

فاز ۱ پروژه درس برنامه نویسی شیء گرا

نیم سال دوم ۱۴۰۳-۰۴

دکتر آراسته، دکتر شیرعلی، دکتر هاشمی

نماینده ای بر نرم افزار های
شبیه سازی مدار های الکترونیک



دانشکده مهندسی برق
دانشگاه صنعتی شریف



فهرست مطالب

۴	۱ مفاهیم پایه مدارهای الکتریکی
۴	۲ المان‌های مداری
۴	۳ مفاهیم مداری و قوانین کیرشهف
۴	۴ ساخت و ذخیره‌سازی گراف مدار
۴	۱.۴ مفهوم گراف مدار
۵	۲.۴ ویژگی‌های المان‌های مداری
۵	۳.۴ مدل‌سازی گراف مدار
۵	۴.۴ قالب ذخیره‌سازی هر المان
۵	۵.۴ ساختار پیشنهادی برای مدار
۵	۶.۴ دلایل استفاده از گراف برای مدل‌سازی مدار
۶	۵ تشکیل ماتریس مدار و روش‌های تحلیل آن
۶	۱.۵ مفهوم ماتریس گره‌ها
۶	۲.۵ ماتریس گره‌ها و روش تحلیل MNA
۷	۳.۵ ساخت ماتریس مدار
۷	۴.۵ حل ماتریس مدار
۸	۵.۵ کاربرد ماتریس در تحلیل‌های مختلف
۸	۶.۵ راهنمای کلی پیاده‌سازی در C++
۱۰	۶ کمی ریاضی
۱۰	۷ معادلات دیفرانسیلی در مدارها
۱۰	۱.۷ مثال: مدار RC
۱۰	۸ روش‌های عددی برای حل معادلات
۱۰	۱.۸ معادلات جبری: تحلیل DC
۱۰	۲.۸ معادلات دیفرانسیلی (تحلیل گذرا)
۱۱	۳.۸ معادلات غیرخطی: روش نیوتن-رافسون
۱۳	۹ مثال کار شده: ساخت ماتریس و حل مدار RC
۱۴	۱.۹ بیان مسئله
۱۴	۲.۹ ساخت ماتریس
۱۵	۳.۹ حل دستگاه ماتریسی

۴.۹	خلاصه	۱۵
۱۰	مثال کار شده: ساخت ماتریس و حل مدار RLC	۱۵
۱.۱۰	بیان مسئله	۱۶
۲.۱۰	ساخت ماتریس	۱۶
۳.۱۰	جایگذاری مقادیر عددی	۱۷
۴.۱۰	حل دستگاه ماتریسی	۱۸
۵.۱۰	اختباری: کد پایتون برای حل مسئله	۱۸
۱۱	گام زمانی تطبیقی	۱۹
۱۲	جدول خلاصه	۱۹
۱۳	معرفی المان های مداری	۲۰
۱.۱۳	ورودی گیری المان های مداری: مقاومت (Resistor)	۲۰
۲.۱۳	ورودی گیری المان های مداری: خازن (Capacitor)	۲۲
۳.۱۳	ورودی گیری المان های مداری: سلف (Inductor)	۲۴
۴.۱۳	ورودی گیری المان های مداری: دیود (Diode)	۲۷
۵.۱۳	ورودی گیری المان های مداری: زمین (Ground)	۲۹
۱۴	تغییر نام نودها و برخی دستورات دیگر	۳۰
۱.۱۴	نمایش لیست نود های موجود که با دستور روبرو انجام میشود	۳۰
۲.۱۴	دستور نمایش المان های خاص مانند خازن و مقاومت	۳۰
۳.۱۴	ساختار دستور تغییر نام نود	۳۱
۱۵	دستورات دریافت خروجی در LTSpice	۳۱
۱.۱۵	ساختار کلی دستورات خروجی	۳۱
۲.۱۵	نمایش ولتاژ	۳۲
۳.۱۵	نمایش جریان	۳۲
۴.۱۵	نمایش همزمان چند متغیر	۳۲
۵.۱۵	ساختار کلی پیام های خطا	۳۲
۱۶	فایل ورودی مدار	۳۳
۱۷	منوی فایل	۳۳
۱۸	تعریف منابع مستقل و وابسته	۳۴
۱.۱۸	منابع مستقل	۳۴
۲.۱۸	منبع سینوسی Source: Sinusoidal	۳۵
۳.۱۸	ورودی های ویژه و بخش امتیازی	۳۶
۴.۱۸	*منابع وابسته	۳۶
۱۹	مدیریت خطاها و پیغام های مناسب	۳۸
۱.۱۹	انواع خطاهای احتمالی در مدار	۳۸
۲۰	راهنمایی برای پیاده سازی مدیریت خطا در برنامه	۳۸
۱.۲۰	تحلیل های اختیاری (امتیازی)	۳۹

