



Arto Tiitto

WAVELETS AND MAXIMUM STRENGTH TECHNOLOGY ALGORITHM

WAVELETS AND MAXIMUM STRENGTH
TECHNOLOGY ALGORITHM

Arto Tiitto
Bachelor's Thesis
25.5.2011
Degree Programme in Medical
Engineering
Oulu University of Applied Sciences

Koulutusohjelma	Opinnäytetyö	Sivuja	+	Liitteitä
Hyvinvointiteknologia	Opinnäytetyö	65	+	1
Suuntautumisvaihtoehto	Aika			
Avopalvelun teknologia	2011			
Työn tilaaja	Työn tekijä			
HYTKE	Arto Tiitto			
Työn nimi				
Aallokkeet ja Maksivoimateknologian algoritmi				
Avainsanat				
Algoritmi, Chronos, maksimivoima, analyysi, kehitys, aallokkeet				

Työn tarkoituksena oli parantaa Oulun seudun ammattikorkeakoulun Tekniikan yksikön Hyvinvointiteknologian tutkimus- ja tuotekehityskeskukseen (HYTKE) suunnitteleman Maksimivoimateknologian (Mustin) toistomäärälaskurin algoritmin toimintaa, joka laskee käyttäjän suorittamat toistot nostettaessa painoja. Toistomäärälaskurista oli jo olemassa Matlabilla toimiva versio, jonka kääntäminen C-kielelle Musti-ranneketta varten oli tätä työtä tehtäessä aloitettu. Entinen toistomäärälaskuri ei kuitenkaan ollut toiminnaltaan täysin luotettava, koska se ei esimerkiksi hyväksynyt ajallisesti huomattavan pitkäkestoisia toistoja.

Ehdotuksia toistomäärälaskurin parantamiseksi oli monia. Tässä työssä testattavaksi valittiin aallokeanalyysi, josta uskottiin olevan apua. Uutta toteutusta kokeiltiin Matlabin ilmaisella WaveLab Toolboxilla ja työkaluna käytettiin lopulta CWT-muunnosta (continuous wavelet transform).

Uudella algoritmilla onnistuttiin saamaan lisää tarkkuutta eripituisten toistojen tunnistamiseen ja sen myös katsottiin olevan riittävän kevyt toimiakseen sulautetussa järjestelmässä. Lopullinen näkemys algoritmin toiminnasta saadaan siinä vaiheessa, kun se tullaan sulauttamaan osaksi nykyistä C-kielistä ohjelmakoodia.

Degree programme	Thesis	Number of pages	+	Appendices
Medical Engineering	B.Sc.	65	+	1
Line	Date			
Option for Home Care Technology	2011			
Commissioned by	Author			
Medical Engineering R&D center	Arto Tiitto			
Thesis title				
Wavelets and Maximum Strength Technology Algorithm				
Keywords				
Algorithm, Chronos, maximum strength, analysis, development, wavelets				

The purpose of this work was to improve repetition counter algorithm of Maximum Strength Technology (Musti) developed by Medical Engineering R&D center of Oulu University of Applied Sciences. Repetition counter counts weight lifts done by user. There was already available a version of the repetition counter for Matlab and it was already started to be compiled into C-language for Musti wristband while this work was started. This earlier repetition counter was not functioning reliably enough, for example because it did not accepted significantly long repetitions by time.

Many methods for improvements were suggested. For this work, wavelet transform was chosen to be tested, because it was believed to be a helpful solution. New implementation was experimented with a free Wavelab Toolbox for Matlab. As a tool, continuous wavelet transform (CWT) was chosen in the end.

Succession for differently timed repetitions to be recognized was possible with the new algorithm, and it was proven to be lightweight enough to be workable in an embedded system. Final vision of operation for the new algorithm will be seen when it will be embedded to be part of the current code of the C-version.

ACKNOWLEDGEMENTS

First of all, thanks to Dr Manne Hannula and Dr Jukka Jauhiainen from Medical Engineering R&D center at Oulu University of Applied Sciences for commissioning Bachelor's Thesis work. Thanks to MSI Kari Jyrkkä for supervising the work, L Aira Huhta for linguistic revision, Eng Henry Hinkula for help and others who have helped me during the timespan of the work. Special thanks to Musti-team for co-operation and creating good atmosphere for working.

Oulu 25.5.2011

Arto Tiitto

CONTENTS

TIIVISTELMÄ.....	3
ABSTRACT.....	4
ACKNOWLEDGEMENTS.....	5
CONTENTS.....	6
GLOSSARY.....	7
1 INTRODUCTION.....	9
2 PHYSICS BASIS OF ACCELERATION.....	10
2.1 2nd law of Newton.....	10
2.2 Acceleration.....	10
2.3 Gravity.....	11
3 ACCELEROMETER.....	12
4 MAXIMUM STRENGTH TECHNOLOGY.....	14
5 DEVICE AND SOFTWARE.....	16
5.1 Hardware of Musti.....	16
5.2 Software of Musti.....	18
6 MUSTI-ALGORITHM IN MATLAB.....	21
6.1 Searching of the reference signal.....	26
6.2 Detection of the sample signal.....	30
6.3 Repetition counter.....	30
7 WAVELETS AND MUSTI-ALGORITHM.....	34
7.1 Wavelet transform.....	34
7.2 Continuous wavelet transform.....	39
7.3 New algorithm.....	39
8 RESULTS.....	50
8.1 Results for early C-version.....	50
8.2 Results for new C-version.....	51
8.3 Results for new Matlab version.....	52
9 DISCUSSION.....	59
10 CONCLUSION.....	61
REFERENCES.....	62
APPENDICES.....	65

GLOSSARY

A/D-conversion	A conversion where analogue input will be converted into digital output.
Algorithm	A well-defined method expressed as a finite list for solving a specific problem.
Biosignal	A term for denoting any signal that can be monitored from biological beings.
Butterworth-filtering	Filtering method, which is not very sensitive to phase shift if small coefficients are used.
C-language	A high-level programming language originally designed to be used with Unix.
Capacitance	Variable for measuring an ability to store charge by a capacitor.
Cross-correlation	A method for measuring similarity of two waveforms.
CWT	Abbreviation for Continuous wavelet transform, which is used for dividing a continuous-time function into wavelets.
Delta (Δ)	In science, used for denoting any change for a variable.
Derivative	Used for defining local or temporal changing speed of function.
Electrocardiogram (ECG)	A graph of electric impulses describing operation of heart.

Fourier transform	Tool for time-frequency localization based on the idea that any continuous signal can be presented by integrals of sine functions.
Gyroscope	A device for measuring orientation.
Interval	Range of numbers in mathematics, relation between two notes in music.
LabVIEW	A visual programming language developed by National Instruments.
Omega (ω)	Here: symbol for denoting angular frequency.
Pi (π)	Mathematical constant (3.14159) representing the ratio of circumference to the diameter of any circle.
QRS complex	Definition of three surges (Q, R and S) seen in ECG-graph representing ventricular contraction.
Spectral analysis	A procedure, where a time series are decomposed into a spectrum of cycles of different lengths.
Spectrum	Distribution of the observable variable to different components by frequency.
Wavelet transform	Tool for time-frequency localization like Fourier transform, but it is more effective zooming in shortlived high frequency phenomena.
Xi (ξ)	In science, used for denoting random variable or frequency.

1 INTRODUCTION

Maximum Strength Technology (Musti) has been started in Medical Engineering R&D center at Oulu University of Applied Sciences in 2004 as a project titled MaVo, originated from Finnish words Maksimi and Voima (Maximum and Strength). The name changed to Maximum Strength Technology (MST) in 2010. The technology is developed for people doing strength training in both beginner and semi-advanced level.

This new technology will become commercial during the period of 2011–2012. The working title for the technology turned out to be Musti in 2011. The goal is to get mock-up prototype without technology for commercial use available by the end of May 2011. Final wireless prototype is planned to be available after summer 2011. During the period of 2011–2012, all the functions and hardware specifications must be developed to the level that is needed to match the quality of a commercial product. Currently, the project involves over 30 members of staff from departments of technology, health and social care, culture and business economics at Oulu University of Applied Sciences, designer students from University of Lapland and engineer students from University of Jyväskylä.

The main functions in Musti-technology are repetition counter, individual movement analysis and series of movement analysis. In this work the goal is to make repetition counter more reliable than in the earlier stage during the development history. Repetition counter is the part that counts all the weight lifts of the product user. This function is part of algorithm which includes all the main function of the product.

This work has been done for Medical Engineering R&D center at Oulu University of Applied Sciences, which develops new technology in the area of medical engineering and is also serving companies and applicants in the areas of research, development and testing. Medical Engineering R&D center has both engineers and students working on the projects.

2 PHYSICS BASIS OF ACCELERATION

2.1 2nd law of Newton

The heavier object needs to be moved the greater force is needed. The quicker wanted terminal velocity trying to get reached, so the greater is acceleration the greater force is needed to be used. When mass \bar{m} is constant, force \bar{F} is directly proportional to acceleration \bar{a} . (Inkinen – Tuohi 1999, 94–95.)

$$F = ma$$

FORMULA 1

Force is defined by derivative variable got from an equation. From that SI-consistent unit can be received, which is *newton*. (Inkinen – Tuohi 1999, 95.)

$$[F] = [m][a] = \text{kg} \cdot \text{m}/\text{s}^2 = \text{newton} = N$$

2.2 Acceleration

If velocity of an object changes, object is in acceleration movement. Average acceleration a of an object equals change of velocity $\Delta v = v_2 - v_1$ divided by used time $\Delta t = t_2 - t_1$. (Inkinen – Tuohi 1999, 49.)

$$a = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1}$$

FORMULA 2

In SI, unit of acceleration is m/s^2 . In case of linear movement, direction of acceleration will be taken noticed by sign. Like velocity, acceleration is vector variable. (Inkinen – Tuohi 1999, 49.)

2.3 Gravity

Gravity \bar{G} is force, which affects to an object by local field of gravity. Gravity is depended by location in earth or space. (Inkinen – Tuohi 1999, 98.) In earth it is varied by latitude and height. For example, at sea level in Helsinki $g = 9,9190$ m/s² and in Oulu 9,8224 m/s². In calculations, approximated normal value 9,81 m/s² is used. (Inkinen – Tuohi 1999, 61.)

$$\bar{G} = m\bar{g}$$

FORMULA 3

3 ACCELEROMETER

In the measured object, signal is formed by a measurement sensor. Sensing element used for probing the object is the most important part of a sensor. For different variables to be probed, different kind of sensing elements are needed. The most important feature of a sensor is that it can give a signal which is informative and suitable for processing. (Aumala – Ihalainen – Jokinen – Kortelainen 1998, 6–7.)

CMA3000-D0X (figure 1) is a small sized, low priced and low power consuming triaxial accelerometer family targeted for high volume products. It consists of a signal conditioning ASIC and a 3D-MEMS sensing element in a wafer level package. (VTI Technologies 2010, 5.)

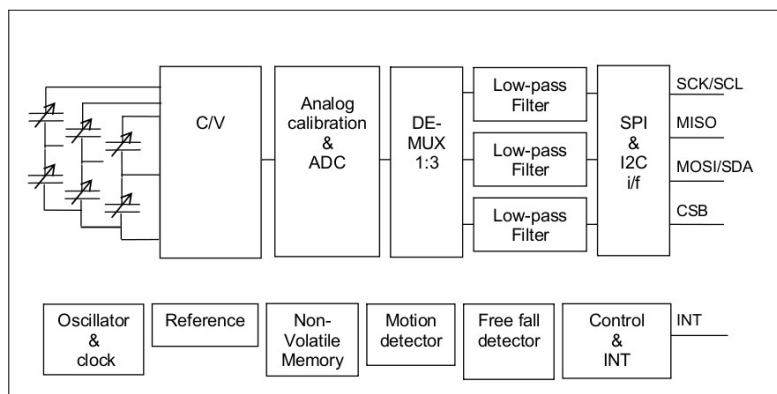


FIGURE 1. CMA3000 D0X Block Diagram (VTI Technologies 2010, 5.)

The sensing element consists of three acceleration sensitive masses is manufactured using proprietary bulk 3D-MEMS process enabling robust, stable and low noise & power capacitive sensors. Acceleration will cause a capacitance change that will be then converted into a voltage change in the signal conditioning ASIC. (VTI Technologies 2010, 5.)

Current and voltage references and internal oscillator are included in CMA3000. There is also non-volatile memory enabling the sensor's autonomous operation

within a system. The sensing element is interfaced by a capacitance-to-voltage (CV) converter. The signal is A/D-converted and then digitally filtered. Digital SPI or I²C interface is user selectable sensor output. (VTI Technologies 2010, 5.)

Acceleration data can be read by the serial bus in measurement mode and in power down mode the device is in-active. Other features, which are supported are motion and free-fall detection. In these modes, the sensor will generate an interrupt when a pre-defined condition has been met. Measurement range can be selected via register command. (VTI Technologies 2010, 5.)

Sensors of CMA3000 are factory calibrated. Trimmed parameters are gain, offset, frequency of the internal oscillator and internal current reference. During sensor startup, calibration parameters will be read automatically from the internal non-volatile memory. (VTI Technologies 2010, 6.)

