The Heartbeat Problem

Clara Hardy and Avnish Jadhav

Abstract:

This report presents a mathematical model of the heartbeat, employing a discrete-time framework implemented through MATLAB code. This model follows a threshold mechanism for triggering the electrochemical wave that induces ventricular contraction (systole), and accurately reflects the rapid transition back to relaxation (diastole) post-contraction. By formulating a system of discrete-time equations derived from the underlying differential equation model, we depict the dynamic behaviour of heart fibre length (x) and electrical conductance (b) over time. Our analysis reveals that a ventricular contraction or heartbeat, corresponds to a swift decrease in heart fibre length as the ventricular muscles contract to pump blood. The triggering of contraction is governed by a predefined threshold value of electrical conductance, indicative of the heart's electrical potential reaching a critical level. The MATLAB simulations demonstrate a coherent pattern where the rapid increase in electrical potential coincides with the contraction phase, facilitating the efficient ejection of blood from the heart into the circulatory system. This model was utilised to explore patterns of irregular heartbeats characteristic of various human diseases. This offers valuable insights into the mechanistic complexity of the heartbeat, with implications for understanding cardiovascular dynamics and potential applications in medical interventions and health management strategies.

Introduction:

During the heartbeat cycle there are two equilibrium states; systole and diastole. Systole corresponds to the contraction of the heart, and diastole corresponds to the relaxed state. The sinoatrial (SA) node is the pacemaker of the heart and generates an electrical stimulus which is then transmitted to the atrioventricular (AV) node. Upon receipt of the electrical stimulus the AV node tells the heart to contract. This is done by sending the signal into the muscles surrounding the ventricles of the heart. The heart's strong muscles around the ventricles contract which forces the blood out of the heart and to through blood vessels to the rest of the body. Each contraction of the heart's ventricles represents one heartbeat. Immediately after contraction the heart enters diastole, its muscle fibres relax and the heart begins to fill with blood again. The heart remains in a diastole state until the AV node receives the correct electrical stimulus to initiate a heartbeat once again. Under normal conditions, this happens 60-100 times per minute. (*Anatomy and Function of the Heart's Electrical System*, n.d.)

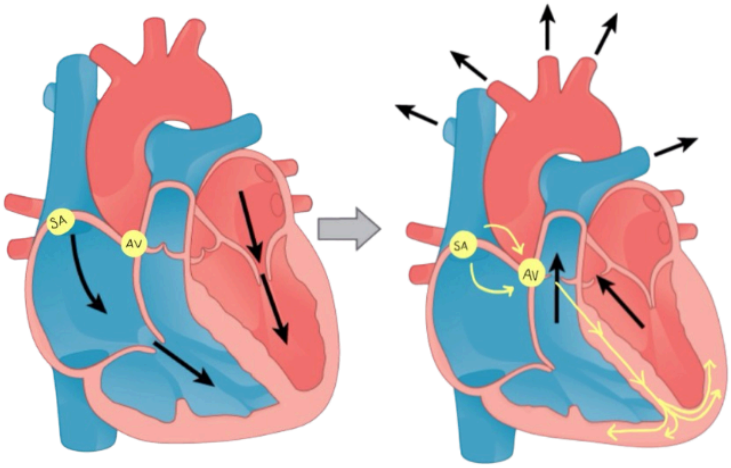

Figure 1: This figure of the heart shows both the pathway of the blood (black arrows) and the pathway of the electrochemical signal (yellow arrows) during a heartbeat. The left hand side shows the heart during diastole, the right hand side shows the heart during systole. (Lumen Learning, 2019)

Our goal is to describe the behaviour of the AV node which uses an electrical potential mechanism. When the AV node receives a signal from the SA node there are two options:

1. The electrical potential may be too high. This means that the heart is not yet ready to contract again, and the AV node ignores the signal, no contraction happens. The heart remains in diastole.
2. The electrical potential is at the proper threshold. This means that the AV node tells the heart to contract into systole. The contraction of the heart causes the electrical potential of the AV node to rapidly increase.

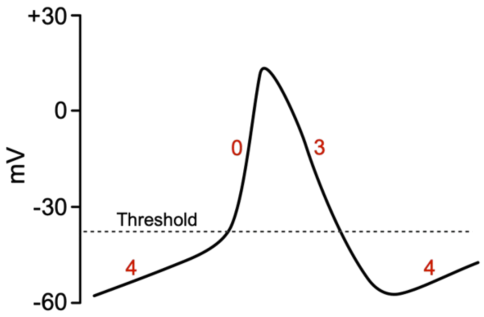

Figure 2: This figure shows the electrochemical signal of a heartbeat which will be represented in our model. As seen in the graph, when the threshold value is reached there is a rapid increase in charge, followed by a rapid decrease back to a value below the threshold.

