

Metropolia Ammattikorkeakoulu
Tietotekniikan koulutusohjelma

Jari Toiviainen

**Automaatiotestauksen tarve hammashoitokoneen
ohjelmistokehityksessä**

Insinööriyö 2.2.2010

Ohjaaja: projektipäällikkö Matti Setälä
Ohjaava opettaja: lehtori Anssi Ikonen

Metropolia Ammattikorkeakoulu Insinööriyön tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Jari Toiviainen Automaatiotestauksen tarve hammashoitokoneen ohjelmistotestauksessa 37 sivua 2.2.2010
Koulutusohjelma	tietotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	projektipäällikkö Matti Setälä lehtori Anssi Ikonen
<p>Tämä opinnäytetyö tehtiin Planmeca Oy:n Hammashoitokone-divisioonan tuotekehitysosastolle. Opinnäytetyössä arvioitiin perusteita ja tarvetta Sovereign-hammashoitokoneen ohjelmistotestausautomaatiolle. Manuaalinen testausmenetelmä oli osoittautunut puutteelliseksi ja hankalaksi.</p> <p>Haluttuun automaatioasteeseen perustuen ehdotettiin puoliautomaattista tai täysautomaattista toteutustapaa, tavoitteena tehostaa testausprosessia ja lisätä testauksen kattavuutta.</p> <p>Tässä opinnäytetyössä tarkasteltiin testaukseen käytettävän laitteiston kehitystä ohjelmistoineen vain periaatetasolla. Työtä voidaan hyödyntää perusteltaessa, miten hammashoitokoneen ohjelmistotestausta voidaan parantaa ja tehostaa testiautomaatiolla sekä arvioitaessa hammashoitokoneen ohjelmistotestiautomaation kehityssuuntaa.</p>	
Hakusanat	ohjelmistotestaus, testaus, automaatio, hammashoitokone, CAN-väylä, Linux

Helsinki Metropolia University of Applied Sciences Abstract

Author	Jari Toiviainen
Title	Test automation need in software testing of a dental care unit
Number of Pages	37 pages
Date	2 February 2010
Degree Programme	Information technology
Degree	Bachelor of Engineering
Instructor Supervisor	Matti Setälä, Project Manager Anssi Ikonen, Principal Lecturer
<p>The thesis project was carried out for the product development department of Planmeca Oy, Dental Care Units Division. The goal of the thesis was to evaluate the basic grounds and need for developing automation to be used in the software testing of the Sovereign dental care unit, a subject necessitated by the inadequacy and complexity of manual software testing.</p> <p>Based on the chosen level of automation, a fully or semi-automatic testing approach was suggested aiming to enhance and broaden the scope and extent of the testing process.</p> <p>The test bench hardware development was studied in this work only on a theoretical level. The results of this thesis can be utilized when justifying the improvement and enhancement of software testing with test automation, and for planning the development path for the software test automation of the dental care unit.</p>	
Keywords	software testing, testing, automation, dental care unit, CAN-bus, Linux

Lyhenteet

ACCU	Advanced Centralized Controlling Unit; kontrolliyksikkö
CAN	Controller Area Network; automaatioväylä
GUI	Graphical User Interface; graafinen käyttöliittymä
HERCO	Headrest Controller; pääntuen ohjaus
ICON	Instrument Controller; instrumenttikontrolliyksikkö
NGU	Next Generation Unit; seuraavan sukupolven yksikkö
PEMS	Programmable Electronic Medical System; ohjelmoitava lääketieteellinen yksikkö
LAN	Local Area Network; paikallisverkko
MAMCO	Multiple Axis Motor Controller; moottorinohjausyksikkö
OLCO	Operation Light Controller; operointivalo
SLED	Single LED; operointivalo
SUCO	Suction Control Unit; imukontrolliyksikkö
TAR	Tape Archive; tiedostomuoto
USB	Universal Serial Bus; sarjaväyläarkkitehtuuri
WMC	Water Management Control; vesipuolen hallintayksikkö

Sisällys

Tiivistelmä

Abstract

Lyhenteet

1	Johdanto	6
2	Sovereign-hammashoitokoneen tekniikka	7
2.1	Sovereignin tekninen yleiskuvaus	8
2.2	CAN-väylä	13
3	Ohjelmistotestaus	14
3.1	Ohjelmistotestauksen yleisiä periaatteita	14
3.2	Hoitokoneen ohjelmistovirheiden mahdollinen vaikutus	17
3.3	Ohjelmiston testaus manuaalisesti	18
4	Testiautomaation kehitystarve	23
4.1	Testiautomaation kustannukset	26
4.2	Yhteenveto testiautomaatiotarpeesta	27
5	Testiautomaation kehityssuunnan arviointia	28
5.1	Ohjelmistotestitapausten toteutustapa	28
5.2	Testiautomaatiovertailu	31
5.3	Testilaitteiston käyttöliittymä	34
5.4	Automaatio	35
6	Päätelmät	36
	Lähteet	37

1 Johdanto

Tämän insinööriyön tavoitteena on arvioida ohjelmistotestauksen automatisointitarvetta perustuen manuaalisen testauksen puutteellisuuksien ja työmäärän analyysiin. Tämän analyysin tuloksena on päädytty hahmottelemaan mahdollista automatisoinnin toteutusta tarkasteltavassa ohjelmistotestauskohteessa.

Työ tehtiin Planmeca Oy:n tuotekehitysosastolle, jossa projekti hammashoitolaitteen ohjelmistotestauksen automatisoimiseksi on alkamassa.

Planmeca Groupin suomalainen emoyhtiö Planmeca Oy suunnittelee, valmistaa ja markkinoi hammashoitolaitteita, hoitoyksiköitä ja niiden oheistuotteita, röntgenlaitteita sekä digitaalisen kuvantamisen ratkaisuja ja ohjelmistoja. Planmeca Oy on maailman kolmanneksi suurin hammaslääketieteen laitevalmistaja sekä alansa suurin perheyrittys. Planmecan tuotteissa hyödynnetään tieto- ja kuvantamistekniikoita. Planmeca on laitevalmistaja, joka tarjoaa monipuolisen tuoteratkaisun potilastyöhön ja hammaslääkärin vastaanoton tiedonhallintaan, johon hoitoyksikön ja kuvantamislaitteiden lisäksi nivoutuvat kaikki tarvittavat ohjelmistot. Planmeca tekee tuotekehitystä yhteistyössä yliopistojen ja korkeakoulujen kanssa. [1.]

Hammashoitokoneen instrumenttivalikoiman monipuolistuessa tarve konfiguroitavuuteen lisääntyy. Asiakkaiden kysyntään pyritään vastaamaan mahdollisimman tarkkaan sellaisilla ominaisuuksilla, joita markkinoilla halutaan. Nämä uudet ominaisuudet lisäävät hoitokoneen ohjelmiston testaustarpeita.

2 Sovereign-hammashoitokoneen tekniikka

Testiautomaatiotarpeiden ymmärtämiseksi on tärkeää hahmottaa Sovereign hammashoitokoneen tekniset ratkaisut. Tässä luvussa kerrotaan, mistä moduuleista ja väyläratkaisusta Sovereign rakentuu.

Planmecan Sovereign-hammashoitoyksikkö (kuva 1) on kokonaisuus, jossa koko yksikön ja potilastuolin liikkeet ovat motorisoituja. Tämä mahdollistaa sen, että hammashoitokone mukautuu käyttäjän ja potilaan tarpeisiin sekä kaikkiin työympäristöihin ja toimenpiteisiin. Kaikki hoitoyksikön toiminnot ovat symmetrisiä, mikä mahdollistaa motorisoidun muutoksen oikeakätisestä vasenkätiseen käyttöön, eli potilastuolin ajamisen kääntömoottorilla oikeanpuoliselta hoitoalueelta vasemmanpuoliselle hoitoalueelle. Hoitoalueet ovat sektoreita, joilla tuolin selkänojan ja nostomoottorin liikkeet ovat sallittuja hoitokoneen logiikan mukaan.

Konfiguroitavissa olevan graafisen kosketusnäytön ansiosta hoitoyksikkö soveltuu hyvin hammaslääkärin työhön ja sen vaatimuksiin. Sovereign-hammashoitoyksikkö ei kilpaile markkinoilla olevien ns. halpakoneiden kanssa, vaan pyrkii kilpailemaan monipuolisilla teknisillä ratkaisuilla. Sovereign-hammashoitokoneen tuotekehitysvaihe on päättynyt, ja se on siirtynyt ylläpitovaiheeseen. [2.]



Kuva 1. Sovereign-hammashoitoyksikkö [2]

2.1 Sovereignin tekninen yleiskuvaus

NGU (Next Generation Unit) Sovereign on sähköisesti ohjattu lääketieteellinen hammashoitoyksikkö (PEMS). Sovereign-hammashoitokoneen käyttöjärjestelmä on Linux. Hoitokoneen käyttöjärjestelmä ja ohjaus sijaitsevat ACCU-ohjainkortilla, jonka alaisuudessa muut toimilaitteet toimivat. ACCU-ohjainkortin ohjelma on Linux-sovellus, jonka laajuus on useita satoja tuhansia koodirivejä. Hammashoitokoneen sisäisenä viestintäkanavana toimilaitteiden ja ACCU-ohjainkortin välillä toimii CAN-väylä. [3.]

Kuvassa 1 potilastuolin yläpuolella näkyy instrumenttitarjotin, jossa on kuusi paikkaa hammaslääkärin instrumenteille. Instrumentit ovat kytkettävissä pikaliittimillä instrumenttitarjottimeen. Instrumentteja voidaan lisätä ja poistaa virran ollessa päällä laitteessa. Kuvassa 1 vasemmalla on hoitajan yksikkö, josta lähtee kolme letkua pitimiinsä.

Ohjausyksikkö

ACCU-ohjainkortti (Advanced Centralized Controlling Unit) hoitaa järjestelmänlaajuisia toimintoja, kuten turvasilmukan hallintaa, hätäpysäytyksen toimintaa, tehon syöttöä alijärjestelmille sekä yksikön pesua ja huuhtelua. ACCU-ohjainkortti tunnistaa kokoonpanon ja alustaa alijärjestelmät.

