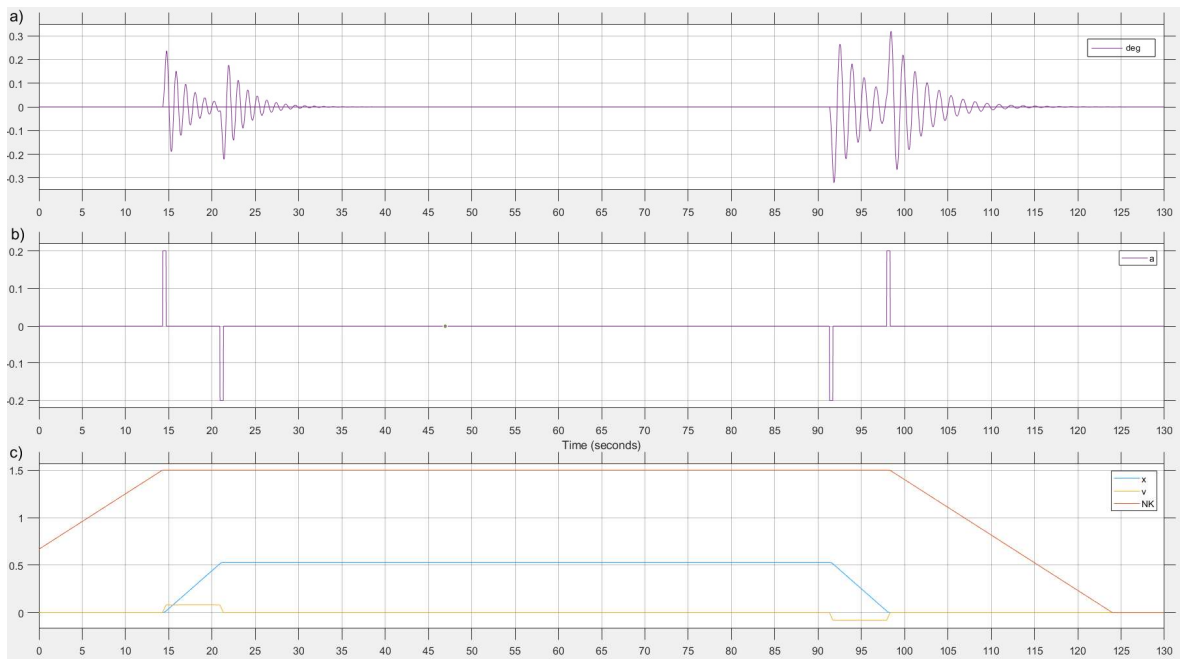
Lastinkäsittelysykli on seuraava:

- Nostokelkka nostetaan haluttuun korkeuteen.
  - Ennen pysähtymistä tyhjän nostokelkan nostaminen oli jo aloitettu, joten alkupiste on 4000 mm.
- Nostokelkka on lakipisteessään (9000 mm) ja työntö ajetaan 500 mm eteenpäin.
  - värähtelyn vaimenemista odotellaan (mikäli sellaista on)
- Kun värähtely on toleranssin sisällä, massa nostetaan kyytiin.
  - Massa otetaan haarukoille ja nostetaan (tämä osuus ei simulaatiossa, mutta lisätty aikana)
  - Kesto: 60 sekuntia.
- Työntö ajetaan takaisin alkupisteeseen ja nostokelkan laskeminen aloitetaan.
- Liikkeelle päästäisiin heti kun värähtely on toleranssin rajoissa ja nostokelkka on tarpeeksi alhaalla liikkeellelähtöä varten. Sykli on esitetty kokonaisuudessaan kuvissa 22 ja 23, eli ajanhetki 0 on alkupiste ja nostokelkan käyrän (oranssi) leikatessa nolla-akselin, on sykli ohi.

### 3.2 Tapaus 1: ei signaalin muokkausta

Kuvassa 22 on esitetty simulaation tulokset, kun signaalia ei ole muokattu. Värähtelyn maksimiampplitudi on noin 0.36 astetta. Masto on niin korkea, että näinkin pieni heilahtelu kertautuu pitkän vipuvarren kautta lähes 70 mm amplitudiksi, eli 140 mm kokonaisliikkeeksi. Tämä voi kuulostaa vähältä, mutta liike eskaloituu todella paljon mennessä suurempiin nostokorkeuksiin. 1500 kg massa nostettuna 11-metrin korkeuteen huajuukin simulaatioiden mukaan jo lähes 400 mm amplitudilla, joten koko liike kertaantuu

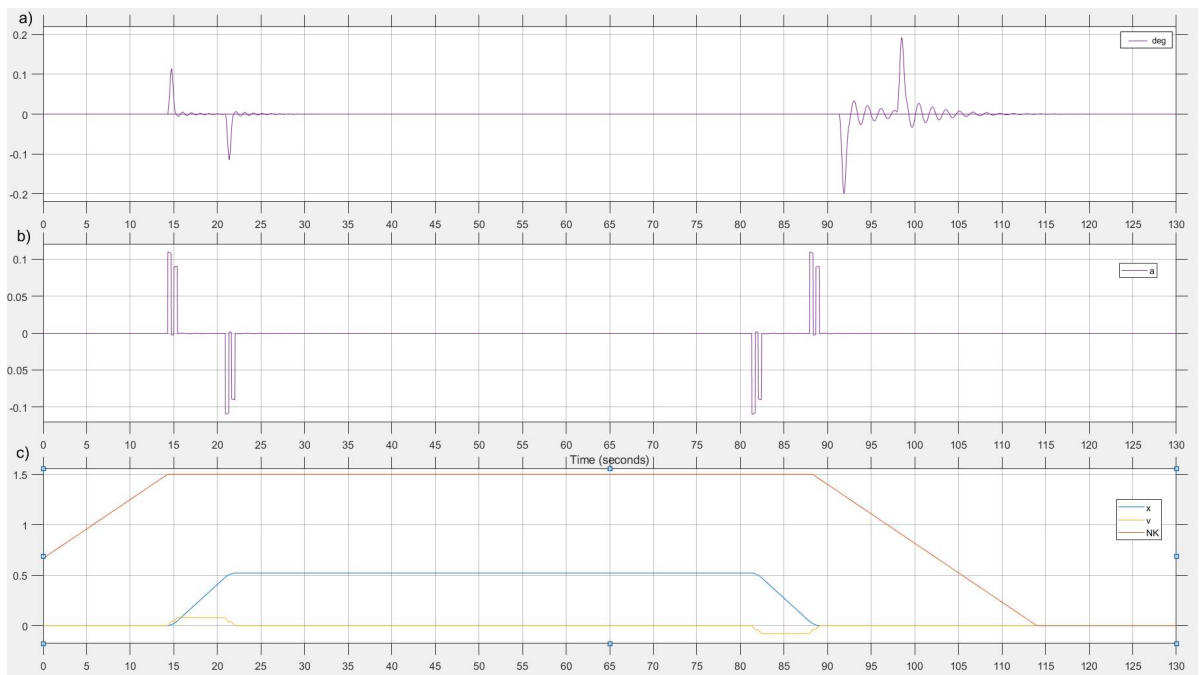
noin 800 mm liikkeeksi. Nostokelkka tyhjänä huojuminen on selkeästi pienempää kuin massan kanssa, mutta huojumista on silti huomattavasti ja sen vaimenemista joudutaan odottamaan. Taakan nostamiseen hyllystä on laskettu 60 sekuntia aikaa alkaen siitä, kun huojuminen on edellä mainitun toleranssin sisällä.



**Kuva 22.** a) Maston värähtely asteina, kun työntösignaalia ei muokattu. b) Mastolle syötetty kiihtyvyyssignaali. c) Nostokelkan korkeus oranssilla, työntöliikkeen paikkatieto sinisellä ja työnnön nopeus keltaisella.

### 3.3 Tapaus 2: ZV-menetelmä

ZV-menetelmällä muokattu tapaus on selkeä parannus edellisen kappaleen muokkaamattomaan simulaatioon. Ensimmäisen liikekäskypulssin aiheuttama heilahdus pienenee suoraan verrannollisesti ZV-muokkaimen ensimmäisen pulssin kanssa. Kuten kuvasta 23 näkee, on ensimmäinen heilahdus selkeästi isompi, kuin sitä seuraavat amplitudit. Muokkaamaton tapaus lähti vaimenemaan lineaarisesti, kun taas signaalinmuokkaimen kehittämä toinen impulssi vaimensi värähtelyn lähes täydellisesti puolen jaksonajan jälkeen.



