ECG-BASED BIOMETRICS

A Primer on Methods and Tools

Afonso Eduardo

aflm.eduardo@gmail.com

March 17, 2017

Preliminaries: Scientific Programming with Python

- **Recommended Python 2.7 distribution**
  - Anaconda: https://www.continuum.io/downloads

- **Libraries/Modules** (other than The Python Standard Library)

SciPy stack Core

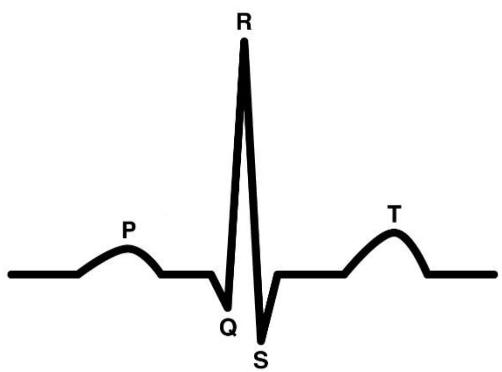
- **pandas**
  data munging (extract, transform, load).
- **numpy/scipy**
  vectorized numerical operations: manipulation, linear algebra, optimisation, signal processing.
- **scikit-learn**
  machine learning tools: preprocessing, dim. reduction, regression/classification. See also: ***statmodels*** *(statistical tests & time series analysis).*
- **matplotlib/seaborn**
  plotting tools: statistical graphics.

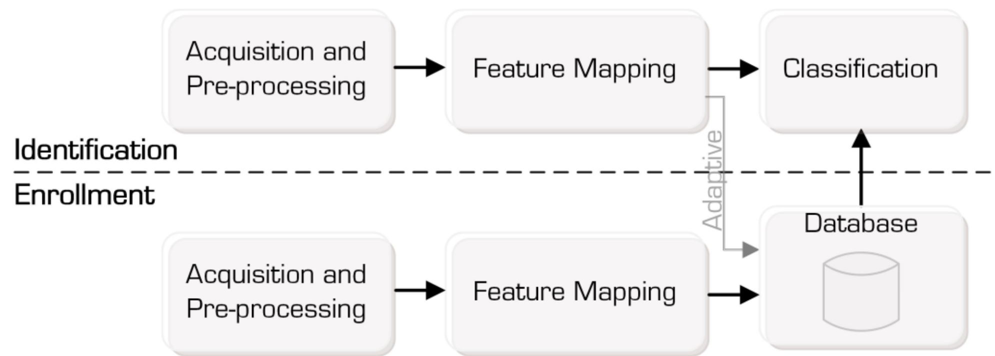
Others

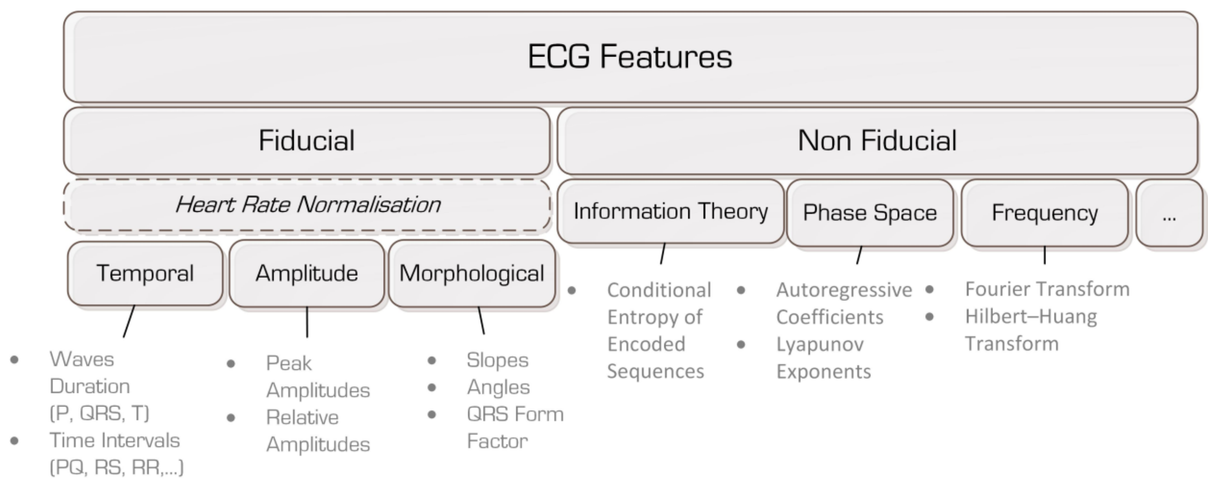
- **jupyter notebook**
  executable documents containing analysis description and results.
- **cython**
  combined power of Python and C/C++ (static compiler and wrapping tools). Similar interface tools exist for other languages (Java, MATLAB, R).
- **theano/tensorflow**
  graph-based numerical operations (allows e.g. automatic differentiation), transparent use of multiple CPUs and GPUs.
- …

Check the corresponding module webpages and technical books published for instance by O'Reilly.

ECG-based Biometrics: Overview







ECG biometrics surveys: (Odinaka, 2012) ECG Biometric Recognition: A Comparative Analysis.
(Fratini, 2015) Individual Identification via Electrocardiogram Analysis.
ECG analysis book: (Clifford, 2006) Advanced Methods and Tools for ECG Data Analysis.
Physiologic signals & open-source software website: http://physionet.org/
Machine Learning book: (Murphy, 2012) Machine Learning: A Probabilistic Perspective.

