



Universidad
Zaragoza

Trabajo Fin de Grado

Diseño de un circuito de adquisición de datos para
matrices de sensores resistivos

Design of a Data Acquisition Circuit for Resistive
Sensor Arrays

Autor

Alexia Suárez Pérez

Director

Carlos Medrano Sánchez

Escuela Universitaria Politécnica de Teruel
2024



Diseño de un circuito de adquisición de datos para matrices de sensores resistivos

Resumen

En este Trabajo Fin de Grado se ha llevado a cabo el diseño de dos modelos distintos de adquisición de datos para matrices de sensores resistivos. Tras la fase de diseño, el proyecto avanzó con la implementación de un script en Python que permite la creación automática de los archivos netlist necesarios. Esta automatización permite que las matrices de sensores resistivos puedan ser simuladas en LTSpice, independientemente de las dimensiones de la matriz, haciendo que el proceso de simulación sea eficiente y adaptable a una amplia variedad de aplicaciones.

El diseño se basó en el circuito de adquisición Zero Potential Method (ZPM), que sirvió como punto de partida para desarrollar los modelos. El circuito ZPM fue analizado y modificado, para adaptarse a los requisitos específicos en este trabajo (alimentación del circuito y valores de las resistencias). Este proceso involucró un examen detallado de las características de rendimiento del circuito, así como un proceso iterativo de pruebas y refinamientos para asegurar que los diseños finales cumplieran con los criterios necesarios de precisión y fiabilidad, así como con los requisitos principales.

Design of a Data Acquisition Circuit for Resistive Sensor Arrays

Abstract

In this Final Degree Project, two distinct data acquisition models for resistive sensor arrays have been designed. Following the design phase, the project progressed with the implementation of a Python script that enables the automatic creation of the necessary netlist files. This automation allows the resistive sensor arrays to be simulated in LTSpice, regardless of the matrix dimensions, making the simulation process efficient and adaptable to a wide range of applications.

The design was based on the Zero Potential Method Circuit (ZPM) acquisition circuit, which served as the starting point for developing the models. The ZPM circuit was analyzed and modified to meet the specific requirements of this work (power supply and resistance values). This process involved a detailed examination of the circuit's performance characteristics, as well as an iterative process of testing and refinement to ensure that the final designs met the necessary criteria for accuracy, reliability, and the main requirements.





ÍNDICE

Listado ilustraciones	viii
Listado tablas	x
Listado siglas y abreviaturas	xi
1 Introducción	1
1.1 Objetivos	1
1.2 Motivación y relevancia del tema	1
2 Fundamentos Teóricos	2
2.1 Introducción a los circuitos de adquisición de datos	2
2.2 Sensores resistivos: principios y aplicaciones	2
2.2.1 Principios de funcionamiento	2
2.2.2 Aplicaciones	2
2.3 Teoría del circuito ZPM	3
2.3.1 Definición y funcionamiento del circuito ZPM.....	3
3 Análisis de las herramientas software	4
3.1 Utilización de LTSpice.....	4
3.1.1 Comandos dot.....	4
3.1.2 LTSpice modo esquemático.....	4
3.1.3 LTSpice modo texto	4
3.2 Python	5
3.2.1 Características	5
3.2.2 Aplicaciones	5



3.3 Google Colab	5
4 Diseño del Circuito de Adquisición de Datos	6
4.1 Especificaciones y requisitos del diseño	6
4.2 Adaptación del circuito	6
5 Implementación del modelo 1	8
5.1 Implementación con componentes ideales	9
5.1.1 Cálculos teóricos	10
5.1.2 Comprobación del circuito ideal mediante LTSpice	15
5.2 Selección de componentes	16
5.2.1 Elección del AO	16
5.2.2 Elección de los switches	16
5.2.3 Elección del dispositivo de obtención del voltaje V'	17
5.3 Implementación del diseño en LTSpice	20
5.3.1 Simulación del circuito matriz 2x2 en LTSpice	21
5.3.1.1 Simulación de los switches y el AO reales	22
5.3.1.2 Simulación de la referencia de tensión real	25
5.3.2 Configuración de la simulación completa	26
6 Implementación del modelo 2	28
6.1 Selección de componentes	29
6.2 Implementación del diseño en LTSpice	30
6.2.1 Simulación del circuito matriz 2x2 en LTSpice	30
6.2.1.1 Simulación del referencia de tensión real	30



6.2.2 Configuración de la simulación completa	34
7 Automatización del Proceso de Simulación con Python	37
7.1 Generación automática de archivos de simulación LTSpice	37
7.1.1 Librerías y herramientas utilizadas	37
7.1.2 Estructura del script en Python	37
7.2 Integración de la simulación con Python y LTSpice	39
7.2.1 Flujo de trabajo automatizado	39
7.2.2 Ejecución, validación y resultados	41
8 Resultados y Discusión	44
8.1 Calculo de errores cometidos con la implementación del diseño	44
8.2 Comparación de ambos modelos.....	45
8.3 Análisis de los valores obtenidos en mediciones reales	45
9 Conclusiones.....	48
9.1 Impacto del trabajo realizado.....	48
9.2 Sugerencias para futuros trabajos	49
10 Referencias	50
Anexos	52
Anexo I: Cálculos	a
Apartado I: Cálculo del valor de la tensión de salida al sustituir las masas por un voltaje denominado V' modelo 1.....	b
Apartado 2: Calculo del valor de V' y rf mediante la aplicación de las condiciones	c
Apartado 3: Calculo del valor de Vo para la resistencia de 400Ω.....	d



Apartado 4: Comprobación del uso de un divisor de tensión para la obtención de 3V en el modelo 1, seleccionando la resistencia de 400Ω.....	f
Apartado 5: Cálculo del valor de la tensión de salida al sustituir las masas por un voltaje denominado V' modelo 2.....	g
Anexo II: Análisis y Soluciones derivados de Problemas en Amplificadores Operacionales	i
Anexo III: Cálculo del error en el modelo 1 y en el modelo 2.	p
Anexo IV: Código	t
Apartado 1: Modelo 1	u
Apartado 2: Modelo 2	y
Anexo V: Implementación modelo 2	dd
Anexo VI: Cálculo del error en matrices reales.....	jj



Listado ilustraciones

Ilustración 1: Circuito ZPM.....	3
Ilustración 2: Matriz de resistencias 2x2.....	10
Ilustración 3: Circuito de selección, resistencia de 100 Ω , modelo 1, matriz 2x2.....	10
Ilustración 4: Circuito de selección, resistencia de 400 Ω , modelo 1, matriz 2x2.....	12
Ilustración 5: Colocación de un divisor de tensión en el circuito de selección de la resistencia de 100 Ω	13
Ilustración 6: Valor de V_o , seleccionando la resistencia de 100 Ω	13
Ilustración 7: Valor de la tensión obtenida mediante el divisor de tensión	14
Ilustración 8: Valor de la V_o , seleccionando la resistencia de 100 Ω , modelo 1 ideal.....	15
Ilustración 9: Valor de la V_o , seleccionando la resistencia de 400 Ω , modelo 1 ideal.....	15
Ilustración 10: Curva característica tensión - corriente, diodo Zener.....	18
Ilustración 11: Implementación modelo 1	20
Ilustración 12: Explicación lógica seguida en las ondas de los switches.....	21
Ilustración 13: Implementación modelo 1, switches y AO reales	22
Ilustración 14: V_o en el modelo 1, switches y AO reales	22
Ilustración 15: Voltaje de la patilla (-) del AO.....	23
Ilustración 16: Modelo 1, AO ideal	24
Ilustración 17: Modelo 1, AO ideal, valor de V_o	24
Ilustración 18: Modelo 1, AO ideal, valor de voltaje de la patilla (-).....	24
Ilustración 19: Modelo 1, V_o , voltaje V' ideal.....	25
Ilustración 20: Conexiones del LT6658-3.....	25



Ilustración 21: Voltaje de salida del LT6658-3	26
Ilustración 22: Circuito final modelo 1	26
Ilustración 23: Circuito final modelo 1, Vo	27
Ilustración 24: Circuito modelo 2	29
Ilustración 25: Circuito modelo 2, referencia de tensión de 3V	31
Ilustración 26: Circuito modelo 2, referencia de tensión de 3V, Vo.	31
Ilustración 27: Circuito modelo 2, referencia de tensión de 2.5V	32
Ilustración 28: Circuito modelo 2, referencia de tensión de 2.5V, Vo.....	32
Ilustración 29: Circuito modelo 2, referencia de tensión de 1.8V	33
Ilustración 30: Circuito modelo 2, referencia de tensión de 1.8V, Vo.....	33
Ilustración 31: Circuito final modelo 2.....	34
Ilustración 32: Circuito final modelo 2, Vo	35
Ilustración 33: Circuito final modelo 2, NPN	35
Ilustración 34: Circuito final modelo 2, NPN, Vo	36
Ilustración 36: Vo, netlist, modelo 1.....	42
Ilustración 37: Vo, resistencia de 100Ω, netlist, modelo 2.....	43
Ilustración 38: Vo, netlist, modelo 2.....	43
Ilustración 39: Representación conductancia matriz ralex2	46
Ilustración 40: Representación conductancia matriz ralex2	47



Listado tablas

Tabla 1: Cálculo de los valores máx. y mín. de V_o para distintas V' y r_f	9
Tabla 2: Valor de V_o para cada una de las resistencias de la matriz 2×2	21
Tabla 3: Voltaje de salida de cada switch en las distintas fases	23
Tabla 4: Voltaje de salida para cada resistencia y cada referencia de tensión	30
Tabla 5: Voltaje de salida para cada resistencia y referencia de tensión de 1.8V	34
Tabla 6: Error para cada resistencia, modelo 1.....	44
Tabla 7: Error para cada resistencia, modelo 2.....	44
Tabla 8: Error para resistencias relevantes de r_{alex1} y r_{alex2}, para ambos modelos.....	46



Listado siglas y abreviaturas

ADC: Conversor analógico digital

AO: Amplificador Operacional

I: Corriente

IoT: Internet of Things

NANC: Contacto Normalmente Abierto Normalmente Cerrado

SPDT: Single Pole Double Through

V: Voltaje

Vo: Voltaje de salida

ZPM: Zero Potential Method



1 Introducción

1.1 Objetivos

El presente trabajo aborda el diseño de un circuito de adquisición de datos para matrices de sensores resistivos. Este tipo de configuración de sensores aparece en la medida de la distribución de presión, temperatura u otras magnitudes relevantes.

Por lo que para este proyecto han sido necesario cumplir con las siguientes metas:

- Análisis y conocimiento del funcionamiento de los circuitos Zero Potential Method (ZPM).
- Selección de los componentes reales requeridos, así como su utilización en un circuito de adquisición de datos comprendiendo sus características más relevantes.
- Comprobación de los diferentes modelos obtenidos, mediante la simulación de LTSpice.
- Utilización del lenguaje Python, automatizando el proceso de creación de netlist. De esta manera podemos crear automáticamente circuitos para matrices con grandes dimensiones para su simulación.
- Presentación de los resultados de forma compacta y comprensible.

1.2 Motivación y relevancia del tema

El diseño de un circuito de adquisición de datos para matrices de sensores resistivos es crucial en una amplia variedad de campos, especialmente en el biomédico, en el cual está centrado el diseño, donde la exactitud y confiabilidad en la captura de señales fisiológicas son esenciales para el diagnóstico, monitoreo y tratamiento de pacientes. Siendo la capacidad de procesar información de múltiples sensores, como los que miden la actividad cardíaca, respiratoria o la presión, una de las características principales, mejorando constantemente la precisión diagnóstica y personalizada de las intervenciones médicas.

En concreto este trabajo está promovido por la necesidad de diseñar un circuito de adquisición de datos mejorado para una matriz de sensores de presión piezoresistivos diseñada en el grupo de investigación EduQTech de la EUPT. La matriz está configurada por 16x16 celdas distribuidas en una superficie de 32x32 cm², sobre una base flexible con forma de esterilla. Esta esterilla permite medir la presión en una superficie, con aplicaciones en ergonomía (presión en asientos, camas), medidas de estabilidad (presión ejercida por los pies) entre otras.



2 Fundamentos Teóricos

2.1 Introducción a los circuitos de adquisición de datos

Los circuitos de adquisición de datos son componentes esenciales en sistemas de medición y control, cuyo objetivo principal es llevar a cabo la conversión de señales físicas en datos digitales que puedan ser procesados, por lo que permiten todos y cada uno de los pasos que ello conlleva (recolección, procesamiento y análisis de datos provenientes de diversas fuentes, tal y como relata Emilio (2013), como por ejemplo puede tratarse de sensores resistivos, los cuales se detallarán en el siguiente punto.

Suelen incluir varias etapas dependiendo del diseño: amplificación de señales (aumentando dicha señal hasta un nivel adecuado para la siguiente etapa), Conversor analógico digital (ADC), las señales son digitalizadas para poder manipular y analizar los datos. El proyecto se centrará principalmente en el diseño de la primera etapa, obteniendo una salida analógica.

2.2 Sensores resistivos: principios y aplicaciones

2.2.1 Principios de funcionamiento

Los sensores resistivos son dispositivos, cuya función es llevar a cabo la medición de variables físicas (temperatura, presión, deformación...), mediante cambios en la resistencia eléctrica. Las resistencias de un material resistivo varían dependiendo de la magnitud de la variable a medir según Tartagni (2021).

En el caso del dispositivo desarrollado en el grupo EduQTech, se utiliza un material piezoresistivo, Velostat, como base para el sensor. Su resistencia cambia con la presión aplicada. La relación entre la presión y la resistencia es compleja. Sólo como primera aproximación puede considerarse que el inverso de la resistencia es proporcional a la presión tal y como afirma Martínez Cesteros (2023):

$$\frac{1}{R} = \alpha P$$

No obstante, en este TFG se considera que la magnitud a medir es la resistencia, independientemente de la magnitud física que produzca sus variaciones.

2.2.2 Aplicaciones

Las aplicaciones derivadas de este tipo de sensores abarcan todos los ámbitos de la industria, ya que pueden ser implementadas tanto en industria automotriz, para medir temperaturas del motor, presión de los fluidos y presión de neumáticos, como en la industria médica, en equipos de monitoreo de pacientes, presión en asientos o camas, o en la industria de la construcción para medir deformaciones.

2.3 Teoría del circuito ZPM

2.3.1 Definición y funcionamiento del circuito ZPM

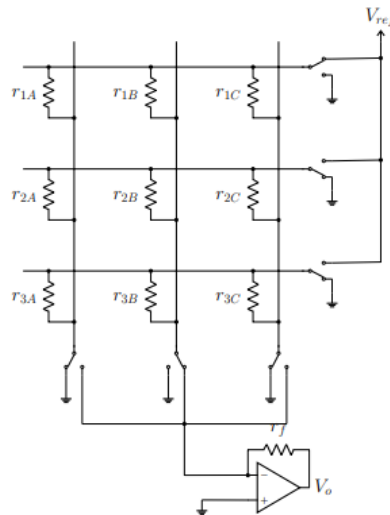


Ilustración 1: Circuito ZPM. Martínez Cesteros, J. (2023). Development of a low cost pressure sensitive mat: Application to stability and posture measurement (Tesis doctoral). Universidad de Zaragoza.

La ilustración 1 muestra una matriz 3x3 de resistencias, con tres filas (1, 2, 3) y 3 columnas (A, B, C). Las conexiones de fila y columna son compartidas por las resistencias de la matriz. Para matrices de ciertas dimensiones es inviable tener conexiones individuales para cada resistencia.

El circuito de adquisición corresponde al ZPM.

Su funcionamiento está basado en la conmutación entre estados activo e inactivo tal y como aparece representado en la ilustración 1, en muchas de las ocasiones se realiza la adquisición de datos cuando el circuito está activo, y cuando se encuentra inactivo se desconectan componentes o se reducen las operaciones, en nuestro caso esa selección se utiliza para llevar a cabo la elección de la celda correspondiente a la resistencia que se desea medir en dicho instante.

Por ejemplo, en la ilustración 1 se selecciona la fila 1, conectándola mediante un interruptor a un voltaje V_{ref} . Las otras filas están conectadas a masa. La columna seleccionada, B, se conecta a la entrada inversora del amplificador operacional (AO), mientras que las otras dos columnas se conectan a masa. De esta forma se selecciona la resistencia r_{1B} . En la segunda y tercera fila, las resistencias están conectadas a masa en ambos terminales (directamente o a la masa virtual de la entrada inversora del AO) y no fluye corriente (I) por ellas. En la primera fila, las resistencias están conectadas entre V_{ref} y masa (real o virtual), pero la única corriente que fluye hacia el AO es la que pasa por r_{1B} . Por ello la salida sólo depende de la resistencia seleccionada.

En la discusión precedente se ha considerado que los componentes son ideales. Si los interruptores, amplificadores y referencias de voltaje no son ideales aparecerá un error en la salida como veremos en las simulaciones presentadas en este trabajo.



3 Análisis de las herramientas software

3.1 Utilización de LTSpice

PSpice es un miembro del grupo de simuladores de SPICE en el cual está incluido LTSpice (perteneciente a Linear Technology Corporate), explica Rashid (2012). LTSpice es un software gratuito que permite realizar simulaciones de circuitos, obteniendo corriente, voltaje y potencia, así como análisis transitorios y de frecuencia. También es útil para la verificación y optimización de diseños, permitiendo ajustar parámetros y modelos de componentes para cumplir con especificaciones de diseño, por lo que se utilizará a lo largo del proyecto para llevar a cabo la simulación y comprobación de cada uno de los pasos, partiendo de componentes ideales hasta obtener el diseño final.

3.1.1 Comandos dot

Comandos dot son comandos específicos utilizado para definir y controlar diversos aspectos de una simulación o configuración del circuito. Estos comandos se introducen en la línea de comandos del simulador o en el archivo de netlist, ajustando parámetros...

Los utilizados en el proyecto son tal y como describe Rashid (2012):

- `.backanno`: activa la anotación de la simulación, permitiendo que LTSpice recupere y muestre la información de simulación directamente en el diseño esquemático después de que se ha ejecutado.
- `.end`: indica el final del archivo de netlist.
- `.lib`: utilizado para incluir un archivo de biblioteca de modelos en el archivo de netlist.
- `.sub`: define un subcircuito en el archivo netlist.
- `.tran`: especifica un análisis transitorio.
- `.gmin`: establece una conductancia mínima en paralelo con todas las uniones en el circuito.
- `.uic`: se usa junto con el comando `.tran` para omitir el cálculo del punto de operación inicial antes de comenzar una simulación transitoria.

3.1.2 LTSpice modo esquemático

LTSpice en modo esquemático se trata de una interfaz gráfica donde el usuario puede ver en todo momento la distribución del circuito, así como los componentes la forman, por lo que la simulación y el diseño es visual en todo momento. Permite añadir, quitar o modificar componentes del circuito en todo momento. Se utilizará el modo esquemático durante las primeras fases de diseño, hasta llegar a la comprobación del circuito final.

3.1.3 LTSpice modo texto

El modo texto de mano de las netlist, se refiere a la entrada de comandos y configuraciones compuestas por un conjunto de instrucciones en texto que describe los componentes y las conexiones de un circuito. Presenta un control más detallado y flexible de nuestro diseño, así



como la creación de componentes, permitiendo a su vez la simulación con la ayuda de los dos. La utilización de este modo de uso de LTSpice se llevará a cabo en la fase final del proyecto donde se programará en Python la creación automática de netlist, de manera que su simulación en LTSpice sea instantánea.

3.2 Python

3.2.1 Características

Python es un lenguaje de programación de alto nivel, creado por Guido van Rossum y lanzado por primera vez en 1991. Diseñado con una sintaxis clara y sencilla, centrado principalmente en la legibilidad del código, lo que facilita su uso tanto por principiantes como por programadores experimentados. Es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea, permitiendo una rápida depuración y un desarrollo ágil.

Python es conocido por su versatilidad, ya que admite múltiples paradigmas de programación, incluyendo la programación orientada a objetos, funcional e imperativa. Además, cuenta con una amplia biblioteca estándar que proporciona herramientas para prácticamente cualquier tarea, desde la manipulación de texto hasta el desarrollo web, lo que lo convierte en una herramienta poderosa y flexible para una variedad de aplicaciones, tal y como explica Chazallet (2018, p.17).

3.2.2 Aplicaciones

Python presenta una gran potencial y flexibilidad por lo que puede ser utilizado en numerosas aplicaciones, entre las cuales cabe destacar:

- Desarrollo web
- Ciencia de datos y aprendizaje automático
- Automatización de tareas
- Desarrollo de Software y juegos
- Redes y seguridad
- Inteligencia artificial
- Desarrollo científico y numérico

En el proyecto utilizaremos Python para la automatización de tareas mediante la creación de una función que permite crear netlist para LTSpice.

3.3 Google Colab

Google Colab se trata de una plataforma gratuita, basada en la nube, permitiendo a los usuarios escribir y ejecutar código en Python3 directamente en el navegador utilizando Jupyter Notebooks. Por lo que dispondremos de esta herramienta para la programación y ejecución del código en Python del proyecto.



4 Diseño del Circuito de Adquisición de Datos

4.1 Especificaciones y requisitos del diseño

Con la finalidad de alcanzar los objetivos propuestos con anterioridad, el diseño deberá contar con las siguientes especificaciones.

La alimentación deberá de ser única cuyo valor será de +5V, adaptando así nuestro dispositivo a la conexión USB de manera que posea una mayor accesibilidad, comodidad y conectividad en situaciones y posibles aplicaciones para todo tipo de dispositivos, así como para todo tipo de públicos. A su vez la resistencia del sensor será 100 Ohmios y la salida V_o deberá encontrarse siempre en rangos positivos. El circuito se probará también con ejemplos de redes 16x16 proporcionadas por el director y que corresponden a la esterilla sensible a la presión desarrollada en EduQTech.

4.2 Adaptación del circuito

Partiendo del circuito ZPM, se analizará la situación en la que se selecciona una de las resistencias del circuito, por ejemplo, la resistencia r_{1B} . Es importante destacar que, a efectos de la ecuación general, la elección de la resistencia es indiferente. El objetivo de este análisis es determinar la ecuación de salida del circuito.

En primer lugar, es necesario calcular las corrientes I_1 e I_2 . La corriente I_1 es la que pasa por la resistencia r_{1B} , mientras que la corriente I_2 es la que pasa por la resistencia r_f . En este análisis, consideramos que el amplificador operacional (AO) es ideal. Esto significa que la tensión en ambas entradas del AO (marcadas como + y -) es igual a 0 V.

Esta condición nos permite simplificar el análisis y obtener el valor de la tensión de salida V_o . Sabiendo que los valores de r_f y r_{1B} son variables y que el valor de la tensión de referencia V_{ref} es conocido, se puede derivar la siguiente ecuación:

$$I_1 = I_2 \rightarrow \frac{V_{ref} - V(-)}{r_{1B}} = \frac{V(-) - V_o}{r_f}$$

Dado que $V(-)$ es 0 V en un AO ideal, la ecuación se simplifica a:

$$\frac{V_{ref}}{r_{1B}} = \frac{-V_o}{r_f}$$

De esta relación, podemos despejar V_o :

$$V_o = \frac{-V_{ref} * r_f}{r_{1B}}$$



Esta ecuación nos indica que la tensión de salida V_o es proporcional a la tensión de referencia V_{ref} , invertida y escalada por la razón de las resistencias r_f y r_{1B} . Corresponde a la salida de un AO inversor, ya que por las características del circuito ZPM el resto de resistencias no influyen, al menos con componentes ideales.

Se considerará que V_{ref} se toma directamente de la alimentación del sistema fijada en 5V. Aunque no es la mejor opción debido a que la alimentación puede ser algo ruidosa, en este caso es admisible porque los sensores tienen otras fuentes de error bastante grandes. Además, utilizar la alimentación directa permite exigir más corriente, lo cual puede ser una ventaja en determinadas aplicaciones. Por lo que la ecuación específica es:

$$V_o = \frac{-5 * r_f}{r_{1B}}$$

En conclusión, el análisis de la selección de la resistencia r_{1B} en el circuito ZPM nos permite determinar que la ecuación de salida V_o depende linealmente de V_{ref} y de los valores de r_f y r_{1B} . Este resultado es fundamental para entender el comportamiento del circuito y su respuesta ante variaciones en los componentes. En cambio, la salida que obtenemos es negativa por lo que se deben de realizar los cambios pertinentes para poder cumplir con las especificaciones. Estos cambios no pueden ir relacionados a la colocación de otra etapa inversora ya que no es posible tratar con voltajes negativos. Se proponen dos soluciones distintas que denominados modelo 1 y modelo 2.

Modelo1: sustitución de las masas del circuito por otro voltaje que se denominará V' .

Aplicando la sustitución y llevando a cabo los cálculos de nuevo se obtiene la siguiente ecuación para la salida de V_o :

$$V_o = r_f * \left(\frac{V' - V_{ref}}{r_{1B}} \right) + V'$$

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 1: Calculo del valor de la tensión de salida al sustituir las masas por un voltaje denominado V' modelo 1). Esta salida corresponde a la suma de un AO inversor con V_{ref} y un AO no inversor con V' .

Pudiendo llegar a la siguiente afirmación:

$$r_f * \left(\frac{V_{ref} - V'}{r_{1B}} \right) < V' \rightarrow V_o > 0$$

Cumpliendo con uno de los requerimientos del diseño.

Modelo 2: en el modelo 2 la entrada no inversora del AO se coloca a un voltaje V' , pero en este caso la fila seleccionada se conecta a masa y las no seleccionadas a dicho voltaje, V' . La corriente por el AO fluye en sentido contrario al anterior y se obtiene la salida que corresponde a un amplificador operacional no inversor que amplifica V' :

$$V_o = V' * \left(1 + \frac{r_f}{r_{1B}} \right)$$



5 Implementación del modelo 1

Para implementar el modelo 1, es necesario calcular los valores de V' y r_f , dado que los valores de V_{ref} y R_{1B} están determinados por las condiciones del circuito. Es fundamental asegurarse de que el voltaje de salida (V_o) del amplificador operacional (AO) se mantenga dentro del rango de 0 a 5V.

A continuación, se procederá a aplicar las condiciones necesarias para determinar los valores de V' y r_f .

Condición 1:

$$r_f * \left(\frac{V' - V_{ref}}{r_{1B}} \right) < V'$$

Al aplicar la condición 1, se consigue una salida positiva ($V_o > 0$).

Condición 2:

$$0 < V_o < 5 \rightarrow 0 < r_f * \left(\frac{V' - V_{ref}}{r_{1B}} \right) + V' < 5$$

Al aplicar la condición 2, se consigue que la salida del AO se encuentre entre 0 y 5 V. Esta condición también consigue que se cumpla la condición 1.

Condición 3:

$$r_{1B} > 100\Omega$$

Condición 4:

$$V_{ref} = 5V$$

Al aplicar las condiciones 3 y 4 se cumple con los requerimientos del sistema.

Mediante el desarrollo de las 4 condiciones se alcanza la siguiente inecuación:

$$\frac{V'}{5 - V'} > \frac{r_f}{r_{1B}}$$

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 2: Calculo del valor de V' y r_f mediante la aplicación de las condiciones)

El límite se encontrará cuando:



$$\frac{rf}{r1B} = \frac{rf}{100}$$

Pudiendo afirmar que V' depende del valor de rf y viceversa, para poder fijar uno de ellos se va a analizar la situación en diferentes escenarios en los que se diferencia el valor típico de V' (valores típicos de referencias de voltaje):

V' (V)	Límite $rf/r1B$	rf (Ω)	$rf/100$	$V_{\text{mín}} (r1B = 100)$	$V_{\text{máx}} (r1B = \infty)$
3	< 1.5	120	1.2	0.6	3
3	< 1.5	100	1	1.0	3
3	< 1.5	50	0.5	2.0	3
2.5	< 1	100	1	0.0	2.5
2.5	< 1	50	0.5	1.25	2.5
1.8	< 0.56	50	0.5	0.2	1.8

Tabla 1: Cálculo de los valores máx. y mín. de V_o para distintas V' y rf .

Es fundamental que la corriente en el circuito sea baja. Para lograr esto, V' debe estar lo más próximo posible a V_{ref} . Esto significa que el valor de V' debe ajustarse de manera que sea muy cercano al valor de V_{ref} , lo cual permitirá que la corriente alcance el nivel deseado. Los AO no suelen mantener bien su salida con corrientes elevadas.

Atendiendo a la justificación anterior, seleccionaremos el valor de $V' = 3V$. Es crucial que el rango de salida sea amplio, permitiendo un margen adecuado respecto a los extremos de 0V y V_{ref} que es 5V. Para asegurar un rango de salida adecuado, determinamos que la relación $rf/100 = 1.2$ será la más apropiada. Esto se traduce en que la resistencia de realimentación rf debe ser de 120Ω .

5.1 Implementación con componentes ideales

Para comprobar la implementación, se elegirá una matriz 2x2 con la finalidad de simplificar el problema en primera instancia. La matriz 2x2 tendrá las siguientes resistencias: 100Ω , 200Ω , 300Ω y 400Ω , distribuidas de la siguiente manera:

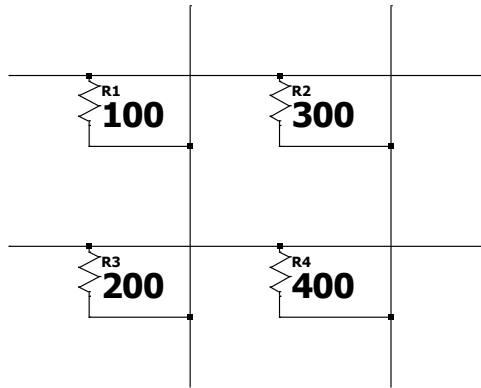


Ilustración 2: Matriz de resistencias 2x2.

Esta configuración inicial facilitará la comprensión y el cálculo de los parámetros necesarios, asegurando que el diseño del circuito sea eficiente y efectivo desde el principio. Con estos valores, $V' = 3V$ y $r_f = 120\Omega$, se logra un diseño que mantiene un rango de salida amplio y eficiente, asegurando que el voltaje de salida se mantenga dentro de los límites de 0V a 5V y que la corriente en el circuito sea suficientemente baja.

En esta etapa del diseño implementamos los voltajes requeridos mediante la utilización de dos fuentes de tensión una de 5V denominada V_{ref} y la otra de 3V denominada V' , así como un AO ideal.

Siguiendo el modelo de ZPM, aunque esta vez en una matriz 2X2 se calculará el voltaje de salida para las resistencias de 100 y 400 Ω , mencionadas con anterioridad.

5.1.1 Cálculos teóricos

Cálculo de la salida de tensión (V_o) para la resistencia de 100 Ω :

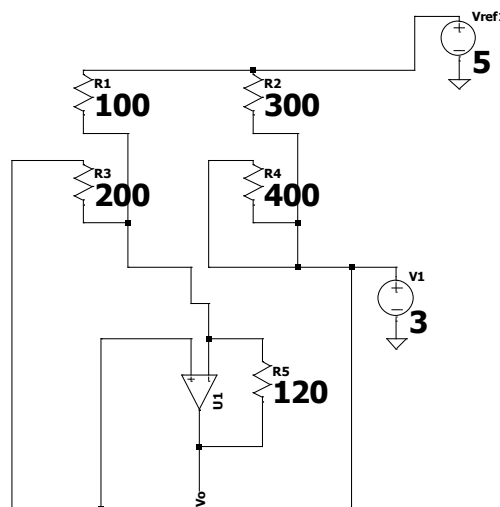


Ilustración 3: Circuito de selección, resistencia de 100 Ω , modelo 1, matriz 2x2.



Como se puede apreciar en la ilustración 3 para seleccionar la resistencia de 100Ω deberemos de conectar dicha fila a V_{ref} (por lo tanto, la otra fila a V') y dicha columna a V - del AO, su entrada inversora (por lo tanto, la otra fila y la otra columna se conectan a V').

La resistencia de 400Ω se encuentra cortocircuitada por lo que podemos eliminarla.

La resistencia de 300Ω puede ser eliminada ya que no se requiere de ella para realizar el cálculo de la V_o dado que la tensión en la patilla menos del AO será $3V$.

Al tratarse de un AO ideal la tensión en ambas patillas es la misma:

$$V_+ = V_-$$

Mediante la utilización de la Ley de Ohm (una de las leyes fundamentales de la electricidad que establece la relación entre el voltaje (V), la corriente (I) y la resistencia (R) en un circuito eléctrico), se halla la corriente que circula por la resistencia de 100Ω :

$$I_{R100} = \frac{V}{R} = \frac{5 - 3}{100} = 20mA$$

Sabiendo que la tensión en la patilla positiva del AO es igual a $3V$, y que a su vez esta es igual a la suma de la tensión de salida V_o y de la tensión de la resistencia r_f , se calcula la V_o :

$$V_o + V_{r_f} = 3V \rightarrow V_o = 3 - V_{r_f}$$

Se aplica la Ley de Ohm para llevar a cabo el cálculo de V_{r_f} , sabiendo que la tensión que circula por I_{r_f} es la misma que al que circula por I_{R100} :

$$V_{r_f} = I_{r_f} * r_f \rightarrow 20 * 10^{-3}A * 120\Omega = 2.4V$$

Se sustituye el valor obtenido en la ecuación anterior:

$$V_o = 3 - V_{r_f} = 3 - 2.4 = 0.6V$$

Cálculo de la salida de tensión (V_o) para la resistencia de 400Ω :

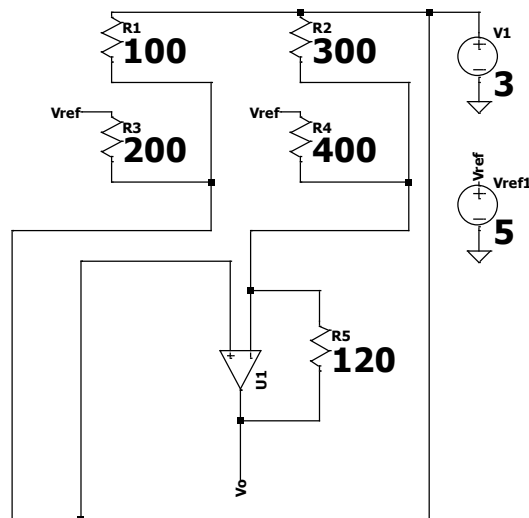


Ilustración 4: Circuito de selección, resistencia de 400Ω, modelo 1, matriz 2x2.

Como se puede apreciar en la ilustración 4 para seleccionar la resistencia de 400Ω deberemos de conectar dicha fila a Vref (por lo tanto, la otra fila a V') y dicha columna a V - del AO (por lo tanto, la otra fila a V').

El valor de la tensión de salida obtenido es de:

$$V_o = 3 - V_{rf} = 3 - 0.6 = 2.4V$$

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 3: Calculo del valor de Vo para la resistencia de 400Ω)

Una vez calculadas las tensiones de salida esperadas se intentó, como primera opción, obtener la tensión de 3V mediante un divisor de tensión, por lo que se calculará el valor de las resistencias pertenecientes a dicho sistema.

Para ello se aplica la formula del divisor de tensión:

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

En nuestro diseño el valor de Vin es de 5V ya que hace referencia al valor de Vref, y el valor de Vin es 3V que es el valor de V' que se desea obtener.

Como se puede intuir la resistencia R2 depende de R1 y viceversa, por lo que presenta multitud de opciones, en nuestro caso se elegirán ambas resistencias del orden de los kΩ, ya que, en un divisor de tensión, es preferible usar resistencias del orden de los kΩ en lugar de los Ω, porque reducen el consumo de corriente y la disipación de calor, lo cual es crítico para la eficiencia y la vida útil de los componentes. Resistencias más altas minimizan el consumo de energía, y evitan

problemas de calentamiento excesivo. Además, al conectar cargas al divisor de tensión, las resistencias más altas mantienen una tensión de salida más estable, ya que la resistencia equivalente cambia menos significativamente en comparación con resistencias bajas.

Por lo que se seleccionan los valores de $R1 = 1k\Omega$ y $R2 = 1k5\Omega$.

Para la resistencia de 100Ω :

Se comprueba mediante LTSpice si al colocar el divisor de tensión se obtiene la tensión de salida esperada al seleccionar la resistencia de 100Ω .

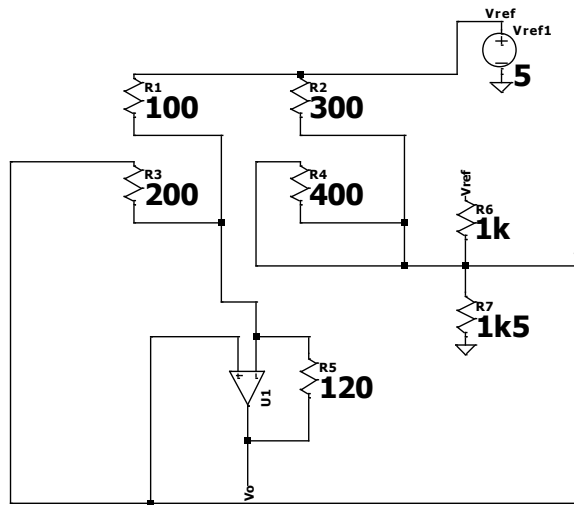


Ilustración 5: Colocación de un divisor de tensión en el circuito de selección de la resistencia de 100Ω .

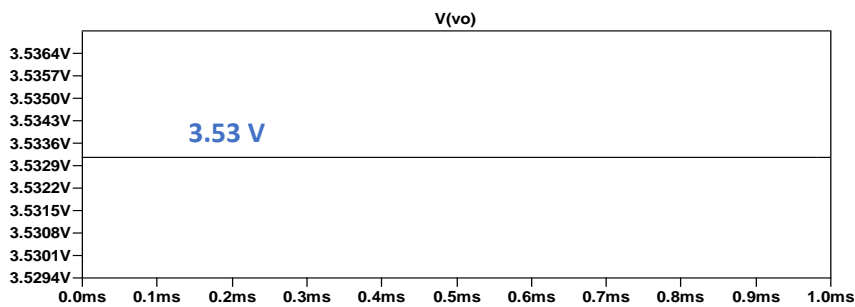


Ilustración 6: Valor de V_o , seleccionando la resistencia de 100Ω .

Como se observa en la ilustración 6, el valor de tensión obtenido es $3.53V$ a diferencia del valor esperado de $0.6V$, lo cual es debido a que la tensión V' no es realmente $3V$, dado que se disipa corriente por la resistencia (en este caso la resistencia de 300Ω), que se encuentra colocada en paralelo a la resistencia de $1k\Omega$, por lo que el divisor de tensión no es el esperado, sabiendo las circunstancias actuales se calcula el valor de V' , aplicando la fórmula del divisor de tensión, esta



vez sustituyendo la resistencia de $1k\Omega$, por la resistencia equivalente resultante de las resistencias de 300Ω y la de $1k\Omega$.

$$V' = \frac{R_{1k5}}{R_{eq} + R_{1k5}} V_{ref}$$

Sabiendo que R_{eq} es:

$$R_{eq} = \frac{R_{1k} * R_{300}}{R_{1k} + R_{300}} = \frac{1k * 300}{1k + 300} = 230.8\Omega$$

Se puede hallar el valor de V' :

$$V' = \frac{R_{1k5}}{R_{eq} + R_{1k5}} V_{ref} = \frac{1k5}{230.8 + 1k5} * 5 = 4.33V$$

Para verificarlo se mide la tensión V' en el diseño de la ilustración 5, mediante LTSpice.

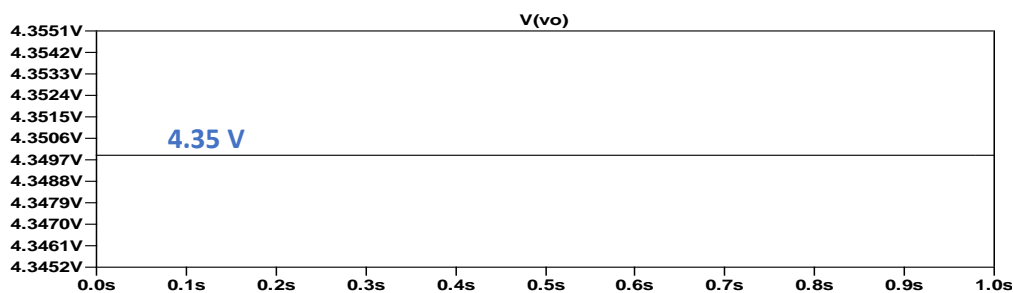


Ilustración 7: Valor de la tensión obtenida mediante el divisor de tensión.

Como se observa en la ilustración 7 en efecto la tensión obtenida es de 4.33V en lugar de 3V que era lo esperado.

Se han realizado las mismas comprobaciones para la resistencia de 400Ω , obteniendo los mismos problemas.

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 2: Comprobación del uso de un divisor de tensión para la obtención de 3V en el modelo 1, seleccionando la resistencia de 400Ω)

Conclusión: no podemos usar un divisor de tensión para obtener V' ya que siempre habrá una resistencia en paralelo a la primera resistencia del divisor, haciendo que el voltaje obtenido no sea 3V. Además, este voltaje variará dependiendo de la fila y columna seleccionada, ya que la resistencia en paralelo cambiará. Por lo tanto, no se puede modelar de esta manera a menos que no haya ninguna carga y el sistema esté en equilibrio. Por esta razón, para realizar los cálculos con el sistema ideal, se utilizará una fuente de tensión de 3V que posteriormente deberá modelarse de otra manera, dado que la única fuente de tensión disponible en el diseño es la de



Vref (+5V). Para obtener la tensión V' en el circuito final utilizaremos un componente denominado referencia de tensión.

5.1.2 Comprobación del circuito ideal mediante LTSpice

Debido a los resultados que se han obtenido en el apartado anterior se realizarán las comprobaciones colocando una fuente ideal de 3V para hallar la V' en lugar de un divisor de tensión.

Para la resistencia de 100Ω :

Por lo que con el circuito de la ilustración 3, compuesto por componentes ideales y seleccionando la resistencia de 100Ω , se obtiene la siguiente V_o :

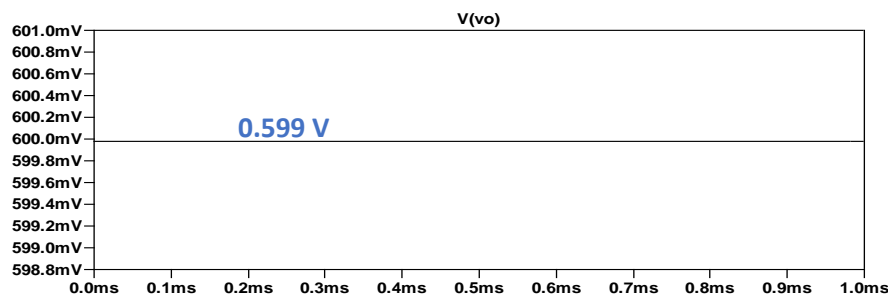


Ilustración 8: Valor de la V_o , seleccionando la resistencia de 100Ω , modelo 1 ideal.

Para la resistencia de 400Ω :

Por lo que con el circuito de la ilustración 4, compuesto por componentes ideales y seleccionando la resistencia de 400Ω , se obtiene la siguiente V_o :

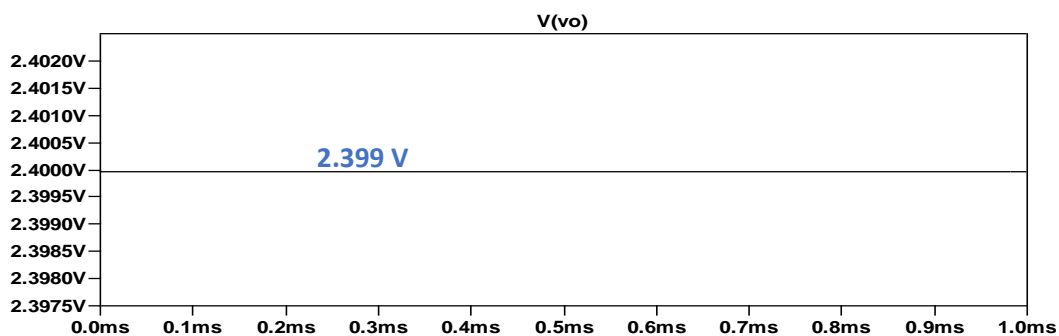


Ilustración 9: Valor de la V_o , seleccionando la resistencia de 400Ω , modelo 1 ideal.

Como se puede apreciar en las ilustraciones 8 y 9, en ambos casos se obtiene el resultado esperado, siendo de $V_o = 0.6V$ para la resistencia de 100Ω y de $V_o=2.4V$ para la resistencia de 400Ω , con unas diferencias mínimas.



5.2 Selección de componentes

5.2.1 Elección del AO

Para seleccionar un AO adecuado, es fundamental elegir uno de precisión que admita una alimentación única (single supply). Los amplificadores operacionales de precisión están diseñados para minimizar errores causados por factores como la variación de temperatura, el ruido y la deriva del voltaje de offset. Estos AOs son especialmente útiles en aplicaciones donde la precisión y la estabilidad a largo plazo son críticas, como en instrumentación y medición.

Es recomendable que el AO seleccionado esté disponible en LTSpice, ya sea en modo texto o en modo esquemático, para facilitar la simulación y validación del diseño.

Para nuestro diseño, necesitamos un AO de precisión para minimizar al máximo el error, garantizando así la exactitud y la estabilidad del sistema. Además, la alimentación del AO debe ser de 5V, alineándose con las especificaciones de nuestro circuito.

En resumen, la elección de un AO de precisión alimentado con una única fuente de 5V es crucial para el rendimiento óptimo y la confiabilidad del diseño, asegurando que los factores externos no comprometan la precisión de las mediciones y operaciones del circuito.

Se ha realizado una búsqueda de componentes en la web de Analog Devices (n.d.-a). Se han seleccionado los amplificadores operacionales de precisión. Posteriormente, se han aplicado una serie de filtros de acuerdo a los requerimientos indicados. se han encontrado en una serie de dispositivos que cumplen con las características requeridas:

- ADA4625-2
- ADA4625-1
- ADA4622-2
- OP777
- OP727

Entre ellos elegiremos el ADA4625-2 ya que, aunque todos tengan single supply y operen a 5V, el ADA4625-2 presenta un asentamiento rápido en single supply por lo que será mejor para nuestro diseño.

5.2.2 Elección de los switches

Se debe llevar a cabo la elección del tipo de switch que será colocado en cada una de las filas y columnas para así poder seleccionar la posición de la matriz deseada.

Para la elección del switch en nuestro diseño nos interesa que la configuración del dispositivo sea 2:1 Single Pole Double Through (SPDT), es un tipo de contacto utilizado en la construcción de interruptores eléctricos y relés. A este tipo de contacto en español, le llamamos Contacto Normalmente Abierto Normalmente Cerrado (NANC). Este tipo de contacto tiene un polo (terminal común) que se puede conectar a uno de dos terminales secundarios (llamados normalmente abiertos y normalmente cerrados). Cuando el polo está en una posición, está



conectado a uno de los dos terminales secundarios, pero cuando se cambia la posición del polo, se desconecta del terminal anterior y se conecta al otro.

Por lo que el contacto SPDT permite cambiar la conexión eléctrica de un circuito entre dos posiciones diferentes, y se utiliza comúnmente en aplicaciones donde se necesita cambiar la dirección del flujo de corriente eléctrica, como en los interruptores de luz, los termostatos y los relés electromecánicos. En el circuito de este trabajo, estos componentes están implementados con tecnología CMOS y son de baja potencia.

Además, necesitaremos una R_{on} (resistencia de entrada, es la impedancia que ofrece el switch a los dispositivos que están conectados a él) baja, la cual permite tiempos de conmutación más rápidos y afecta menos a la medición de las resistencias.

Debido a los requerimientos de nuestro diseño la señal de control deberá ser independiente para cada switch.

Por lo que mediante la aplicación de dichos filtros para conseguir las características deseadas se han encontrado en Analog Devices (n.d.-b) una serie de dispositivos que cumplen con las características requeridas:

- MAX20337
- AD5419
- ADG5236
- ADG5234
- ADG5436
- ADG852
- ADG824

En cuanto al MAX20337, AD5419 y ADG5234 se descartan ya que no presentan modelo en LTSpice por lo que dificultan la simulación y comprobación posterior del diseño.

En cuanto al resto de dispositivos observando el Datasheet de cada uno de ellos podemos concluir que las características del ADG5236 y del ADG5436 son similares pero este último presenta una R_{on} menor, en cuanto al ADG852 presenta single supply entre 1.8V y 5, teniendo en cuenta que nuestra fuente de tensión es de 5V y que además presenta la R_{on} menor podemos afirmar que es el más indicado, ya que aunque comparándolo con el ADG824 esté presente una R_{on} menor, los voltajes de operación también lo son.

Por todo ello finalmente se decide optar por el ADG852.

5.2.3 Elección del dispositivo de obtención del voltaje V'

Debido a que nuestro diseño cuenta únicamente con una fuente de tensión deberemos colocar un dispositivo con la finalidad de obtener el voltaje V' gracias a la fuente de referencia.

Para ello podemos pensar en diferentes maneras de obtenerlo:

Mediante un Zener:

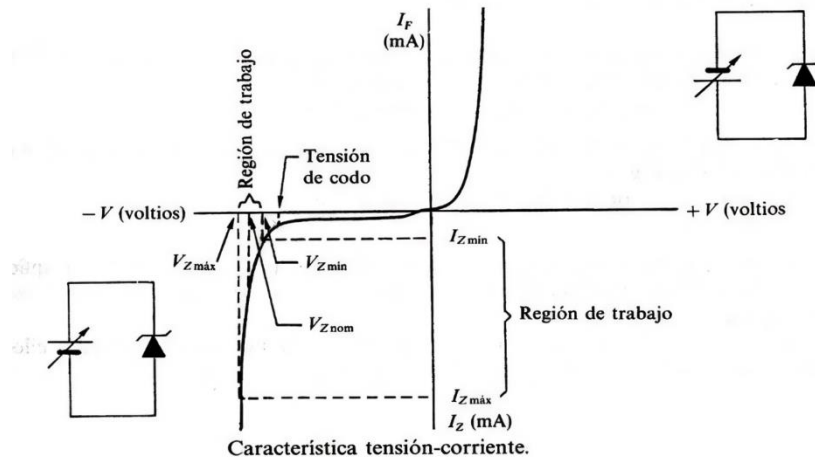


Ilustración 10: Curva característica tensión - corriente, diodo Zener. Fuente: Adaptado de "Diodos (5) - Zener", por ProtoBoard, 2013, Arduino y solo Arduino. Recuperado de <https://soloarduino.blogspot.com/2013/09/diodos-5-zener.html>.

Para obtener el valor de V' utilizaremos un divisor de tensión compuesto por una resistencia conectada a V_{ref} (5V) y a V' , y por un diodo Zener (cuya curva característica es la que aparece en la imagen ...) conectado a V' y a masa.

En cuanto a las características del diodo Zener deberemos elegir el que menor r_z presente, por lo que de esta manera la pendiente también será mayor. La presencia de una r_z implica un cambio en el voltaje del Zener cuando se exige una corriente I :

$$r_z * I$$

Sabiendo que la I_{max} será 20mA, se da cuando se selecciona la resistencia de 100Ω , ya que es la más restrictiva por lo que la $I_{max} = I_{R100}$ (calculada con anterioridad en el apartado 5.1.1).

También es recomendable que se encuentre disponible en LTSpice para la posterior comprobación del diseño.

En nuestro caso se buscará un diodo donde V_z sea 3V, por lo que en las características del diodo elegiremos uno cuya V_{zmin} sea de 2.8V y la $V_{zmáx}$ de 3.2V, de tal manera que el error se minimice al máximo ya que el valor deseado se encuentra en el medio de este rango.

Por lo que mediante la aplicación de dichos filtros para conseguir las características deseadas se han encontrado en Rohm (n.d.) una serie de dispositivos que cumplen con las características requeridas:

- BZX84C3V0LY



- BZX84C3V0LYFH

Observando las hojas de características se puede llegar a la conclusión de que la V_z es para una corriente I_z de 5mA, por lo que no soportan la corriente requerida en nuestro diseño. A su vez, el diodo Zener necesita mantener una corriente mínima (I_r) para garantizar que el voltaje de referencia (V') sea estable y preciso. Si la corriente cae por debajo de este valor, el voltaje de salida puede fluctuar o desviarse de su valor nominal.

Analizando las características podemos afirmar que no es la mejor opción, ya que un diodo Zener puede ser una opción adecuada para aplicaciones donde la precisión no es crítica y donde las condiciones de carga y temperatura son relativamente estables. Sin embargo, en aplicaciones que requieren un voltaje de referencia muy preciso y estable, un referenciador de tensión es generalmente la mejor opción debido a su menor tolerancia, mejor coeficiente de temperatura, y mayor estabilidad ante cambios en la corriente de carga. En el caso del diodo Zener la resistencia dinámica (r_z) provoca una variación en el voltaje Zener (V_z) con cambios en la corriente. Esta resistencia puede causar inestabilidad en el voltaje de salida, especialmente bajo condiciones de carga variable, como es nuestro caso.

Por lo tanto, en nuestro diseño se requiere un voltaje de referencia de 3V con alta precisión y estabilidad, siendo un referenciador de tensión una opción más adecuada que un diodo Zener. Esto se debe a las limitaciones inherentes de los diodos Zener en términos de resistencia dinámica, tolerancia, y variabilidad con la temperatura y la corriente de carga.

Mediante una referencia de tensión:

Las referencias de tensión son componentes que permiten obtener un voltaje de salida preciso, si bien no suelen admitir grandes corrientes. En primer lugar, se deberá de seleccionar aquellos referenciadores de tensión cuya I_{out} Sourcing e I_{out} Sinking sea el máximo posible para poder afirmar que el dispositivo soportará la corriente máxima de 20mA, calculada con anterioridad, que circula por el dispositivo cuando se selecciona la resistencia de 100Ω . En el circuito del modelo 1, la corriente es de entrada (sinking).

En cuanto al fabricante Analog Devices (n.d.-c), el máximo de corriente I_{out} Sinking es de 20mA por lo que se encontrara al límite.

En cuanto a otras especificaciones, se deberá limitar la V_{out} a 3V, con un máximo de 3.3V, ya que el diseño no se puede permitir acumular mayor error.

Atendiendo a las características necesarias aparecen los siguientes dispositivos:

- ADR3630
- LT6658
- ADR433

En cuanto al ADR3630 y ADR433 se descartan ya que no presentan modelo en LTSpice por lo que dificultan la simulación y comprobación posterior del diseño.

Analizando el datasheet del dispositivo LT6658 se puede afirmar que cumple con las características requeridas por el diseño (el dispositivo presenta la posibilidad de obtener 3V, puede ser conectado a una fuente de 5V y presenta modelo en LTSpice).

5.3 Implementación del diseño en LTSpice

Siguiendo los dispositivos seleccionados en los apartados anteriores el diseño completo se dispone como en la ilustración 11.

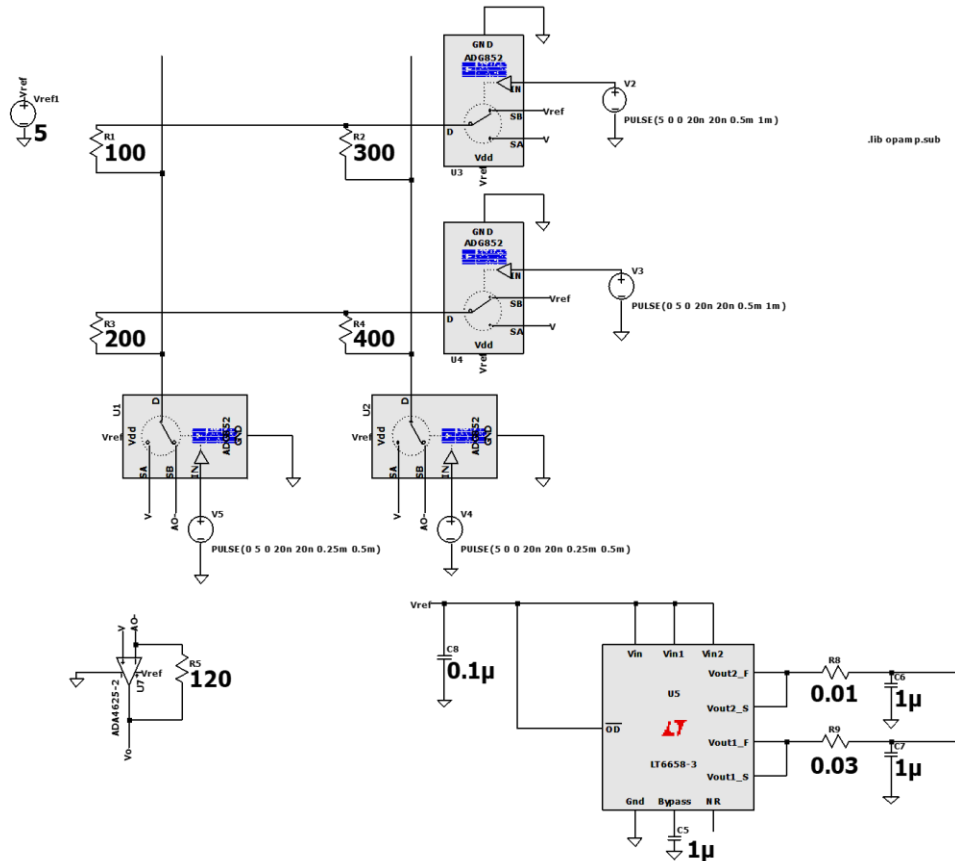


Ilustración 11: Implementación modelo 1.

En cuanto a los switches, la onda generada en la entrada de cada uno de ellos hace que se seleccionen consecutivamente en la simulación de manera automática, con la finalidad de que se pueda presenciar cada uno de los V_o (Voltajes de salida) de cada una de las resistencias presentes en la matriz. De tal manera que cuando se seleccione una fila se pueda seleccionar una de las dos comunes por ello las columnas presentan un periodo de la mitad con respecto a las filas, cuya representación gráfica en modo árbol sería la siguiente.

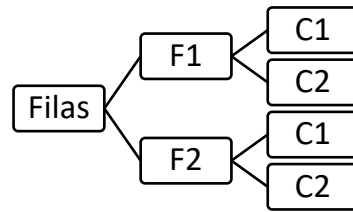


Ilustración 12: Explicación lógica seguida en las ondas de los switches.

Donde F1: fila 1; F2: fila 2; C1: columna 1; C2: columna 2.

Donde el orden de selección de las resistencias es:

- Fase 1: 300 Ω
- Fase 2: 100 Ω
- Fase 3: 400 Ω
- Fase 4: 200 Ω

En la onda de salida, mediante los cursores de LTSpice, se pueden obtener los valores de los voltajes.

Relacionado con el resto de pines de cada uno de los dispositivos se han conectado tal y como aparece en las especificaciones del datasheet.

5.3.1 Simulación del circuito matriz 2x2 en LTSpice

Antes de llevar a cabo la simulación completa de la matriz 2x2, se ha llevado a cabo simulación de los distintos elementos por separado comprobando así que cumplen la función deseada dentro del sistema diseñado.

Además, se debe calcular con anterioridad las salidas V_o esperadas en cada una de las resistencias, mediante la utilización de la ecuación para V_o vista en el apartado 4.2, cuyos valores se representan en la tabla 2.

Resistencias	V_o (V)
300 Ω :	2.2
100 Ω	0.6
400 Ω	2.4
200 Ω	1.8

Tabla 2: Valor de V_o para cada una de las resistencias de la matriz 2x2.

5.3.1.1 Simulación de los switches y el AO reales

Para llevar a cabo la simulación colocamos el AO y los switches seleccionados en nuestro sistema, véase en la ilustración 13. En este apartado el valor de V' proviene de una fuente ideal. De esta forma podemos ir comprobando el funcionamiento de los componentes por separado.

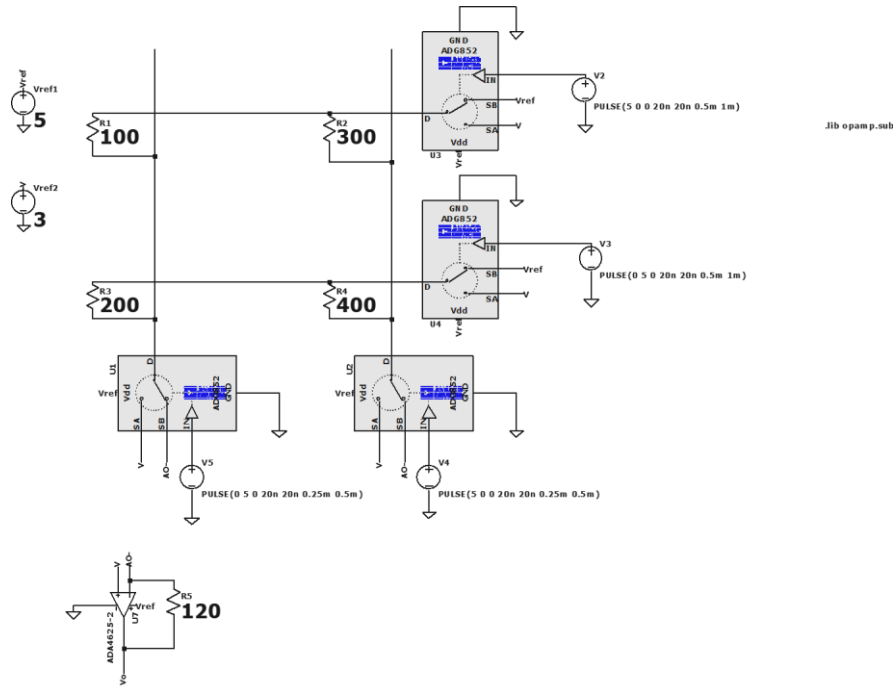


Ilustración 13: Implementación modelo 1, switches y AO reales.

Obteniendo el siguiente gráfico en la salida de Vo.

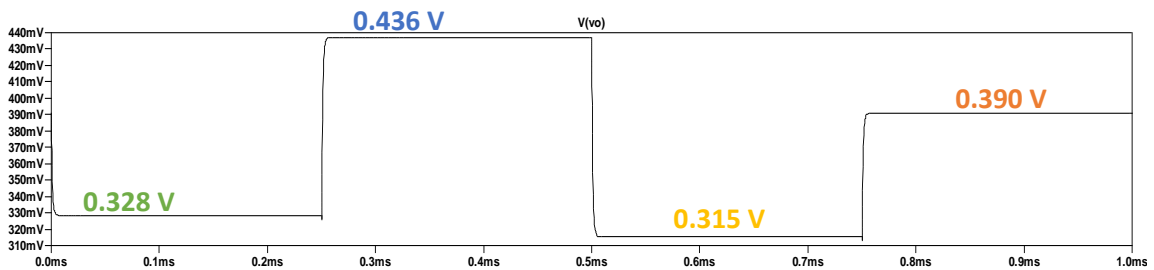


Ilustración 14: Vo en el modelo 1, switches y AO reales.

Por lo que como se observa no se han obtenido los resultados esperados. El voltaje no llega ni siquiera al valor mínimo de 0.6 V indicado en el apartado anterior. Se llega a la conclusión de que nuestro sistema presenta algún error.



Para ello se comprueba el voltaje de salida de cada uno de los switches, véase en la tabla 3.

Switch	Fase 1	Fase 2	Fase 3	Fase 4
Switch 1 (Fila 1)	4.98V	2.99V	4.98V	2.99V
Switch 2 (Fila 2)	2.99V	4.98V	2.99V	4.98V
Switch 3 (Columna 1)	3.01V	2.93V	3.00V	2.49V
Switch 4 (Columna 2)	1.89V	3.00V	1.77V	3.00V

Tabla 3: Voltaje de salida de cada switch en las distintas fases.

En los switches 1 y 2 obtenemos el voltaje esperado, a diferencia de los switches 3 y 4 donde debería de ser el valor de 3V continuamente, debido a que el switch se puede encontrar seleccionando V' o seleccionando la patilla (-) del AO, siendo el valor de estos 3V en los dos casos.

A la vista de las gráficas de la tabla 3, podemos concluir que el voltaje erróneo es el que está relacionado con la patilla (-) del AO.

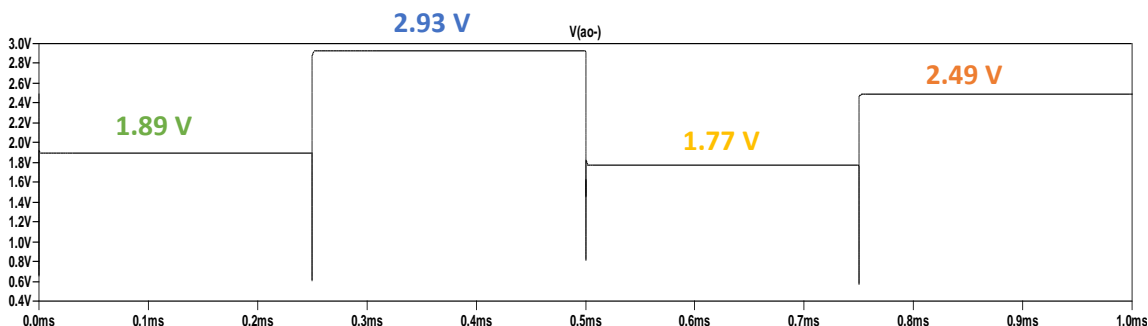


Ilustración 15: Voltaje de la patilla (-) del AO.

En cuanto a la gráfica, se observa que la tensión debería ser constante, cuyo valor sería de 3V, en cambio esto no sucede así.

Una de las causas que puede provocar un funcionamiento incorrecto y una salida que se aleje de la esperada es una corriente excesiva para el AO, o bien que la tensión se acerque demasiado a los límites.

Para comprobar lo sucedido se colocan los switches reales y el AO ideal, como aparece en la ilustración 16.

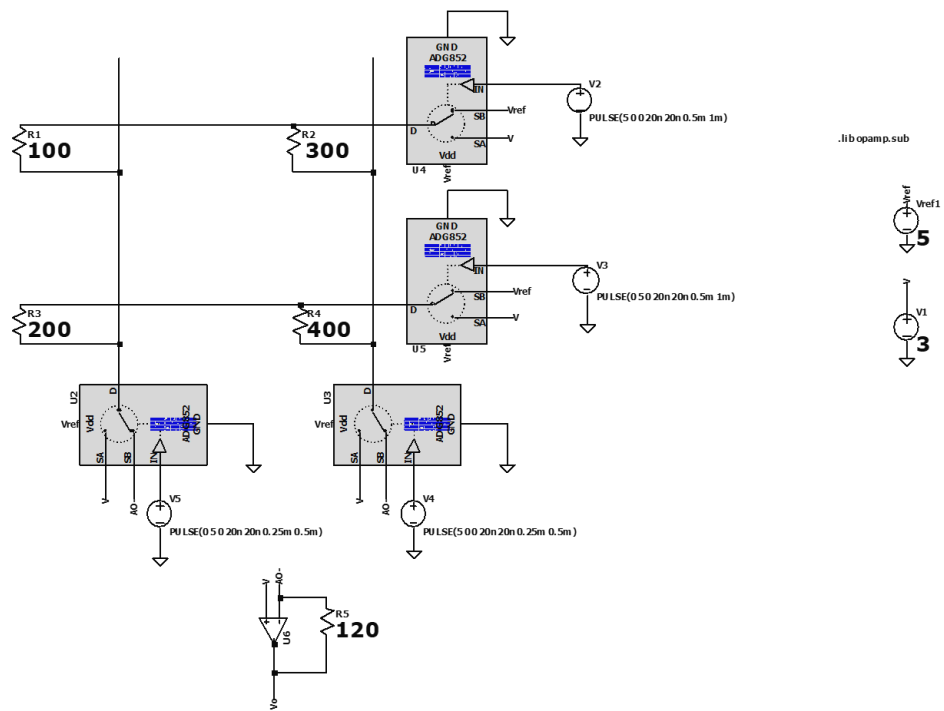


Ilustración 16: Modelo 1, AO ideal.

Obteniendo la salida V_o que aparece en la ilustración 17.

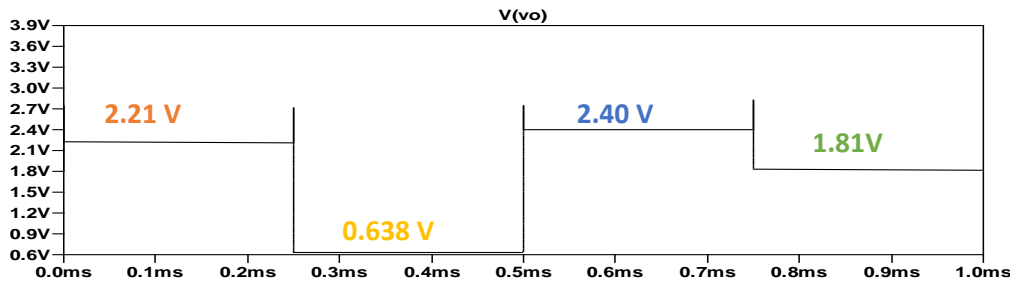


Ilustración 17: Modelo 1, AO ideal, valor de V_o .

Siendo el voltaje en la patilla (-) del AO el perteneciente a la ilustración 18, cuyo valor es 3V.

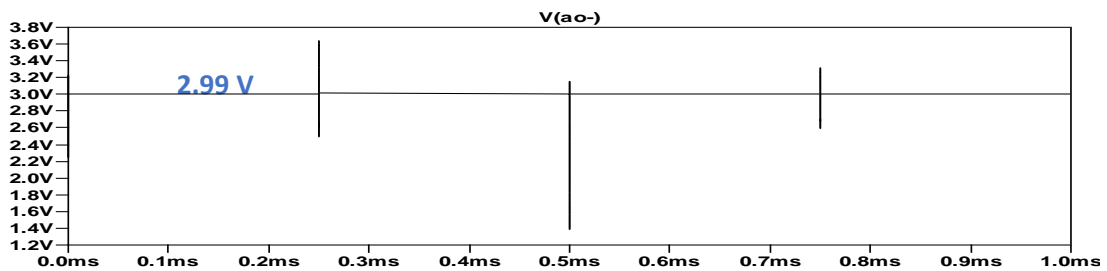


Ilustración 18: Modelo 1, AO ideal, valor de voltaje de la patilla (-).

Llegando a la conclusión final de que el problema viene del AO por lo que se cambiará este con la finalidad de solucionar el problema.

Por ello se sustituirá por el AO: ADA4622-2.

El gráfico reflejado en la ilustración 19 afirma que la utilización de este AO sería la mejor opción para nuestro diseño.

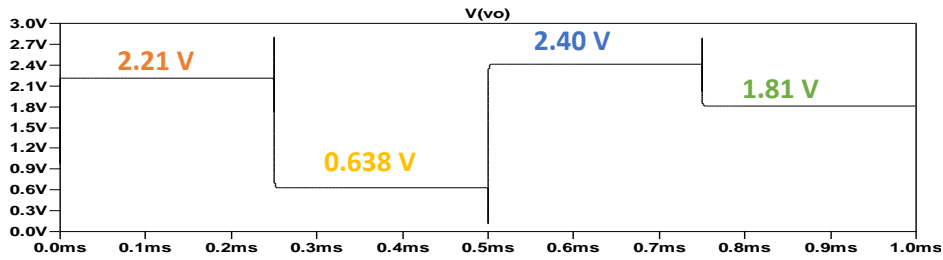


Ilustración 19: Modelo 1, V_o , voltaje V' ideal.

Hasta llegar a dicha solución se han obtenido otras posibles soluciones al problema, optando finalmente por esta ya que es la que más conserva la idea del diseño original.

(Para mayor detalle sobre los pasos y soluciones halladas a dicho problema véase el Anexo II: Análisis y Soluciones derivadas de Problemas en Amplificadores Operacionales)

5.3.1.2 Simulación de la referencia de tensión real

Se conectará el componente LT6658-3 tal y como aparece en el datasheet, representado en la ilustración 20.

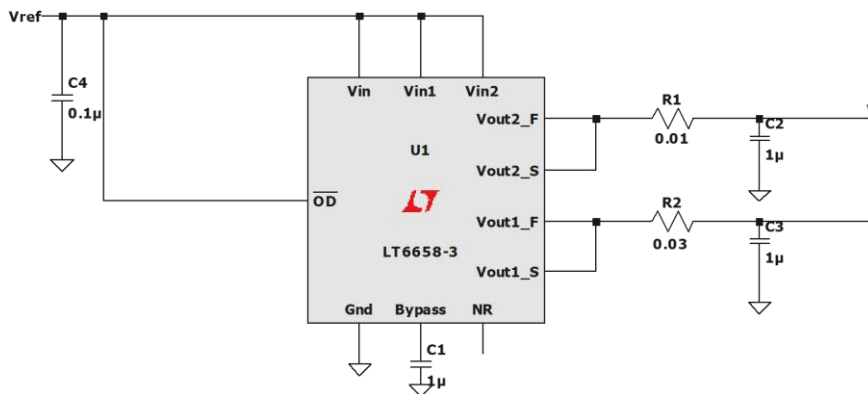


Ilustración 20: Conexiones del LT6658-3.

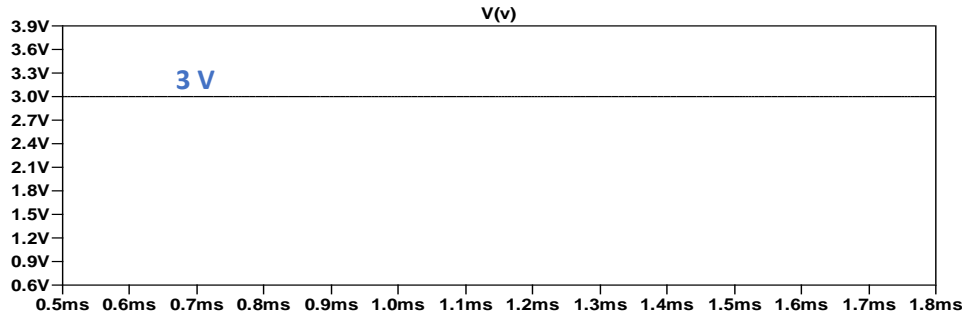


Ilustración 21: Voltaje de salida del LT6658-3.

Como se puede observar en la ilustración 21, se obtiene el voltaje esperado. En realidad, este componente tiene dos salidas. Se han conectado entre sí a través de resistencias de muy bajo valor como indica el fabricante para mejorar la corriente que puede recibir.

5.3.2 Configuración de la simulación completa

Representación del modelo final en la ilustración 22.

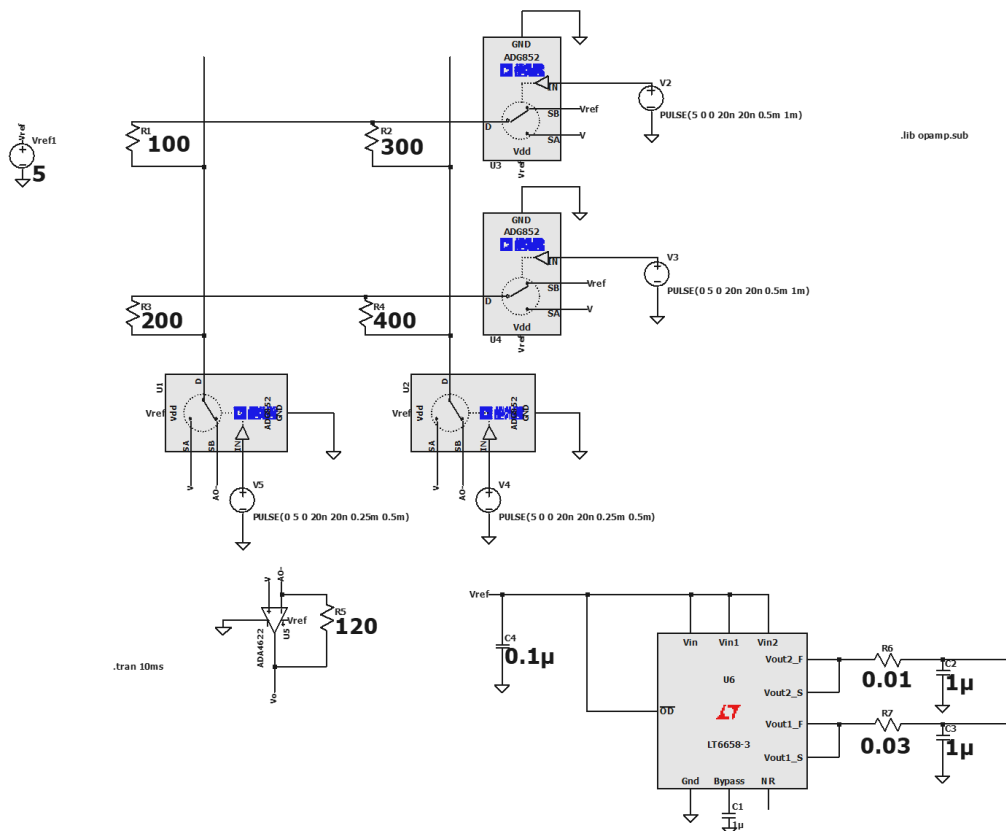


Ilustración 22: Circuito final modelo 1.

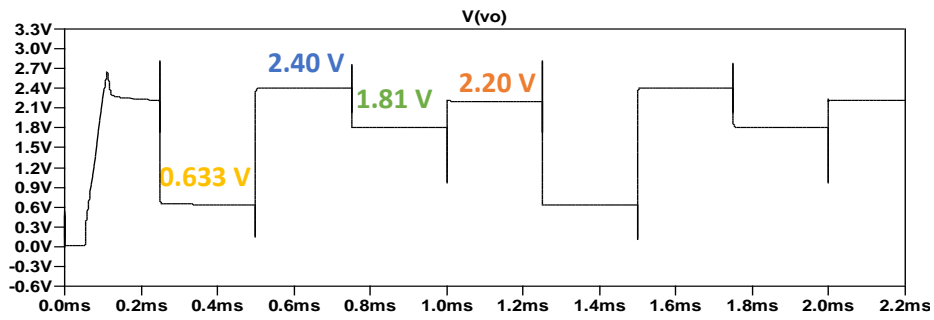


Ilustración 23: Circuito final modelo 1, Vo.

Por lo que obtenemos el resultado esperado.



6 Implementación del modelo 2

Para implementar el modelo 2, es necesario calcular los valores de V' y r_f , dado que los valores de V_{ref} y R_{1B} están determinados por las condiciones del circuito. Es fundamental asegurarse de que el voltaje de salida del amplificador operacional (AO) se mantenga dentro del rango de 0 a 5V.

El modelo 2 presenta las mismas características que el modelo 1, pero en este caso se utiliza una única referencia de tensión en el diseño, por lo que simplificamos notablemente el sistema.

Se va a llevar a cabo el cálculo de los valores de V' y r_f fijando los valores típicos de referencia de tensión. Sabiendo que ambos diseños son equivalentes en términos de circuito utilizaremos las mismas ecuaciones calculadas con anterioridad.

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 2: Calculo del valor de V' y r_f mediante la aplicación de las condiciones)

Siendo el valor de la tensión de salida (V_o):

$$V_o = V' * \left(1 + \frac{r_f}{r_{1b}}\right) < 5$$

Una vez calculado el valor, hallamos r_f para los valores típicos de referencias de tensión de 1.8V, 2.5V y 3V.

- Referencia de 3V:

$$V_o = 3 * \left(1 + \frac{r_f}{100}\right) < 5 \rightarrow r_f < 66.7\Omega$$

- Referencia de 2.5V:

$$V_o = 2.5 * \left(1 + \frac{r_f}{100}\right) < 5 \rightarrow r_f < 100\Omega$$

- Referencia de 1.8V:

$$V_o = 1.8 * \left(1 + \frac{r_f}{100}\right) < 5 \rightarrow r_f < 177.8\Omega$$

Para realizar los cálculos, teniendo en cuenta que para seleccionar el referenciador de tensión adecuado deberán de cumplirse todas las condiciones, se elige una $r_f = 50\Omega$. Además, este valor es preferible que no sea demasiado pequeño para la salida presente una mayor amplitud

6.1 Selección de componentes

Dado el diseño del circuito utilizaremos los mismos componentes que para el modelo 1, ya que los componentes deberán seguir el mismo criterio de selección que para el modelo 1 (aparece en el apartado 4.2 Selección de componentes).

- AO: ADA4622-2
- Switches: ADG852
- Referenciador de tensión: LT6658

En cuanto a la elección del voltaje de referencia se va a llevar a cabo la realización de diferentes pruebas con LTSpice con los voltajes de referencia típicos (1.8V, 2.5V y 3V). Este componente tiene variantes que permiten obtener dichos voltajes.

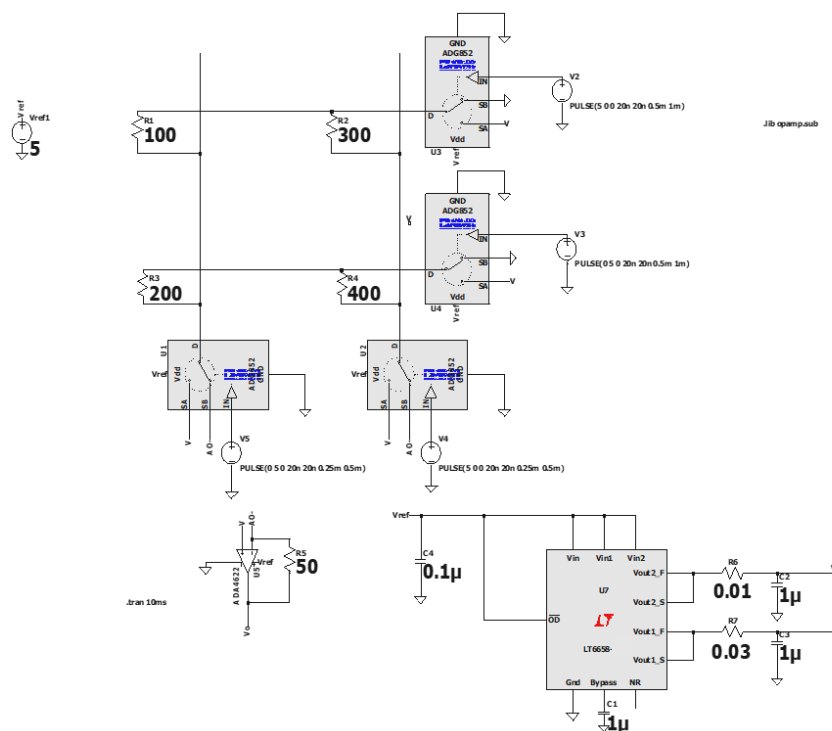


Ilustración 24: Circuito modelo 2.

A continuación, se calculará el valor de la salida para cada una de las resistencias y cada uno de los voltajes de referencia, representados en la tabla 4, aplicando la fórmula que siguiente:

$$V_o = V' - \left(\frac{0 - V'}{r1B} * rf \right) = V' - \left(\frac{0 - V'}{r1B} * 50 \right)$$

(Véase con mayor detalle en el Anexo I: Cálculos, apartado 5: Cálculo del valor de la tensión de salida al sustituir las masas por un voltaje denominado V' modelo 2)



Valores típicos de referencias de tensión (V)			
	1.8	2.5	3.0
Resistencias (Ω)	Valor de la tensión de salida (V_o) en (V)		
300	2.10	2.92	3.5
100	2.70	3.75	4.5
400	2.03	2.81	3.38
200	2.25	3.13	3.75

Tabla 4: Voltaje de salida para cada resistencia y cada referencia de tensión.

6.2 Implementación del diseño en LTSpice

Véase en la ilustración 24 la representación del modelo en LTSpice

En cuanto a los switches, como sucedía en el modelo 1, la onda de entrada sigue el mismo razonamiento por lo que la explicación aparece en el apartado 5.3 Implementación del diseño en LTSpice

6.2.1 Simulación del circuito matriz 2x2 en LTSpice

En cuanto a la simulación de la matriz de resistencias 2x2 se van a llevar a cabo 3 simulaciones dependiendo del valor de voltaje de referencia seleccionado, analizadas y representadas a continuación, por lo que una vez estudiados cada uno de los casos se decidirá el valor de voltaje de referencia idóneo para nuestro diseño, representando finalmente la configuración completa del modelo 2.

6.2.1.1 Simulación del referencia de tensión real

Seguidamente se representarán cada uno de los circuitos pertenecientes a las referencias de voltaje, así como las gráficas del voltaje de salida (V_o).

- Referencia de voltaje de 3V:
Dicho circuito aparece representado en la ilustración 25.

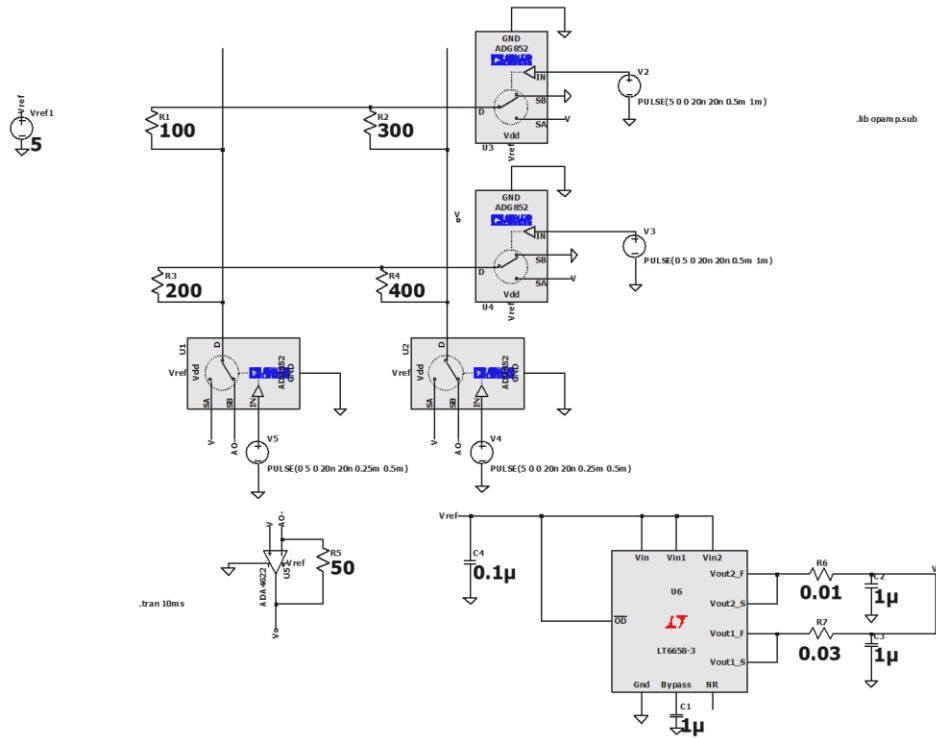


Ilustración 25: Circuito modelo 2, referencia de tensión de 3V.

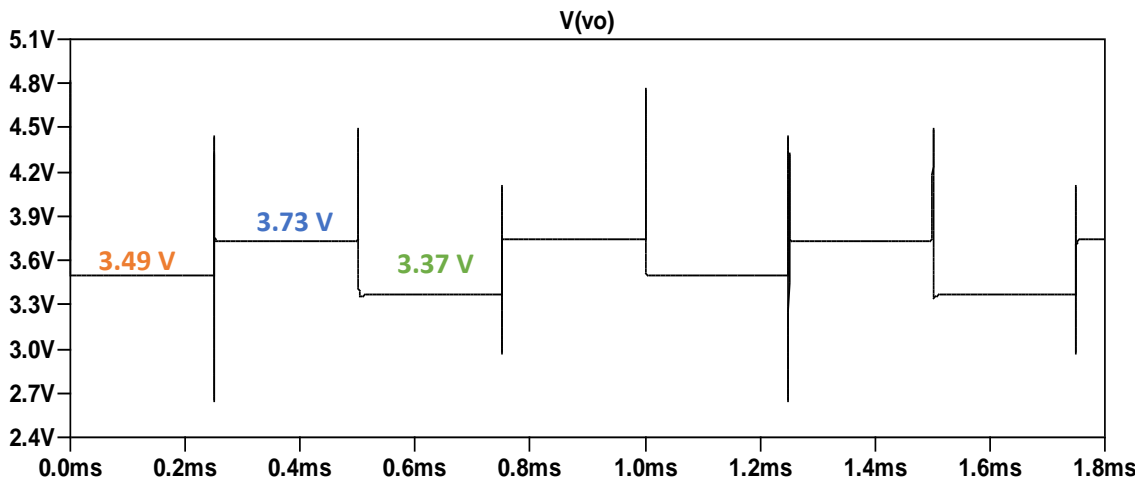


Ilustración 26: Circuito modelo 2, referencia de tensión de 3V, Vo.

Según los valores pertenecientes a la tabla 4, se puede concluir que obtenemos un valor similar al esperado en todos los voltajes de salida a excepción del perteneciente a la resistencia de 100 Ω , cuyo valor dista 0.8V por debajo de lo esperado.

- Referencia de voltaje de 2.5V:
Dicho circuito aparece representado en la ilustración 27.

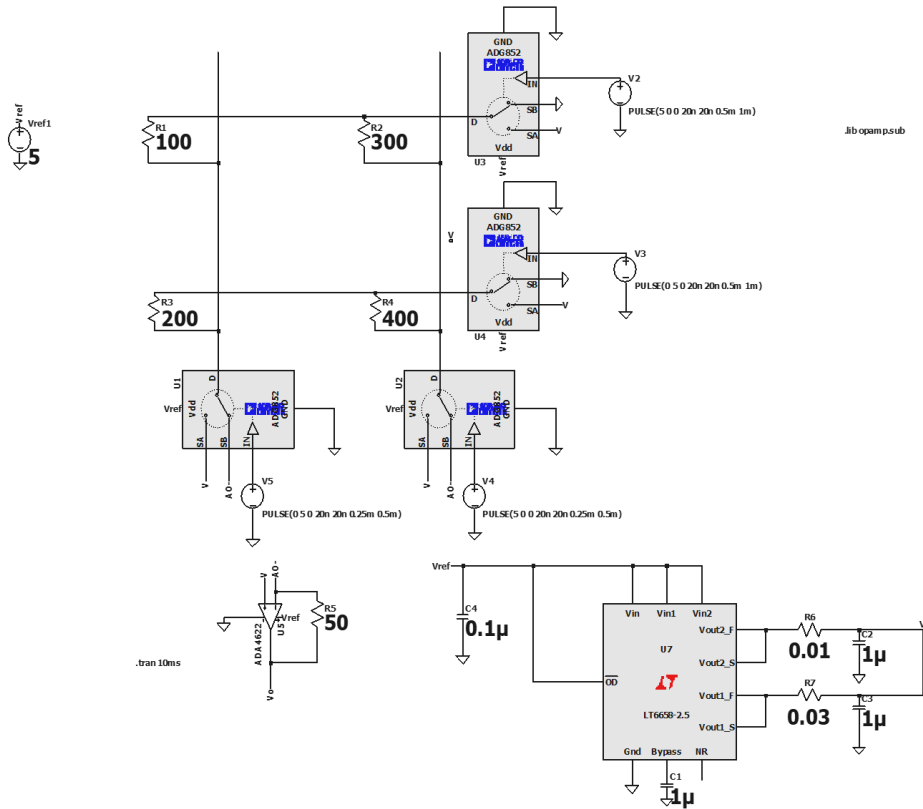


Ilustración 27: Circuito modelo 2, referencia de tensión de 2.5V.

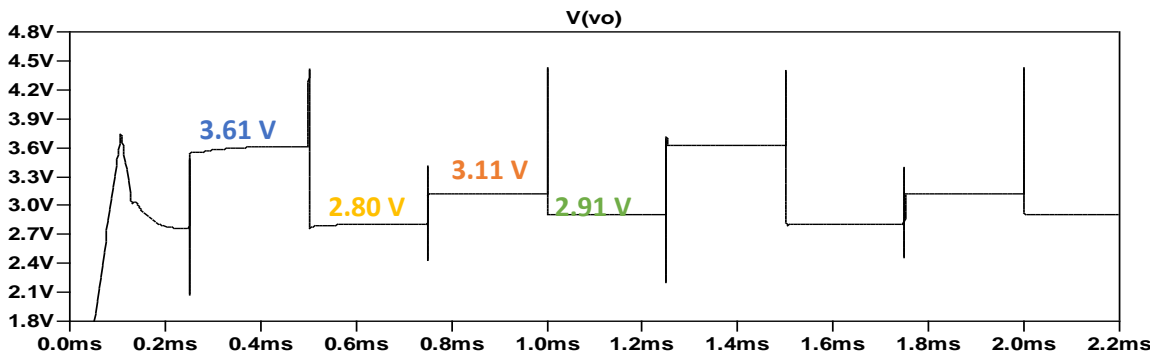


Ilustración 28: Circuito modelo 2, referencia de tensión de 2.5V, Vo.

En el caso de utilizar el referenciador de voltaje de 2.5V, el valor obtenido en la gráfica perteneciente a la ilustración 28, la cual hace referencia al voltaje de salida dista mucho menos de los valores de la tabla 4, siendo la máxima diferencia de 0.1V por debajo del valor esperado para la resistencia de 100Ω.

- Referenciador de voltaje de 1.8V:

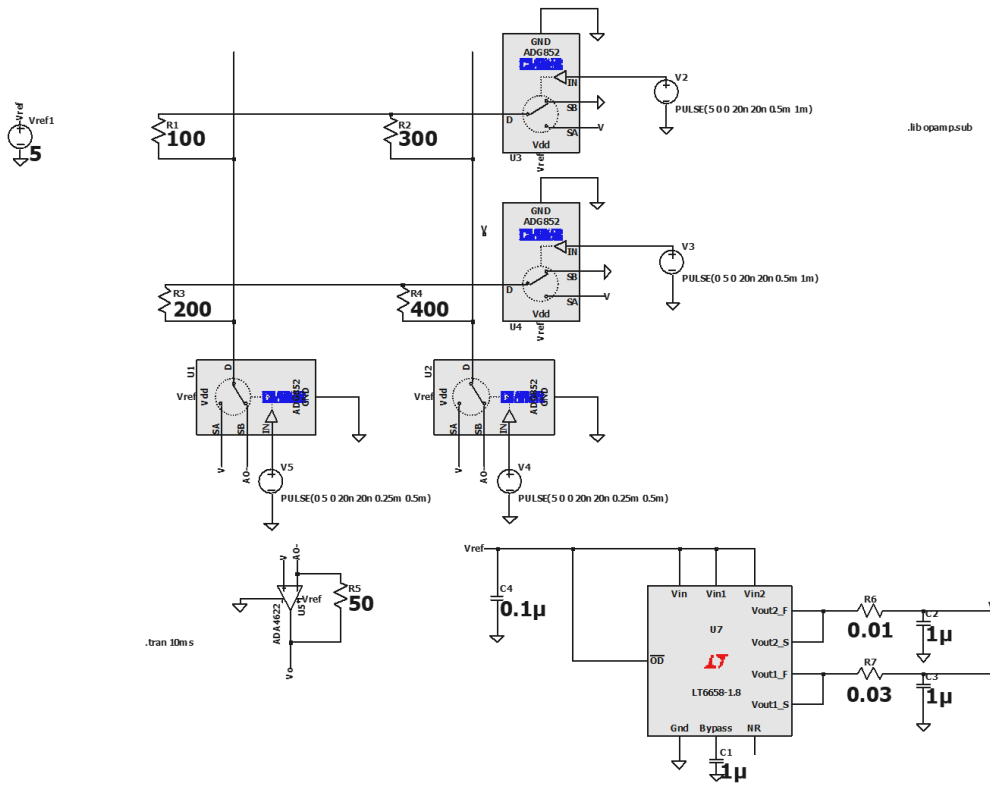


Ilustración 29: Circuito modelo 2, referencia de tensión de 1.8V.

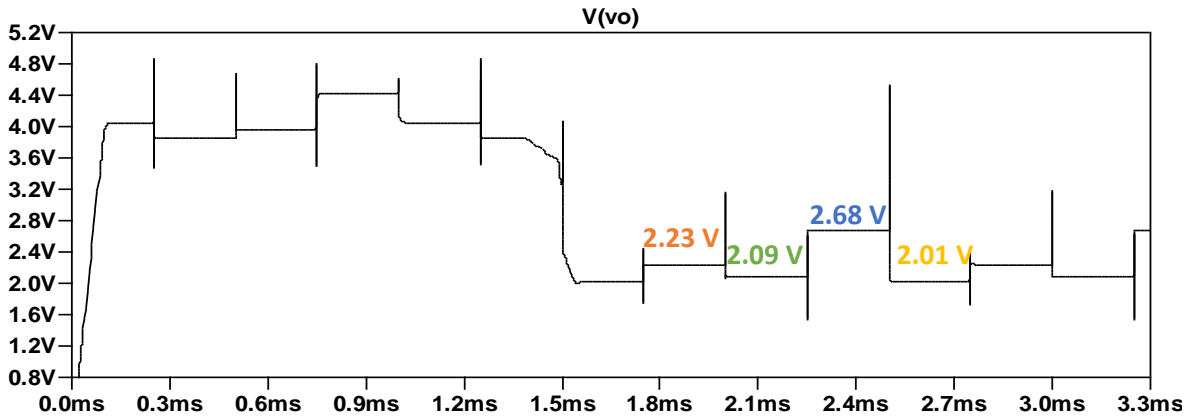


Ilustración 30: Circuito modelo 2, referencia de tensión de 1.8V, V_o .

Analizando las ilustraciones 26,28 y 30, se llega a la conclusión de que comparándolo con los valores de la tabla 4 es la mejor opción, cuyos valores son próximos a los esperados.

6.2.2 Configuración de la simulación completa

Dado que el voltaje de referencia seleccionado es el de 1.8V., podemos cambiar el valor de r_f , ya que al calcularlo se ha obtenido que: $r_f < 177.8\Omega$, por lo que lo cambiaremos a $r_f = 100\Omega$, obteniendo el diseño de la ilustración 29, por lo que deberemos de calcular previamente los valores de salida esperados:

El valor de V_o (calculado con anterioridad en el apartado 2, para el modelo 2) se representa para las resistencias de la matriz en la tabla 5.

Resistencias	$V_o(V)$
300 Ω	2.4
100 Ω	3.6
400 Ω	2.25
200 Ω	2.7

Tabla 5: Voltaje de salida para cada resistencia y referencia de tensión de 1.8V.

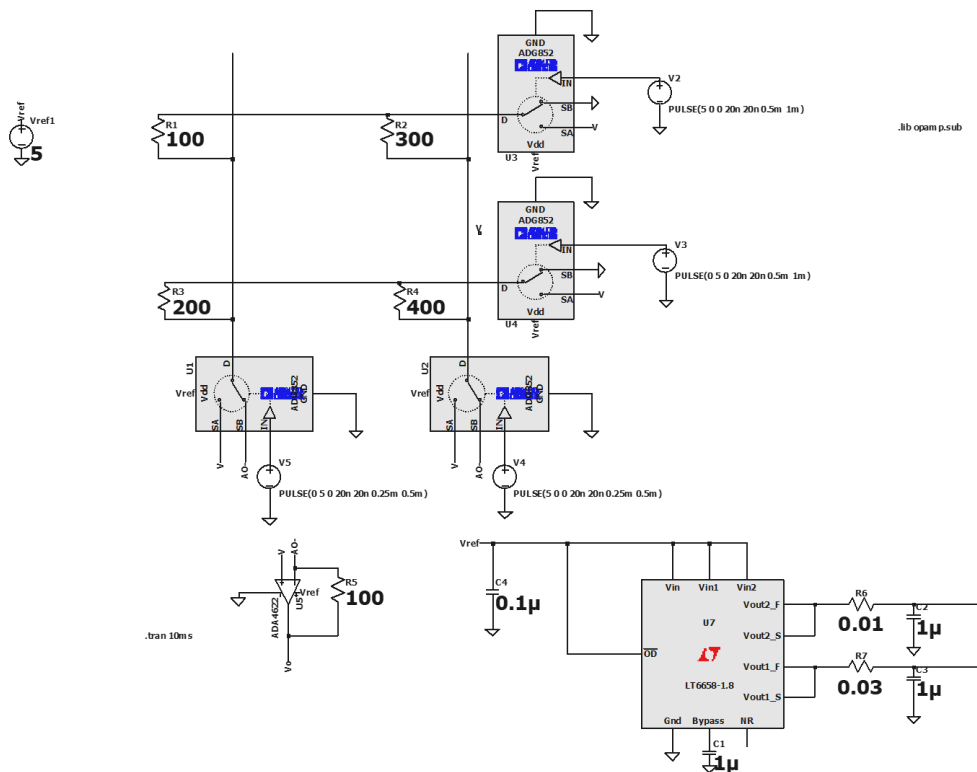


Ilustración 31: Circuito final modelo 2.

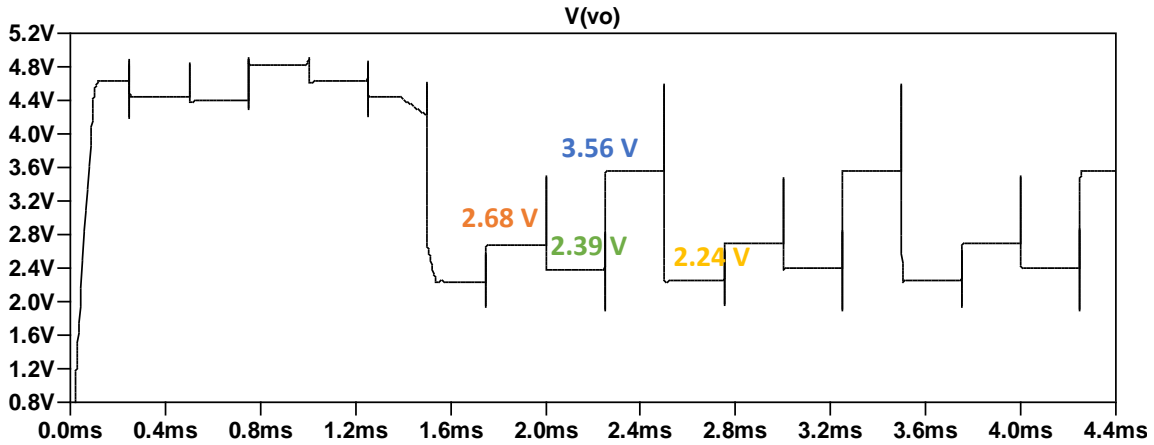


Ilustración 32: Circuito final modelo 2, V_o .

Véase en la ilustración 32 la concordancia de dicha gráfica con los resultados esperados, a diferencia del periodo inicial donde tarda 1.5ms en estabilizarse, por lo que se coloca una etapa de colector común utilizando un transistor NPN con la finalidad de actuar como un buffer, aumentando la capacidad de corriente y mejorando la respuesta en frecuencia del circuito. Esto ayuda al AO a mantener una salida más estable y rápida, lo que explica por qué V_o se estabiliza más rápidamente cuando utilizas un transistor NPN. Además, la corriente de salida en el AO es menor, ya que corresponde a la base del transistor NPN.

Comprobaremos dicha premisa en LTSPice.

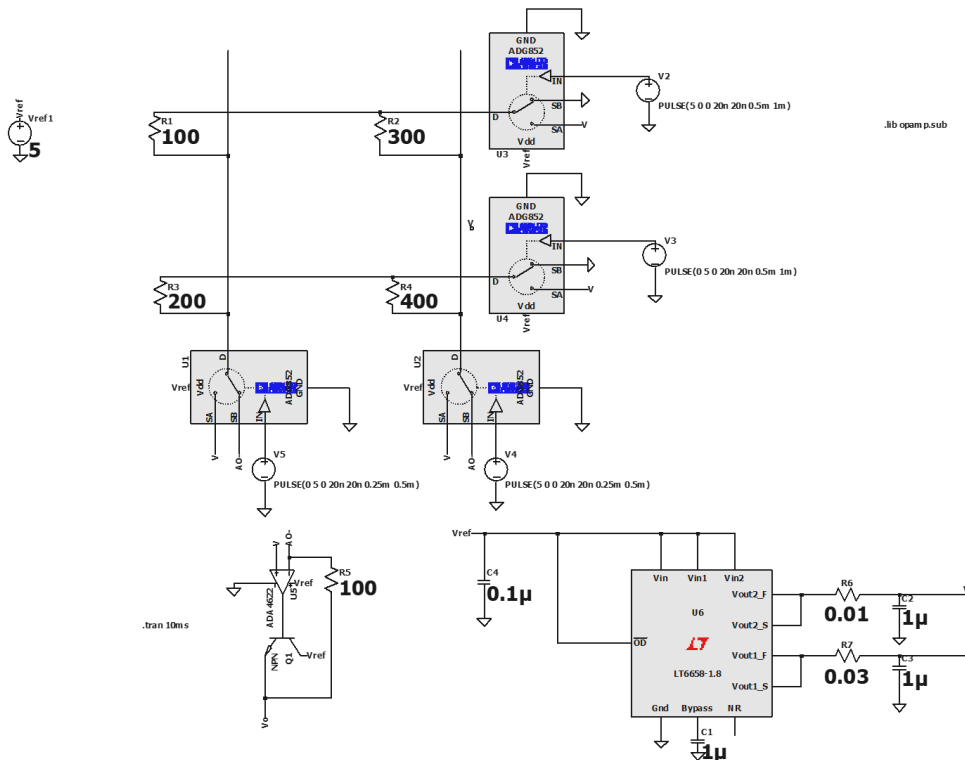


Ilustración 33: Circuito final modelo 2, NPN.

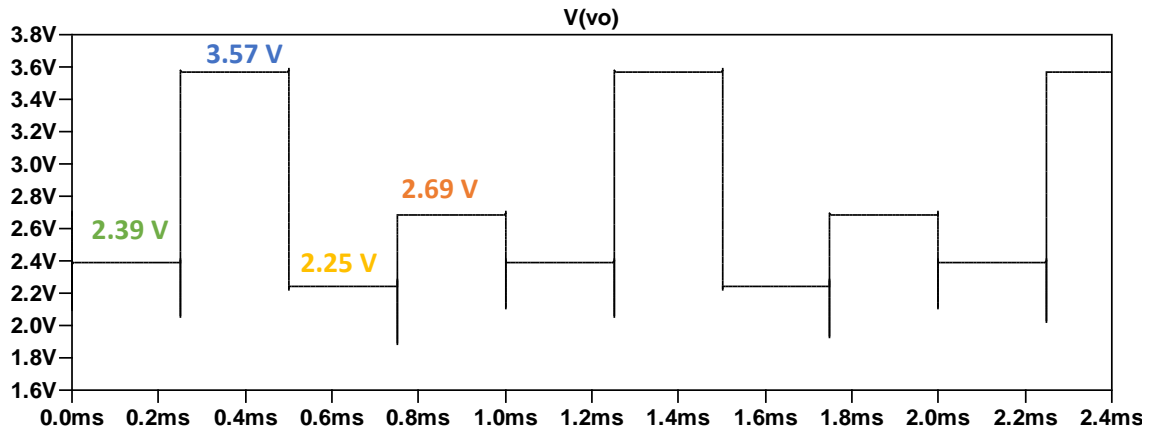


Ilustración 34: Circuito final modelo 2, NPN, V_o .

En efecto, tal y como se aprecia en la ilustración 34, se ha corregido el problema, por lo que este será el circuito final para el modelo 2.

Dicho modelo se ha implementado mediante adaptaciones en la asignatura de Diseño electrónico cursada el 2º cuatrimestre del 4º curso, para ello véase el Anexo V: Implementación modelo 2.



7 Automatización del Proceso de Simulación con Python

7.1 Generación automática de archivos de simulación LTSpice

Con el objetivo de automatizar todo el proceso, se ha desarrollado una función en Python que, al proporcionar una serie de parámetros específicos, genera un archivo .cir (netlist). Este archivo puede ser importado y simulado directamente en LTSpice, lo que permite analizar el comportamiento en un punto determinado de la matriz de resistencias o incluso examinar el rendimiento del circuito en su totalidad. Esta herramienta no solo facilita la creación y modificación de esquemas en LTSpice, sino que también optimiza el flujo de trabajo al permitir simulaciones precisas y rápidas basadas en los parámetros de entrada definidos. Así, se posibilita un estudio exhaustivo de las condiciones operativas del circuito, mejorando la eficiencia y precisión en el análisis de resultados.

7.1.1 Librerías y herramientas utilizadas

Para la programación de ambos modelos se ha utilizado la librería Numpy ya que es fundamental para poder trabajar en el proyecto.

NumPy, abreviatura de Numerical Python, es capaz de proporcionar las estructuras de datos, algoritmos y conectores necesarios para aplicaciones enfocadas a datos numéricos. Además de análisis de datos, su uso puede enfocarse como contenedor para datos que se pasan entre algoritmos y bibliotecas, manipulando y almacenando datos, permitiendo que otras bibliotecas en lenguaje de bajo nivel (como el C) puedan operar con los datos almacenados en una matriz de NumPy sin copiar los datos en otra representación de memoria, según McKinney (2017)

Dicha librería aparece en la primera línea de código dentro del script tal y como se indica a continuación:

```
import numpy as np
```

Importando la librería NumPy y asignándola al alias np.

7.1.2 Estructura del script en Python

En cuanto a la generación de la matriz de resistencias necesaria en ambos modelos pueden generarse de 3 maneras diferentes, por lo que dependiendo que script se ejecute, la matriz pertinente se guardará en el valor rarray.

- Introducción directa de los valores de la matriz.

```
rarray = np.array([[ 100, 300],[200,400]])
```



np.array es una función que crea un objeto **ndarray**, una matriz multidimensional, perteneciente a la biblioteca **NumPy** mencionada con anterioridad, seguidamente aparece la lista de listas, representando una matriz 2x2, donde la primera lista corresponde a la primera fila, y la siguiente a la segunda. Finalmente, el array se almacena en la variable **rarray**.

- Creación de una matriz aleatoria.

```
rarray = np.random.choice([1e3, 10e3, 100e3], size = (20,2) )
```

np.random.choice es una función que selecciona aleatoriamente elementos de una lista o array, perteneciente a la biblioteca **NumPy** mencionada con anterioridad, seguidamente aparece una lista de tres valores que son los posibles valores a elegir de manera aleatoria (estos valores podrían ser cambiados o incluso añadir más), representando una matriz 20x2, tal y como especifica a continuación en la línea de código. Finalmente, la función devuelve una matriz 20x2 donde cada valor es uno de los valores especificados, la matriz se almacena en la variable **rarray**.

- Introducción de una matriz guardada en un fichero adjunto cargado en Google colab.

```
rarray = np.load('ralex1.npy')
```

np.load es una función que se utiliza para cargar archivos cuyo formato sea **.npy** o **.npz**, perteneciente a la biblioteca **NumPy** mencionada con anterioridad, seguidamente aparece el nombre del archivo en el que están los datos (para poder ejecutar esta línea de código previamente se debe haber cargado el archivo en el mismo directorio), en el caso de que se desee cargar otro archivo se deberá cambiar el nombre por el pertinente. Finalmente, la función carga el array almacenado en el archivo y lo asigna a la variable **rarray**.

El siguiente script pertenece a la creación de una función cuya finalidad es desarrollar el fichero cir que se podrá ejecutar en LTSpice simulándolo. La función recibe una serie de parámetros:

- La matriz de resistencias (descrita con anterioridad).
- Los componentes a utilizar, en el caso de ambos modelos son los siguientes componentes:
 - ADA4622, AO.
 - ADG852, switches.
 - LT6658, referencia de voltaje.
- Fila y columna seleccionadas o ninguna (ya que cabe la posibilidad de realizar una simulación completa, donde a lo largo del espacio del tiempo se van simulando cada una de las posiciones de la matriz una por una).
- Tiempo de simulación.
- Nombre de un fichero.

En el interior de la función se crea una variable de tipo string (texto) a la que se le van añadiendo elementos hasta conformar una netlist que se puede simular en spice. Este texto se guarda en



el fichero. Existe la opción también de que se vayan seleccionando todas resistencias de la matriz, para lo cual se generan ondas de voltaje adecuadas para la selección en los switches.

(Véase con mayor detalle en el Anexo IV: Código, apartado 1: Modelo 2).

7.2 Integración de la simulación con Python y LTSpice

7.2.1 Flujo de trabajo automatizado

Se ha logrado automatizar por completo el proceso de creación del archivo de simulación cir para ambos modelos, independientemente de la matriz resistiva utilizada, ya sea en términos de valores, dimensiones o configuraciones específicas. Gracias a esta automatización, el código desarrollado permite simular cualquier matriz resistiva de manera directa en LTSpice, sin necesidad de ajustes manuales adicionales.

A continuación, aparecen dos ejemplos de las netlists generadas con dicha automatización del trabajo.

- Netlist del modelo 1, con la matriz 2x2 utilizada en el diseño, sin selección concreta de fila y columna, ya que, aunque el programa permite la selección de dicha posición de la matriz, la generación de la netlist que permita una simulación completa es más elaborada.

```
* Automatic generation - Creation cir R
Vref1 ref1 0 5
R0_0 NF0 NC0 100
R0_1 NF0 NC1 300
R1_0 NF1 NC0 200
R1_1 NF1 NC1 400
RAO AON Vo 120
XU_0 NS_0 V NC0 AON ref1 0 ADG852
XU_1 NS_1 V NC1 AON ref1 0 ADG852
XU0_ NS0_ V NF0 ref1 ref1 0 ADG852
XU1_ NS1_ V NF1 ref1 ref1 0 ADG852
V0_ NS0_ 0 PULSE(0 5 0.0000m 20n 20n 0.5000m 1.00001m)
V1_ NS1_ 0 PULSE(0 5 0.5000m 20n 20n 0.5000m 1.00001m)
V_0 NS_0 0 PULSE(0 5 0.0000m 20n 20n 0.2500m 0.50000m)
V_1 NS_1 0 PULSE(0 5 0.2500m 20n 20n 0.2500m 0.50000m)
XAO V AON ref1 0 Vo ADA4622
X3V ref1 ref1 ref1 ref1 N3V1 N3V2 N3V2 N3V3 N3V3 0 N3V4
LT6658-3
C1 N3V1 0 1u
C2 V 0 1u
C3 V 0 1u
R3V1 V N3V3 0.01
R3V2 V N3V2 0.03
C4 ref1 0 0.1u
.lib opamp.sub
.tran 10ms
.lib ADA4622.lib
.lib ADG852.sub
```



```
.lib LT6658.lib  
.backanno  
.end
```

- Netlist del modelo 2, con la matriz 2x2 utilizada en el diseño, sin selección concreta de fila y columna.

```
* Automatic generation - Creation cir R  
Vref1 ref1 0 5  
R0_0 NF0 NC0 100  
R0_1 NF0 NC1 300  
R1_0 NF1 NC0 200  
R1_1 NF1 NC1 400  
RAO AON Vo 100  
XU_0 NS_0 V NC0 AON ref1 0 ADG852  
XU_1 NS_1 V NC1 AON ref1 0 ADG852  
XU0_ NS0_ V NF0 0 ref1 0 ADG852  
XU1_ NS1_ V NF1 0 ref1 0 ADG852  
RC_0 NS_0 0 10  
RC_1 NS_1 0 10  
RC0_ NS0_ 0 10  
RC1_ NS1_ 0 10  
V0_ NS0_ 0 PULSE(0 5 0.0000m 20n 20n 0.5000m 1.00001m)  
V1_ NS1_ 0 PULSE(0 5 0.5000m 20n 20n 0.5000m 1.00001m)  
V_0 NS_0 0 PULSE(0 5 0.0000m 20n 20n 0.2500m 0.50000m)  
V_1 NS_1 0 PULSE(0 5 0.2500m 20n 20n 0.2500m 0.50000m)  
XAO V AON ref1 0 Vo ADA4622  
X3V ref1 ref1 ref1 ref1 ref1 N3V1 N3V2 N3V2 N3V3 N3V3 0 N3V4  
LT6658-1.8  
C1 N3V1 0 1u  
C2 V 0 1u  
C3 V 0 1u  
R3V1 V N3V3 0.01  
R3V2 V N3V2 0.03  
C4 ref1 0 0.1u  
.lib opamp.sub  
.tran 0 10ms 0 10u UIC  
.option gmin=1e-12  
.lib ADA4622.lib  
.lib ADG852.sub  
.lib LT6658.lib  
.backanno  
.end
```

Ambas netlist siguen el mismo patrón, en la primera línea aparece el nombre de la fuente de voltaje, así como el nombre del nudo al que va conectada y el valor. Seguidamente podemos ver la matriz de resistencias descrita cada una por el nombre y los puntos a los que va conectada donde NF0 hace referencia a la fila 0, NC0 hace referencia a la columna 0 y así sucesivamente, por último, aparece el valor. La última resistencia es la que se conecta en el AO uniendo la patilla negativa del componente con la salida Vo. Los subcircuitos XU hacen referencia a los switches y las fuentes V0_, V1_ etc. hacen referencia a los voltajes de control de dichos switches. La línea de X3V está relacionada con la referencia de tensión, por lo que en ella aparecen los nudos y conexiones que presenta seguido por los condensadores y resistencias siguientes que también están relacionados con las conexiones en la referencia de tensión. Finalmente aparecen los dots necesarios, de los componentes y de la simulación. Ambos códigos presentan diferencias que cabe destacar, como son los diferentes valores de referencia de tensión, siendo en el modelo 1



3V y en el modelo 2 1.8V, así como utilización de `.gmin` en el modelo 2 que introduce una pequeña conductancia en paralelo con todas las conexiones a tierra para mejorar la convergencia.

7.2.2 Ejecución, validación y resultados

Para llevar a cabo la validación, se han creado ficheros cir de ambos modelos con la matriz 2x2 que se ha utilizado para crear el diseño, comparando así los resultados con los obtenidos con anterioridad en las simulaciones del circuito esquemático, por lo que la matriz será:

```
rarray = np.array([[ 100, 300],[200,400]])
```

Modelo 1:

Se llevarán a cabo dos comprobaciones.

En la primera se elegirá una columna y una fila para comprobar que se obtiene la salida esperada, por lo que la función quedará definida de la siguiente manera.

```
s = creation_cir(rarray, 'ADA4622', 'ADG852', 'LT6658', 0, 0,  
10, 'Cir_array_LTSpice.cir')
```

Por lo que la fila y columna seleccionada es la 0, correspondiendo con la resistencia de 100Ω .

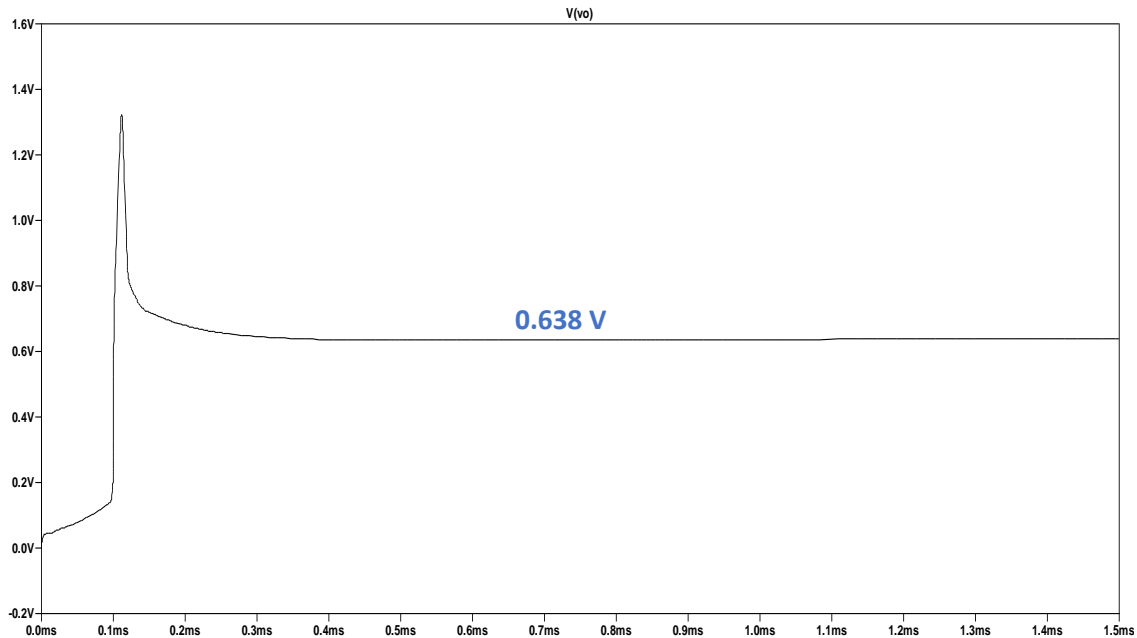


Ilustración 35: V_o , resistencia de 100Ω , netlist, modelo 1.

Como se aprecia en la ilustración 35, se obtiene el resultado esperado.



En la segunda se elegirá una columna y una fila para comprobar que se obtiene la salida esperada, por lo que la función quedará definida de la siguiente manera.

```
s = creation_cir(rarray, 'ADA4622', 'ADG852', 'LT6658', None, None,  
10, 'Cir_array_LTSpice.cir')
```

Obteniendo la gráfica de la ilustración 36, para la salida de la tensión Vo.

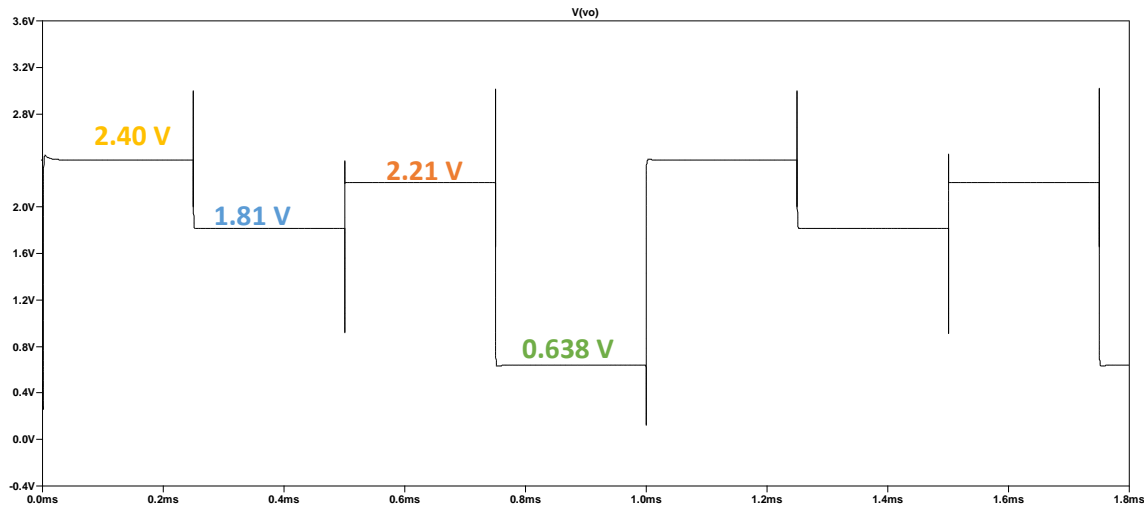


Ilustración 36: Vo, netlist, modelo 1.

Por lo que en la ilustración 36, se observa que en efecto se consiguen los resultados esperados, obteniendo la misma gráfica que se obtenía con el esquemático, corroborando así el programa creado.

Modelo 2:

Se llevarán a cabo dos comprobaciones, al igual que se ha realizado con el modelo 1.

En la primera se elegirá una columna y una fila para comprobar que se obtiene la salida esperada, por lo que la función quedará definida de la siguiente manera.

```
s = creation_cir(rarray, 'ADA4622', 'ADG852', 'LT6658', 0, 0,  
10, 'Cir_array_LTSpice.cir')
```

Por lo que la fila y columna seleccionada es la 0, correspondiendo con la resistencia de 100Ω .

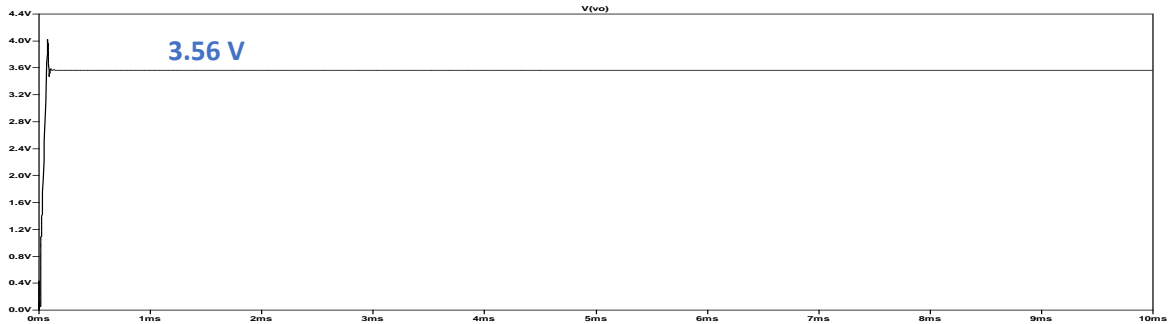


Ilustración 37: Vo, resistencia de 100Ω, netlist, modelo 2.

Como se aprecia en la ilustración 37 se obtiene el resultado esperado.

En la segunda se elegirá una columna y una fila para comprobar que se obtiene la salida esperada, por lo que la función quedará definida de la siguiente manera.

```
s = creation_cir(rarray, 'ADA4622', 'ADG852', 'LT6658', None, None,  
10, 'Cir_array_LTSpice.cir')
```

Obteniendo la gráfica de la ilustración 38 para la salida de la tensión Vo.

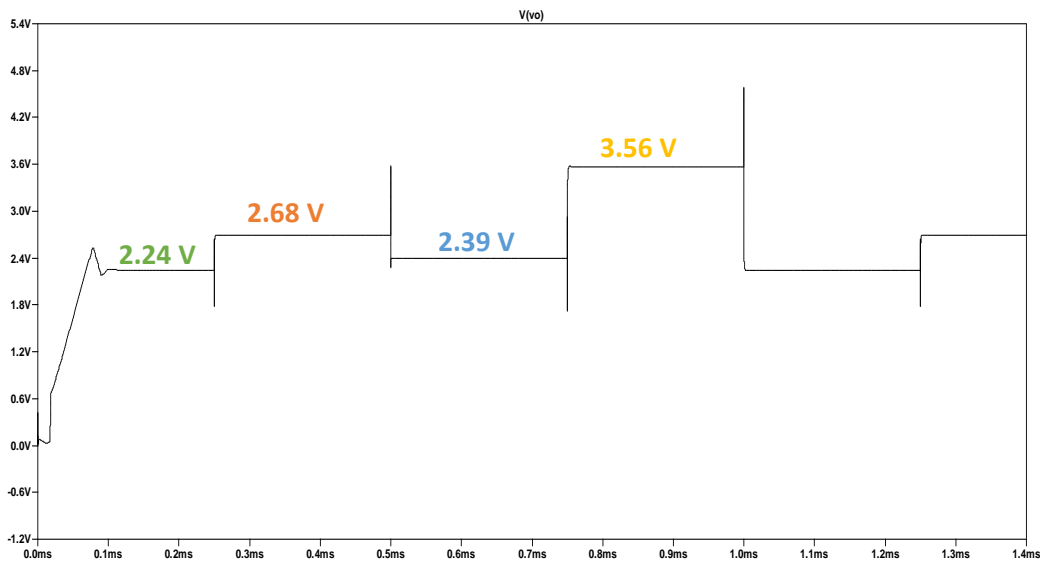


Ilustración 38: Vo, netlist, modelo 2.

Por lo que en la ilustración 38, se observa que en efecto se consiguen los resultados esperados, obteniendo la misma gráfica que se obtenía con el esquemático, corroborando así el programa creado.



8 Resultados y Discusión

8.1 Calculo de errores cometidos con la implementación del diseño

Se llevará en calculo de error producido en cada una de las resistencias de la matriz 2x2 en ambos modelos.

Modelo 1:

Valor real (Ω)	Valor calculado (Ω)	Error (Ω)	Error (%)
300	301.13	1.13	0.38
100	101.39	1.39	1.39
400	401.34	1.34	0.34
200	201.17	1.17	0.59

Tabla 6: Error para cada resistencia, modelo 1.

Presentando en todos los casos un error no superior al 1.5%, por lo que sigue siendo una alta precisión.

Modelo 2:

Valor real (Ω)	Valor calculado (Ω)	Error (Ω)	Error (%)
300	304.05	4.05	1.35
100	101.81	1.81	1.81
400	403.59	3.59	0.89
200	202.93	2.93	1.47

Tabla 7: Error para cada resistencia, modelo 2.

Las tablas presentan en todos los casos un error no superior al 2%, aunque es bueno este modelo presenta una menor precisión en comparación con el modelo 1.



(Véase con mayor detalle en el Anexo III: Cálculo del error en el modelo 1 y en el modelo 2).

8.2 Comparación de ambos modelos

El modelo 1 presenta un error menor en las 4 resistencias analizadas. En la respuesta temporal obtenida en los apartados anteriores, se observa que el modelo 1 presenta un tiempo de respuesta algo mayor que el modelo 2., debido a que en el modelo 2 se ha colocado un transistor NPN en la salida del AO. Una vez pasado dicho tiempo, la respuesta se repite en cada una de las fases de selección de las resistencias. No obstante, este tiempo es bajo, menor de 1 ms y no parece relevante en el sistema de medida. Corresponde simplemente a un transitorio al conectar la alimentación del sistema.

8.3 Análisis de los valores obtenidos en mediciones reales

Debido a que este trabajo está promovido por la necesidad de diseñar un circuito de adquisición de datos mejorado para una matriz de sensores de presión piezoresistivos diseñada en el grupo de investigación EduQTech de la EUPT, se han probado los diseños también con datos proporcionados por el director del trabajo. La matriz está configurada por 16x16 celdas distribuidas en una superficie de 32x32 cm², se han tomado dos medidas de dicha configuración, las cuales han sido guardadas con el nombre de ralex1 y ralex2.

Analizando dichas matrices se observan varias celdas de interés, que cubren un rango de valores amplio de resistencias, los cuales se van a analizar con ambos modelos, calculando así el valor obtenido para dichas resistencias.

Ralex1

Posición	Valor real (Ω)	Valor obtenido modelo 1 (Ω)	Valor obtenido modelo 2 (Ω)	Error modelo 1 (Ω)	Error modelo 2 (Ω)
[13,4]	374.86	378.75	379.54	3.89	4.68
[3,4]	1121.17	1126.71	1130.44	5.54	9.27
[3,11]	19858.99	19672.13	20454.55	186.86	595.56

Ralex2

Posición	Valor real (Ω)	Valor obtenido modelo 1	Valor obtenido modelo 2	Error modelo 1 (Ω)	Error modelo 2 (Ω)
[12, 4]	481.14	485.12	486.11	3.98	4.97
[11, 12]	980.98	985.05	987.44	4.07	6.46
[4, 9]	6059.75	6080.57	6139.15	20.82	79.4

Tabla 8: Error para resistencias relevantes de ralex1 y ralex2, para ambos modelos.

(Véase los cálculos con mayor detalle en el Anexo VI: Cálculo del error en matrices reales).

Dichas matrices han sido representadas de manera gráfica, por lo que se puede observar aproximadamente la presión ejercida en cada punto, siendo posible ya que en las ilustraciones aparece mostrada la conductancia (inverso de la resistencia). Se tomaron con una persona de pie sobre la esterilla, mientras se balanceaba, observándose la forma de los pies.

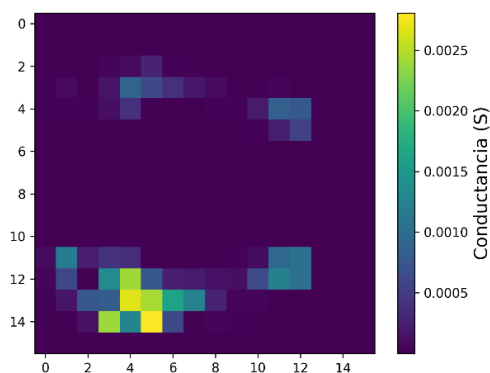


Ilustración 39: Representación conductancia matriz ralex1.

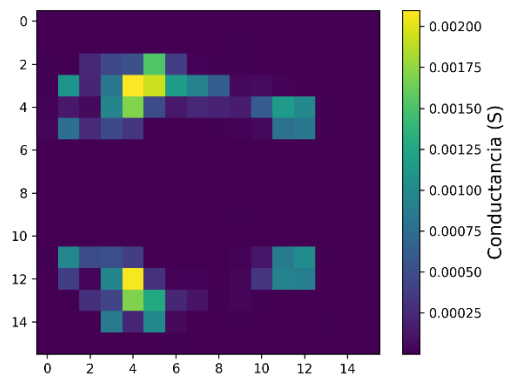


Ilustración 40: Representación conductancia matriz ralex2.



9 Conclusiones

9.1 Impacto del trabajo realizado

El diseño y desarrollo de este proyecto representan un avance en la evolución de sistemas de medición precisos y eficientes en matrices de sensores resistivos, lo que refuerza la relevancia y aplicabilidad del trabajo realizado. Durante el desarrollo, se han enfrentado y superado numerosos desafíos técnicos y de diseño, que incluyen la transición de un circuito ideal a uno real, la selección adecuada de componentes, y la resolución de inconvenientes que surgieron en la implementación del circuito. Cada uno de estos problemas ha sido abordado de manera meticulosa, asegurando que se cumplieran los requisitos específicos del proyecto y que se mantuviera la integridad y funcionalidad del diseño final.

La creación de dos modelos de circuito distintos, junto con la capacidad de simular matrices de sensores de tamaño indefinido mediante la automatización por Python, ha sido una de las contribuciones más significativas del proyecto. Este enfoque no solo ha mejorado la eficiencia del proceso de diseño, sino que también ha optimizado el proceso de simulación. La automatización en la generación de netlists no solo permite un ahorro significativo de tiempo, sino que también reduce la probabilidad de errores humanos, lo que es crucial para asegurar que las simulaciones reflejen con precisión el comportamiento real del circuito bajo diversas condiciones. Este nivel de precisión es esencial para validar el diseño antes de proceder con su implementación física.

Otro aspecto destacado del proyecto es la elección de componentes utilizados en ambos modelos. La selección cuidadosa de componentes estándar y fácilmente accesibles asegura que el circuito pueda ser implementado físicamente sin la necesidad de recurrir a componentes especializados o difíciles de adquirir. Esto hace que el diseño sea no solo teóricamente viable, sino también práctico y realizable en aplicaciones reales. Al utilizar componentes convencionales como switches, amplificadores operacionales y fuentes de alimentación de 5V, el circuito no solo cumple con los requisitos técnicos, sino que también es económicamente accesible y fácilmente replicable, lo que abre la puerta a su implementación en diversas aplicaciones industriales, científicas y comerciales. En el caso de la esterilla sensible a la presión realizada en EduQTech, los circuitos diseñados permiten medir resistencias menores de 1 K, las que acumulan la mayor parte de la presión ejercida, con un error menor del 2%.

En resumen, el proyecto ha logrado no solo cumplir con los objetivos planteados inicialmente, sino también superar los desafíos asociados con el diseño e implementación de un circuito de adquisición de datos robusto y eficiente. La combinación de un diseño modular, la automatización de simulaciones y la utilización de componentes estándar garantiza que el sistema no solo es teóricamente sólido, sino también práctico y preparado para su aplicación en entornos reales, donde puede contribuir significativamente a la mejora de los sistemas de medición y monitoreo.



9.2 Sugerencias para futuros trabajos

A pesar de haber cumplido con los objetivos principales, el desarrollo futuro en esta área ofrece un amplio margen de mejora y optimización. Un aspecto clave que podría abordarse es la reducción del consumo de energía, especialmente considerando que los switches y la fuente de alimentación de 5V podrían verse optimizados para operar de manera más eficiente en aplicaciones de larga duración. Este avance permitiría que el circuito mantuviera su rendimiento durante períodos prolongados sin comprometer la autonomía del sistema. De esta forma se podría considerar la alimentación con baterías para hacer el instrumento portátil. Además, la integración de microcontroladores en ambos modelos podría ser un paso significativo, ya que no solo aumentaría la capacidad de análisis de datos. Estos sistemas, al estar integrados en la programación, maximizarían la precisión de los datos obtenidos, llevando el diseño a un nuevo nivel de sofisticación. En esta misma línea, la implementación de protocolos de comunicación avanzados como SPI abriría la puerta a la integración del sistema en entornos de Internet of Things (IoT), expandiendo su aplicabilidad y conectividad.

Por otro lado, la validación del circuito, hasta ahora realizada mediante simulaciones con componentes ideales, podría evolucionar hacia la construcción de un prototipo físico. Este paso permitiría evaluar el desempeño del circuito en condiciones reales, proporcionando una visión más precisa de su viabilidad y eficiencia en aplicaciones prácticas, y eventualmente podría conducir a su implementación comercial. Además, aunque los modelos actuales están diseñados específicamente para matrices de sensores resistivos, existe la posibilidad de explorar el uso de materiales y componentes avanzados en futuros trabajos. Estos materiales podrían mejorar la sensibilidad y durabilidad del sistema, lo que a su vez permitiría una mejor adaptación a las otras mejoras propuestas, creando un diseño más robusto y versátil.

Todas estas sugerencias no solo son complementarias, sino que, al integrarse, tienen el potencial de mejorar significativamente ambos diseños, elevando su rendimiento y aplicabilidad en un amplio espectro de contextos.



10 Referencias

Emilio, M. D. P. (2013). *Data Acquisition Systems: From Fundamentals to Applied Design*. Springer.

- **Capítulo relevante:** Basics of Data Acquisition Systems

Tartagni, M. (2021). *Electronic sensor design: Principles*. Cambridge University Press.

- **Capítulo relevante:** Selected Topics on Mechanical and Thermal Transduction

Martínez Cesteros, J. (2023). *Development of a low cost pressure sensitive mat: Application to stability and posture measurement* (Tesis doctoral). Universidad de Zaragoza.

Rashid, M. H. (2012). *Spice for power electronics and electrical power* (3rd ed.). CRC Press.

Chazallet, S. (2018). *Python 3: Los fundamentos del lenguaje* (2a ed.). Ediciones ENI.

Analog Devices. (n.d.-a). *Analog Devices: búsqueda de dispositivos, AO*.

<https://www.analog.com/en/product-category/single-supply-analog-switches-multiplexers.html#category-detail>

Accedido el 15 de marzo de 2024.

Analog Devices. (n.d.-b). *Analog Devices: búsqueda de dispositivos, switch*.

<https://www.analog.com/en/parametricsearch/12830#/p4683=2:1%20SPDT>

Accedido el 22 de marzo de 2024.

Protoboard. (2013, septiembre 5). *Diodos (5) - Zener*. Arduino y solo Arduino.

<https://soloarduino.blogspot.com/2013/09/diodos-5-zener.html>

Accedido el 23 de junio de 2024.

Rohm. (n.d.). *Rohm: búsqueda de dispositivos, diodo zener*. <https://www.rohm.com/>

Accedido el 5 de abril de 2024.

Analog Devices. (n.d.-c). *Analog Devices: búsqueda de dispositivos, referenciador de tensión*.

<https://www.analog.com/en/parametricsearch/13004#/d=4994|454|s6|s7|s25|s3|s5|4995|5071|4997|5124|5125|4007|2841&sort=454,desc>

Accedido el 10 de mayo de 2024.

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media.