**Kuva 23.** Aikaa säästy 9.94s koko kuormankäsittelysyklin yli. Viimeisessä kuvaajassa nostokelkan korkeus oranssilla, työntöliikkeen paikkatieto sinisellä ja työnnön nopeus keltaisella. a) Maston värähtely asteina, kun työntösignaali muokattiin. b) Mastolle syötetty kiihtyvyyssignaali. c) Nostokelkan korkeus oranssilla, työntöliikkeen paikkatieto sinisellä ja työnnön nopeus keltaisella

### 3.4 Vertailu ja data

Simulaatioista saadut tulokset ovat hyvin selkeitä, mutta silti niitä ajettiin myös eri massalla ja eri korkeuksilla edellisen kappaleen simulaatioiden lisäksi. Tulokset on esitetty taulukoissa 2-4 ja niitä avataan hieman enemmän tässä kappaleessa. Taulukossa on tyhjää suurilla nostokorkeuksilla, koska aivan ylös asti ei voida suuria taakkoja nostaa.

Taulukosta 2 näkee erittäin selkeästi, kuinka tehokas signaalinmuokkaus on simuloimassamme tapauksessa. Erityisesti matalilla taajuuksilla, alkuheilahduksen jälkeen saadaan jäännösvärähtelyä pidettyä todella pienenä. Tämä tarkoittaa yksinkertaisesti sitä, että värinän laantumista ei tarvitse odottaa ennen seuraavaa toimintoa. Datasta myös näkee, että algoritmin teho pienenee noustessa kohti suuria nostokorkeuksia, kun massana on suuri kuorma. Nämä simulaatiot ovat ajettu lähes ideaalisilla arvoilla, mutta kovin poikkeavia muuttujia ei pitäisi kentällä tulla vastaan. Tästä kuitenkin lisää seuraavassa luvussa.



*Taulukko 2. Prosentuaalinen vaimenemisajanmuutos annetusta käskystä alle 5 % jäännösvärähtelyyn. Alle yhdeksän metrin nostokorkeuksissa huojuminen ei lähes millään realistisella massalla ylitä 5 % värähtely toleranssia.*

	<b>6000 mm</b>	<b>9000 mm</b>	<b>11000 mm</b>
<b>0 kg</b>	-73,52 %	-92,17 %	-78,83 %
<b>500 kg</b>	-81,22 %	-92,31 %	-69,10 %
<b>700 kg</b>	-82,54 %	-93,02 %	-69,80 %
<b>1000 kg</b>	-86,11 %	-93,15 %	-71,66 %
<b>1500 kg</b>	-88,23 %	-95,31 %	-74,88 %
<b>2000 kg</b>	-90,37 %	-85,96 %	
<b>2500 kg</b>	-90,50 %		

Prosentuaalisesti aikasäästö vaikuttaa huikealta, kuten se onkin, mutta on korostettava, että kyseessä on ideaalitalanne. Lähes tulkoon joka tilanteessa, alle yhdeksän metrin nostokorkeudessa, kyetään odottelu-aika leikkaamaan kokonaan pois. Prosenttilukemat eivät kuitenkaan ole -100 %, sillä ZV-muokkaimen viive on vielä laskettava odotusajaksi. Tämä tarkoittaa laajasti sanottuna, että esimerkkitapauksemme kuormankäsittelysykliä on saatu lyhennettyä noin 1-18 % kun puhutaan alle yhdeksän metrin korkeuksista. Mitä korkeammalla mennään, sitä enemmän säästetään ajallisesti, vaikka prosentuaalinen vaimennusluku alkaakin kärsiä korkeissa nostoissa. Taulukossa 3 on sekunteina säästetty aika, eli kuinka paljon lyhyempi aika ollaan toimitettomana ennen kuorman käsittelyä. Kuten taulukon 3 viimeisessä 11000 mm sarakkeesta voidaan havaita verratessa taulukkoon 2, on säästetty aika sekunteina hyvin suuri, vaikka prosentuaalisesti enää noin 70-80 % ajasta saadaan vähennettyä.

*Taulukko 3. Säästetty huojumisen vaimenemisaika sekunteina.*

	<b>6000 mm</b>	<b>9000 mm</b>	<b>11000 mm</b>
<b>0 kg</b>	-0,87 s	-6,01 s	-11,08 s
<b>500 kg</b>	-1,79 s	-8,34 s	-15,25 s
<b>700 kg</b>	-2,08 s	-9,94 s	-16,38 s
<b>1000 kg</b>	-3,10 s	-11,53 s	-21,04 s
<b>1500 kg</b>	-4,12 s	-19,94 s	-37,53 s
<b>2000 kg</b>	-5,72 s	-21,50 s	
<b>2500 kg</b>	-6,35 s		

Taulukossa 4 on vielä esitetty säästetty aika prosentteissa, kun katsotaan koko kuormankäsittelysykliä. Matalissa nostokorkeuksissa odottamista ei ole paljoa, joten ajansäästöt ovat maltillisia. Koko tilanne kuitenkin muuttuu, kun mennään toiseen ääripäähän, lähes kolmasosan aikasäästö on todella suuri ja kun on kyse lähes 38:stä sekunnista, alkaa sillä olla jo suuri vaikutus sekä yksittäisen koneen kapasiteettiin, että koko fleetin kapasiteettiin. Liitteessä I on esitetty tarkemmin kuormankäsittelysykli vaihe vaiheelta ja yksittäisten simulaatioiden tasolla.

*Taulukko 4. Kuormankäsittelysyklissä signaalin muokkauksella säästetty aika prosentuaalisesti*

	6000 mm	9000 mm	11000 mm
<b>0 kg</b>	-0,73 %	-5,04 %	-9,09 %
<b>500 kg</b>	-1,50 %	-6,97 %	-12,14 %
<b>700 kg</b>	-1,74 %	-8,32 %	-13,01 %
<b>1000 kg</b>	-2,60 %	-9,64 %	-16,55 %
<b>1500 kg</b>	-3,45 %	-16,64 %	-28,56 %
<b>2000 kg</b>	-4,79 %	-17,57 %	
<b>2500 kg</b>	-5,32 %		

Näillä tuloksilla voimme lähteä laskemaan jo aiemmin mainitun 15 vaunuisen fleetin päivittäisiä aikasäästöjä. Keskimäärin kaikissa nostokorkeuksissa yli kuuden metrin, lastinkäsittelysyklin ajasta saatiin vähennettyä 9.9 %. Oletetaan siis, että 6/11 nostoista ei nopeudu ollenkaan ja 5/11 nopeutuu keskimäärin 9.9 %. Ainoastaan lastinkäsittelysyklit nopeutuvat, sillä ajonopeudet pysyvät täysin samana kuin ennen. Keskimääräinen lastinkäsittelysykli yli kuuden metrin nostoissa on noin 133 sekuntia. Oletetaan myös, että logistiikkakeskus on toiminnassa kello 06-22. Järjestelmän maksimikapasiteetti on 15 tehtävää/tunti/kone, mutta käytännössä keskimääräinen tehtävämäärä per tunti lienee noin 10 tehtävän paikkeilla. Säästetty aika lasketaan noin puolella kapasiteetilla (8 tehtävää/tunti) ja täydellä kapasiteetilla (15 tehtävää/tunti), jotta aika voidaan skaalata tarpeen mukaan tältä väliltä. Näillä arvoilla saamme laskettua, että aikaa säästyy seuraavasti:

$$\frac{5}{11} \cdot (22 - 6) \cdot 8 \frac{\text{teht}}{\text{h}} \cdot 15 \text{ konetta} \cdot 133 \text{ s} \cdot (-9.6\%) = -3 \text{ h } 6 \text{ min} \quad (31)$$

$$\frac{5}{11} \cdot (22 - 6) \cdot 15 \frac{\text{teht}}{\text{h}} \cdot 15 \text{ konetta} \cdot 133 \text{ s} \cdot (-9.6\%) = -5 \text{ h } 48 \text{ min} \quad (32)$$

Koko fleetin tasolla saavutetut säästöt ovat suuria ja kapasiteetin nousukin on merkittävä. Jos yhtälössä (31) on 8 tehtävää tunnissa, tämä tarkoittaa 7.5 minuutin keskimääräistä kestoja jokaiselle tehtävälle. Tästä voimme laskea, että puolella kapasiteetilla yllämainitun fleetin kapasiteetti nousisi 25:llä tehtävällä päivässä. Maksimikapasiteetilla toimiessa yhden tehtävän kesto on 4 minuuttia, jolloin järjestelmän kapasiteetti nousi 87:llä tehtävällä. Maksimikapasiteetti 15:llä vaunulla ja yllämainituilla aukioloajoilla on 3600 tehtävää päivässä. Näin ollen näillä säästöillä ei vielä pienennetä fleetin kokoa, mutta sillä on pitkällä aikavälillä vaikutus fleetin koneiden käyttöikäen ja sitä kautta huoltokustannuksiin. Mikäli yhdestä koneesta voitaisiin luopua, säästäisi tämä huomattavasti asiakkaan kannalta alkuinvestointia, ylläpitokustannuksia ja vikaantumiskohteita. Valmistajan kannalta taas, valmistuskustannus vähenisi sekä materiaaleina, että työtunteina, ylläpidontarve ja vikaantumistodennäköisyys pienenesi, jolloin myös ylläpitotyössä säästetään.

