Tilt Sensing Using a Three-Axis Accelerometer

Yue Sun

1 Introduction

- Accelerometers are sensitive to both linear acceleration and the local gravitational field.

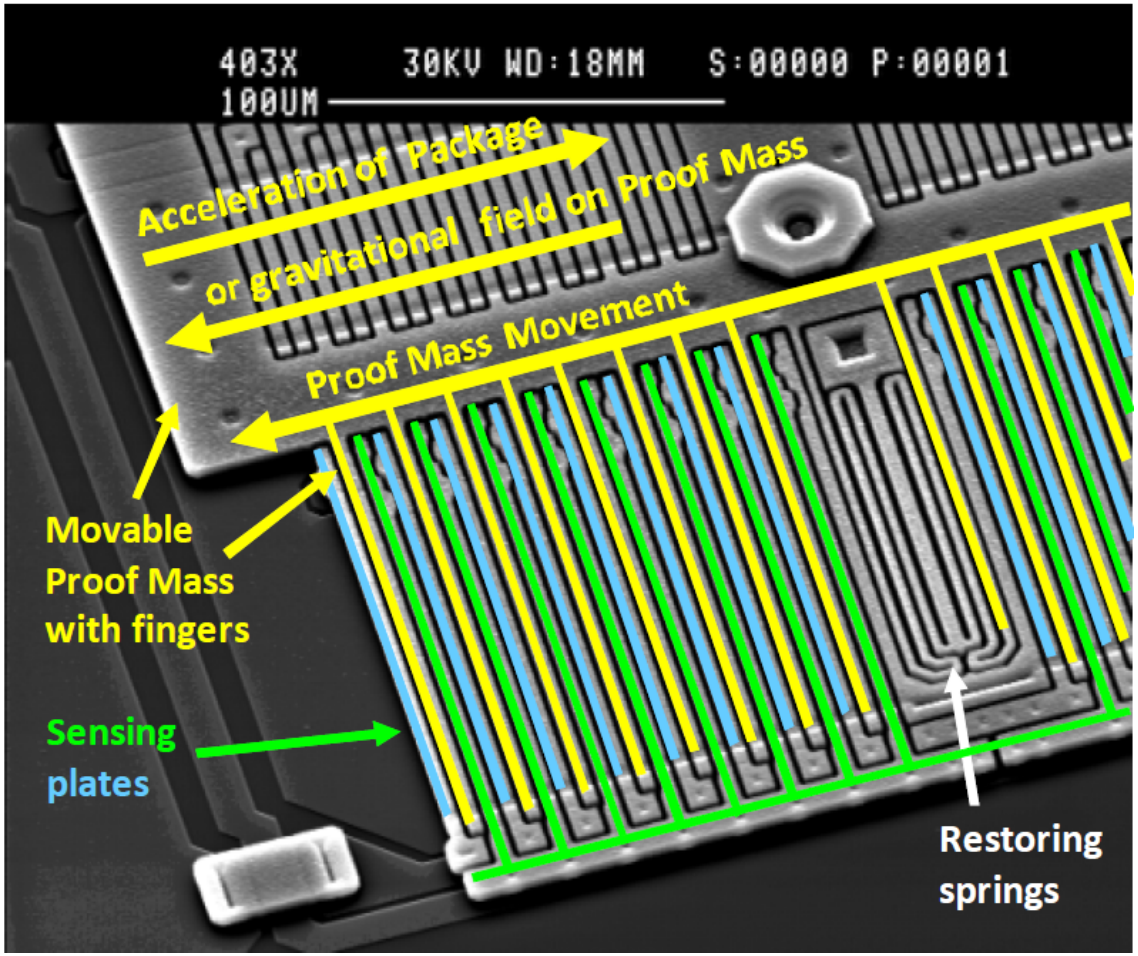
- The former allowing the development of 'gesture' user interfaces

- The latter allows a smart phone or tablet display to automatically switch between portrait and landscape settings.

2 Accelerometer Output Under Gravity and Acceleration

- A linear acceleration aligned in the direction of x, y and z axes will give a positive accelerometer output.

- Accelerometers are sensitive to the difference between the linear acceleration of the sensor and the local gravitational field.

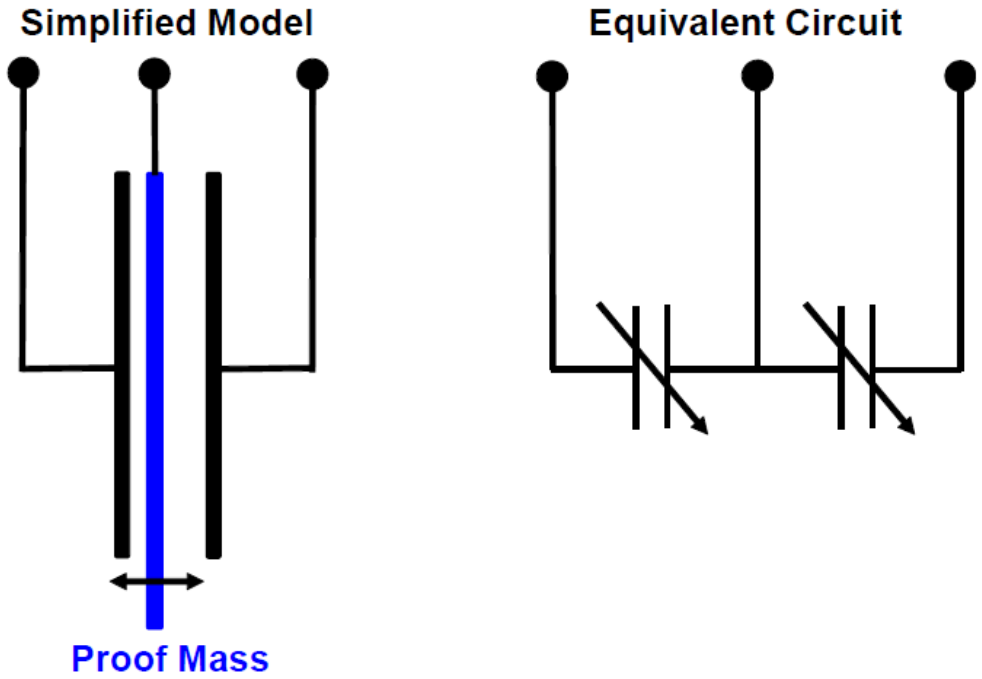


A gravitational field component aligned along the same axes directions will result in a negative reading on the accelerometer. The upper proof mass is suspended by the restoring springs.

Both a gravitational field directed to the left and a linear acceleration of the package to the right will deflect the proof mass to the left.

- The deflection of the proof mass is measured from the change in capacitance between the fingers of the proof mass and the sensing plates.

- Circuitry internal to the accelerometer sensor converts the tiny capacitance to a voltage signal, is then digitized and output as a digital word over a serial bus.



- Changes in orientation are described by rotations in roll ϕ, pitch θ and yaw ψ about the *x, y and z axes* respectively.

- The accelerometer output is negated to give value +1g in any axis aligned with the earth‘s downward gravitational field.
- With this assumption:

$$\boldsymbol{G}_p = \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \boldsymbol{R}(\boldsymbol{g} - \boldsymbol{a}_r)$$

R is the rotation matrix describing the orientation of the smart phone relative to the earth’s coordinate frame.

For further discussion we assume:

- The accelerometer has no linear acceleration .
- The initial orientation of the Smartphone is lying flat with the earth's gravitational field.

With these additional assumptions, the Smartphone accelerometer output is:

$$\boldsymbol{G}_p = \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \boldsymbol{R}\boldsymbol{g} = \boldsymbol{R}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

3 Pitch and Roll Estimation

- The orientation of the Smartphone can be defined by its roll, pitch and yaw rotations from an initial position.

$$\boldsymbol{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}$$

$$\boldsymbol{R}_y(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$\boldsymbol{R}_z(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

There are six possible orderings of these three rotation matrices.

The rotation matrices do not commute, meaning that the composite rotation matrix R depends on the order in which the roll, pitch and yaw rotations are applied.

$$\boldsymbol{R}_{xyz}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}-\sin\theta\\ \cos\theta\sin\phi\\ \cos\theta\cos\phi\end{pmatrix}$$

$$\boldsymbol{R}_{yxz}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi)\boldsymbol{R}_z(\psi)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}-\sin\theta\cos\phi\\ \sin\phi\\ \cos\theta\cos\phi\end{pmatrix}$$

It is conventional to select either the rotation sequence Rxyz or the sequence Ryxz to eliminate the yaw rotation ψ and allow solution for the roll ϕ and pitch θangles.

$$\boldsymbol{R}_{xzy}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_x(\phi)\boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}-\cos\psi\sin\theta\\ \cos\theta\sin\phi+\cos\phi\sin\psi\sin\theta\\ \cos\phi\cos\theta-\sin\theta\sin\phi\sin\psi\end{pmatrix}$$

$$\boldsymbol{R}_{yzx}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)\boldsymbol{R}_x(\phi)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}\cos\theta\sin\phi\sin\psi-\cos\phi\sin\theta\\ \cos\psi\sin\phi\\ \cos\theta\cos\phi+\sin\theta\sin\phi\sin\psi\end{pmatrix}$$

$$\boldsymbol{R}_{zxy}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_z(\psi)\boldsymbol{R}_x(\phi)\boldsymbol{R}_y(\theta)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}\cos\theta\sin\phi\sin\psi-\cos\psi\sin\theta\\ \cos\psi\cos\theta\sin\phi+\sin\theta\sin\psi\\ \cos\theta\cos\phi\end{pmatrix}$$

$$\boldsymbol{R}_{zyx}\begin{pmatrix}0\\0\\1\end{pmatrix} = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi)\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}\sin\phi\sin\psi-\cos\phi\cos\psi\sin\theta\\ \cos\psi\sin\phi+\cos\phi\sin\psi\sin\theta\\ \cos\theta\cos\phi\end{pmatrix}$$

It is not possible to solve for three unique values of the roll ϕ, pitch θ and yaw ψ angles.

The unknown yaw angle ψ represents the smartphone rotation from north but its determination requires the addition of a magnetometer sensor to create an eCompass.