ECG-based Biometrics: Toolbox (I)
(in development)

- **Temporary location: PIANAS/Database/biometrics_code/**

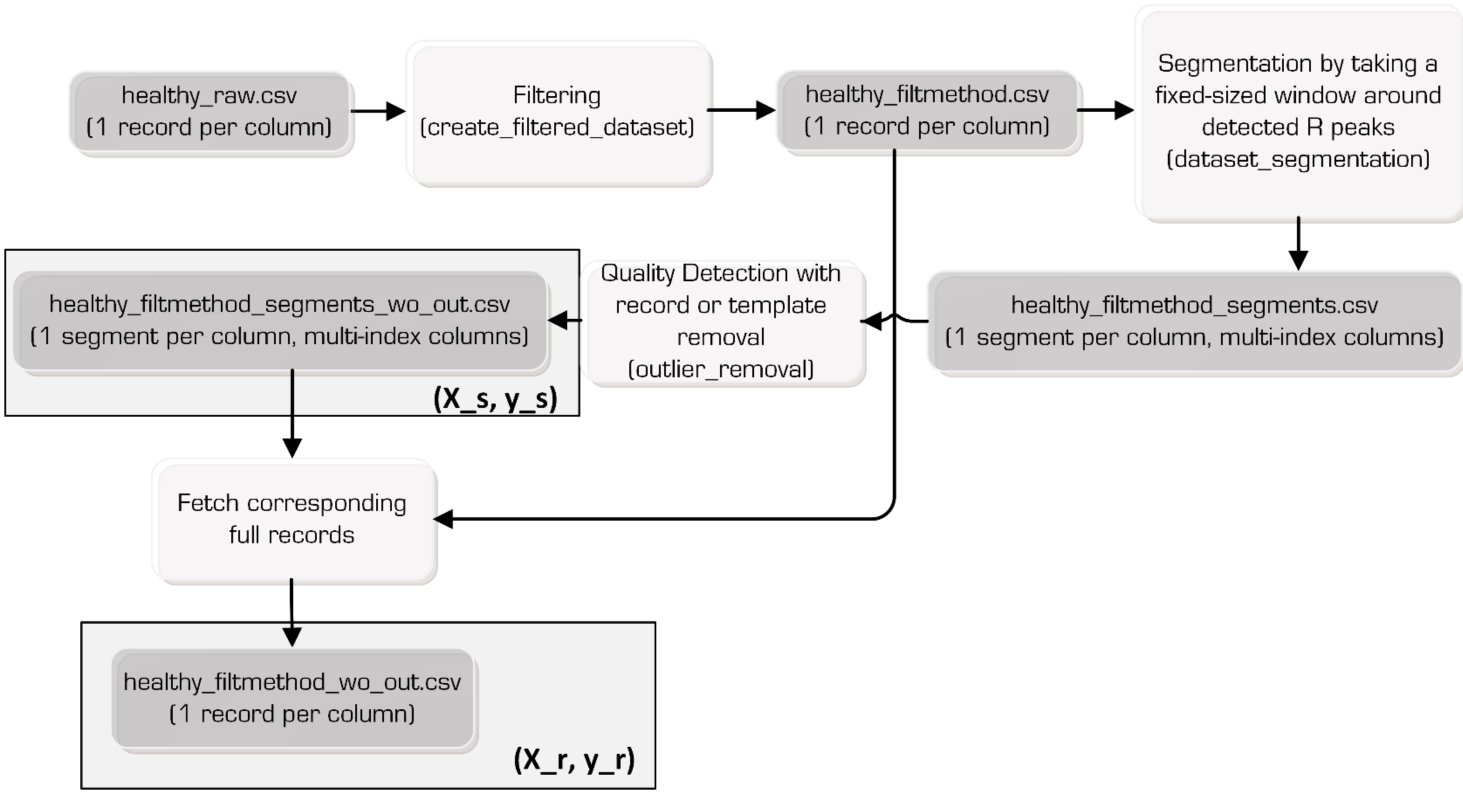
```
Overview:
---------
This project aims to provide functions that should ease the process of setting up pattern recognition systems, namely those whose purpose is biometric
identification based on ECG (Electrocardiogram).
    - The Preprocessing folder contains functions that allow to create, plot and filter an ECG dataset.
    - The Methods folder contains feature extraction and classification methods. In order to guarantee these functions share the same signature,
    wrappers.py has been created.
    - The module main.py implements the pipeline that consists of data selection, feature extraction, classification and logging. To create the
    filtered datasets, the preprocessing step has to be run beforehand.
    - For more information, check the corresponding modules.

Other Notes:
------------
To use the autoencoder, theano and keras (https://keras.io/) must be installed. Installing these on Linux shouldn't offer any complications. However,
if one wants to use Windows, this tutorial should be followed (it should also work for win7/8): https://github.com/philferriere/dlwin.

Python version:
    2.7
Python packages:
    scipy stack - https://www.scipy.org/install.html
    scikit-learn - http://scikit-learn.org/stable/documentation.html
    theano - http://deeplearning.net/software/theano/
    keras - https://keras.io/
    pywavelets - https://pywavelets.readthedocs.io/en/latest/
    seaborn - http://seaborn.pydata.org/
    joblib - https://pythonhosted.org/joblib/parallel.html
    matlab - https://www.mathworks.com/help/matlab/matlab_external/install-the-matlab-engine-for-python.html
             (Requires R2014b or later)
Version requirements:
    scipy: >=0.18
    scikit-learn: >=0.18
```

