MAE 3260 Final Group Work: Exploring a System of Interest

Report

Outline (two examples given below, there could be others):

Option #1: one report, same grade for all	**Option #2: separate sections/grades for each**
Page 1: cover page Pages 2-9: any format you want in these pages, just include a few section headings in this outline) Page 10: References	Page 1-2: cover page Page 3-4: Logan technical summary Page 5-6: Thomas technical summary Page 7-8: Ana technical summary Page 9-11: Jess technical summary Page 12: References

Title: Rushing or Dragging? Modelling a Pacemaker.

Topic of Interest: Cardiovascular System <3🫀 (specifically pacemakers)

Abstract: As a group, our mutual interests guided us to choose a cardiovascular system to explore. More specifically, we decided to model a single-lead pacemaker due to its well-defined mechanisms that share familiar facets to what we've seen through our experience in 3260. We plan to study the block diagram(s) that make up the system, – potentially the ODEs as well – the steady state behavior of the open-loop part of the system, and the feedback control law associated with the active control of the pacemaker. We intend to produce a system diagram and a matlab script including different heart conditions where pacemakers apply. It is our hope to use engineering principles to demonstrate the connection between naturally occurring dysfunctions of the heart and the man-made technology induced upon it to return rhythm and beat pace to normal/acceptable values.

Students/Roles:

Student	**Task/Role**	**Portfolio**
Thomas Wells	Create block diagrams of the transfer function model proposed by [1]. Recreate the model in MATLAB and apply the model to a new heart condition.	https://cornell-mae-ug.github.io/Thomas-Wells-Portfolio/projects/2025-2-ECG-Model/
Logan Roberts	I described the P-wave as a second order transfer function modeling atrial depolarization, showing how its	Submitted via Canvas

	gain, delay and pole/zero locations control the wave's amplitude, width, and timing in the ECG. I then analyzed how changing these parameters alter the P-waves shape and position without affecting the QRS and T waves.	
Ana Badea	I found and read the VDP paper, from which I produced the ODE. I then wrote a matlab script to model the heartbeat based on the paper and a pacemaker model more so based on intuition and general descriptions of the pacemaker.	https://cornell-mae-ug.github.io/fa25-portfolio-amb675-Cornell/projects/2025-pacemaker-model/
Jess Rakoczy	Analyze the dynamics of using the VDP model to model a heartbeat and its connection to a pacemaker, displaying the state space and block diagram of the heart-pacemaker system.	https://cornell-mae-ug.github.io/fa25-portfolio-jlr444/projects/Model-Pacemaker-MAE3260/

List of MAE 3260 concepts or skills used in this group work:

Models:

- **ODEs**
- **Block diagrams**

Open-loop system:

- **Steady state behavior**

Active control:

- **Feedback control law**
- **Command following**

P-WAVE MODEL INTRODUCTION

The P-wave model used in this project is based on a transfer function framework that synthesizes electrocardiogram (ECG) singles using three parallel linear subsystems, one for each of the principal ECG waves: P, QRSQ, and T[1]. In a physiological ECG, the P wave is generated by the depolarization of the atria, occurring before ventricular activation and thus before the QRS complex. Since it reflects atrial conduction and pacemaker behavior, accurate modeling of the P wave is essential for capturing sinus rhythm and many atrial conditions. Many detailed biophysical heart models can reproduce this behavior, but they are very complex and expensive to stimulate. The idea in [1] is to replace those large models with a much smaller linear system that still produces waveforms that look like clinical ECGs. In this setup, the P wave is treated as its own second order system, tuned directly from ECG data, so we can adjust atrial behavior in a clear and controlled way [1]

Within this framework, the P-wave is represented by the first term of a three branch transfer function written in the Laplace domain.The complete heartbeat model is expressed as a sum of three delayed second order transfer functions, but the P-wave can be written as

$$HB(s) = \sum_{i=1}^{3} k_i e^{-r_i s} \frac{a_i s - b_i}{s^2 + c_i s + d_i}, \quad (12)$$

Which is the $i = 1$ term of Equation (12) in [1]. Here, k_1 is the gain associated with the P-wave subsystem, r_1 is its time delay relative to the excitation impulse, and the coefficients a_1, b_1, c_1 and d_1 define the location of one zero and a pair of complex conjugate poles in the transfer function. Since this is a proper second order system with a first order numerator, its impulse response exhibits the damped, asymmetric shape that is typical of atrial depolarization. In contrast to a standard second order form, the non standard representation used here explicitly includes the numerator zero, allowing more flexible control of the initial slope and skewness of the waveform [1].

Each parameter in the P-wave transfer function alters a specific, visible feature of the waveform. The gain k_1 stretches the P-wave up or down without changing its width, so it controls the amplitude, which in real signals is usually on a few tenths of a millivolt. The delay r_1 shifts the entire P-wave in time and is used to match the observed PR interval (the time from the start of the P-wave to the start of the QRS complex) [1]. The coefficient a_1 sets the location of the zero. Moving this zero changes the initial slope and asymmetry between the upstroke and the downstroke of the P-wave, so it can model slower or faster atrial conduction. The pair b_1 and d_1 are linked to the natural frequency of the underlying second order oscillator. Changing them makes the P-wave narrower or wider in time. The coefficient c_1 controls the damping: higher damping makes the waveform quickly settle back to baseline with little ringing, while lower damping would allow small oscillations after the peak.

Rodríguez-Abreo et al. chose a non standard form for these second order transfer functions, meaning that the numerator and denominator coefficients are left free instead being written only in terms of natural frequency and damping ratio [1]. This choice is important for the P-wave model. Since the zero location is an independent variable parameter, the model can fine tune the skew and tilt of the P wave in addition to its basic width and height. That extra flexibility is useful because real P-waves are often slightly asymmetric and can vary in shape with different atrial conditions. In other words, the non-standard form makes it easier to match the subtle details of the P-wave without disturbing the QRS or T-wave branches.

To find acceptable parameter values for the P-wave model, the authors use a generic algorithm rather than manual tuning [1]. The algorithm starts with many random guesses for the parameters a_1, b_1, c_1, d_1, k_1 and r_1, simulates the heartbeat for each guess, and measures how close the result is to a target ECG beat. The best guesses are kept and combined to form a new generation of parameters, and P-wave branch is separated from the other branches, the algorithm can automatically adjust its parameters to better fit the small atrial deflection while the QRS and T-wave model that is consistent with the full ECG but still controlled by a small set of clear parameters.

The heartbeat formula in Equation (12) describes only one beat. To create a continuous ECG over many beats, the model uses Equation (13), which adds a periodic input term. For the P-wave branch, this becomes

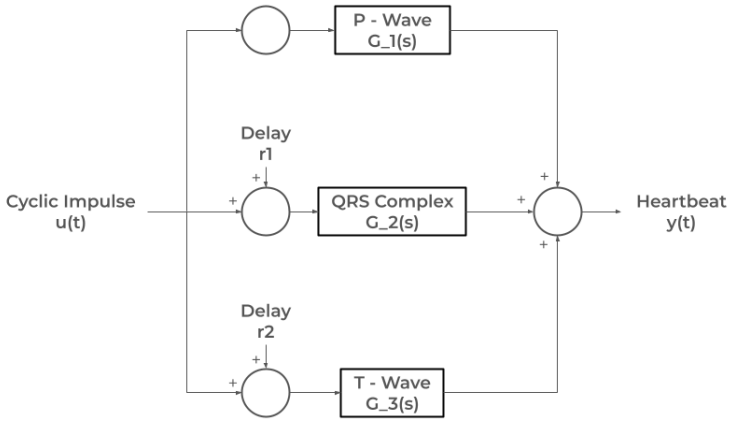
$$ECG(s) = \sum_{i=1}^{3} \frac{k_i e^{-r_i s}}{1 - e^{\frac{s}{f}}} \frac{a_i s - b_i}{s^2 + c_i s + d_i}, \quad (13)$$

Where f is the frequency of the impulse train (beats per second) [1]. The factor $1/(1 - e^{-s/f})$ is the Laplace transform of an infinite sequence of impulses separated by a time $T = 1/f$. This means the P-wave response of Equation (12) is simply repeated once every beat, always with the same shape and delay, while the value of f sets the heart rate. A slow heart rate uses a small f, so the P-waves are farther apart in time. A fast heart rate uses a larger f, bringing the P-waves closer together. The important point is that changing f does not change the shape of the P wave itself. Shape is controlled only by a_1, b_1, c_1, d_1, k_1 and r_1, while beat to beat timing is controlled by f.

Since each ECG component is modeled as its own branch, the P-wave subsystem can also be modified or removed independently, which is useful for representing atrial arrhythmias. It is a controlled, interpretable handle on atrial behaviors within a simple linear structure that still captures the essential features of human ECG signals.

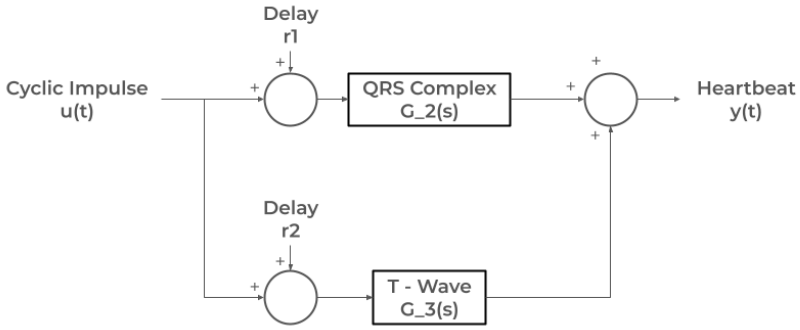
TF MODEL DYNAMICS

A matlab script was made that provides an input to the transfer functions given by equations (12) and (13) in [1]. A block diagram of the system is given below.



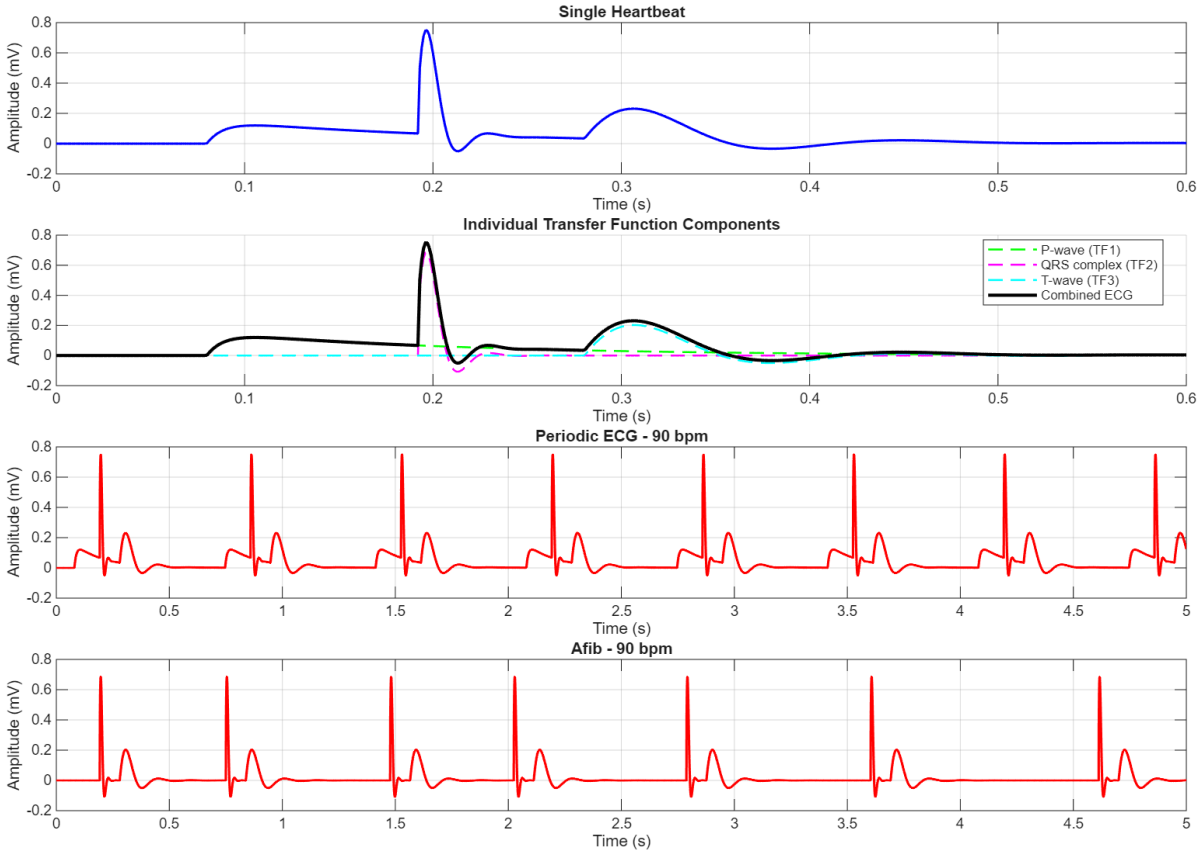
While the transfer functions for each system are of the same form, the value of each parameter used to describe each transfer function differs. While sample values for these parameters are provided in [1], a description of the input u(t) is only described as a “cyclic impulse”. Through experimentation, a reasonable amplitude for the input function was determined, and produced outputs that were similar to those achieved by the paper. Another successful method involved setting the impulse at a constant amplitude, and re-tuning parameters k1, k2, and k3, the gains for each function. This method was also effective, but was less efficient than varying the input, as tuning each gain independently did not meaningfully improve the result.

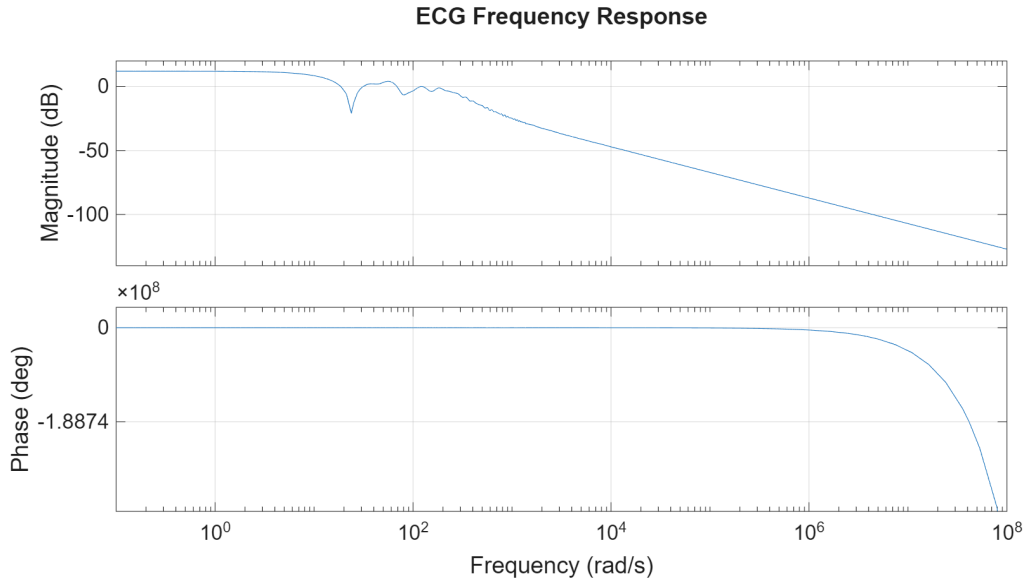
Omar et al. [1] applied this model to several common heart conditions, showcasing its ability to reproduce the signals generated by these conditions. This inspired me to model atrial fibrillation (afib), a condition that I was diagnosed with at nine years of age but have since recovered from. The defining characteristics of afib are an irregular heartbeat at a rate of 90bpm, and the absence of a P-wave [2]. To simulate the irregular heartbeat, a random delay was added to the cyclic impulse signal. Removing the P-wave involved removing the transfer function that described the P-wave from the overall system, an updated block diagram is shown below.



The modified model was able to generate a signal that somewhat resembles that of a patient with afib, but failed in portraying the condition in its entirety.

The full output of the matlab script is given below. This includes plots showing a single heartbeat along with the contributions of the individual transfer functions, a “perfect” ECG signal at 90bpm, and an ECG signal exhibiting characteristics of afib at 90bpm. Bode plots for the system are also provided, with a major valley at 23.7 rad/s.





VAN DER POL INTRODUCTION

A second model we found of the heart and the pacemaker was based off of this paper by Yahalom & Puzanova [3]. We chose to do two separate models as that gives us two different approaches to modeling the heart. With the P-wave model modeling a more electrical approach and this Van der Pol model modelling the physical oscillations of the heartbeat. Having both models helps us understand the system as a whole and how this rhythmic behavior can be modelled mathematically.

This is the Van der Pol equation:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$$

It's a second order system with a nonlinear damping term, $(1 - x^2)x'$, which can model the "spike-and recovery" shape seen in biological processes like heartbeats.

The paper took this system and modified it to this equation to model a beating heart. Which gives us a forced, nonlinear, 2nd order system.

$$\frac{d^2x}{dt^2} + a(x - v_1)(x - v_2)\frac{dx}{dt} + \frac{x(x+d)(x+e)}{ed} = F(t)$$
$$F(t) = Asin(\omega t)$$

Where:

- x(t) represents the internal pacemaker signal
- a, v_1, v_2 help give us the nonlinear damping system and model asymmetries in contraction and relaxation of the heart
- d and e help modify our restoring force and control cycle timing
- And F(t) is the heart's natural driving force

The parameters to simulate this system with irregular heartbeats are provided in the paper.

```
parameters from paper for just a heartbeat
a = 0.5;
v1 = .97;
v2 = -1;
d = 3;
e = 6;
A = 2.5;
w = 1.5;
tspan = [0,30];
```

```
% initial conditions
X0 = [-0.1; 0.02];
```

This equation can be rewritten into 2 1st order ODEs to make our MATLAB simulation easier.

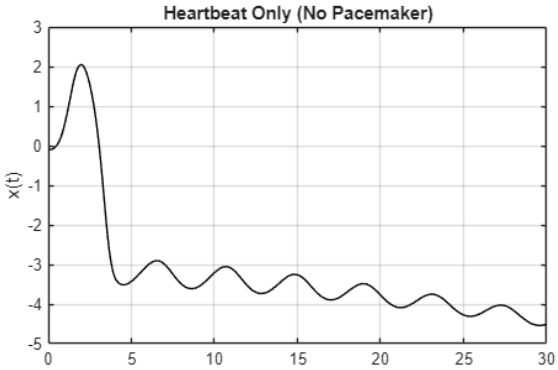
(1) $x'(t) = y(t)$

(2) $y'(t) = -a(x(t) - v_1)(x(t) - v_2)y(t) - \frac{x(t)(x(t)+d)(x(t)+e))}{ed} + Asin(\omega t)$

Where y(t) is the amplitude of the heartbeat signal voltage. Below is the ODE model in MATLAB:

```
f_hb = @(t,X) [X(2);
-a*(X(1)-v1)*(X(1)-v2)*X(2) - (X(1)*(X(1)+d)*(X(1)+e)/(e*d)) + A*sin(w*t)];
```

```
[t1,X1] = ode45(f_hb,tspan,X0);
```

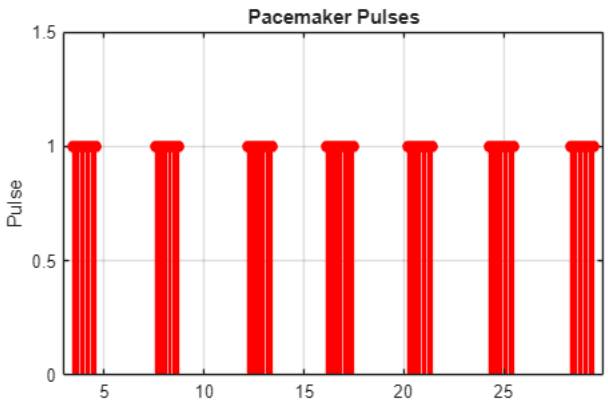


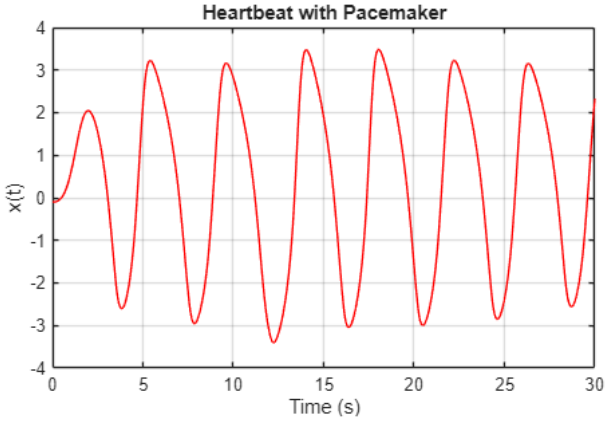
Instead of using the integral control method mentioned in the paper, we chose a more intuitive and open loop model for our pacemaker.
Our pacemaker's logic is that the pacemaker detects the natural heartbeat, and intercept (by actuating our pacemaker and sending out a step input to go into our F(t)) if too much time passes between heartbeats.

I modeled this is MATLAB like so:

```
    %detect our nat beat
    if x > xth && (t - last_beat) > 0.2
        last_beat = t;
    end
    %pacemaker pulse
    pacer = 0;
    if (t - last_beat) >= Tp && (t - t_stim) > pulse_dur
        t_stim = t;
    end
    if (t - t_stim) <= pulse_dur
        pacer = I0;
        record_pulse(t); % store pulse event
    end
    %heart ODE w pulse added
    dx = y;
    dy = -a*(x-v1)*(x-v2)*y - (x*(x+d)*(x+e)/(e*d)) + A*sin(w*t) + pacer;
    dX = [dx; dy];
```

```
f_pacemaker = @(t,X)
pacemaker_sys(t,X,a,v1,v2,d,e,A,w,T_p,x_th,I0,pulse_dur);
[t2,X2] = ode45(f_pacemaker,tspan,X0);
```





VDP DYNAMICS & BLOCK DIAGRAM

Mathematical models of cardiac electrophysiology often rely on nonlinear oscillators to capture the periodic spike-and-recovery structure of real heartbeat signals. Among these, the modified Van der Pol (VDP) oscillator has become a standard because its nonlinear damping term naturally generates the asymmetric depolarization-repolarization waveform observed in atrial and ventricular tissue. In our project, the VDP model functions as the underlying open-loop heart dynamics, and the pacemaker behaves as an external feedback controller that senses missed or irregular beats and injects small corrective pulses.

The key idea is that the heart alone is modeled as a nonlinear second-order ODE, and the pacemaker adds an input forcing term to regulate the oscillator. Once the VDP equation is rewritten as a system of first-order equations, it becomes straightforward to construct a block diagram showing the flow of signals and a state-space representation compatible with MATLAB simulations.

We model the heart as a modified Van der Pol oscillator. The version we use follows the model used in the Yahalom & Puzanov [3] paper. It represents the pacemaker cell's voltage-like activity with a scalar variable $x(t)$:

$$x'' + a(x - v_1)(x - v_2)x' + \frac{x(x+d)(x+e)}{ed} = F(t)$$

The parameters shape different aspects of the cardiac spike:

$$a(x - v_1)(x - v_2)x'$$

produces asymmetric nonlinear damping, giving a sharp upstroke and slower recovery.

$$\frac{x(x+d)(x+e)}{ed}$$

replaces the simple linear stiffness term in the classical oscillator with a biophysically shaped restoring force, affecting contraction timing.

$F(t)$ is an external forcing term, typically modeled in heart-only simulations as:

$$F(t) = Asin(\omega t)$$

representing background physiological simulation.

When parameters are chosen as in the literature (e.g., $a = 0.5$, $v_1 = 0.97$, $v_2 = -1$, $d = 3$, $e = 6$, $A = 2.5$, $\omega = 1.9$), the oscillator spontaneously generates chaotic or fibrillation-like rhythms, making it a suitable model of a dysfunctional heartbeat that needs pacing.

We convert the VDP ODE into state-space form. To create a dynamics block diagram or implement the system in MATLAB, the second-order differential equation is rewritten as two first-order equations:

Let

$$x_1 = x,\ x_2 = \dot{x}$$

Then our state equations are:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\ a(x_1 - v_1)(x_1 - v_2)x_2 - \frac{x_1(x_1+d)(x_1+e)}{ed} + F(t) + u(t)$$

Where $F(t)$ is natural forcing and $u(t)$ is the pacemaker's electrical stimulus, which is zero unless the pacemaker fires.

The state vector becomes:

$$X(t) = \left[x_1\ x_2\right]^T$$

And the output is:

$$y(t) = x_1(t)$$

as the "measureable" electrical signal sensed by the pacemaker lead.

Because the system is strongly nonlinear, the state-space model is nonlinear rather than the linear $AX + Bu$ form. But the format above is all we need for nonlinear MATLAB solutions.

We model the single-lead pacemaker as a feedback controller. A single-lead ventricular pacemaker performs two tasks:

1. Sensing the intrinsic heartbeat signal (voltage from the ventricular lead)
2. Pacing: injecting a short stimulus pulse if a heartbeat fails to occur within the programmed escape interval T_p.

One part of the model becomes beat detection:
If

$$x_1(t) > x_{threshold}$$

and a minimum refractory time has elapsed, the pacemaker detects a natural beat.

Another part of the model becomes escape-interval logic:
If no beat is sensed for

$$t - t_{LastBeat} > T_p$$

the pacemaker fires.

The third part of the model becomes the electrical stimulus:
A pacing pulse is represented as:

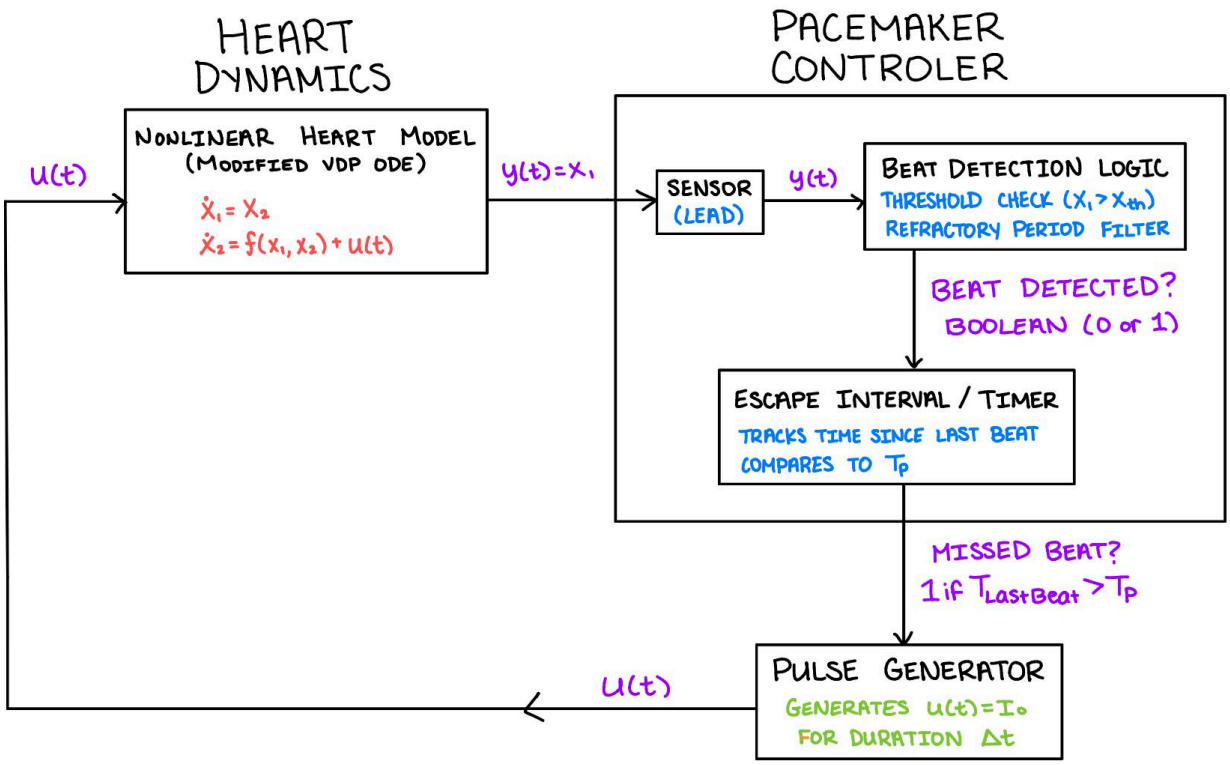
$$u(t) = I_0 \quad \text{during the pulse duration } \Delta t,$$

$$u(t) = 0 \quad \text{otherwise.}$$

This stimulus appears as an additive input in the $\dot{x}_2$ equation, matching how a real pacemaker injects current into cardiac tissue.

Because the pacemaker modifies the heart's dynamics based on measured output, it is explicitly a feedback control system, allowing us to build a complete block diagram.

We construct the block diagram. A block diagram clarifies the feedback loop and the interaction between the nonlinear heart model and the pacemaker controller:



REFERENCES

[1] O. Rodríguez-Abreo, M. Cruz-Fernandez, C. Fuentes-Silva, MA. Quiroz-Juárez, JL. Aragón, “Modeling the Electrical Activity of the Heart via Transfer Functions and Genetic Algorithms,” *Biomimetics (Basel, Switzerland)* vol. 9,5 300. 18 May. 2024. Available: National Library of Medicine, https://www.ncbi.nlm.nih.gov/. [Accessed December 7, 2025]

[2] Protrainings, “Atrial Fibrillation”, *ACLS Wiki*. www.proacls.com/wiki/ekg-rhythms/atrial-fibrillation/.

[3] Yahalom, Asher & Puzanov, Natalia. (2023). “Feedback stabilization applied to heart rhythm dynamics with integro-differential equations method.” *Research Square.* July 4th, 2023 0.21203/rs.3.rs-2853196/v1.