Solving *Rxyz and Ryxz for the Pitch and Roll Angles*

- Rxyz can be rewritten in the form of relating the roll ϕ and pitch θ angles to the normalized accelerometer reading ***Gp:***

$$\frac{\boldsymbol{G}_p}{\|\boldsymbol{G}_p\|} = \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{pmatrix} \Rightarrow \frac{1}{\sqrt{G_{px}^{\ 2} + G_{py}^{\ 2} + G_{pz}^{\ 2}}} \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{pmatrix}$$

$$\tan\phi_{xyz} = \left(\frac{G_{py}}{G_{pz}}\right) \quad \tan\theta_{xyz} = \left(\frac{-G_{px}}{G_{py}\sin\phi + G_{pz}\cos\phi}\right) = \frac{-G_{px}}{\sqrt{G_{py}^{\ 2} + G_{pz}^{\ 2}}}$$

- For Ryxz it is the same:

$$\frac{\boldsymbol{G}_p}{\|\boldsymbol{G}_p\|} = \begin{pmatrix} -\sin\theta\cos\phi \\ \sin\phi \\ \cos\theta\cos\phi \end{pmatrix} \Rightarrow \frac{1}{\sqrt{G_{px}^{\ 2} + G_{py}^{\ 2} + G_{pz}^{\ 2}}} \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \begin{pmatrix} -\sin\theta\cos\phi \\ \sin\phi \\ \cos\theta\cos\phi \end{pmatrix}$$

$$\tan\phi_{yxz} = \frac{G_{py}}{\sqrt{G_{px}^{\ 2} + G_{pz}^{\ 2}}} \qquad \tan\theta_{yxz} = \left(\frac{-G_{px}}{G_{pz}}\right)$$

Eliminating Duplicate Solutions by Limiting the Roll and Pitch Ranges

Evaluating Rxyz, the accelerometer measurement for pitch angle π − θ and roll angle ϕ + π is the same as that resulting from rotations θ and ϕ.

$$\begin{pmatrix} -sin(\pi-\theta) \\ cos(\pi-\theta)sin(\phi+\pi) \\ cos(\pi-\theta)cos(\phi+\pi) \end{pmatrix} = \begin{pmatrix} -sin\,\theta \\ cos\,\theta sin\,\phi \\ cos\,\theta cos\,\phi \end{pmatrix}$$

Similarly, evaluating Ryxz for pitch angle θ + π and roll angle π − ϕ also shows that the accelerometer measurement is identical to that resulting from rotations θ and ϕ.

$$\begin{pmatrix} -sin(\theta+\pi)cos(\pi-\phi) \\ sin(\pi-\phi) \\ cos(\theta+\pi)cos(\pi-\phi) \end{pmatrix} = \begin{pmatrix} -sin\,\theta cos\,\phi \\ sin\,\phi \\ cos\,\theta cos\,\phi \end{pmatrix}$$

1) Solution: restrict roll or pitch angle between -90°and +90°.
2) Convention In the aerospace sequence, roll angle: [-180°,+180°] ,pitch :[-90° , 90°].
3) The convention used by Android™ smartphones and by Microsoft for its Windows 8 sensor platform: roll angle [-90°,90°], pitch angle [-180°,180°].

Regions of Instability

Both Equation $\tan\phi_{xyz} = \left(\frac{G_{py}}{G_{pz}}\right)$ and $\tan\theta_{yxz} = \left(\frac{-G_{px}}{G_{pz}}\right)$ have a region where the calculation of the roll angle ϕ and pitch θ angle respectively become unstable.

For the first one:
This condition occurs when Gpy and Gpz both equals to zero, it will produces an unstable and essentially random estimate of the roll angle. (smart phone is aligned with its x-axis vertically)

For the second one:
This condition occurs when Gpx and Gpz are zero. (smart phone is aligned with its y-axis vertically).

Solution:
Is to modify the one for the roll angle ϕ by mixing a fraction μ into the denominator to prevent the denominator ever being zero. The resulting equation is:

$$\tan\phi_{xyz} = \frac{G_{py}}{sign(G_{pz})\sqrt{G_{pz}^{2} + \mu G_{px}^{2}}}$$

sign(Gpz) has a value +1 if Gpz is non-negative and -1 if Gpz is negative.

- The roll angle error Δϕ is given by:

$$\Delta\phi = tan^{-1}\left(\frac{G_{py}}{\sqrt{G_{pz}{}^2 + \mu G_{px}{}^2}}\right) - tan^{-1}\left(\frac{G_{py}}{G_{pz}}\right) = tan^{-1}\left(\frac{sin\,\phi\left(cos\,\theta cos\,\phi - \sqrt{cos^2\theta cos^2\phi + \mu sin^2\theta}\right)}{cos\,\theta sin^2\phi + cos\,\phi\sqrt{cos^2\theta cos^2\phi + \mu sin^2\theta}}\right)$$

Figures below plot the absolute value of the resulting error |Δϕ| in degrees as a function of roll ϕ and pitch angles θ between -90° and 90° and with the contour mesh set to interval of 3° and μ equal to 0.01 and 0.1.

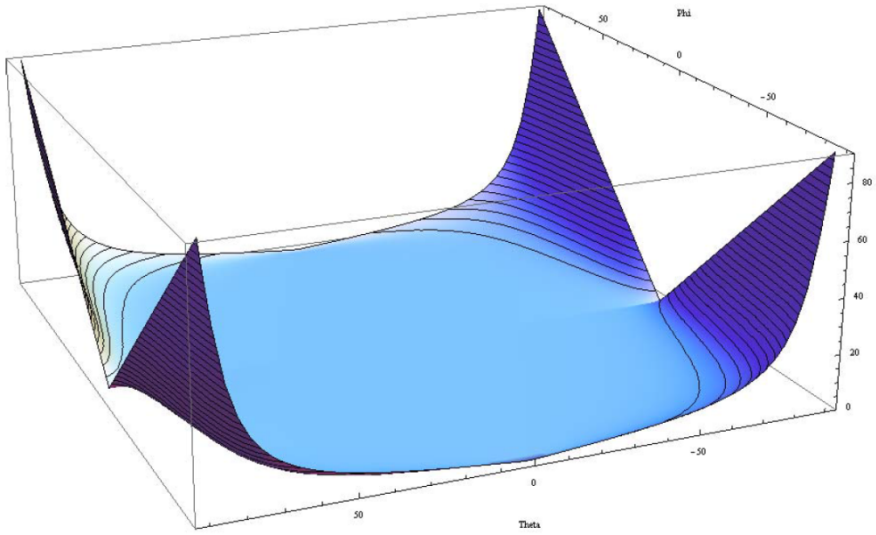


Figure 4. Roll Angle Error Surface $|\Delta\phi|$ for μ = 0.01

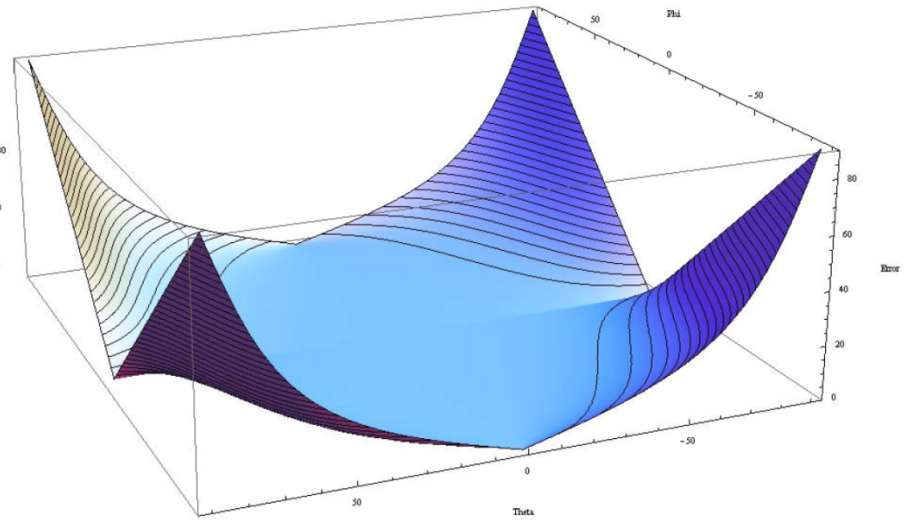


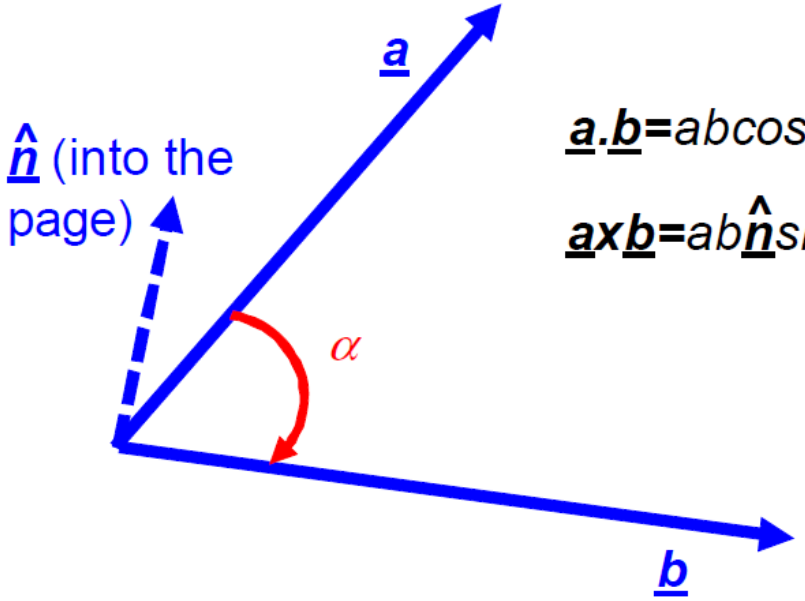
Figure 5. Roll Angle Error Surface $|\Delta\phi|$ for μ = 0.1

A similar strategy can be used for $tan\,\theta_{yxz} = \left(\frac{-G_{px}}{G_{pz}}\right)$

Calculating the Angle Between Two Accelerometer Readings

- By using scalar product:

$$\boldsymbol{a}.\boldsymbol{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} . \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$



$$\underline{a}.\underline{b} = ab\cos\alpha \qquad \frac{a_x b_x + a_y b_y + a_z b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}\sqrt{b_x^2 + b_y^2 + b_z^2}}$$

$$\underline{a}x\underline{b} = ab\hat{\underline{n}}\sin\alpha$$

- By using vector p

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$

$$\Rightarrow \hat{\boldsymbol{n}} \sin\alpha = \left(\frac{1}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right)\left(\frac{1}{\sqrt{b_x^2 + b_y^2 + b_z^2}}\right)\begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