## 4 POHDINTA

Tämä kappale keskittyy pohtimaan ja analysoimaan edellisistä saatuja tuloksia ja havaintoja. Myös jatkotutkimussuuntauksia, sekä tämän työn rajoitusten ohi meneviä tutkimusmahdollisuuksia avataan ja analysoidaan hieman ohimenevää mainintaa tarkemmin. Erilaisia simulaatioissa esiin nousseita ominaisuuksia pohditaan ja analysoidaan tarkemmin.

### 4.1 Simulaatioanalyysi

Kuten asiaa jo aiemmin sivuttiin, pelkän teorian perusteella pystyttiin eliminoimaan kolme menetelmistä pois, niissä ilmenneiden rajoitteiden perusteella. Kaksi pudotettiin vasta mallintamisen ja simuloimisen jälkeen, jolloin piti karsia pois kaikki paitsi yksi menetelmä. Jäljelle jäi ZV-muokkain, joka yksinkertaisuudessaan jakaa tulosta saatavan käskyn kahteen portaaseen, jotka amplitudeillaan ja viiveillään johtavat jäännösvärähtelyarvoon nolla ideaalimaailmassa. ZV-menetelmän valintaan johti suurimpana syynä sen nopeus. Koska tässä käyttötapauksessa tavoitteena on nopeuttaa kuorman käsittelyä, ei ole kannattavaa käyttää monen sekunnin muokkausalgoritmeja, ainakaan matalilla nostokorkeuksilla. ZV-menetelmä tekee mitä tarvitsee, nopeasti. Esimerkiksi kuvan 21 tapauksessa, kyseinen menetelmä tasoittaa värähtelyä huomattavasti, vaikkei missään tapauksessa täysin jäännösvärähtelystä eroon pääsekään. Näin ollen tilanne on parempi, kuin ennen signaalin muokkausta, eli jotakin on saavutettu.

Korkeissa nostoissa, esimerkiksi taulukosta 2, havaittava vaimenemisen heikkeneminen oli odotettavissa oleva ilmiö. Mitä korkeammalle mennään, sitä herkempi koko järjestelmä on minimaalisillekin virheille tai epälinearisuuksille. Ilmiön saisi eliminoitua käyttämällä hitaampia muokkausalgoritmeja, mutta tällöin odottamisaika pitenisi matalammissa korkeuksissa. Korkeisiin nostoihin voisi mahdollisesti lisätä yhden impulssin enemmän, muuttamalla ZV-menetelmä ZVD-menetelmäksi, kun nousee esimerkiksi yli yhdeksän metrin nostokorkeuksiin. Algoritmit ovat melko yksinkertaiset mallintaa ja toiseen vaihtaminen ei vaatisi koodiin kuin IF-lauseen joka vaihtaa laskenta-algoritmia, kun mennään yli määrätyn nostokorkeuden. Näin saataisiin yhdistettyä matalilla korkeuksilla

nopea ja riittävän tehokas ZV-menetelmä, sekä korkeille nostoille hitaampi, mutta selkeästi tehokkaammin vaimentava ZVD-menetelmä. Tämä pari olisi paras yhdistelmä, sillä ZVD-menetelmä on seuraavaksi nopein tämän työn simulaatioiden mukaan. Tietysti myös ZVDD on vaihtoehtona, eikä sen puolentoista jaksonajan viivekään kuulosta enää kovin pahalta, jos simulaatioissa jouduttiin ZV-menetelmälläkin lisäksi odottamaan 11 metrin korkeudessa 3-13 sekuntia.

Kappaleen 3 tulokset todistavat, että mikäli kaikki arvot vastaavat todellisuutta, on signaalinmuokkaus erittäin hyvä tapa vaimentaa ohjauskäskyistä johtuvaa värähtelyä. Erityisesti automaattikäytössä, kun liikkeet suunnitellaan ennen niiden suorittamista, on huimat säästöt mahdollisia korkeissa ja painavissa nostoissa.

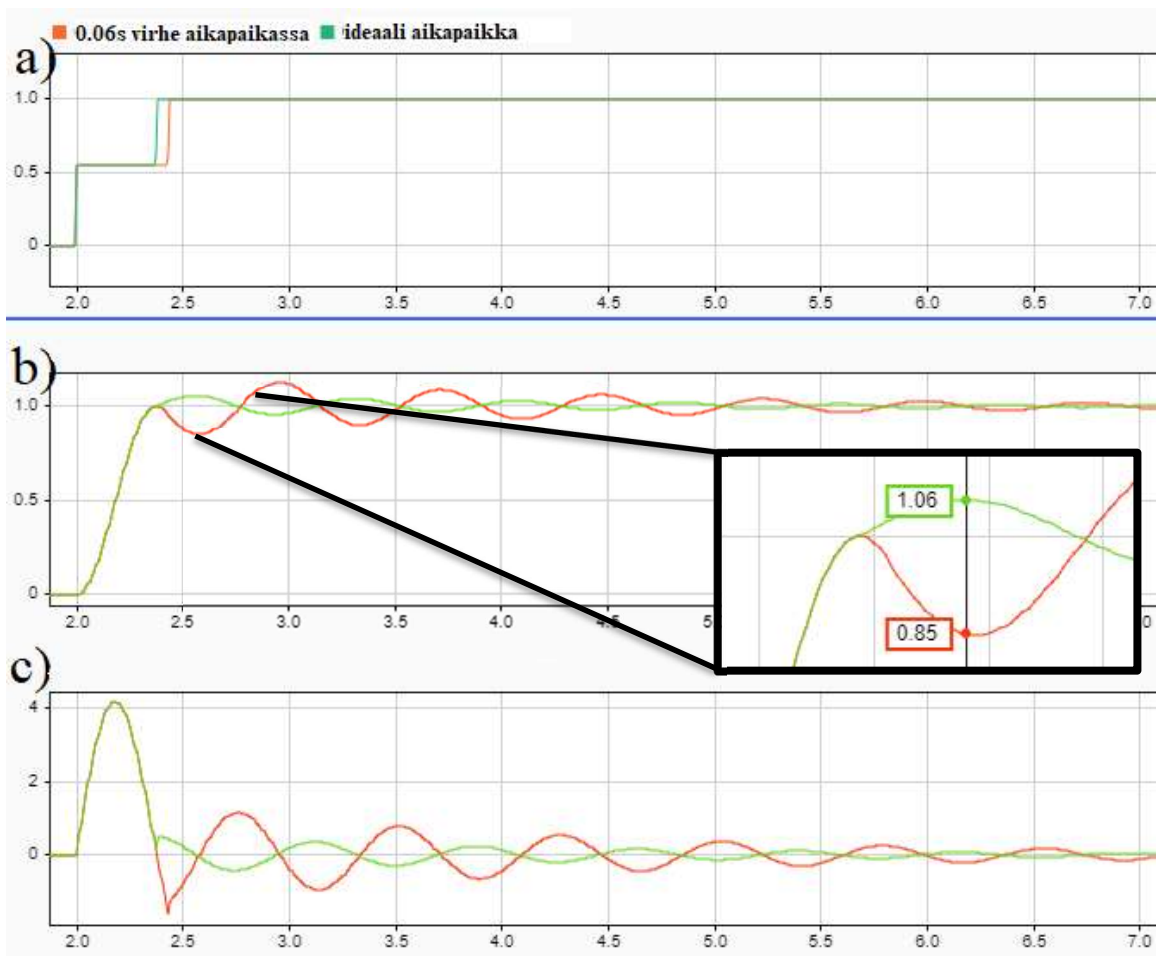
Kiihtyvyyden nostolla työntöliikkeessä ei ole paljoa saavutettavaa lastinkäsittelysyklin keston kannalta ajateltuna. Kiihdytykset ovat niin lyhyitä, että kiihtyvyyden arvo pitäisi moninkertaistaa, jotta minkäänlaista hyötyä saataisiin näkymään ajallisena muutoksena. Lisäksi, koska lähes kaikki heilurit aiheuttavat ensimmäiseen heilahdukseen melko suuren amplitudin, ei kiihtyvyyttä voi rajattomasti lisätä. Muokkaimet vain jakavat annetun käskyn osiin niin, että mitä enemmän askelia on, sen pienemmiksi pulssit pilkkoutuvat. Mikäli kiihtyvyyttä nostetaan, lähestyy ZV-muokkaimen ensimmäisen pulssin arvo koko ajan alkuperäisen pulssin arvoa ja täten myös ensimmäisen heilahduksen vastetta. Marginaalisen hyödyn takia, ei kiihtyvyyttä täten kannata lähteä juurikaan nostamaan.

