Motion Recognition with Android devices

Gabor Paller
gaborpaller@gmail.com
Sfonge Ltd.

Setting the context

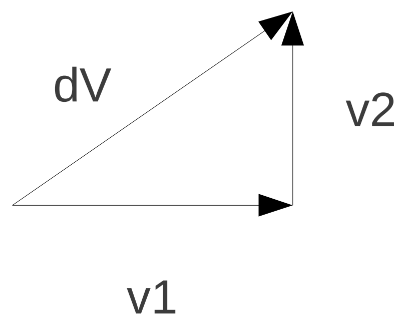
- Sensors in smart phones are part of the context-aware computing vision.
  - *“context awareness refers to the idea that computers can both sense, and react based on their environment.”*
  - Location is the most traditional context variable
- Our goal:
  create context-aware applications on *today's smart phones* that process the *motion* context variable
  - Generate “motion” information from raw sensor data – we will concentrate on this
  - Use that information in the application – that's your job :-)

Motion sensors

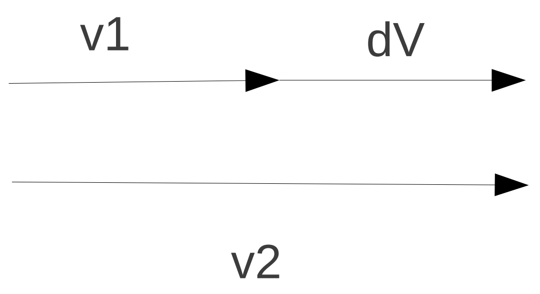
- Sensors that can be used to detect motion
  - Location sources (WiFi, GPS, cellular network)
  - Light sensors (e.g. proximity detection)
  - Accelerometer
  - Gyroscope

Acceleration

Acceleration caused by the change of direction



Acceleration caused by the change of velocity



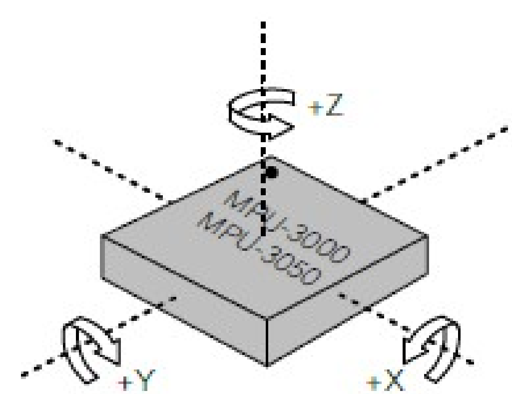
$$a = \frac{\Delta V}{\Delta t}$$

Accelerometer

- First smart phone with built-in accelerometer: Nokia 5500 “sport device”: 2005 Q3 (Symbian)
- Android supports accelerometer sensor even from pre-1.0 API
- One frequent type in Android phones: Bosch BMA150
  - One-chip solution with internal electro-mechanic structure
  - -2g - +2g/-4g - +4g/-8g - +8g (10 bit resolution)
  - “Any motion” interrupt mode
  - Built-in thermometer

Gyroscope

- Very new phenomenon as gyroscopes suitable for consumer electronic devices appeared very recently
- First appearance: Wii Motion Plus accessory, 2009 June
- Android supports gyroscope sensor even from pre-1.0 API
- First Android smart phone: Nexus S (end of 2010)
- Example: InvenSense MPU-3000
  - One-chip solution with internal electro-mechanic structure
  - Measures rotation along 3 axes
  - 16 bit resolution



Compass

- Measures the device orientation wrt. the magnetic vector of the Earth
  - This vector points toward the magnetic center of the Earth – mainly down
  - It has a component that points to the magnetic North pole – that's what we use for orientation
  - If, however, the device is not held horizontally, the downward vector element influences the measurement
  - Also sensitive for all sorts of metal objects
- Consequence: can be used for motion detection only in special cases

Android sensor support

- In android.hardware.SensorManager
  - Listing sensors
  - Sensor sampling
  - Some processing functions like vector transformations
- In competing platforms:
  - Snap, shake and double tap (with accelerometer) detection in bada
  - Shake detection in iOS

Sampling the sensors

- With what frequency?
- With what sampling precision?
- With what battery budget?

A word from our sponsor

- The samples were captured with this tool: http://mylifewithandroid.blogspot.com/2010/04/monitoring-sensors-in-background.html
- The samples were analyzed with the open-source mathematics software called Sage http://www.sagemath.org/

Sampling frequency

- Android API allows you to define the sampling frequency only in relative, symbolic way.
  - SENSOR_DELAY_NORMAL
  - SENSOR_DELAY_UI
  - SENSOR_DELAY_GAME
  - SENSOR_DELAY_FASTEST

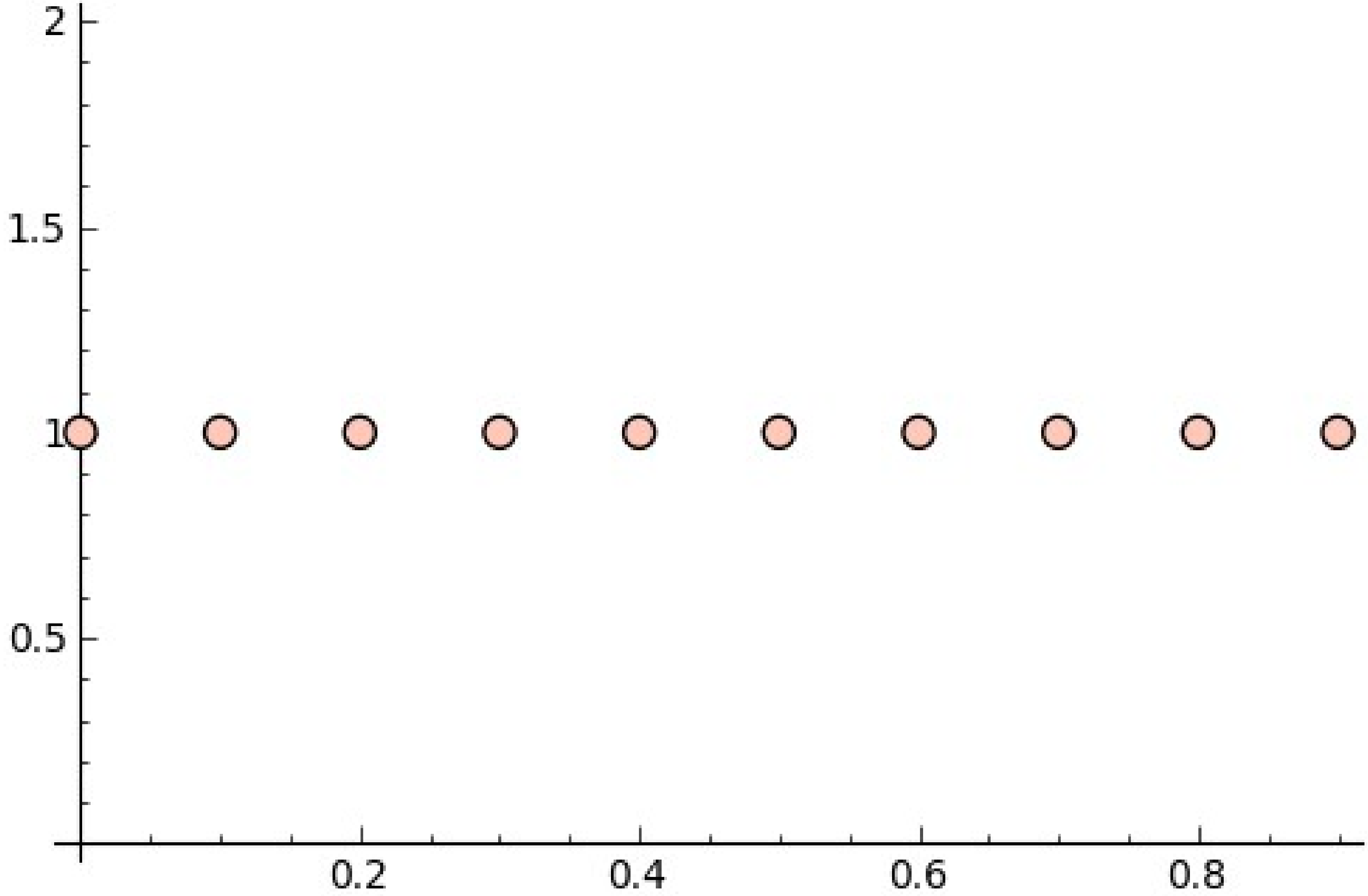
Nexus 1

Label	Average frequency	Maximum frequency	Minimum frequency
NORMAL	4.09 Hz	4.34 Hz	2.08 Hz
UI	9.87 Hz	11.11 Hz	4.14 Hz
GAME	16.16 Hz	20.24 Hz	4.33 Hz
FASTEST	24.45 Hz	33.13 Hz	4.34 Hz

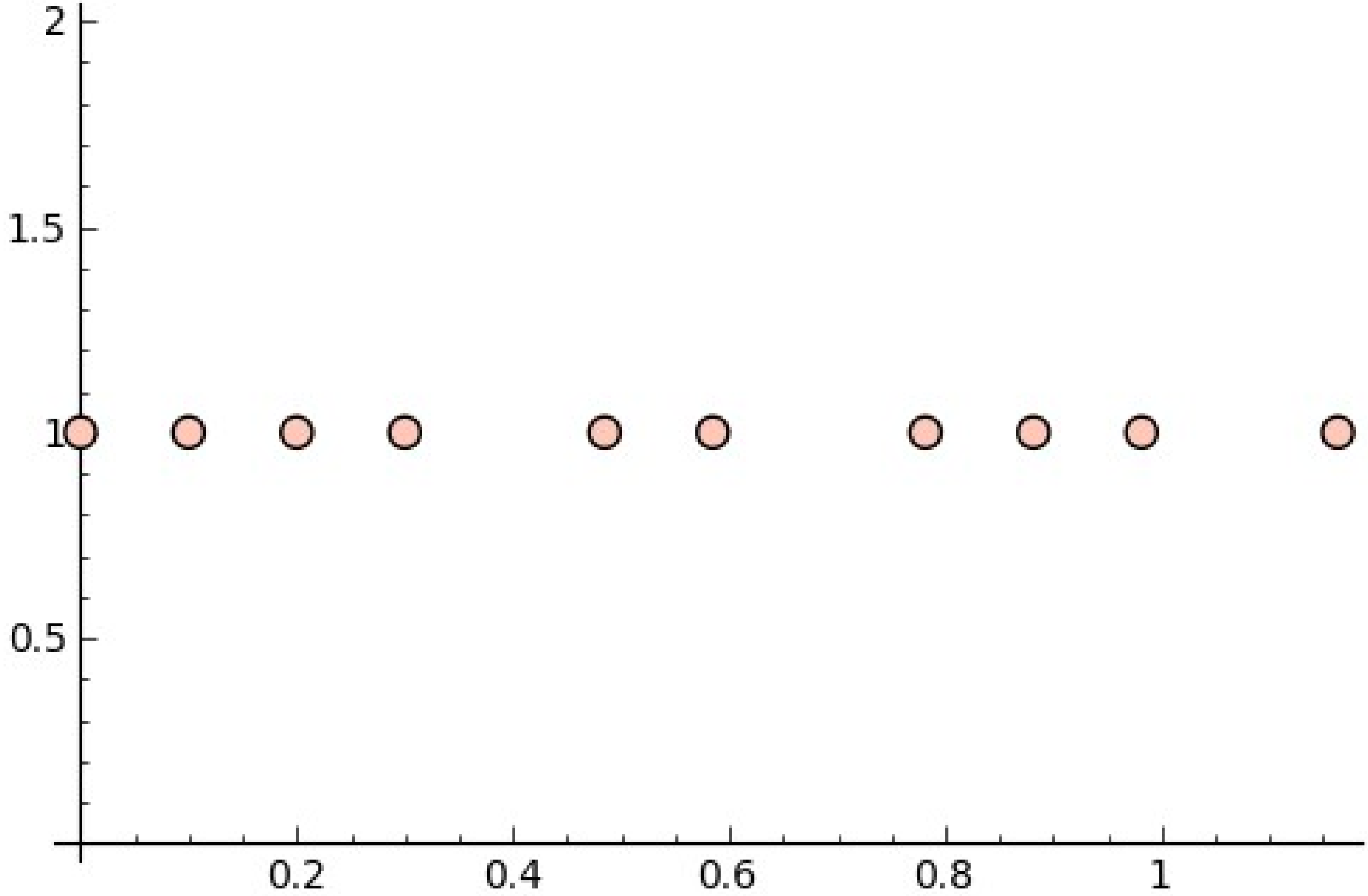
Sony-Ericsson x10 mini

Label	Average frequency	Maximum frequency	Minimum frequency
NORMAL	4.74 Hz	4.81 Hz	4.49 Hz
UI	14.15 Hz	14.19 Hz	11.64 Hz
GAME	32.55 Hz	32.84 Hz	22.75 Hz
FASTEST	94.77 Hz	96.44 Hz	37.68 Hz

Sampling precision: even sampling

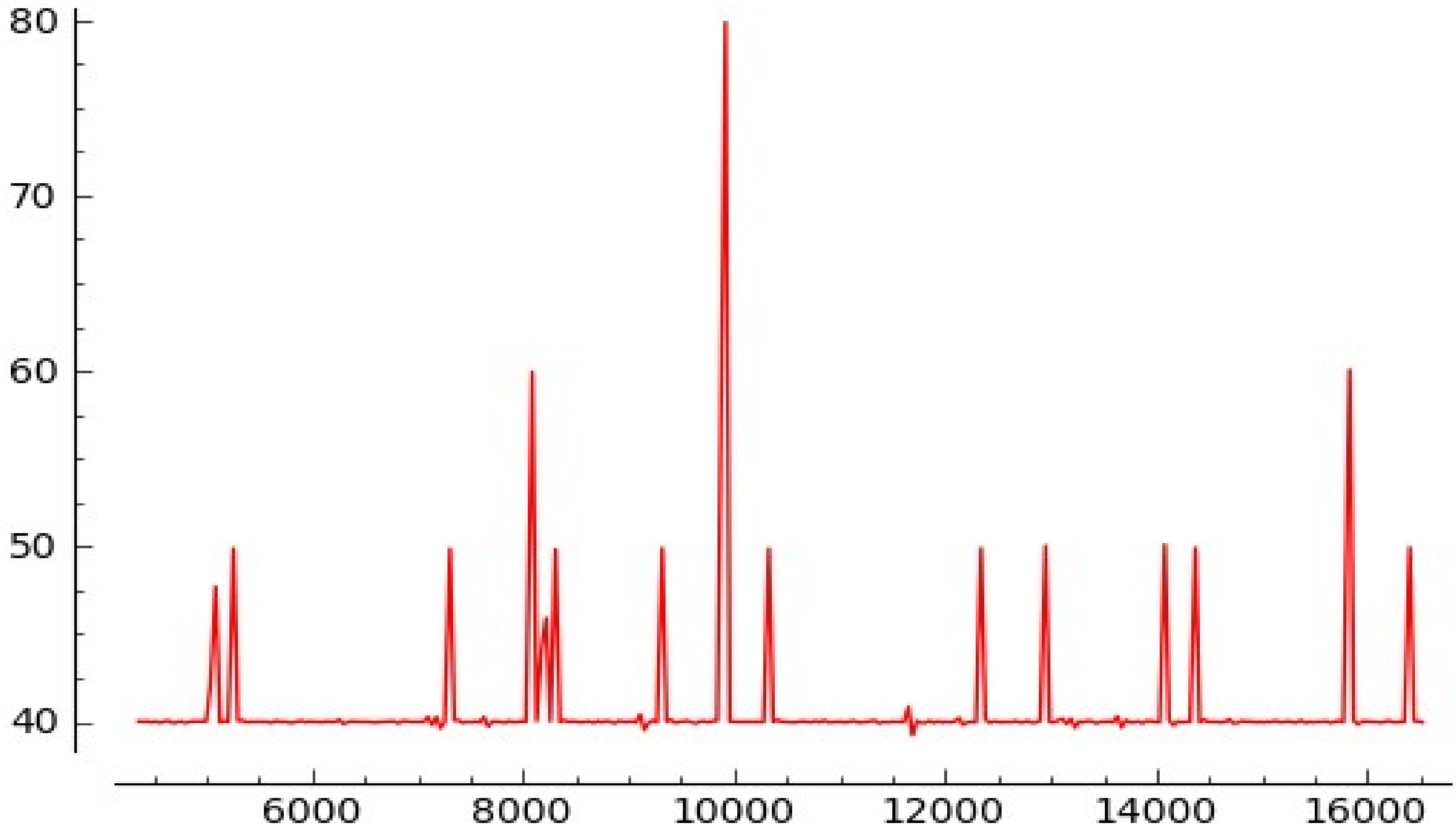


Sampling precision: uneven sampling



Variation of sampling period

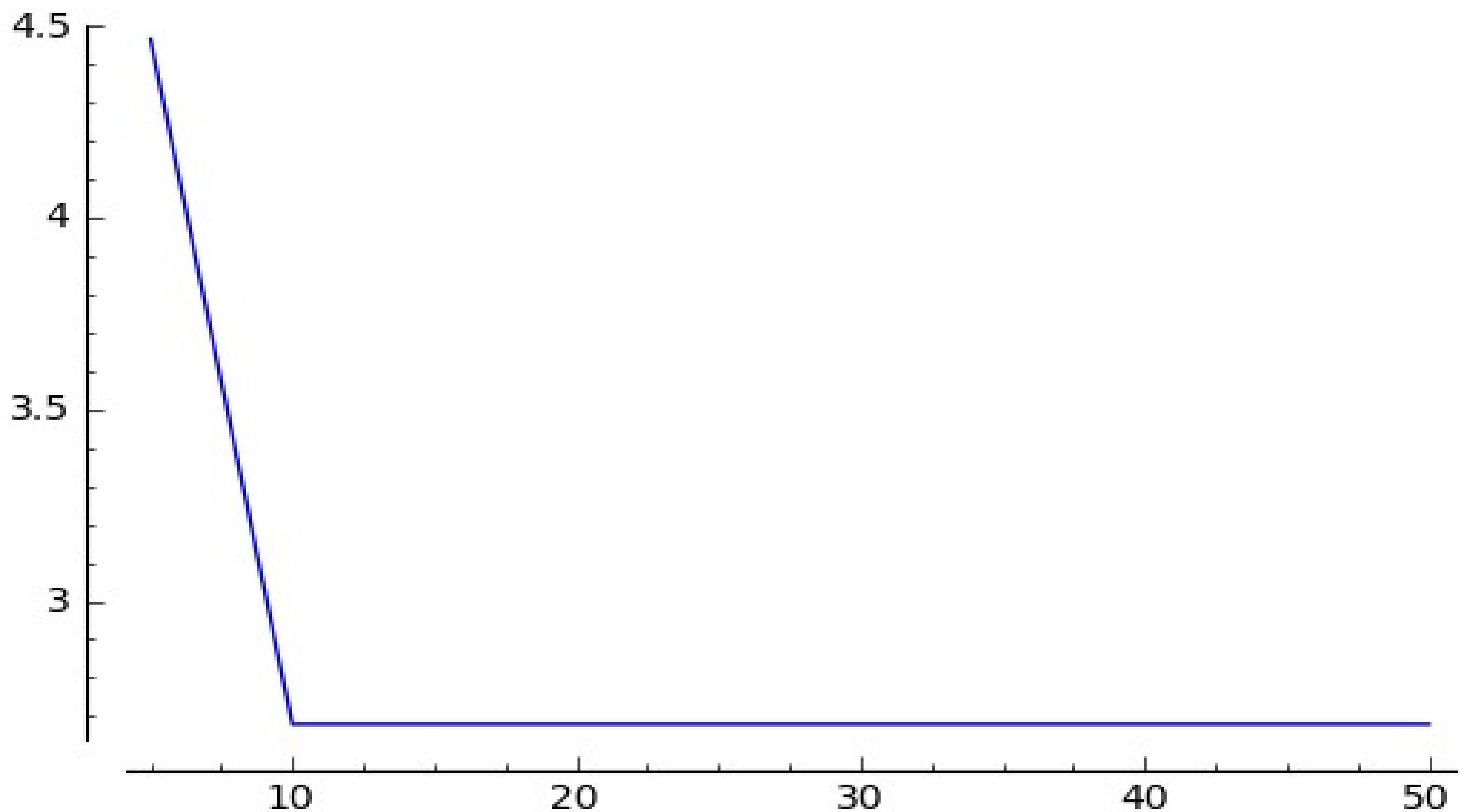
Consecutive sampling periods in milliseconds



Nexus 1, FASTEST sampling rate

Percentage of uneven sampling periods (NORMAL)

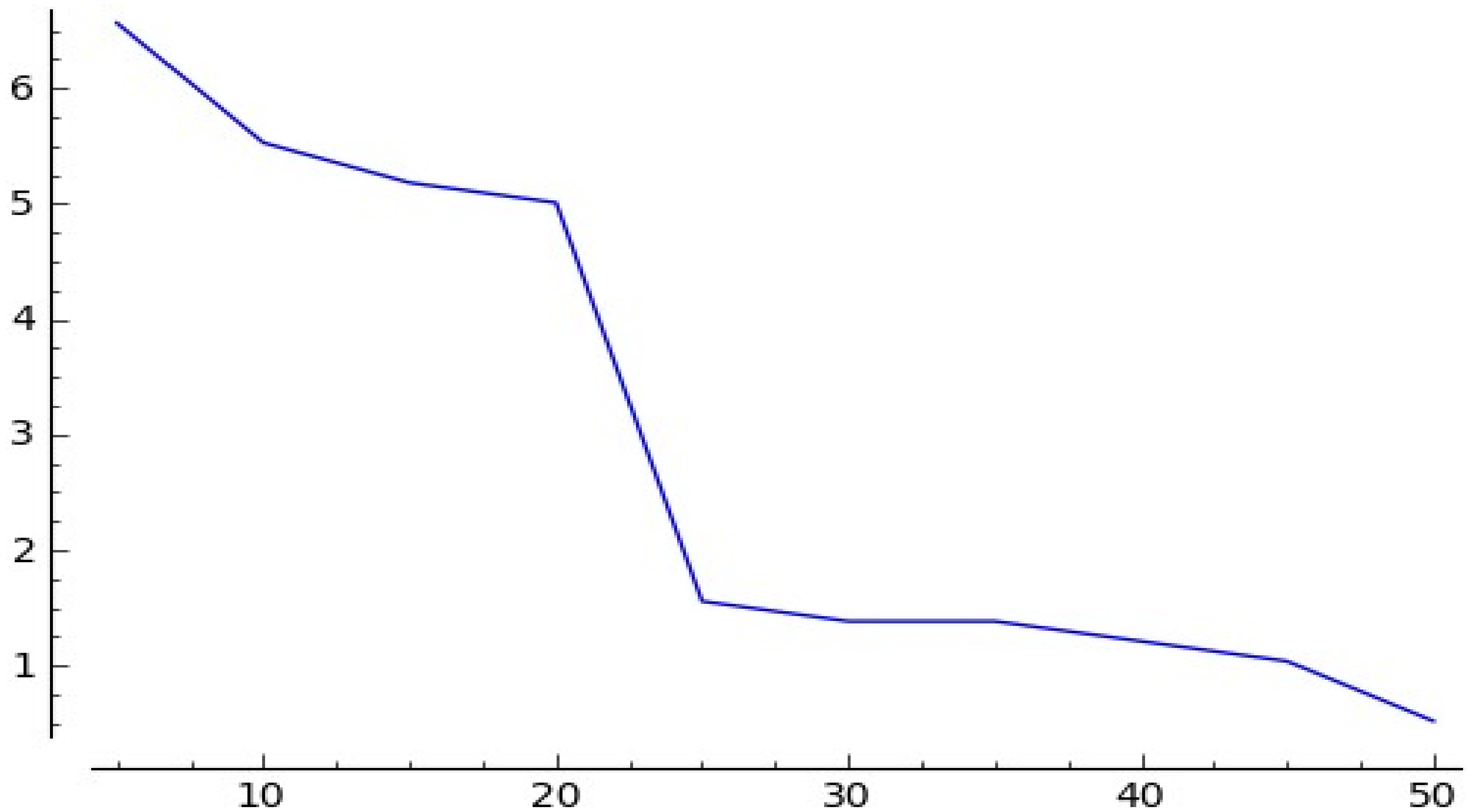
Percentage of sampling period times out of the x% band compared to the average sampling period



Nexus1, NORMAL sampling rate

Percentage of uneven sampling periods (FASTEST)

Percentage of sampling period times out of the x% band compared to the average sampling period



Nexus1, FASTEST sampling rate

Battery cost

- Measurement method:
  - Nexus1
  - Reference measurement:
    - airplane mode
    - fully charged
    - undisturbed for the whole night (about 8 hours)
    - Consumption measured by the battery graph application in %
    - 0.25% battery consumption per hour
  - Measurements
    - airplane mode
    - fully charged
    - undisturbed for the whole night
    - sensor sampling with the specified speed
    - empty sensor event processing callback

Battery cost

Sampling speed	Battery consumption (%/hour)
NORMAL	1.71
UI	3.19
GAME	3.27
FASTEST	3.41

Experiences

- There are significant differences in the sampling frequency values of different phone models
- The sampling frequency is not stable, there are interruptions (sampling is not the most important activity of the phone)
- The lower the sampling frequency is, the smaller the sampling period variation is
- There is significant battery cost associated with sampling
- NORMAL sampling frequency has significantly lower battery consumption than the others

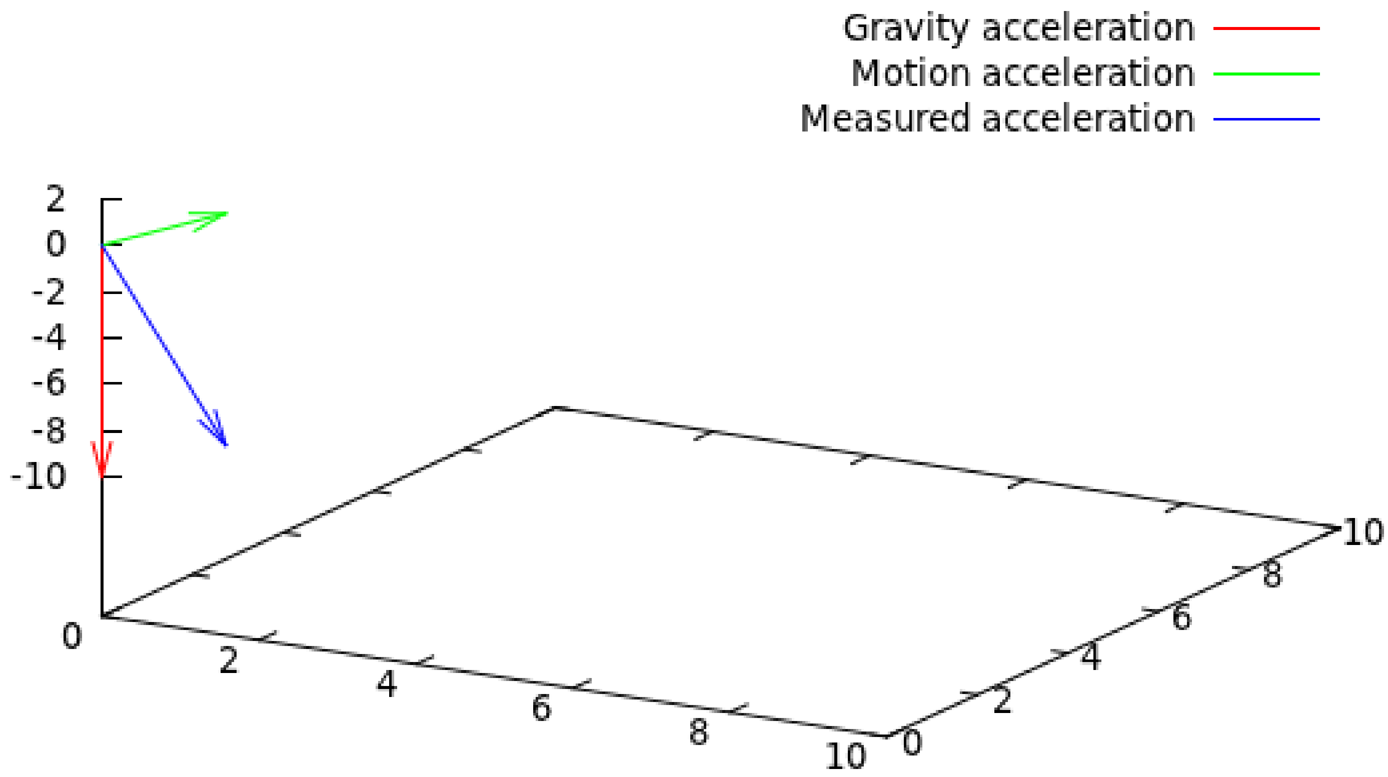
Consequences

- Try to use NORMAL sampling frequency if you can
- If you have to measure precisely, use the time stamp that comes with the sensor event and interpolate
- Deactivate sampling in case you don't need it
  - This advice does not help background motion monitoring applications

Extract motion information from accelerometer data

- Accelerometer data is a vector, having 3 axes (x,y,z)
- This vector has the following components:
  - Gravity acceleration
    - Pointing toward the center of the Earth
    - Value of about 10 m/s^2
    - That's what we measure when the accelerometer is used to calculate tilt
  - Any other acceleration the device is subject to
    - Added to the gravity acceleration
    - “Disturbs” tilt measurement in gaming (swift movements cause acceleration) – hence the reason for gyroscopes
    - Can be used for movement detection

Measured acceleration



Absolute value vs. orientation

- In most of the use cases we don't know the orientation of the device
  - E.g. the device is in the trouser pocket or held casually in the hand – how is the device's z axis orientated wrt. the downward axis (pointing to the Earth)?
- In these cases we can use only the length of the acceleration vector
  - Vector-addition of the gravity acceleration and the motion acceleration

Absolute value

- x, y, z: acceleration vector components
- g – value of the gravity acceleration (can be approximated as 10)

$$a = \sqrt{x^2 + y^2 + z^2} - g$$

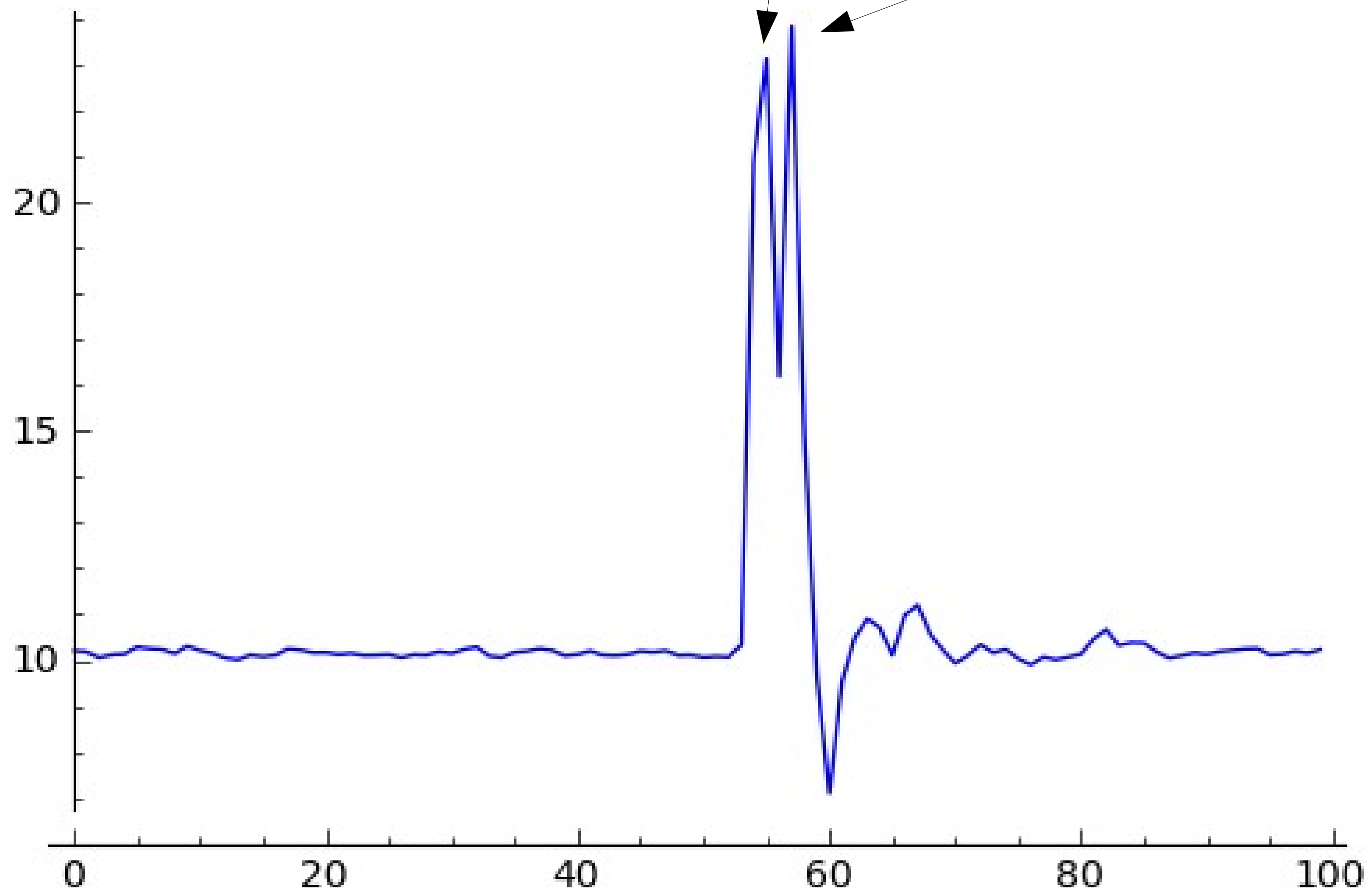
Some simple movements

- Useful gestures (inspired by bada)
  - Snap – one shake
  - Shake – repeated shakes
  - Double-tap – tapping on the device (anywhere, not only on active touch screen)
    - On inactive touch screen
    - On the device body