۱ مفاهیم پایه مدارهای الکتریکی

[پتانسیل الکتریکی] پتانسیل الکتریکی در یک نقطه، میزان انرژی پتانسیلی الکتریکی به ازای واحد بار در آن نقطه است. اختلاف پتانسیل میان دو نقطه که به آن ولتاژ گفته می شود، عامل حرکت بارهای الکتریکی است.

[زمین (گراند)] زمین در مدار، به عنوان مرجع صفر ولتاژ انتخاب می شود. همه ولتاژهای دیگر نسبت به این نقطه سنجیده می شوند.

[جریان الکتریکی] جریان، مقدار عبور بار الکتریکی از یک مقطع در واحد زمان است. جهت جریان به صورت قراردادی در جهت حرکت بار مثبت در نظر گرفته می شود.

[توان الکتریکی] توان الکتریکی بیانگر نرخ انتقال انرژی الکتریکی است و از رابطه زیر به دست می آید:

$$P = V \times I$$

که در آن P توان، V ولتاژ و I جریان الکتریکی است.

۲ المانهای مداری

المانهای اصلی مدارهای الکتریکی عبارتند از:

- **مقاومت (Resistor):** مانعی برای عبور جریان که باعث اتلاف انرژی به صورت گرما می شود.
- **خازن (Capacitor):** ذخیره کننده انرژی در میدان الکتریکی.
- **سلف (Inductor):** ذخیره کننده انرژی در میدان مغناطیسی.
- **منبع ولتاژ (Voltage Source):** تولیدکننده اختلاف پتانسیل الکتریکی ثابت.
- **منبع جریان (Current Source):** تولیدکننده جریان الکتریکی ثابت.

۳ مفاهیم مداری و قوانین کیرشهف

[قانون جریان کیرشهف (KCL)] مجموع جریانهای ورودی به یک گره برابر مجموع جریانهای خروجی از آن گره است:

$$\sum I_{in} = \sum I_{out}$$

[قانون ولتاژ کیرشهف (KVL)] مجموع ولتاژهای یک حلقه ی بسته برابر صفر است:

$$\sum V = 0$$

این قوانین بر اساس اصل بقای بار و انرژی بوده و پایه تحلیل مدارهای الکتریکی می باشند.

۴ ساخت و ذخیره سازی گراف مدار

۱.۴ مفهوم گراف مدار

در تحلیل مدارهای الکتریکی، می توان مدار را به صورت یک گراف مدل کرد. در این مدل:

- **گره ها (Nodes):** نقاط اتصال المانهای مدار به یکدیگر هستند.
 - **یال ها (Edges):** المانهای مدار هستند که دو گره را به هم متصل می کنند.
- بنابراین، هر مدار الکتریکی یک گراف متشکل از مجموعه ای از گره ها و یال ها می باشد.

۲.۴ ویژگی های المان های مداری

هر المان مداری باید اطلاعات زیر را در خود ذخیره کند:

- نوع المان (مقاومت، خازن، سلف، منبع ولتاژ، منبع جریان)
- نام یا شناسه ی یکتا
- شماره ی گره ی اول (Node 1)
- شماره ی گره ی دوم (Node 2)
- مقدار المان (مقدار مقاومت، ظرفیت خازن، اندوکتانس یا مقدار ولتاژ/جریان)

۳.۴ مدل سازی گراف مدار

برای مدل سازی گراف مدار، می توان یک لیست از تمام المان ها تعریف کرد. هر المان به صورت یک شیء از یک کلاس مناسب مدل می شود. به طور خلاصه:

- هر گره با یک شماره مشخص می شود (مثلاً ۰ برای زمین).
- هر یال، دو گره را به هم متصل می کند و اطلاعات ویژگی های المان را شامل می شود.