ACCU-ohjainkortilla käsitellään seuraavat tehtävät:

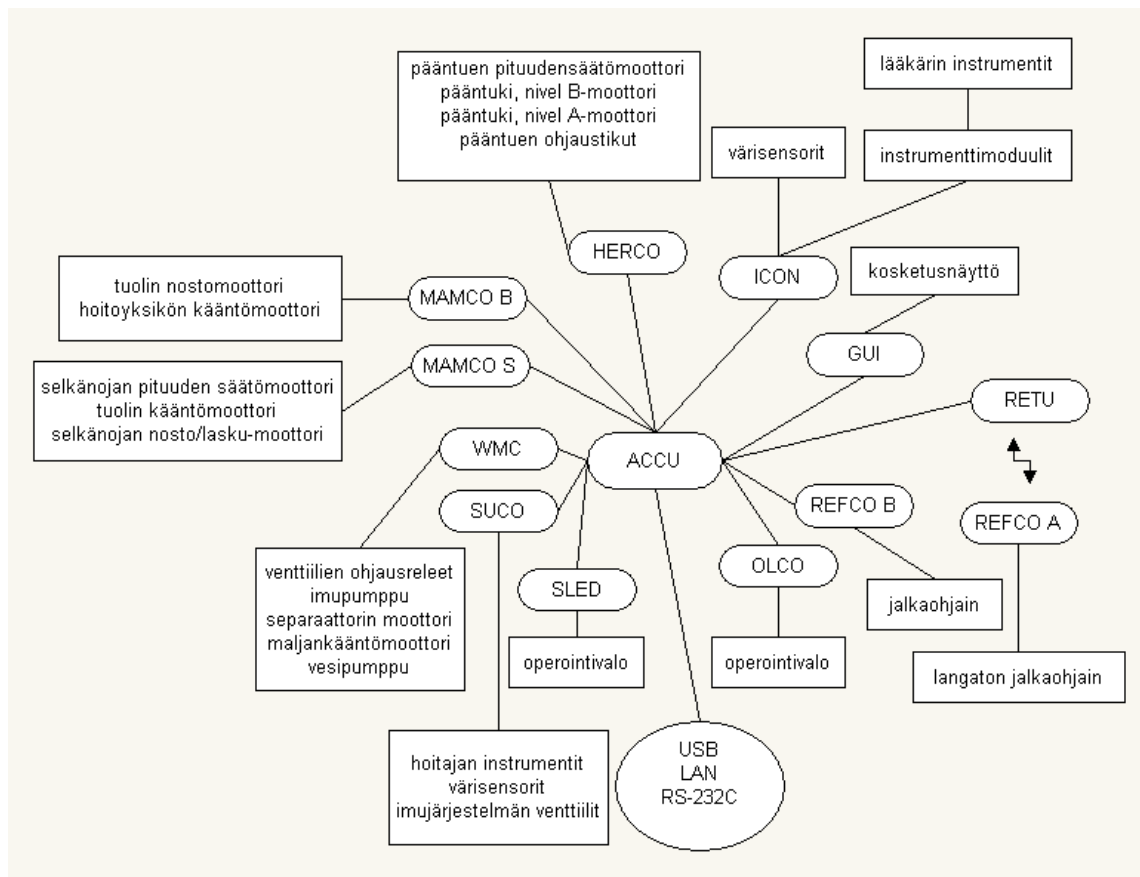
- järjestelmän käynnistys ja alustus
- järjestelmän hallinta
- järjestelmän tilojen hallinta
- ohjelmiston päivitys
- käyttöliittymän käsittely ja hyödyntäminen
- turvamekanismin käsittely
- tilakirjanpidon hallinta.

ACCU-ohjainkortti ohjaa myös hoitokoneen ohjelmistopakettien päivityksen, joka suoritetaan liittämällä Linux-arkistotiedosto (TAR-tiedosto) USB-muistin kautta ACCU-ohjainkortille ja käynnistämällä päivitysprosessi hoitokoneen kosketusnäytöltä. (TAR-tiedosto sisältää kaikki toimilaitteohjelmat). [4.]

Sovereign-hammashoitokoneen toiminnalliset osat

Hammashoitokoneen keskeiset toiminnalliset osat ovat motorisoitu potilastuoli, operointivalo, imu- ja erottelujärjestelmä (separaattori) ja instrumenttikontrolleri. Hoitokone käsittää vesijärjestelmän, jonka tarkoituksena on sisääntuloveden eristys, koska hoitokoneen vesijärjestelmä ei saa olla suorassa yhteydessä vesijohtoverkkoon bakteerivaaran vuoksi. Vesijärjestelmän ohjauksesta vastaavat WMC- ja SUCO-kortit (kuva 2). Graafisella kosketusnäytöllä (GUI) ja jalkaohjaimella ohjataan instrumentteja sekä potilastuolin liikkeitä. Pääntuen ohjaustikuilla ohjataan pääntuen asentoa ja pituutta. Kuvasta 2 käy ilmi, miten toimilaitteet kytkeytyvät itse toimilaitteisiin sekä ACCU-ohjainkorttiin, joka kontrolloi järjestelmää. Kuvassa 2 näkyvät langaton ja langallinen jalkaohjain ovat vaihtoehtoisia, eli molemmat eivät saa olla yhtä aikaa

kytkettynä järjestelmään, koska järjestelmä on suunniteltu käytettäväksi vain yhdellä jalkaohjaimella. Sama koskee operointivalaisimia OLCO ja SLED, joista jompikumpi on valittava hoitokoneen operointivaloksi. Instrumenttien ajo edellyttää, että instrumentti on valittu aktiiviseksi. Tämä voi tapahtua instrumenttivarren taivutuksella tai instrumenttipitimen värisensorilla riippuen siitä, kumpi laiteratkaisu on kyseessä. Värisensori lukee instrumenttipitimestä lähtevän sinisen ja punaisen LED-valon suhdetta instrumenttiholkista palautuvasta heijastuksesta. Heijastuksesta hoitokone pystyy päättelemään, onko kyseinen instrumentti, esimerkiksi hammaspaikkamateriaalin valokovetin, aktiivinen vai passiivinen kyseisellä hetkellä.



Kuva 2. Sovereign-hammashoitokoneen periaatteellinen lohkokkaavio

Hoitokoneen ohjelmisto

Hammashoitokoneen ohjelmisto aktivoi kulloinkin tarvittavia toiminnallisia osia kuten vesijärjestelmä, moottorit ja vastaavat järjestelmästä, käyttäjän tekemien valintojen perusteella. Ohjelmistot on toteutettu C- ja C++ -kielillä ISO 9000 -standardin 60601-1-4 mukaan, joka koskee yleisiä kansainvälisiä turvallisuusvaatimuksia ohjelmoitavissa sähköisissä lääketieteellisissä järjestelmissä. Turvaominaisuuksien johdosta hoitokoneen ohjelmistossa on määritelty, mitkä toiminnot ovat kulloinkin mahdollisia. Potilastuolin moottoriajo on suojattu turvaominaisuuksilla, moottoriajo ei ole mahdollisia, kun lääkärin instrumentti on aktiivinen ja päinvastoin. Pelkästään tuolin liikkeissä on paljon ohjelmallisia turvallisuusrajoituksia, koska tuolin nostovoima on 300 kilogrammaa. Ohjelmisto valvoo moottorien ajoa tarkkailemalla reaaliajassa asema-anturien tiloja, moottorien ylivirtoja, aikarajoja, asema-anturien ja hall-elementtien ristiriitoja.

Ohjelmoitavat toimilaitteet

Kuvasta 3 on havaittavissa ACCU-ohjainkortin alainen hierarkia. Toimilaitteet ovat liittyneet ACCU-ohjainkorttiin, ja toimilaitteet, kuten moottorit ja käyttöliittymät ovat toimilaitteiden alaisuudessa. Toimilaitteet sisältää itsenäisen tietokoneen, jolla on oma mikroprosessori, muisti ja liityntä automaatiiväylään.

Kaikki toimilaitteet käyttävät CAN-väylää viestintään, ja niillä on oma toimilaitteohjelma. Toimilaitteet ohjaavat toimilaitteita, kuten moottoreita ja venttiilejä, sekä lukevat hoitokoneen anturien tilaa.

Graafinen käyttöliittymä (GUI) piirtää grafiikkaa näytölle ja lukee kosketuspaneelia. GUI liittyy CAN-väylällä muuhun järjestelmään.

Pääntuen ohjaus (HERCO, Headrest Controller) ohjaa kolmea pääntuen moottoria ottaen vastaan komentoja CAN-väylän kautta tai toimien ohjaustikkukontrollissa.

Instrumenttikontrolli (ICON, Instrument Controller) ohjaa instrumenttimoduuleja, lukee antureita ja tunnistimia. ICON myös kytkee sähköjä instrumenttipaikkoihin ja tunnistaa instrumentit lukemalla instrumenttiletkuissa olevilta muistipiireiltä tunnisteiden.

Moottoriohjain (MAMCO, Multiple Axis Motor Controller) ohjaa potilastuolin moottoreita ja koko hammashoitoyksikön kääntömoottoria sekä lukee moottorien asematunnistimia, joita ovat säädettävät vastukset eli potentiometrit sekä magneettiset anturit eli hall-elementit.