ECG-based Biometrics: Toolbox (II) *(in development)*

- **Preprocessing pipeline** *(heartbeat-based quality detection)*

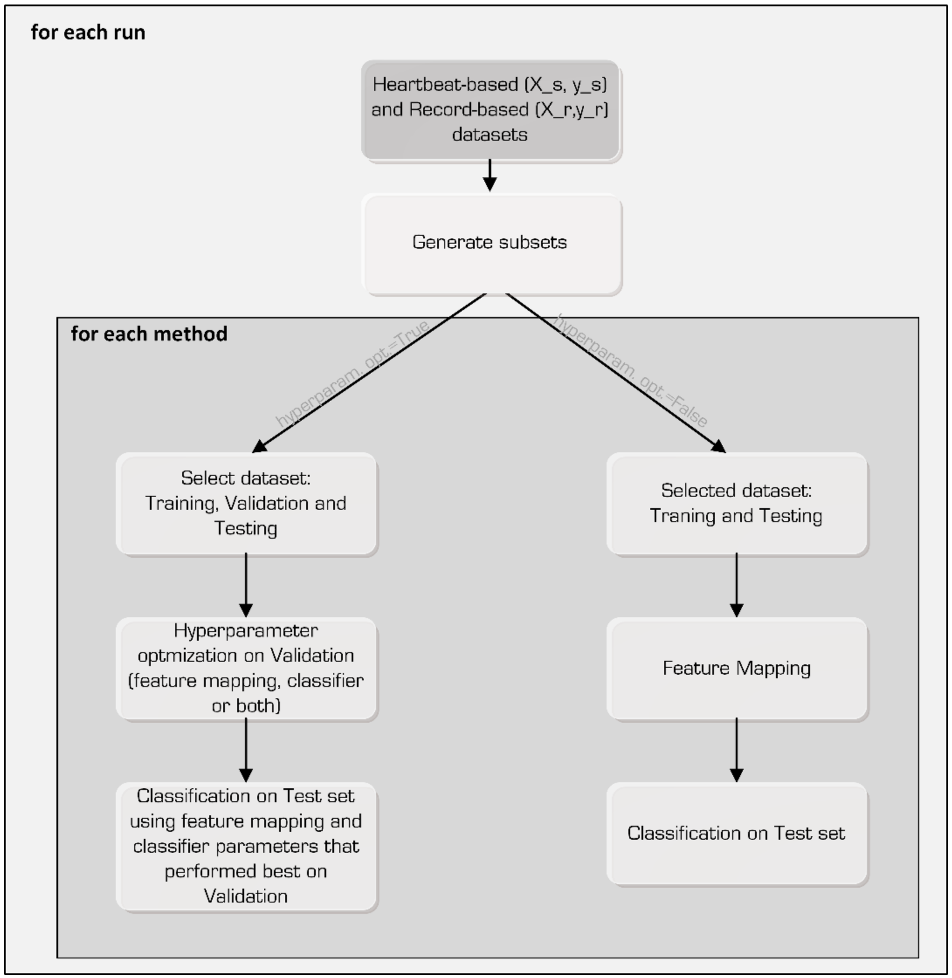


- **Preprocessing pipeline** *(record-based quality detection)*: does not require segmentation beforehand.[1]

[1] For instance, UNSW quality detection: (Khamis, 2016) QRS Detection Algorithm for Telehealth Electrocardiogram Recordings .

ECG-based Biometrics: Toolbox (III)
(in development)

- **Pattern recognition pipeline**



ECG-based Biometrics: Toolbox (IV)
(in development)

All feature extraction methods have the option of truncation and resampling of data arguments (this is accomplished with a function decorator).

Feature Extraction (FE):
Baseline (with option of dimensionality reduction: + DCT/LDA/PCA/KPCA)
Autocorrelation (+ DCT/LDA/PCA/KPCA) [1][2][3]
Short-Time Fourier Transform (stft) [4]
Wavelet decomposition (wt) [5]
Autoencoder (Shallow or Deep) [6]

Classifiers:
kNN (works with all FE methods)

Classifier/Stand-alone system (to use these select baseline with no dimensionality reduction as feature extraction and the following as classifiers):
Time-Frequency Robust Feature Selection (clf_tf_rbfs) [4]
Wavelets with measures 'prd', 'ccorr', 'wdist' (clf_wavelet) [5]
Phase representation with measures 'nSC', 'MNPD', 'MNPM' (clf_rec_phase) [7]

PCA: Principal Component Analysis
KPCA: Kernel PCA
LDA: Linear Discriminant Analysis
DCT: Discrete Cosine Transform

[1] (Plataniotis, 2006) ECG biometric recognition without fiducial detection.
[2] (Agrafioti, 2008) ECG Based Recognition Using Second Order Statistics.
[3] (Hejazi, 2016) ECG biometric authentication based on non-fiducial approach using kernel methods.
[4] (Odinaka, 2010) ECG biometrics A robust short-time frequency analysis.
[5] (Chan, 2008) Wavelet distance measure for person identification using electrocardiograms.
[6] (A. Eduardo, 2017) ECG-based Biometrics using a Deep Autoencoder for Feature Learning.
[7] (Fang, 2013) QRS detection-free electrocardiogram biometrics in the reconstructed phase space.