#### 4.2 Kontrollerin sykliajan vaikutus

Koneessa, johon tätä algoritmia luultavimmin sovitetaan, on kontrollerin sykli aika 60 ms. Tämä asettaa tietyt rajoitteet sen suhteen, kuinka korkea taajuuksista värähtelyä voidaan vaimentaa. Sykliajan ollessa 60 ms, syklijä on noin 16.66 kappaletta sekunnissa, ja kontrolleri kykenisi mittaamaan maksimissaan puolet tästä, eli 8.33 Hz värähtelyä. Koska käskyjä annetaan algoritmin mukaan puolen jaksonajan viiveellä, tarkoittaa se, että korkeimman mahdollisen taajuuden puolikas jaksonaika on 60 ms ja loogisesti koko jaksonaika on 120ms. Näin päästään samaan 8.33 Hz taajuuteen, joka on maksimissaan mitattavissa. Samalla se on suurin taajuus, joka voidaan teoriassa vaimentaa nollavärähtelyvaatimuksella. Tämä on hyvinkin riittävä ainakin tämän työn kohteena

olevaan trukki/masto yhdistelmään, sillä mittaustuloksien perusteella taajuudet eivät nouse 2 Hz yläpuolelle. Koska ensimmäistä korkeamman moodin värähtelyt on rajattu pois jo työn alussa, ei niihinkään tarvitse kiinnittää huomiota, joten sykli aika ei tule olemaan ongelma ainakaan taajuuden puolesta.

Käytännössä täydellistä vaimennusta ei tule tapahtumaan ja simulaatioitakaan harvoin kuvaavat täydellisesti fyysisen järjestelmän käyttäytymistä. Järjestelmän sykli aika aiheuttaa myös epätäydellisyyttä signaalinmuokkausalgoritmin toteuttamisessa, sillä 60 ms on melko pitkä aika tällaisessa toteutuksessa. Jos käsky pitäisi antaa ajan hetkellä  $t$ , mutta se lähtee vasta ajanhetkellä  $t+60\text{ ms}$ , aiheuttaa tämä jäännösvärähtelyä. Kyseistä virhettä käskyn aikapaikassa voidaan myös käsitellä suoraan virheenä ominaistajuudessa. Kuvassa 24 on havainnollistettu 60 ms virheen vaikutus vaimennustapahtumaan.



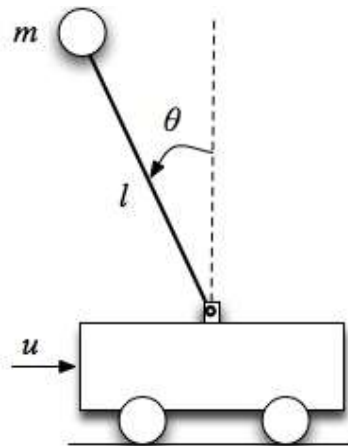
**Kuva 24.** 60 ms virheen vaikutus värähtelyn hallinnan tehokkuuteen. a) järjestelmään annettava signaali b) systeemin sijaintivaste c) systeemin kiihtyvyydvaste

Kuten kuvan 24b suurennoksesta voi lukea, lähtee virheettä muokattu signaali vaimenemaan 6 % jäännösvärähtelystä, kun taas 60 ms viiveellä ideaaliin nähden annettu käsky aiheuttaa 15 % jäännösärähtelyn. Tämä on suuri ero, mutta kuvaan on valittu tilanne, joka on pahimmasta päästä maston normaalissa käytössä. Massaa on 300 kg ja se on nostettuna vain kuuden metrin korkeuteen. Ominaisaajuus on tällöin noin 1.3 Hz, mutta mitä korkeammalle mennään tai mitä painavampaa tavaraa nostetaan, sitä pienemmäksi taajuus laskee, kuten myös prosentuaalinen virheen aiheuttama värähtely. Myös viive on kuvan 24 simulaatiossa maksimi 60 ms, eli se aiheuttaa noin 6 % virheen. Tämä on pahimmasta päästä oleva tapaus, massan ollessa matalalla ja sen ollessa kevyt. Teoreettisesti pahin tilanne sykliajan kannalta on 6 metrin korkeudelle nostettu tyhjä nostokelkka. Tällöin jaksonaika on vain 0.55 sekuntia, mikä taas johtaa suurimpaan prosentuaaliseen virheeseen (noin 11 %). Kuitenkin värähtely on edellä mainitussa nostokorkeudessa amplitudiltaan niin pientä ja nopeasti vaimenevaa, että sillä ei ole käytännön merkitystä.

Tarkan raja-arvon määrittäminen syklijalle on todella monimutkaista. Raja-arvoon vaikuttaisi nostokorkeus, nostettu massa, mikä muokkausmenetelmä on käytössä ja kuinka paljon on sallittava määrä värähtelyä. Tässä on jo niin monta muuttujaa, että raja-arvon määrittämisestä ei sinänsä olisi mitään hyötyä, koska sitä ei voida pitää absoluuttisena totuutena joka tilanteessa. Nämä asiat huomioon ottaen voidaan kuitenkin todeta, että sykli aika vaikuttaa ja se on hyvä ottaa huomioon, mutta tässä tapauksessa, pieni virhe ei haittaa. Kokonaistilanne paranee kuitenkin huomattavasti, vaikka virhe jätetään huomiotta.