Mathematical Model:

There are three criteria that our model must show:

1. There must be an equilibrium state in between contractions representing diastole.
2. The model should contain a threshold for triggering the electrochemical wave emanating from the AV node causing the heart to contract into systole.
3. The model should reflect the rapid return to diastole after contraction occurs.

Parameters:

$T > 0$ *is a constant that represents tension*
$e > 0$ *is a very small constant that depends on the timescale*

x_d *is the heart muscle fibre length during diastole*
x_s *is the heart muscle fibre length during systole*
b_d *is the electrical control variable of diastole*
b_s *is the electrical control variable of systole*
Note that: $x_d > x_s$

$x(t)$ *is the length of heart muscle fibre at time* t
$b(t)$ *is the electrical potential at time* t

System of Differential Equations:

$$\frac{dx}{dt} = -\frac{1}{e}(x^3 + Tx + b)$$
$$\frac{db}{dt} = x - x_d + (x_d + x_s)u$$

And we can define u by:

u: $b_s \leq b \leq b_d$ *and* $x^3 + Tx + b > 0$ *then* $u = 1$ (signal is within range to trigger contraction)

$b > b_d$ *then* $u = 1$ (there is no contraction because the signal is too high)

otherwise $u = 0$ (there is no contraction because the signal is too low)

This model is based off of a slightly improved version of E. C. Zeeman's model which was published in 1970. (Modelling in a heartbeat, 2018) (ORAL et al., 2019).

Assumptions:

- The heart contracts all at once. This is an assumption different from what really occurs in a heartbeat because we are neglecting the small atria contraction prior to ventricular contraction. In a realistic heartbeat there is an initial small contraction sending the blood within the heart from the atria to the ventricles, followed by the larger contraction which sends the blood from the ventricles to the rest of the body. We are only modelling one contraction of the heart therefore we neglect this small contraction.
- The increase of the electrical potential of the AV node is a constant and sharp pulse.
- The threshold for triggering the contraction is a constant value for every heartbeat.
- We assume a rapid and uniform return to diastole immediately following contraction.

Discrete Time Model:

We are able to define a system of discrete time equations based on our differential equation model to graph the system as a function of time.

$$x(t+1) = x(t) + \Delta t\left(-\frac{1}{e}\left(x^3 + Tx + b\right)\right)$$
$$b(t+1) = b(t) + \Delta t(x - x_d + (x_d + x_s)u)$$

For $\Delta t = 0.001$

Our goal with these discrete time equations is to plot both $x(t)$, the heart fibre length, and $b(t)$ the electrical conductance of the heart, as functions of time. By plotting this we can examine their behaviour during pulses, which represent contractions of the heart's ventricles (a heartbeat). What we want to show is that a pulse corresponds to a rapid decrease in heart fibre length x, as the muscles surrounding the ventricles squeeze together, pumping blood out of the heart.

The reason that a pulse is fired is because the electrical potential in the heart reaches a threshold level. When this threshold is reached this signals a rapid increase in electrical potential which makes the muscles contract and shorten. To set up this model we want to create a trigger where a certain b value sets off this rapid increase in electrical potential, causing a contraction into

systole. We did this in MATLAB by creating a threshold *b(i)* value to trigger an electrical pulse. Using the update formula *b(i+1)*, the heart's electrical potential determines its actions. This is achieved by reducing *b(i)* over time to begin natural decay, while increasing the response to stimulus, represented by *u*. Triggering the pulse depends on *b(i)'s* comparison with a defined threshold. This is shown when we check *if b(i) < b_threshold && ~inPulse*, ensures that a new contraction is initiated only when the electrical potential falls below a critical level, reflecting the heart's readiness for the next beat. (See MATLAB code in appendix).

