

Elastância

November 25, 2021

```
[1]: using Plots
```

Universidade Federal de Alagoas Instituto de Computação Engenharia de Computação
Disciplina: Modelagem do Sistema Cardiovascular Humano **Professor:** Thiago D. Cordeiro
Alunos: Hugo Tallys Martins Oliveira e Valério Nogueira Rodrigues Júnior

1 Implementação da função Elastância

Implemente a função elastância descrita por Lim *et al.* [1] e compare a curva desta função com aquela descrita por Simaan *et al.* [2]. Sabendo que o parâmetro elastância máxima é utilizado para representar a condição de contratilidade do ventrículo, e que quanto maior o seu valor maior a força de contratilidade do mesmo, utilize dois valores para este parâmetro de modo a representar duas condições para o ventrículo.

1.1 Lim *et al.*

$$e_v(t) = \begin{cases} \sin\left(\frac{\pi(t-t_{sh,i})}{2T_{r,i}}\right)^2 & t_{sh,i} \leq t < (t_{sh,i} + T_{r,i}) \\ \cos\left(\frac{\pi(t-t_{sh,i}-T_{r,i})}{2(T_{sys,i}-T_{r,i})}\right)^2 & t_{sh,i}(t_{sh,i} + T_{r,i}) \leq t < (t_{sh,i} + T_{sys,i}) \\ 0 & 0 \leq t < (t_{sh,i}) \text{ or } (t_{sh,i} + T_{sys,i}) \leq t < T \end{cases}$$

Definindo o modelo:

```
[2]: T = 1.0
k_sys = 0.1
t_sh = 0

function e(t, T_sys0 = 0.54, k_r = 0.83)

    T_sys = T_sys0 - (k_sys / T)
    T_r = k_r * T_sys

    t %= T
    if ((t_sh + T_sys) <= t < T) || (0.0 <= t < t_sh)
        return 0.0
    elseif t_sh <= t < (t_sh + T_r)
        num = * (t - t_sh)
        den = 2T_r
        return sin(num / den)^2
    end
```

```

elseif (t_sh + T_r) <= t < (t_sh + T_sys)
    num = * (t - t_sh - T_r)
    den = 2.0 * (T_sys - T_r)
    return cos(num / den)^2
end
end;

```

Definindo o intervalo de tempo:

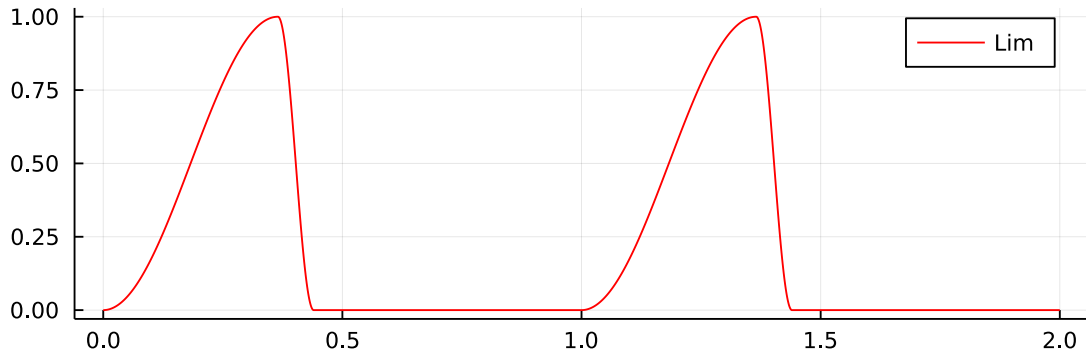
```
[3]: t = 0:0.0001:2T;
```

Plotando a cruva descrita por *Lim. et al.*. Os parâmetros escolhidos são:

- $T_{sys0} = 0.54$ (...)
- $k_r = 0.83$ (...)

```
[4]: lim = map(e , t)
plot(t, lim, label="Lim", size = (600, 200), color="red")
```

[4]:



1.2 Simaan *et al.*

$$E(t) = (E_{max} - E_{min})E_n(t_n) + E_{min}$$

onde:

$$E_n(t) = 1.55 * \left[\frac{\left(\frac{t_n}{0.7}\right)^{1.9}}{1 + \left(\frac{t_n}{0.7}\right)^{1.9}} \right] * \left[\frac{1}{1 + \left(\frac{t_n}{1.17}\right)^{21.9}} \right]$$

Definindo o modelo:

```
[5]: E  = 2.0
     E  = 0.06
     HR = 60
     tc = 60 / HR
     T  = 0.2 + 0.15tc
```

```

function E (t)
    n = (t / 0.7) ^ 1.9
    n = (t / 1.17) ^ 21.9
    return 1.55 * (n / (1 + n)) * (1 / (1 + n))
end;

function E(t)
    return (E - E) * E (t/T) + E
end;

```

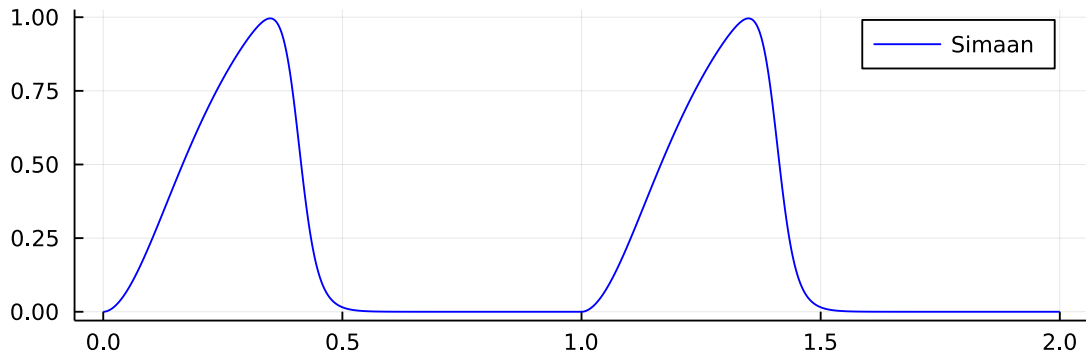
Plotando a curva descrita por *Simaan*:

```

[6]: simaan = map(E, t .% T ./ T)
plot(t, simaan, label="Simaan", size = (600, 200), color="blue")

```

[6]:



1.3 Comparando os dois modelos

Para comparar os dois modelos, vamos tentar aproximar ao máximo a curva descrita por *Lim* daquela descrita por *Simaan*. Para isso, variar os parâmetros T_{sys0}, k_r realizando uma busca completa de modo a minimizar a seguinte função:

$$\min_{(T_{sys0}, k_r)} \{(S - L(T_{sys0}, k_r))^2\}$$

```

[7]: function compute_error(a, b)
    return sum((e.(t, a, b) .- simaan) .^ 2)
end

x = 0.0:0.01:1.0
p = Iterators.product(x, x)

grid_values = Array{Float64}(undef, length(p))

i = 1

```

```

for t in p
    grid_values[i] = compute_error(t[1], t[2])
    i += 1
end

i = 1
min_i = argmin(grid_values)
min_t = nothing
for t in p
    if i == min_i
        min_t = t
    end
    i += 1
end

println(min_t)

```

(0.58, 0.73)

Encontramos um valor mínimo do erro no ponto onde:

$$(T_{sys0}, k_r) = (0.58, 0.73)$$

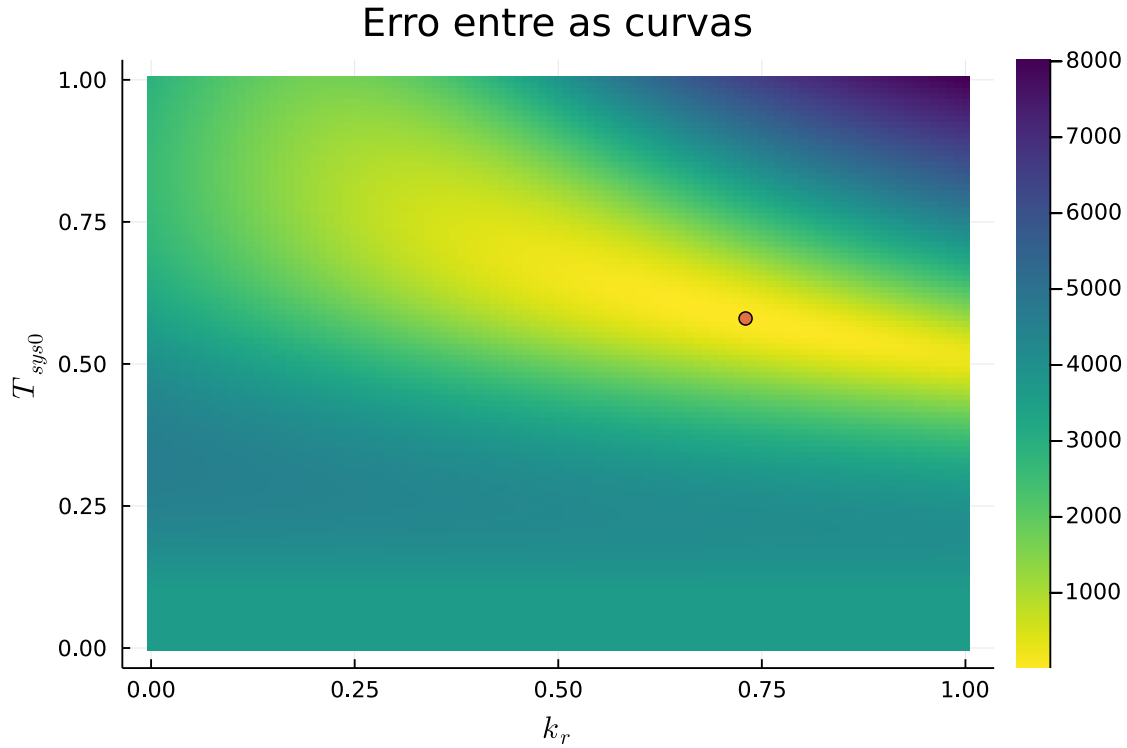
Visualizando o erro em função dos parâmetros:

```

[11]: n = Int64(√length(p))
      heat = reshape(grid_values, (n, n))
      heatmap(x, x, heat, c=cgrad(:viridis, rev=true), title="Erro entre as curvas",
      ↪xlabel="\$k_r\$", ylabel="\$T_{sys0}\$")
      scatter!([min_t[2]], [min_t[1]], legend=false)

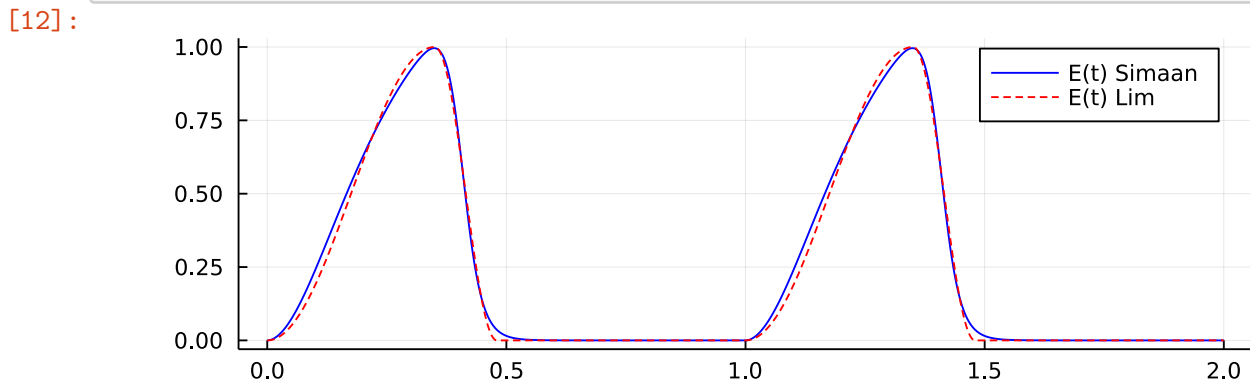
```

[11]:



Resultado final:

```
[12]: plot(t, simaan, label = "E(t) Simaan", color="blue")
      lim = e.(t, min_t[1], min_t[2])
      plot!(t, lim, line = :dash, label = "E(t) Lim", size = (600, 200), color="red")
```



2 Referências

[1] Lim E, Dokos S, Cloherty SL, Salamonsen RF, Mason DG, Reizes JA, Lovell NH. **Parameter-optimized model of cardiovascular-rotary blood pump interactions**. IEEE Transactions

on Biomedical Engineering. 2009 Sep 18;57(2):254-66.

[2] Simaan MA, Ferreira A, Chen S, Antaki JF, Galati DG. **A dynamical state space representation and performance analysis of a feedback-controlled rotary left ventricular assist device.** IEEE Transactions on Control Systems Technology. 2008 Dec 22;17(1):15-28.