- *(show a notebook with some results)*

Phase-wrapped ECG

1. **R peak detection**
2. **Phase assignment**
   1. Generate linear ramp from 0 to 2π for each RR segment
   2. Subtract 2π where $\theta \geq \pi \rightarrow$ each heartbeat in $[-\pi, \pi[$
3. **Binarisation**
   1. Split each heartbeat according to a given number of bins
   2. Take the mean amplitude and phase for each bin (alternatively, median/other sample statistic)

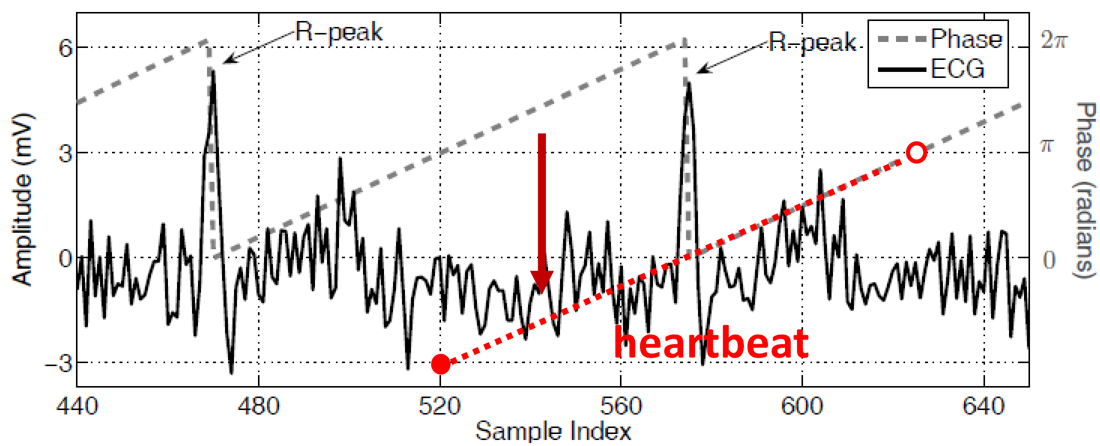


Fig. 1. An illustration of the phase assignment approach

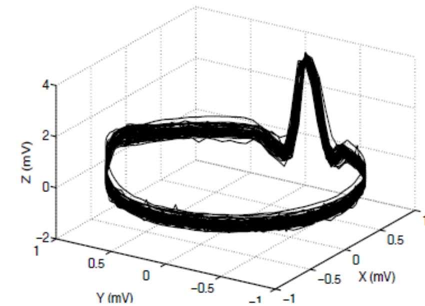


Fig. 2. Several cycles of the ECG phase-wrapped in the state space

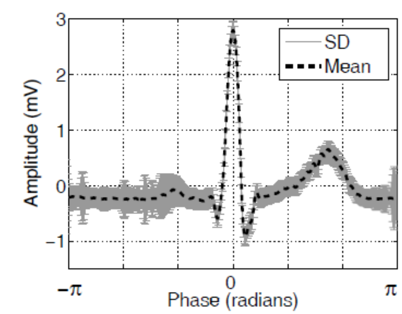


Fig. 3. An average and standard deviation-bar of 30 ECG cycles of a noisy ECG

- **A similar procedure can be applied to other quasi-periodic signals (e.g. BVP) using the most easily-detectable fiducial event!**

(Sameni, 2007) A Nonlinear Bayesian Filtering Framework for ECG Denoising.
The Open-Source Electrophysiological Toolbox (OSET) - MATLAB: http://www.oset.ir/

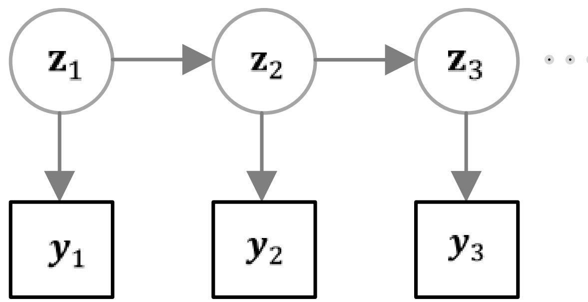
State Space Models: Overview (I)

- **Generic form:**

$$\begin{aligned} \mathbf{z}_t &= g(\mathbf{u}_t, \mathbf{z}_{t-1}, \boldsymbol{\epsilon}_t) \\ \mathbf{y}_t &= h(\mathbf{z}_t, \mathbf{u}_t, \boldsymbol{\delta}_t) \end{aligned}$$

$\mathbf{z}_t$: hidden state
$\mathbf{u}_t$: (optional) control signal
$\mathbf{y}_t$: observation
g: transition model
h: observation model
$\boldsymbol{\epsilon}_t$: system noise at time t
$\boldsymbol{\delta}_t$: observation noise at time t

- **Graphical representation:** *(identical to Hidden Markov Model, states are now continuous)*



- **Notable special case:** linear-gaussian SSM (LG-SSM) or linear dynamical system (LDS)

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$$

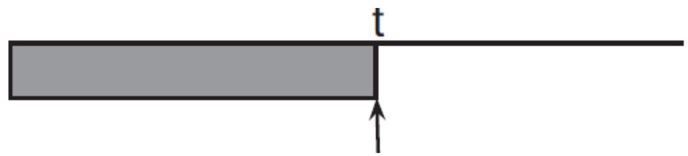
$$\mathbf{y}_t = \mathbf{C}_t \mathbf{z}_t + \mathbf{D}_t \mathbf{u}_t + \boldsymbol{\delta}_t \qquad \boldsymbol{\delta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$$

  - Supports exact inference: if inital state is Gaussian, all subsequent states are Gaussian.