#### 4.3 Jarrun vaikutus huojumiseen

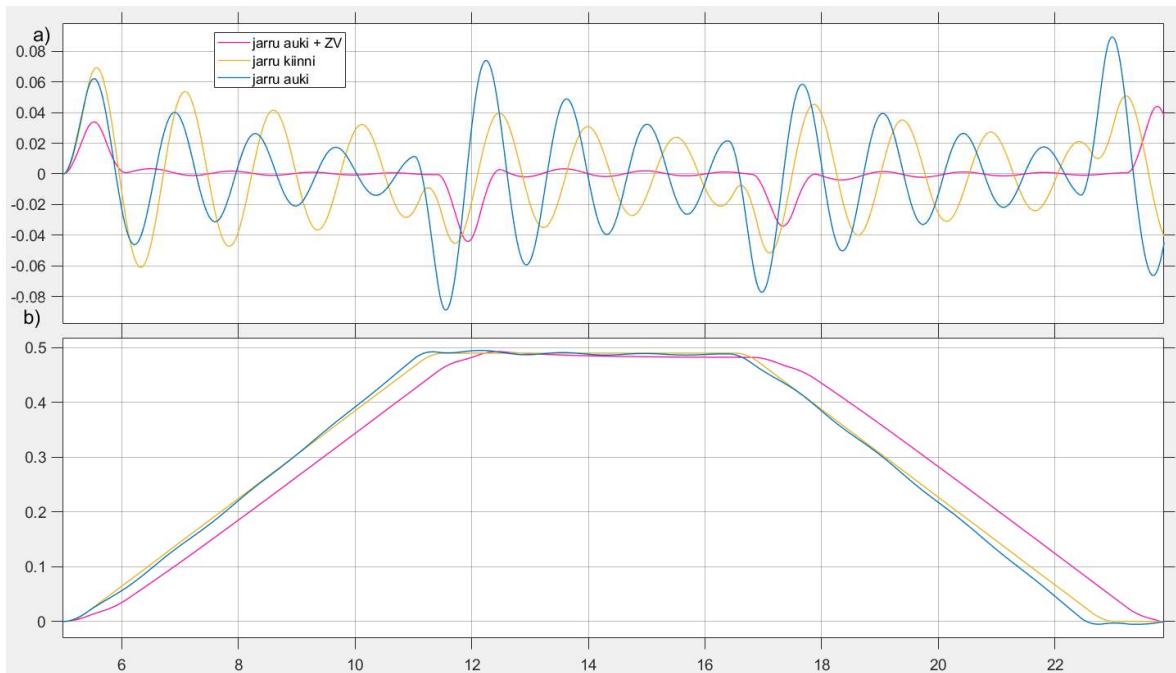
Se, onko jarru päällä vai ei, vaikuttaa huomattavasti tuloksiin. Maston korkeuden ja toisinaan suuren massan huojumisen johdosta, alkaa heiluminen liikuttaa myös maan pinnalla olevaa trukkia. Perusajatus tässä tilanteessa on kahden yksinkertaisen periaatteen summa. On siis kuvan 25 mukainen käänteinen heiluri, eli käytännössä masto, jota mikään voima ei pidä pystyssä. Masto itsessään muuttuu kuitenkin heiluriksi, kun kuvan 25 tilanteessa maston ja vaunun yhdistävään niveleen laitetaan jousivoima, joka pyrkii pitämään mastoa pystyssä. Näin maston huojumisen välittyy alustaan, jossa se on kiinni, mutta masto ei kuitenkaan kaadu kallistuessaan, vaan värähtelee työssä jo aiemmin kuvatulla tavalla.



**Kuva 25.** Käänteinen heiluri, joka kuvastaa melko hyvin ilman jarrua tapahtuvaa maston liikuttamista. (CyPhy Lab. 2010)

Vaunun ja maston liitoskohdassa oleva jousivakio pysyy teoriassa samana, mutta koska vaunu pääsee liikkumaan, on tällä selkeä vaikutus maston huojumiseen. 500 kg taakalla nostettuna yhdeksään metriin saatiin kuvan 26 mukaiset käyrät koneen kiihtyvyydestä ja maston kallistuskulmasta. Kuvassa 26 lähdetään liikkeelle 5 s kohdalla, pysähdytään 12 s kohdalla ja lähdetään taas takaisin alkupistettä kohti 17 s kohdalla. Taakka on koko ajan nostettuna 11-metriin, sekä koneen omamassaksi on laskettu 3000 kg. Vaunun lineaariseen liikkeeseen on mallinnettu 0.1 vierintäkierroin, jotta saadaan lähempänä käytäntöä olevia tuloksia. Maston huojuminen aiheuttaa jo tasaisen liikkeen aikana pientä oskillointia itse koneeseen. Myös huojumisen taajuus muuttuu hieman, sillä koneen liike kompensoi maston huojumista vastakkaiseen suuntaan. Huomion arvoista on myös se, että jarrun ollessa kytkemättä, huojuminen vaimenee selvästi nopeammin.





**Kuva 26.** ZV-muokkain toimii erittäin hyvin myös, kun jarru ei ole kytkettynä. a) heilumisamplitudi asteina b) työnnon sijainti

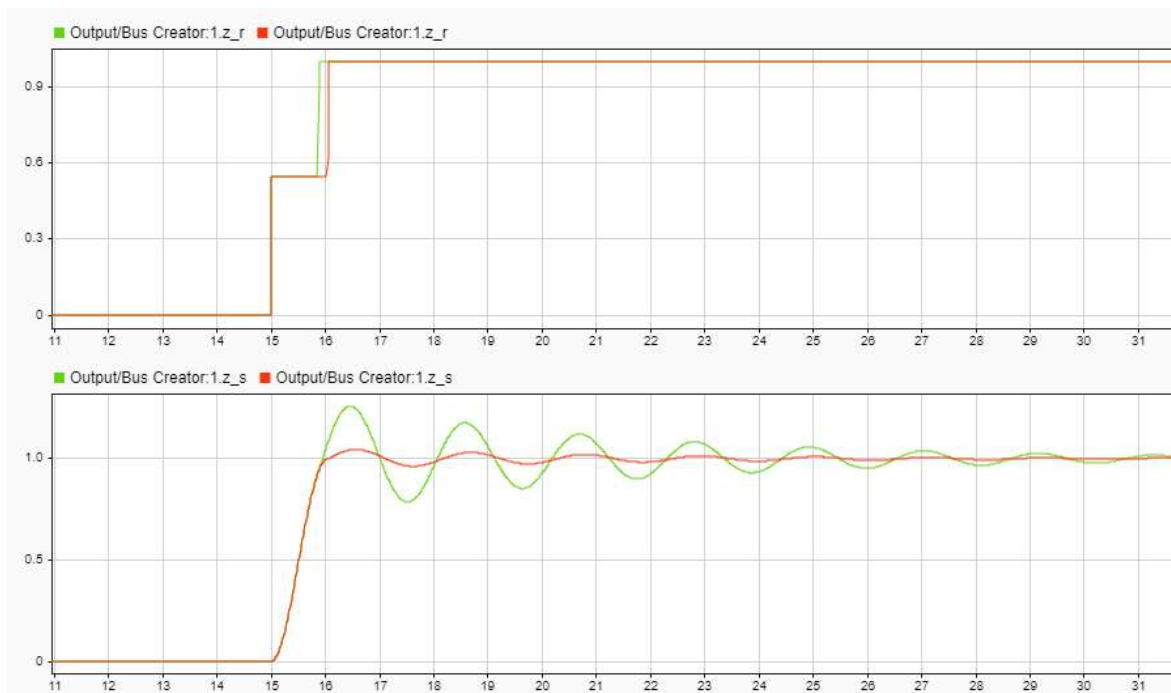
Kuten kuvasta näkee, on jarru kytkemättä olevassa tilanteessa selkeästi nopeampi vaimeneminen, sekä ensimmäisessä liikkeessä myös pienempi amplitudi. Amplitudi kuitenkin saattaa summautua jatkoliikkeissä, mikäli huojumisen aikana tehdään lisää liikkeitä, jolloin huojumista vahvistetaan tahattomasti. Koska maston huojumista on vaikea monitoroida automaattitrukissa, vaihtoehdoksi jää odottaa, että huojuminen laantuu. Tämän takia ZV-muokkain on tehokkaampi vaimennusvaihtoehto, sillä jarrun ollessa kytkemättä muokkaimen tehokkuus ei pienene ollenkaan. Yhdessä näiden vaimennuskeinojen käyttäminen on ainakin tämän simulaation perusteella entistä toimivampi ratkaisu. Jatkokehitys ajatuksena jarrulla tehtävää vaimennusta voisi tutkia pidemmälle jarrun päälle ja pois kytkemisellä puolen jaksonajan jaksoilla. Näin voitaisiin mahdollisesti parantaa vaimennusta, kun signaalin muokkaus ei ole käytössä. Tämä on kuitenkin jo niin laaja tutkimusalue, että se siirretään jatkotutkimuskohteiden alle, sillä se ei suoranaisesti kuulu enää tämän työn rajauksien alle. On kuitenkin turvallista todeta, että signaalinmuokkaus toimii myös, kun jarru on kytkemättä.

#### 4.4 Herkkyysanalyysi

Mittausdatasta tiedettiin, että oikea fyysinen järjestelmä ei ole täysin lineaarinen. Tämä huomattiin myös simulaatioissa, kun ominaistajuuksiin tuli virhettä maksimissaan noin  $\pm 5$  % riippuen siitä, oltiinko korkealla painavan massan kanssa vai matalalla ilman massaa. Tämä kuitenkin pystyttiin korjaamaan manuaalisesti korjauskertoimella jokaisessa simuloidussa korkeudessa. Itse koneessa tämän ei pitäisi olla ongelma, sillä taulukoidut arvot ovat oikeasta mastosta mitattuja ja täten virhettä ei pääse syntymään, kun seurataan näitä mittausarvoja. Simulaatioissa ongelmana oli lineaarisesti skaalautuva malli, joka kasvatti virhettä ääripäitä kohti.