Calculating the Tilt Angle

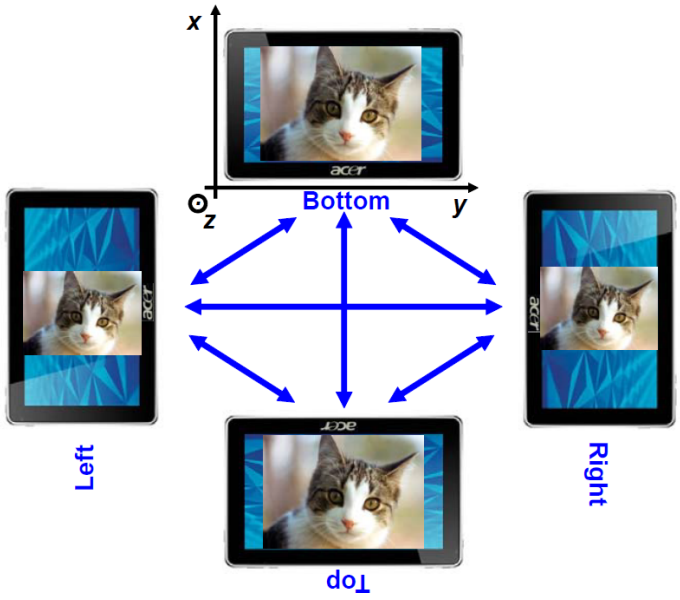
- The techniques of the previous section can be used to calculate the angle ρ between the gravitational vector measured by the accelerometer and the initial orientation with the gravitational field pointing downwards along the *z-axis.*

If the accelerometer reading is Gp, then in the absence of linear acceleration:

$$\boldsymbol{G}_p \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = G_{pz} = \left|\boldsymbol{G}_p\right| \cos\rho \Rightarrow \cos\rho = \frac{G_{pz}}{\sqrt{{G_{px}}^2 + {G_{py}}^2 + {G_{pz}}^2}}$$

6 Selecting Portrait and Landscape Modes

- The first high volume use of accelerometers in consumer products was to switch a display between landscape and portrait orientations.



Problem: a direct mapping between the display orientation and the accelerometer reading is not ergonomically suitable

Solution: Simple state transition rules are listed below.

- (|*Gpz*| < 0.5g) AND (*Gpx* > 0.5g) AND (|*Gpy*| < 0.4g): *Change orientation to Top*
- (|*Gpz*| < 0.5g) AND (*Gpx* < -0.5g) AND (|*Gpy*| < 0.4g): *Change orientation to Bottom*
- (|*Gpz*| < 0.5g) AND (*Gpy* > 0.5g) AND (|*Gpx*| < 0.4g): *Change orientation to Right*
- (|*Gpz*| < 0.5g) AND (*Gpy* < -0.5g) AND (|*Gpx*| < 0.4g): *Change orientation to Left.*

|*Gpz*| < 0.5g *known as z-axis lockout* ensures that the screen orientation remains unchanged as the user lowers the tablet to a flat orientation.

A Comparison of Pedestrian Dead-Reckoning Algorithms using a Low-Cost MEMS IMU

introduction

- Local Positioning Systems:
- Sensor network technology
- Ultrasound, radio or vision technology

- Pedestrian Dead-Reckoning solutions:
- Do not depend on a pre-installed infrastructure
- Use Inertial Measuring Units (IMU)
- Integrate step lengths and orientation estimations at each detected step, so as to compute the absolute position and orientation of a person

Some PDR approaches assume a smooth walk on horizontal surfaces, and others are valid for uneven terrain with complicated gait patterns. PDR has been proposed for a large range of applications, such as defense, emergency rescue workers, smart offices, etc.

In this paper…

- They use low-performance Micro-Electro-Mechanical (MEMS) inertial sensors attached to the foot of a person.

- This sensor has triaxial orthogonal accelerometers, gyroscopes and magnetometers.

- They describe, implement and compare several of the most relevant algorithms for step detection, stride length, heading and position estimation.

Challenge: MEMS sensors have a significant bias and therefore suffer large drifts after integration.

IMU

- The IMU has three orthogonally-oriented accelerometers, three gyroscopes and three magnetometers.

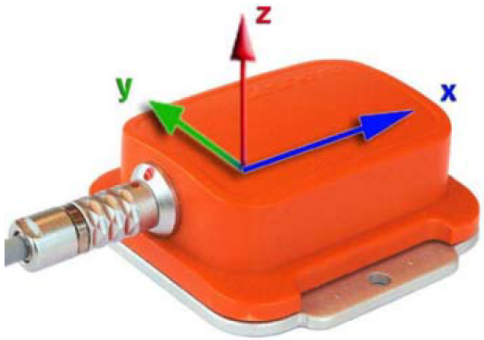


Fig. 1. MTi Xsens IMU with annotated sensor cartesian coordinates.

	accelerometers	gyroscopes	magnetometers
Axes	3	3	3
Full Scale (FS)	$\pm 50\ m/s^2$	$\pm 300\ deg/s$	± 750 mGauss
Linearity	0.2% of FS	0.1% of FS	0.2% of FS
Bias stability	$0.02\ m/s^2$	$1\ deg/s$	0.1 mGauss
Bandwidth	30 Hz	40 Hz	10 Hz
Max update rate	512 Hz	512 Hz	512 Hz

TABLE I
PERFORMANCE OF INDIVIDUAL SENSORS IN XSENS IMU

The MTi sensor has a built-in algorithm that provides the absolute heading and attitude of the unit(rotation matrix RGS).

RGS can transform the readings from the sensor (S) to the global (G) cartesian coordinates frames.

Static accuracy (roll/pitch)	<0.5 deg
Static accuracy (heading)[1]	<1 deg
Dynamic accuracy	2 deg RMS
Angular resolution	0.05 deg

[1] in homogeneous magnetic environment

TABLE II
PERFORMANCE OF ATTITUDE AND HEADING AS PROVIDED BY XSENS FUSION ALGORITHM IN MATRIX $\mathbf{R}_{GS}$.

Performance is quite good whenever the earth magnetic field is not disturbed

IMU placement

- Waist or trunk
- less intrusive
- the most reliable position for heading estimation using gyroscopes or magnetometers.

- Foot
- It is applicable the zero velocity update (ZUPT) strategy to diminish drifts after integrating accelerations.
- the step detection is robustified.

Fig. 2. Xsens IMU attached to the right foot using the shoe's laces.

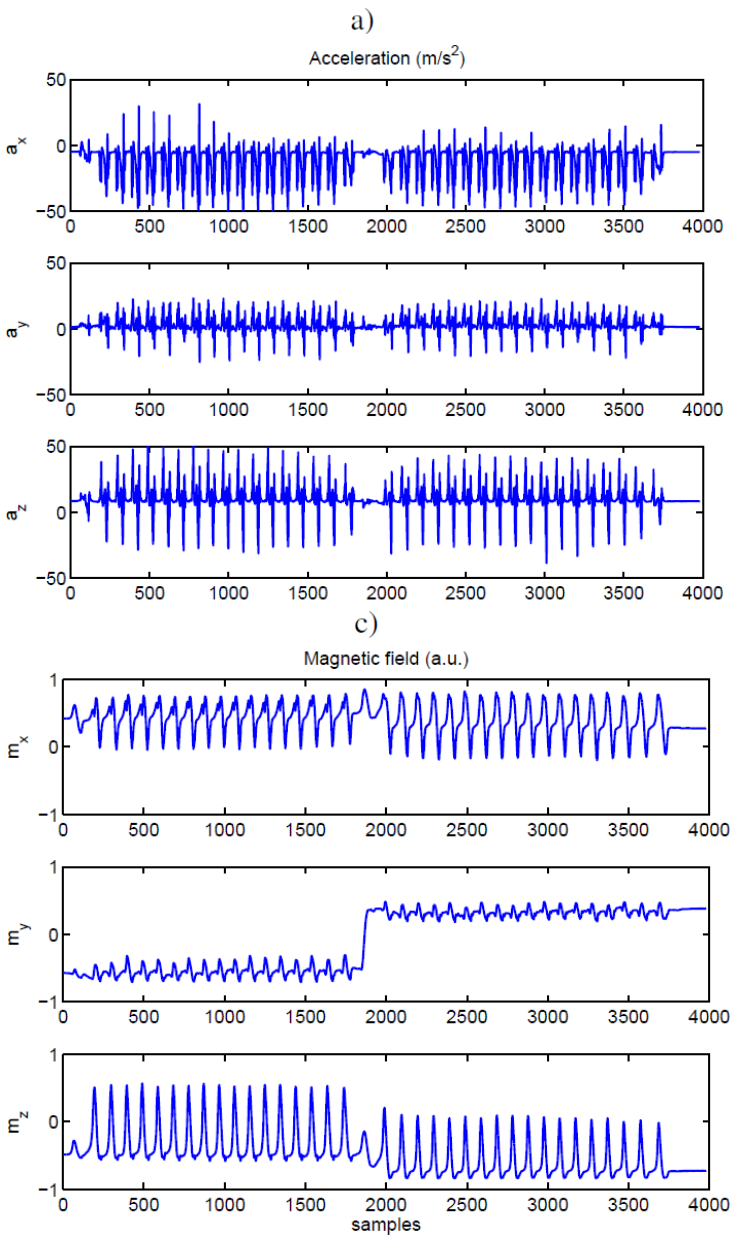
STEP DETECTION

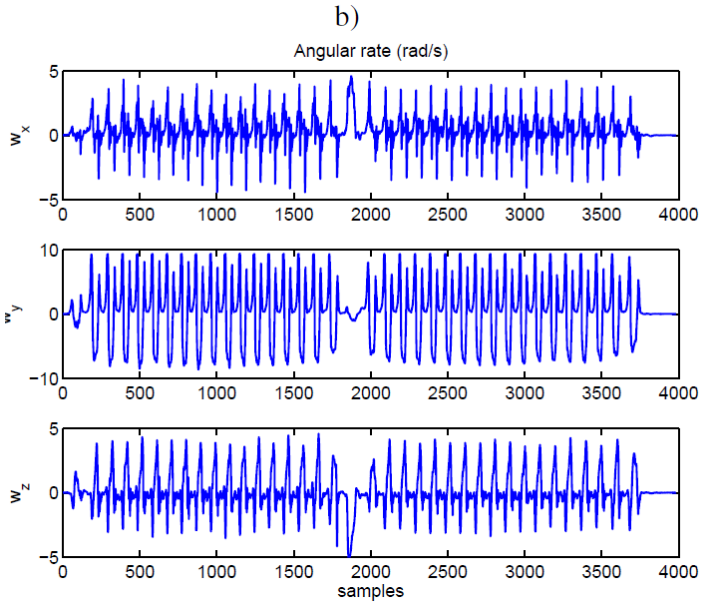
- **Preliminary walking tests:**
- 1) A one-way straight walk (120 meters long)

- 2) A go and return walk (60 m long with a 180o turn)

- 3) A rectangular path (30 x 5 m; 70 meters in total)
- Each of them were done at three different walking speed (slow, normal, and fast), and repeated three times to check the repetitivity of results.