4 MAXIMUM STRENGTH TECHNOLOGY

Maximum strength is the highest personal level of strength, which can be produced by a muscle or musclegroup in volitional single contraction. It takes 0,5–2 seconds to reach maximum strength level depending on age, sex, muscle working levels, musclegroup to be measured and training background. (Mero – Nummela – Keskinen – Häkkinen 2004, 285.)

Maximum strength can be indicated as maximal force, Newtons (N), torque (Nm) or kilograms (kg). Being competent to do testing movements is the condition for producing maximum strength. It is essential to control directing of strength production towards measuring sensors by strength measurement. Measuring strength production is specific, so each test can be used only for measuring strength production characteristics of a specific nerve-musclesystem. (Keskinen – Häkkinen – Kallinen 2004, 138.) The highest load that can be lifted once for a certain movement with proper technique is called one repetition maximum (1-RM). Typical testing movements can be bench press, squat, or other movements that can be done with different kind of gym equipment. (Mero et al. 2004, 285.)

For getting reliable results, first thing needed to do is that testee does warm-up series of 5–10 for 1–3 times with estimated 40–60 % load of maximum. After that, close-up series of 3–5 for 1–3 times with estimated 60–80 % load of maximum. After that, load will be increased little by little and every time (for example 90 %, 95 % and 100 % of maximum), only one repetition will be performed until maximum is finally found. The goal is to find maximum weight after 3–5 attempts. Resting from 3–5 minutes is allowed between each repetition. (Mero et al. 2004, 285.)

1-RM method is considered dangerous in due to usage of heavy weights and consumption of time. (Hannula – Leskinen – Linnamo – Rontu – Salmi 2007, 4.) Because of that, many different submaximal methods have been developed. In addition to these methods, there have been studies to implement methods

using anthropometric dimensions in predicting 1-RM performance. (Hannula et al. 2007, 17.)

During the testing of maximum strength with 1-RM method, usually heavy weights are being used. This may lead to the risk of injury. With a new method developed by Medical Engineering R&D center at Oulu University of Applied Sciences it is possible to measure maximum strength safely. The method does not require usage of maximal weights. The maximum repetition will be done with submaximal weight and maximum strength will be estimated based on the acceleration of the lift. In this way, the risk of injury will be reduced. (Hannula et al. 2007, 4.)

Musti is based on earlier researches. In research "1-RM bench press performance estimated with accelerometer method" by Manne Hannula, Sami Leskinen, Vesa Linnamo, Jari-Pekka Rontu and Jukka Salmi estimating reliably 1-RM with a new accelerometer-based method was studied. The subjects of the research were 22 active floorball players between 16–30 years, 164–186 cm and 61.0–86.6 kg. The submaximal bench press lift was recorded with a triaxial accelerometer integrated to a wrist equipment and a data acquisition card. The maximum acceleration was calculated from the measurement data of the sensor and analyzed with LabVIEW based software in PC. Then the same 1-RM tests were done with traditional method and the results were compared to each other. The new method was evaluated by using different loads (50, 60, 70, 80 and 90 % of the measured 1-RM) and for each load level, own estimation equation, i.e. five different estimation equations have been used based on the measured 1-RM values of the subjects. The differences between the methods were -0.11–0.01 kg, so they were not significantly different from each other. As the conclusion, the new method based on one submaximal lift gave promising bench press 1-RM estimation results. (Hannula et al. 2007.)

5 DEVICE AND SOFTWARE

5.1 Hardware of Musti

The very first prototype of the wristband was developed by Manne Hannula and Sami Äijälä in summer 2005. During the measurements, wristband was attached to the wrist of the testee before performing a bench press lift. During the test, the wristband sends acceleration data of X-, Y- and Z-axis to a computer software. With the software it was possible to calculate the maximum weight the testee can lift. In autumn 2006, a new version of the prototype was developed by medical engineering student Henry Hinkula (figure 2). (Lammela 2007, 8.)



FIGURE 2. 2006 prototype of maximum strength meter wristband (Similä 2010b, 24.)

The wristband prototype designed in 2006 has an integrated accelerometer based on condenser with sensitivity of $V_{dd}/5$ and offset (V_{off}) of $V_{dd}/2$. The model was LIS3L02AQ3 manufactured by ST Microelectronics. The wristband has a display type HPDL-1414 including parallel data controlled display

controller. The display control worked with 8-bit part of digital side in NI-6009 measurement board. (Lammela 2007, 20.) The sample rate of the accelerometer was 100 Hz. (Lammela 2007, 25.) The wristband has a middle button and minus- and plus-buttons on its side. The program is controlled by these buttons and is used for functions like starting the program or changing the weight. (Leiviskä 2009, 10.)

Analogical technology had problems. Because the cable was very long, it tended to cause inconvenience to the users. Long cable length caused also bad noise to the signal. It was also obvious that if the user knocked the pole somewhere during the lift, it caused false peaks which could be later calculated as results. The next step was to investigate for wireless digital technology and for that purpose Medical Engineering R&D center chose TI eZ430-Chronos.

TI eZ430-Chronos (figure 3) is a fully functional sports watch based on the CC430F6137, MSP430 MCU with integrated <1GHz wireless transceiver. The watch has an integrated triaxial accelerometer (VTI CMA3000 Series). The device includes an eZ430 USB emulator that connects the Chronos to a PC for realtime. (Texas Instruments Embedded Processors Wiki. eZ430-Chronos.) Chronos software developed by Medical Engineering R&D center uses sample rate of 30 Hz.



FIGURE 3. TI eZ430-Chronos

Musti is being developed in the way that the wristband will use the same technology as TI eZ430-Chronos. A chest strap for measuring heart rate and a gyroscope for measuring the body movements will be also included. A PC will not be required anymore, because It would be inconvenient in the gym environments. A PC would still be needed for keeping training diary at home by the user. The plan is to make the algorithm controlled by a mobile device – or an external pocket device and thus the algorithm needs to be developed to be enough lightweight to run in such conditions.

5.2 Software of Musti

The very first version of the Musti-software was developed with LabVIEW in Medical Engineering R&D center. The improved LabVIEW-version with better usability was coded by Petteri Leiviskä (figure 4). The first version written in C including repetition counter for the first lime was coded by Antti Kylmänen. It was designed for analogical accelerometer used in the first prototypes. For digital accelerometer used in TI eZ430-Chronos, the C-based software was re-written by Harri Haataja. (Hinkula 2011.)

In 2010, the following features were planned to be included to the Musti-technology:

- Repetition counter for notifying the user by voice after each performed repetition.
- Individual movement analysis for notifying the power of the repetitions done. It compares the power of each repetition to the most powerful one (calibration).
- Series of movement analysis notifies by voice in which training effect area each performed movement is (maximum strength, muscle mass or endurance). (Similä 2010a, 4.)

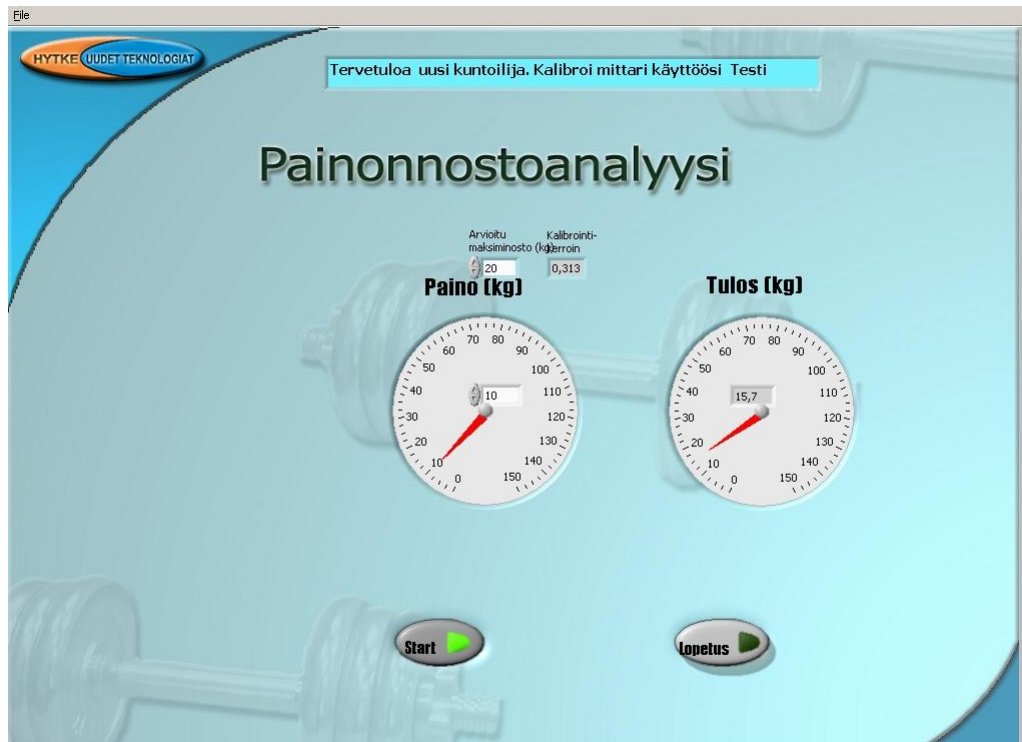
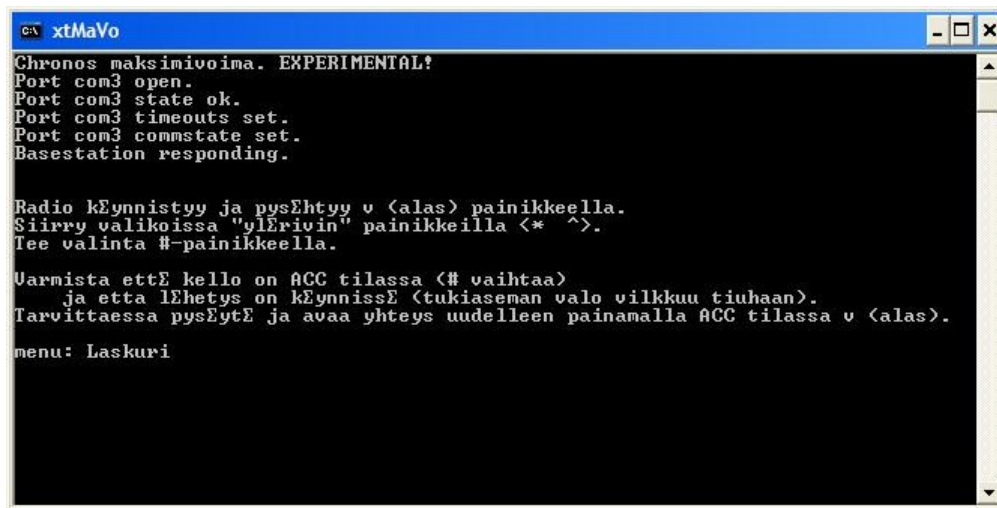


FIGURE 4. LabVIEW-software in 2009 – window for calibration (Leiviskä 2009, 35.)

The current repetition counter used, works in software written in C (figure 5), based on a simple algorithm implemented by thresholds. This algorithm inspects the accelerometer signal with two thresholds if the lift is found. Detailed implementation appears from the source codes which can be found from Medical Engineering R&D center laboratory. (Hannula 2011.)

xtMaVo for TI eZ430-Chronos is controlled wirelessly from TI eZ430-Chronos itself. Buttons from TI eZ430-Chronos are reserved for certain functions like moving left, right and doing selections in the way 2006 prototype has. All the main functions are available via menu and if a logger is wished to be used by the user, it will be selected from menu and data will be stored to a pre-defined CSV-file which will overwrite itself every time when the logger is used.



```
GN xtMaVo
Chronos maksimivoima. EXPERIMENTAL!
Port com3 open.
Port com3 state ok.
Port com3 timeouts set.
Port com3 commstate set.
Basestation responding.

Radio k ynnistyy ja pys htyy v (alas) painikkeella.
Siirry valikoissa "yl rivin" painikkeilla <* ^>.
Tee valinta #-painikkeella.

Varmista ett  kello on ACC tilassa (# vaihtaa)
ja ett  l hetys on k ynniss  (tukiaseman valo vilkkuu tiuhaan).
Tarvittaessa pys yt  ja avaa yhteys uudelleen painamalla ACC tilassa v (alas).
menu: Laskuri
```

FIGURE 5. xtMaVo – The second C-based software version written for Musti-technology

A new version of the algorithm was designed and coded by medical engineering student Veli-Pekka V limaa during summer and autumn 2010 in Matlab-language. This is the version which will be developed further during this work.

The measurement data is split in two sections: reference signal (calibration) and sample signal, which includes all the repetitions. Calibration (recorded reference signal) is personal and will be performed for each user every time before selected movement will be performed for the first time. It will be done with submaximal weight (about 60 % of estimated maximum) with maximum strength as quick as possible. The performed signal will be kept in the memory even if it has been decided by the user to select a different movement to be performed, like switching from repetition counter to series of movement analysis if the performed movement remains the same. (Simil  2010a, 5.)

