Simulating Healthy Human Heart Rate: A Markovian Model

CC Yang, CH Chang, SS Hseu, HW Yien

Taipei Veterans General Hospital, Taipei, Taiwan
School of Medicine, National Yang-Ming University, Taipei, Taiwan

Abstract

This study is performed within the scope of Computers in Cardiology Challenge 2002 on simulating 24 hours RR interval time series. We construct a computational model to characterize short- and long-term complex dynamics of healthy human heart rate. The cardiac dynamics is simplified via map the increase and decrease of the interbeat interval to 1 and 0, respectively. The probability of transition from current binary sequence to next state is then determined by following two factors: 1. Prior history of binary sequences, and 2. Current value of RR interval. Probability tables were constructed from real data . We used 8-bits and 2-bits binary sequences to simulate short- and long-term heart rate fluctuations. The magnitude of increment was chosen randomly. Finally, we implemented two simple functions to simulate the circadian rhythm and temporal structures of fluctuations during rapid eye movement stage sleep. The model reached a score of 0.689 in event 1 (entry142). In summary, our preliminary study indicated Markovian model may apply to different levels of physiologic regulations. Further study is needed to examine the correlation with physiologic mechanisms.

1. Introduction

Human cardiac dynamics are driven by the complex nonlinear interactions of two competing forces: sympathetic regulation increases and parasympathetic regulation decreases the heart rate. These cardiac regulations form complex fluctuations that were known as heart rate variability. Although various techniques were developed to analyze human heart rate and many physical properties have been quantified [1], it is still difficult to create a simulation of heart rate with sufficient realism to mislead an experienced observer, or has similar quantities with real heart rate time series. Here we present a Markovian model to simulate the healthy human heart rate, and use 24 hours ECG reference databases from PhysioNet (16 subjects) and Taipei Veterans General Hospital (7 subjects) for model construction.

2. Method

The heart rate simulation algorithm consists of 4 components, includes Markovian model for short- and long-term heart rate regulating mechanisms. The short term fluctuation represents beat-to-beat variation, which is mostly regulated by parasympathetic system. The long term fluctuation represents minute-to-minute fluctuations, which is mainly regulated by sympathetic system. In addition, we implemented two simple functions into program to simulate circadian rhythm and temporal structures of heart rate during rapid eye movement stage (REM) sleep.

2.1. Markovian model for short- and long-term fluctuations

To build a Markovian model, we have to specify a set of states and associated probabilities that RR interval will move from one state to another. Therefore, the first step of heart rate simulation is to map RR increments into specific states. Since heart rate is mainly regulated by autonomic nervous system which consists of two competing forces: sympathetic regulation increase and parasympathetic regulation decrease the heart rate. It is then plausible to use these two states for model construction [2,3]. Consider a time series: $\{X_0, X_1, X_2, X_3, X_4, \cdots\cdots, X_n\}$. For each pair of successive sequences, we can classify it into one of the 2 states that represent increase in X, and decrease in X. These 2 states are mapped to the symbols 0 and 1, respectively.

$$w_n(X_n, X_{n+1}) = \begin{cases} 0 : X_n - X_{n+1} <= 0 \\ 1 : X_n - X_{n+1} \quad > 0 \end{cases} \tag{1}$$

The next state of RR intervals (increase or decrease) is then determined by following 2 factors (Fig 1): 1. Prior history of binary sequences, and 2. Current value of RR interval. The probability of transition from current binary sequences to next possible states can be calculated statistically from real data. In addition, the magnitude of increment is chosen randomly by fitting a Gaussian distribution curve to histograms of RR increments.

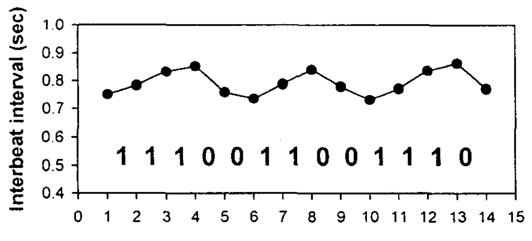

Figure 1. Illustration of mapping procedure from part of real RR interval time series. Each increment is mapped to 0 or 1 according to decrease or increase of RR intervals, respectively.