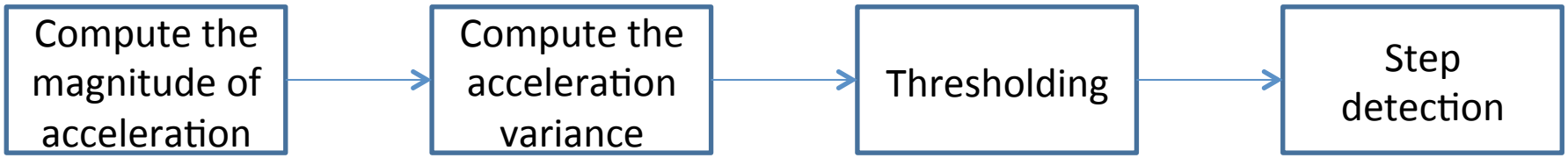
- Captured Signals





the signal magnitudes are very low at the start and the end of the motion; this makes it difficult to detect all steps using simple threshold-based algorithms.

Step Detection using accelerations



1) Compute the magnitude of the acceleration, a_i, for every sample i:

$$a_i = \sqrt{a_{x_i}^2 + a_{y_i}^2 + a_{z_i}^2}. \quad (1)$$

- So that the placement of the sensor won't affect the result

2) Compute the local acceleration variance, to highlight the foot activity and to remove gravity:

$$\sigma_{a_i}^2 = \frac{1}{2w+1} \sum_{j=i-w}^{i+w} (a_j - \overline{a_j})^2, \quad (2)$$

where $\overline{a_j}$ is a local mean acceleration value, computed by this expression: $\overline{a_j} = \frac{1}{2w+1} \sum_{q=i-w}^{i+w} a_q$, and w defines the size of the averaging window (w=15 samples).

- The fixed size of averaging window is not good!

3) Thresholding. A first threshold is applied to detect the swing phase with high accelerations (T1=2 m/s^2).

$$B_{1i} = \begin{cases} \text{T1} & \sigma_{a_i} > \text{T1} \\ 0 & \text{otherwise} \end{cases} . \qquad (3)$$

A second threshold (T2=1 m/s^2) is used to detect the stance phase ($B_{2i} = \text{T2}$, if $\sigma_{a_i} < \text{T2}$).

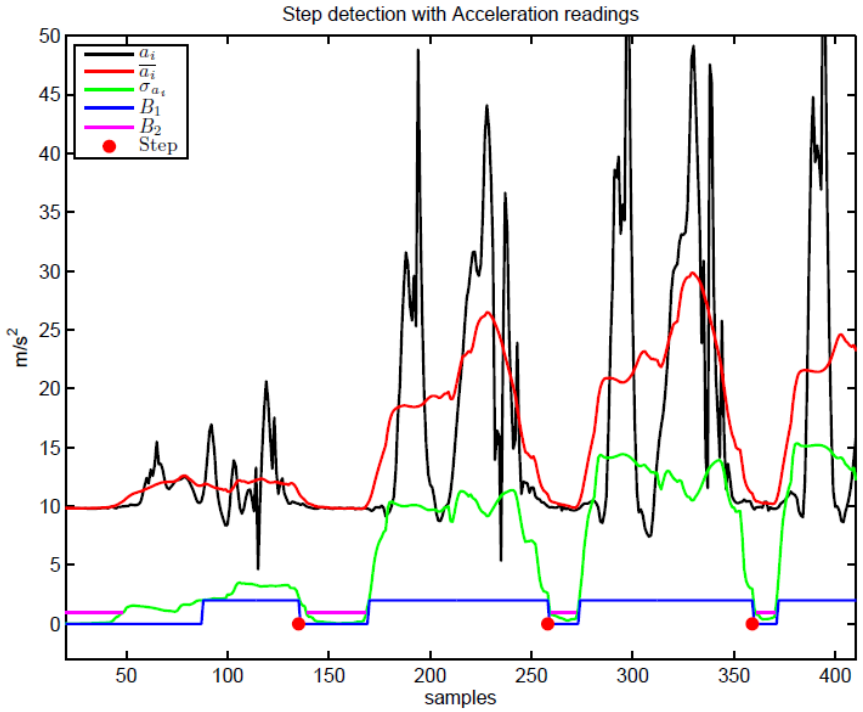


Fig. 4. Steps detection using accelerations. The detected steps are marked with red circles. Additional line plots represent intermediate processing values as explained in equations 1 to 3.

Disadvantage:
The fixed threshold is not good too.
The speed of walking have a strong influence on the accuracy of the result.

- A step is detected in sample i when a swing phase ends and stance phase starts.
- These two conditions must be satisfied:
- A transition from high to low acceleration
- There must be at least one low acceleration detection in a window of size w ahead of current sample i, i.e.: max(B2i:i+w) = T2.

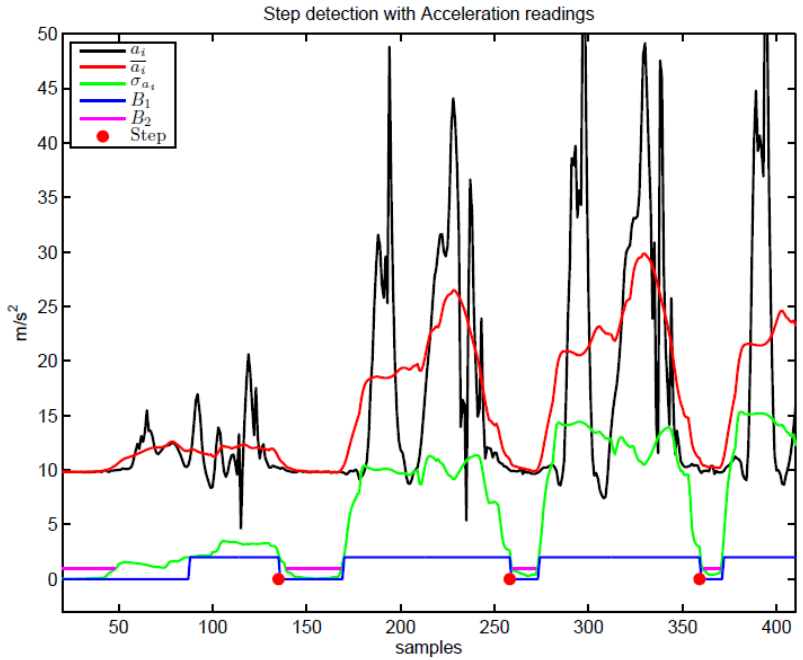


Fig. 4. Steps detection using accelerations. The detected steps are marked with red circles. Additional line plots represent intermediate processing values as explained in equations 1 to 3.

Step detection using gyroscopes and magnetometers

- Gyroscopes:
- Computes the total angular rate magnitude using the three individual gyroscopic sensors,
- Performs a threshold at 1 rad/s
- Applies a median filter to remove outliers
- Detects transitions to a motionless state.

- Magnetometer:
- Starts with a high pass filter for removing the DC components of the magnetic field
- The remaining is the same as last algorithm.

Results

Sensor & Algorithm type	Real number of steps	Number of steps failed	Percentage of errors
Acceleration	955	1	0.1%
Gyroscopes	955	2	0.2%
Magnetometric	955	9	0.94%

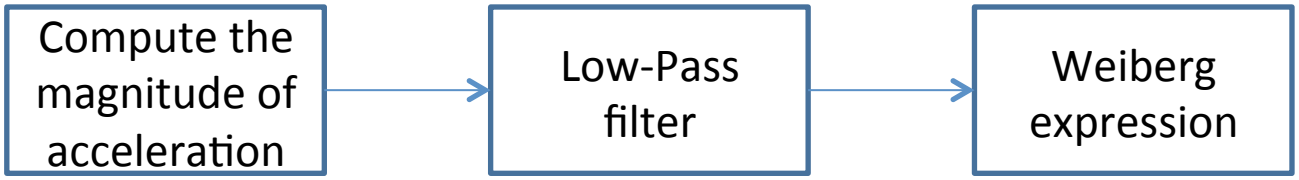
TABLE III
PERFORMANCE OF THREE STEP DETECTION ALGORITHMS

- The step detection is very reliable at continuous walk, but it is more difficult to detect steps robustly at the beginning/end of motion, and at very low speeds (e.g. museum-like style of walking).

STRIDE LENGTH ESTIMATION

- There are three main procedures to estimate a SL:
- 1) Foot-to-foot range measuring devices (Not implementable just with IMUs).
- 2) Modelling human gait with inertial measurements.
- 3) Foot inertial integration (INS) with zero velocity updates (ZUPT) at stance detections

- **The Weiberg SL Algorithm:**
- It assumes that SL is proportional to the bounce, or vertical movement, of the human hip. This hip bounce is estimated from the largest acceleration differences at each step.



1) Compute the magnitude of the acceleration, a_i, for every sample i:

$$a_i = \sqrt{a_{x_i}^2 + a_{y_i}^2 + a_{z_i}^2}. \quad (1)$$

2) Low-Pass filter this signal ($\tilde{a}_i = LP(a_i)$). We use a filter of order 4 and cut-off frequency at 3 Hz.

