ZHAW School of Engineering

Projektarbeit

IT

Title

Authors:
Linda Helen Boedi
Valentin Bossi

Supervisor:
Hans-Joachim Gelke

December 7, 2017

Acknowledgement

Declaration of Autonomy

The undersigned certify with their signature that the work has been written independently and put into written form, that the involvement of other persons has been limited to consulting and proofreading, and that all documents and guarantors used are listed.

Place and Date	Linda Helen Bödi
Place and Date	Valentin Bossi

Abstract

Every mechanical watch must be checked for accuracy both during manufacture and use. Most of the equipment used for this purpose is expensive. In this work it is determined whether a Raspberry Pi and a Raspberry Pi camera can be turned into a cheaper measuring device that can compete with the accuracy of a professional device. This camera also differs from conventional measuring devices in the type of measurement. Almost all devices on the market perform acoustic measurements and a few complement this measurement with an optical measurement by means of a laser beam. In this work a new approach is used; the measurement is carried out completely optically, but without laser beam, but completely by means of image processing.
The following questions should be clarified in the work: Is a complete optical measurement by means of image processing possible? How accurate will these measurements be and can they keep up with the results of professional equipment? What are possible sources of error? How can the measurement be further improved?
In a first step, possible methods were then evaluated in order to carry out the optical measurement of the frequency of the balance of a watch. A closer look was taken at the OpenCV library, the gstreamer, a one-to-one comparison of images, tracking a specific pixel and the Picamera library. Since the Picamera library has a very well-founded documentation and offers many of the functionalities needed for this work, the decision was made in the end for this library. In order to better understand the motion vectors, various experiments were carried out, whereby the camera settings were also improved. After this basis had been established, a first rough calculation of the frequency was done by hand in Excel and due to a promising result the implementation of a Python program was carried out. During the course of the work, after several measurements with professional instruments, it was found that the watch reacts very sensitively to temperatures, runs very inaccurately and therefore it is difficult to get good reference measurements. Therefore, a comparison was made with a blinking LED which flashes in time to show how the test clock should ideally oscillate back and forth. The

measurements of the flashing LED showed that the determination of the frequency by means of image processing works very well and that sufficient good results can be achieved. In order to improve the results even further, it is possible, for example, to investigate which camera settings are best suited for the measurement.

Contents

Acknowledgement i

Declaration of Autonomy ii

Abstract iii

1 Motivation 1

2 Fundamentals 2
2.1 Competitor Analysis . . . 2
2.1.1 Optical versus acoustical measurement . . . 2
2.1.1.1 Acoustical measurement . . . 2
2.1.1.2 Optical measurement . . . 2
2.1.2 Companies . . . 3
2.1.2.1 Witschi - WisioScope S . . . 3
2.1.2.2 Lepsi - WatchScope/WatchAnalyzer . . . 3
2.1.2.3 Greiner Vibrograf - Compact 900 . . . 3
2.2 Camera module . . . 4
2.2.1 Exposure time . . . 6
2.2.1.1 Minimal exposure time . . . 6
2.2.1.2 Maximum framerate . . . 6
2.2.1.3 Maximum exposure time . . . 7
2.2.2 Hardware limits . . . 7
2.3 Mechanical watches . . . 7
2.3.1 Operation of a mechanical watch . . . 7
2.3.2 Oscillation of the balance wheel . . . 9
2.3.2.1 Period . . . 9
2.3.2.2 Half a period . . . 10
2.3.3 Number of impacts . . . 10
2.3.4 Test watch . . . 11

3 Technologies **12**
3.1 Evaluation of technologies 12
3.1.1 OpenCV 12
3.1.2 Gstreamer 12
3.1.3 Picamera 12
3.1.4 Frame analysis versus given GPU motion vectors . . . 12
3.1.5 Frame analysis via OpenCV 13
3.1.6 Motion vectors from high level library picamera 13
3.1.7 Conclusions 14

4 Experimental approaches **15**
4.1 Test setup 15
4.2 Motion vectors 16
4.2.1 Gesture detection 16
4.2.2 Records of balance wheel 17
4.2.3 Animation of motion vectors 17
4.2.4 Motion vectors visualized in a frame 19
4.2.5 Analysis of motion vectors in excel 19
4.2.6 Influences of light, ANGLE? and camera parameters . 20
4.3 Crude calculation of frequency 21
4.4 Verification of accuracy 24

5 Calculation of the frequency **26**
5.1 Implementation of the algorithms 26
5.1.1 Calculation of the frequency by counting minima with upper limit 26
5.1.2 Calculation of the frequency by counting minima with stepsize 27

6 Results **28**
6.1 Measurements 28
6.2 Calculation of rate deviation from frequency 29
6.3 Evaluation 29

7 Discussion **30**
7.1 Difficulties 30
7.2 Outlook 30

Glossary **31**

Chapter 1

Motivation

Today's watch technology requires measuring instruments to perform a wide variety of measurements and analyses on almost all types of watches with great precision and reliability. Mechanical watches must be checked in production and service for their rate deviation. There are different techniques to measure this deviation; on one hand acoustically and optically or only acoustically. The aim of this work is to develop a handy device that can visually measure the rate deviation using inexpensive and simple means. To achieve this, an existing camera consisting of the best-known single-board computer Raspberry Pi and the corresponding camera from Raspberry Pi is used. The aim of this work is to use image processing to measure and analyse the rate deviation of mechanical clocks. The vision of this project is to be able to perform real-time measurements as established providers do with a much cheaper device.

Chapter 2

Fundamentals

2.1 Competitor Analysis

2.1.1 Optical versus acoustical measurement

To measure the frequency of the balance wheel of a mechanic clock there exist mainly two methods on the market, acoustical and optical measurement. The common used method is by analyzing the beat noises of the lever escapement. More expensive devices use both acoustical and optical feedback. In order to be able to classify the rate deviation well enough, it is usually measured in different positions of the watch.

2.1.1.1 Acoustical measurement

The first used method to get the frequency measurement of the balance wheel was by using a vibrograph was used. For every "tick" of the clock a line was drawn onto a ongoing strip of paper. Out of the distance between of these lines the frequency was calculated [**Zeitwaage**]. But as this method isn't the most accurate other techniques are used nowadays. Modern watch timing machines use an oscillating quartz crystal as comparison for the frequency of the balance wheel [**Zeitwaage**]. The beat noises of the lever escapement are recorded and amplified with a microphone whereat unwanted background noises need to be filtered out.