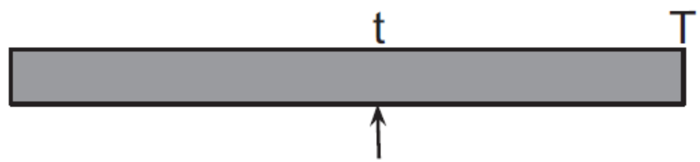
(Murphy, 2012) Machine Learning: A Probabilistic Perspective.

State Space Models: Overview (II)

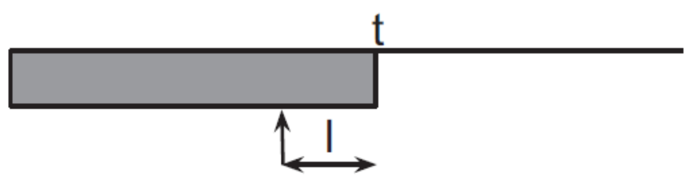
- **Types of inference problems for time series:**
  - **Filtering:** Compute belief state $p(\mathbf{z}_t|\mathbf{y}_{1:t})$ online as the data streams in.



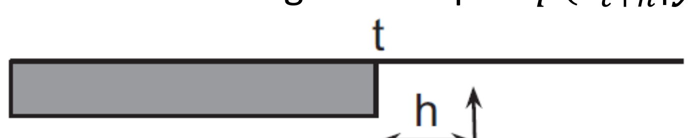
  - **Offline Smoothing:** Compute $p(\mathbf{z}_t|\mathbf{y}_{1:T})$ offline, given all evidence. Uncertainty is reduced by incorporating all future observations.



  - **Fixed lag Smoothing:** Compromise between online and offline estimation. Compute $p(\mathbf{z}_{t-l}|\mathbf{y}_{1:t})$, with $l > 0$ being the lag. Better performance than filtering, but at the cost of a slight delay.



  - **Prediction:** Predict the future given the past $p(\mathbf{z}_{t+h}|\boldsymbol{y}_{1:t})$, where h is the horizon.



State Space Models: Overview (III)

- **If the model is a LG-SSM, Kalman filter is the algorithm for exact filtering**

$$p(\mathbf{z}_t|\mathbf{y}_{1:t}, \mathbf{u}_{1:t}) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$

- **Prediction step:**

$$\begin{aligned}
p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) &= \int \mathcal{N}(\mathbf{z}_t|\mathbf{A}_t\mathbf{z}_{t-1} + \mathbf{B}_t\mathbf{u}_t, \mathbf{Q}_t)\mathcal{N}(\mathbf{z}_{t-1}|\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})d\mathbf{z}_{t-1} \\
&= \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}) \\
\boldsymbol{\mu}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\mu}_{t-1} + \mathbf{B}_t\mathbf{u}_t \\
\boldsymbol{\Sigma}_{t|t-1} &\triangleq \mathbf{A}_t\boldsymbol{\Sigma}_{t-1}\mathbf{A}_t^T + \mathbf{Q}_t
\end{aligned}$$

- **Measurement step** (after receiving observation y_t)**:**

$$p(\mathbf{z}_t|\mathbf{y}_t, \mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{u}_t)p(\mathbf{z}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t})$$

$$\begin{aligned}
p(\mathbf{z}_t|\mathbf{y}_{1:t}, \mathbf{u}_t) &= \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \\
\boldsymbol{\mu}_t &= \boldsymbol{\mu}_{t|t-1} + \boxed{\mathbf{K}_t\mathbf{r}_t} \\
\boldsymbol{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{C}_t)\boldsymbol{\Sigma}_{t|t-1}
\end{aligned}$$

Correction term: *the amount of weight placed on the error depends on the gain*
If $\mathbf{C}_t = \mathbf{I} \rightarrow \mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{S}_t^{-1}$: ratio of prior and measure error covariances.
E.g. strong prior or noisy sensors → $|\mathbf{K}_t|$ is small.