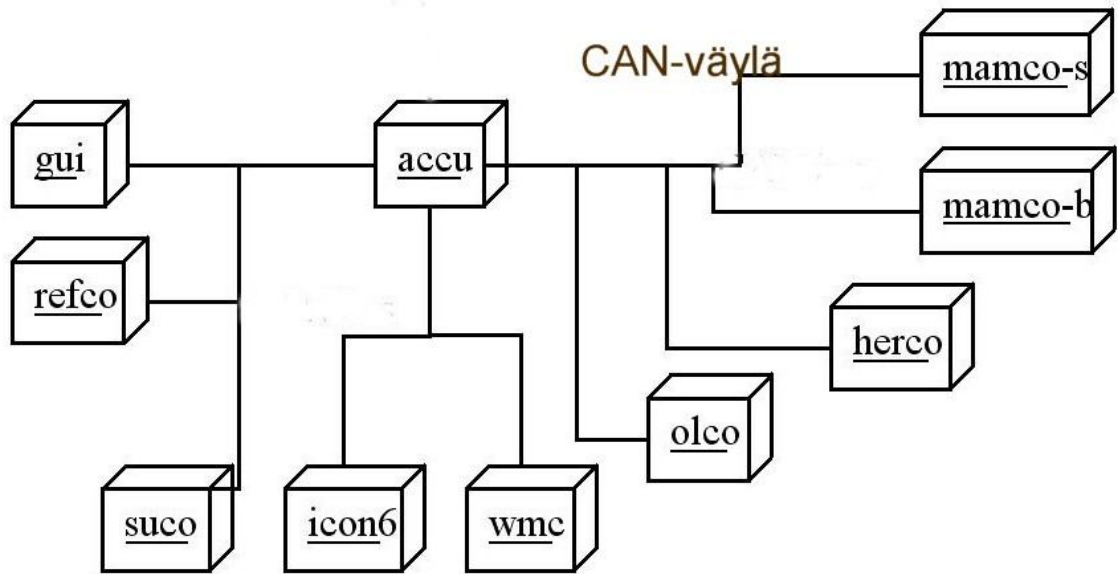
Operointivalo (OLCO, Operation Light Controller) ohjaa operointivaloa ja lukee optista etäisyystunnistinta kytkeäkseen valon päälle, eli valon voi sytyttää ja sammuttaa käden heilautuksella operointivalon edessä.

Operointivalo (SLED, Single LED Operating Light) on yhteen LED:iin perustuva valaisintuote, jolla on samat päällekytkentä ja kirkkaudensäätötoiminnot kuin OLCO:ssa.

Jalkaohjain (Foot Controller System) ohjaa instrumentteja. Jalkaohjaimella voidaan tehdä osin samoja valintoja kuin kosketusnäytöltäkin, mutta poran käynnistys voi tapahtua vain jalkaohjaimella. Tuolin liikkeitä voidaan ohjata jalkaohjaimella. Jalkaohjaimesta on myös ”langaton versio”, joka kommunikoi radiotaajuuksilla.

Imukontrolli (SUCO, Suction Controller Unit) ohjaa hoitajan käyttämiä instrumentteja.

Vesikontrolli (WMC, Water Management Controller) kontrolloi veden pinnan tasoa vesisäiliössä, käynnistää tarvittaessa imupumpun sekä avaa ja sulkee venttiileitä tarpeen mukaan. [5.]



Kuva 3. Järjestelmäarkkitehtuuri [5]

2.2 CAN-väylä

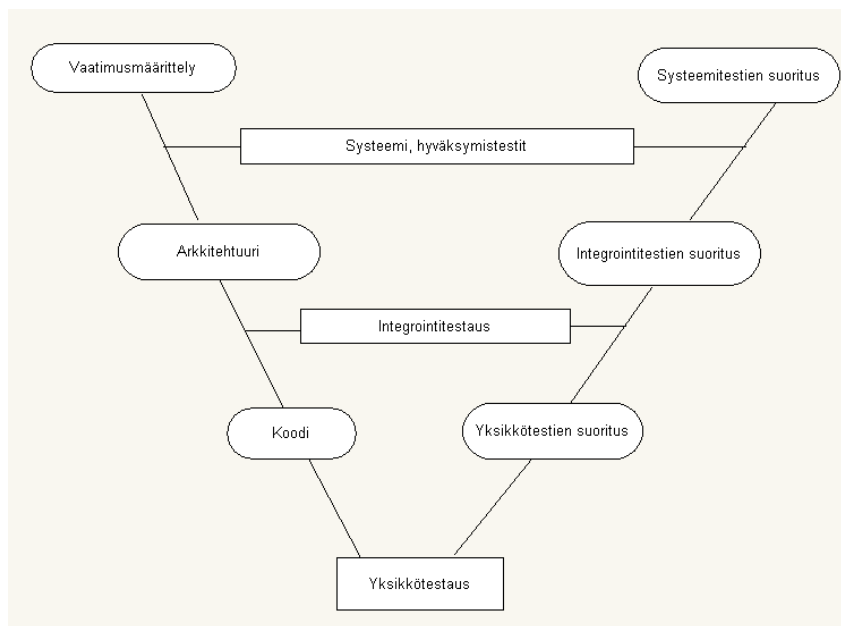
CAN-automaatioväylä (Controller Area Network) on Sovereign-hammashoitokoneen toimilaitteiden välinen väyläratkaisu (kuva 3). Koska väyläratkaisu (toimilaittekorttien liitäntä ACCU-ohjainkottiin) on keskeinen hoitokoneessa, hoitokoneen CAN-väylän toimivuus tulee testata. Näitä testejä ovat esimerkiksi hoitokoneen ja CAN-väylän kuormitusaste sekä reagointi väärin CAN-viesteihin. CAN-väylä on mutkaton tapa rakentaa liitäntä hammashoitokoneen ohjelmistotestausautomaatiolle. Hoitokoneen johdotusmäärä on pystytty minimoimaan, sillä väylä tarvitsee vain yhden johdinparin toimiakseen, lisäksi toimilaitte tarvitsee minimissään erillisen virransyötön. Näin on esimerkiksi OLCO-operointivalon tapauksessa. Tämä väyläratkaisu myös mahdollistaa muunneltavuuden, eli toimilaitteita, kuten operointivalo tai jalkaohjain, voidaan helposti liittää järjestelmään.

3 Ohjelmistotestaus

3.1 Ohjelmistotestauksen yleisiä periaatteita

Ohjelmistokehitystä voi kuvata V-mallilla (kuva 4), siitä on havaittavissa miten kehitys ja testaus ovat sidoksissa toisiinsa. Vasemmalla puolella kuvaa on esitetty ohjelmiston kehitys alkaen ylhäältä vaatimusten määrittelystä ja päättyen alas ohjelmointiin.

Oikealla puolella on esitetty testausvaiheita ohjelmistokehitysprosessin aikana. Kutakin kehitysvaihetta kuvassa vastaa testausvaihe. Poikittain kuvassa kulkevat suunnittelu ja toteutus. Periaatteena on, että jokainen testausvaihe vähentää virheitä seuraavista vaiheista.

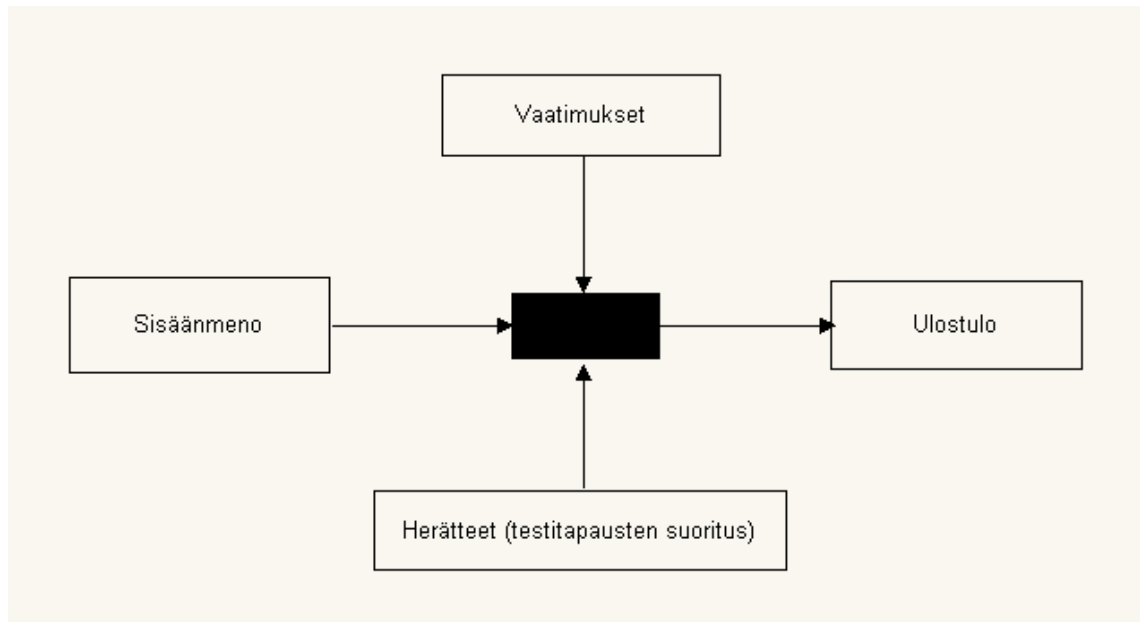


Kuva 4. V-malli [6]

Koska Sovereign-hammashoitokoneen ohjelmistolle asetetut vaatimukset ja vasteet ovat tunnettuja (tiedetään, miten ohjelmiston pitää käyttäytyä), voidaan ohjelmiston systeemitestaukseen soveltaa mustalaatikkoperiaatetta. Tämä toimintaperiaate eliminoi mahdolliset väärät painotukset testauksessa. Seuraavassa kappaleessa eritellään lyhyesti tämä periaate.

Mustalaatikkotestaus

Kuvassa 5 on esitetty mustalaatikkotestauksen periaate.



Kuva 5. Mustalaatikkotestaus [6]

Mustalaatikkotestauksessa ohjelmistoa testataan tuntematta ohjelmakoodia tai ohjelmakoodin rakennetta. Tunnetaan vain se, mitä ohjelman tulisi tehdä, ja testataan, toteutuuko tämä. Tätä tekniikkaa voidaan käyttää monessa eri testausvaiheessa. Mustalaatikkotestauksella saadaan siis ulkoapäin testattua ohjelmiston rajapinnat.