- To remove the noise in the signal

3) Estimate the SL using the Weiberg expression:

$$\mathrm{SL}_{\mathrm{Weiberg}_k} = K \cdot \{\max_{j=[i_{(k)} \pm w]}^{\tilde{a}_j} - \min_{j=[i_{(k)} \pm w]}^{\tilde{a}_j}\}^{1/4}, \quad (4)$$

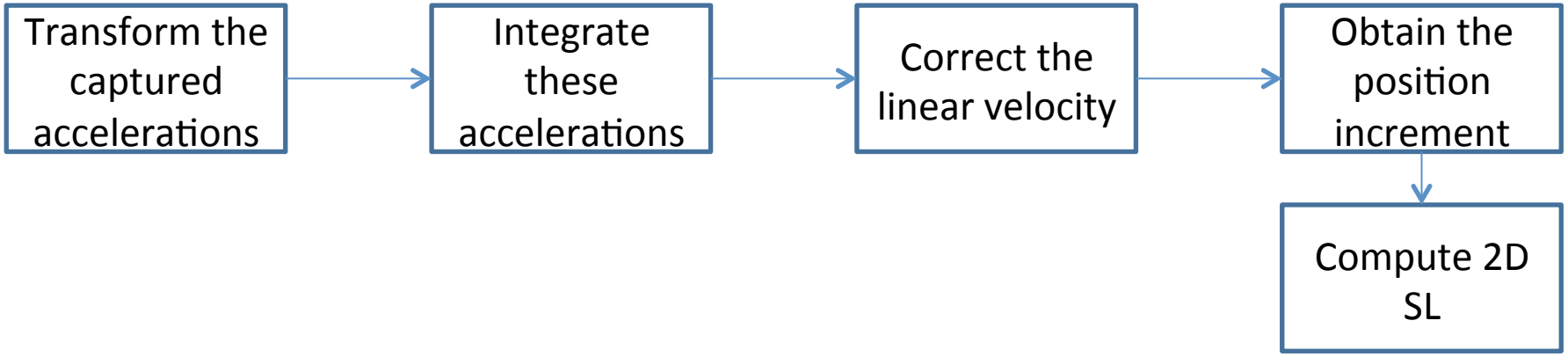
- The maximum and minimum operations are applied over the filtered accelerations ˜aj in a window of size 2w+1 around the sample i(k) corresponding to the k stance detection.

- K is a constant that has to be selected experimentally or calibrated.

- The ZUPT SL Algorithm (can be used during walk, run, side walk, cris-cross, climb):

- This inertial mechanization method performs zero velocity updates every time a step is detected.

- At foot stance the velocity is known to be zero, so the idea is to correct the linear velocities obtained after integrating the accelerometer.



1) Transform the captured accelerations, a_i refered to the sensor frame (S), into the global north-east frame (G), using the rotation matrix: $a_i^G = \mathbf{R}_{GS} \cdot a_i$

2) Integrate these accelerations, a_i^G, to obtain the linear velocities v_i^G. Thats is: $v_i^G = v_{i-1}^G + a_i^G / f_s$, where f_s is the sampling frequency (100 Hz).

3) Correct the linear velocity v_i^G from drift by using the ZUPT update at every stance event:

a) Compute the mean velocity value, μ, around the stance event, k

$$\mu_k = \sum_{j=i_{(k)}-w}^{s_{(k)}+w} v_j^G / (2w + 1)$$

1) i(k) represents the sample index of the k stance occurrence.
2) μk represents the velocity error accumulated at step k (it should be zero).

b) Correct all the velocity samples in the whole step.

$$\check{v}_i^G = v_i^G - [\mu_k (i - i_{(k-1)}) + \mu_{k-1} (i_{(k)} - i)] / m_k$$

1) mk is the number of samples in step k.

4) Obtain the position increment at step k

$$\Delta P_k = (\Delta P_k(north), \Delta P_k(east), \Delta P_k(up))$$

$$\Delta P_k = \sum_{j=i_{(k-1)}}^{i_{(k)}} \breve{v}_j^G / f_s.$$

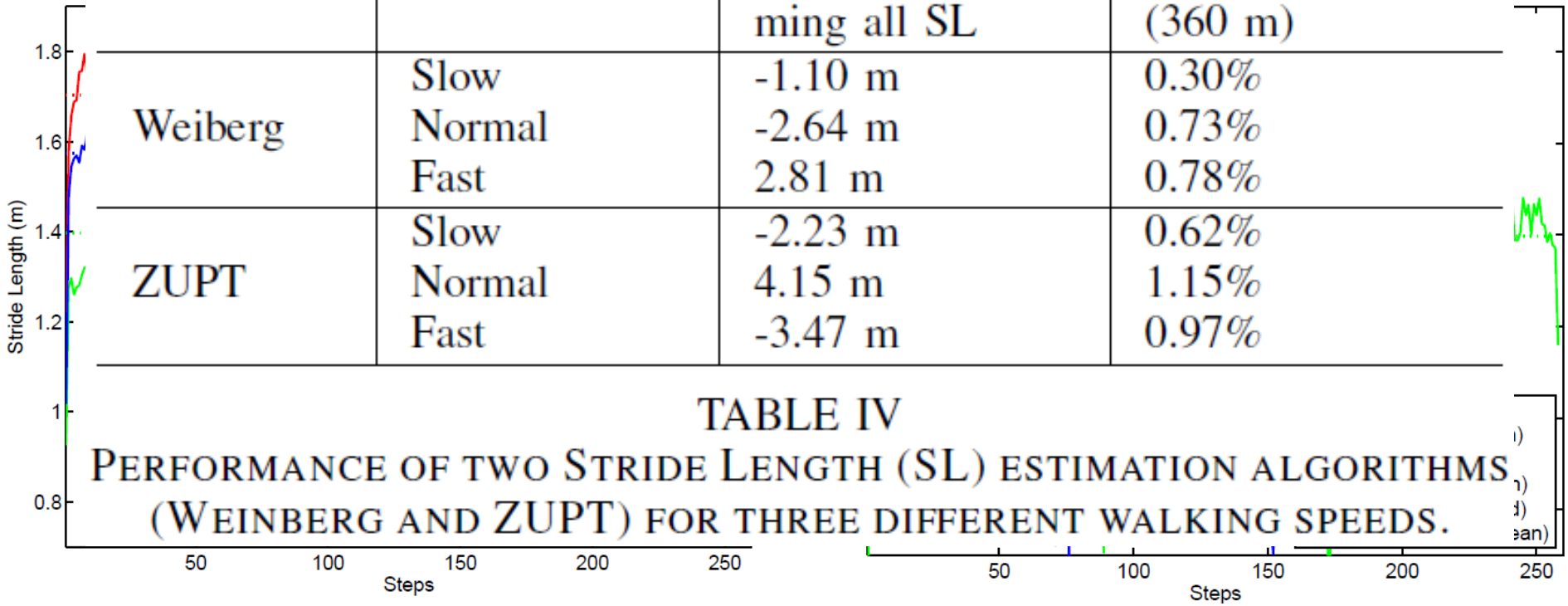
5) The 2D SL is computed by taking the horizontal cartesian distance of the position increment:

$$\mathrm{SL}_{\mathrm{ZUPT}_k} = \sqrt{\Delta P_k(north)^2 + \Delta P_k(east)^2}. \quad (6)$$

Results

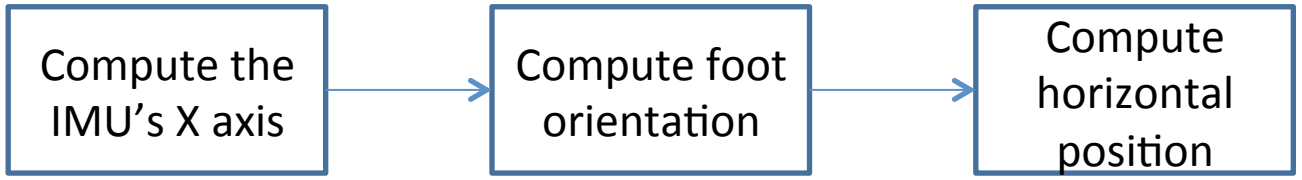
SL algorithm	Walking speed	Error of the total distance calculated by summing all SL	Error in percentage of the total travelled distance (360 m)
Weiberg	Slow	-1.10 m	0.30%
	Normal	-2.64 m	0.73%
	Fast	2.81 m	0.78%
ZUPT	Slow	-2.23 m	0.62%
	Normal	4.15 m	1.15%
	Fast	-3.47 m	0.97%

TABLE IV
PERFORMANCE OF TWO STRIDE LENGTH (SL) ESTIMATION ALGORITHMS (WEINBERG AND ZUPT) FOR THREE DIFFERENT WALKING SPEEDS.



Weinberg methodology, using a fixed K value, is valid for accurately estimating SL even at different walking speeds. The ZUPT method gives also good results with a larger variation between consecutive SL; this is because ZUPT method does not filter any signal

2D POSITION ESTIMATION



1) Compute the IMU's X axis in global coordinates, for every detected step k at stance occurrence i(k).

$$X^{G}_{\mathrm{IMU}_{i_{(k)}}} = \mathbf{R}_{GS_{i_{(k)}}} \cdot [1,0,0]$$

2) Assuming that IMU X axis is aligned with the foot heading, then the foot orientation at stance is:

$$\theta_{\mathrm{stance}_k} = arctan\left(\frac{-X^{G}_{\mathrm{IMU}_{i_{(k)}}}(\mathrm{west})}{X^{G}_{\mathrm{IMU}_{i_{(k)}}}(\mathrm{north})}\right). \qquad (7)$$

It will cause errors when the X axis is not perfectly aligned with the foot heading

3) The horizontal position $P_k = (P_k(\text{north}), P_k(\text{east}))$ of the foot at step k is computed as

$$\begin{cases} P_k(\text{north}) = P_{k-1}(\text{north}) + \text{SL}_k \cdot cos(\theta_{\text{stance}_k}) \\ P_k(\text{east}) = P_{k-1}(\text{east}) + \text{SL}_k \cdot sin(\theta_{\text{stance}_k}), \end{cases} \quad (8)$$

We get SL from previous algorithms

The third INS-ZUPT-based algorithm for foot position estimation just accumulates the positioning increments, ΔPk,

c $\Delta P_k = \sum_{j=i_{(k-1)}}^{i_{(k)}} \breve{v}_j^G / f_s$. ation:

$$\begin{cases} P_k(\text{north}) = P_{k-1}(\text{north}) + \Delta P_k(\text{north}) \\ P_k(\text{east}) = P_{k-1}(\text{east}) + \Delta P_k(\text{east}). \end{cases}$$