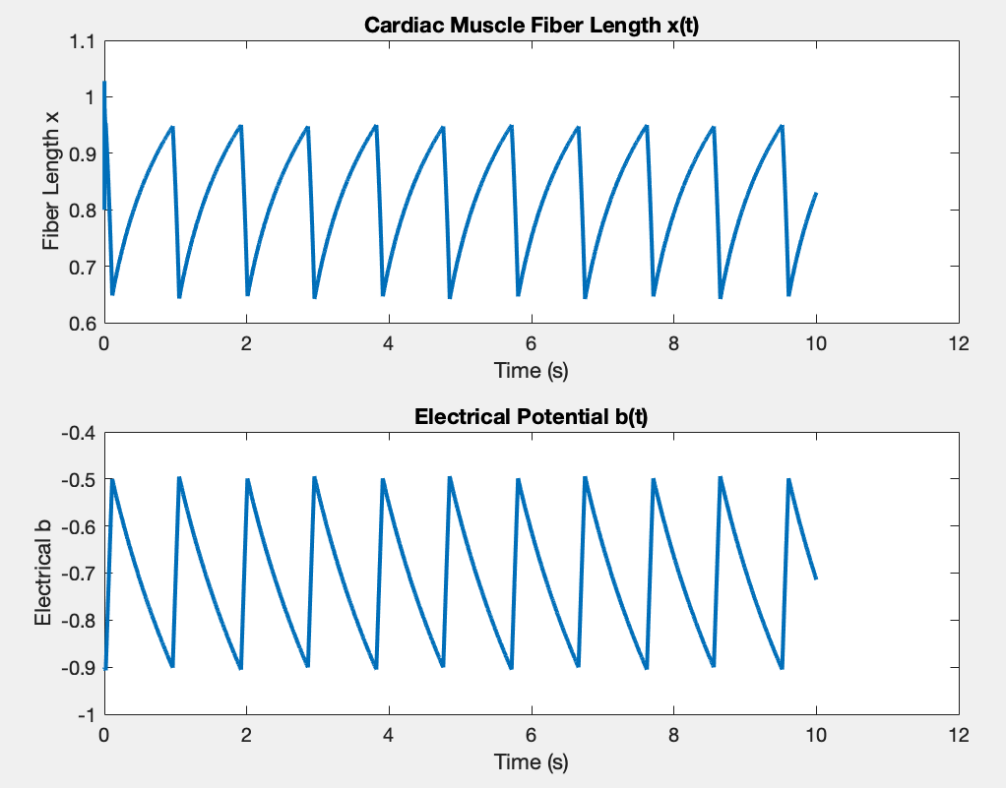

Figure 3: This graph represents our regular heartbeat model where the first graph, x(t), is the change in length of heart muscle fibres and the second graph, b(t) is the electrical potential of the heart.

Figure Results:

As seen in graphs, as the electrical potential rapidly increases the muscle fibre length rapidly decreases. When the electrical potential b(t) is at its peak the muscle fibre length x(t) is at a minimum because the heart is contracted. This is consistent with the heart entering systole and blood rapidly being pumped out into the body. After contraction, the electrical signal b(t) decreases and the muscle fibre length x(t) increases as the heart relaxes and fills with blood. This represents the heart returning to diastole. This graph shows a heartbeat every second, which refers to a heart rate of 60 BPM (beats per minute). This heart rate is common for an individual in a resting state, sitting or lying down. The y-axis on the x(t) graph shows a relative amount that

the heart fibre contracts. If the muscle fibre length when most relaxed and filled with blood is at its largest value of 1 (100% expanded), then as contraction occurs the fibres decrease to about 65% of their length (to value of 0.65). This is consistent with experimental data that we found in the literature (Molnar & Gair, 2019). The y-axis on the b(t) graph represents the electrical potential value. In our model the threshold value is -0.9, which is the electrical signal triggering a pulse. After a pulse is triggered there is a rapid increase in this electrical signal which represents systole. After the contraction occurs, the electrical signal decreases again until it reaches another threshold of -0.9 and another pulse is fired.

Irregular Heartbeat Patterns:

We want to apply our model to three different irregular heartbeat patterns which are real human conditions. The three medical conditions which we examined are bradycardia, first degree atrioventricular block and Wenckebach atrioventricular block. Bradycardia is often associated with damage to heart tissue and refers to an overall low heart rate. As mentioned, regular daily human heart rate is considered to be between 60 to 100 BPM, and bradycardia is considered to be any heart rate lower than 60 BPM (Mayo Clinic, 2017). First degree AV block is when every other signal is partially or completely blocked from the AV node. In other words, when the signal reaches the AV node either only a fraction of this signal is actually sent, or no signal is sent at all. This causes the heart to partially or completely skip every other heart beat (*Atrioventricular Block - Cardiovascular Disorders*, n.d.). Wenckebach AV block is when the signal gets randomly interrupted, so sometimes the heart rate is normal and sometimes the heart skips a beat. We modelled all of these cases based on altering parameters of our regular heartbeat model. To model the Wenckebach AV block we introduced a random function which gives a different output every time we run the code. This was done to reflect the nature of heartbeats for someone suffering from Wenckebach AV block. The *random_pulse* mimics the irregular behaviour, where heartbeats are skipped at random intervals. The randomness factor (0.1 in this case) influences the frequency of skipped beats, and adjusting this threshold can vary the model to show different severities.