6 MUSTI-ALGORITHM IN MATLAB

Matlab (figure 6) is an interactive environment and high-level language, which enables the user to perform numeric computation and other scientific tasks, such as analyzing, accessing and visualising data and developing algorithms and applications. For that purpose, it is faster than than traditional high-level programming languages like FORTRAN, C or C++. (The MathWorks, Inc. 2011.)

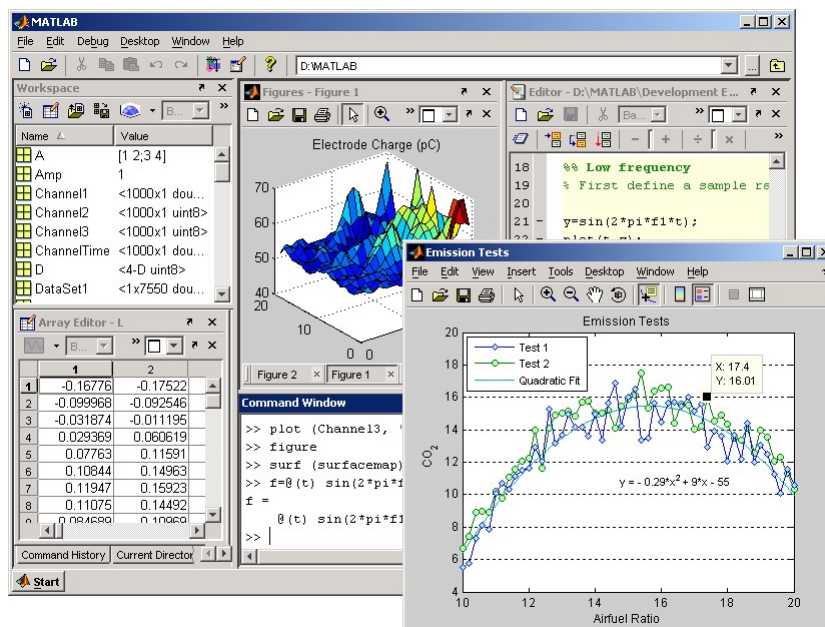


FIGURE 6. Matlab environments (The MathWorks, Inc. 2011.)

The current Matlab version of the algorithm has not been tested by using a prototype of the wristband yet. It will only work on a computer screen by using pre-recorded data. In this chapter, pre-recorded data for biceps curl has been used as an example. The version has been coded to C by engineering student Arto Rajaniemi during spring 2011 and the improved version will be coded after that if the new Matlab version proves to be successful. Appendix 1. contains a comprehensive block diagram of the old Matlab-implementation.

In the Matlab version, the first thing to do by user is to pick up a file where

recorded reference and sample signals are stored in CSV-format including four columns of data: sample number and acceleration data for X-, Y-, and Z-axis of accelerometer (figures 7–9). As can be seen, all the axes will provide different acceleration values. Both signals will be transposed and after that signals will be filtered with Butterworth-filter, so most of the noise will be eliminated. Comparing figures 10 and 11 the difference between filtered and non-filtered signals are shown. Software will also create a sum signal (figure 10) which is calculated from X-, Y- and Z-axis data with formula 4. (Välilmaa 2010.)

$$\sqrt{(x^2 + y^2 + z^2)}$$

FORMULA 4

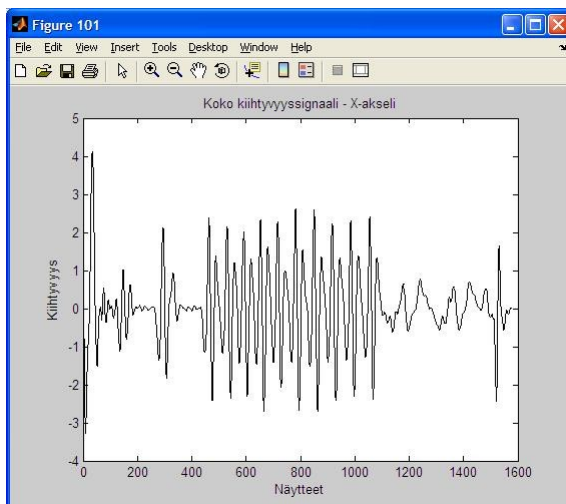


FIGURE 7. Full signal for biceps curl – X-axis

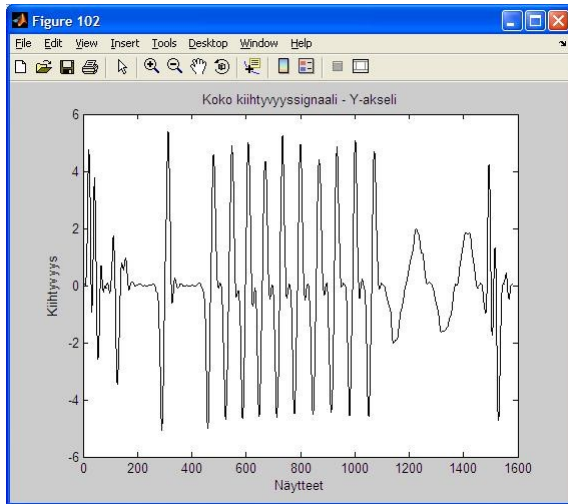


FIGURE 8. Full signal for biceps curl – Y-axis

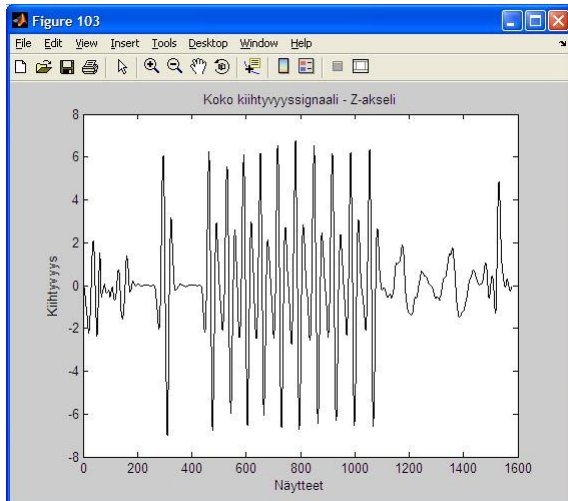


FIGURE 9. Full signal for biceps curl – Z-axis

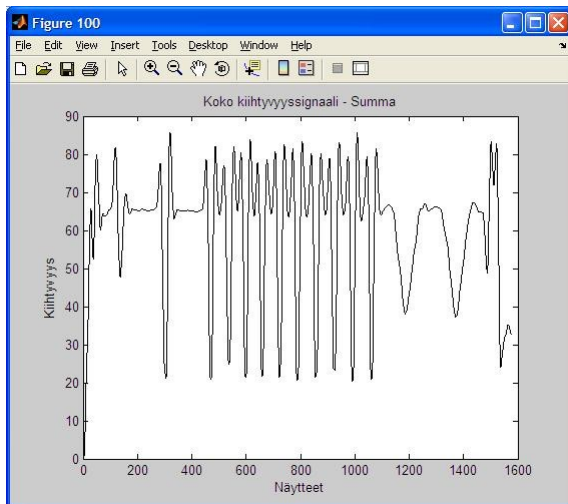


FIGURE 10. Sum signal for biceps curl

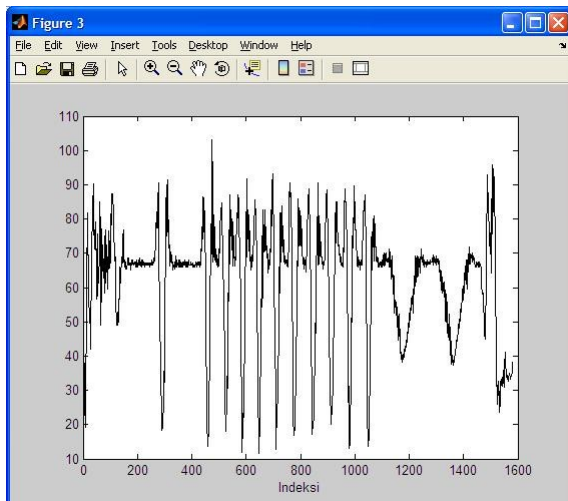


FIGURE 11. Unfiltered sum signal for biceps curl

The user will be prompted to select reference signal and sample signal separately from pre-recorded data and it will be analyzed after that. Then it will be needed to place two cursors to the figures showing the signals by user. One for defining the starting point of the reference signal, other for defining the ending point of the reference signal and the same will be done for sample signal (figure 12). (Välimala 2010.) From now on, every figure containing both reference and sample signal, red color will be used for indicating reference signal and blue for sample signal.

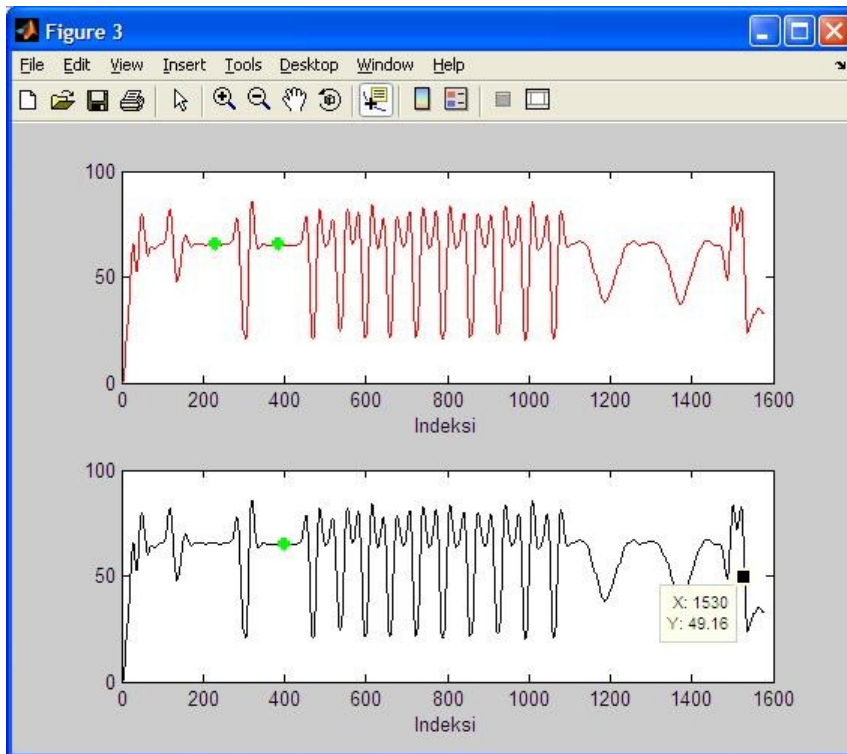


FIGURE 12. Picking the biceps curl reference signal (upper window) and sample signal (lower window) in Matlab version (Please note the cursors for selections made by user.)

As already stated, the Matlab version does not currently support realtime signal analysis. For the final C-based version, the algorithm must be made in the way that reference signal and sample signal will be detected automatically. Basically, this problem may not need any workaround in this work as it will be solved during the compiling into C. On the other hand, a new method would be needed in the Matlab version, because the current code has several errors. An example of an error can be seen in figure 13.

```

Klikkaa referenssingaalin alkukohta.
Kiitos!
Klikkaa referenssingaalin loppukohta.
Kiitos!
Klikkaa samplesingaalin alkukohta.
Kiitos!
Klikkaa samplesingaalin loppukohta.
Kiitos!
??? Attempted to access aloituskohta_ref_lepotilassa(:,1); index out of
bounds because size(aloituskohta_ref_lepotilassa)=[1,0].

Error in ==> Chronos_toistolaskuri_siistitty at 336
naytteet_lepotila_alku_alkaa=aloituskohta_ref_lepotilassa(:,1)-lepotila_
keskiarvoon_hyvaksytyt_naytteet

>> |

```

FIGURE 13. Error right after picking up signals – program terminated

6.1 Searching of the reference signal

If the program does not terminate itself with an error, the analysis will be started automatically. Before the final reference signal searching and sample signal detection, all the starting and ending points will be rounded to even numbers, because it was thought that some errors, such as problems during possible signal divisions, could be avoided. (Välímää 2011.) After that, both signals will be created to new vectors for the analysis in the sizes which are decided by user after giving the starting and ending points (figures 14–15). After that, total accelerations (sum signals) will be calculated for both signals and derivated sum signals will be made too. Derivated signals are only used for detecting resting values and trigger thresholds, where it is needed to know the calculated zero-levels of the signals. (Välímää 2010.)