Sykliajan aiheuttama virhe on suurimmillaan kevyellä kuormalla ja matalalla nostokorkeudella. Tällöin kuitenkin värähtelyn amplitudi on niin pientä ja värähtely niin nopeaa, ettei  $\pm 5$  % virhe taajuudessa ole kovin merkityksellinen. ZV-menetelmä toimii vielä kohtalaisesti näidenkin virheiden jälkeen. Toisessa ääripäässä, eli suurella massalla ja isossa nostokorkeudessa jaksonaika on lähes kolme sekuntia, jolloin sykliajan vaikutus on niin häviävän pieni, että sen aiheuttamaa virhettä (2 %) ei tarvitse ottaa huomioon. Se on hyvä tiedostaa, mutta ei vaadi tässä vaiheessa kehitystä toimenpiteitä.

Kuvassa 3 nähtiin, kuinka herkkä ZV-muokkain on virheelle. Se tuli esille myös simulaatioissa. Alla olevassa kuvassa 27 on mallinnettu +10 % virhe ominaistajuuteen, kun lasketaan viivettä. Tämä lyhentää jaksonaikaa, koska taajuus nousee, aiheuttaen näin liian lyhyen välin ZV-impulssien välille. Kuten kuvan 27 vasteesta voi huomata, kasvaa värähtelyn amplitudi järjestelmässä huomattavasti. Jäännösvärähtely kasvaa 4 % ylityksestä 25 % ylitykseen. Mikäli virhe ominaistajuudessa puolitetaan viiteen prosenttiin, aiheuttaa se vain 10 % jäännösvärähtelyn, joten nämä tulokset tukevat teoriassa esiteltyä ZV-menetelmän herkkyyttä ominaistajuudelle. On siis todella tärkeää, että taulukoidut taajuusarvot ovat mahdollisimman tarkat, kuten ne tässä projektissa ovat. Näiden tuloksien perusteella voidaan sanoa, että olisi hyvä saada pidettyä maksimivirhe ominaistajuudessa alle viidessä prosentissa. Näin saadaan pidettyä jäännösvärähtely 10 % sisällä referenssistä, joka on erittäin hyvä tulos. Myös syklijasta aiheutuva virhe sisältyy näihin toleransseihin ja on pyrittävä minimoimaan, jotta virhe ei kasva sitä kautta liian suureksi.



**Kuva 27.** +10 % virhe mallinnettuna ominaistaajuuden laskentakaavaan. Ylitys kasvoi 4 %:sta 25 %:iin. Oranssilla ideaalitalanne ja vihreällä virheellisellä ominaistaajuudella laskettu tilanne.

Yleisellä tasolla simulaatioita voidaan pitää luotettavina ja onnistuneina. Ulkoisia tekijöitä niissä ei otettu huomioon, mutta teorian tasolla on todistettu, että periaate on toimiva ja sen käyttäminen tehokasta. Kovin merkittäviä ulkoisia tekijöitä ei varastohallissa myöskään ole, joten luultavimmin tulokset ovat simulaatioiden suuntaisia. Käytännön toteutuksessa on pidettävä huolta, että taulukoidut taajuusarvot ovat mahdollisimman tarkat ja että käskyt saadaan lähtemään mahdollisimman oikea-aikaisesti, jotta saadaan parhaita mahdollisia tuloksia.

#### 4.5 Jatkotutkimusmahdollisuudet

Jatkotutkimuskohteita on tämän työn tekemisen aikana muodostunut useita, jotta tämän työn painopiste on saatu pidettyä itse signaalin muokkaamisessa. Tässä kappaleessa lyhyet kuvaukset näistä mahdollisista jatkotutkimusaiheista.

Manuaalikäytön liikkeiden optimointia ei tässä työssä lähdetty testaamaan teoriaa ja simuloiteja pidemmälle. Kuten aiemmin jo mainittiin, vaatisi tämä käyttäjätuntuman optimointia, paljon testausta ja paljon testihenkilöitä. Käyttäjän saama tuntuma on hyvin kvalitatiivinen ominaisuus, joten sen mittaaminen sekä analysoiminen olisi melko suuritöinen kokonaisuus. Sitä ei näin ollen ollut mahdollista tämän työn puitteissa toteuttaa.

Tutkimusta olisi mahdollista jatkaa myös muiden liikkeiden signaalien muokkaamismahdollisuuksien kartoittamiseen. Muiden liikkeiden signaalien muokkaamiseen on kuitenkin otettava heti alussa huomioon värähtelyn taajuus, kontrollerien sykli aika, sekä mahdolliset hydrauliiikkarajoitteet, joihin tässäkin työssä on jo törmätty. Näin tulee heti alussa varmistettua, että signaalien muokkausta voidaan käyttää kyseisen liikkeen värähtelytaajuuskaalan vaimentamiseen. Nyrkkisääntönä kuitenkin tämän työn löydöksiä perusteella on, että signaalien muokkaaminen on toimiva tapa värähtelyn vähentämiseen, kun taajuudet ovat matalia.

Kuten kappaleessa 4.3 simuloitiin, on jarrun käyttäminen vaimentamiseen yksi mahdollisuus lisävaimennusta hakiessa. Tuloksien perusteella maanpinnalla olevan vaunun heilahtelu kompensoi päinvastaiseen suuntaan ylhäällä huojuvan massan liikettä. Näin ollen, jos jarrun kytkisi päälle puoleksi jaksonajaksi ja taas pois puoleksi jaksonajaksi, vaunu liikkuisi hieman, mutta saisi puolen jakson aikana kompensoitua maston heilahdusta jonkin verran. Se, miten tämä toimisi ZV-menetelmällä muokatun signaalien kanssa, tai toimitisiko se ylipäättään, jää jatkotutkimusaiheeksi.

UMZV-menetelmä todettiin simulaatioissa todella hyväksi tuloksien perusteella. Ongelmaksi kuitenkin muodostui todella nopea impulssien välinen tauko, sekä hydrauliiikan mahdolliset rajoitteet. Ensimmäinen näistä aiheuttaa ongelmia jo heti, jos liikettä ohjaavan kontrollerin sykli aika on liian hidasta. Mikäli nämä haasteet saadaan ratkaistua, olisi työntöliikkeestä aiheutuva huojumisen vaimenemisen odottaminen mahdollisesti jopa täysin eliminoidavissa. On myös hyvä huomioda, että tämän menetelmän herättämät mahdolliset seuraavan moodin värähtelyt voivat aiheuttaa entistä suurempia ongelmia.

## 5 JOHTOPÄÄTÖKSET

Tämän työn päätavoitteena oli tutkia ja vertailla erilaisia signaalin muokkausalgoritmeja, joiden käyttäminen trukin maston huojumisen vaimentamisessa olisi mahdollista ja mahdollisimman tehokasta lastinkäsittelysyklin lyhentämiseksi. Todentamiskeinona oli pääasiassa Matlab Simulink -simulaatiot erilaisine Matlabin lisäosineen. Itse käytännöntoteutusta tai prototyypä ei tämän työn puitteissa lähdetty rakentamaan.

Tutkimuskysymykset olivat seuraavat:

- Mikä algoritmi olisi paras tähän käyttötarkoitukseen?
- Kuinka materiaalinkäsittelysykli määritellään ja minkälaisiin osiin se saadaan jaettua?
- Mikä osuus materiaalin käsittelysyklistä antaa suurimman parannukset?
- Onko materiaalinkäsittelysykli nopeampi signaalin muokkauksen ja liikkeiden optimoinnin jälkeen?
- Kuinka toteuttaa pysähtymisrampin approksimointi manuaalikäyttöön?
- Kuinka pysähtyminen radan päätyyn tehdään mahdollisimman tehokkaasti manuaalikäytössä?

Näihin kysymyksiin saatiin hyvän tasoisia vastauksia, vaikka hieman avoimiakin aspekteja jäi tietyille osa-alueille. Pääosin kysymyksiin, sekä ongelmiin saatiin hyväksyttävä vastaus. Parhaaksi algoritmiksi valittiin ZV-muokkain, sen yksinkertaisuuden ja nopeuden takia. Muut menetelmät todettiin pääasiassa joko hitauden, tai hankalan toteutuksen takia huonommiksi ja näin ollen pudotettiin pois vaihtoehdoista. Kappaleessa 3 kaikkia menetelmiä simuloitiin ja niiden tuloksia vertailtiin. ZV-muokkaimella päästään riittävän hyvään vaimennukseen suhteellisen pienellä viiveellä. Näin ollen se todettiin myös käytännön toteutukseen parhaiten sopivaksi ja sen jatkojalostaminen jatkunee tämän työn jälkeen prototyypivaiheeseen.