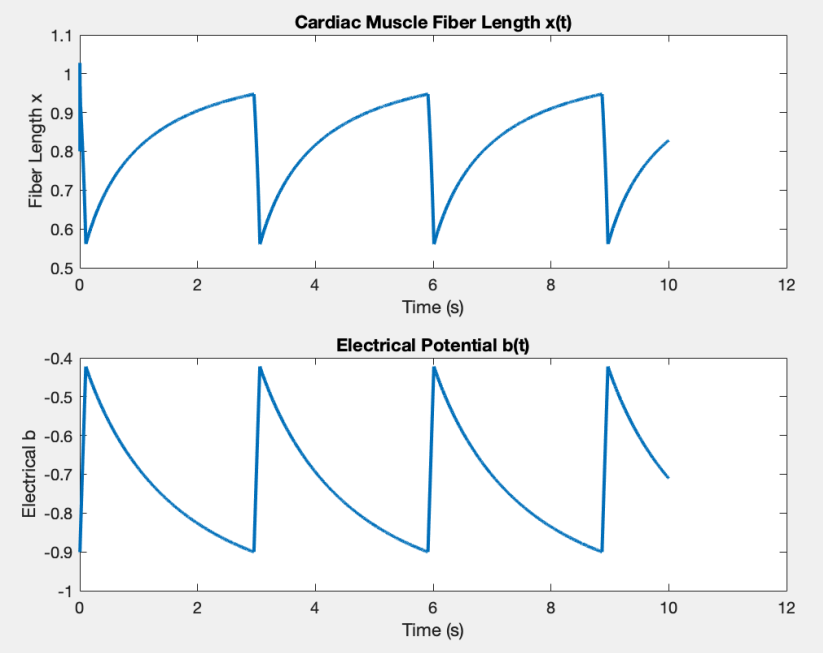

Figure 4: This graph represents the heartbeat model of bradycardia, a condition where the heart rate is very low. See that contraction only happens every three seconds.

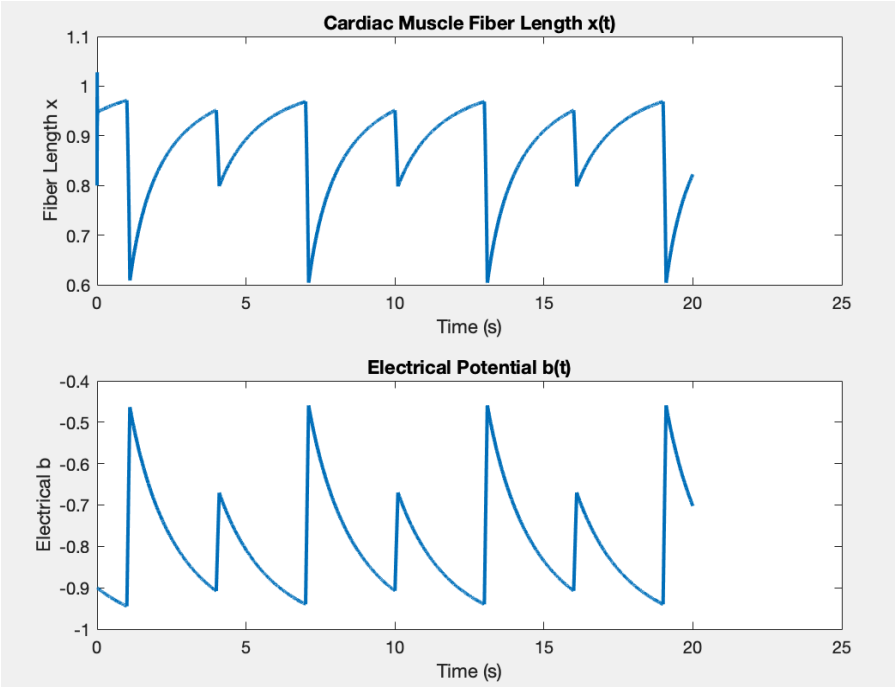

Figure 5: This graph represents the heartbeat model of first degree AV block where the heart only completes a regular contraction every second beat. As seen in the graph, every second pulse is small which represents a partial electrical signal, b, and a partial contraction of the muscle fibre, x.