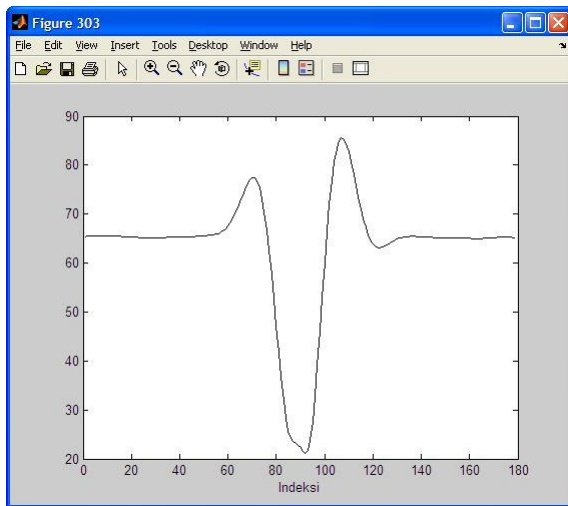


FIGURE 14. Reference signal selected by user for biceps curl

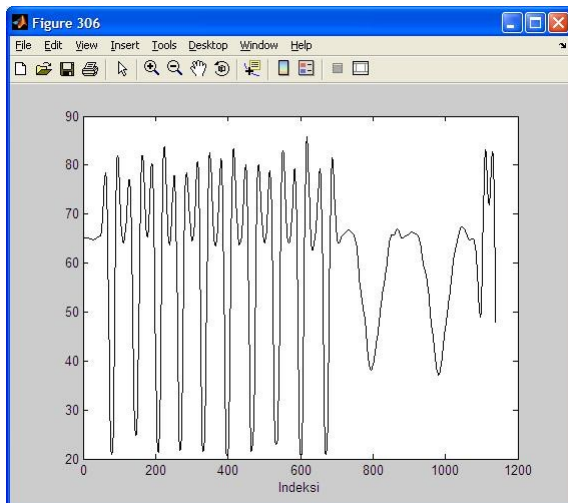


FIGURE 15. Sample signal selected by user for biceps curl

After the signals are initially created, threshold value from reference signal for resting area will be defined. This is when lift has not been started or has just stopped. When the derivate stays between defined threshold values, there will be no major changes in the acceleration data. Starting and ending points will be defined by detecting where threshold values are going higher or lower than allowed during the rest. Then some extra samples from the start and the end of the signal will be picked. It has been made to pick user defined amount of values of the resting period from both ends of the signal pointing the beginning and the end of the signal. This should be done for making sure that the detection of the beginning and the end of the signal are as correct as possible.

Now, the final reference signal is created (figure 16). For calculating mean from the voltage values of the beginning and end of the reference signal, less samples will be used. This will be done, because if the movement the user has done is not clean, it is possible to set the value manually to limit where the cleanly performed movement has been done in the code. So, it is possible that the lift does not end to the zero level as it started, but somewhat above or below the zero level defined with derivation. The resting level at the beginning will be compared to the resting level at the end of the signal and it will be examined if the ending will be accepted or not. (Välilmaa 2010.) This check has caused errors at some points in the development of the algorithm. (Välilmaa 2011.)

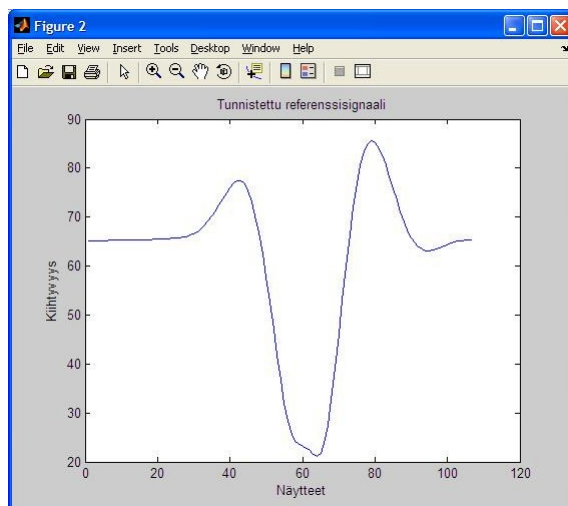


FIGURE 16. Detected reference signal for biceps curl

After the detected reference signal is found, values for signal length, minimum, maximum and value from the beginning and the end will be picked. For getting comprehensive results, all local minimums and maximums are detected by using by a subprogram called `localMaximum.m` by MathWorks. (Välilmaa 2010.) Minimums or maximums from a vector or matrix will be examined by the subprogram after user defined minimum distance between two peaks. It is also possible to make the subprogram to prevent detecting same value twice. IF-ELSE-conditions for the procedure are used by the subprogram.

The beginning and the end of the signal will be filtered by the algorithm for

eliminating minor minimums and maximums. After that, places for detected minimum and maximum values will be detected and put in order for defining the shape of the sample signals. When the places of minimums and maximums are known, it is possible to guess the length of the sample signal and the way how minimums and maximums are located in the sample signal. Trigger thresholds will be created for setting limits to the values for minimums and maximums while detecting the shapes of the repetitions within sample signal. (Välismaa 2010.)

In this section, the biggest problem is the amount of errors after given starting and ending points by user for both signals. Even when minimums and maximums are detected with two different methods, there may still be too many of them detected. It is not always possible to know which value is truly the most meaningful in the end, because the highest hilltop or lowest valley may appear in many places, specially when the movement is slow and choppy. It is suggested that the method for detecting minimums and maximums should be more accurate and possible more simple. It is possible to split signal to two pieces and look for only those minimums and maximums which are needed for the analysis.

Because the speed for doing reference signal is individual, the way extra samples from beginning and end of the signal picked by the algorithm does not give realistic results every time. Assuming that the speed for doing lift at the time of recording the reference signal is very slow, the real beginning and end of the signal may stay within levels set for resting values. So it is possible that the signal will be cut too late at the beginning and too soon at the end. (Välismaa 2011.)

It has been suggested that the running signal should be sniffed all the time whenever recording reference or sample signal data. This is the way it should work in realtime picking the right signal starting and ending points in the right time. The same could be done with trigger thresholds, so the assumed places for them could be found dynamically without a need to trying just to guess their locations. (Välismaa 2011.)

6.2 Detection of the sample signal

After searching for the points and their locations where the conditions are met for minimums and maximums defined during analyzing the reference signal, a vector containing both location information and value will be created. Then, by only using location information, a vector containing all value and location information above or below limit values is created. Then, value amounts for suitable spots will be defined. Minimum and maximum candidates of sample signal must be compared separately, because some of them must be higher or lower than in reference signal. Candidates will be compared to trigger thresholds. Then it will be examined, how much extra samples must be given to the beginning and end of the sample data by using help from trigger order. Now, the final reference signal is created (figure 17).

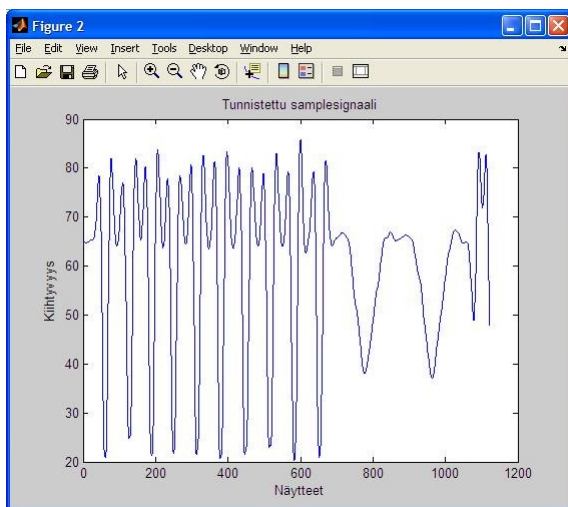


FIGURE 17. Detected sample signal for biceps curl including all repetitions

6.3 Repetition counter

Full sample signal will be gone through by the algorithm shape matching with the reference signal will be looked for. If minimums and maximums will meet conditions, then cross-correlation will be done for each axis (X, Y and Z). If the calculated values are within the defined values (> 0.80 for all axis), movement will be counted as a repetition (figures 18–20). This procedure will continue as

long as sample signal ends.

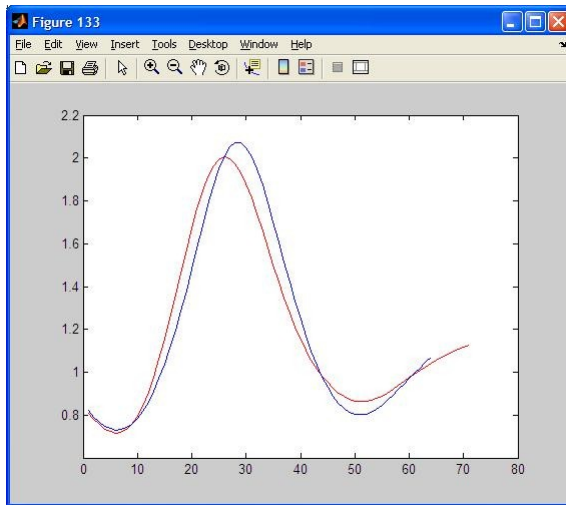


FIGURE 18. Accepted repetition after cross-correlation for X-axis

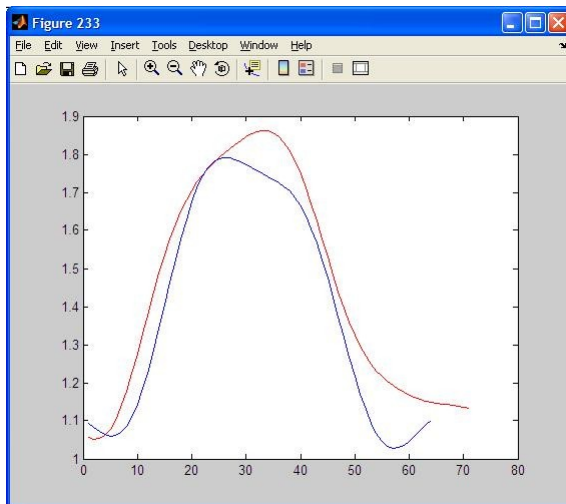


FIGURE 19. Accepted repetition after cross-correlation for Y-axis

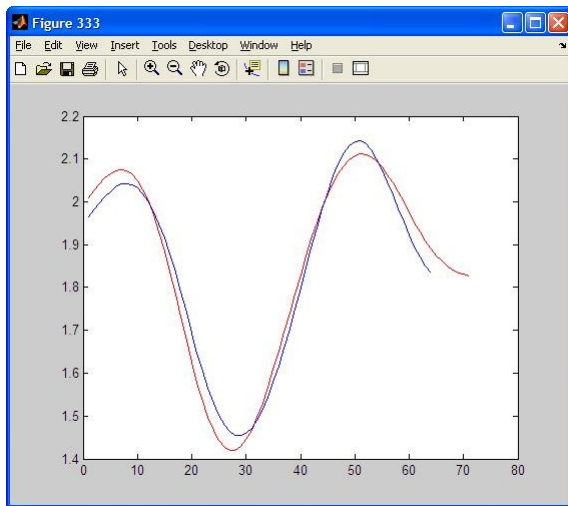


FIGURE 20. Accepted repetition after cross-correlation for Z-axis

The biggest problem in repetition counter is the situation if fatiguing is happening while doing the series by user. Then the movements will sometimes get much slower than reference signal (calibration) and they may be not detected as repetitions at all. The values can not be compared by the algorithm, because distances will get longer between them. Also, the sum signal shape may give too weak hilltops or valleys to be noticed.

As the current method goes through the sample signal very densely, it has been suggested that it could be analyzed in like 100 samples at a time in a window and going through the sample signal that way, which would make it easier and realistic with realtime analysis. The only problem there would be that if the signal start point would remain to the window already checked through, the rest of the signal would remain without confirmation. It might be possible to make algorithm to memorize the point where the signal would have been started, but it might not be very easy and simple to code. (Välimaa 2011.)

It has been suggested that for shape recognition it would be wiser to use a single axis signal for shape recognition, while for calculating total acceleration, sum signal would be used. It is noticed that a recognizable shape can be given by a single axis signal even during a very slow movement. For slowness, the reference signal needs to be scaled somehow to match sample signals within different lengths. There are many methods which could be tested. The simplest one would be to detect the points where each sample signal starts and ends

and by that, using scaling for reference signal. It would also be possible to keep up with current system, but using wavelet transform for shape recognition.

7 WAVELETS AND MUSTI-ALGORITHM

In the meeting arranged in 14th February 2011 it was decided that Matlab-based algorithm will be improved by using wavelets in the repetition counter if the method appears to be useful. Idea was originally made by Manne Hannula and it was found useful to try by the team. It was noted that with current algorithm, it was not believed to get great improvements without help of a gyroscope, so it was better to look for a completely new solution.

7.1 Wavelet transform

Wavelets provide a tool for time-frequency localization. The standard Fourier transform,

$$(Ff)(\omega) = \frac{1}{\sqrt{2\pi}} \int dt e^{-i\omega t} f(t) , \quad \text{FORMULA 5}$$

also gives a representation of the frequency content of f , but information concerning time-localization like high frequency bursts cannot be taken easily off from Ff . After windowing the signal f , time-localization can be achieved. It can be done by cutting off only a well localized slice of f , and then taking a Fourier transform:

$$(T^{win} f)(\omega, t) = \int ds f(s) g(s-t) e^{-i\omega s} . \quad \text{FORMULA 6}$$

Above is the standard technique for time-frequency localization: windowed Fourier transform. It is more familiar in its discrete version to signal analysts:

$$T_{m,n}^{win}(f) = \int ds f(s) g(s-nt_0) e^{-im\omega_0 s} . \quad \text{FORMULA 7}$$

Gaussian g is a very popular choice in this context. It is supposed to be well concentrated in both time and frequency, if g and \hat{g} are both concentrated around zero. Then $(T^{win} f)(\omega, t)$ can be interpreted loosely as the content of f near time t and frequency ω . (Daubechies 1992, 1–2.)

A similar time-frequency description can be provided by the wavelet transform, with a few important differences. The wavelet transforms analogous to formulas 6 and 7 are

$$(T^{wav} f)(a, b) = |a|^{-1/2} \int dt f(t) \psi\left(\frac{t-b}{a}\right) \quad \text{FORMULA 8}$$

and

$$T_{m,n}^{wav}(f) = a_0^{-m/2} \int dt f(t) \psi(a_0^{-m} t - nb_0) \quad \text{FORMULA 9}$$

In both cases can be assumed that ψ satisfies

$$\int dt \psi(t) = 0 \quad \text{FORMULA 10}$$

The functions $\psi^{a,b}$ are called wavelets. The function ψ is sometimes called mother wavelet. A typical choice for ψ is the second derivative of the Gaussian ($\psi(t) = (1-t^2)\exp(-t^2/2)$), sometimes called the Mexican hat function (figure 21), which is well localized in both time and frequency, and satisfies formula 10. The $\psi^{a,0}(s) = |a|^{-1/2}\psi(s/a)$ cover different frequency ranges, where large values of the scaling parameter $|a|$ correspond to small frequencies, or large scale $\psi^{a,0}$, small values of $|a|$ correspond to very fine scale or high frequencies. Each $\psi^{a,0}(s)$ is localized around $s = b$, so changing the parameter b , too, allows to move the time localization center. All the $g^{\omega,t}$, regardless of value ω , have the same width, while $\psi^{a,b}$ have time-widths adapted to their frequency: low frequency $\psi^{a,b}$ are very broad,

while high frequency $\psi^{a,b}$ have them much narrower. As a conclusion: wavelet transform is better for zooming in on very short-lived high frequency phenomena than the windowed Fourier transform. (Daubechies 1992, 3.)

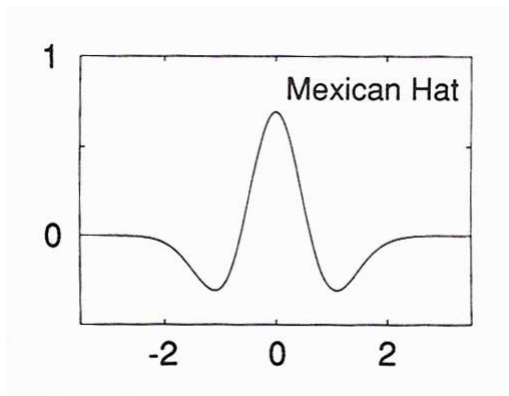


FIGURE 21. Mexican hat function (Daubechies 1992, 80.)

The absolutely basic for wavelet analysis is the idea of multiresolution. The signal is rather divided into different scales of resolution than different frequencies. Multiresolution divides the frequencies into octave bands, from ω to 2ω , instead of uniform bands from ω to $\omega + \Delta\omega$. When $f(t)$ is replaced by $f(2t)$, the compression of a graph means expansion of it's Fourier transform from $F(\omega)$ to $\frac{1}{2}F(\frac{\omega}{2})$. Frequencies shift upward by an octave, when time is rescaled by two. Time-frequency plane is partitioned naturally into rectangles of constant area (figure 22). (Strang – Nguyen 1996, 3.)

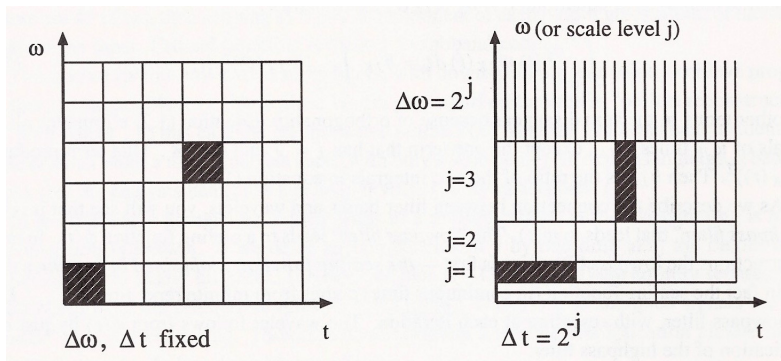


FIGURE 22. For Fourier decompositions, time-frequency squares become rectangles for wavelets (Strang – Nguyen 1996, 4.)

An article on wavelets was published in the American Scientists of May 1994. There wavelet notation was introduced through musical notation, where each note specifies a position in frequency and time. Its horizontal placement indicated time while its vertical placement gives frequency. A musical score is almost a wavelet decomposition – except that it has fractional jumps in frequency. Notes can be placed between middle C and high C, but wavelets jump by octaves. As a conclusion, in the wavelet decomposition all instruments play the same tune (figure 23)! (Strang – Nguyen 1996, 3–4.)

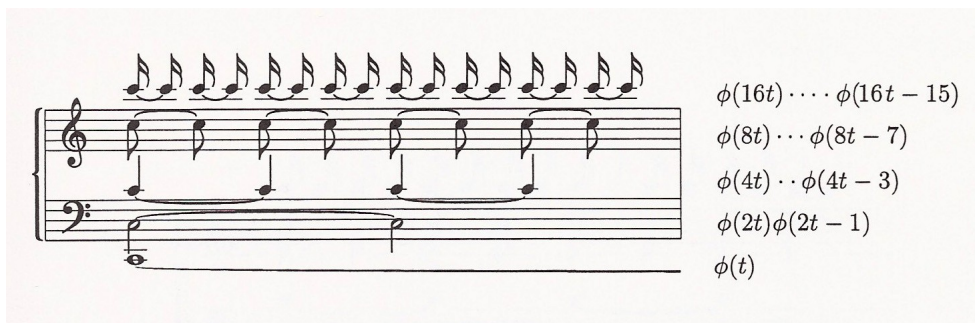


FIGURE 23. Musical score represents the wavelet decomposition to nontechnical audience (Strang – Nguyen 1996.)

The choice between waves or wavelets or between Fourier analysis or filter analysis depends on the signal, because signals coming from different applications have different characteristics. It is helpful to see the big picture:

Audio: Use a windowed (short time) Fourier transform or many subband filters.

Speech: Because the time variation is irregular, nonuniform intervals are required.

Images: Special treatment for finite length signals is needed to reduce blocking.

Video: Use space-time filtering or motion prediction (optical flow or images).

For medical imaging, the best choice is not clear. The legal questions arising from lossy compression are not clear either. (Strang – Nguyen 2011, xv.)

Continuous wavelet transform (CWT) is commonly used in interpreting segments of Electrocardiogram (ECG) signal. CWT is also used for detecting QRS complex (figure 24) using Klauder wavelets, or to perform realtime analysis of the Ventricular Fibrillation Waveform. (Ilić 2007, 33.) Because acceleration signal of weight lift has high resemblance between QRS complex and many other biosignals, CWT was a safe choice for new algorithm.

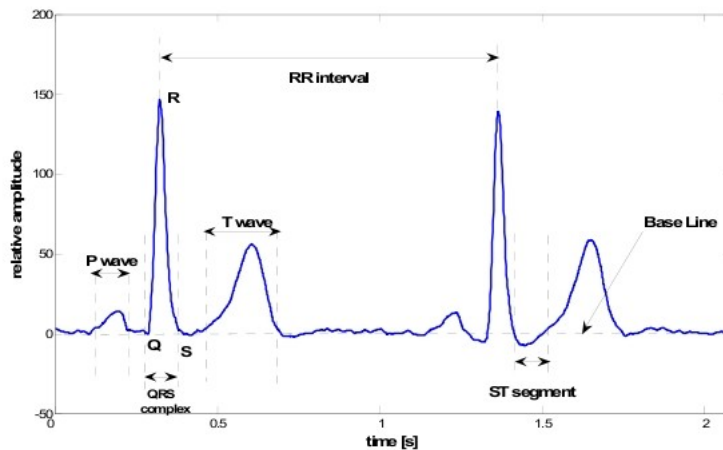


FIGURE 24. Typical ECG waveform of healthy patient (Ilić 2007, 33.)

The problem is to represent typical signals with a small number of conveniently computable functions. The traditional bases like Fourier, Bessel, etc. Come from differential equations, while wavelets do not come from them. (Strang – Nguyen 1996, xv.)

7.2 Continuous wavelet transform

In continuous wavelet transform, the dilation and translation parameters a, b vary continuously over \mathbb{R} (when $a \neq 0$). The wavelet transform is given in the formula 8. A function is reconstructable from its wavelet transform by means of the "resolution of identity" formula

$$f = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{da db}{a^2} \langle f, \psi^{a,b} \rangle \psi^{a,b}, \quad \text{FORMULA 11}$$

where $\psi^{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right)$, and $\langle \cdot, \cdot \rangle$ denotes the L^2 -inner product. The constant C_ψ is depended only on ψ and is given by

$$C_\psi = 2\pi \int_{-\infty}^{\infty} d\xi |\hat{\psi}(\xi)|^2 |\xi|^{-1}, \quad \text{FORMULA 12}$$

it is assumed that $C_\psi < \infty$ (because otherwise formula 11 does not make sense). If ψ is in $L^1(\mathbb{R})$, then $\hat{\psi}$ is continuous, so that C_ψ can be finite only if $\hat{\psi}(0) = 0$, i.e., $\int dx \psi(x) = 0$. (Note that it has been implicitly assumed that ψ is real, so for complex ψ , $\bar{\psi}$ should be used instead of ψ in formula 8.) There are two different ways by how Formula 11 can be viewed: (1) as a way of reconstructing f once its wavelet transform $T^{wav} f$ is known, or (2) as a way to write f as a superposition of wavelets $\psi^{a,b}$, the coefficients in this superposition are given by the wavelet transform of f . (Daubechies 1992, 7–8.)

7.3 New algorithm

There were many possibilities for adopting wavelets into Musti-algorithm, but there was one method which was rather easy to develop on top of the old algorithm and to meet the demands defined by low CPU and memory capacity of TI eZ430-Chronos.

The new algorithm has been developed for Matlab with WaveLab 850 Toolbox by Stanford University - Department of Statistics. (Stanford University; 2, link Contact.) WaveLab is a collection of functions that have been used by its authors and collaborators to implement a variety of algorithms related to wavelet analysis in Matlab-environment. (Stanford University.)

For keeping up with the performance of the embedded system and the accuracy high enough, four voices (notes), two octaves and four scales have been used. These can be altered in the next C-version if the performance proves to be too low or better than expected. As mother wavelet, the Mexican hat function has been used, because it is well localized in both time and frequency, but it is also easier to code for C than complex Morlet-wavelet.

Windowing method for the reference signal remained the same as introduced in the earlier Matlab version. During the development of the new algorithm, it was proven to work very well in the freshly developed C-version. This new algorithm also requires windowing for each sample signal separately, which was never done in Matlab version. This feature was introduced in the new C-version, too, so it was not necessary to develop a fully-working method for the new Matlab version. From this point, this new algorithm is not complete in Matlab version, because it will be embedded to the C-version later. Then all the gaps in the development will be filled.

Windowing method for sample signal in the new C-version works in similar way than it does for reference signal. Starting and ending points of the signal will be defined by using original sample signal and its derivated signal. (Rajaniemi 2011, 19.) (See chapter 6.1 for details.) The case assuming sample signal will be delayed in relation to reference signal can not happen in realtime optimized C-version.

New algorithm was introduced with two different versions: one, which uses the sum signal (figure 10) only for the analysis and second, which uses all of the three axis (figures 7–9). These two versions has major differences. The first version detects sample signals easier and uses less memory and CPU-power.

Triaxial version is more strict and uses more memory and CPU-power. From chapter 8.3 early test results for getting the idea of difference between strictness of the two different versions can be seen. It will be decided later by the team which version would suit better for the demands of the system specifications and needs. Triaxial version may suit better for detecting false movements, but it will need more research than have been done already. The results will not tell the whole truth, because this new algorithm allows much fine tuning during the finalization of the future C-version. In the following introduction, triaxial version and the same biceps curl data used in chapter 6 (figures 16–17) has been used.

After the both signals are picked up, reference signal (figure 16) has been windowed and sample signal (figure 17) detected, both signals will be rounded to the nearest dyadic length, because otherwise wavelet transform will fail. The first wavelet transform will be done for reference signal. In this point, a matrix will be created. The matrix will contain vectors, where each wavelet transform are stored. The CWT-matrix is ordered from low to high frequencies. The amount of wavelet transforms done are defined by the pre-defined number of voices, octaves and scales, but also sample length has affection to it (figure 25).