We apply this model to different time scales of real RR time series: beat-to-beat and minute-to-minute fluctuations. The real RR time series were taken from PhysioNet [4] and Taipei Veterans General Hospital [5]. Fig 2 shows original time series of 2 minutes (upper panel) and coarse-grained time series of 2 hours (lower panel) which values are calculated by averaging of every 60 beats. Both time scales show complex fluctuations that are regulated by different physiologic systems.

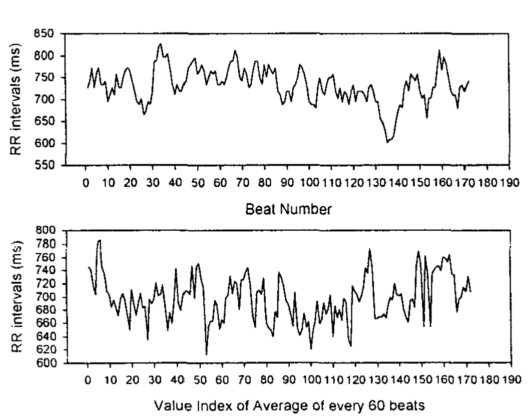

Figure 2. Illustration of beat-to-beat heart rate fluctuations (upper panel) and coarse-grained long-term fluctuations (lower panel)

To apply the model to real heart rate time series, we use 8- and 2-order of Markovian model to simulate short- and long-term fluctuations, respectively. In terms of m-order Markovian model, the next state of RR increment is determined by prior m-bit binary sequences. For example of modelling short-term (beat-to-beat) fluctuations, real RR time series is first transformed into binary sequences via mapping decrease or increase of RR increments into 0 or 1, respectively. Then the binary sequences are further partitioned into a set of 8-bits binaries. The probability of transition from each binary to its next possible states can be determined by comparing occurrence of current sequence with that of next possible sequence.

However, it is apparent that transition probability is not constant for all range of RR intervals. For example, heart rate tends to decrease while the heart beats fast and in vice versa. To apply this concept to our model, we partitioned RR intervals into bins of 100 ms and calculated its associated transition probability of each binary within each bins.

Figure 3 illustrated a probability table of transition from 8-bits binary. The table defines the transition probability of a specific binary (vertical axis) within a certain range of RR intervals (horizontal axis). Both short- and long-term probability tables were constructed from real data. To initiate the heart rate simulation, we randomly assign a binary sequence and a normal value of RR interval. In each step, the next state (increase or decrease) is determined by both tables and magnitude of increment is chosen randomly to generate a new RR interval.

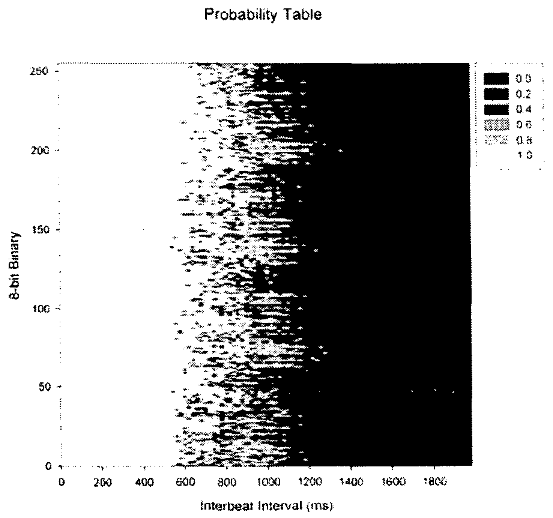

Figure 3. Example of short-term (beat-to-beat) probability table which defines transitions of a binary to its next possible states according to current value of RR intervals.

2.2. Circadian rhythm

Markovian model can provide statistical simulation of short- and long-term fluctuations based on real time series. However, for longer scale of fluctuations such as circadian rhythm, there is no sufficient data length to reach significant statistics. Therefore, based on physiologic principles, the program randomly determine the time of sleep as well as time to wake up, and use a simple step function to simulate gradually increased RR interval (decreased heart rate) while falling asleep and vice versa while waking up.

2.3. Fluctuations during REM sleep

In young adults, REM sleep occupies about 25% of sleep time. The first episode of REM sleep occurs about 80-90 minutes after falling asleep and come in 4-5 episodes during the night. In REM sleep, breath and pulse tend to be rapid and often irregular. [6] Fig 4 shows an illustration of heart rate during REM sleep.

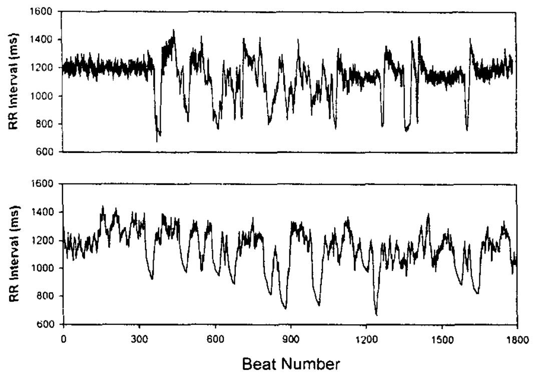

Figure 4. Illustration of real fluctuations during REM sleep (upper panel) and synthetic fluctuations (lower panel).

The original ECG signals were examined by physicians and subjects were reviewed for excluding any pathologic causes such as sleep apnea. We found these fluctuations existed in most recordings in both databases. Based on this observation, the program first randomly assigns the number of cycles of REM sleep, and sets a higher REM occurrence during those periods. We further implemented an iterated function to gradually decrease and increase RR interval while REM occurs. The resulting simulation was shown as lower panel of Fig 4.

3. Result

The representative recording from Taipei Veterans General Hospital databases was shown as upper panel in Fig 3. with a randomly generated RR timer series shown in lower panel. The model of entry 142 finally reached a score of 0.693 in event 1. The source code of generator function was shown in appendix. The complete source code with probability table can be downloading via website [5]. Comparing to real time series, our synthetic series shows distributions of 1/f noise on power spectral density analysis and comparable results in low frequency and high frequency domains. In overall, our synthetic series shows realistic behaviour on short-term scales and features of REM fluctuations indeed mislead some physicians.

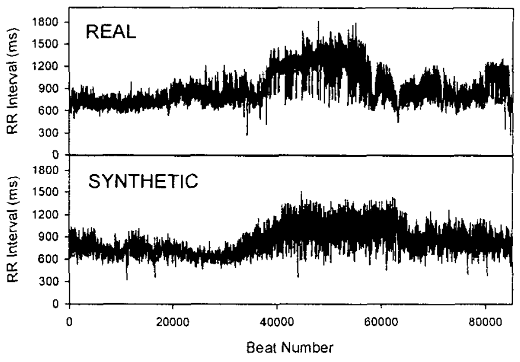

Figure 5. Result of synthetic RR time series (lower panel) compared to real RR time series (upper panel).

4. Conclusion

There are some limitations of our model. First, we assume the magnitude of increment is random. However, previous study reveals it has long-range correlations properties [3]. Second, one probability table is only capable to generate one type of signals. Therefore, our algorithm can not generate a diversity of RR interval time series. In comparing with real RR time series by currently used heart rate variability parameters, our model can captures short- and mid-term (up to 30 minutes) physiologic fluctuations. This may suggest that Markovian model can apply to different level of physiologic system once the signal has sufficient sample number for significant model construction.

Acknowledgements

The author wishes to thank PhysioNet and Taipei Veterans General Hospital for providing databases that were used for model construction, and Ary L. Goldberger, C.-K. Peng, J. E. Mietus, J. Healey and I. C. Henry for discussions of these and related issues. We also thank support from the National Science Council of Taiwan (NSC89-2314-B-075-160)

References

[1] *Heart Rate Variability*, edited by M. Malik and A. J. Camm (Futura, Armonk, NY, 1995) .
[2] H. Kantz and T. Schreiber-Nonlinear Time Series Analysis (Cambridge University Press, Cambridge, 1999);
[3] Y. Ashkenazy et al., Phys. Rev. Lett. 86, 1900 (2001).
[4] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, 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* 101(23):e215-e220

[5] See http://cps.vghtpe.gov.tw
[6] Somers V. K., Dyken M. E., Mark A. L., Abboud F.M. N Engl J Med 1993; 328:303-307. Feb 4, 1993.

Address for correspondence.

Huey-Wen Yien
Address
Department of Anesthesia, Taipei Veterans General Hospital
201 Shipai Rd. Sec 2., Taipei
Taiwan R.O.C.
hwyiin@vghtpe.gov.tw

Appendix

The C program below is a generator function of RR interval time series. It is called once per RR interval.

```
float generate(void)
{

  float m;
  float p;
  int position;

  /* APC switcher */

if ((((RAND_MAX-(float)rand())/RAND_MAX)
<=((float)incidence_APC/100000)))
   { flag_rhythm = 1;
     flag_APC = 1;     }

  /* REM switcher */

if (((((RAND_MAX-(float)rand())/RAND_MAX) <=
((float)incidence_REM[(int)(total_t)/300]/1000)))
&& (flag_rhythm==0))
   { flag_rhythm = 2;
     inc_REM = 0;
      REM_RRlowlimit=0.6+(RAND_MAX-
(float)rand())/RAND_MAX*0.25;   }

switch(flag_rhythm)
{
   case   0:     /* normal sinus rhythm */

   m=copysignf(grandom()/660/(1+0.25*expf(logf(6
)/10*(index_LF+index_diurnal[(int)(total_t)/900]
)))*25,1);

   if (((int)(rr*1000) % 20)==0 )
  { position = (int)(rr*1000) / 20 ;   }
   else
  { position = ( (int)(rr*1000) / 20 )+1;   }

   p = Prob_short[seq_SF][position + index_LF +
index_diurnal[(int)(total_t)/900] ];

   if (((RAND_MAX-(float)rand())/RAND_MAX) < p )
   { seq_SF = ((seq_SF-((seq_SF>>7)<<7))<<1)+1;
     rr+=m; }
   else
   {seq_SF = (seq_SF-((seq_SF>>7)<<7))<<1;
    rr-=m; }
    break;

   case    1:     /* APC generator */

   switch(flag_APC)
   {
      case 1:           /* premature beat */

       rr=oldrr/(2+((RAND_MAX-
(float)rand())/RAND_MAX*0.1));
      flag_APC = 2;
      break;

       case    2:      /* Compensatory beat */

        rr=oldrr/2*(5+((RAND_MAX-
(float)rand())/RAND_MAX*0.1));
        flag_APC = 3;
        break;

       case    3:
        rr=oldrr/(5+((RAND_MAX-
(float)rand())/RAND_MAX*0.1))*4;
        flag_APC = 0;
        flag_rhythm=0;
        break;
         }

     break;       /* end of case 1 */

    case    2:  /* simulate REM sleep */

    if ((oldrr>=REM_RRlowlimit) && (inc_REM<30))
    { inc_REM+=1;
      m=copysignf(grandom()/500/inc_REM*55,1);
      if (m>0.1)
       m=copysignf(grandom()/500/inc_REM*30,1);
       rr-=m; }
      else
        { m=copysignf(grandom()/500*20,1);
          rr+=m;
          flag_rhythm=0; }

          break;          /* end of case 2 */

}
        total_t+=rr;      /* count time */

        if       (((int)(total_t)/60-(int)(total_t-
rr)/60)>0 )
       {      m=copysignf(grandom()/1000/((unsigned
int)index_diurnal[(int)(total_t)/900]+1)*16,1);

        if (   ((int)(mean_rr*1000) % 100)==0 )
        {position=(int)(mean_rr*1000) / 100 ;}
        else
        {position=((int)(mean_rr*1000) / 100)+1;}

        p=Prob_long[seq_LF][position];

        if (((RAND_MAX-(float)rand())/RAND_MAX)< p)
        {mean_rr+=m;
         seq_LF =((seq_LF-((seq_LF>>1)<<1))<<1)+1;}
        else
        {mean_rr-=m;
         seq_LF=(seq_LF-((seq_LF>>1)<<1))<<1; }

        index_LF=((int)(mean_rr*1000))/20-40;
        }

        oldrr = rr;
        return (rr);
   }
```