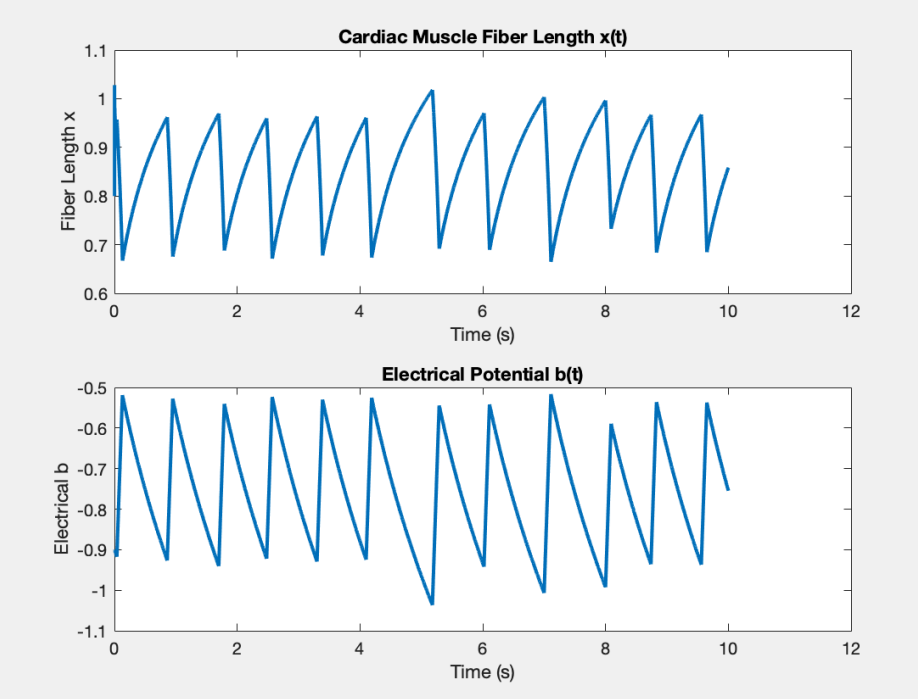

Figure 6: This figure represents the heartbeat model of the Wenckebach AV block where there are random impartial beats mixed in with normal heart beats.

Irregular Heartbeat Figure Results:

Figure 4 represents bradycardia. We can see on the graph that the heart rate is much slower as a heartbeat is happening only every 3 seconds, giving a heart rate of only 20 BPM. 20 BPM is much lower than regular resting heart rate (of around 60 BPM) and likely requires medical attention. A very low heart rate can be very dangerous as your body is pumping blood to its muscles and brain much less frequently, therefore the muscles and brain are receiving a much lower amount of oxygen. This can lead to hypoxia, where the brain is starved of oxygen. Severe hypoxia results in permanent brain damage, seizures or death. It is also important to notice from the graph that the electrical signal and the magnitude of muscle fibre contraction is consistent with the regular heartbeat model, the only difference is that the beat occurs much less frequently.

Figure 5 represents first degree AV block. We can see on the graph that There is a normal heartbeat and then there is a partially blocked heartbeat, and this pattern continues. The blocked beat is demonstrated by a smaller electrical signal sent, which results in a smaller contraction of the heart muscle fibres associated with this pulse. This means that when the AV node blocks the signal and a smaller electrical pulse is sent the heart's size doesn't contract as much and a smaller amount of blood is squeezed from the heart to the rest of the body. Therefore every other heartbeat, the muscles and brain receive a smaller amount of oxygen. Severe cases of starving the muscles and brain of oxygen can have similar consequences as discussed in bradycardia.

Figure 6 represents the Wenckebach AV block. This graph is only one of the many outcomes that are possible when running this code, as we modelled this with a random function so every time running the code results in a different outcome. We can see on this graph that the heartbeat is regular until around 5 seconds and then an irregular pattern starts. During this irregular random pattern there are some larger electrical signals as well as some smaller electrical signals sent. The larger signals correspond to a larger magnitude of heart muscle fibres contracting, and the smaller signals correspond to a smaller heart fibre contraction. Wenckebach AV block can have severe consequences if the pulses deviate far from the regular beating pattern and there are too many small electrical signals sent.

Critiques of the Model:

There are many critiques to our model mostly due to the fact that although this is a very complex problem, we did have to make it manageable for us to solve. This involved making many assumptions which are listed at the beginning of this report. All of these assumptions lead to our model being slightly less realistic, but without assuming them our model starts to become too complex. In addition, there are many factors that affect a person's heart rate. Heart rate varies within the person as well as between different people. For example depending on if someone is sitting vs standing vs walking vs sprinting, or if they are too hot or cold, or if they are highly stressed,... (and many more factors), all affect an individual's heart rate at any given time. Between different people heart rate is different depending on the person's health, physical activity, age, smoking habits, and the list goes on (Nikolova Georgieva-Tsaneva & Gospodinova, n.d.). Another critique is that heart rate is rarely as uniform as seen in our model. It isn't common in real life that someone is keeping a completely constant heart rate, therefore we believe our model makes the most sense for someone sleeping or maintaining rest lying down. Lastly, a critique is based on our model showing a standard amount which the heart muscle fibre shortens. In real life the amount that the heart fibre shortens can actually vary between different people, so this could be something that could be taken into consideration to create a more robust model. (Nikolova Georgieva-Tsaneva & Gospodinova, n.d.).