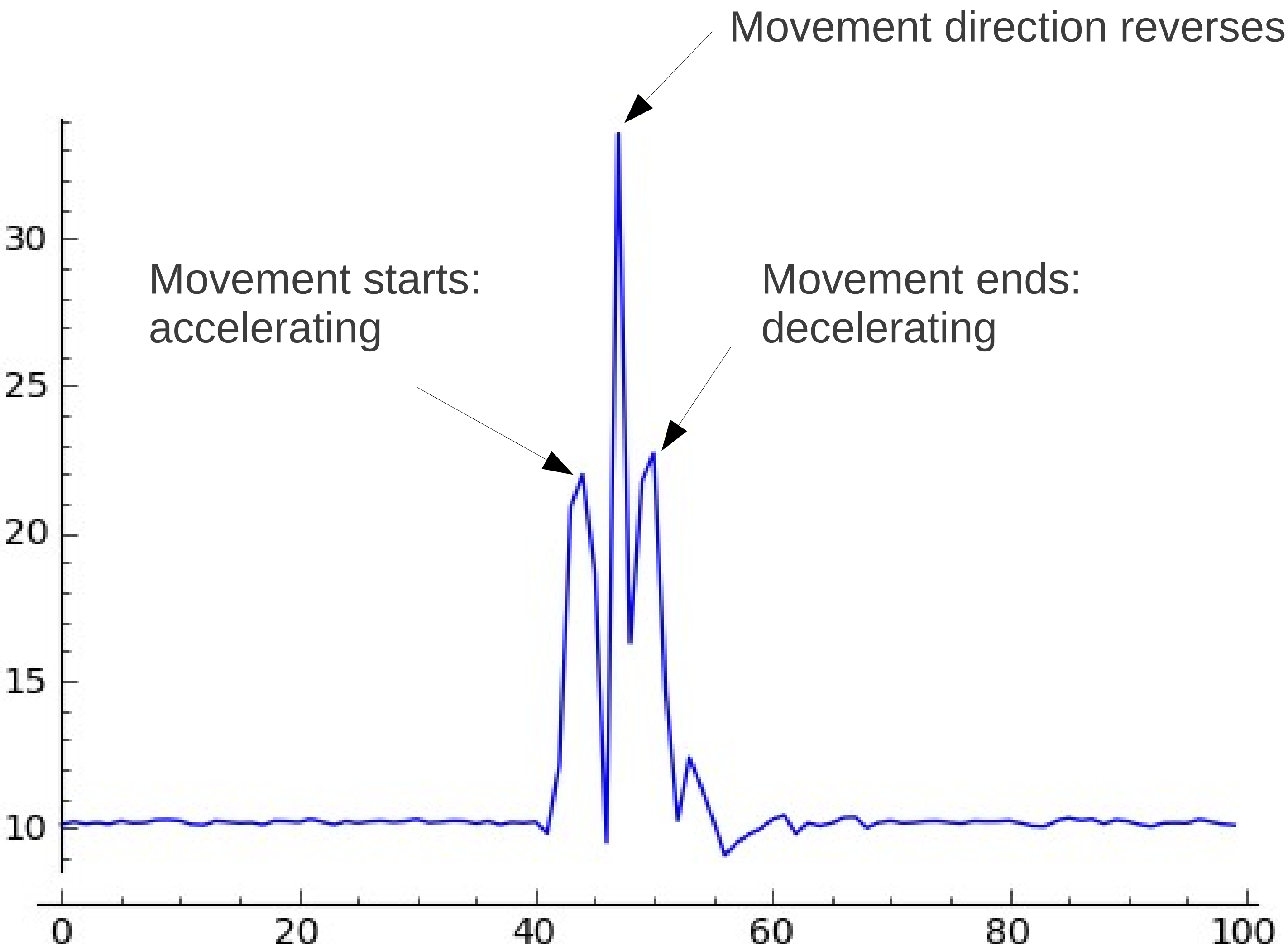
Snap – one way

Movement starts: accelerating

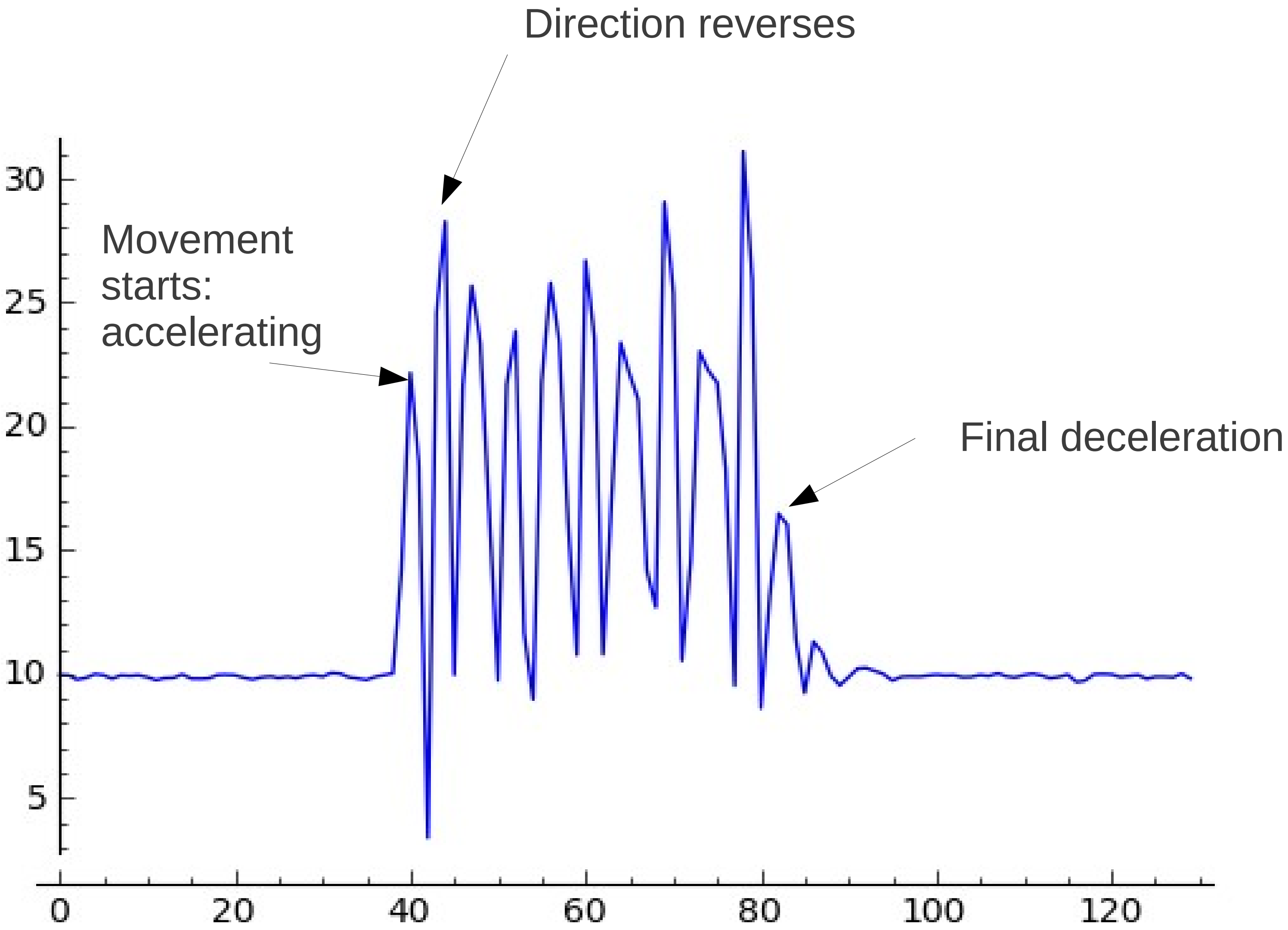
Movement ends: decelerating



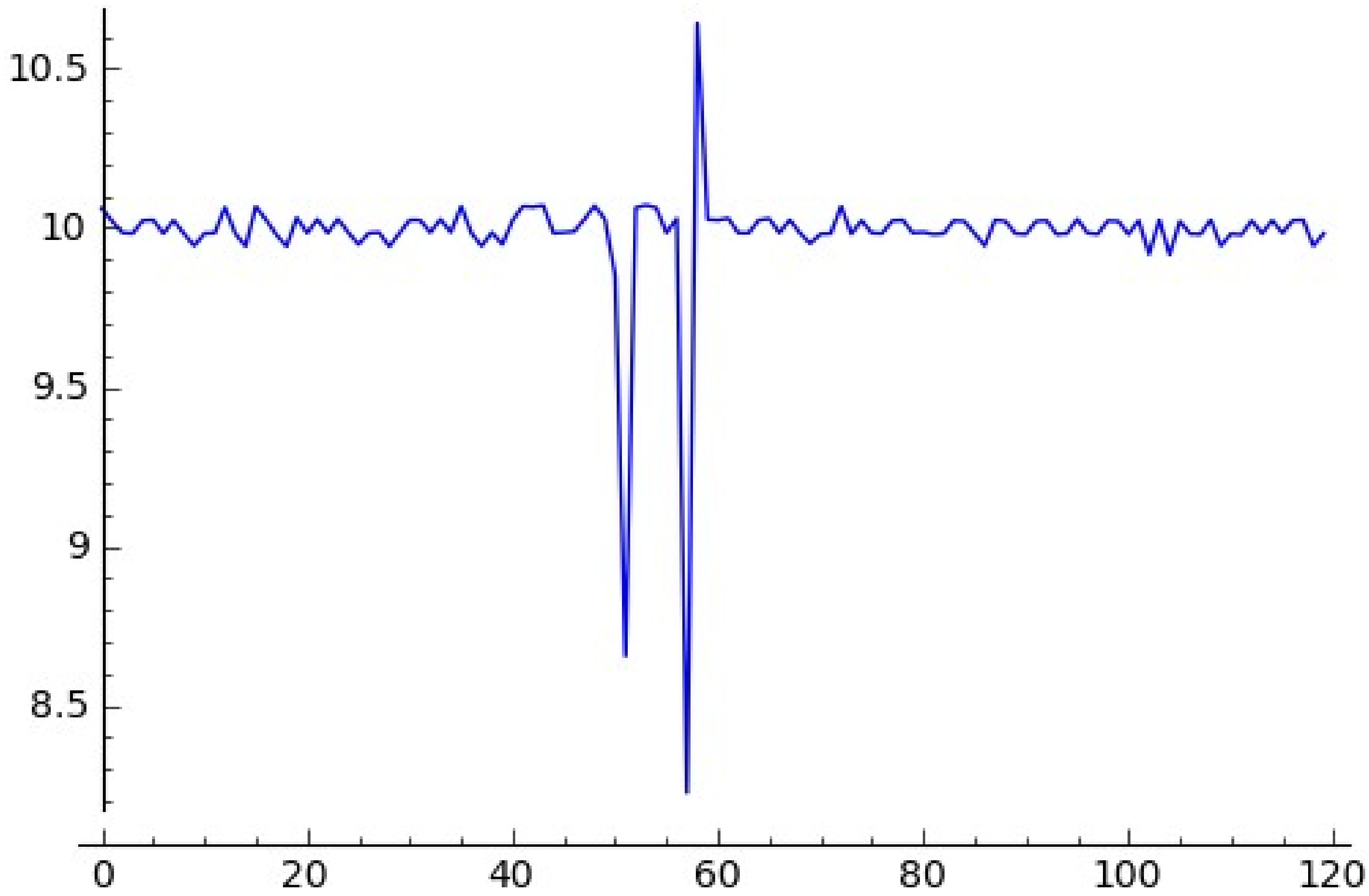
Snap – two way


Movement direction reverses
Movement starts:
accelerating
Movement ends:
decelerating
30
25
20
15
10
0
20
40
60
80
100

5 shakes

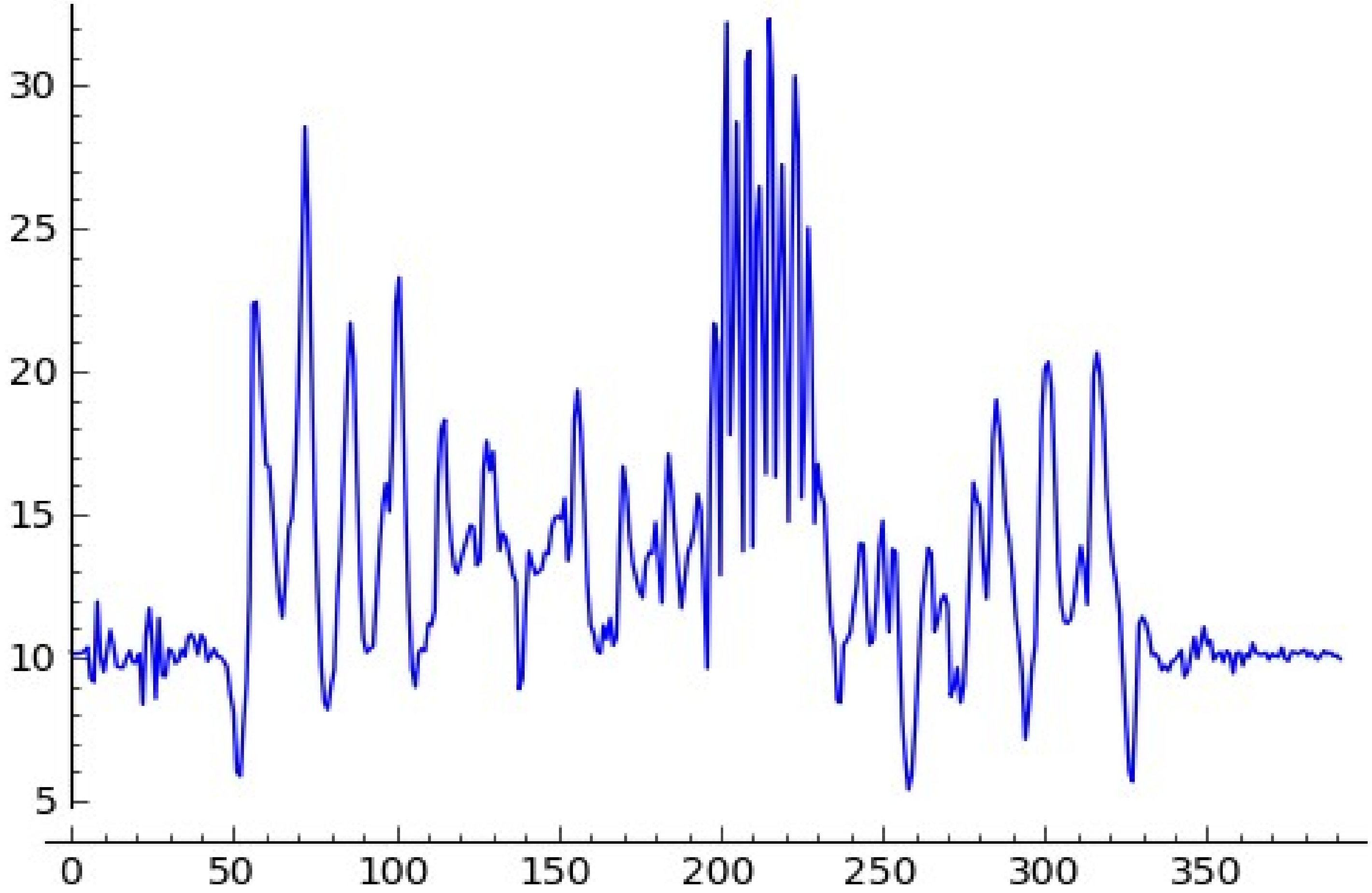

Direction reverses
Movement starts: accelerating
Final deceleration
30
25
20
15
10
5
0
20
40
60
80
100
120

Double-tap



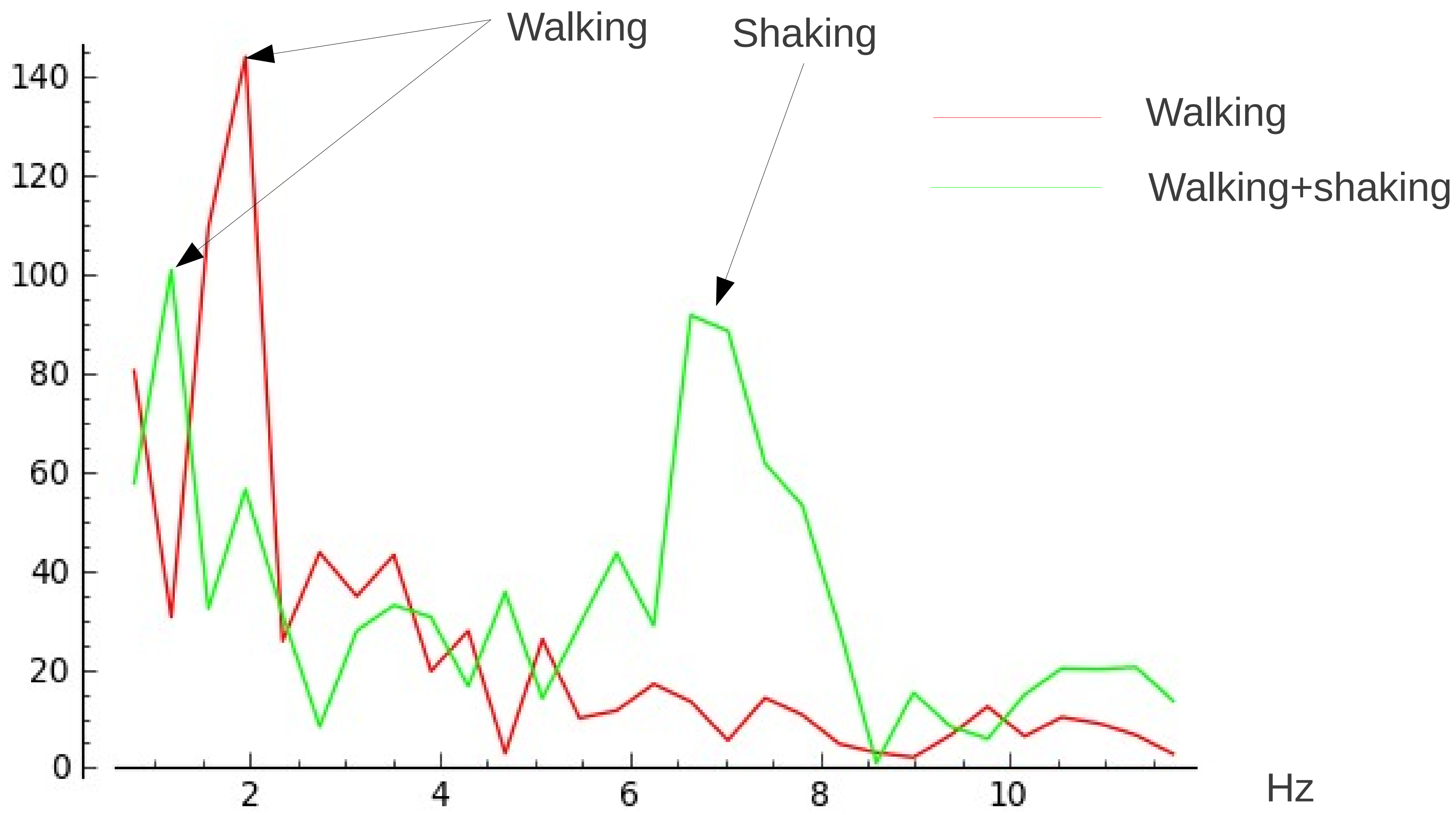
Phone was laying on the table, its body (not touch screen) was tapped

5 shakes while walking



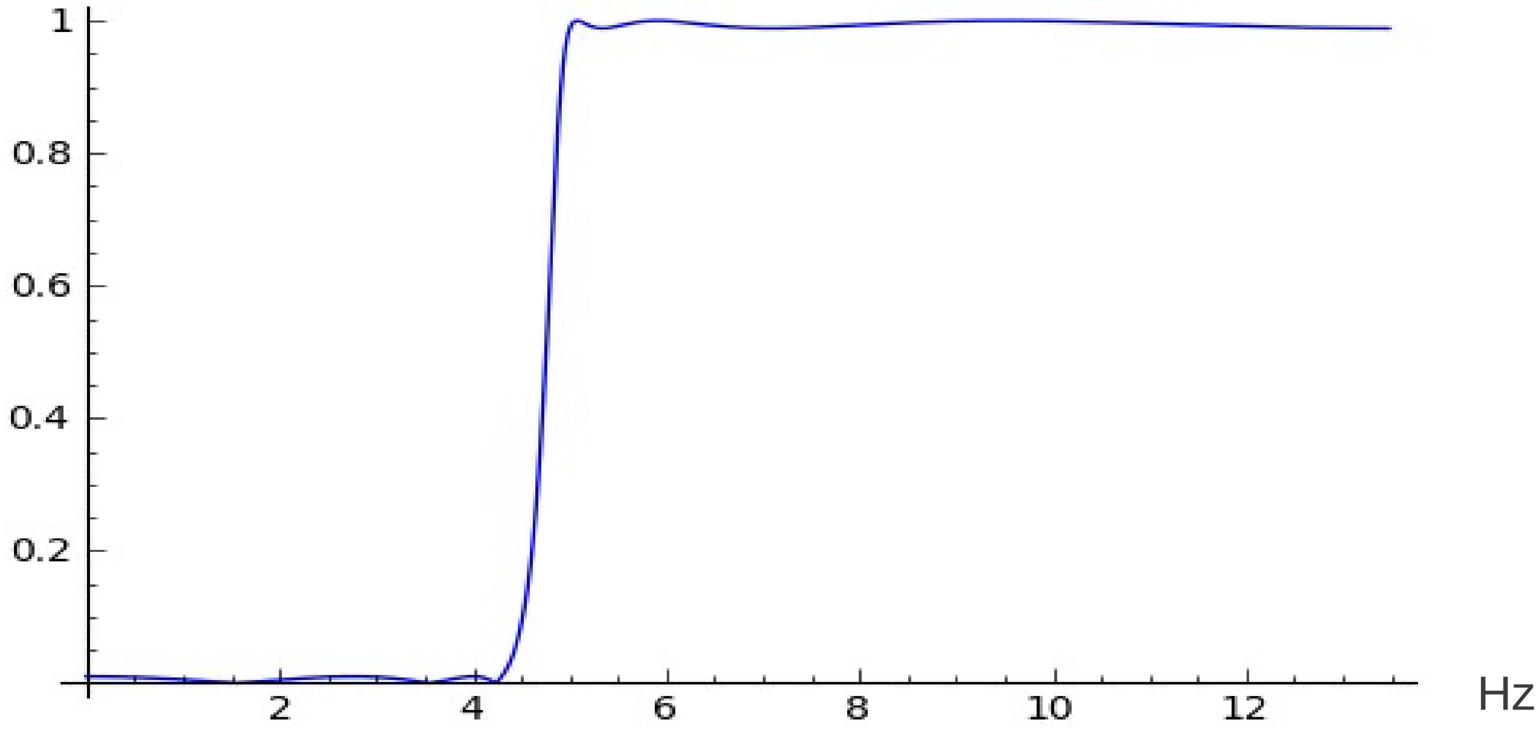
Walking, phone held in swinging hand, 5 shakes mid-stride

Separating movements in the frequency domain



64-point Fast Fourier Transform performed at samples 50 and 200

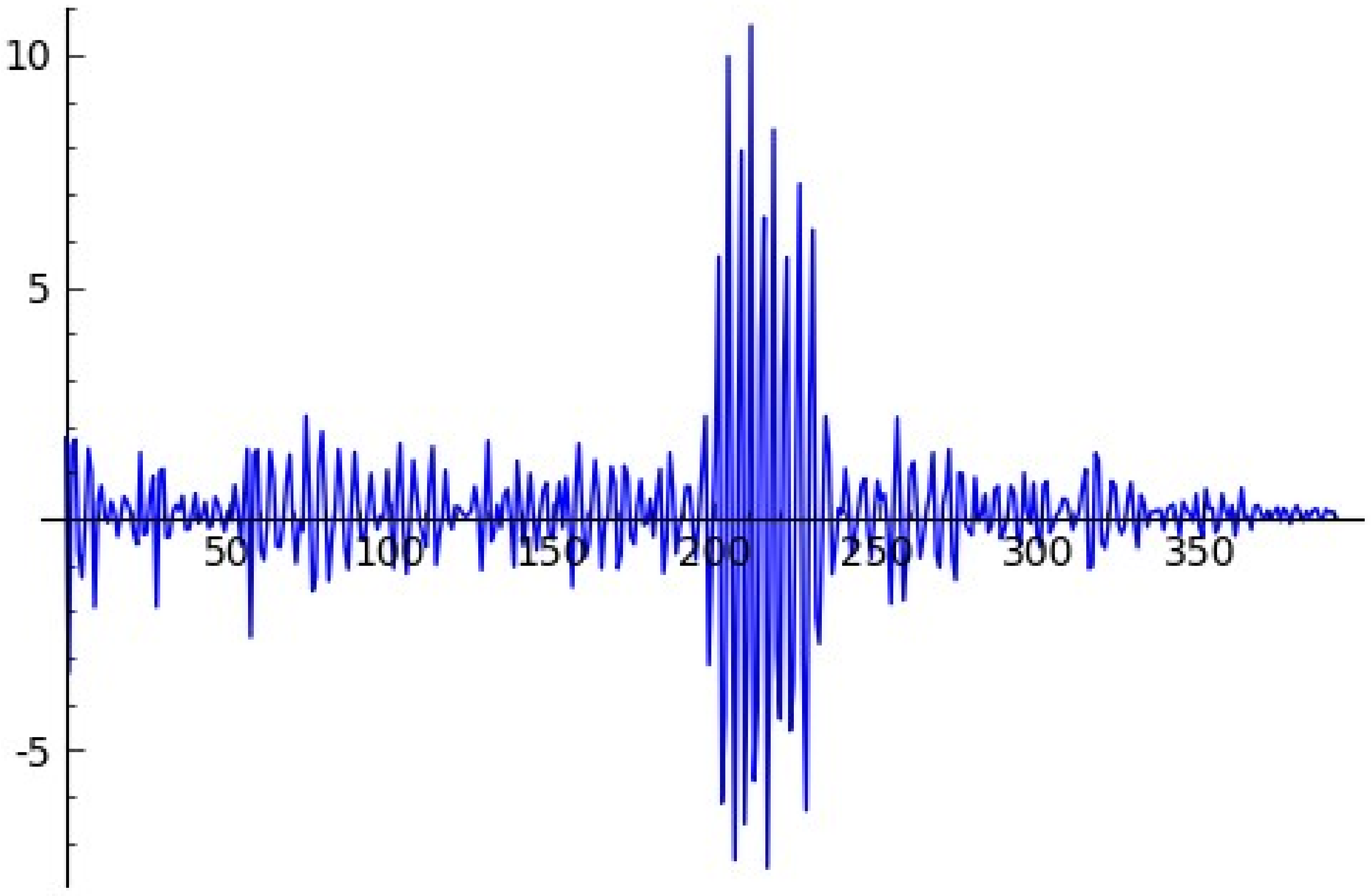
High-pass filter for separating the shakes



Frequency characteristics of the IIR filter designed with the following Sage command:

```
scipy.signal.iirdesign(5.0/12.5,4.0/13.5,0.1,40.0)
```

Applying the filter

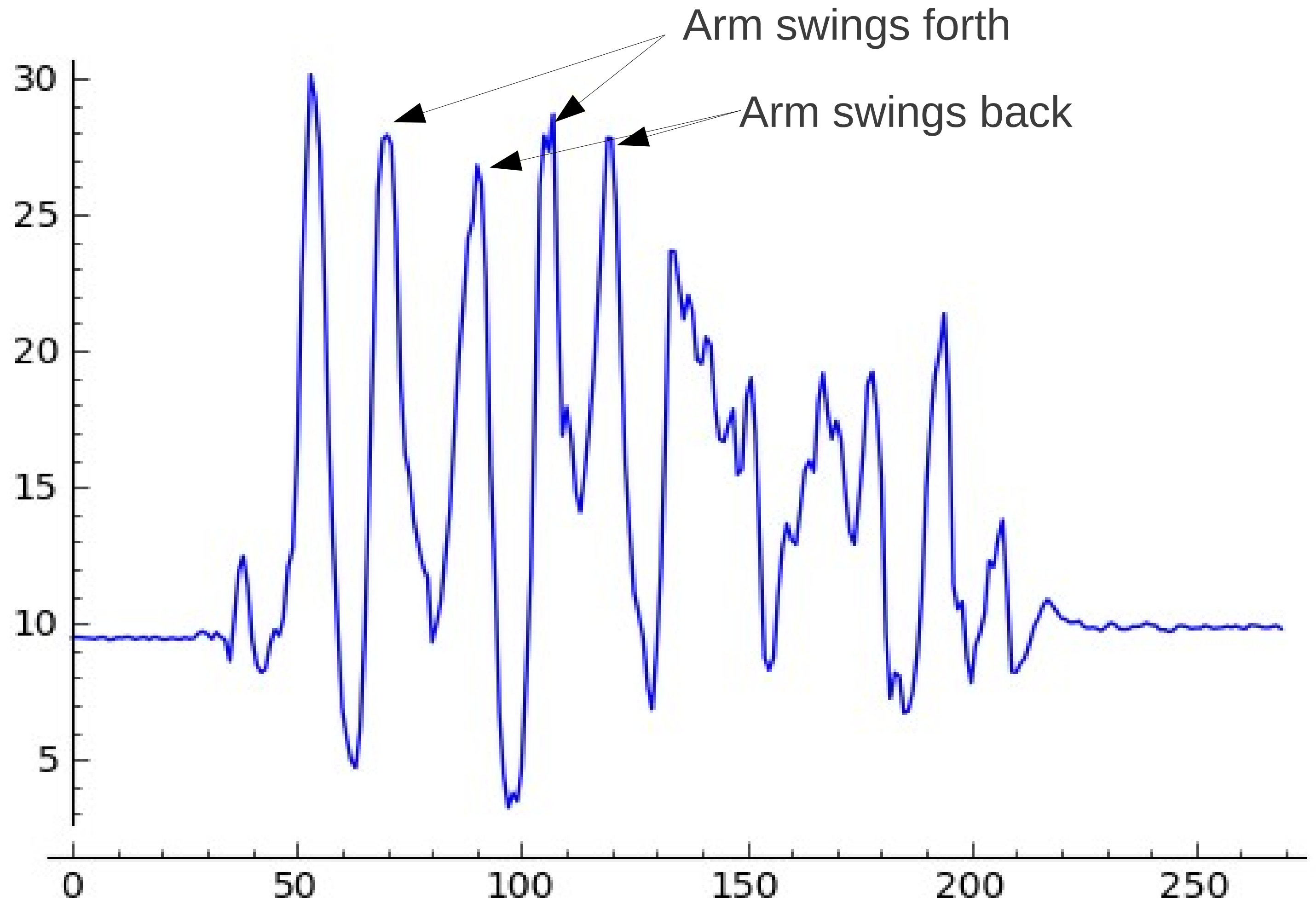


4 shakes can be reliably recognized

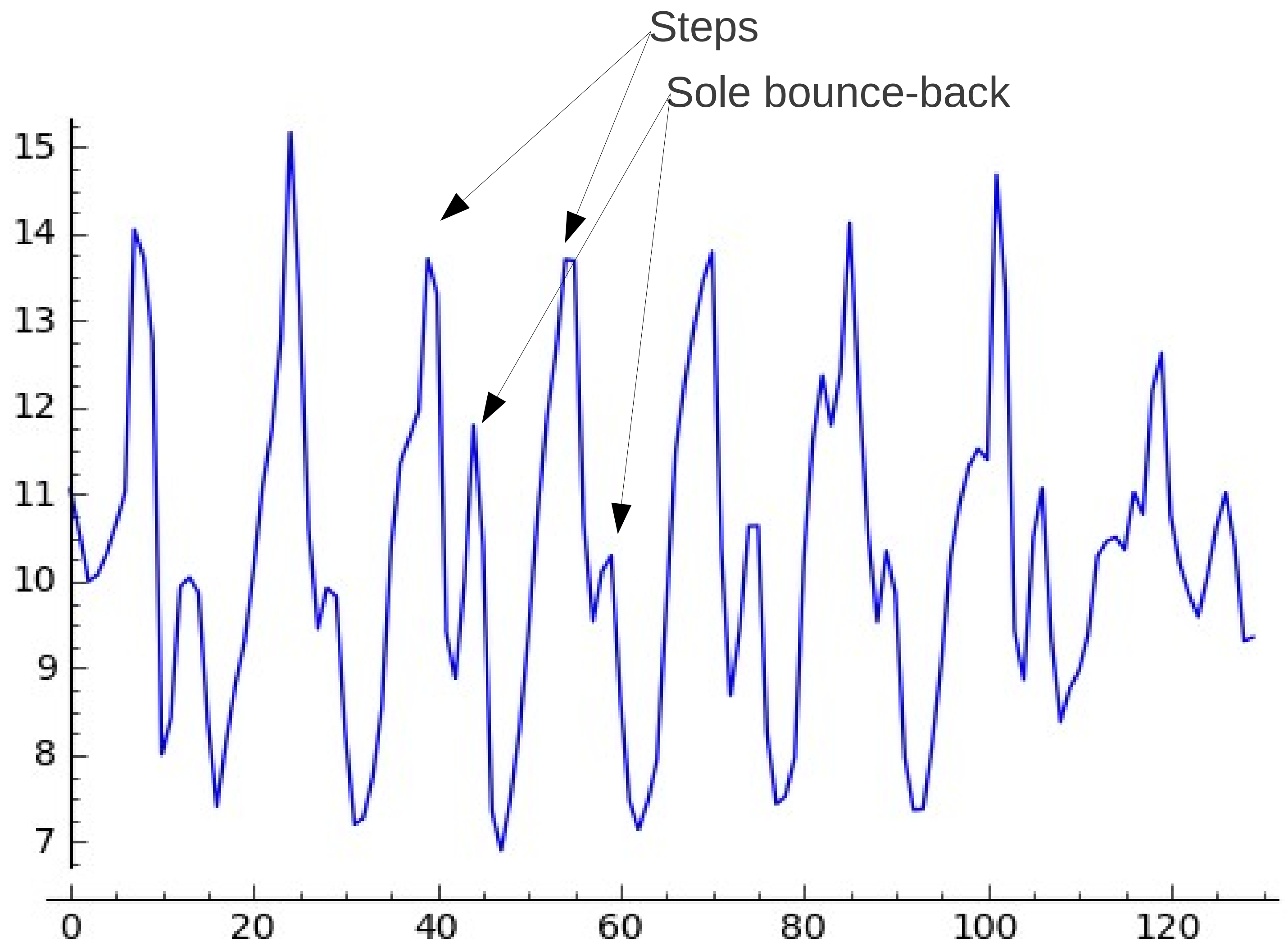
Some complicated movements

- Periodic movements can often be recognized
  - Beware of sum of movements – often they cannot be separated into components, particularly if they have similar frequency
  - You can try to use an additional sensor like the gyro to help the separation
- Placement of the phone is crucial
  - Where on the body?
    - Torso (shirt, trouser pocket)
    - Hand, swinging
    - Hand, fixed (like user watching the display while walking)
    - Legs
  - Although placement is “casual” (as opposed to precise placement used in professional measurements), extreme cases make movement recognition impossible
    - Like phone in a loose trouser pocket which moves partly independently from the body

Phone in swinging hand while walking



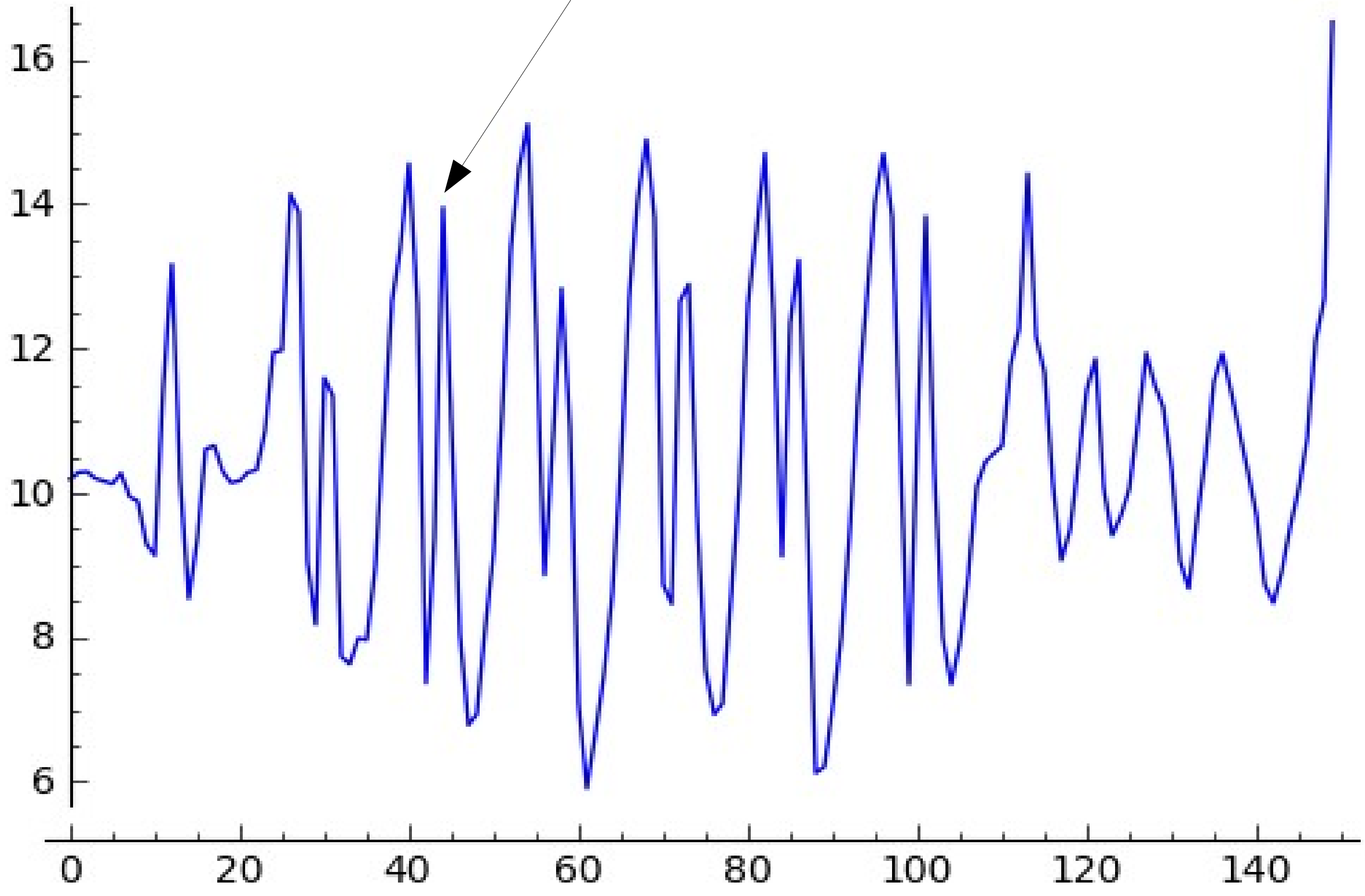
Phone in steady hand while walking



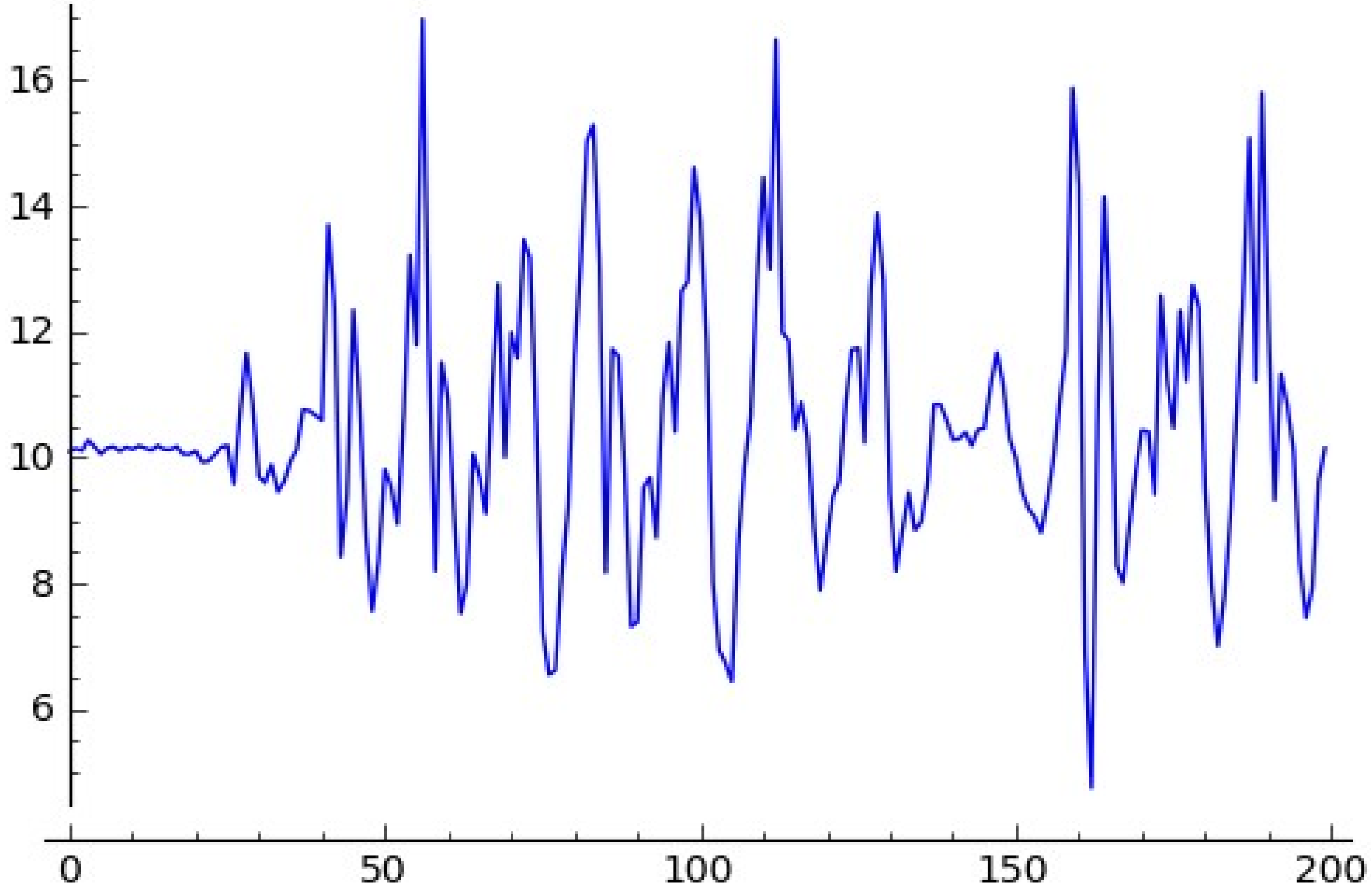
Phone held in hand with display upward,
position relative to the body is (reasonably) fixed

Shirt pocket, walking

More pronounced sole bounce-back



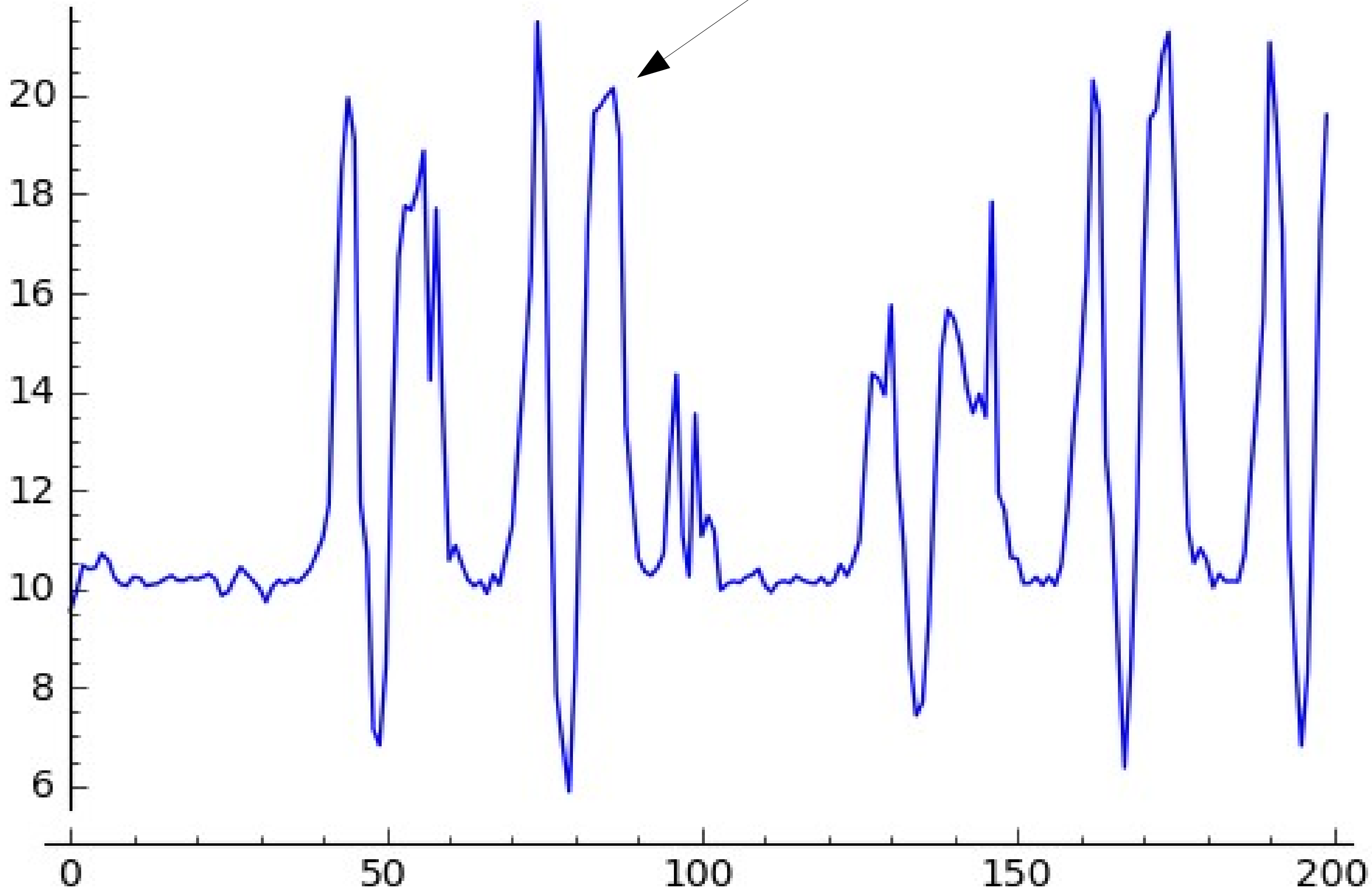
Trouser pocket, walking



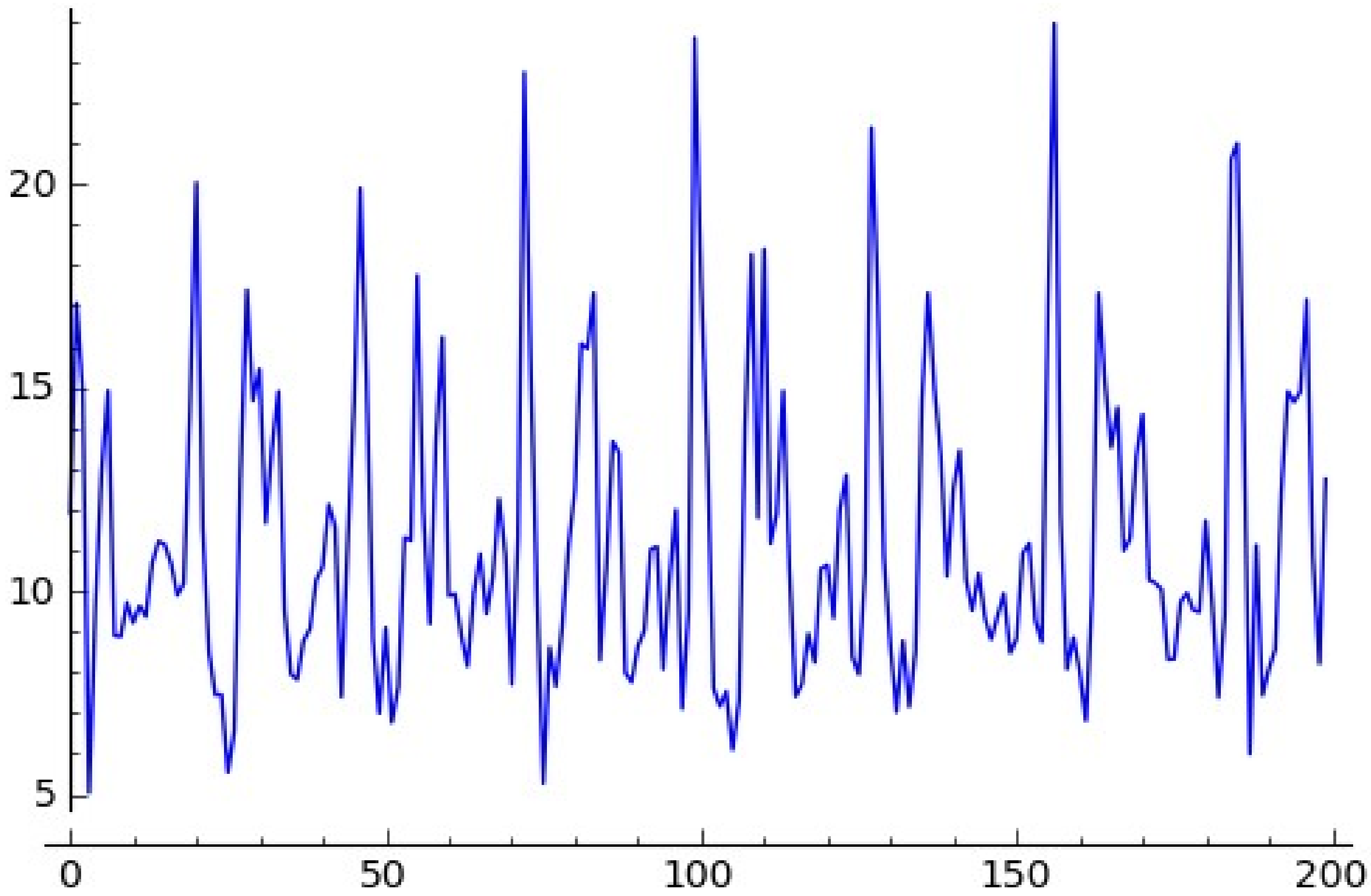
High level of noise caused by the complex movement of the waist when walking

Shoe, walking

Very strong bounce-back effect



Horror: loose trouser pocket



Example application: Activity Level Estimator (ALE)

- Current research at the University of Geneva
  (Contact persons: Dr. Katarzyna Wac and Jody Hausmann)
- An Android application made for users that want to monitor their level of physical activity during the whole day and not only during sports activities
- Detects the intensity of the body movement to determine the user's level of activity and calories burned
- Prototype actually developed at the University of Geneva and for the European project called TraiNutri

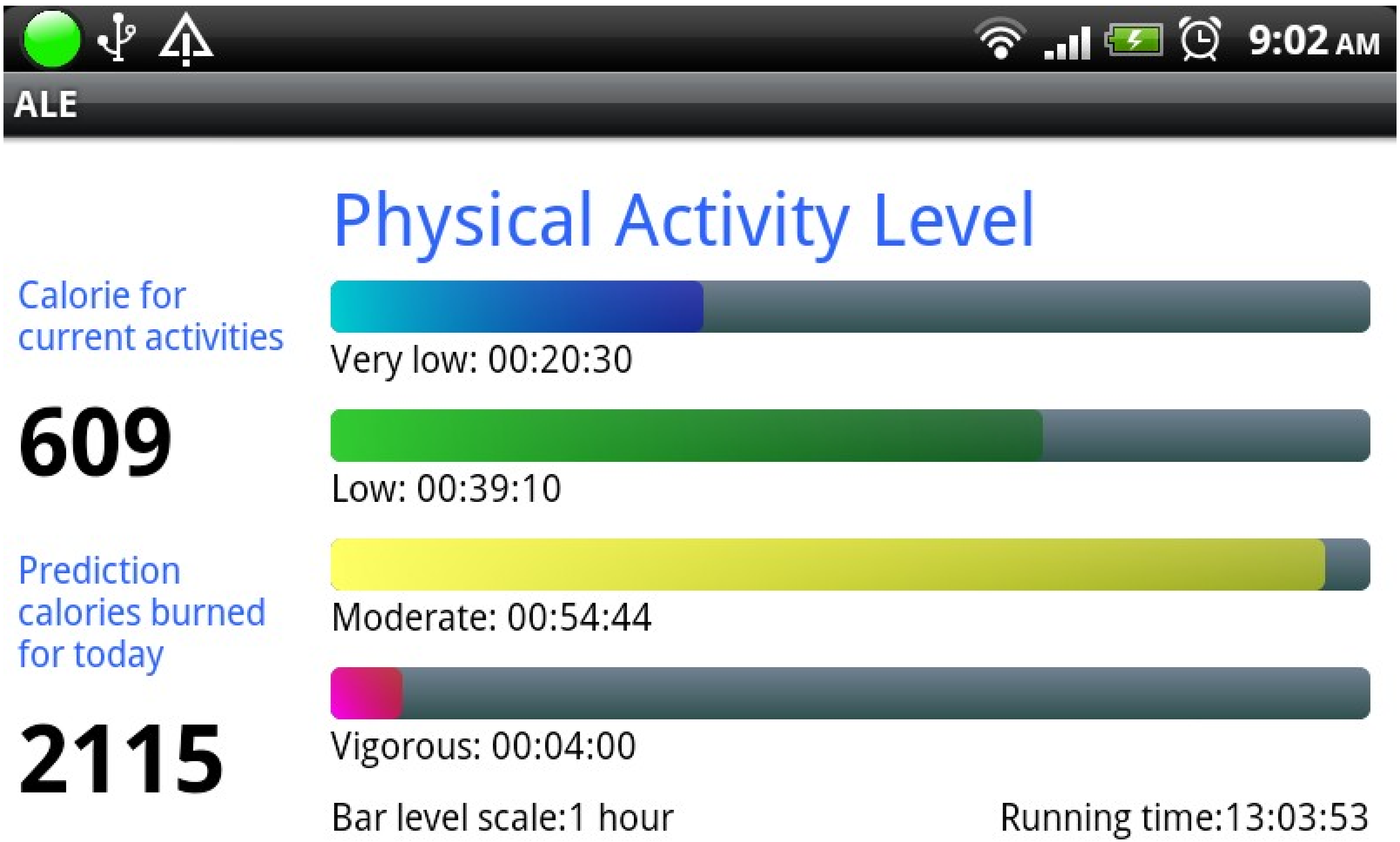
Example application: Activity Level Estimator (ALE)

- Current research at the University of Geneva
- An Android application made for users that want to monitor their level of physical activity during the whole day and not only during sports activities
- Detects the intensity of the body movement to determine the user's level of activity and calories burned
- Prototype actually developed at the University of Geneva and for the European project called TraiNutri

How it works

- The raw accelerometer signal is
  - filtered, analyzed, converted to kcal
- The accelerometer is running in the background as long as the application is active
- The battery consumption is high, but possible to have more than 22 hours per day in a normal use
- Tested and validated in a controlled lab vs Indirect Calorimetry (the gold standard in medicine)
- Average accuracy : 90.23% for walking activities
- http://www.qol.unige.ch/research/ALE.html

Activity Level Estimator



Room for improvement

- Low-hanging fruit: framework for “easy” signals like snap, shake and double-tap
  - iOS and bada are ahead of Android with these ones
- More difficult but important: exploit the sensor circuits' “wake on motion” feature to facilitate background motion analysis without excessive battery consumption
- Nice to have: less variance in sampling period times