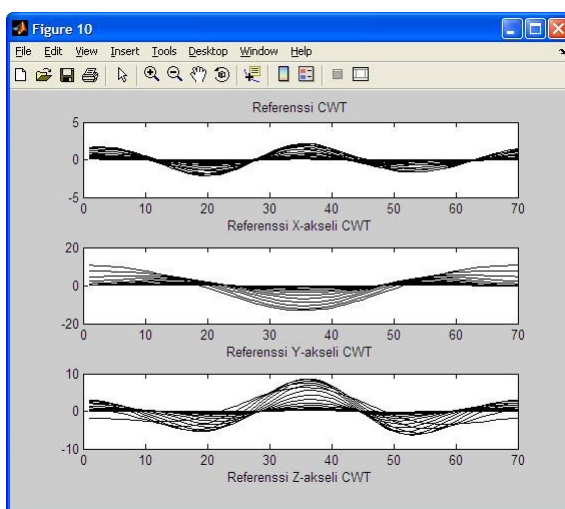


FIGURE 25. Continuous wavelet transform of the reference signal

After that, the movement for the analysis will be chosen. This will be done,

because the Matlab version does not window each sample signal in realtime. Derivated signal can not be used for detecting resting values in Matlab version for sample separation, because resting time in the used test data will hardly go higher than zero-crossing times in the acceleration data within each sample.

This version uses `peakdet.m`-subprogram by Eli Billauer for detecting the points for windowing sample signal (figure 26). All the maximums and minimums from the signal are collected by the subprogram and a point is considered a peak if it has the maximal value, and was preceded (to the left) by a value lower by user defined value delta. (Billauer 2010.)

For biceps curl, Y-axis of sample signal (figure 8) is used for detecting the gaps (maximums) between each sample. As already mentioned, this operation is not needed in the future C-version, because windowing-related problem is already solved.

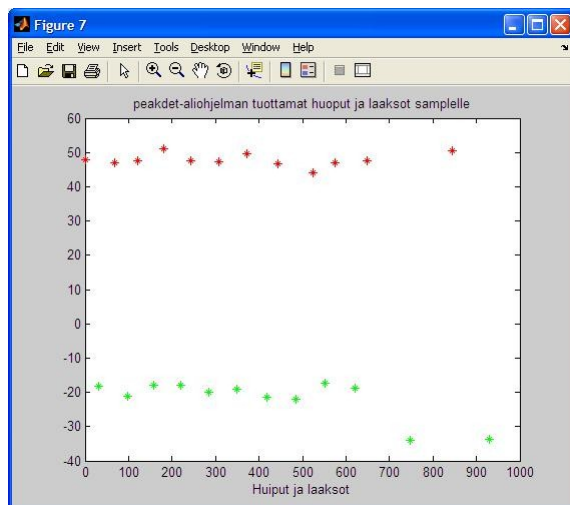


FIGURE 26. *Detected maximums and minimums for biceps curl signal by `peakdet.m`-subprogram*

When each starting and ending point for each sample signal has been detected, each sample candidate will be analyzed in a FOR-loop, which will run until there are no detected maximums left. The last sample of the signal has been added to the vector including maximums so the last sample signal will be detected as well. When the first sample signal has been windowed (figure 27), it

will be expanded to match the size of the reference signal. There are two reasons for this: firstly the signal needs to be at dyadic length, and secondly the CWT-transform function within WaveLab Toolbox can not scale the signal over the time frame if sample signal is wider than reference signal. The scaling works only within the pre-defined time frame. Then the wavelet transform will be done (figure 28).

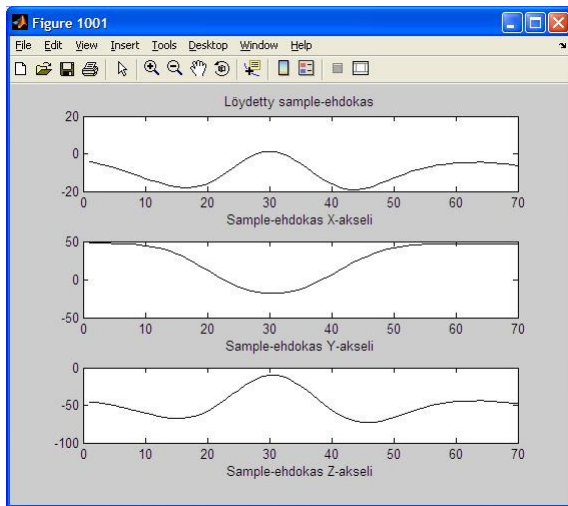


FIGURE 27. *Windowed sample candidate*

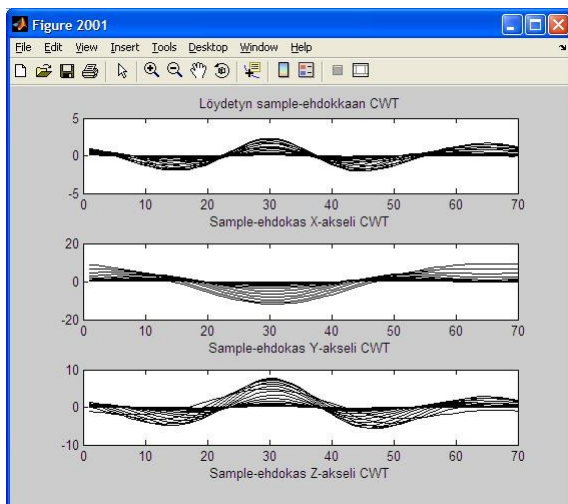


FIGURE 28. *Continuous wavelet transform of the windowed sample candidate*

The analysis will be done in the same way as in the earlier Matlab version. Cross-correlation will be used to check the similarity between reference signal

and sample signal. Acceptance value is still at > 0.80 . Inverse wavelet transform is not needed in this part and it is not included in this algorithm. It might be considered to be included in the future if it has benefits over current method.

The analysis will be done in three parts and it will be interrupted at any time if sample signal has been identified as a repetition. In the first part of analysis, vectors from matrices created by CWT-transforms will be picked in relation of 1:1 for both reference and sample signal. 50 % of the vectors will be gone through within matrices. 50 % was chosen, because during testing it was noticed that going beyond it will not give any benefits and also setting the value as low as possible helps keeping the algorithm as lightweight as possible. In triaxial version, each axis may be accepted in different scale, because movement may vary in different axis even at the same length of sample signal. The first sample candidate (figure 27) is recognized as a repetition at the first part with only two vectors gone through (figure 29). Cross correlation for the first vectors (figure 29; 1.) are accepted for Y- and Z-axis with values 0.9324 and 0.9741 while X-axis has not yet been accepted with the value of 0.7035. X-axis will be accepted when running the second vectors (figure 29; 2.) with the value of 0.8043. Analysis will stop here and the next sample candidate will be windowed. It should be noted that in the given cross-correlation figures, none of the axis figures are frozen after the acceptance of cross-correlation. During this series of repetitions, all sample candidates are recognized until no. 11 (figure 30).

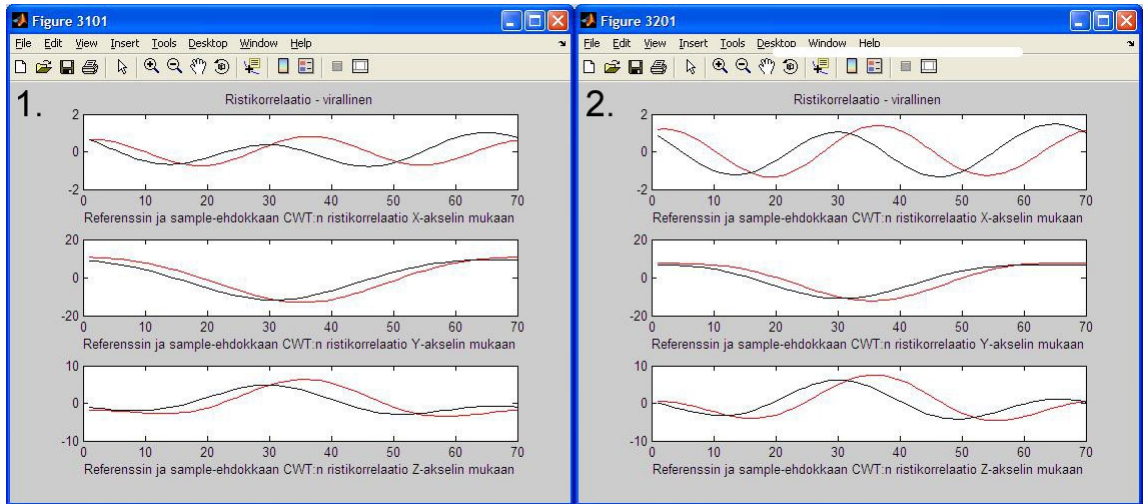


FIGURE 29. Cross-correlation sequence between reference signal and windowed sample candidate

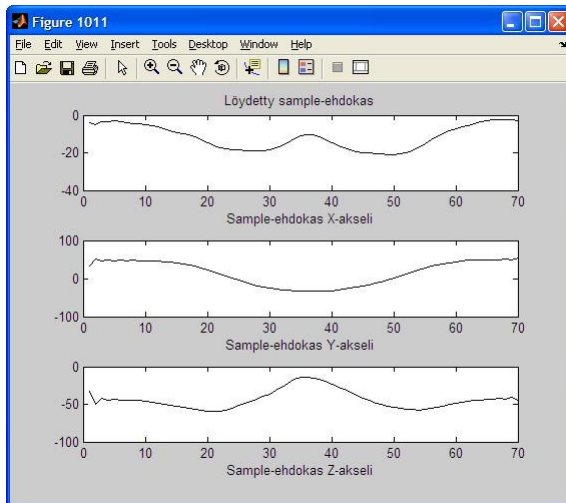


FIGURE 30. Windowed rejected sample candidate

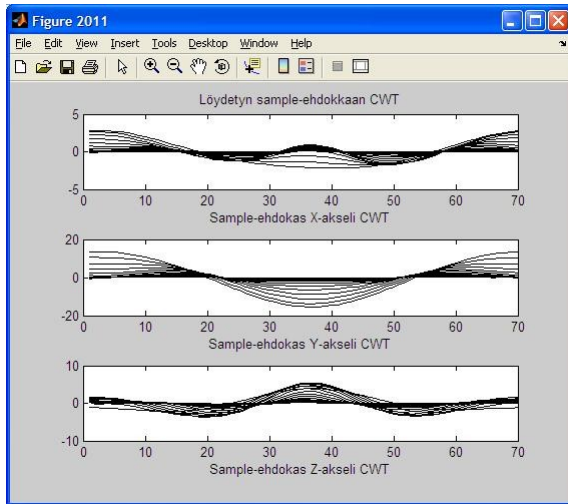


FIGURE 31. Continuous wavelet transform of the windowed rejected sample candidate

For sample candidate no. 11, 1:1 analysis (figure 32), Y- and Z-axis are accepted with values 0.9884 and 0.9861 while X-axis has not yet been accepted with the value of 0.3714 (figure 32; 1.). Analysis will continue. At vector 3 (figure 32; 2.), cross-correlation for X-axis goes to 0.4449. At vector 5 (figure 32; 3.) it will be 0.6672 and for vector 7 (figure 32; 4.) it will be 0.7028. Analysis will stop on vector 8 and because X-axis remains unidentified, the second part of the analysis will be done.

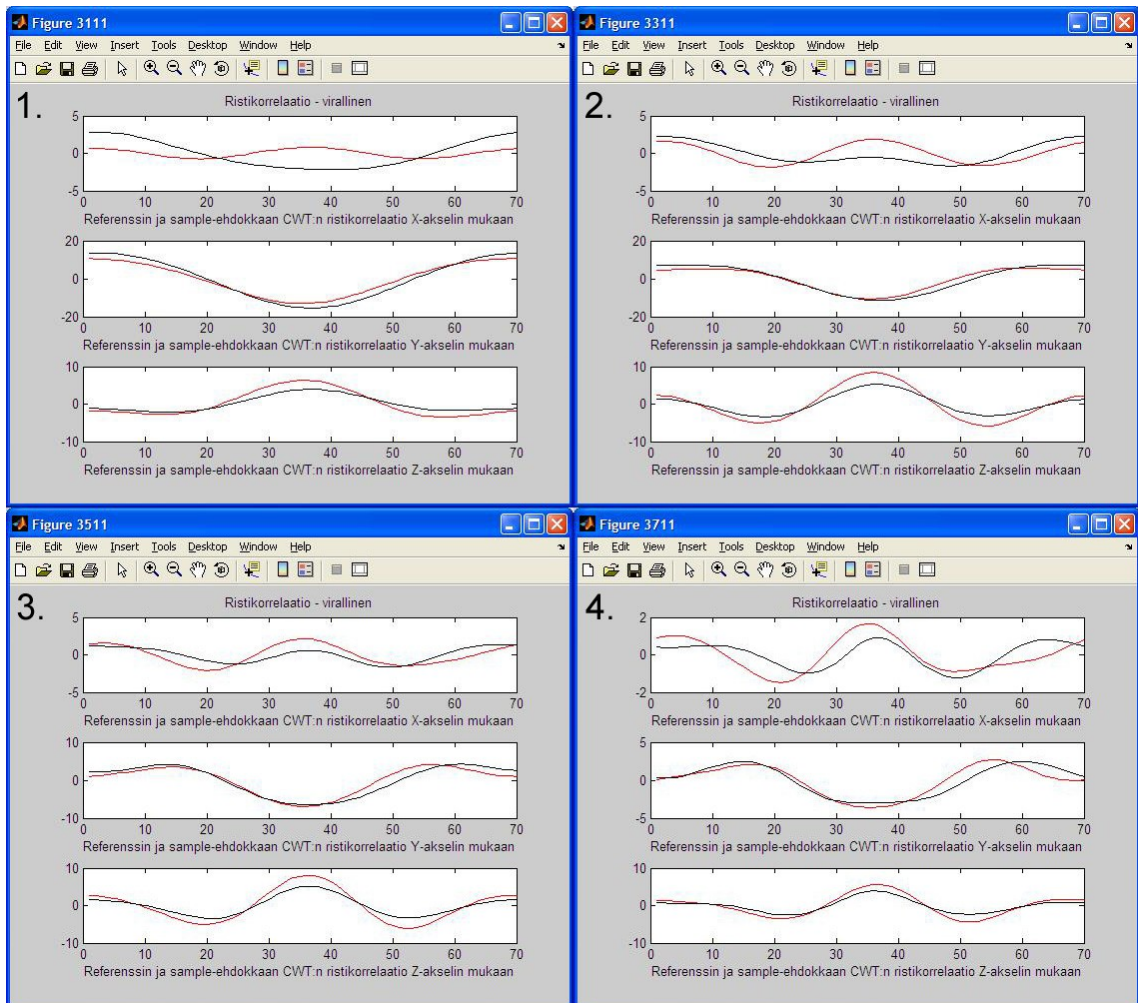


FIGURE 32. Cross-correlation sequence between reference signal and rejected windowed sample candidate

In part two (figure 33), reference signal will be scaled up and 50 % of the vectors within reference signal CWT-matrix will be gone through. As the first vector has been already used in 1:1 analysis, parts two and three will begin with vector 2, where (figure 33; 1.) cross-correlation for X-axis now goes to 0.3421 and for vector 8 (figure 33; 4.) it will be 0.3699. Analysis will stop on vector 8 and because X-axis still remains unidentified, the third part of the analysis will be done.