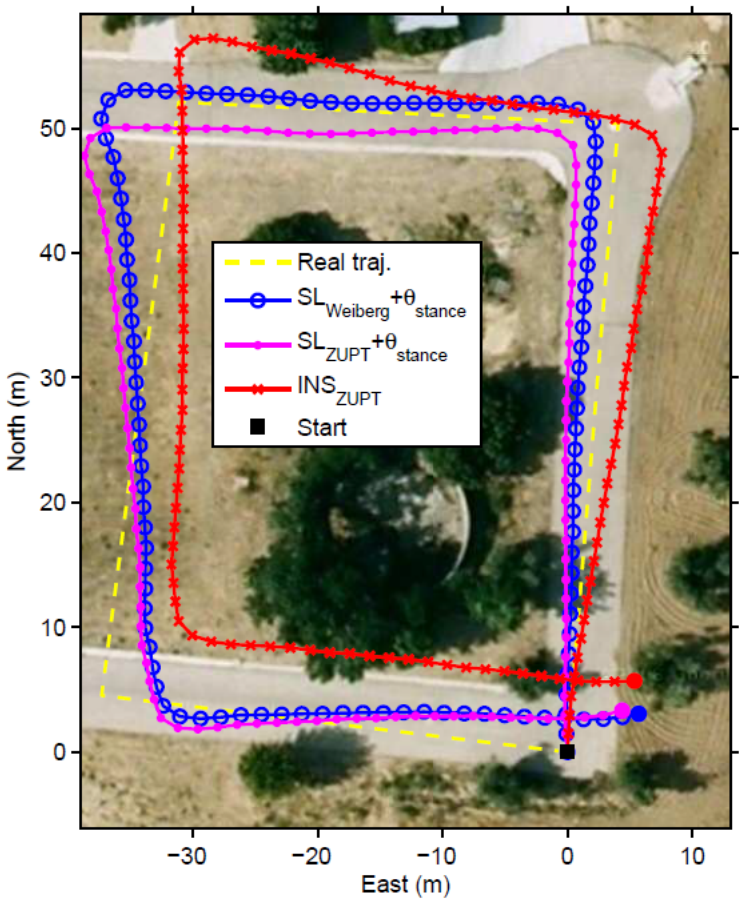


Fig. 6. An outdoors 160 meter-long closed-CCW trajectory and three position estimations using different PDR algorithms.

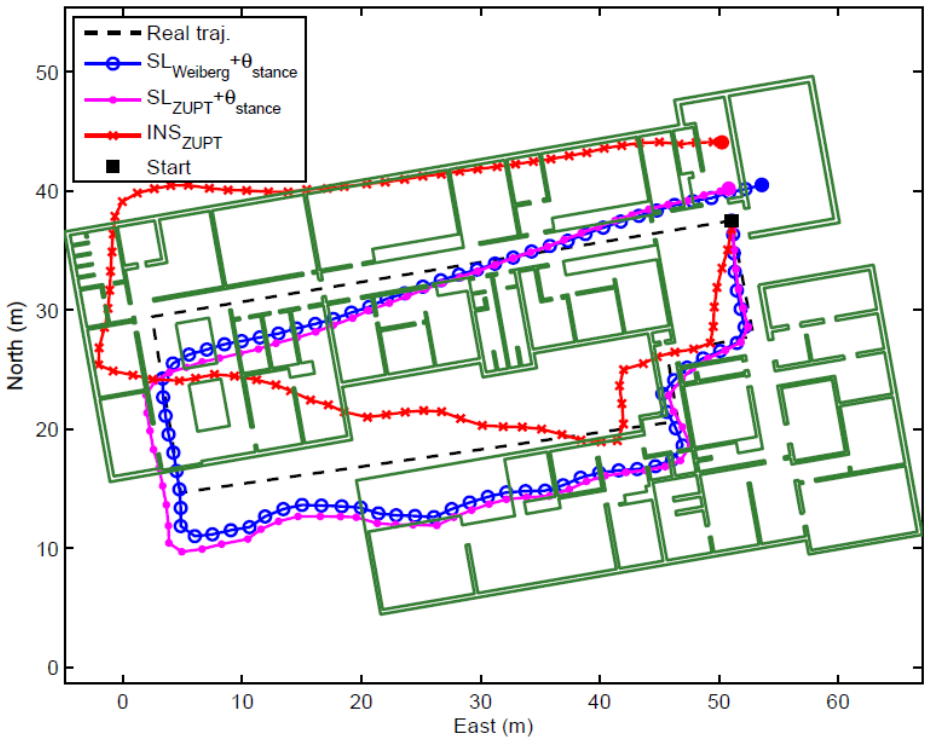


Fig. 7. A trajectory along the main building of IAI-CSIC and three position estimations using different PDR algorithms.

1) Orientation error in the SL and foot orientation based methods can come from the misalignment between the foot and the sensor X axis.
2) The INS-ZUPT method uses all orientation samples in a step, however the SL + θ methods use the orientation at one step sample (the sample at stance detection). The orientation during swing phase are not as reliable as orientation at stance

Conclusion

- The algorithms (error below 5%) over come the low performance of MEMS-based IMU sensors.
- The INS-ZUPT positioning method is suitable for many situations, however other methods, based on SL plus foot orientations at stance give better performance in smooth surfaces.

Automatic Step Detection in the Accelerometer Signal

- **Pan-Tompkins method：**
- This algorithm includes a series of filters and methods that perform low-pass, derivative, squaring, integration for preprocess and adaptive thresholds for peak searching.

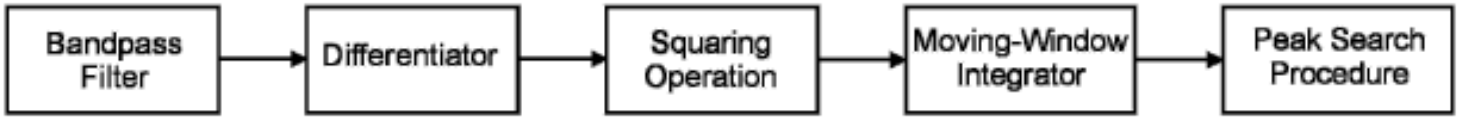


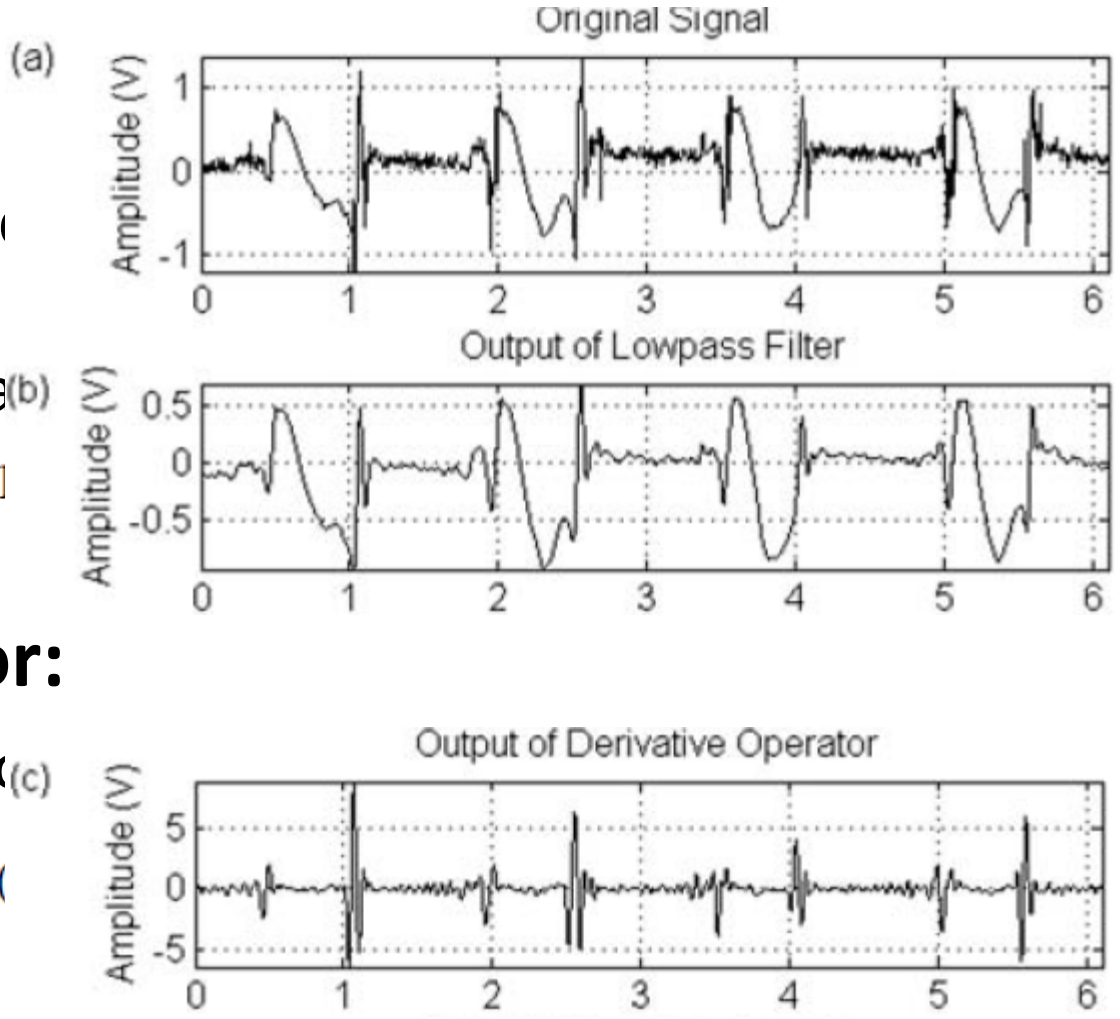
Fig. 3 Block diagram of the Pan-Tompkins algorithm