Materiaalinkäsittelysykli saatiin määriteltyä suhteellisen yksinkertaiseksi, eli ajaksi, jonka kone on pysähdyksissä ajosta taakan kuormaamiseksi tai purkamiseksi. Koko aika pysähtymisestä taas liikkeelle lähtemiseen laskettiin mukaan lastinkäsittelysyklin

simulaatioon, joten tämän takia säästetty aika jäi prosentuaalisesti pieneksi koko sykliä tarkastellessa. Ainoastaan värähtelyn vaimenemisaikaa tarkastellessa asia on toisin. Matalilla nostokorkeuksilla odottaminen eliminoituu kokonaan ja suurilla nostokorkeuksilla se vähenee murto-osaan.

Kuormankäsittelysyklistä saatiin selkeästi lyhennettyä odottamisaikaa riippuen nostokorkeudesta ja massasta. Syklin nopeuttamiseen kriittisin ja helpoin aikasäästö on juurikin tämä odottamisen eliminoiminen. Kiihdytysarvojen nostaminen työntöliikkeessä vaikuttaa hyvin marginaalisesti, ja huomattavan muutoksen aikaan saaminen vaatii kiihtyvyyksarvon moninkertaistamisen. Tämä oli toki arvattavissa, työntöradan ollessa vain reilun puolen metrin mittainen. Kiihdytykset ovat yleisesti niin lyhyitä, ettei niiden nopeuttaminen vaikuta kovin huomattavasti kokonaisaikaan. Suurempia säästöjä saadaan aikaiseksi liikenopeuden nostamisella riippuen liikeradan pituudesta. Nopeuden nostaminen, esimerkiksi nostoliikkeessä, säästäisi huomattavasti enemmän aikaa kuin työntöliikkeessä.

Toisen algoritmin lisääminen korkeisiin nostokorkeuksiin pitäisi ZV-menetelmän nopeuden matalilla nostokorkeuksilla ja tehokkaan värähtelyvaimennuksen korkeissa nostoissa. Kuten kappaleessa 4 jo mainittiin, joudutaan simulaation perusteella edelleen odottamaan värähtelyn vaimenemista maston ollessa nostettuna 11 metriin, vaikka ZV-menetelmää on käytetty. Noin 3-13 sekunnin odottaminen on niin pitkä aika, että vaikka käytössä olisi ZVD tai jopa ZVDD –menetelmä päästäisiin aika säästöihin. ZVD-muokkaimen lisääminen korkeisiin nostoihin on melko yksinkertaista, joten kahden algoritmin -vaimennus on ehdottomasti testaamisen arvoinen.

Manuaalikäyttö on melko hankala osa-alue ja juurikin se aspekti, joka jäi tässä työssä hieman vähemmälle huomiolle. Ihmismieli on ennalta-arvaamaton ja mahdoton ennustaa, mutta tietysti asiakkaalle halutaan myydä tuote, joka on looginen käyttää ja käyttäjäystävällinen tuntumaltaan. Tämä taas johtaa siihen, että erilaisia signaalinmuokkausalgoritmeja pitäisi testata käyttäjätuntuman kannalta suurehkoilla otoksella ihmisiä, jotta saadaan luotettavia tuloksia. Manuaalikäyttö on joka tapauksessa oltava jonkinlainen kompromissi, sillä korkeissa nostokorkeuksissa ZV-muokkaimen viive on 1-2 sekuntia, joka voi olla hidas ja turhauttava käyttäjän mielestä. Tämä tekniikka on kuitenkin jo käytössä, eikä viive käyttäjien mukaan ole huomattavan häiritsevän tuntuinen. Viiveen lyhentäminen olisi

kuitenkin toivottavaa ja radan loppuun asti mahdollisimman nopeasti ajamisessa on vielä parantamisen varaa. Tämä osa-alue työstä on siis melko pinnallisesti tutkittu, mutta kuitenkin simuloitu kuvassa 16. Tällä saralla olisi mahdollista tehdä selkeästi lisää tutkimusta aiheen laajuuden ja suuren työmäärän johdosta.

Kokonaisuudessaan tämän diplomityön tavoitteisiin päästiin onnistuneesti. Tulokset olivat lähellä odotettuja simulaatioissa ja niiden fyysinen todentaminen tapahtuu tämän työn ulkopuolella. Kuormankäsittelysykli lyheni nostokorkeudesta riippuen noin 1-29 % ja odotteluajanjaksoja saatiin lyhennettyä vähintään 69 % kaikissa nostokorkeuksissa. Oikeat tulokset tuskin vastaavat täydellisesti tässä projektissa simuloituja tuloksia, sillä oikea ympäristö ei ole ideaali, mutta ne ovat ainakin skaalattuna nostokorkeuksien välillä samankaltaiset.

## LÄHTEET

- CyPhy Lab. 2010. [Cyber-Physical Systems Laboratory -verkkosivu]. [viitattu 17.7.2020].  
 Saatavissa:  
<https://sites.google.com/a/cyphylab.ee.ucla.edu/pessoa/documentation/examples-1/inverted-pendulum>
- Kang, C.-G. 2011. On the derivative constraints of input shaping control. *Journal of Mechanical Science and Technology*. Vol. 25 (2). 549–554 s.
- Kang, C.-G. Ha, M.-T. 2014. Partially Analytical Extra-Insensitive Shaper for a Lightly Damped Flexible Arm. *IEEE International Conference on Intelligent Robots and Systems* 31 October 2014.2401-2406 s.
- Karonen, I. Jousi/vaimennin/massa –systeemin kuva. [viitattu: 14.5.2020]. Saatavilla:  
<http://vyznev.net/misc/Mass-Spring-Damper.svg>
- Maghsoudi, M. Mohamed, Z. Tokhi, M. Husain, A. Abidin, M. 2017. Control of a gantry crane using input-shaping schemes with distributed delay. *Transactions of the Institute of Measurement and Control*. Vol 39(3). 361–370 s.
- Mathworks. 2018. [Mathworks: Inverted Pendulum -esimerkki] [viitattu: 1.6.2020].  
 Saatavilla: <https://www.mathworks.com/help/control/ug/control-of-an-inverted-pendulum-on-a-cart.html>
- MLE. 2020. [Mitsubishi Logisnext Europe -verkkosivu]. [viitattu on 3.4.2020].  
 Saatavissa: <https://logisnext.eu/About.html>
- Poty, A. Melchior, P. Orsoni, B. Levron, F. Oustaloup, A. 2003. ZV and ZVD shapers for explicit fractional derivative systems.



Rocla AGV. 2020. [Rocla AGV -verkkosivu]. [viitattu on 5.5.2020]. Saatavissa: [https://www.rocla-agv.com/sites/default/files/sample-files/art\\_brochure\\_en\\_website.pdf](https://www.rocla-agv.com/sites/default/files/sample-files/art_brochure_en_website.pdf)

Shome, S.K. Mukherjee, A. Datta, U. 2013. Input Shapers against System Parametric Uncertainty. International Conference on Machine Intelligence and Research Advancement. Katra. 299-304 s.

Singh, T. Singhose, W. 2002. Input shaping/time delay control of maneuvering flexible structures. Proceedings of the 2002 American Control Conference. Vol. 3. 1717-1731 s.

Singhose, W.E. Seering, W.P. and Singer, N.C. 1996. Input shaping for vibration reduction with specified insensitivity to modeling errors. Japan-USA Sym. on Flexible Automation. 1. 307-323 s.

Singhose, W.E. Porter, L. Tuttle, T. Singer, N. 1997. Vibration Reduction Using Multi-Hump Input Shapers. Journal of Dynamic Systems Measurement and Control. Vol 119. 320-326 s.