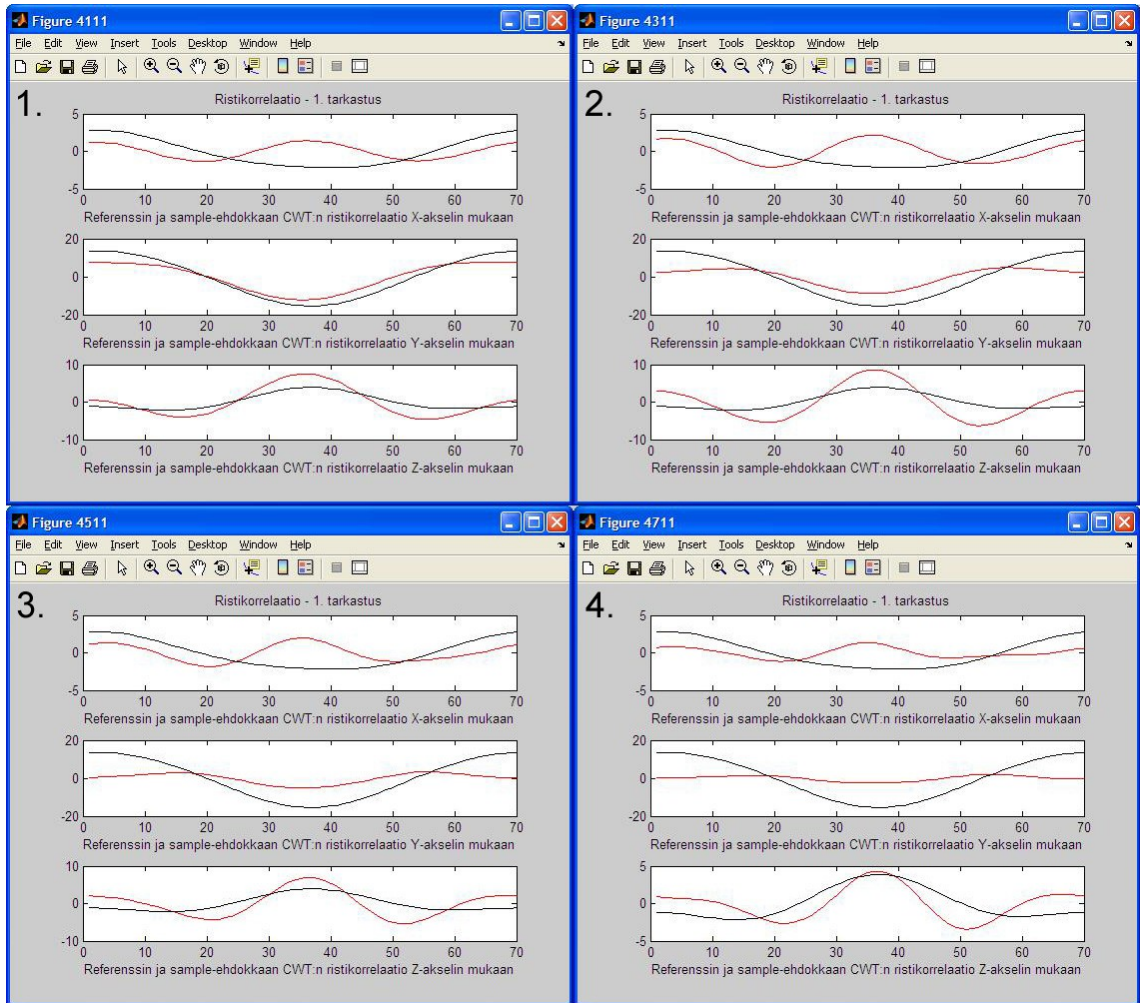


FIGURE 33. Cross-correlation sequence (scaling reference) between reference signal and rejected windowed sample candidate

Part three (figure 34) is similar than part two, except that now sample signal will be scaled up and 50 % of the vectors within sample signal CWT-matrix will be gone through. At vector 2 (figure 34; 1.), cross-correlation for X-axis goes to 0.3946 and for vector 8 (figure 34; 4.) it will be 0.6010. Analysis will stop on vector 8 and because X-axis remained unidentified after all the parts of the analysis, the eleventh sample candidate will not be recognized as a repetition at all. In this series, there are 12 sample candidates in total and 11 of them are recognized as repetitions. In comparison, first 10 repetitions were recognized by the old Matlab version and all 12 repetitions are recognized by sum signal based version of the new algorithm.

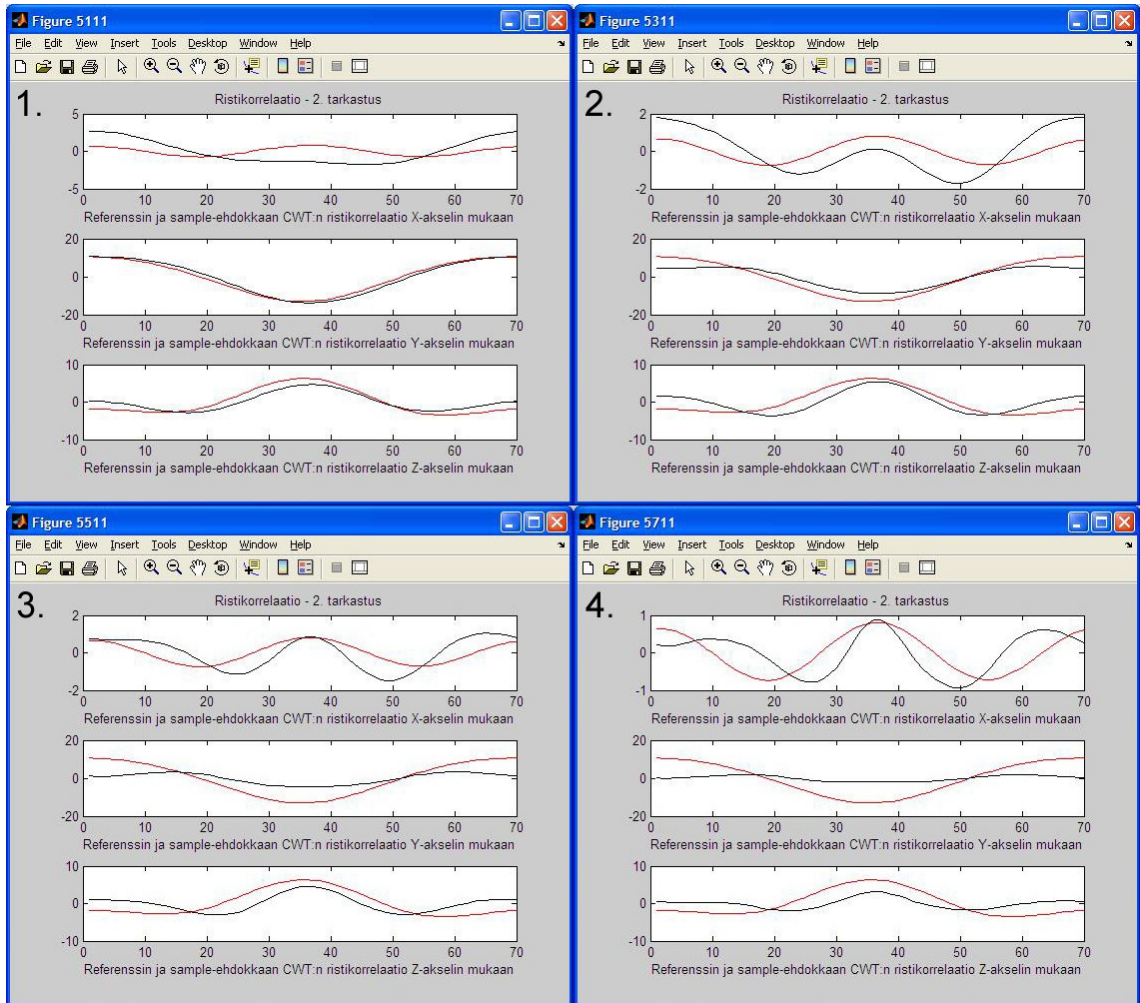


FIGURE 34. Cross-correlation sequence (scaling sample) between reference signal and rejected windowed sample candidate

8 RESULTS

In this chapter, extracts of the results for three different versions of the algorithm will be introduced. Firstly, the results for early C-version collected by Tiina Vaaraniemi. For the first Matlab version coded by Veli-Pekka Välimaa there are no available test results, but after coding the C-version, test results were collected by Arto Rajaniemi from his own testing data. In the last part of this chapter, extract of the early results for this new Matlab version are introduced.

8.1 Results for early C-version

Reliability of repetition counter in Musti-technology was tested in real gym environments including 11 female and 12 male testees (23 in total). Seven of them performed the test according to testing plan in Kuntoklubi OZ in Kempele and the remaining testees performed the testing in a gym located in Oulu University of Applied Sciences, School of Engineering. All the data was stored and performances were recorded with a camcorder. The functioning of repetition counter was compared with both stored data and recorded video. (Vaaraniemi 2011, 8.)

The selected movements for the test were squat and lap bow. Training history of the testees varied a lot. Some had no experience on the testing movements and others had both movements in their weekly training calendar. (Vaaraniemi 2011, 8.)

Squat was very balanced as a movement. From the starting position, the movement was well controlled and naturally more relaxing than returning, so Musti-meter registered a movement. Only 3,3 % of the repetitions remained without registering by the Musti-meter. (Vaaraniemi 2011, 10.)

Unlike squat, lap bow appeared to be a very unsure movement. Finding the rhythm and returning the pole to the starting position slower than lifting to the

chest appeared to be problems. Multi-meter registered only 88 % of the repetitions and about 10 % of these repetitions were to the wrong way registered repetitions. (Vaaraniemi 2011, 20.)

8.2 Results for new C-version

The algorithm written for C was tested with bench press movement. Reference movement has been done separately. In table 1, series of ten repetitions have been tried to do as successfully as possible. It can be seen that program has registered repetitions relatively well. (Rajaniemi 2011, 35.)

TABLE 1. Test results for bench press – Movements are tried to make as flawlessly as possible. (Rajaniemi 2011, 36.)

	Repetitions	Succesed	Failed
Series 1	10	10	0
Series 2	10	10	0
Series 3	10	9	1
Series 4	10	8	2
Series 5	10	10	0
Series 6	10	10	0
Series 7	10	9	1
Series 8	10	10	0
Series 9	10	10	0
Series 10	10	10	0

Table 2 shows the test results for series of ten repetitions, which have been tried to do wrong. It can be seen that program has recognized false movements relatively well. False movement must be relatively considerable, though. (Rajaniemi 2011, 36.)

TABLE 2. Test results for bench press – Movements are tried to make consciously wrong. (Rajaniemi 2011, 36.)

	Repetitions	Successed	Failed
Series 1	10	0	10
Series 2	10	2	8
Series 3	10	1	9
Series 4	10	1	9
Series 5	10	0	10
Series 6	10	1	9
Series 7	10	1	9
Series 8	10	0	10
Series 9	10	0	10
Series 10	10	1	9

8.3 Results for new Matlab version

For new Matlab version, it was possible to use test data from more than 40 testees. Because analyzing all the data would have been time-consuming, data from four testees was used. In 17th March 2011, tests were made by Musti-team with physiotherapy students from other department. These tests were observed by three students and a camcorder, while fourth student was performing the movements, so all the false repetitions were marked during the testing. Some of them were done consciously wrong. Also, the tests were performed by the both algorithm coders for Matlab version, but without observation. All these tests presented here are performed with biceps curl, so these four sets of results are comparable. In every following table in this chapter, letter F after repetition number indicates a fast repetition and letter S indicates a slow one. Green color indicates a successful repetition and red means failed by algorithm. All the tests are performed with both versions of the new algorithm: triaxial and sum. Windowing errors were occurred, so in those cases, all false repetitions were re-checked by analyzing them one by one.