Valkolaatikkotestaus

Valkolaatikkotestaus on vaihtoehtoinen tapaa toteuttaa ohjelmistotestausta. Se perustuu ohjelmiston toiminnan yksityiskohtiin. Loogiset polut testataan herätteillä, jotka testaavat toimintaa eri haarautumis- ja ehtotilanteissa. Hoitokoneen ohjelmiston testauksessa valkolaatikkotestaus johtaisi mahdottomaan työmäärään eikä kattava testaus olisi lähimainkaan mahdollista.

Testisuunnitelma

Testisuunnitelma on olennainen osa ohjelmistotestausta riippumatta, käytetäänkö testauksessa musta- vai valkolaatikkotestausta. Testisuunnitelma sisältää tiedot siitä, mitä ja miksi testataan. Testisuunnitelmassa voidaan kertoa esimerkiksi yleiskatsaus, käytetyt tekniikat, testaustasot, käytetyt välineet ja testin suoritustavat.

Testitapaukset

Hammashoitokoneen testisuunnitelma koostuu yleiskatsauksen lisäksi testitapauksista, joissa kerrotaan miten testi suoritetaan. Testitapaukset vastaavat vaatimuksia vaatimusmäärittelyssä. Niillä varmistetaan, että ohjelmisto toimii vaatimusten mukaisesti. Ohjelmistolle asetettavia vaatimuksia ovat viranomaisvaatimukset, kuten turvaominaisuudet ja tuotevaatimukset. Tuotevaatimuksia ovat ohjelmistojulkaisusuunnitelmiin kirjatut suunnitellut ominaisuudet.

Ohjelmistotestaus on tärkeä osa hammashoitokoneen ohjelmistokehitystä, joka jatkuu koko ohjelmistotuotteen elinkaaren ajan. Se alkaa jo ohjelmointiprosessissa, jolloin ohjelmoijat testaavat omat käänöksensä. Ensimmäinen testi onkin se, että onnistuuko ohjelman kääntäminen. Hammashoitokoneen ohjelmistosta tehdään säännöllisin väliajoin julkaisuja, jotka käyvät läpi laajan systeemitason testauksen, jolloin myös uudet ominaisuudet testataan. Viikoittain kehitetään ja korjataan toimilaitteiden ohjelmia. Toimilaitteiden ohjelmista käännetään ohjelmistoversio, josta testaamalla etsitään virheitä. Tätä testausta kutsutaan integrointitestaukseksi, koska testattava ohjelmistopaketti on koostettu (integroitu) toimilaitteiden ohjelmista.

Testauksella varmistetaan ohjelmistovaatimusten toteutuminen, laatu ja asiakastyytyväisyys. Sovereign-hammashoitokoneen monimutkaisuuden vuoksi ohjelmistovikamahdollisuuksia on paljon. Toimilaitteiden välisessä kommunikoinnissa voi esiintyä vääriä viestejä, tai lääkärin instrumenttien parametreissa voi olla puutteita ja virheitä. Hammashoitokoneessa voi esiintyä myös laitevikoja. Näissäkin tapauksissa ohjelmiston tulisi hallita tilanne ja toimia turvallisesti. [6.]

3.2 Hoitokoneen ohjelmistovirheiden mahdollinen vaikutus

Hammashoitokoneen ohjelmistovirheet voivat teoriassa vaarantaa potilasturvallisuuden. Riskienhallinnassa on kuitenkin pyritty huomioimaan näitä tilanteita. Teoriassa on mahdollista, että potilastuolin ajokontrollia ei ohjelmistovian vuoksi kyetä hallitsemaan. Jos tässä tilanteessa myös potilastuolin liikkeen turvakytkin olisi rikki, siitä voisi seurata raajan puristuminen tuolin ja kiinteän kohteen väliin. Vaaratilanteita voisi syntyä ohjausjärjestelmän vioista. Esimerkiksi mikromoottori- tai turbiinipora voisi jäädä pyörimään tai vesisuihku instrumentista (spray) voisi loppua tai alkaa hallitsemattomasti. Mikromoottoriporan kierrosnopeus, vääntömomentti ja pyörimissuunta ovat ohjelmallisesti asetettavia. Jos ohjelmistovirheen seurauksena todelliset arvot ovat jotain muuta, kuin käyttäjä luulee, sillä voisi olla vahingollisia seurauksia poratessa hammasta. Vaaratilanne tai haitta voi muodostua hoitokoneen pesutoiminnon suorituksen jälkeen, jos virheellisen toiminnon seurauksena vesijärjestelmään jää desinfiointiainetta. Koska desinfiointiaine on haitallista ihmiselle, olisi tärkeää huuhdella järjestelmä hyvin pesun jälkeen.

Ohjelmistovirheet voivat myös hankaloittaa hammaslääkärin työskentelyä tai estää työskentelyn kokonaan. On haitallista, jos tuolin liikeradat eivät toimi oikein ohjelmistovirheen takia. Tästä voi aiheutua tilanne, jossa potilasta joudutaan hoitamaan hankalassa asennossa. Ohjelmistovirheet voivat aiheuttaa tulonmenetyksiä, jos potilaita ei kyetä hoitamaan. Näin voi käydä, jos pora tai jokin muu oleellinen osa hammashoitokonetta, kuten imujärjestelmä, ei toimi.

Imujärjestelmän on käynnistytävä automaattisesti, kun sitä tarvitaan hoito-operaatioissa. Jos imujärjestelmä pettää, kun imuletkut aktivoidaan, siitä seuraa tulva hoitajanyksikössä, koska hoitajanyksikössä käynnistyy huuhteluveden juoksutus imuletkujen aktivoinnin seurauksena. Tulvatilanteessa piirikortit ja muut sähköiset osat hoitajanyksikössä voivat kastua ja oikosulun seurauksena vahingoittua.

On mahdollista, että ohjelmistovirhe aiheuttaa käynnistyssilmukan. Se tarkoittaa sitä, että kun hammashoitokone on käynnistynyt, ohjelmistovirhe aiheuttaa laitteen uudelleenkäynnistymisen.

Koska Sovereign-hammashoitokoneessa useita jännitteitä ja virtoja säädetään ohjelmallisesti, saattaisi ohjelmistovirheen seurauksena elektroniikan komponenteille ohjautua liikaa tehoa. Tämä johtaisi elektroniikan käyttöiän lyhenemiseen tai välittömään tuhoutumiseen. Esimerkiksi viiden voltin jännitteitä pienennetään ohjelmisto-ohjatusti 32 voltista, jolloin ohjelmistovian seurauksena voisi viiden voltin elektroniikalle päätyä 32 voltia, josta aiheutuisi elektroniikan välitön tuhoutuminen. Laitteistossa on kuitenkin sisäänrakennettuna suojauksia, kuten älykkäissä moottoriohjaimissa, joiden suojaukset aktivoituvat, jos ohjelmisto yrittää toimia laitetta vahingoittavasti.

3.3 Ohjelmiston testaus manuaalisesti

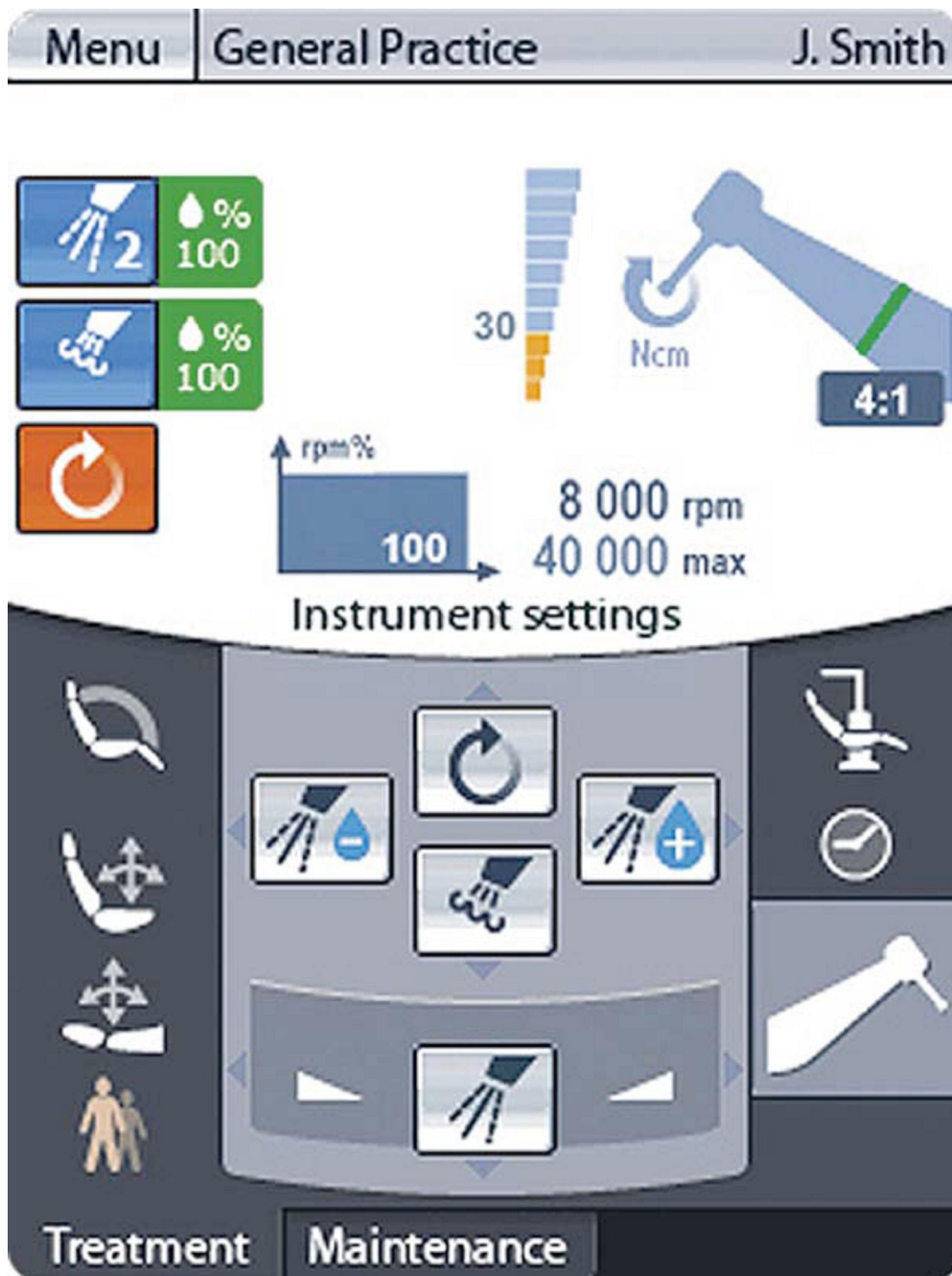
Manuaalinen testaus suoritetaan systeemitasolla, eli koko hammashoitokonejärjestelmä ohjelmistoinen testataan kokonaisuutena. Manuaalisessa testauksessa hammashoitokoneen toimintoja käytetään ”käsini” hammashoitokoneen tuotekehityksessä kehitettyjen testien määrittämällä tavalla. Nämä testitapaukset on suunniteltu ennen suoritusvaihetta, ja ne perustuvat ohjelmistolle asetettuihin järjestelmävaatimuksiin, eli jokaista vaatimusta vastaa yksi testitapaus.