## Lastinkäsittelysyklin simulaatiotuloksia

*Taulukko I.1. Lastinkäsittelysyklin laskennalliset ajat. Tulokset saatu Simulink simulaatiolla.*

*Nostokorkeudet 9 metriä ja 11 metriä.*

<u>@ 9m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	16,25	7,02	6,52	60,00	7,05	28,50	>>	125,34
500kg			9,03				>>	127,85
700kg			10,69				>>	129,51
1000kg			12,38				>>	131,20
1500kg			20,92				>>	139,74
2000kg			25,01				>>	143,83
<u>@ 9m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	16,25	7,02	0,51	60,00	7,05	28,50	>>	119,33
500kg			0,69				>>	119,51
700kg			0,75				>>	119,57
1000kg			0,85				>>	119,67
1500kg			0,98				>>	119,80
2000kg			3,51				>>	122,33
<u>@ 11m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	22,75	7,02	14,05	60,00	7,05	34,83	>>	132,87
500kg			22,07				>>	140,89
700kg			23,46				>>	142,28
1000kg			29,36				>>	148,18
1500kg			50,12				>>	168,94
<u>@ 11m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	22,75	7,02	2,97	60,00	7,05	34,83	>>	121,79
500kg			6,82				>>	125,64
700kg			7,08				>>	125,90
1000kg			8,32				>>	127,14
1500kg			12,59				>>	131,41

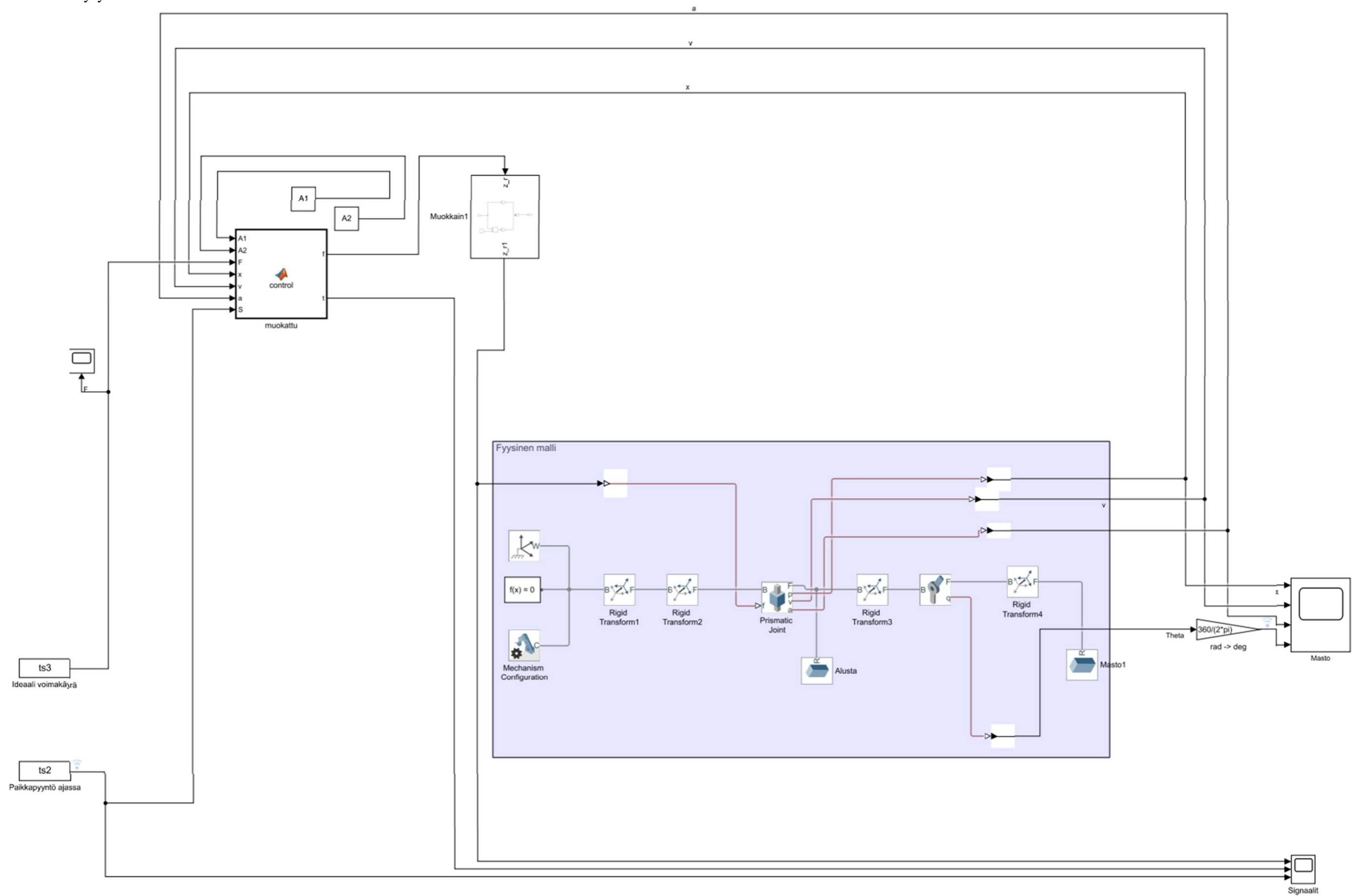
## Lastinkäsittelysyklin simulaatiotuloksia

*Taulukko I.2. Lastinkäsittelysyklin laskennalliset ajat. Tulokset saatu Simulink simulaatiolla.*

*Nostokorkeus 6 metriä.*

<u>@ 6m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	6,50	7,02	1,18	60,00	7,05	19,00	>>	120,00
500kg			2,20				>>	121,02
1000kg			3,60				>>	122,42
1500kg			4,67				>>	123,49
2000kg			6,33				>>	125,15
2500kg			7,02				>>	125,84
<u>@ 6m</u>	nosto	reach eteen	odotus	lavankäsittely	reach taakse	lasku	Liikkeelle	Kok aika
0kg	6,50	7,02	0,31	60,00	7,05	19,00	>>	119,13
500kg			0,41				>>	119,23
1000kg			0,50				>>	119,32
1500kg			0,55				>>	119,37
2000kg			0,61				>>	119,43
2500kg			0,67				>>	119,49

Kuormankäsittelysyklien simulaatiomalli



## Päätason simulaatiomallin ajotiedosto

## LIITE III

```
%Tuomas Huuskonen 17.6.2020

%Tämä koodi ohjaa simulointikärryä, jossa masto on kiinni.
%Manuaalikäyttöön, eli lähinnä rajoja päin ajamista varten. Käytännössä
%tehdään käyrä jolla oltaisiin menossa yli rajojen, jonka jälkeen seurataan
%kun koodi pelastaa tilanteen.

%% Ohjaus voimalla(lasketaan kiihtyvyyt)

%massa on nyt todella iso, jotta se ei vaikuta tulokseen. Oletuksena on että
%trukki ei liiku huojumisen takia.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Alkuarvoja
maxV = 0.08;
maxA = 0.2 ;
S=(input('Kuinka kauas mennään(mm)?:'))
a = [0:0.01:60];

x = (round((S/maxV),1)*100);      %aika loppupisteeseen maksimi vauhtia
F = ((60000)*maxA)              %F=ma

%Rakennetaan kiihtyvyyt käyrä kysytyyn pisteeseen S pääsemiseksi
b4 = zeros(500,1);              %Nollasignaalia
b5 = F + zeros(round(maxV*100/maxA),1);      %F-signaalia
b1 = zeros(round(x-maxA*length(b5)),1);      %nolla signaalia
b2 = zeros((floor(((length(a)-length(b1)))/2)),1);      %nolla signaalia
b3 = zeros((floor(((length(a)-2*length(b5)-2*length(b4)-2*length(b1))))),1);      %nolla signaalia
%nolla signaalia
b6 = S + zeros(((length(a)-length(b4)-length(b3))),1); %S-kokoista signaalia
if S > 0.5
    b = [b4 b5 b1 b4 -b5 b1 b3];
else
    b3 = zeros((floor(((length(a)-4*length(b5)-2*length(b4)-2*length(b1))))),1);
    b = [b4 b5 b1 -b5 b4 -b5 b1 b5 b3];
end

%muodostetaan timeseriet maston ja kelkan liikkeille
C = [b4 b6 b3];

ts2 = [a(:), C(:)];              %aika + signaali matriisiksi
timeseries(ts2);
ts3 = [a(:), b(:)];              %aika + signaali matriisiksi
timeseries(ts3);
disp('Nyt voi simuloida')
```