2.1.1.2 Optical measurement

About hundred years later, optical watch timing machines joined the acoustical ones. There are barely devices on the market, which use only optical measurement, but several companies started to combine their acoustical with

optical metering. One way to scale the frequency of the balance wheel optically is by using a laser. The beam will be periodically interrupted by the balance wheel and thus the frequency can be calculated [**Lombardi2011**]. In this paper the frequency should be quantified optically by using image processing. There is hardly to none company out there, which uses the last technique for measuring the frequency.

Comparison acoustical and optical measurement		
	acoustical	**optical**
since	around 19th century (2)	end of 20th century (2)
accuracy	0.1 s/d	0.1 s/d
advantages	most experience (exists for about 200 years)	measurement can be done anywhere
disadvantages	background noises need to be filtered out	

Table 2.1: Comparison of acoustical and optical measurement

2.1.2 Companies

2.1.2.1 Witschi - WisioScope S

WisioScope S tests mechanical watches acoustically and optically. The measurement is done parallel and in this way is more accurate as both signals are used for the calculation of the frequency. The optical metering is done using a laser and lighting, a camera helps to adjust the watch properly. The costs of their product are: CHF 10450.-

2.1.2.2 Lepsi - WatchScope/WatchAnalyzer

The WatchScope and WatchAnalyzer from Lepsi are especially for lovers and not really for production. Both products are Swiss technologies. Both devices only work with an acoustic input, with measurements of either a few seconds or up to 24 hours. All data can then be queried with the smartphone. WatchScope is the smaller, more convenient version. With the WatchAnalyzer you can easily measure the rate deviation of the watch in several positions. The prices for this product are CHF 369.00, resp. CHF 929.-

2.1.2.3 Greiner Vibrograf - Compact 900

The Compact 900 measures only the beat noises of the ever escapement. The price is CHF 4070.-

The following table summarizes all important information for the three mentioned professional devices.

Measurement rate deviation			
	Witschi - WisioScope S	**Lepsi - Watch-Scope/WatchAnalyzer**	**Greiner Vibrograf - Compact 900**
scope	+/- 999.9 s/d	+/- 1000 s/d	+/- 1000 s/d
resolution	0.1 s/d	0.1 s/d	0.1 s/d
price	CHF 10450.-	CHF 369.-, resp. CHF 929.-	CHF 4070.-

Table 2.2: Measurement rate deviation with different professional devices

2.2 Camera module

The documentation Readthedocs for the picamera library [**ReadTheDocsPicamera**] provides a good introduction to the operation of the Raspberry Pi with camera. This chapter will briefly summarize this introduction to better understand the hardware and operation of the Raspberry Pi camera.

The camera module of the Pi is basically a mobile phone camera module. Among other things, it uses a rolling shutter to take pictures. The pixel values of a frame are not captured completely at once, but are read line by line.

However, the sensor is configured via the registers with the number of lines to be read and a corresponding time. The sensor reads the lines and pushes the data with the configured speed to the Raspberry Pi.

This keeps the readout time of each line constant. However, the CPU does not speak directly to the camera, but processing takes place on the GPU (VideoCore IV) of the Raspberry Pi, which operates its own real-time operating system (VCOS).

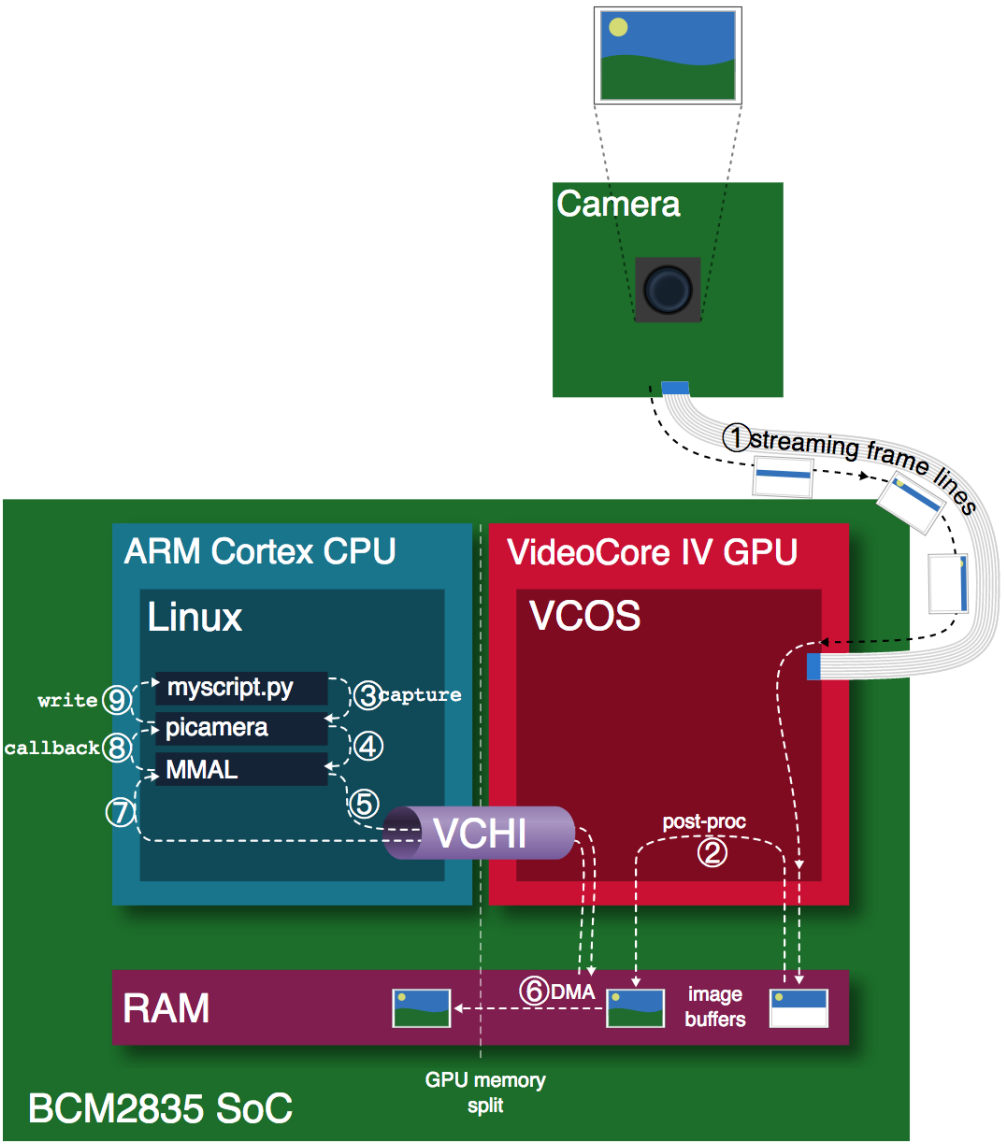

Figure 2.1: Camera architecture. [**ReadTheDocsPicamera**]

The illustration above illustrates the processing flow of a frame, with the associated steps explained in the following section.

1. The camera's sensor is configured and continuously streams frame lines to the GPU.

2. The GPU builds complete frame buffers from these lines and performs the post-processing of these buffers.

3. Meanwhile, myscript.py makes a capture call with the picamera on the CPU.
4. The picamera library uses the MMAL API to meet this requirement.
5. The MMAL API sends a message via VCHI requesting a frame capture.
6. The GPU then initiates a DMA transmission of the next full frame from its RAM portion to the CPU portion.
7. Finally, the GPU sends back a message via VCHI that the capture is complete.
8. This causes an MMAL thread to trigger a callback in the picamera library, which in turn retrieves the frame.
9. Last picamera calls "write" on the output object provided by myscript.py.

2.2.1 Exposure time

The camera sensor detects how many photons hit the sensor elements, because the more impact, the more they increase their counter values. The sensor can perform exactly two operations; reset a set of elements or read a set of elements.

2.2.1.1 Minimal exposure time

Reading out a series of elements takes a certain amount of time, thus there is a limit to the minimum exposure time. Assuming one has 500 lines on a sensor and reading each line takes at least 20ns, then it will take at least $500 * 20\text{ns} = 10\text{ms}$ to read a full screen.

2.2.1.2 Maximum framerate

The frame rate is the number of frames the camera can capture per second. The exposure time determines the maximum number of images that can be taken in a given time. Assuming it takes 10ms to read a complete image, then no more than $\frac{1\text{s}}{10\text{ms}} = \frac{1\text{s}}{0.01\text{s}} = 100$ images in one second.

So it is valid:

$$\frac{1\text{s}}{\text{min exposure time in s}} = \text{max framerate in fps.}$$

The lower the minimum exposure time, the higher the maximum frame rate and vice versa.

2.2.1.3 Maximum exposure time

To maximize shutter speed, the framerate needs to be reduced.

Therefore the following is valid :

$$\frac{1\text{s}}{\text{min framerate in fps}} = \text{max exposure time in s}$$

2.2.2 Hardware limits

- The maximum resolution for MJPEG recording is partially dependent on the GPU memory.
- The maximum horizontal resolution for the standard H264 recording is 1920 (this is a limitation of the H264 block in the GPU).
- The maximum frame rate of the camera depends on several factors. With overclocking 120fps can be achieved, but 90fps is the maximum supported frame rate.

2.3 Mechanical watches

2.3.1 Operation of a mechanical watch

If a mechanical watch is wound up over the crown, the locking wheel (3) is moved by the winding shaft (1) and winding wheels (2). It turns the spring core (4) and pulls the tension spring in the mainspring barrel (5). The spring transmits the stored energy via the external toothed mainspring barrel (5) to the minute wheel (6), third wheel (7) and fourth wheel (8).

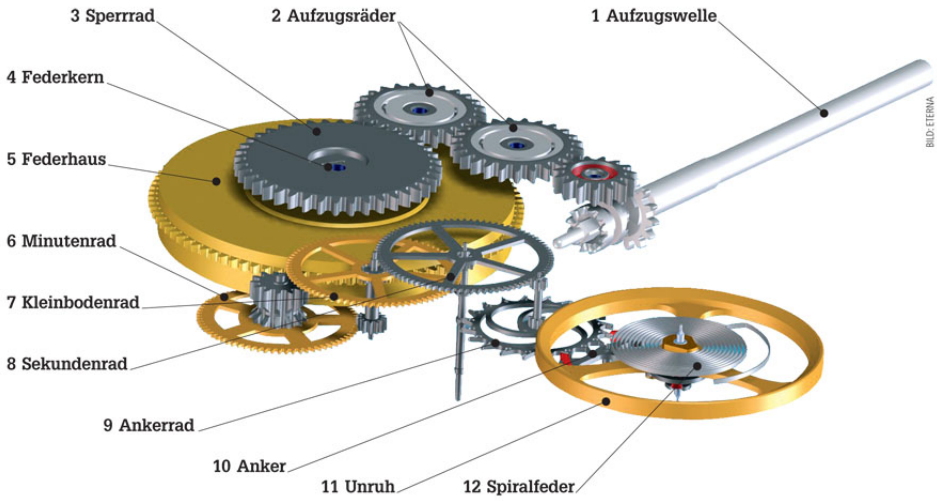

Figure 2.2: Operation of a mechanical watch [**Uhrwerk**]

The so-called escapement ensures that the gear train runs at the correct speed. The fourth wheel (8) drives the escape wheel (9); it gives an impulse to the pallet fork (10) which it passes on to the balance wheel (11), after which the armature, which moves back and forth like a seesaw, blocks the escape wheel. The balance wheel rotates, but is then retracted by the hairspring (12). It moves the armature back, which now releases the armature wheel again a little bit, which gives the armature with its rotation the next impulse. Due to the sudden braking and acceleration of the escapement wheel, the second hand of a mechanical watch also moves stepwise. [**Uhrwerk**]

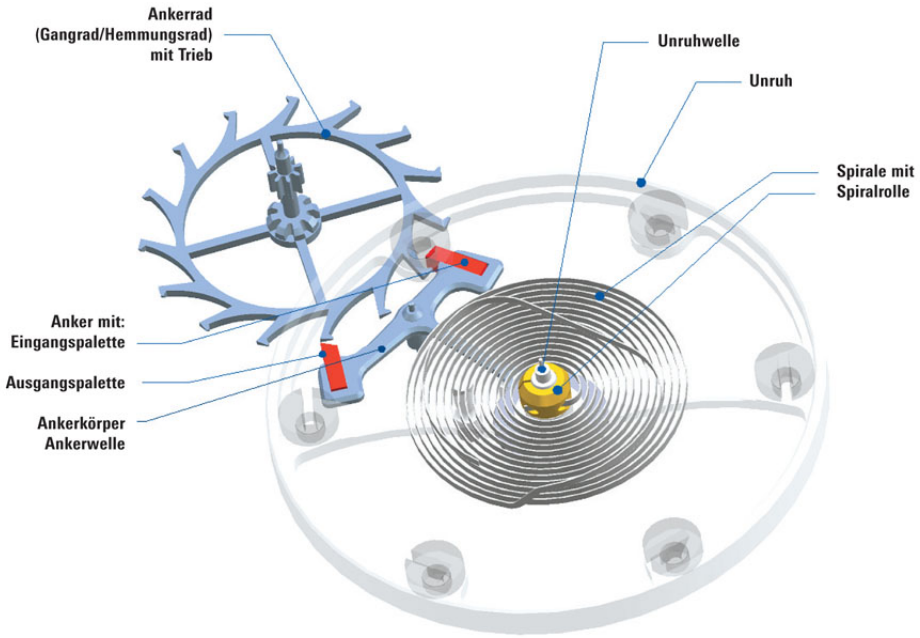

Figure 2.3: lever escapement and balance wheel [**Uhrwerk**]

The balance wheel is the heart of the oscillation system. It generates a time-defined movement, which in turn is transferred to the gear train and passed on to the watch hands. The occurring error, if the balance wheel does not run accurate, is called rate deviation. The deviation can be caused by different external and internal matters. Internal motives often are wear, resp. erosion or dirt. Whereas there are a lot more external causes as changes in temperature, air pressure or magnetism.

2.3.2 Oscillation of the balance wheel

2.3.2.1 Period

The period of the balance wheel is the path of a point from one turning point to the other and back (A - B - A). [**Witschi_basics**]

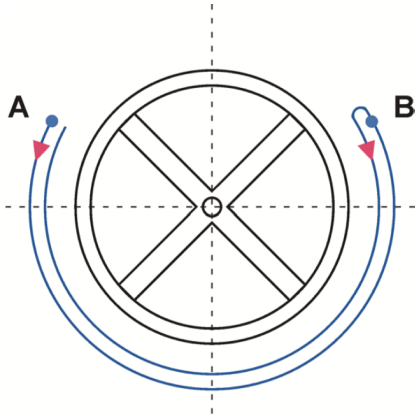

Figure 2.4: period of the balance wheel [**Witschi_basics**]

2.3.2.2 Half a period

Half of a period of the balance wheel is the path of a point from one turning point to the other (A - B). [**Witschi_basics**]

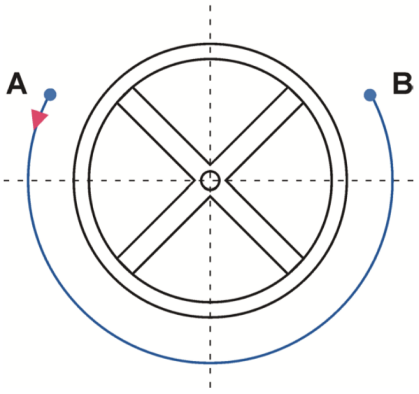

Figure 2.5: half of a period of the balance wheel [**Witschi_basics**]

2.3.3 Number of impacts

Since the rate deviation has been calculated in seconds per day and the number of strokes per hour, i. e. the number of audible impulses (beats) of a two-armed pallet fork of a mechanical watch per hour, is known, these two

Number of strokes [1/h]	**Stroke duration** [s]	**Oscillation number** [1/h]	**Period duration** [s]	**Frequency** [Hz]
18'000	0.200	9'000	0.400	2.50
19'800	0.182	9'900	0.364	2.74
21'600	0.166	10'800	0.333	3.00
28'800	0.125	14'400	0.250	4.00
36'000	0.100	18'000	0.200	5.00

Table 2.3: Number of strokes, period duration and frequency of the balance of automatic wristwatches [**Krug1987**]

quantities must first be brought to a unit. The relationship between the beat number (n*) and frequency (f) is as follows [**Krug1987**]:

$$n^* = 2f * 3600$$

and therefore:

$$f = \frac{n^*}{7200}$$

2.3.4 Test watch

At first some general information to the test watch, which was used.

The test watch is a watch of the caliber ETA C01.211 it is a chronograph, with a diameter of 31 centimeters. It strokes 21600 times per hour and therefore has a frequency of 3 Hz. Additionally the watch has a power reserve of 43 hours. [**Caliber**]

In order to obtain an exact reference value of the frequency of the balance wheel of the test watch, multiple acoustic measurements were carried out with professional measuring devices, the Greiner Vibrograf - Compact 900 and Watch Expert II from Witschi, and the rate deviation was determined.

In order to calculate the frequency from the rate deviation of the balance wheel, first some transformations had to be carried out.

Chapter 3

Technologies

was für uns wichtig ist bei einem tool/library. wie ein bild aufgebaut ist zeigen (rgb etc)

- stillstand erkennen
- merkmal verfolgen
- geschwindigkeit berechnen
- timestamps als wichtige komponente, da für uns die bewegung über die zeit interessiert

3.1 Evaluation of technologies

3.1.1 OpenCV

3.1.2 Gstreamer

3.1.3 Picamera

The Picamera library is an open source project in the programming language Python. The library is an interface to the MMAL —written in C— which in turn is an interface to the firmware of the GPU. This means that the camera can be accessed using a simple programming language such as Python. [**ReadTheDocsPicamera**]

3.1.4 Frame analysis versus given GPU motion vectors

One approach is to compare each image with the subsequent image and to mark the displacements with motion vectors. This would mean a considerable

computational effort. Such motion vectors are already made by the GPU, in order to compress videos in the H.264 codec.

3.1.5 Frame analysis via OpenCV

With the library OpenCV it is possible to determine the optical flow of an image sequence.

The optical flow represents the vector field of the velocities of the points of an image sequence. It is a useful representation of motion information in image processing. The local optical flow is an estimate of patterns in an image in the vicinity of a viewed pixel.

One possibility would be to follow the course of a velocity vector and thereby calculate a period of the frequency of the balance wheel. The calculation of the flow at selected points is also called feature point tracking. Another approach would be to calculate the frequency based on the velocity obtained from the optical flow, since it is more or less a circular movement.

One difficulty with this procedure is that one has to commit to certain pixels that one examines. This can cause problems if the clock is placed differently, because wrong points (worst-case scenario: points in which movement never takes place) can cause problems for the calculation.

The OpenCV library uses the Lucas-Kanade method to calculate the optical flow. The Lucas-Kanade method also assumes that the flow in the local environment of the pixel for which the flow is intended is the same. So more or less a whole set of pixels is considered. The flow can thus be determined by calculating the derivatives (= gradients).

One characteristic of this method is that it does not provide a dense flow, i. e. the flow information disappears quickly with the distance from the edges of the image. The advantage of the method is its relative robustness against noise and smaller defects in the image.

3.1.6 Motion vectors from high level library picamera

The picamera library is capable of storing all motion vector estimations of the h.264 encoder in an object. This object with all motion vectors for all frames can be later accessed and used. The motion data is at the level of macro-blocks where the values are 4-bytes long and consist of a signed 1-byte x vector, a signed 1-byte y vector and an unsigned 2-byte SAD (Sum of Absolute Differences) value. All motion data values can be easily accessed frame by frame. [**ReadTheDocsPicamera**]

The sum of absolute differences is a positive number resulting from the formation of the difference between two digital images. It serves as a measure

of the difference between two images.

The SAD is obtained by subtracting the color values of the images pixel by pixel from each other and adding them up by amount.

The advantage of the Picamera library is that compared to the possibilities of the OpenCV library, the information about the motion vectors is available faster, since this library already has an internal object (motion-data), which stores all necessary information as well as the SAD value.

3.1.7 Conclusions

Chapter 4

Experimental approaches

- ev begründen, wieso wir picamera lib gewählt haben?
- ev nicht hier erwähnen, aber iwo einbetten, dass wir die MV's gewählt haben, weil sie auch universell sind. rgb zum beispiel ist abhängig von der farbe und weil die uhren verschiedene metalle haben, wäre es dann nicht möglich mit rgb zu messen. nach mustern im bild suchen bzw nach der unruhe suchen und so objekttracking machen ist auch nicht sinnvoll, weil es sehr viele verschiedene unruhen gibt. (ev bilder von verschiedenen unruhen?)

4.1 Test setup

- bild ausarbeiten
- kurze beschreibung dazu

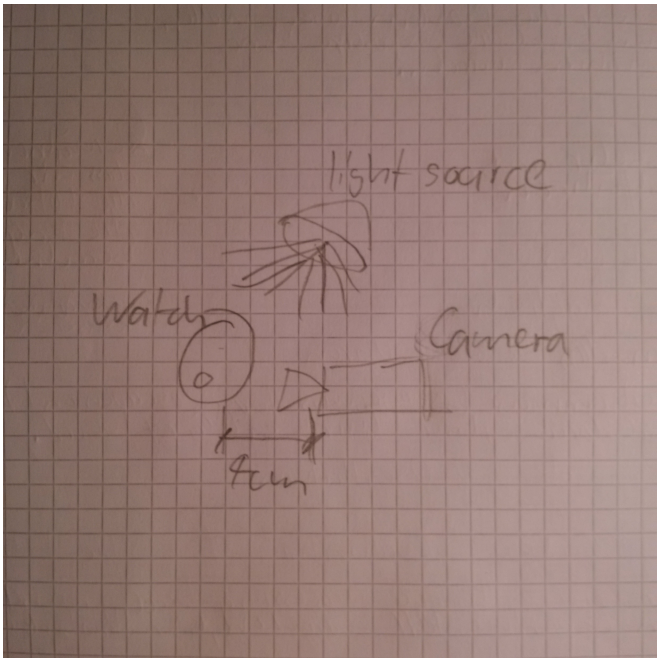

Figure 4.1: Test setup for records

4.2 Motion vectors

- kurz erklären wie mv funktionieren-> makro blocks. mv -> wird für video kompression genutzt
- erklären, dass die idee einen punkt zu verfolgen ist und dann den stillstand zu erfassen, um die periode bzw die frequenz herauszufinden.
- ev erwähnen, dass so ev geschwindigkeitsunterschiede festgestellt werden könnten, dachten wir...

4.2.1 Gesture detection

- wir haben zuerst getestet, ob wir mit der kamera real time erkennen können, ob sich ein objekt vor der kamera nach links/rechts/runter/rauf bewegt
- das konnten wir und sind weitergegangen und wollten das an der uhr testen und visualisieren (animation of motion vectors)

4.2.2 Records of balance wheel

- dann haben wir aufnahmen der unruhe gemacht und versucht die ergebnisse zu interpretieren

4.2.3 Animation of motion vectors

After deciding which library to use, some experimental sessions took place to get a better understanding of the so called motion data values or motion vectors. With the picamera library it is easy to access these motion data values so the hard part wasn't to get the values but to understand how they are constructed. The motion data object consists of multiple values; a signed 1-byte x vector, a signed 1-byte y vector and and unsigned 2-byte SAD value for each macro-block of a frame.

In a first step all SAD values were animated with colors to get a better understanding of how good the information they keep is and how as well as if they can be used to calculate the frequency of the movement of the balance wheel.

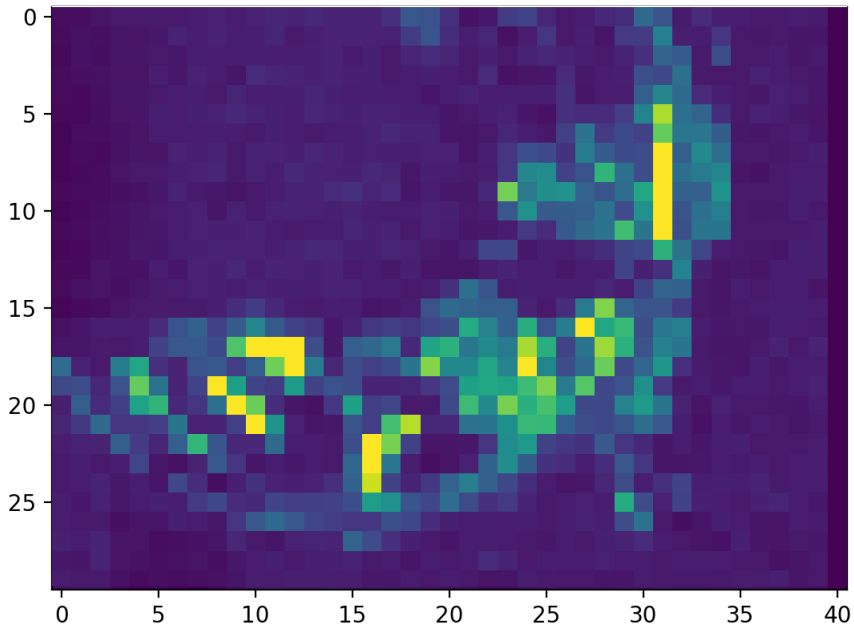

Figure 4.2: Snapshot of the animation of the SAD values, the lighter the color the higher the value of the SAD

Further the x and y vector were analyzed and also animated. But as

those vectors itself are hard to use and only give partial information (y-vector: up or down, x-vector: right or left), the hypotenuses were calculated with Pythagoras' theorem and displayed similar to the SAD values. The hypotenuses give information about the amount as well as the direction of the motion.

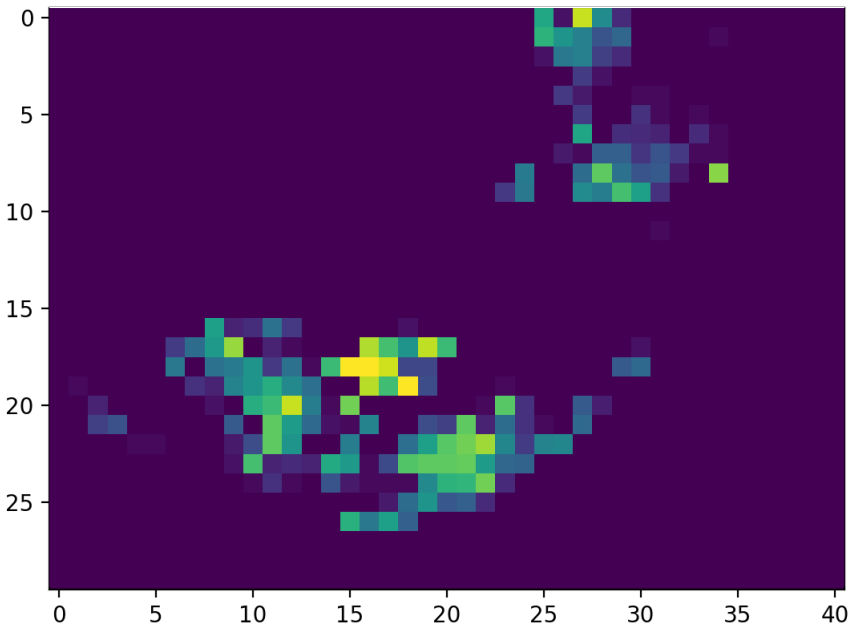

Figure 4.3: Snapshot of the animation of the hypotenuses, the lighter the color the higher the value of the hypotenuse

Another experimental approach was displaying the motion vectors as arrows directly in the video. This approach was helpful to get a better understanding under which conditions the best result of the motion vectors is reached.

4.2.4 Motion vectors visualized in a frame

Codecvisa was used to display the motion vectors of a frame. The result is shown in Figure 3.

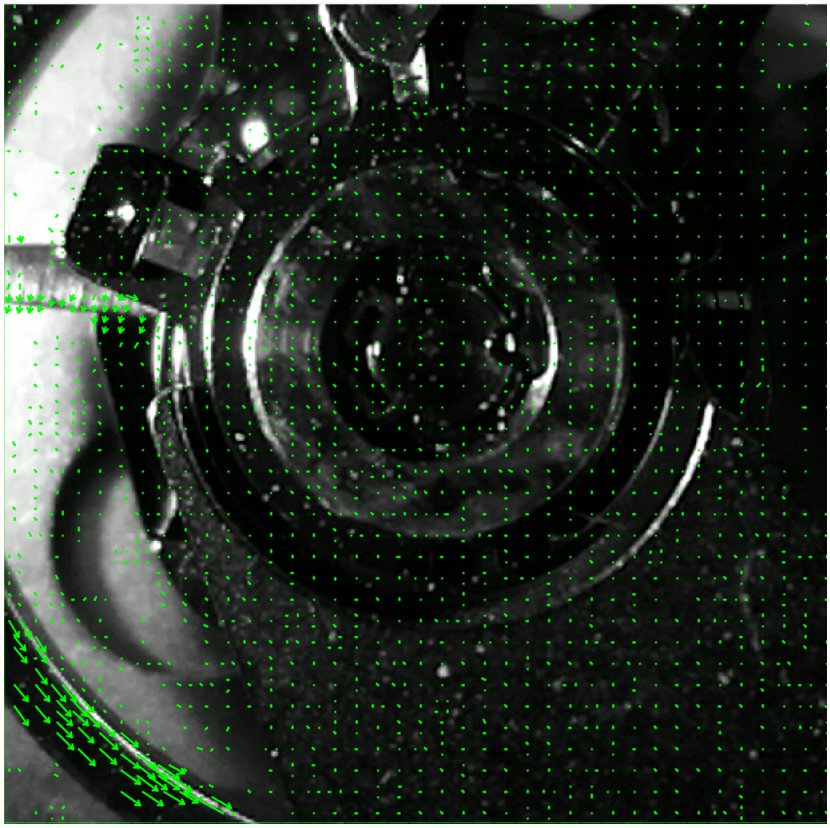

Figure 4.4: Motion vectors drawn in one frame

4.2.5 Analysis of motion vectors in excel

- ev bild, wie das gemeint ist...

Another approach used Excel was to determine whether a motion vector from one frame is the predecessor of the motion vector in the next frame. The hope was that a moving pixel could be followed and therefore the location of the pixel would always have been known. The result was sobering because one frame and motion vector cannot be associated with another frame and motion vector. This experiment looked at the idea of the optical flux with the movement vectors of the MMAL (Multi-Media Abstraction Layer).

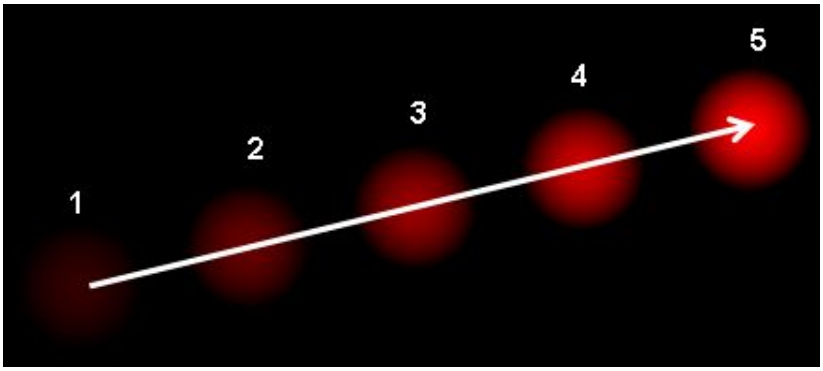

Figure 4.5: It shows a ball moving in 5 consecutive frames. The arrow shows its displacement vector.

4.2.6 Influences of light, ANGLE? and camera parameters

- mehr licht ergab deutlichere messwerte
- bewegungsunschärfe (ev bild) minimieren hat auch deutlichere messwerte ergeben, braucht aber sehr strake lichtquelle(mit smartphone)
- anderer winkel hat keine auswirkungen...

Testing with different settings has shown that the results can be improved. For example, the black and white shots are the better choice. Reducing the shutter speeds can reduce motion blur, which also contributes to a better result. The consequence of the shorter exposure time is that the light source must be very strong and close to the subject, otherwise the image is too dark and has a negative effect on the result.

4.3 Crude calculation of frequency

With the help of the motion vectors and the corresponding time stamps per frame, a rough calculation of the frequency can be made. All x, y or SAD values contained in the image are summed up. If the sum of all values is small, it is very likely that the balance wheel is at a standstill before turning back again. For small values, the time stamp is registered and the difference to the next standstill is calculated. Since the balance wheel triggers half a oscillation, it has moved from one stop to the next half point. In order to roughly calculate the frequency, an Excel sheet was created in which all the accumulated x- and y-values were entered and the standstills of the balance wheels were determined by eye. The half-oscillations and the frequency were calculated from the time intervals of the standstills. Afterwards, the average of all calculated frequencies was determined and it was recognized that the results with this method are good enough to determine the frequency and thus the rate deviation.

xSumAbs	ySumAbs	timestamp			timestamp stillstand	half period	period	hertz	hertz average
0	0	None			88645	166210	-		3.032792904
3050	2704	11081			254855	166209	332419	3.00825163	
2540	1770	22161			421064	177290	-		
1106	1464	33242			598354	155129	332419	3.00825163	
3422	2870	44324			753483	177289	-		
3070	2352	55403			930772	166210	343499	2.91121663	
1924	1658	66484			1096982	166210	-		
1012	1438	77565			1263192	166209	332419	3.00825163	
66	146	88645	88645		1429401	166209	-		
500	982	99726			1595610	166210	332419	3.00825163	
21266	20532	110807			1761820	166209	-		
2860	2780	121887			1928029	166210	332419	3.00825163	
3202	2546	132967			2094239	166209	-		
3322	2986	144048			2260448	166209	332418	3.00826068	
3342	2412	155129			2426657	166210	-		
2164	2730	166210			2592867	166209	332419	3.00825163	
3192	3012	177291			2759076	166209	-		
3142	2736	188371			2925285	166210	332419	3.00825163	
2236	2146	199452			3091495	166209	-		
2392	3192	210533			3257704	166209	332418	3.00826068	
3108	3434	221613			3423913	166210	-		
3480	2130	232693			3590123	166209	332419	3.00825163	
2204	1290	243775			3756332	166211	-		
354	440	254855	254855		3922543	166208	332419	3.00825163	
754	496	265935			4088751	166209	-		
3160	2034	277016			4254960	177292	343501	2.91119968	
3156	2560	288097			4432252	166208	-		
3264	3118	299178			4598460	166209	332417	3.00826973	
3426	2736	310258			4764669	166209	-		
2282	2934	321338			4930878	166210	332419	3.00825163	
3046	3390	332419			5097088	166209	-		
3682	4380	343500			5263297	166210	332419	3.00825163	
3168	2472	354580			5429507	166210	-		
2294	2394	365661			5595717	166208	332418	3.00826068	
3178	3328	376742			5761925	166210	-		
3354	2864	387822			5928135	166210	332420	3.00824258	
2708	2438	398903			6094345	166209	-		
1394	1604	409984			6260554	166209	332418	3.00826068	
192	456	421064	421064		6426763	166210	-		
516	908	432145			6592973	166209	332419	3.00825163	
1724	1894	443226			6759182	166210	-		
3006	2476	454306			6925392	166209	332419	3.00825163	
3732	2800	465387			7091601	166209	-		
3586	3248	476467			7257810	166211	332420	3.00824258	
3152	2598	487548			7424021	166208	-		

Figure 4.6: Excel sheet with summed x-, y-values, timestamps and calculated frequency

Figure 4.7: Illustration of summed x-(blue), y-values (red) over time. The x-axis is the time and the y-axis are the x- and y-values.

Duration [s]	**run 1** [µs]	**run 2** [µs]	**run 3** [µs]	**run 4** [µs]	**run 5** [µs]
10	166591.5	166585.0	166397.2	165655.4	166585.0
20	166679.0	166117.1	166675.0	166675.0	166581.9
40	166486.4	166673.1	166675.0	166440.3	166673.1
60	166673.7	166672.4	166673.7	166672.4	166673.7
300	166665.2	166671.7	166640.4	166665.3	166640.4
600	166668.2	166649.6	166668.2	166655.8	166668.2
900	166667.2	166667.3	166638.4	166638.4	166667.2
1200	166660.4	166668.3	166669.7	166668.2	166668.2
1500	166668.8	166650.3	166650.3	166668.8	166668.8
1800	166652.8	166668.2	166669.2	166668.2	166669.2
2100	166668.6	166668.6	166668.6	166655.4	166654.5
2400	166668.1	166652.7	166656.5	166668.1	166668.1

Table 4.1: Measured half periods of led board using RGB values of pixels. Desired value 166666

4.4 Verification of accuracy

The calculated frequency is far too imprecise. In theory, however, the calculation should become more precise after a certain period of time and should even be accurate to the microsecond. But this has never been achieved. For this reason, a reliable artificial clock was used in the form of an LED that is clocked by a 50 Mhz quartz and is therefore sufficiently accurate.

Duration [s]	**run 1** [µs]	**run 2** [µs]	**run 3** [µs]	**run 4** [µs]	**run 5** [µs]
10	166636.1	166628.1	166628.1	166628.1	166628.1
20	166700.3	166602.3	166700.3	166602.3	166602.3
40	166685.6	166638.1	166638.1	166638.1	166638.1
60	166649.5	166649.5	166649.5	166649.5	166649.5
300	166667.4	166667.4	166673.6	166667.4	166667.4
600	166666.5	166666.5	166666.5	166669.6	166666.5
900	166668.2	166666.2	166668.3	166668.3	166651.7
1200	166669.1	166656.7	166658.3	166658.4	166669.3
1500	166669.7	166659.8	166658.5	166669.6	166659.7
1800	166661.7	166660.7	166668.9	166669.9	166668.9
2100	166668.4	166661.4	166668.4	166669.3	166662.3
2400	166669.7	166669.6	166663.5	166663.5	166662.7

Table 4.2: Measured half periods of led board using motion vectors. Desired value 166666

Chapter 5

Calculation of the frequency

5.1 Implementation of the algorithms

5.1.1 Calculation of the frequency by counting minima with upper limit

In a next step, the results of the rough calculation from the Excel sheet were transferred into a Python program.

The greatest difficulty proved to be the detection of the minimia, which characterize the standstill of the balance wheel. These were relatively easy visible to the unaided eye, but in an automatic calculation, appropriate conditions had to be set in order to take into account the correct values and correct the inaccuracies caused by the noise. The setting of an upper limit for the minima proved to be a suitable condition. All minima below this limit are therefore taken into account.

Furthermore, the calculation of the frequency in the Python program has been further simplified by not calculating each period individually, but by selecting the first and the last minimum over the whole period of time, in which it was recorded. This time span is then calculated by half of the number of minima minus 1. Half, because every half of the period the balance wheel stops and minus 1, because otherwise a half-period would be taken into account too much. This approach reduces the number of rounding errors, resulting in higher accuracy. After this calculation, the period duration T is now available, from which the frequency f can be calculated quite simply:

$$f = \frac{1}{T}$$

5.1.2 Calculation of the frequency by counting minima with stepsize

Chapter 6

Results

6.1 Measurements

Reference measurements were performed on the test watch on three days. Two measurements were carried out with the Greiner Compact 900 and one with the Wisio Scope meter. Calculations using the Greiner Compact diverge greatly from those taken with the Wisio Scope. In the next section, this divergence and its causes will be discussed in more detail.

As the test watch was in a distinctly colder environment (about 5 degrees Celsius) shortly before the Greiner Compact meter was used to measure the reference values, it is highly likely that the small metal spring has contracted and the watch ran faster as a result. Before the calculation with the Wisio Scope time scale, the test watch was in an ambient temperature of about 25 degrees Celsius for a long time. A difference of only 5 degrees Celsius can affect the accuracy of the watch. As the reference measurements are based on a difference of about 20 degrees Celsius, the above-mentioned difference occurs with the measured values obtained. The optical measurements of the picamera were carried out on an average at about 25 degrees Celsius, therefore the reference values determined by the Wisio Scope time scale are best suited for the comparison.

All measurements were taken in different positions of the watch with different length of recordings whereas each position was recorded in 10, 20, 40 and 80 minutes.

Figure 5 and Table 3 show the deviation per day based on the following calculation: Taken the first timestamp (t1) of a stillstand and taken the last timestamp (t2) of a stillstand, the counted number of stillstands (s1), number of strokes per hour (s2) (21600):

Positions	Greiner Compact 900	Watch Exper II	10 min	20 min	40 min	80 min
up	-13	-31	-48.7	-48.0	-47.6	-47.0
right	-30	-64	-58.3	-65.5	-54.4	-49.4
down	-20	-46	-122.2	-122.2	-125.4	-121.4
left	-14	-34	-87.0	-87.1	-87.1	-83.1

Table 6.1: variations per day in seconds

$t2 - t1 = \triangle t$
$\triangle t/s = f * 2$
$(f * 2 * s2) - 3600 =$ deviation per hour

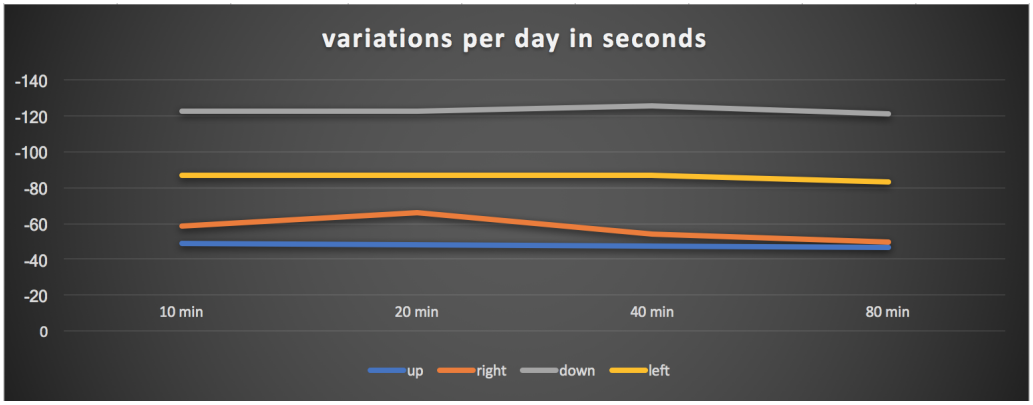

Figure 6.1: Variation of different time and position recordings

6.2 Calculation of rate deviation from frequency

6.3 Evaluation

Chapter 7

Discussion

7.1 Difficulties

7.2 Outlook

List of Figures

2.1 Camera architecture. [**ReadTheDocsPicamera**] 5
2.2 Operation of a mechanical watch [**Uhrwerk**] 8
2.3 lever escapement and balance wheel [**Uhrwerk**] 9
2.4 period of the balance wheel [**Witschi_basics**] 10
2.5 half of a period of the balance wheel [**Witschi_basics**] 10

4.1 Test setup for records 16
4.2 Snapshot of the animation of the SAD values, the lighter the color the higher the value of the SAD 17
4.3 Snapshot of the animation of the hypotenuses, the lighter the color the higher the value of the hypotenuse 18
4.4 Motion vectors drawn in one frame 19
4.5 It shows a ball moving in 5 consecutive frames. The arrow shows its displacement vector. 20
4.6 Excel sheet with summed x-, y-values, timestamps and calculated frequency 22
4.7 Illustration of summed x-(blue), y-values (red) over time. The x-axis is the time and the y-axis are the x- and y-values. . . . 23

6.1 Variation of different time and position recordings 29