- **Bandpass-filter:**
- The bandpass filter re
  the signal. The followi
  frequency of 20 Hz wa

$$H(z) = \frac{1}{16}\frac{(1-z^{-4})^2}{(1-z^{-1})^2} \qquad (1$$

- **Drivative operator:**
- The derivative operat

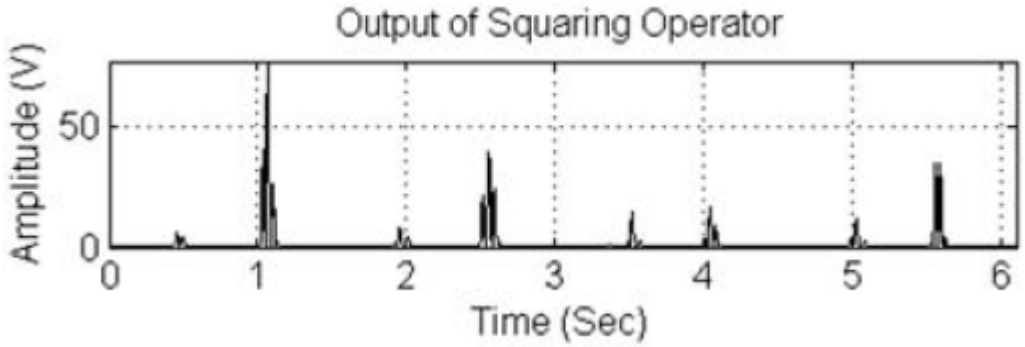
$$y(n) = \frac{1}{8}[2x(n) + x(n-1) - x($$



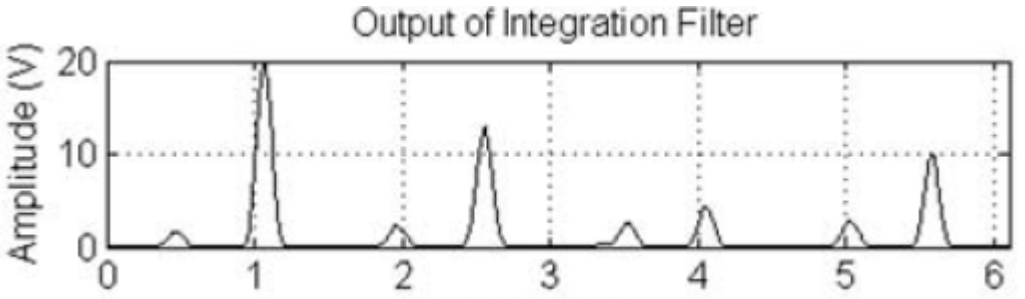
- It suppresses the low-frequency components and enlarge the high frequency components from the high slopes.

- **Squaring:**
- The squaring op
and enhances la
values.

(d)



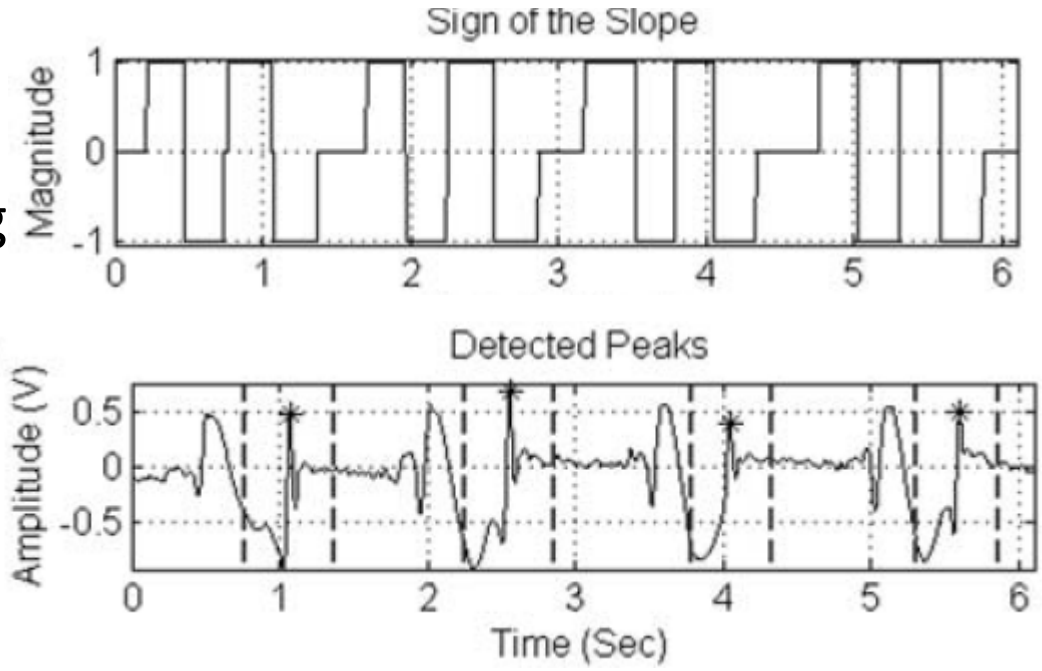
- **Integration:**
- The output result
moving-window i



$$y(n) = \frac{1}{N}[x(n-(N-1)) + \ldots$$
$$x(n-(N-2)) + \cdots + x(n)] \qquad (3)$$

- where N is chosen to be 20 empirically.

- **Fiducial mark:**
- The peak detection using solution.



- The property of the succ transformed signal is util

- One step cycle signal is converted into a pair of consecutive 1 and -1 value, i.e., [1 -1 1 -1]
- The second [1 -1] interval, corresponding to the onset and offset of the second peak in Fig 4 (e) is defined as peak-searching interval.
- The local maximum within the peak-searching interval on the filtered signal is detected as step.

- **Template-matching method:**
- The main concept of the template-matching method is to generate a template, which represents a typical step cycle.

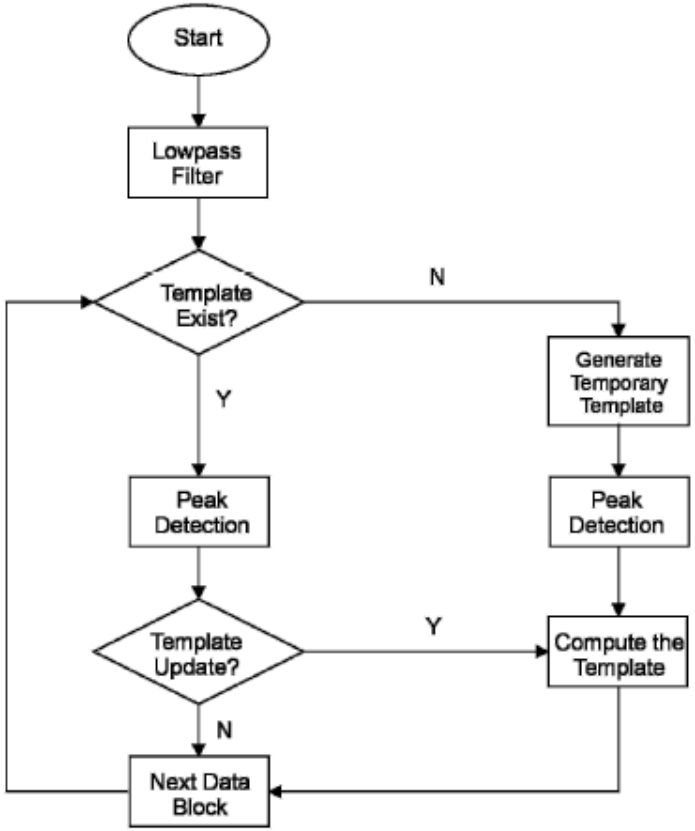
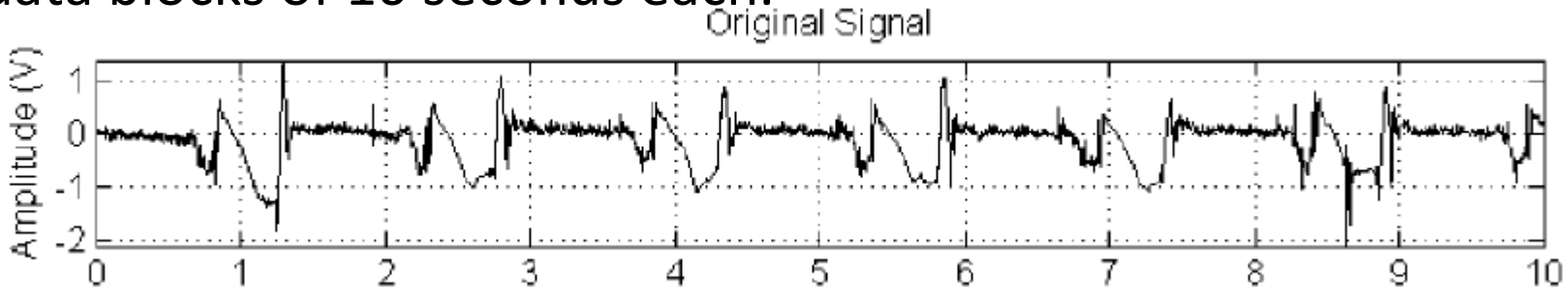


Fig. 5 Flowchart of the template matching method

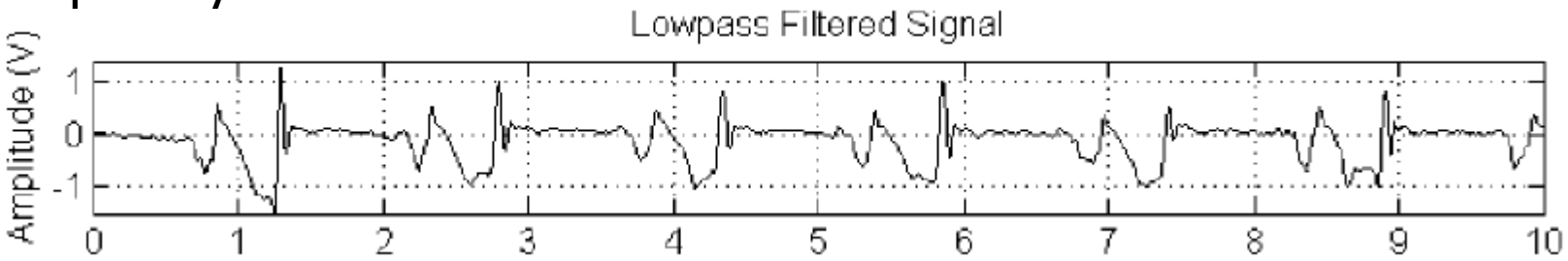
- the whole recording is broken into several non-overlapping data blocks of 10 seconds each.