Residual *(diff. between observed and predicted)*

$$\begin{aligned}
\mathbf{r}_t &\triangleq \mathbf{y}_t - \hat{\mathbf{y}}_t \\
\hat{\mathbf{y}}_t &\triangleq \mathbb{E}\left[\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] = \mathbf{C}_t\boldsymbol{\mu}_{t|t-1} + \mathbf{D}_t\mathbf{u}_t
\end{aligned}$$

Kalman Gain

$$\begin{aligned}
\mathbf{K}_t &\triangleq \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^T\mathbf{S}_t^{-1} \\
\mathbf{S}_t &\triangleq \operatorname{cov}\left[\mathbf{r}_t|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] \\
&= \mathbb{E}\left[(\mathbf{C}_t\mathbf{z}_t + \boldsymbol{\delta}_t - \hat{\mathbf{y}}_t)(\mathbf{C}_t\mathbf{z}_t + \boldsymbol{\delta}_t - \hat{\mathbf{y}}_t)^T|\mathbf{y}_{1:t-1}, \mathbf{u}_{1:t}\right] \\
&= \mathbf{C}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}_t^T + \mathbf{R}_t
\end{aligned}$$

State Space Models: Overview (IV)

- **There is also a very efficient smoother for LG-SSM:** *Rauch-Tung-Striebel (RTS) smoother* aka *Kalman smoothing* algorithm. Kalman filter performs the forward pass and the smoother performs the backward pass (using information from the forward pass). This is related to message passing in graphical models.
- **What if the model is not linear in the parameters?** Approximate Inference. For instance, <u>Extended Kalman Filter (EKF)</u>: linearise g and h about the previous state estimate using a first order Taylor series expansion and then apply the standard Kalman filter equations. The same applies to the smoother.

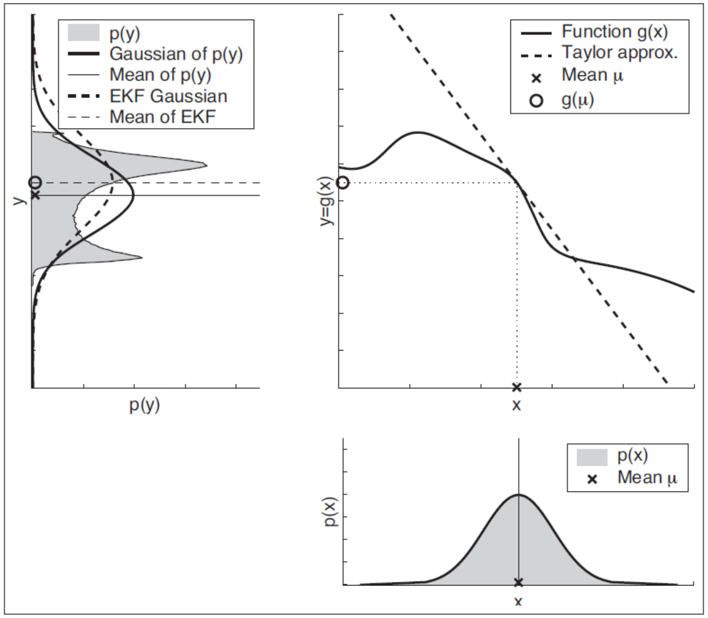
$$\begin{aligned} \mathbf{z}_t &= g(\mathbf{u}_t, \mathbf{z}_{t-1}) + \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \\ \mathbf{y}_t &= h(\mathbf{z}_t) + \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \end{aligned}$$

Approximate system model:

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{u}_t) &\approx \mathcal{N}(\mathbf{z}_t|\mathbf{g}(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) + \mathbf{G}_t(\mathbf{z}_{t-1} - \boldsymbol{\mu}_{t-1}), \mathbf{Q}_t) \\ G_{ij}(\mathbf{u}) &\triangleq \frac{\partial g_i(\mathbf{u}, \mathbf{z})}{\partial z_j} \\ \mathbf{G}_t &\triangleq \mathbf{G}(\mathbf{u}_t)|_{\mathbf{z}=\boldsymbol{\mu}_{t-1}} \end{aligned}$$

Approximate measurement model:

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{z}_t) &\approx \mathcal{N}(\mathbf{y}_t|\mathbf{h}(\boldsymbol{\mu}_{t|t-1}) + \mathbf{H}_t(\mathbf{z}_t - \boldsymbol{\mu}_{t|t-1}), \mathbf{R}_t) \\ H_{ij} &\triangleq \frac{\partial h_i(\mathbf{z})}{\partial z_j} \\ \mathbf{H}_t &\triangleq \mathbf{H}|_{\mathbf{z}=\boldsymbol{\mu}_{t|t-1}} \end{aligned}$$



<u>Unscented Kalman Filter (UKF)</u>: Instead of a linear approximation, pass a deterministically set of points (sigma points) through the function and fit a Gaussian to the resulting transformed points.

ECG Dynamical Model: Overview

- **(McSharry, 2003) A Dynamical Model for Generating Synthetic Electrocardiogram Signals**
  - Set of differential equations that generates a trajectory in a space with coordinates (*x*, *y*, *z*):

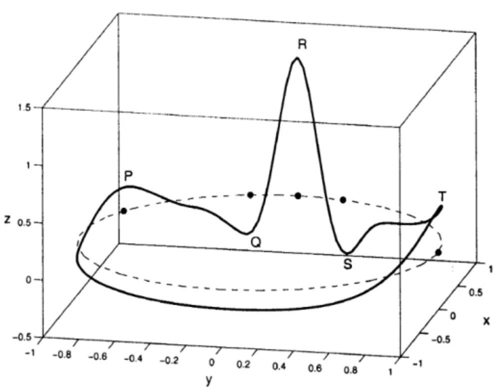
Sum of Gaussian kernels

$$\begin{cases} \dot{x} = \gamma x - \omega y \\ \dot{y} = \gamma y + \omega x \\ \dot{z} = \boxed{-\sum_{i \in \{P,Q,R,S,T\}} a_i \Delta\theta_i \exp\left(-\frac{\Delta\theta_i^{\,2}}{2b_i^{\,2}}\right)} - \boxed{(z - z_0)} \end{cases}$$