Future Research:

In the future it would be interesting to examine some other related questions and how we can incorporate them into our model. Such as:

1. How can humans control their heart rate (by controlling breathing), therefore controlling the firing of the SA and AV nodes?
    Humans are obviously able to control their breathing. Our model only considered a continuous and constant heart rate, but it would be interesting to examine how humans can actively pause breathing or increase breathing rate.
2. How do mechanical pacemakers work for people with various types of heart disease where their SA node is unable to act as a regular pacemaker?
    Mechanical pacemakers are a strategy for people who suffer from various irregular beating pattern diseases. It would be interesting to examine how these machines work by initiating an electrical signal, and to try to fit this to our model.

References:

Anatomy and Function of the Heart's Electrical System. (n.d.). Www.hopkinsmedicine.org. https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-and-function-of-the-hearts-electrical-system#:~:text=An%20electrical%20stimulus%20is%20generated

Atrioventricular Block - Cardiovascular Disorders. (n.d.). Merck Manuals Professional Edition. https://www.merckmanuals.com/professional/cardiovascular-disorders/specific-cardiac-arrhythmias/atrioventricular-block

Lumen Learning. (2019). *The Cardiac Cycle | Biology for Majors II*. Lumenlearning.com. https://courses.lumenlearning.com/wm-biology2/chapter/the-cardiac-cycle/

Mayo Clinic. (2017). *Bradycardia - symptoms and causes*. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/bradycardia/symptoms-causes/syc-20355474

Modelling in a heartbeat. (2018, February 14). Chalkdust. https://chalkdustmagazine.com/blog/heartbeats/

Molnar, C., & Gair, J. (2019, May). *19.4 Muscle Contraction and Locomotion – Concepts of Biology – 1st Canadian Edition*. Opentextbc.ca. https://opentextbc.ca/biology/chapter/19-4-muscle-contraction-and-locomotion/

Nikolova Georgieva-Tsaneva, G., & Gospodinova, P. (n.d.). *Mathematical technologies for modeling of cardiological data: Heart Rate Variability*. Retrieved April 11, 2024, from https://www.dpublication.com/wp-content/uploads/2020/11/8-8008.pdf

ORAL, P., MERDAN, M., & BEKİRYAZICI, Z. (2019). Analysis of a Random Zeeman Heartbeat Model with Differential Transformation Method. *Cumhuriyet Science Journal*, *40*(2), 285–298. https://doi.org/10.17776/csj.460984

Appendix:

MATLAB CODE:

Regular Heart Beat Model:

```
% Initialization
clear all;
deltat = 0.001;
xd = 1;     % Typical relaxed fiber length during diastole
xs = 0.5;   % Assumed contraction fiber length
bd = 1;     % Baseline parameter for diastole
bs = 1.5;   % Adjusted baseline parameter for systole, assuming it should be
higher
T = 1;      % Constant proportional to tension
e = 0.001; % Small constant for timescale
N = 10000; % Number of iterations to cover more heartbeats
% Parameters for automatic pulsing based on 'b' value
b_threshold = -0.9;  % b value threshold for triggering a pulse
pulseWidth = 90;     % Width of the pulse in iteration steps, simulate the pulse
duration
pulseFrequency = 20; % Frequency of pulses in terms of iteration steps
AV_decay_rate = 0.3; % Rate of potential decrease when no signal received
inPulse= false;
% Initial values
x = zeros(1, N+1);
b = zeros(1, N+1);
x(1) = 0.8;
b(1) = -0.9;
u = 0;
% Simulation with automatic pulsing based on 'b' value
for i = 1:N
   t = i * deltat; % Current time step

   % Check if it's time to start a new pulse based on frequency
   if mod(i, pulseFrequency) == 0
       % Check if 'b' falls below the threshold to trigger a pulse
       if b(i) < b_threshold && ~inPulse
           inPulse = true;
           pulseCounter = pulseWidth; % Reset pulse counter
       end
   end

   % Generate pulse if in pulse period
   if inPulse
       u = 1;
       pulseCounter = pulseCounter - 1;
       if pulseCounter <= 0
           inPulse = false; % End of pulse
       end
   else
       u = 0;
   end

   % Update b based on the current state and presence of a pulse
   b(i+1) = b(i) - AV_decay_rate * deltat + deltat * ((x(i) - xd) + 10 * (xd -
xs) * u);
```

```
    % Update x with added non-linearity for realism
    x(i+1) = x(i) + deltat * ((-1/e) * (x(i)^3 - x(i)^2 + T * x(i) + b(i)));
end
% Plot results
t_start = 0;  % Start time
t_end = 10;   % End time
idx_start = round(t_start / deltat) + 1;
idx_end = round(t_end / deltat) + 1;
figure(1)
subplot(2,1,1);
plot((idx_start:idx_end) * deltat, x(idx_start:idx_end), 'LineWidth', 2);
title('Fiber Length x(t)');
xlabel('Time');
ylabel('x');
subplot(2,1,2);
plot((idx_start:idx_end) * deltat, b(idx_start:idx_end), 'LineWidth', 2);
title('Electrical Control Variable b(t)');
xlabel('Time');
ylabel('b');
```

Bradycardia Model:

```
% Initialization
clear all;
deltat = 0.001;
xd = 1;     % Typical relaxed fiber length during diastole
xs = 0.5;   % Assumed contraction fiber length
bd = 1;     % Baseline parameter for diastole
bs = 1.5;   % Adjusted baseline parameter for systole, assuming it should be
higher
T = 1;      % Constant proportional to tension
e = 0.001; % Small constant for timescale
N = 10000; % Number of iterations to cover more heartbeats
% Parameters for automatic pulsing based on 'b' value
b_threshold = -0.9;  % b value threshold for triggering a pulse
pulseWidth = 100;    % Width of the pulse in iteration steps, simulate the
pulse duration
AV_decay_rate = 0.005; % Rate of potential decrease when no signal received
% Initial values
x = zeros(1, N+1);
b = zeros(1, N+1);
x(1) = 0.8;
b(1) = -0.9;
u = 0;
% Flag to control pulse generation
inPulse = false;
pulseCounter = 0;
% Automatic pulsing based on 'b' value
for i = 1:N
    t = i * deltat; % Current time step

    % Check if it's time to start a new pulse
    if b(i) < b_threshold && ~inPulse
```

```
        inPulse = true;
        pulseCounter = pulseWidth; % Reset pulse counter
    end

    % Generate pulse if in pulse period
    if inPulse
        u = 1;
        pulseCounter = pulseCounter - 1;
        if pulseCounter <= 0
            inPulse = false; % End of pulse
        end
    else
        u = 0;
    end

    % Update b based on the current state and presence of a pulse
    b(i+1) = b(i) - AV_decay_rate * deltat + deltat * ((x(i) - xd) + 10 * (xd -
xs) * u);

    % Update x with added non-linearity for realism
    x(i+1) = x(i) + deltat * ((-1/e) * (x(i)^3 - x(i)^2 + T * x(i) + b(i)));
end
% Plot results
t_start = 0;  % Start time
t_end = 10;   % End time
idx_start = round(t_start / deltat) + 1;
idx_end = round(t_end / deltat) + 1;
figure(1)
subplot(2,1,1);
plot((idx_start:idx_end) * deltat, x(idx_start:idx_end), 'LineWidth', 2);
title('Cardiac Muscle Fiber Length x(t)');
xlabel('Time (s)');
ylabel('Fiber Length x');
subplot(2,1,2);
plot((idx_start:idx_end) * deltat, b(idx_start:idx_end), 'LineWidth', 2);
title('Electrical Potential b(t)');
xlabel('Time (s)');
ylabel('Electrical b');
```

First Degree Block Model:

```
% Initialization
clear all;
deltat = 0.001;
xd = 1;    % Typical relaxed fiber length during diastole
xs = 0.5;  % Assumed contraction fiber length
bd = 1;    % Baseline parameter for diastole
bs = 1.5;  % Adjusted baseline parameter for systole, assuming it should be
higher
T = 1;     % Constant proportional to tension
e = 0.001; % Small constant for timescale
N = 20000; % Number of iterations to cover more heartbeats
% Parameters for automatic pulsing based on 'b' value
b_threshold = -0.9;  % b value threshold for triggering a pulse
pulseWidth = 100;    % Width of the pulse in iteration steps, simulate the
```

```
pulse duration
pulseFrequency = 1000; % Frequency of pulses in terms of iteration steps
AV_decay_rate = 0.005; % Rate of potential decrease when no signal received
inPulse = false;
atrialSignalCounter = 0; % Initialize counter for atrial signals
% Initial values
x = zeros(1, N+1);
b = zeros(1, N+1);
u_values = zeros(1, N+1); % To store the pulse state for visualization
x(1) = 0.8;
b(1) = -0.9;
u = 0;
% Simulation with automatic pulsing based on 'b' value
for i = 1:N
   t = i * deltat; % Current time step

   % Check if it's time to start a new pulse based on frequency
   if mod(i, pulseFrequency) == 0
       atrialSignalCounter = atrialSignalCounter + 1; % Increment atrial signal
counter
       % Check if 'b' falls below the threshold to trigger a pulse
       if b(i) < b_threshold && ~inPulse
           inPulse = true;
           pulseCounter = pulseWidth; % Reset pulse counter
           if mod(atrialSignalCounter, 2) == 0
               pulseAmplitude = 0.5; % Reduce the amplitude for every other
pulse
           else
               pulseAmplitude = 1; % Normal pulse
           end
       end
   end

   % Generate pulse if in pulse period
   if inPulse
       u = pulseAmplitude; % Use the modified amplitude
       pulseCounter = pulseCounter - 1;
       if pulseCounter <= 0
           inPulse = false; % End of pulse
       end
   else
       u = 0;
   end

   u_values(i) = u; % Store the current value of u for visualization

   % Update b based on the current state and presence of a pulse
   b(i+1) = b(i) - AV_decay_rate * deltat + deltat * ((x(i) - xd) + 10 * (xd -
xs) * u);

   % Update x with added non-linearity for realism
   x(i+1) = x(i) + deltat * ((-1/e) * (x(i)^3 - x(i)^2 + T * x(i) + b(i)));
end
% Plot results
t_start = 0;  % Start time
t_end = 20;   % End time
idx_start = round(t_start / deltat) + 1;
```

```
idx_end = round(t_end / deltat) + 1;
figure(1)
subplot(2,1,1);
plot((idx_start:idx_end) * deltat, x(idx_start:idx_end), 'LineWidth', 2);
title('Cardiac Muscle Fiber Length x(t)');
xlabel('Time (s)');
ylabel('Fiber Length x');
subplot(2,1,2);
plot((idx_start:idx_end) * deltat, b(idx_start:idx_end), 'LineWidth', 2);
title('Electrical Potential b(t)');
xlabel('Time (s)');
ylabel('Electrical b');
%subplot(3,1,3); % Plotting the pulse signal
%plot((idx_start:idx_end) * deltat, u_values(idx_start:idx_end), 'LineWidth',
2);
%title('Pulse Signal u(t)');
%xlabel('Time');
%ylabel('u');
```