Results in table 3 are for physiotherapy student A. 20 repetitions were recorded,

while the first one was used as a reference. Repetitions 12, 14, 17 and 19 were reported to be technically false. Only number 17 was registered as a false repetition by both versions of the new algorithm for this series.

TABLE 3. Test results for biceps curl – Physiotherapy student A

Repetition	Succession (triaxial)	Succession (sum)
1		
2		
3		
4		
5 (F)		
6 (F)		
7 (F)		
8 (F)		
9 (F)		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		

Results in table 4 are for physiotherapy student C. (Student B failed in all repetitions and student D had more false repetitions than she had planned to do.) Repetitions 7, 8, 9, 11, 12, 14, 16, 18 and 19 were reported to be technically false. Numbers 3, 5, 6, and 7 were registered as false repetitions by the triaxial version of new algorithm for this series.

TABLE 4. Test results for biceps curl – Physiotrerapy student C

Repetition	Succession (triaxial)	Succession (sum)
1		
2		
3		
4		
5 (F)		
6 (F)		
7 (F)		
8 (F)		
9 (F)		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		

Results in table 5 are for coder A. This test data has been used for introducing both versions of the Matlab-based algorithm seen in figures 7–10. This series was not observed by professionals, so it is not known how well these repetitions are performed. Repetitions 11 was reported to false by the triaxial version of new algorithm for this series as already stated in chapter 7.3. It was also stated that the earlier Matlab version recognized first 10 repetitions.

TABLE 5. Test results for biceps curl – Coder A

Repetition	Succession (triaxial)	Succession (sum)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11 (S)		
12 (S)		

Finally, results in table 6 are for coder B. This series was also not observed by professionals, so it is not known how well these repetitions are performed. Repetitions 8 and 12 were reported, surprisingly, to be false by the sum signal version of new algorithm.

TABLE 6. Test results for biceps curl – Coder B

Repetition	Succession (triaxial)	Succession (sum)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16 (S)		
17 (S)		

Because the new algorithm has not yet been coded to C-language and it has not been fine tuned, the results seen in this chapter are not final. In chapter 8.2 it can be seen that earlier Matlab version has proven to be much more accurate after fine tuning than it was expected, specially for detecting false repetitions. In those results it was not mentioned if any of the repetitions was made fast or slow, but at least it has been already a huge step forward.

From the results presented in this chapter it can be seen that slow repetitions are recognized very well by the new algorithm. For some reason, many of the fast repetitions were reported to be false by triaxial version of the new algorithm seen in table 4, while in table 3, all the fast repetitions have been reported to be clean. It was planned with physiotherapy students that pace for doing movements would temporary go quicker for repetitions 5–9, so personal style switching the pace may explain a lot. Confirmed false repetitions were reported to be false by the new algorithm only occasionally. At the moment it still seems

that the old algorithm is better for detecting false repetitions. It should be noted that there are no official test results observed by professionals available yet.

When this new algorithm will be written in C-language, fine tuning can be approached from many ways. Because gyroscope has been planned to be added to the device, the role of acceleration data for detecting false repetitions will be reduced. If gyroscope will be found very reliable for this purpose, acceleration data can be only needed for detecting only signal with heavy peaks and thus detecting false repetitions while gyroscope data could be used for detecting minor faults in technique, where acceleration data can not give the wanted accuracy. This is one aspect needed to be considered in the future development.

The second aspect for fine tuning is the technical aspect. Because the device does not have very much memory and CPU-power, it is needed to be checked how well the current Matlab-implementation meets the requirements set by the technical specifications of the device. The amount of voices, octaves and scales used in the analysis can be changed in order to get more precise analysis or lighter analysis with less vectors needed to go through. The percentage of the vectors to be used in the analysis can be changed as well. 50 % was the optimum at the moment when the new algorithm was developed. There was no need for more and using less would have given smaller chance for some repetitions to be detected. Less can be safely used if the algorithm needs more strictness. Acceptance value for cross-correlation is the last thing to be considered, but of course it does not have as much affection to technical aspect than the other above-mentioned have. The most important thing is: will it be triaxial or sum signal based version to be used in future C-version? It has the biggest significance to the performance.

The third aspect is the technical implementation itself. As mother wavelet, Mexican hat function was used in the current implementation because it is well localized in both time and frequency, but also easier to code for C than complex Morlet-wavelet. For example, Haar wavelet can be considered, because it is the simplest wavelet implementation available. The other thing is: would addition of

inverse wavelet transform give benefits over the current method? At least it would require more calculations from the device, but on the other hand it could lead to implementation where only one CWT-transform could be used, for example to reference signal and that would be the only signal needed to be scaled if inverse wavelet transform would easily give scaled versions of regular signal. The last possibility would be to use completely different method for checking similarity between reference and sample signal. That would be spectral analysis. This method was not tested in due to limited time and would require heavy modifications to the current implementation. In conclusion, the way of fine tuning of the new algorithm will be depended on the results given by the gyroscope data from the aspect of how well false repetitions can be estimated only from there and the requirements set by the technical specifications of the wristband.

9 DISCUSSION

Musti-technology has been developed for many years. At the early stages of the development, repetition counter was not included and the first C-version had a very simple method for counting repetitions. It was a huge step forward when Matlab version was developed in 2010. Because an other person had done the job and at that moment the documentation was limited to commented source code, it took much time for me to understand how the first Matlab version worked. It was also needed to list all the problems which the algorithm had at the time.

At first it was not clear how the algorithm would have to be developed further. Many ideas had came up by the Musti-team. Some of them were suggested from the viewpoint of continuing to improve the current method used in the new C-version and some of the ideas were completely new. The idea of using wavelets was the biggest of the latter. Because there were very many different ideas and it was important to keep the time used for the further development reasonable, it was finally decided to go for that. It was known that coding wavelet-based algorithm was not an easy task with C-language, but later the team was sure that it was possible to find a coder from the students who could do that. Later, it was also found that Stanford University had released C-based sources for WaveLab Toolbox functions, which could make the task even easier. From technical aspect, using wavelets was a reasonable choice at the perspective of performance.

At first, I felt the task a bit risky, but wavelets proven to be very versatile and they allowed for looking into many different possibilities for improving the algorithm. CWT was finally chosen, because it was used in similar purposes before. Keeping cross-correlation in the algorithm felt suitable and lightweight for embedded wristband system. It would have also been possible to look for something else like a method based by spectral analysis, but it was not tested mainly because there was not enough time.

As seen from the results, the biggest problems in the old algorithm has been solved. The new algorithm is able to recognize long repetitions and different scales helps recognizing the shape of the signal better. All the repetitions are still not beign recognized. Firstly, because this Matlab version is tuned to be as lightweight as possible and only a limited number of voices, octaves and scales have been used. Everything can not be tested in a short timespan reserved for the Thesis work, so it was needed to limit the time reserved for testing. Secondly, some of the repetitions are done technically wrong by the testees. Also, too noisy signal may be rejected automatically. This version of the algorithm can not see exactly all the mistakes the user may do during the weight lift as the old one did not either. When this algorithm will be coded to C, there are many possibilities for tweaking it. Windowing did not need a fix, because the new C-version stops storing data after each repetition, which have been already done. Gyroscope may be used to help with this task in the future, because with it, location information can be precisely known. Gyroscope can be possibly used for detecting false repetitions, too. It was not solved if inverse wavelet transform would make the algorithm better or not. It was encouraging to see that the algorithm works without it, because otherwise extra calculation operations would be needed.

This work was educational and helped me to improve Matlab-skills to a great extent. In the beginning it felt very difficult to start, but in the end this subject appeared to be interesting. Partly because I am interested in sports by general and partly because I have never worked in a project involving a large number or people aiming for a commercial product. I believe that this experience will be beneficial for the future.

10 CONCLUSION

The goal for this work was to improve Matlab version of the algorithm developed for Musti wristband. The biggest problem in the algorithm developed in 2010 was the recognition of slow repetitions. The chosen method, wavelets, gave a solution for this problem. When the new C-version came out, the windowing method for reference signal was proven to work well with the older Matlab-based algorithm, so changes were not needed for that part. New Matlab-algorithm required windowing method for sample signal, which was not done perfectly in Matlab, because the new C-version already included this feature implemented in similar way than windowing for reference signal. Because the team decided to add a gyroscope to the device, it can be used for detecting false movements and windowing sample signals can be possibly made to be even more reliable if needed.

From the current results it can be seen that the new algorithm can recognize slow repetitions better than the old version. Currently, from the results for new C-version, it can be seen that the old version of the algorithm is still better for detecting false movements. Tests were done in different manner, so the results are not fully comparable. The new algorithm needs to be coded to C, before it is possible to see its full potential.

REFERENCES

Aumala, Olli – Ihalainen, Heimo – Jokinen, Heikki – Kortelainen, Juha 1998. Mittaussignaalien käsittely. Tampere: Klingeldahl Paino Oy.

Billauer, Eli 2010. Matlab-language programmed peakdet.m-source code with comments. Available at: <http://www.billauer.co.il/peakdet.html>. Date of retrieval 12.4.2011.

Daubechies, Ingrid 1992. Ten Lectures on Wavelets. Montpellier: Capital City Press.

Hannula, Manne – Leskinen, Sami – Linnamo, Vesa – Rontu, Jari-Pekka – Salmi, Jukka 2007. 1-RM bench press performance estimated with accelerometer method. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Jyväskylä: University of Jyväskylä.

Hannula, Manne 2011. Doctor, Oulu University of Applied Sciences. Interview 31.1.2011.

Hinkula, Henry 2011. Project coordinator, Oulu University of Applied Sciences. Interview 9.2.2011.

Ilić, Siniša S. 2007. Detection of the Left Bundle Branch Block in Continuous Wavelet Transform of ECG Signal. Available at: http://www.ktu.lt/lt/mokslas/zurnalai/elektros_z/z74/07_T121_%20Detection_of_LBBB_in_CWT_of_ECG_signal_Sinisa_Ilic.pdf. Date of retrieval 17.3.2011.

Inkinen, Pentti – Tuohi, Jukka 1999. Momentti 1 – Insinöörifysiikka. Keuruu: Otavan Kirjapaino Oy.

Keskinen, Kari L. – Häkkinen, Keijo – Kallinen, Mauri 2004. Kuntotestauksen käsikirja. Tampere: Tammer-Paino Oy.

Lammela, Jaana 2007. Algoritmi penkkipunnerruksen maksimituloksen arvioimiseksi. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Bachelor's Thesis.

Leiviskä, Petteri 2009. Maksimivoimamittalaitteen rakentaminen ja testaus. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Bachelor's Thesis.

The MathWorks, Inc. 2011. MATLAB - The Language Of Technical Computing. Available at: <http://www.mathworks.se/products/matlab/>. Date of retrieval 25.4.2011.

Mero, Antti – Nummela, Ari – Keskinen, Kari – Häkkinen, Keijo 2004. Urheiluvalmennus. Jyväskylä: VK-Kustannus Oy.

Rajaniemi, Arto 2011. Maksimivoimamittarin algoritmikehitys. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Bachelor's Thesis.

Similä, Ville 2010a. Maksimivoimamittari – käyttöohjeet. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center.

Similä, Ville 2010b. MST – konseptin kehitys ja testaus. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Bachelor's Thesis.

Stanford University. WAVELAB 850. Available at: <http://www-stat.stanford.edu/~wavelab/>. Date of retrieval 20.4.2011.

Strang, Gilbert – Nguyen, Truong 1996. Wavelets and Filter Banks. Wellesley: Wellesley-Cambridge Press.

Texas Instruments Embedded Processors Wiki. EZ430-Chronos. Available at: <http://processors.wiki.ti.com/index.php?title=EZ430-Chronos>. Date of retrieval 6.2.2011.

Vaaranieni, Tiina 2011. Maksimivoimamittarin konseptin testaus loppukäyttäjien kanssa. Oulu: Oulu University of Applied Sciences, Medical Engineering R&D center. Bachelor's Thesis.

VTI Technologies 2010. Product Family Specification – CMA3000-D0X Series 3-axis accelerometer. Available at: http://www.vti.fi/midcom-serveattachmentguid-1e0126c994791f0126c11e09138e559c13fb5afb5af/cma3000-d0x_product_family_specification_8281000a.04.pdf. Date of retrieval 17.2.2011.

Välimaa, Veli-Pekka 2010. Matlab-language programmed Chronos_toistolaskuri.m-source code with comments. Available at: Oulu University of Applied Sciences, Medical Engineering R&D center.

Välimaa, Veli-Pekka 2011. Student, Oulu University of Applied Sciences. Meeting 14.2.2011.

APPENDICES

Appendix 1. Block diagram of the old Matlab version