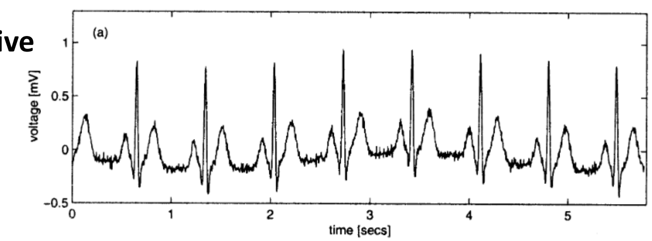
Baseline wander term (z_0 is a low sinusoidal component)

$$\gamma = 1 - \sqrt{x^2 + y^2},\ \Delta\theta_i = (\theta - \theta_i) \bmod 2\pi,\ \theta = atan2\,(y, x)$$

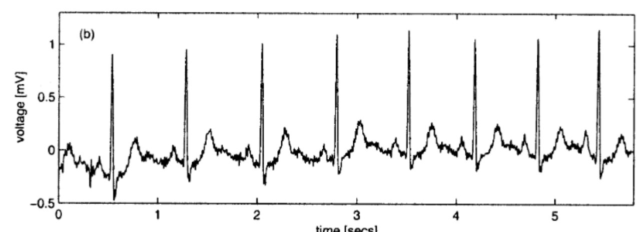
$\omega = 2\pi f$ Angular velocity of the trajectory (related to the beat-to-beat frequency)



Synthetic with additive Gaussian noise



Real



Some applications

Denoising normal ECG: (Sameni, 2007) A Nonlinear Bayesian Filtering Framework for ECG Denoising.

Attacking Biometric Systems: (Eberz, 2017) Broken Hearted: How to Attack ECG Biometrics.

ECG Dynamical Model: Denoising (I)

- **(Sameni, 2007) A Nonlinear Bayesian Filtering Framework for ECG Denoising**
  - Modification of the state equations to polar coordinates:

$$\begin{cases} \dot{r} = r(1-r) & \text{Radial variable } \textit{(can be dropped because there is no coupling)} \\ \dot{\theta} = \omega & \text{Angular variable} \\ \dot{z} = -\sum\limits_{i\in\{P,Q,R,S,T\}} \frac{\alpha_i \omega}{b_i^2}\Delta\theta_i exp(-\frac{\Delta\theta_i^2}{2b_i^2}) - \boxed{(z-z_0)} & \text{Amplitude} \end{cases}$$

Baseline wander *(can be dropped)*

  - Discrete form:

Sampling period

$$\begin{cases} \theta_{k+1} = (\theta_k + \omega\boxed{\delta})mod(2\pi) \\ z_{k+1} = -\sum\limits_i \delta\frac{\alpha_i \omega}{b_i^2}\Delta\theta_i exp(-\frac{\Delta\theta_i^2}{2b_i^2}) + z_k + \boxed{\eta} \end{cases}$$

Random additive noise

$$\Delta\theta_i = (\theta_k - \theta_i)mod(2\pi)$$

  - How to denoise *(simplified):*
    1. Compute the phase-wrapped ECG mean and standard deviation.
    2. Nonlinear least squares optimisation to determine the parameters (amplitude, angular width and position of PQRST waves): α_i, b_i, θ_i.
    3. Linearisation of the model, Kalman filter followed by the Kalman smoother equations.
  - In this model, the parameters α_i, b_i, θ_i are fixed, but it is possible to augment the state equations to introduce variability:

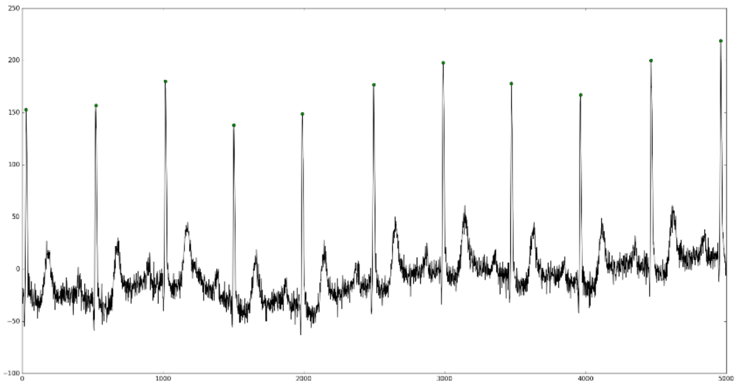
(Su, 2013) [Master Thesis] ECG Noise Filtering Using Online Model-Based Bayesian Filtering Techniques.