Testitapaukset käsittävät seuraavia alueita:

- instrumentit
- motorisoidun potilastuolin ajot
- turvallisuusominaisuudet
- ylläpidettävyys (päivitettävyys ja kalibroinnit)
- valmistuksen kannalta oleelliset ominaisuudet
- ohjelmistoasennus
- laiteviat ja käyttäjän tekemät virheet.

Manuaalisessa testauksessa toimintoja ohjataan hammashoitokoneen instrumenttikonsoliin kiinnitetyn graafisen kosketusnäytön ja jalkaohjaimen avulla. Kuvassa 6 on manuaalisen testauksen kannalta tärkeä instrumentin ohjausnäkyvä. Tästä näkymästä hammaslääkäri asettaa oikeat asetusarvot mikromoottoriporalle. Ohjelmistotestauksella varmistetaan asetusarvojen oikeellisuus ja toiminta.

Manuaalisessa testauksessa tarvitaan myös ulkopuolista tietokonetta, johon soveltuu Linux-käyttöjärjestelmällä varustettu kannettava tietokone tai Telnet-ohjelmalla varustettu Windows-tietokone. Testauksessa tarvitaan myös ulkoisia mittareita, kuten yleismittari (jännitteet) ja painemittarit (instrumenttien ajoilman ja vesien paineiden tarkistus). Mitta-astioilla voidaan todentaa instrumenttien tuottamien säädettävien vesimäärien tilavuus. Ulkoisilla painoilla voidaan todentaa potilastuolin nostokyky. Näiden mittaustulosten hyväksymisrajat on määritelty testitapauksissa.



Kuva 6. Mikromootorin graafinen ohjausnäky (GUI)

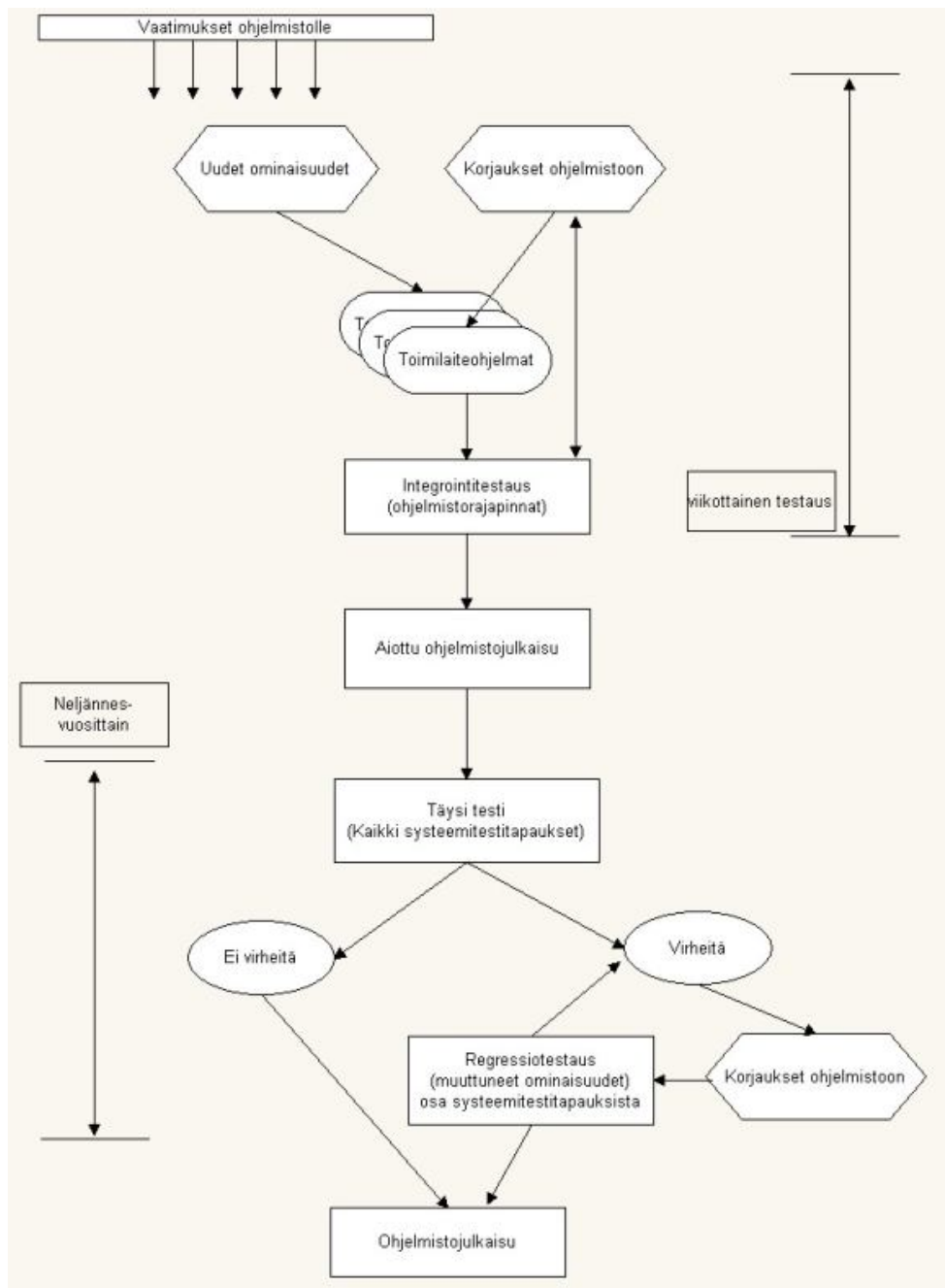
Testauksen apuna tarvitaan ulkoinen tietokone, jolla voidaan lähettää niitä komentoja, joita ei voi lähettää graafisen kosketusnäytön avulla. Hammashoitokone kalibroidaan käyttäen ulkoista tietokonetta ja konsoliyhteyttä. Tietokone voi olla liitettynä hammashoitokoneeseen joko sarjaväylän tai LAN-lähiverkon kautta.

Jos ohjelmapaketissa on tuotu laitteeseen uusia ominaisuuksia, tarvitaan yleensä täysi testauskierros ja vastaavasti, jos kyse on ohjelmavirhekorjauksesta, riittää usein regressiotestauskierros. Systeemitestauksella pyritään varmistamaan tuotemäärittely. Regressiotestistä ja täydestä testistä täytetään testiraportti.

Testikierrokset

Integrintitestauksella tarkoitetaan viikoittain tapahtuvaa vapaavalintaista ohjelmiston väliversioiden testausta. Ohjelmiston väliversiot koostuvat hammashoitokoneen toimilaitteiden itsenäisistä ohjelmista, jotka koostetaan (integroidaan) yhdeksi ohjelmistopaketsiksi, joka ladataan hammashoitokoneeseen. Integrintitestaus perustuu säikeittäin testattaviin systeemin ominaisuuksiin. Integrintitestauksessa testataan ohjelmamoduulien väliset rajapinnat. Integrintitestauksessa todetaan myös edellisen viikon aikana tehtyjen korjausten toimivuus. Regressiotesti koostuu toiminnallisista perustesteistä ja tarveharkinnan mukaisista lisätesteistä. Regressiotesti on uudelleentestausta, jossa on tarkoitus kattaa ominaisuuksia, jotka ovat muuttuneet uudessa ohjelmistojulkaisussa. Regressiotestin laajuudesta keskustellaan ohjelmistosuunnittelijoiden kanssa. Kuvasta 7 voidaan havaita, miten testikierrokset aikataulutetaan aiottuun ohjelmistojulkaisuun nähden.

Hammashoitokoneen uusi ohjelmisto pyritään julkaisemaan neljännesvuosittain, ja se sisältää uudet ominaisuudet ja korjaukset (kuva 7). Tyypillisesti täyteen testausjaksoon varataan noin kuukausi aikaa.



Kuva 7. Ohjelmistotestausprosessi

Täysi testi sisältää kaikki manuaalisesti suoritettavat systeemitestitapaukset, joihin siis lukeutuvat sekä integrointi- että regressiotestitapaukset. Nämä testit tehdään kahdessa erilaisessa hoitolaitekokoonpanossa, joita ovat tuotteet nimeltä Sovereign ja Sovereign-Classy. Manuaalisesti suoritettavia testitapauksia on 134.

4 Testiautomaation kehitystarve

Hammashoitokoneella ohjataan instrumentteja, joilla voidaan hoidon aikana ”työstää” potilaan hampaita. On ensiarvoisen tärkeää, että potilas ja lääkäri voivat luottaa näiden instrumenttien toimintaan. Luottamuksen lisäämisessä on ohjelmistotestauksella tärkeä rooli. Ohjelmistotestuksen automatisoinnilla voidaan saavuttaa merkittäviä etuja. Automaatioon sisään rakennetut anturit voivat tehdä tarkempia mittauksia kuin manuaalisesti suoritettuja, ja halutuista ominaisuuksista on mahdollista muodostaa raportti, joka myös tallentuu automaattisesti. Ohjelmistotestausautomaatiolaitteisto voi systemaattisesti testata hammashoitokoneen ominaisuuksia ja toimintoja ja täten parantaa tetauksen kattavuutta.

