**UNIVERSIDAD DEL VALLE DE GUATEMALA**

Facultad de Ingeniería

A picture containing logo

Description automatically generated

Proyecto No. 1

Simulación De Comportamiento De Un Tubo De Rayo Catódicos (CRT)

Fernando José Garavito Ovando, #18071

Luis Pedro Gonzalez Aldana #20008

Guatemala, miércoles 6 de octubre de 2021

Sumario

Con este proyecto se buscó mediante una simulación programada, graficar figuras de Lissajous las cuales se le aplican un voltaje sinusoidal. Además de manipular en el programa voltaje de aceleración de los electrones, voltaje de placas de deflexión verticales, voltaje de placas de deflexión horizontales, botón de cambio de modo, control de la señal sinusoidal y el tiempo de latencia de un punto en pantalla. Para ello, se realizó un programa en Python el cual es capaz de mostrar la pantalla resultante según se establezca una función sinusoidal o la manipulación de los voltajes de cada una de los pares de placas. Finalmente, se concluyó que la simulación es posible, aunque el lenguaje Python no fue el más apto para establecer el factor de continuidad.

Objetivos

1. Entender a profundidad el funcionamiento de un tubo de rayos catódicos controlado por placas.
2. Elaborar una simulación que muestra el movimiento de electrones dentro de un tubo de rayos catódicos con fines didácticos.
3. Poner a prueba el sistema simulado alimentando las dos señales de entrada con funciones sinusoidales controlables por el usuario y que como resultado se vea en la simulación las Figuras de Lissajous y que se asemejen a lo que predice la teoría al respecto.

Ecuaciones

Ecuación 1: Posición en eje x, según la influencia por el voltaje entre placas verticales y el resto del recorrido hasta la pantalla

Donde…

* q es la carga del rayo
* es el voltaje entre las placas verticales
* es la masa del rayo
* es el tiempo que pasa, atravesando las placas verticales
* es la velocidad al salir de las placas verticales
* es el tiempo que pasa entre pasar por las placas verticales hasta que llega hasta las placas horizontales
* es el tiempo que pasa entre las placas horizontales
* es el tiempo que pasa entre salir de las placas horizontales hasta llegar a la pantalla
* posición x del rayo en la pantalla
* distancia entre placas verticales

Ecuación 2: Posición en eje y, según la influencia por el voltaje entre placas horizontales y el resto del recorrido hasta la pantalla

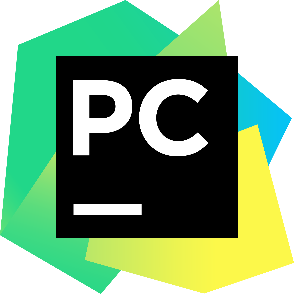
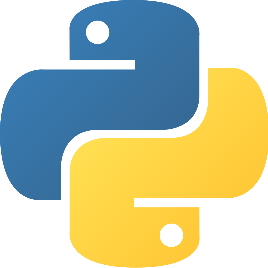
Donde…

* q es la carga del rayo
* es el voltaje entre las placas horizontales
* es la masa del rayo
* es la velocidad al salir de las placas verticales
* es el tiempo que pasa entre las placas horizontales
* es el tiempo que pasa entre salir de las placas horizontales hasta llegar a la pantalla
* posición x del rayo en la pantalla
* distancia entre placas horizontales

Materiales

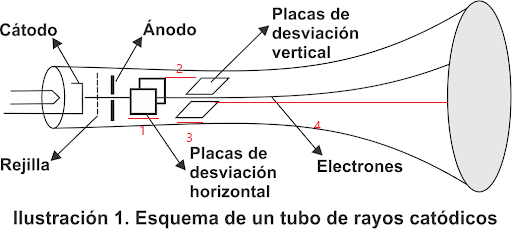
Para este proyecto utilizamos el lenguaje de programación *python* y como entrono de desarrollo *PyCharm.* Cabe mencionar que para Python utilizamos las siguientes librerías:

* Math
* Time
* Numpy
* Matplotlib
* Threading



Métodos

Consideramos el siguiente modelo de CRT.

**

Denotamos de él que podemos dividir como se desvía en 4 fases

1. **Fase 1**: Mientras pasa a través de las placas verticales, se mueve a través de un campo eléctrico uniforme, por lo que su movimiento horizontal es acelerado. Sin embargo al salir de este quedará con una velocidad horizontal que persistirá durante todo el trayecto
2. **Fase 2**: Espacio entre placas verticales y placas horizontales, el rayo no es influenciado pero la velocidad horizontal causada por las placas verticales persiste (y lo hará hasta llegar a la pantalla).
3. **Fase 3**: Mientras pasa a través de las placas horizontales, nuevamente pasa a través de un campo eléctrico, sin embargo, este altera su posición vertical con aceleración constante. Al terminar de pasar por las placas, quedará una velocidad vertical que persistirá por el resto del trayecto. El movimiento en x, persiste a velocidad constante.
4. **Fase 4**: Espacio entre placas horizontales y pantalla, la posición es alterada por las velocidades constantes causadas por las placas verticales y horizontales para las posición en x y y, respectivamente.

Además de ello, notar que no se altera la trayectoria sobre el eje z, por lo que el rayo se moverá a velocidad constante en ese eje y será nuestro punto de referencia para saber cuánto tiempo pasará a través de cada una de las fases.

Cálculos y resultados

# Valores prestablecidos por experimentador

|  |  |  |
| --- | --- | --- |
| Cantidad | Valor | Unidades |
|  |  | m/s |
| Largo de placas (horizontales y verticales) [L] | 0.1 | m |
| Trayecto desde placas verticales hasta horizontales [D] | 0.05 | m |
| Trayecto entre placas horizontales hasta pantalla [] | 0.2 | m |
| Distancia entre placas () | 0.1 | m |

## Entre placas verticales (1ra fase) Mov en X

Con dicha aceleración consideramos las ecuaciones de MRUV y obtenemos

Al finalizar fase 1

.

## Espacio entre placas verticales y placas horizontales (2da fase)

## Entre placas horizontales (3ra fase) movimiento y, movimiento x

## De placas horizontales a pantalla

## Movimiento Total

Eje X:

Eje Y

## Voltaje Máximo y mínimo

Despejando, utilizando las medidas y el movimiento entre placas descrito:

## Posiciones máximas (calcular tamaño de pantalla)

Utilizamos el voltaje máximo y mínimo para cada uno de los ejes y obtenemos

# Limitantes obtenidas

|  |  |
| --- | --- |
| Dato | Cantidad |
| Voltaje máximo |  |
| Voltaje mínimo |  |
| Posicion x máxima alcanzar |  |
| Posicion x mínima alcanzar |  |
| Posicion y máxima alcanzar |  |
| Posicion y mínima alcanzar |  |

Conclusiones y recomendaciones

* Los objetivos se cumplieron al ejecutar el programa, y la simulación correspondió con el modelo.
* Considerar el factor continuidad, no implementado para simulaciones en entornos de programación mejor adaptados como unity
* El efecto de placas paralelas resulta sencillo si consideramos todos los trayectos y las variables que actúan sobre ellos en cada uno de ellos
* Es recomendable dividir el movimiento en dos ejes
* Considerar otro entorno de programación mejor adaptado

Video

<https://youtu.be/9owtEpo4XAQ>

Anexos

Repositorio GITHUB: <https://github.com/LPELCRACK896/Simulacion_TRC.git>

Cálculos a mano:

