An Open-Source Method for Simulating Atrial Fibrillation Using ECGSYN

J Healey[1], GD Clifford[2], L Kontothanassis[1], PE McSharry[3,4,5]

[1]Hewlett-Packard Laboratories, Cambridge MA, USA
[2]Harvard-MIT Division of Health Sciences & Technology, Cambridge MA, USA
[3]Department of Engineering Science, University of Oxford, Oxford, UK
[4]Mathematical Institute, University of Oxford, Oxford, UK
[5]Centre for the Analysis of Time Series, London School of Economics, London, UK

Abstract

A method for simulating atrial fibrillation has been developed using the open source simulator ECGSYN (available at physionet.org). In this method episodes of atrial fibrillation are simulated by combining data streams generated by two ECGSYN engines driven from a correlated stochastic process. The underlying process represents atrial activity during fibrillation. This process generates a series of atrial inter-beat intervals with the same statistical properties as those found in the MIT-BIH atrial fibrillation database (afdb). The atrial beats from this process are then selected to be either conducting beats that generate subsequent ventricular activity or non-conducting beats that generate only atrial activity. The intervals for conducting beats are processed by an ECGSYN engine with appropriate parameters to create PQRS and T waves while the non-conducting intervals are processed by a second ECGSYN engine with parameters to generate only P waves. The data streams from the two engines are superimposed to create an artificial atrial fibrillation waveform. This waveform generator has been made into an operator and has been incorporated into a stream based ECG simulator. The simulator uses a timing operator to switch from generating normal ECG morphologies to atrial fibrillation.

1. Introduction

In order to effectively test the performance of signal processing algorithms for analyzing biomedical signals, a noise-free signal is often desired [1]. A realistic artificial biomedical signal generator that is able to encompass the range of signals observed for both normal and abnormal subjects is therefore a useful tool. Furthermore, the ability to rapidly create a regenerateable time series enables a researcher to quickly prototype applications and test theories such as signal mixing[1, 2]. Algorithms can be tested under unusual circumstances and as a function of the model parameters such as sampling rate [3]

In this paper we present a method for generating a simulated atrial fibrillation signal using the superimposed output of two ECGSYN engines. This method is based on an atrial fibrillation model proposed by Zeng and Glass [4] in which a number of atrial waves conduct and cause ventricular activation. To create a synthetic AF signal we first generated a series of atrial beat intervals using the ECGSYN program *rrprocess* with statistical parameters from the MIT-BIH atrial fibrillation database[5][4]. We then determined which of the intervals generated by this process would be conducting vs. non-conducting. Two ECGSYN engines with customized parameters separately processed the conducting and non-conducting intervals to generate two ECG waveforms. These waveforms were then superimposed to create the synthetic AF signal.

2. The ECG model

The ECG model is explained in detail in [6] with modifications in [1]. Briefly, the ECG is generated by a a fourth order Runge-Kutta integration of three ordinary differential equations in a 3-space (x, y, z)

The ode's that for the dynamical equations of motion are given by

$$\begin{aligned} \dot{x} &= \alpha x - \omega y, \\ \dot{y} &= \alpha y + \omega x, \\ \dot{z} &= -\sum_{i\in\{P,Q,R,S,T\}} a_i \Delta\theta_i \exp(-\Delta\theta_i^2/2b_i^2) - (z - z_0) \quad (1) \end{aligned}$$

where $\alpha = 1 - \sqrt{x^2 + y^2}$, $\Delta\theta_i = (\theta - \theta_i) \bmod 2\pi$, $\theta = \operatorname{atan2}(y, x)$ and ω is the angular velocity of the trajectory as it moves around the limit cycle. The ECG is reproduced using the motion of the trajectory in the z-direction.

Distinct points on the ECG, such as the P,Q,R,S and T are described by *events* corresponding to negative and pos-

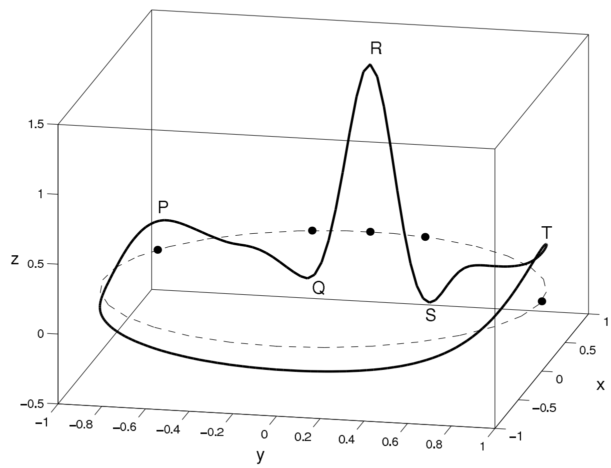

Figure 1. A typical trajectory generated by the dynamical model (1) in the three-dimensional space given by (x, y, z). The dashed line reflects the limit cycle of unit radius while the small circles show the positions of the P,Q,R,S,T events.

Table 1. ECG & model parameters for normal beats in (1). $\beta = \frac{\sqrt{\mathrm{HR}}}{60}$

Index (i)	Time (s)	θ_i (rads)	a_i	b_i
P	-0.2	$-\frac{1}{3}\pi\sqrt{\beta}$	1.2	0.25β
Q	-0.05	$-\frac{1}{12}\pi\beta$	-5.0	0.1β
R	0	0	30.0	0.1β
S	0.05	$\frac{1}{12}\pi\beta$	-7.5	0.1β
T	0.3	$\frac{1}{2}\pi\sqrt{\beta}$	0.75	0.4β

itive attractors/repellors in the z-direction. These events are placed at fixed angles along the unit circle given by θ_P, θ_Q,θ_R,θ_S and θ_T (see Fig. 1). When the trajectory approaches one of these events, it is pushed upwards or downwards away from the limit cycle, and then as it moves away it is pulled back towards the limit cycle.

The beat-to-beat variation in the morphology is induced by the variation in the integration step dt to reflect changes in the RR interval; the time to complete one revolution around the attracting limit cycle in the (x, y)-plane. Shorter RR intervals (higher HR's) compress the waveform, resulting in shorter QT intervals and lower RS amplitudes (i.e. RSA). To mimic Bazett's law, a further compression factor is added; the θ_i^{BP} are therefore pre-multiplied by a factor proportional to $\beta = \sqrt{\mathrm{HR}}/60$, causing a decreasing delay between the θ_i^{ECG} and the θ_i^{BP} for increasing mean HRs.

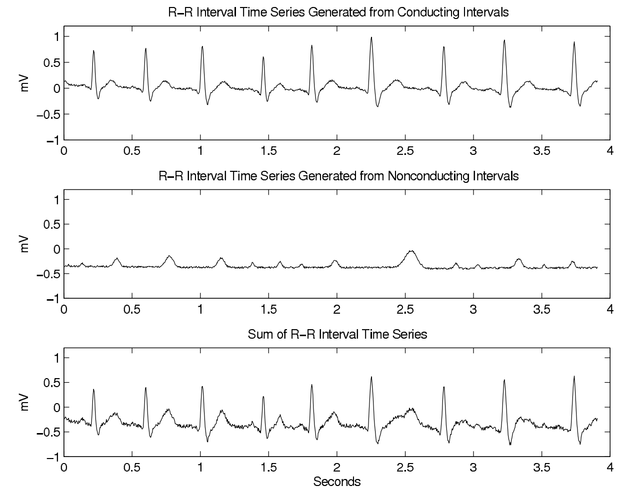

Figure 2. The model for synthetic atrial fibrillation (bottom) is the sum of the conducting atrial beats (top) and the non-conducting beats (middle)

2.1. AF model

The AF model begins with the generation of a series of atrial beat intervals. This was accomplished using the Matlab version of the ECGSYN *rrprocess* function as follows:

```
[rrp]=
rrprocess(256,256,0,410,94,0.5,256)
```

The variable rrp now contains an atrial beat time series with the statistical properties of atrial fibrillation as found in the MIT-BIH atrial fibrillation database (afdb). From this process atrial intervals are next assigned to be either conducting or non-conducting according to the following rules:

- The first atrial beat is conducting.
- An atrial beat occurring during ventricular activity is non-conducting.
- Otherwise, a beat has a 76% chance of conducting.

The value for the probability of conduction after the cessation of ventricular activity, 76% was determined experimentally as the probability that resulted in a similar ratio (three to one) of atrial to ventricular beats as derived from the MIT-BIH database[4].

To generate synthetic AF signal, the conducting and non-conducting intervals were then separated and used to seed two different ECGSYN engines. The conducting interval time series was processed by an ECGSYN engine with parameters a_i and b_i (see Table 2) to generate PQRS and T waves for every interval. The non-conducting interval time series was processed by an ECGSYN engine with parameters a_i and b_i (see Table 2) set to generate only P waves for every interval as described in Table 2. These two

Table 2. Modified ECG & model parameters in for generating synthetic AF where a_1 and b_1 are the parameters used for the conducting intervals and a_2 and b_2 are the parameters used for the non-conducting intervals.

Index (i)	P	Q	R	S	T
a_1	0.5	-5.0	30.0	-7.5	0.5
b_1	0.1β	0.1β	0.1β	0.1β	0.4β
a_2	0.5	0	0	0	0
b_2	0.2β	0.9β	0.9β	0.9β	0.9β

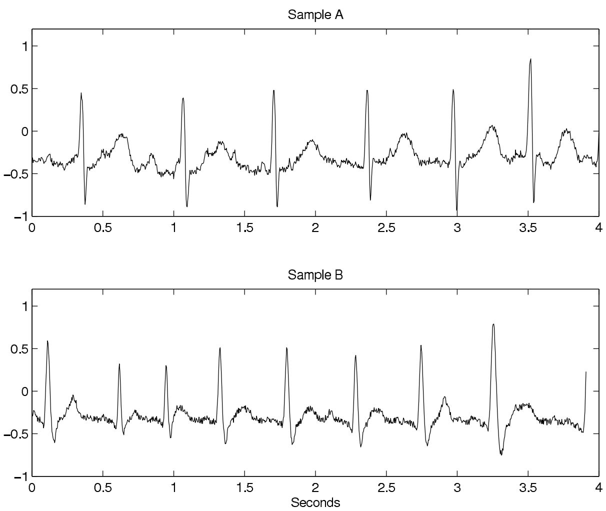

Figure 3. A comparison of natural atrial fibrillation (Sample A) with a segment of synthetic atrial fibrillation (Sample B)

synthetic ECG signals were then superimposed to create a synthetic AF signal as shown in Figure 2.

3. Discussion and Future Work

The results of the synthetic AF generation are visually and statistically similar to real signals taken from the MIT-BIH atrial fibrillation database. Figure 3 shows a comparison of a real atrial fibrillation signal (samples 5706021 through 570702 of record 04746 of the MIT-BIH atrial fibrillation database[5]) with a signal generated using this synthetic AF model. This model was created to test a real-time stream processing system for processing physiological signals[7]. To improve the model in the future we are considering a more sophisticated model for determining which of the atrial beats are conducting beats, following more closely the ideas presented by Zeng and Glass[4].

4. Author's Note

Open source C, Matlab and Java code for the models is available from Physionet [8].

References

[1] Clifford G, McSharry PE. A realistic coupled nonlinear artificial ecg, bp, and respiratory signal generator for assessing noise performance of biomedical signal processing algorithms. Proc of SPIE International Symposium on Fluctuations and Noise 2004;5467(34):290–301.

[2] McSharry P. E.and Clifford GD. A comparison of nonlinear noise reduction and independent component analysis using a realistic dynamical model of the electrocardiogram. Proc of SPIE International Symposium on Fluctuations and Noise 2004;5467(09):78–88.

[3] McSharry PE, Clifford GD. Open-source software for generating electrocardiogram signals. ARXIV preprints June 2004;0406017.

[4] Zeng W, Glass. Statistical properties of heartbeat intervals during atrial fibrillation. Phys Rev E 1996;54:1779–1784.

[5] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–e220.

[6] McSharry PE, Clifford GD, Tarassenko L. A dynamical model for generating synthetic electrocardiogram signals. IEEE Trans Biomed Eng 2003;50(3):289–294.

[7] Bar-Or A, Healey J, Kontothanassis L, Van Thong J. Biostream: A system architecture for real-time processing of physiological signals. Proc IEEE EMBS September 1-5 2004;San Francisco, CA USA.

[8] http://www.physionet.org/physiotools/ecgsyn/.

Address for correspondence:

J. A. Healey
Hewlett-Packard Cambridge Research Laboratory
One Cambridge Center
Cambridge MA, 02142 USA
jennifer.healey@hp.com