(a)



- Then the signal is filtered by a low-pass filter with cutoff frequency of 20 Hz.

(b)



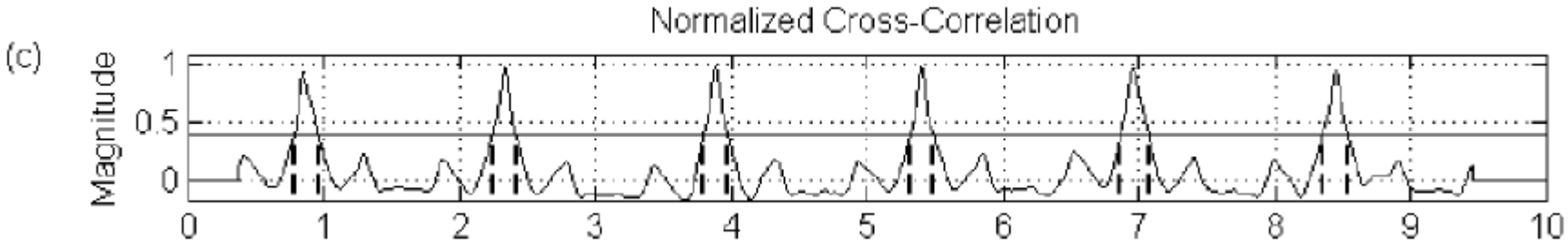
- Next, the algorithm will check whether there is any template present in memory. If not present, the first step cycle is extracted as a temporary template.

- This template is slid across the whole data block and the normalized cross-correlation, is calculated between the template and signal.

- The normalized cross-correlation indicates the similarity between two vectors X and Y, which is given

$$R_N[k] = \frac{< X, Y >}{\sqrt{\|X\| \cdot \|Y\|}} = \frac{R_{XY}(k)}{\sqrt{R_{XX}(0) \cdot R_{YY}(0)}} \qquad (4)$$

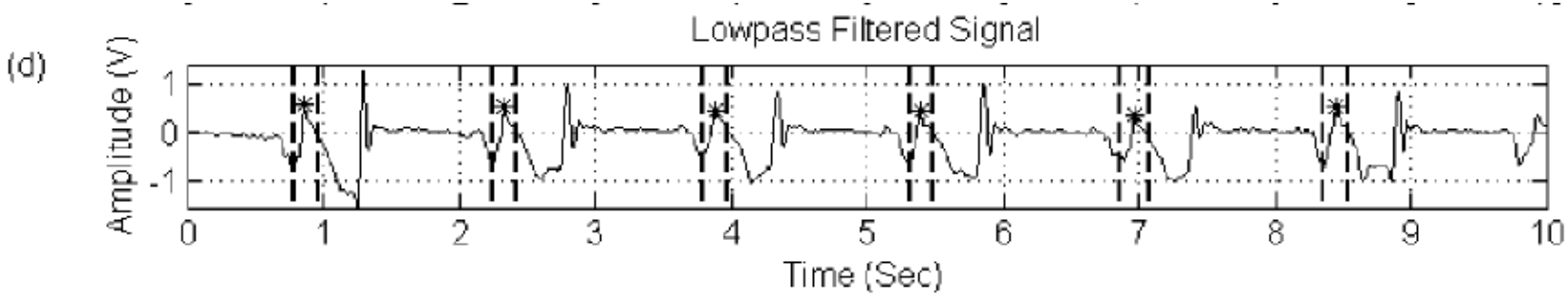
where $< X \cdot Y >$ is the inner product of X and Y, $\|X\|$ is the norm of the Vector X, $R_{XY}(k)$ is cross-correlation of X and Y for arbitrary k, and $R_{XX}(0)$ is autocorrelation of X[n] at point of zero.



- The peaks in the normalized cross-correlation indicate great similarity between the step template and signal segment, thus the occurrence of the event.

- The interval, in which the cross-correlation exceeds the threshold 0.4, is defined as peak-searching interval.

- The local maxima falling within the peak-searching interval in the filtered signal are marked as fiducial points of the steps



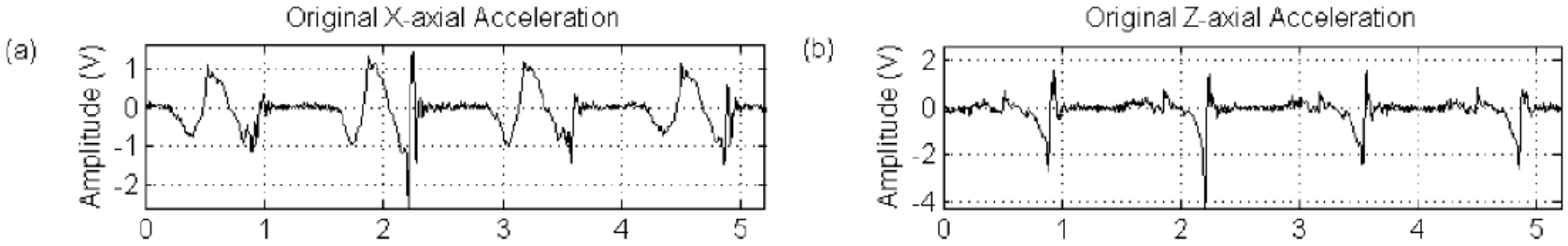
- Before the next data block is processed, a determination need to be made as to whether the template will be updated.

- The step signal may change dynamically with time.

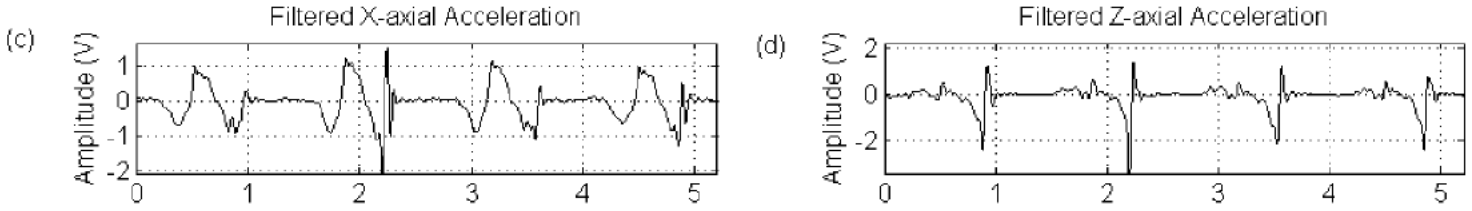
- If the major peaks in the normalized cross-correlation are lower than 0.55, the new template is generated using the steps cycles in the current data block.

- **Peak-detection method based on combined dual-axial signals:**
- Idea: Observe from the acceleration signal of each step that the negative wave in the x-axial signal occurs coincident with the negative wave in the z-axial signal.

Fig. 8 Block diagram of the peak detection algorithm on the combined dual-axial signal



- At first, both signals are passed through a lowpass filter with cutoff frequency 20 Hz.

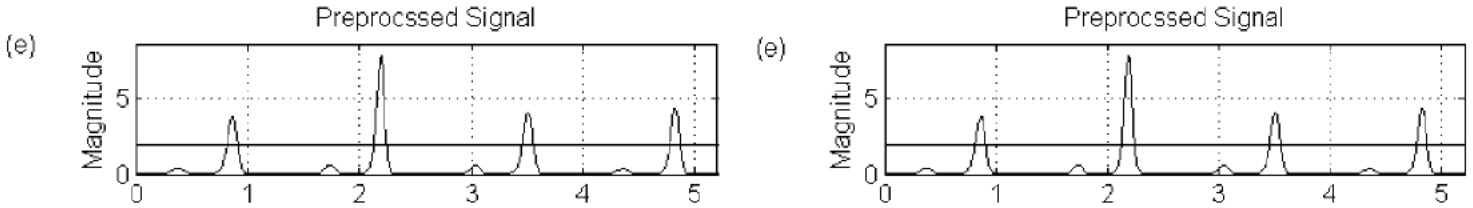


- Then the positive elements in both arrays are set to zero, whereas the negative elements remain. Both arrays are summed up entry-by-entry.

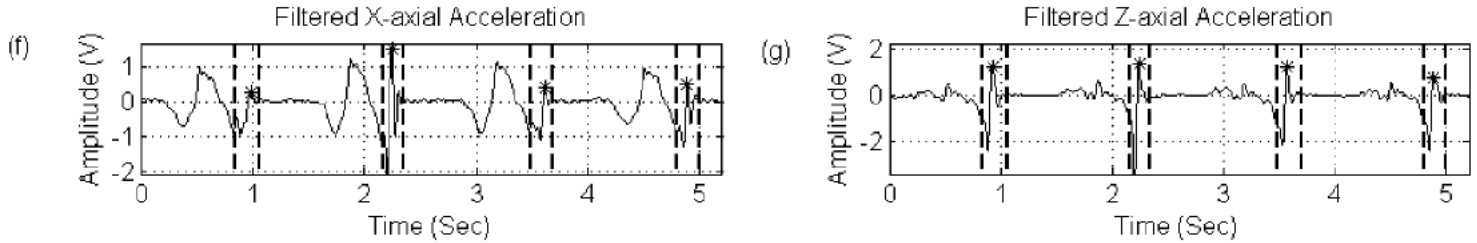
- Next, the intermediate results are smoothed using the moving-window integration filter.

$$y(n) = \frac{1}{N}[x(n-(N-1)) + \ldots \qquad (3)$$
$$x(n-(N-2)) + \cdots + x(n)]$$

- Then the squaring operator boosts the large value, which is associated with the deceleration before the feet striking on the ground, and suppresses the other waves of small amplitudes.



- The peaks in the preprocessed signal are detected using a threshold, one fourth of the maximum in array.
- The location of these peaks is defined as the onset of the peak-searching interval in the filtered signal and twice half-width of the sloping waves is defined as the width of the interval.



- **Pan-Tompkins method:**
- Easy to implement
- The fluctuation in the signal, yielding the positive and negative slopes as the useful feature, can result in false peak-searching interval.
- **Template-matching method:**
- Capable of detecting the steps self-adaptively and generating the representative template according to the current step signal.
- The first template is estimated since the parameters are unknown; hence it may not approximate the real step signal.
- **Peak detection based on combined dual-axial signals**
- The fastest and easiest among the three algorithms.