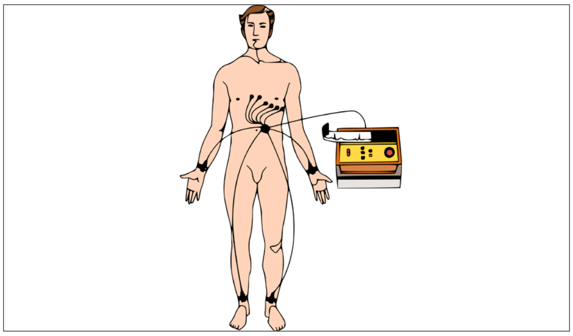
Philipp Oehler

Tina Zweifel

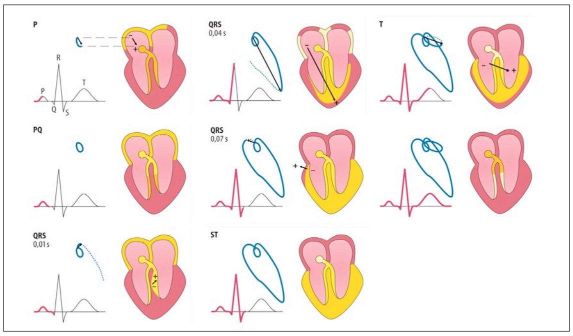
Graduate Candidates	Philipp Oehler, Tina Zweifel
Examiner	Prof. Dr. Guido Schuster
Co-Examiner	Ramun Schmid, Schiller AG, Baar ZG
Subject Area	Digitale Signalverarbeitung
Project Partner	Schiller AG, Baar ZG

Pseudo-vectorcardiogram

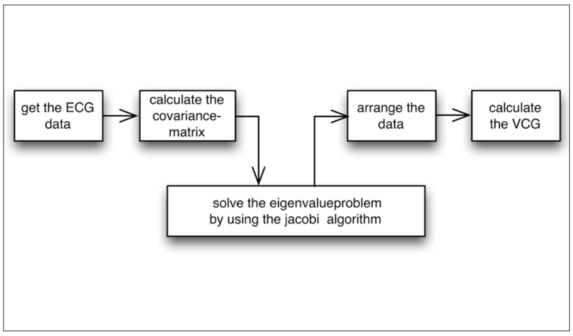
From derivation to adaptive use



Electrocardiogram with 12 standard leads I, II, III, aVR, aVL, aVF and V1 to V6



Stimulation of the heart muscle



Procedure of the Principal Components Analysis

Introduction: It is a fact that the heart is one of the most important organs in the human body, which makes it vital that medical practitioners diagnose abnormal functionalities. To record the electrical stimulation of the heart muscle, a helpful piece of equipment is the Electrocardiogram (ECG). The electrodes must be fixed on different positions on the body, so the difference of potential can be measured (so-called leads). There are several possibilities to get the vectorcardiogram (VCG) out of the ECG. Some of them are purely mathematical approaches; one other method is to measure it directly. In the following, we go into the mathematical way in detail.

Approach / Technologies: One method to convert the ECG into a VCG is Principal Components Analysis (PCA). PCA makes it possible to identify patterns in data. It searches the components with the highest variance in a certain direction, so that the new dataset is uncorrelated and orthogonal. Taking only the three most important components results in the VCG. The next step is to make it possible to use PCA in an adaptive way, so that the data can be handled consecutively. The Jacobi algorithm offers a brilliant opportunity for that. The eigenvectors of the covariance matrix are used to find the principal components. Instead of calculating the eigenvectors for every new symmetrical covariance matrix it is assumed that the diagonal matrix (containing the eigenvalues of the covariance matrix) do not change much. The Givens rotations do the rest to diagonalise it again and create new adapted eigenvectors for the next data block.

Solution: The results clearly show the performance of the Jacobi Algorithm. It is a perfectly tailored method for our particular need. Having integrated the code, the eigenvectors converge very fast. Moreover, it can be easily implemented and with very few lines of code in an embedded system. In case of any signal perturbations, the Jacobi Algorithm reacts in a stable way and is still able to calculate eigenvectors.