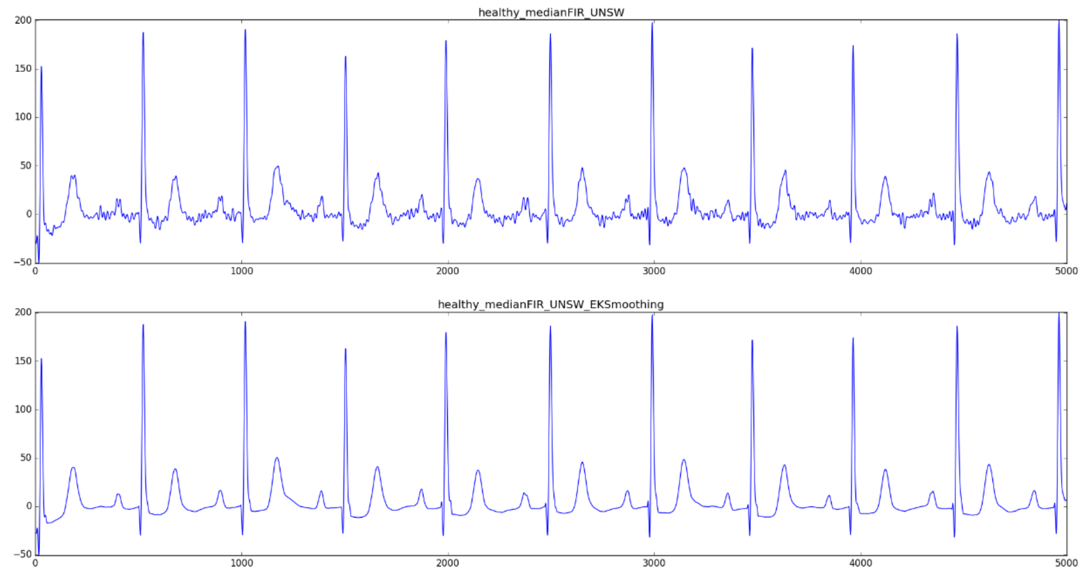
ECG Dynamical Model: Denoising (II)

- **(Sameni, 2007) A Nonlinear Bayesian Filtering Framework for ECG Denoising**

  - Other quasi-periodic signals can be modelled using a similar framework (e.g. BVP, PCG): (Almasi, 2011) A dynamical model for generating synthetic Phonocardiogram signals.

  - Denoising on data from Hospital Santa Marta - DBCarlos, subject 2016 (lead I):



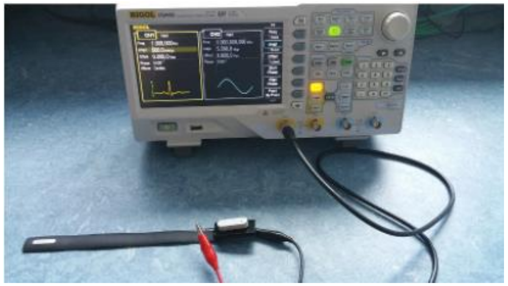


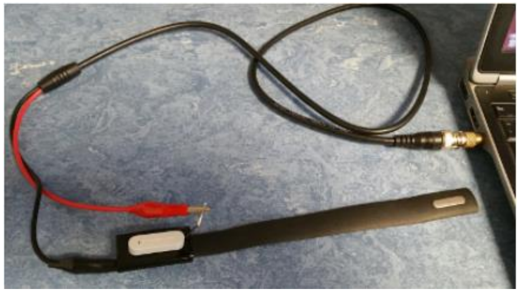
- **Popular alternatives are based on Empirical Mode Decomposition (intrinsic mode function rejection) and Wavelet Transform (coefficient thresholding/shrinkage):** e.g.
  (Kabir, 2012) Denoising of ECG signals based on noise reduction algorithms in EMD and wavelet domains.
- **Other observations:** Some papers on denoising only consider white noise. Real noise spectrum can be colored and noise samples are often correlated in time – see (Sameni, 2007).

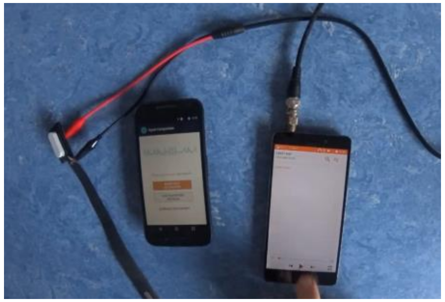
ECG Dynamical Model: Spoofing Attack (I)

- **How to impersonate a legitimate user in ECG Biometrics?** We could generate synthetic ECGs using a dynamical model where parameters are extracted from previous measurements.

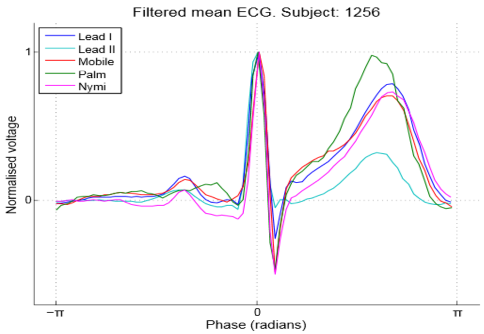
- **(Eberz, 2017) Heart Broken: How to Attack ECG Biometrics**
  - <u>Signal injection methods:</u> HW waveform generator, laptop soundcard with SW-based waveform generator, playback of .wav-encoded ECG signal.







  - <u>Other considerations:</u>
    Attacker takes photo from the victim's ECG → image analysis
    Source device ≠ target device (nymi band) → ≠ waveform morphology among devices



ECG Dynamical Model: Spoofing Attack (II)

- **(Eberz, 2017) Heart Broken: How to Attack ECG Biometrics**
  - Training cross-device mapping and generation of attack signals: