

Modelo dinámico del comportamiento eléctrico del corazón

Nieva Cesar Rubén, Navarro Silvia, Juarez Gustavo
Universidad Nacional de Catamarca- Argentina
juarezgustavoadolfo@yahoo.com.ar

El objetivo de este trabajo es brindar conceptos y técnicas empleadas para analizar, modelar y simular la actividad eléctrica del corazón, a través de un estímulo eléctrico mediante la carga y descarga de un condensador, con el fenómeno implícito de almacenamiento de energía eléctrica asistido por un osciloscopio en el procesamiento de datos, permitiéndole afianzar los conceptos físicos en un aprendizaje significativo para facilitar y transferir los conocimientos adquiridos a otros contextos de su quehacer profesional.

Se desarrolló un modelo matemático empleando la Dinámica de Sistemas para demostrar que el comportamiento del marcapaso artificial puede describirse como la carga y descarga de un condensador dependiente de factores dinámicos del sistema.

PROBLEMA MATEMÁTICO DEL MARCAPASO CARDÍACO

El problema con valor inicial (PVI) que describe el funcionamiento de un marcapaso cardíaco consta de un condensador C , una batería de voltaje V y un interruptor que se mueve periódicamente con un período de carga $t_1 < t < t_2$ y un período de descarga $t_2 < t < t_1$ durante el cual el condensador envía un estímulo eléctrico al corazón, el cual actúa como un resistor de resistencia R .

Con el resultado matemático se procede a la creación del Modelo Matemático y a su verificación mediante la simulación dinámica, que corrobore resultados experimentales obtenidos.

SIMULACIÓN AJUSTADA AL MODELO MATEMÁTICO DINÁMICO

Se plantea la creación de un Modelo Matemático Dinámico del proceso de carga y descarga de un capacitor. Para se necesita una variable que mida la carga del capacitor en cada instante, llamando a tal variable: *nivel de carga del capacitor*.

El proceso de carga y descarga se define a través de las variables *carga del capacitor* y *descarga del capacitor*. Si bien cada una de ellas se define por medio de:

- Carga del capacitor: $Q_f (1 - e^{-t/RC})$ donde Q_f es la carga final del proceso, y es donde el nivel de carga toma su máximo valor, t es el tiempo que transcurre para el proceso de cargado, y RC es una medida de la velocidad de carga del capacitor y por ello se llama *constante de tiempo*.
- Cuando RC es pequeña, el capacitor se carga rápidamente; cuando es más grande, el proceso de carga toma más tiempo.
- En forma análoga se define la Descarga del capacitor como $Q_f e^{-t/RC}$.
- Por lo tanto, pretendemos definir al nivel de carga en términos de los dos procesos como: $r Q_f (1 - e^{-t/RC}) + (1-r) Q_f e^{-t/RC}$ donde la carga se produce cuando r es 1 y la descarga cuando r es 0.
- Además, el proceso carga - descarga se da en forma alternada desde la carga nula hasta la máxima Q_f . Queremos que esto se de entre un valor mínimo que llamaremos Q_m , por lo que las definiciones anteriores se ven modificadas en ese valor

$$r[(Q_f - Q_m)(1 - e^{-t/RC}) + Q_m] + (1-r)[(Q_f - Q_m)e^{-t/RC} + Q_m].$$

- Por otro lado, debió reemplazarse esta constante r por una alternativa en la función nivel mediante la aplicación de la función IF THEN ELSE:
 - *Nivel de carga del capacitor: IF THEN ELSE (Time<21, carga del capacitor, descarga del capacitor)*. Suponiendo que durante los primeros 20 segundos carga y luego descarga.
- Además el proceso de carga - descarga continúa, por lo que fue necesario volver a cero el tiempo después de la descarga, y esto se logra con una variable auxiliar:
 - *Tiempo modular: Time-40*INTEGER (Time/40)*. Donde 40 es el tiempo de carga - descarga.
 - De esa manera la descarga luego de la carga se expresa con la variable *demora:Demora: IF THEN ELSE (tiempo modular<20, 0, tiempo modular-20)*

Por tanto, se muestra en la Figura 1 el Diagrama de Forrester de los flujos de carga y descarga sobre un nivel que acumula las cargas.

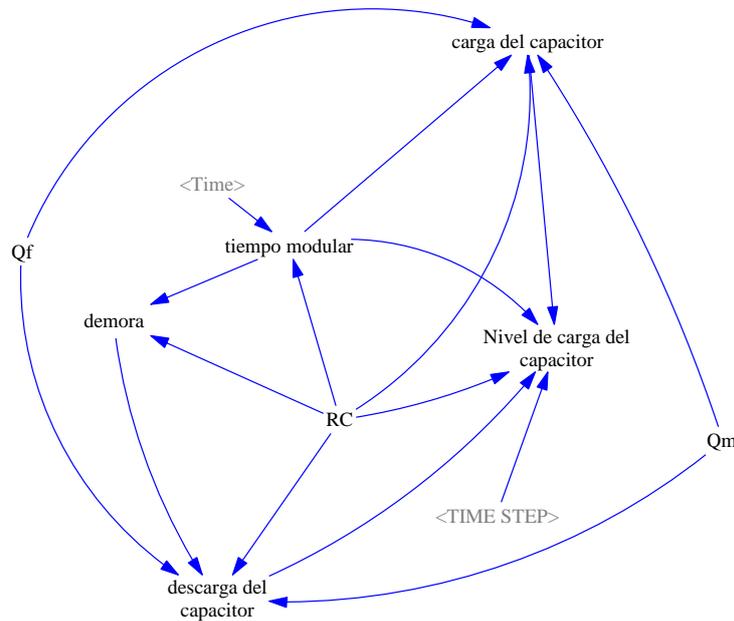


Figura 1: Diagrama de Forrester de la carga y descarga del condensador

Al realizar la corrida del programa se obtiene la carga y descarga del condensador como se muestra en la Figura 2, donde al alimentarse el circuito con una tensión continua de 9V estando el condensador descargado, el LED no conduce y el condensador empieza a cargarse a través de la resistencia de carga; al llegar éste a un cierto valor de tensión de carga, el LED empieza a conducir y el condensador se descarga a través de la resistencia de descarga; cuando la tensión del condensador ha descendido hasta un cierto valor crítico de tensión, el LED deja de conducir y se inicia un nuevo proceso de carga y de descarga.

Por tanto, al cabo de un tiempo igual a RC , la corriente en el circuito ha disminuido a $\frac{1}{e}[0.368]$ de su valor inicial. En este momento la carga del capacitor ha alcanzado una fracción $(1 - \frac{1}{e})[0.632]$ de su valor final.

Así, la *constante de tiempo* $\tau = RC$ es en consecuencia una medida de la velocidad de carga del capacitor, cuando RC es pequeña el capacitor se carga rápidamente y cuando es más grande, el proceso de carga toma más tiempo, tal como se muestra en la Figura 2.

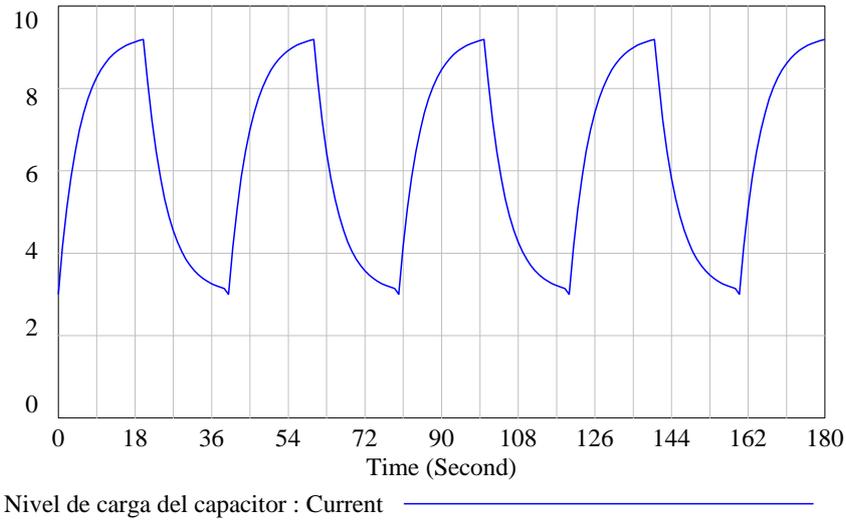


Figura 2: Simulación de la carga y descarga del condensador usando VENSIM 5.2

Debemos ahora ajustar a los valores reales obtenidos experimentalmente, pues en la Figura 2 se toma 180[s] o sea 3 [min], donde se busca lograr los niveles de carga y descarga y el módulo de congruencia del tiempo dependiente de la constante de tiempo, tal se muestra en la Figura 3.

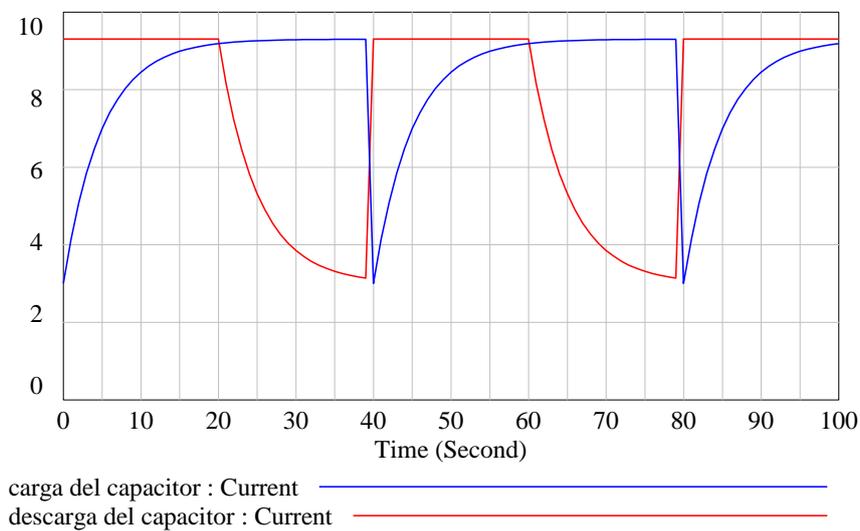


Figura 3: Carga y descarga del condensador que simula el marcapaso cardíaco

Por tanto, se comprueba que el circuito electrónico construido genera una onda en forma de dientes de sierra, donde el tiempo crece conforme la carga y descarga del condensador varíe en un rango entre 3 y 9.3, que se determina por el valor de la constante de tiempo del circuito.

De acuerdo, a las distintas afecciones cardiacas hemos tenido en cuenta al registro de los datos experimentales obtenidos, en relación a los pulsos generados en un periodo de tiempo de 60 [s], se realizó las correspondientes simulaciones dinámicas usando el software Vensim PLE 5.11 y se obtuvo lo siguiente:

- ◆ **Automatismo:** Para una pulsación de 79 latidos por minuto correspondiente al latido cardiaco normal se tiene con $\tau = RC = 0.095$, la generación de las ondas de carga y descarga al inicio del proceso de activación cardiaca, el cual se registra durante 30 segundos (Figura 4).

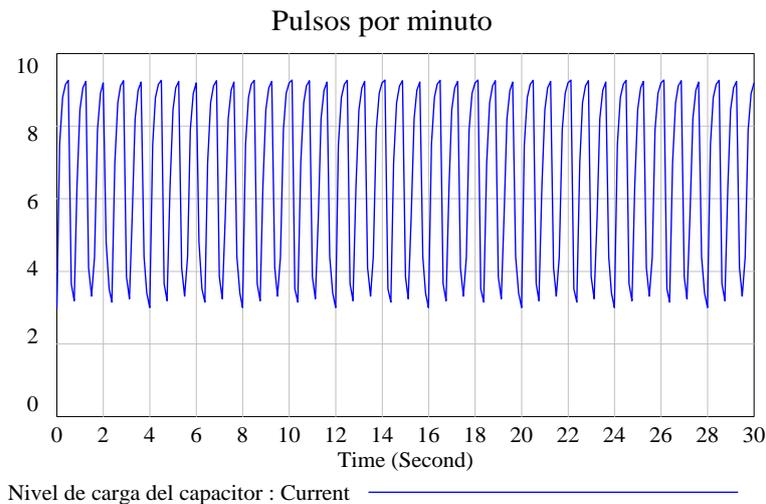


Figura 4: Simulación Dinámica de la pulsación normal que genera el marcapaso cardíaco en el tiempo de 30[s].

- ◆ **Bradicardia sinusal:** Para una pulsación menor a 60 latidos por minuto se tienen con $\tau = RC = 0.123$, la generación de las ondas de carga y descarga al inicio del proceso de activación cardiaca. (Figura 5).

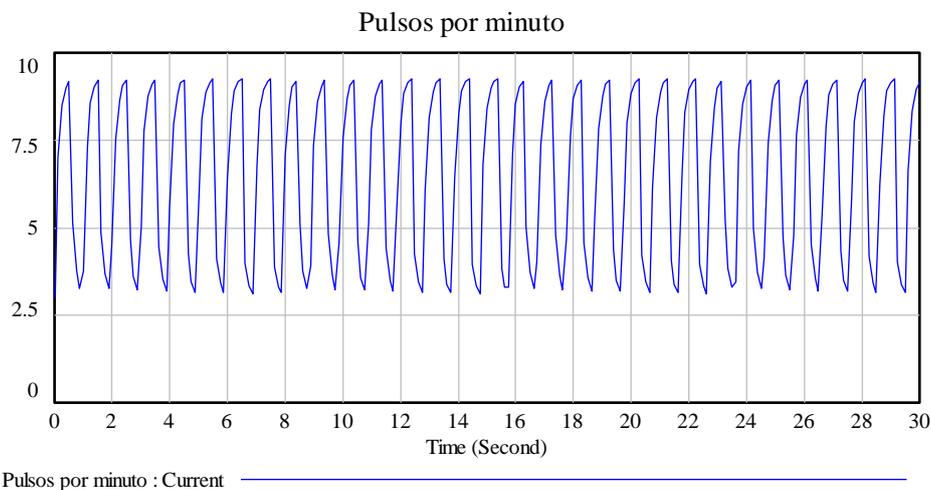


Figura 5: Simulación Dinámica de una bradicardia en el tiempo de 30[s].

- ◆ **Taquicardia:** Para una pulsación de 115 latidos por minuto se tienen con $\tau = RC = 0.065$, la generación de las ondas de carga y descarga (Figura 6).

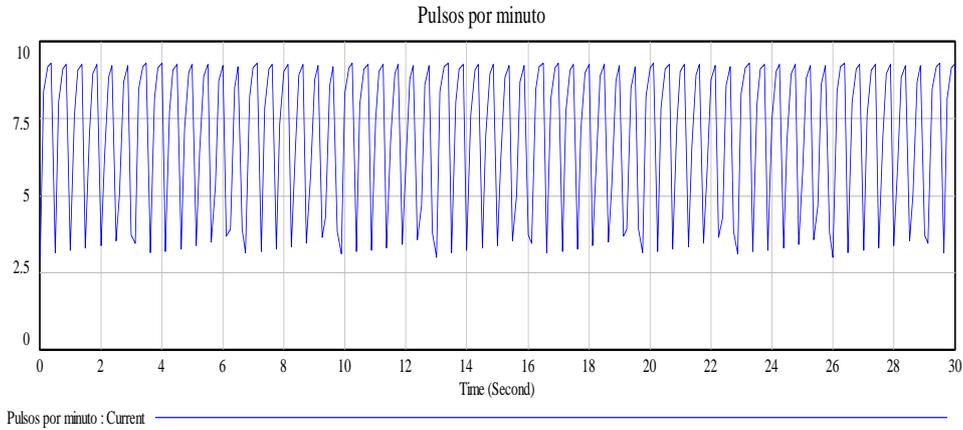


Figura 6: Simulación Dinámica de una taquicardia en el tiempo de 30[s].

- ◆ **Arritmia:** Para una pulsación de 94 latidos por minuto se tienen con $\tau = RC = 0.08$, la generación de las ondas de carga y descarga (Figura 7).

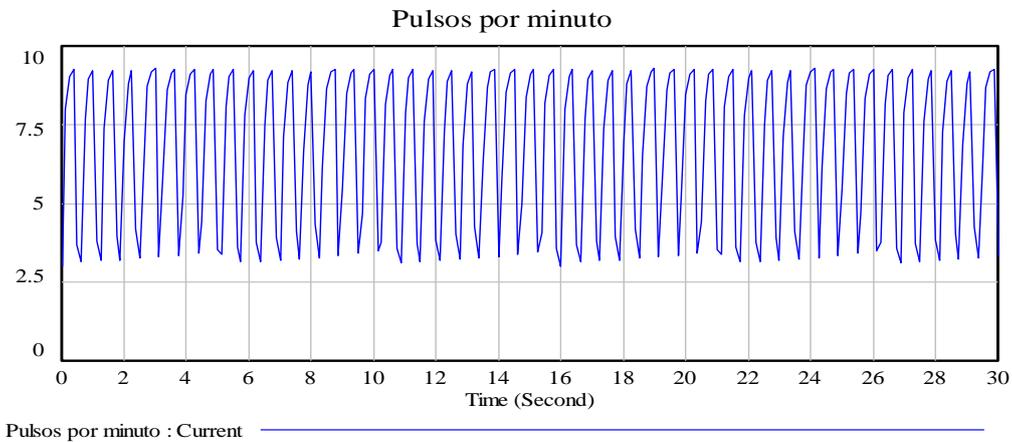


Figura 7: Simulación Dinámica de una arritmia cardíaca en el tiempo de 30[s].

Conclusión

La propuesta de modelizar bajo dinámica de sistema del comportamiento eléctrico del corazón fue llevada a cabo en forma favorable, pues representan a lo observado experimentalmente, y permite describir las anomalías que ocurren en su funcionamiento.



<http://dinamica-de-sistemas.com/>

Distribuidor Oficial Vensim:

<http://atc-innova.com/>

Libros

Cursos Online



[Ejercicios](#)



[Curso Básico Intensivo en Dinámica de Sistemas](#)



[Avanzado](#)



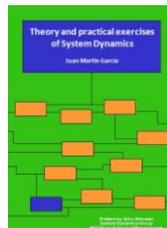
[Curso Superior en creación de modelos de simulación](#)



[Conceptos](#)



[Modelos de simulación en ecología y medioambiente](#)



[English](#)



[Planificación de empresas con modelos de simulación](#)



[Português](#)



[System Thinking aplicado al Project Management](#)