Wenckebach Model:

```
% Initialization
clear all;
deltat = 0.001;
xd = 1;     % Typical relaxed fiber length during diastole
xs = 0.5;   % Assumed contraction fiber length
bd = 1;     % Baseline parameter for diastole
bs = 1.5;   % Adjusted baseline parameter for systole, assuming it should be
higher
T = 1;      % Constant proportional to tension
e = 0.001; % Small constant for timescale
N = 10000; % Number of iterations to cover more heartbeats
% Parameters for automatic pulsing based on 'b' value
b_threshold = -0.9;  % b value threshold for triggering a pulse
pulseWidth = 90;     % Width of the pulse in iteration steps, simulate the pulse
duration
pulseFrequency = 20; % Frequency of pulses in terms of iteration steps
AV_decay_rate = 0.4; % Rate of potential decrease when no signal received
inPulse = false;
% Initial values
x = zeros(1, N+1);
b = zeros(1, N+1);
x(1) = 0.8;
b(1) = -0.9;
u = 0;
% Simulation with automatic pulsing based on 'b' value
for i = 1:N
   t = i * deltat; % Current time step

   %Randomness to pulse generation
   %generates a random number between 0 and 1, and if it's less than 0.1
(adjust this threshold as needed), a pulse is generated.
   random_pulse = rand < 0.1; % Adjust the probability to control pulse
frequency
```

```
    % Check if it's time to start a new pulse based on frequency and randomness
    if mod(i, pulseFrequency) == 0 && random_pulse
        % Check if 'b' falls below the threshold to trigger a pulse
        if b(i) < b_threshold
            inPulse = true;
            pulseCounter = pulseWidth; % Reset pulse counter
        end
    end

    % Generate pulse if in pulse period
    if inPulse
        u = 1;
        pulseCounter = pulseCounter - 1;
        if pulseCounter <= 0
            inPulse = false; % End of pulse
        end
    else
        u = 0;
    end

    % Update b based on the current state and presence of a pulse
    b(i+1) = b(i) - AV_decay_rate * deltat + deltat * ((x(i) - xd) + 10 * (xd -
xs) * u);

    % Update x with added non-linearity for realism
    x(i+1) = x(i) + deltat * ((-1/e) * (x(i)^3 - x(i)^2 + T * x(i) + b(i)));
end
% Plot results
t_start = 0;  % Start time
t_end = 10;   % End time
idx_start = round(t_start / deltat) + 1;
idx_end = round(t_end / deltat) + 1;
figure(1)
subplot(2,1,1);
plot((idx_start:idx_end) * deltat, x(idx_start:idx_end), 'LineWidth', 2);
title('Cardiac Muscle Fiber Length x(t)');
xlabel('Time (s)');
ylabel('Fiber Length x');
subplot(2,1,2);
plot((idx_start:idx_end) * deltat, b(idx_start:idx_end), 'LineWidth', 2);
title('Electrical Potential b(t)');
xlabel('Time (s)');
ylabel('Electrical b');
```