Ohjelmistotestausautomaatio voi keventää testaajan työkuormaa, koska hammashoitokoneen manuaalinen testaus on työlästä, aikaa vievää eikä sillä voi kattaa kaikkia hoitokoneessa ohjelmallisesti valittavia muuttujia kaikilla mahdollisilla arvoilla. Lisäksi laitekonfiguraatioiden ja lääkärin instrumenttien lisääntyessä testaustarpeet kasvavat moninkertaisiksi. Tällöin myös virheiden määrä ja mahdollisuus siihen, että virheet jäävät huomaamatta, kasvaa.

Taulukosta 1 voidaan havaita Bienair MX -mikromoottorin ominaisuuksien suuri määrä. Hammashoitotyöskentelyssä näistä voidaan valita mikä tahansa yhdistelmä. Yhdistelmien suuren määrän vuoksi täydellistä yhdistelmätestausta manuaalisessa testauksessa ei kannata edes yrittää. Testiautomaatio on väsymätön käymään läpi erilaisia parametriyhdistelmiä. Tällöin havaitaan paremmin viat, jotka tulevat esiin vain tietyt ehdot täyttävässä tilanteessa. Taulukossa 1 sekä seuraavassa kappaleessa tuodaan esiin, miten yhdistelmien suuri määrä muodostuu. Taulukossa 1 ominaisuudella tarkoitetaan toimintoa, jonka suoritustapaan voidaan vaikuttaa muuttujan arvolla. Suluissa esitetty ”toisistaan riippumattomat” tarkoittaa, että kaikki arvot ovat vapaavalintaisia (vapaavalintaisten yhdistelmien määrä on suuri). Sarakkeessa arvoja on kyseisen toiminnon rivillä mahdollisten muuttuja-arvojen suurin lukumäärä.

Taulukko 1. Mikromoottori Bienair MX:n parametrit

Ominaisuus (toisistaan riippumattomat)	muuttuja-arvo	arvoja
sprayvesimoodi	1, 2, pois	3
sprayvesimäärä	0–100 %, säätöaskel 5 %	21
pyörimissuunnan valinta	peruutus, pois	2
automaattinen lastun puhallus	päällä, pois	2
manuaalinen lastun puhallus	poljintoiminto	2
kuituvalo päälle/pois -valinta	päällä, pois	2
vääntömomentin valinta	A, B, C, D, off	5
vääntömomentti	10–90 %, säätöaskel 5 %	19
vääntömomenttitoiminto	automaattinen eteneminen, automaattinen peruutus	2

Lääkärin konsolissa on kuusi instrumenttipaikkaa, joista jokaisella on omat parametrinsa. Kuvassa 8 on kuusi instrumenttia instrumenttipaikoissaan, vasemmalta oikealle lueteltuina ruisku, paikan valokovetin, hammaskiven poistolaite, suukamera, mikromoottoripora ja turbiinipora.



Kuva 8. Hammaslääkärin instrumentteja [10]

Samaa instrumenttityyppiä, kuten mikromoottori-poria tai hammaskiven poistolaitteita, voi olla useampikin yhdessä konsolissa, ja niitä voi vaihtaa eri paikkoihin sammuttamatta sähköjä ottaen huomioon tietyt laitepohjaiset poikkeukset. Taulukon 1 mikromoottori Bienair MX:n parametrien lukumäärään perustuen on laskettu mahdollisten yhdistelmien määräksi $3 \times 21 \times 2 \times 2 \times 2 \times 2 \times 5 \times 19 \times 2 = 191520$. Jos instrumenttikonsoli on asennettu täyteen (kuusi instrumenttia) mikromoottoreita (poria), testattavia kombinaatioita tulisi **49349731983342298007076864 miljoonaa**. (Kun yhden instrumentin parametrien lukumäärä on 191520 siitä seuraa, että kuuden instrumentin yhdistelmiä on $191520 \times 191520 \times 191520 \times 191520 \times 191520 \times 191520 = 49349731983342298007076864$ miljoonaa).

Hoitajan konsolissa voi olla viisi hoitajan laitetta (ei poria). Edellinen laskelma on vain yksi esimerkki suuresta testauskuormasta, jonka suuri testattavien yhdistelmien määrä aiheuttaa. Muilla instrumenteilla on vastaavasti omat parametrinsa.

Jos testausautomaatiolaitteisto on hyvin suunniteltu ja toteutettu, sillä voi suorittaa isomman otannan tai kaikki testit tarvittavilla laitekoonpanoilla. Näin voidaan tehokkaasti havaita mahdolliset ohjelmistovirheet ja muodostaa niistä tarvittaessa halutun kaltainen kattava raportti. Testausautomaatiolaitteisto voi myös tehdä työtä kellon ympäri, eikä se vaadi palkkaa. Voidaan todeta, että Sovereignhammas-hoitokoneen ohjelmistotestausautomaatio on tarpeellinen ja perusteltu taloudellisista ja teknisistä syistä.

Testausautomaation sovellusalueita

Ohjelmistotestauksen automatisointi voi kattaa sovellusalueita, kuten yhdistelmätestaus, suorituskyykytestaus ja luotettavuustestaus. Yhdistelmätestaus kohdistuisi lähinnä instrumenttikontrolliin ja instrumentteihin, koska instrumenttien asetusarvojen yhdistelmiä on lukematon määrä (esimerkkinä taulukko 1). Suorituskyykytestaus soveltuisi automatisoitavaksi hyvin, koska testiautomaatiolaitteisto voisi mitata jännitteitä, kierrosnopeuksia, vääntömomenteja ja lämpötiloja testaajaa tehokkaammin. Tämä on mahdollista toteuttaa testipenkkiin rakennettavilla antureilla. Toipumistestauksessa testipenkki voi automaattisesti suorittaa hammashoitokoneen prosessien pysäytystoimintoja (hoitokoneen käynnistyminen ja hätä-katkaisija-toiminto) sekä analysoida laitteen toipumista näistä tilanteista.

Luotettavuustestauksessa voidaan toistaa haluttua toimintoa (käynnistyminen, aktivointi tai vastaava) haluttu toistomäärä tai niin kauan kunnes havaitaan virhe. Tämänkaltaisella testauksella voidaan parantaa hammashoitokoneen luotettavuutta. Manuaalisia testitapauksia ei ole ryhmitelty edellä mainittujen sovellusalueiden mukaan, koska tähän ei ole ollut tarvetta.

4.1 Testiautomaation kustannukset

Ylivoimaisesti suurin osa testiautomaation kehityskustannuksista muodostuu työkustannuksista, koska aikaa vievät testien määrittely ja suunnittelu. Testiautomaation valmistelu alkaa vaatimusmäärittelystä, jolloin päätetään, miten automaatiolaitteisto

sekä testiohjelmisto toteutetaan. Testiautomaation vaatimusmäärittelyyn osallistuu useita henkilöitä, mikä vie paljon työaika palaverineen. Kun alustava testaussuunnitelma on tehty, alkaa varsinainen suunnitteluvaihe testiautomaation testitapausten luomiseksi. Kaikki määritelty ja suunniteltu täytyy dokumentoida, mikä aiheuttaa oman työkuormansa. Prototyypit maksavat. Kun ensimmäiset prototyypit on valmistettu, ohjelmoidaan testit, testataan ja dokumentoidaan laitteisto. Elektroniikkasuunnittelun työmäärää on vaikea arvioida. Ensimmäisen version aikaansaamiseen voisi kulua yksi vuosi.

Lisenssit ovat välttämätön kuluerä hankittaessa testiohjelmistoa. Testitapausten ohjelmointi vaatii koulutuksen valittuun ohjelmointiympäristöön ja myös koulutuksen hammashoitokoneen ohjelmiston testaukseen. Automaatiotestitapausten ylläpitokustannukset riippuvat paljon siitä, miten hyvin testiohjelmiston koodi on dokumentoitu ja miten selkeästi testit on ohjelmoitu.

Ohjelmistotestausautomaation hyödyt ovat yleensä saavutettavissa vasta myöhäisemmässä vaiheessa, kun suurin osa testiautomaatiolaitteiston ja testiautomaatio-ohjelmiston kehitystyöstä on tehty. Taloudellinen hyöty alkaa näkyä, kun suoritetaan testausta automaattisesti ilman testaajan jatkuvaa läsnäoloa. Testaus on osa ohjelmistojen jatkuvaa kehitystyötä, ja ohjelmistotestausautomaatio on ohjelmistotestauksen todennäköinen kehityssuunta.

4.2 Yhteenveto testiautomaatiotarpeesta

Testaus varmistaa ohjelmiston luotettavuuden ja läpi päässeet virheet, voivat tulla kalliiksi ja olla jopa potilasturvallisuutta uhkaavia. Potilasturvallisuus edellyttää, että hammashoitokoneen ohjelmisto on kattavasti testattu. Hammashoitokoneen ohjelmisto on niin monimutkainen kokonaisuus, että manuaalisella testauksella ei voida taloudellisella tavalla kattaa riittävästi eri tilanteita (eri herätteiden yhdistelmiä), joita ohjelmisto joutuu käsittelemään. Testiautomaatio ratkaisee tämän ongelman, koska automaatiolaitteisto voidaan ohjelmoida suorittamaan järjestelmällisesti eri käyttötapauksia kaikilla mahdollisilla parametreilla, kuten esimerkiksi instrumenttien

käyttämät vesi ja ilmamäärät. Testiautomaatio mahdollistaa myös ohjelmiston luotettavuustestauksen toistamalla testausta kattavan tilaston luomiseksi. Samalla toistomenetelmällä saadaan esille vain harvoin toistuvat ohjelmistoviat. Testiautomaatio voi siis tehostaa testausta, vähentää läpi päässeistä virheistä aiheutuvia kustannuksia ja myös testauskustannuksia.

5 Testiautomaation kehityssuunnan arviointia

Testausautomaatioprojektin mahdollisia vaiheita ovat seuraavat:

- välineiden ja ohjelmistojen valinta
- henkilöiden kouluttaminen
- automaation tason päättäminen
- laitteiston rakentaminen
- ohjelmien hankinta ja asennus
- automatisointitapausten valinta
- scriptien luonti
- testituloksen raportointi
- kokonaisuuden testaus ja analysointi.

5.1 Ohjelmistotestitapausten toteutustapa

Kehitystyössä kannattaisi huomioida, mitkä automaattiset testitapaukset suunniteltaisiin ensin ja mikä on niiden työmääräarvio. Kehitystyössä kannattaisi myös huomioida, miten testitapaukset on ajateltu suoritettavaksi. Testisuorituksen etenemiselle pitäisi tarvittaessa voida asettaa hyväksytyt/hylätty-ehtoja, sillä ei ole järkevää siirtyä testilistassa eteenpäin, jos jokin tärkeä osa-alue on edellisessä testissä todettu hylätyksi, esimerkiksi käyttöjännite. On myös huomioitava, miten testiautomaation kehitystuki on järjestetty. Tuen tulee sisältää ylläpitoa ja pitkäaikaisen kehityksen, joka on tarpeellinen, jos ohjelmistotukea laajennetaan käsittämään uusia instrumentteja tai ominaisuuksia. Seuraavassa otetaan kantaa toteutukseen. Koska tämän työn tarkoituksena ei ole laitteistosuunnittelu, on laitteistoja käsitelty vain toteutuksen kannalta.

Puoliautomaattinen testilaitteisto

Puoliautomaattinen laitteisto on helppo ja kustannuksiltaan edullinen toteuttaa. Puoliautomaattisella laitteistolla suoritettavat testit kuitenkin vaativat paljon testaajan huomiota, koska instrumentteja on tarkkailtava suorituksen aikana ja arvioitava, toimivatko ne hyväksytyllä tavalla. Puoliautomaattinen laitteisto vaatii testaajan huomiota myös silloin kun testattava instrumentti on siirrettävä ”käsini” toiseen instrumentin pikaliitinpaikkaan.

Täysautomaattinen testilaitteisto

Mikäli halutaan ohjelmistotestauksen täysautomaattinen toteutus, tarvitaan automaatiolaitteisto, joka pystyy suorittamaan kaikki tarvittavat testaustoimenpiteet (instrumenttien aktivoinnit ja ohjaukset) automaattisesti ja mittaamaan lääkärin ja hoitajan instrumenttien toimintaa fysikaalisina suureina, kuten seuraavasta instrumenttikohtaisesta ominaisuustaulukosta 2 ilmenee. Automaatiolaitteiston on testattava mikromoottoriporan ja turbiiniporan terän pyörimisnopeus, moottorin vääntömomentti (vain mikromoottori), moottorin pyörimissuunnan vaihdon ja kuituvalon ohjauksen toimivuus. Koska porat voidaan asettaa ajon aikana suihkuttamaan vettä halutulla voimakkuudella (veden tuotto) sekä puhaltamaan ilmaa, on nämä toiminnot myös testattava. Hammaskiven poistolaitteen tärkeimmät mitattavat suureet ovat ultraääniteho ja veden suihkutus. Valokovettimen automaatiotestiksi riittää ohjauksen toimivuuden tarkistus.

Taulukko 2. Instrumenttien ominaisuudet

Mikromoottori	Suure
pyörimisnopeus	rpm
pyörimissuunta	CW/CCW
vääntömomentti	Ncm
veden tuotto eri tilanteissa	ml
ilman tuotto eri tilanteissa	mbar
mikromoottorin kuituvalo	lux
Hammaskiven poistolaite	
veden tuotto eri tilanteissa	ml
ultraääniteho	W
kuituvalo	lux
Turbiini	
pyörimisnopeus	rpm
veden tuotto eri tilanteissa	ml
turbiinin kuituvalo	lux
ilman tuotto eri tilanteissa	ml
Valokovetin	
valo	lux

Täysautomaattisen laitteiston tulee pystyä tuottamaan testitulokset, josta käy ilmi, onko testi hyväksytty vai hylätty, ja tallentamaan virheloki. Täysautomaattinen laitteisto on kaikkein vaativin ja kallein toteuttaa, koska se vaatii paljon antureita fyysikaalisten ominaisuuksien lukemiseen instrumenteilta ja muualta järjestelmästä. Laitteiston tulee voida lukea veden ja ilman painetta sekä mitata jännitteitä halutuista kohteista. Tietoja voi kerätä PC:hen liitettävällä tiedonkeruunmoduulilla, joka sisältää riittävästi analogisia sekä digitaalisia tuloja ja lähtöjä.

Hankalinta on tehdä järjestelmä, joka sisältää venttiilit, joilla ohjataan veden ja ilman kulkua järjestelmässä. Toteutusta helpottaa, jos voidaan osittain tukeutua emuloituihin toimintoihin, koska silloin voidaan tehdä toteutus ohjelmallisesti (tietokoneella) ilman tarvetta kalliiseen elektroniikkasuunnitteluun. Näitä voivat olla kosketusnäyttö- ja

instrumenttiemulaatio, instrumentinpitimien anturien, instrumenttitunnisteiden ja jalkaohjaimen emulaatio sekä CAN-simuloitu moduuli.

Ohjelmistotestausautomaatiolaitteiston on pystyttävä seuraamaan käynnistyssekvenssiä ja veden virtausta, keräämään anturidataa CAN-viesteillä ja lukemaan korttitunnisteita. Toteutuksessa on otettava huomioon, että testattavan hoitokoneohjelmiston kannalta testilaitteiston tulee näyttää hoitokoneelta, muuten testitulokset eivät ole välttämättä sovellettavissa. Tämä on mahdollista kytkemällä haaroituskappale instrumenttien ilma- ja vesilinjan väliin. Näin painetieto voidaan tuoda kyseessä olevasta ominaisuudesta antureille.

Kun automaatiolaitteisto on valmis, päätetään, miten testitapaukset automatisoidaan. Luultavasti tässäkin täysiautomaattisen laitteiston tapauksessa kannattaa toteuttaa asiat vaiheittain. Täysin automaattista testausta ei voida ottaa käyttöön täysimittaisesti, ennen kuin testauslaitteisto on testattu ja todettu riittävän luotettavaksi.

5.2 Testiautomaatiovertailu

Taulukossa 3 on analysoitu pääpiirteittäin puoliautomaattisen ja täysiautomaattisen laitteiston vahvuuksia ja heikkouksia ohjelmistotestauksen automatisoinnissa.

Taulukko 3. Testilaitteistojen vahvuudet ja heikkoudet

Puoliautomaattinen laitteisto	
Vahvuudet	Heikkoudet
Toteutuksen voi aloittaa asteittain, ominaisuus kerrallaan.	Testien suoritus vaatii testaajan läsnäoloa.
Ei vaadi heti isoja investointeja.	Jos toteutetaan asteittain, kokonaisuus voi rönnyillä.
Voi joskus saavuttaa täysautomaatioasteen.	
Voidaan toteuttaa hoitokoneeseen liitettävillä moduuleilla, jotka suorittavat tarvittavat mittaukset ja ohjaukset.	
Täysautomaattinen laitteisto	
Vahvuudet	Heikkoudet
Kattaa koko systeemitestauksen.	Vaatii ison investoinnin.
Testien suoritus ei vaadi testaajan läsnäoloa.	Toteutus perustuisi hoitokoneeseen, johon olisi lisätty tarvittava elektroniikka ja sähkömekaniikka.
Kerralla suunniteltaessa kokonaisuuden suunnittelu olisi hyvin hallinnassa.	

Puoliautomaattinen laitteisto on mielestäni järkevin vaihtoehto toteuttaa, koska toteutuksen voi aloittaa asteittain ja tehdä testejä osa-alueittain, jolloin voidaan heti todeta automaation toimivuus ja osoittaa automaatiosta saatava hyöty. Kumpikin toteutustapa vaatii myös laitteistosuunnittelua, mutta siihen ei tässä yhteydessä puututa.

Graafinen ohjelmointiympäristö

Toinen puoli testiympäristöä on testausohjelmisto. Lähtökohtana testausohjelmiston valinnassa on testitapausten ohjelmoinnin helppous ja monipuolisuus. On siis aika selvää, että merkkipohjaiset ohjelmointikielet eivät tule kyseeseen. Graafisista

testausohjelmistoista National Instrumentsin *LabVIEW* on hyvä vaihtoehto testausohjelmistoksi helpon käytettävyyden ja laajan laitetukensa vuoksi. *LabVIEW-ohjelmistoon* on saatavilla National instrumentsin valmistama CAN-liitäntämoduuli, mutta se tukee myös muiden valmistajien vastaavia CAN-moduuleja. *LabVIEW-ohjelmisto* toimii molemmissa tapauksissa, sekä täysautomaattisessa että puoliautomaattisessa testilaitteistossa.

LabVIEW-ohjelmien rakenne

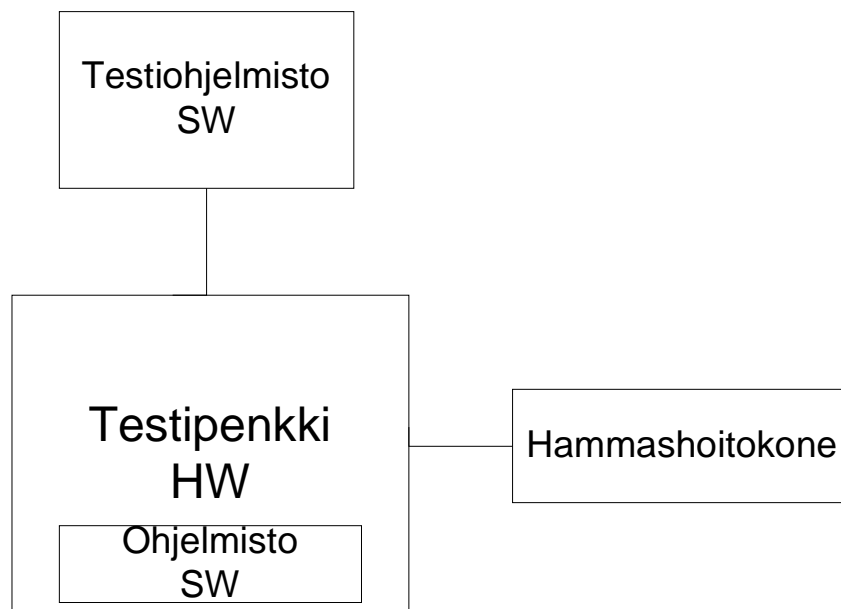
LabVIEW-ohjelmistossa yksittäinen ohjelmamoduuli on nimeltään virtuaali-instrumentti eli *VI*. Käytännössä ohjelmamoduulien nimen laajenteena on tuo sama *VI*, eli ohjelma voi olla nimeltään vaikka *Mittari.VI*. *VI* muodostuu kahdesta eri osasta, kontrollipaneelistä (Front Panel), joka on käyttöliittymä, ja kaaviosta (Diagram), joka on oikeastaan varsinainen ohjelma eli ohjelman looginen toiminta.

Laajemmin tarkasteltuna huomataan, että *VI* on aliohjelma. *VI:lle* voidaan määritellä tulot ja lähdöt kytkentäpisteillä (ns. konnektoreilla), jotka vastaavat "tavallisen" aliohjelman kutsuparametreja. Näin ollen *LabVIEW* tukee modulaarista ja hierarkkista ohjelmointia. [7.]

Virtuaali-instrumentit

LabVIEW-kaavion yksinkertaistamisessa on tärkeää hahmottaa, mitkä osat voivat toimia aliohjelmina. Sekä testipenkin ohjelmisto että testiohjelmisto on järkevää toteuttaa *LabVIEW-virtuaali-instrumentteina*, vaikka kuvassa 9 ne näkyvät eri ohjelmistolohkoina. Testipenkkiosuudessa virtuaali-instrumentit voi tehdä toiminnoittain, esimerkiksi *mikromoottori.VI* tai *GUI.VI*. Tämä mahdollistaa myös paremman version hallittavuuden testausohjelmiston suhteen, sillä matalamman tason virtuaali-instrumentit eli *VI:t*, jotka muodostavat ohjelmistorajapinnan laiteohjaimiin, ovat versiohallittuja. Versionhallinnalla tarkoitetaan Planmecan tuotekehityksen järjestelmää, johon julkaistut ohjelmistot tallennetaan.

Testitapaukset, joita luodaan ohjelmistotestausautomaatiota varten, eivät tule olemaan aivan samoja kuin manuaalisessa testauksessa. Tarvitaan esimerkiksi automaattista raportointia, jotta jälkeempään on mahdollista todeta, miten lähellä hyväksymisrajaa laitteen käyttämät fyysiset suureet ovat olleet testiajon aikana. Näytön tapahtumien seuranta ei ehkä pystytä toteuttamaan kuin CAN-viestien tasolla. Täydellinen seuranta mahdollistuu vain konenäköjärjestelmällä. Pitää olla myös mahdollista ajaa testisarjoja tai toistaa jotain tiettyä testiä halutun ajan. Testitapaukset rakentuvat testipenkin toiminnallisista virtuaali-instrumenteista (kuten *mikromoottori.VI*) sekä niiden ympärille rakennetusta logiikasta, käyttöliittymästä ja raportoinnista.



Kuva 9. Testipenkki ohjelmistoinen

5.3 Testilaitteiston käyttöliittymä

Sovereign-hammashoitokoneen toimintoja ohjaa ACCU-ohjainkortti. Lähettämällä CAN-komentoja ACCU-ohjainkortille saadaan ohjelmistotestauksen kontrolli toteutettua systeemitasolla. Toinen mahdollisuus ohjauksen toteutukseen on käyttää kosketusnäytön *button press*- ja *button click* -viestejä, joita ACCU-ohjainkortti

tulkitsee. Tämä ohjaustapa on hammashoitokoneen ohjelmiston kannalta lähellä käyttäjän toimintaa kosketusnäytön kanssa.

Voi olla järkevää komentaa hammashoitokonetta button-viesteillä (CAN) ja lukea palautetietoja CAN-väylältä. Button-viestit ovat kolminumeroisia lukuja, esimerkiksi 341 liikuttaa potilastuolia ylöspäin. 341 on tyyppiä `button_press`, jonka painallus on voimassa, kunnes testiohjelmassa seuraa `button_release 341` (release-viesti vapauttaa numerolla määritellyn napin), vaikutusaika määritellään viiveellä (`sleep`).

Kolmas tapa välittää komentoja hoitokoneelle on simuloida kosketusnäytön kommunikaatiota, eli lähettää x- ja y-koordinaattitietoja.

Testipenkin GUI-kosketuspaneelisimulaatio voidaan toteuttaa *LabVIEW-ohjelmistolla* ohjattavan I/O-moduulin (liitäntämoduuli) tuottamilla jännitteillä, jotka vastaavat oikean kosketuspaneelin painallusten tuottamia jännitteitä. Kun PC:n näytöllä painetaan nappia, lähetetään painallusta vastaava jännite I/O-moduulin kautta testipenkille, joka prosessoi sitä eteenpäin. Ohjelmistotestitapaukset voidaan muodostaa käyttäen runkona testipenkin virtuaali-instrumentteja, kuten kosketusnäyttöä, jakajaa (multiplexer), jalkaohjainta, turvakytkintä, CAN-viestiä ja raportointia. Tämän lisäksi tarvitaan mahdollisuutta valita testikokonaisuuksia suoritettavaksi sekä raportointiominaisuuksia.

5.4 Automaatio

Testitapausten runsaan määrän (yli sata) vuoksi niiden graafiset kontrollipainikkeet eivät mahdu yhteen kontrollipaneeliin, vaan ne kannattaa jakaa useampaan kontrollipaneeliin, jotka voidaan jaotella moduuleittain. National instruments -yhtiöltä on saatavilla myös Test Stand -niminen sovellus, jolla voidaan helposti rakentaa ”runko” LabVIEW-testeille. Tarkemmin sanottuna Test Standilla voidaan määrittää käytetyt virtuaali-instrumentit, mahdolliset toistokerrat, muut ehdot ja hyväksymisrajat sekä tulostaa raportti. Edellä mainitut asiat on mahdollista toteuttaa pelkällä LabVIEWillä, mutta se on hankalampaa. Edellä mainituista syistä Test Standin valintaa LabVIEW-testitoteutuksien automatisointiin kannattaa harkita.

6 Päätelmät

Insinööriyön lopputuloksena syntyi arvio Sovereign-hammashoitokoneen ohjelmistotestauksen testiautomaatiotarpeesta sekä toteutusmahdollisuuksien vertailu perustuen puoliautomaatiseen tai täysautomaattiseen testiautomaatiolaitteistoon ja *LabVIEW-ohjelmistoon*.

Automaatioprojektin esivalmistelut osoittivat testiautomaation rakentamisen pitkäjänteiseksi prosessiksi. Testiautomaation kehitys jäi tämän työn puitteissa vain alustukseksi tälle prosessille, ja aika näyttää, miten paljon siitä saadaan hyötyä tulevaisuudessa. Insinööriyötä tehdessä kävi ilmi, että hammashoitokoneen ohjelmistotestausautomaation suunnittelu ja toteutus tulee vaatimaan paljon aikaa ja resursseja. Työssä osoitetaan, että olisi hyödyllistä muuttaa manuaalinen ohjelmistotestaustyö automatisoiduksi. Työ antaa myös perustietoa systeemitason ohjelmistotestauksesta Planmeca Oy:n hammashoitokoneen tuotekehityksessä.

Testiautomaation kehitystyötä jatkavat elektroniikka- ja ohjelmistosuunnittelijat sekä testaajat. Testiautomaatiolaitteisto ohjelmistoinen tulee dokumentoida versionhallintajärjestelmään ja mahdollisesti myös kalibroida, että sen tuottamat testitulokset voivat olla hyväksytyjä. Mahdollisia riskejä ovat testiautomaatiolaitteiston muodostuminen liian monimutkaiseksi tai hoitokoneohjelmiston ja automaatio-ohjelmiston yhteensopimattomuus. Hyvin toteutettuna Sovereign-hammashoitokoneen ohjelmistotestausautomaatio saavuttaa tavoitteen: suorittaa kattavaa testausta nopeasti ja taloudellisesti.

Lähteet

- 1 Planmeca groupin yritykset. Planmeca Oy. (www-dokumentti.) Planmeca. www.planmeca.com/index.php?lng=11&page=63137, Luettu 15.6.2009.
- 2 Planmeca Sovereign. (www-dokumentti.) Planmeca_ <http://www.planmeca.com/index.php?lng=11&page=63137>. Luettu 18.6.2009.
- 3 Järvinen Juha. NGU. Planmeca NGU project, (Planmeca Oy:n intranet.) 2009. Luettu 26.6.2009.
- 4 Finnilä Panu. ACCU Master Object. (Planmeca Oy:n intranet.) 2009. Luettu 26.6.2009.
- 5 Finnilä Panu, Haapajärvi Teemu, Korhola Antti, Karppanen Petri, Haapajärvi Teemu, Tarvainen Mika ja Matosevic Mladen . NGU Operating Device Configuration, Planmeca, (Planmeca Oy:n intranet.) 2009. Luettu 17.6.2009.
- 6 Burnstein Ilene, Practical Software Testing. Department of Computer Science Illinois Institute of Technology, New York: Springer-Verlag, 2003.
- 7 LabVIEW ohjelmien rakenne. (www-dokumentti.) Jyväskylän yliopisto www.mit.jyu.fi/opiskelu/seminaarit/ohjelmistotekniikka/LabVIEW/#2.%20Johdanto. Luettu 12.8.2009.
- 8 Lääkäriin instrumentteja (www-dokumentti.) Planmeca. http://www.planmeca.com/image_gallery/low_res/?dir=%2F01_Planmeca_Sovereign. Luettu 12.8.2009.