۴.۴ قالب ذخیره سازی هر المان

اطلاعات هر المان را می توان به صورت زیر ذخیره کرد:

- نوع المان (type): مانند Resistor، Capacitor و غیره
- نام (name): مانند R1، C2، L3
- گره اول (node1): شماره ی اولین گره متصل به المان
- گره دوم (node2): شماره ی دومین گره متصل به المان
- مقدار (value): مثلاً ۱۰۰۰ اهم یا ۵ ولت

به عنوان مثال، تعریف یک مقاومت ۱۰۰۰ اهم بین گره های ۱ و ۲ به صورت زیر خواهد بود:

```
type: Resistor
name: R1
node1: 1
node2: 2
value: 1000
```

۵.۴ ساختار پیشنهادی برای مدار

کل مدار را می توان با دو مجموعه ذخیره کرد:

- لیستی از گره ها (فقط شماره ی گره ها)
- لیستی از المان ها (هر المان شامل اطلاعات فوق)

۶.۴ دلایل استفاده از گراف برای مدل سازی مدار

- نمایش صریح ارتباطات بین اجزای مدار
- تسهیل در تحلیل مدار با استفاده از الگوریتم های عددی (مثل روش گره، روش مش)
- امکان توسعه ی ساده برای مدارهای پیچیده تر

استفاده از مدل گرافی، مبنای تحلیل مدار در نرم افزارهای پیشرفته ای چون SPICE می باشد.

۵ تشکیل ماتریس مدار و روش های تحلیل آن

۱.۵ مفهوم ماتریس گره ها

در تحلیل مدار به روش گره ای، بر اساس قوانین کیرشهف برای جریان و ولتاژ، می توان یک دستگاه معادلات خطی تشکیل داد که رفتار مدار را مدل می کند. این دستگاه معادلات در قالب یک ماتریس مربعی (ماتریس هدایت یا Conductance Matrix) نمایش داده می شود.

هر سطر این ماتریس متناظر با معادله ی جریان-ولتاژ در یک گره، و هر ستون متناظر با تأثیر ولتاژ یک گره خاص بر جریان گره ها است.

۲.۵ ماتریس گره ها و روش تحلیل MNA

در تحلیل مدار، هدف اصلی یافتن ولتاژ گره ها و جریان های شاخه ها است. روش Nodal Analysis بر مبنای قانون کیرشهف برای جریان در گره ها عمل می کند، ولی در حضور منابع ولتاژ، دچار محدودیت می شود. برای حل این مشکل، روش Modified Nodal Analysis (MNA) توسعه یافته است.

در روش MNA:

- علاوه بر ولتاژهای گره ها، جریان های منابع ولتاژ نیز به عنوان متغیرهای مجهول وارد دستگاه معادلات می شوند.
- ماتریس معادلات به صورت زیر ساخته می شود:

$$\begin{bmatrix} G & B \\ C & D \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} J \\ E \end{bmatrix}$$

روش MNA روشی استاندارد در بسیاری از شبیه سازهای مدار مانند SPICE است و به راحتی می تواند المان های وابسته، منابع کنترل شده، و المان های فرکانسی را نیز مدیریت کند.

منابع:

- [Modified Nodal Analysis – Wikipedia](#)
- [Swarthmore College – Modified Nodal Analysis](#)

در آن

- ماتریس G (هدایت بین گره ها):

– ابعاد: $n \times n$

– تشکیل شده از مقاومت ها (یا هدایت ها) بین گره ها

– مقدار روی قطر اصلی برابر با جمع هدایت های متصل به گره

- ماتریس B (اتصال منابع ولتاژ):

– ابعاد: $n \times m$

– ثبت اتصال منابع ولتاژ بین گره ها

– مقدار ۱ یا -۱ بسته به جهت اتصال

- ماتریس C (ترانهادی B):

– ابعاد: $m \times n$

– برابر با B^T

- ماتریس D (ویژگی منابع ولتاژ):

– ابعاد: $m \times m$

– معمولاً ماتریس صفر در منابع ولتاژ مستقل

- بردارهای V و I :

– V : ولتاژ گره‌ها، ابعاد: $n \times 1$

– I : جریان‌های منابع ولتاژ، ابعاد: $m \times 1$

- بردارهای J و E :

– J : جمع منابع جریان مستقل در هر گره، ابعاد: $n \times 1$

– E : مقادیر ولتاژ منابع ولتاژ مستقل، ابعاد: $m \times 1$

ماتریس	ابعاد	طرز ساخت
G	$n \times n$	بر اساس هدایت المان‌های مقاومتی بین گره‌ها
B	$n \times m$	بر اساس اتصال منابع ولتاژ بین گره‌ها
C	$m \times n$	ترانهاده‌ی ماتریس B
D	$m \times m$	معمولاً ماتریس صفر (برای منابع ولتاژ ایده‌آل)
J	$n \times 1$	بردار منابع جریان مستقل روی گره‌ها
E	$m \times 1$	ولتاژ منابع ولتاژ مستقل

جدول ۱: ابعاد و نحوه‌ی ساخت ماتریس‌های MNA

۳.۵ ساخت ماتریس مدار

برای ایجاد ماتریس دستگاه معادلات:

۱. گره‌ها و منابع ولتاژ را شماره‌گذاری کنید.

۲. برای هر المان:

- مقاومت‌ها: افزایش رسانایی بین گره‌های مربوطه.

- منابع جریان: افزودن مقدار جریان به بردار J .

- منابع ولتاژ: افزودن سطر و ستون جدید برای جریان و ولتاژ آن‌ها مطابق با ساختار MNA.

این ماتریس پایه‌ای برای تمامی تحلیل‌های مدار در پروژه خواهد بود.

۴.۵ حل ماتریس مدار

پس از تشکیل دستگاه $Ax = b$ ، باید مجهولات حل شوند. روش‌های اصلی عبارتند از:

۱.۴.۵ روش حذفی گاوس (Gaussian Elimination)

یک تکنیک مستقیم برای حل دستگاه‌های معادلات خطی از طریق کاهش ماتریس به فرم مثلثی بالا و سپس حل معکوس است. این روش به خصوص برای ماتریس‌های کوچک تا متوسط مناسب است.

در زبان C++ می‌توان این روش را با استفاده از حلقه‌های تو در تو برای عملیات صفربندی و سپس یک حلقه‌ی معکوس برای حل اجرا کرد.

منابع:

- [Gaussian Elimination – Wikipedia](#)

- [Gaussian Elimination – Wolfram MathWorld](#)

۲.۴.۵ روش تجزیه LU (LU Decomposition)

در این روش ماتریس A به حاصلضرب دو ماتریس L (مثلثی پایین) و U (مثلثی بالا) تقسیم می‌شود. حل دو مرحله‌ای ابتدا $Ly = b$ و سپس $Ux = y$ را شامل می‌شود.

این روش برای مدارهایی که در آن‌ها نیاز به حل چندباره‌ی دستگاه با تغییر در بردار b وجود دارد (مانند تحلیل پارامتری یا تحلیل زمانی) بسیار مؤثر است.

در زبان C++ معمولاً این روش با تعریف کلاس‌های مخصوص L و U و استفاده از دو فاز حل پیاده‌سازی می‌شود.

منابع:

- [LU Decomposition – Wikipedia](#)
- [LU Decomposition – Wolfram MathWorld](#)

۵.۵ کاربرد ماتریس در تحلیل‌های مختلف

۱.۵.۵ تحلیل زمانی (Transient Analysis)

در این نوع تحلیل:

۱. مدار در بازه‌های زمانی کوچک مدل می‌شود.

۲. المان‌های خازن و سلف با مدل‌های عددی (مانند Backward Euler یا Trapezoidal Rule) به مدل مقاومت-جریان معادل تبدیل می‌شوند.

۳. در هر گام زمانی، ماتریس MNA به‌روزرسانی شده و با یکی از روش‌های حل دستگاه حل می‌شود.

روش‌های عددی پرکاربرد:

- Backward Euler: ساده و پایدار، مناسب مدارهای دارای المان‌های مقاومتی.
- Trapezoidal Rule: دقت بالاتر، مناسب مدارهای با نوسانات سریع.

منابع:

- [Backward Euler Method – Wikipedia](#)
- [Trapezoidal Rule – Wikipedia](#)

۲.۵.۵ تحلیل پارامتری (Parameter Sweep)

در این نوع تحلیل:

۱. یک یا چند پارامتر مشخص (مثل ولتاژ یا مقاومت) در یک بازه تغییر می‌کند.

۲. در هر مقدار جدید پارامتر، دستگاه معادلات به روز شده و حل می‌شود.

روش LU به دلیل ثابت بودن ماتریس A ، برای تحلیل پارامتری بسیار مناسب‌تر از Gaussian است، چرا که می‌توان L و U را یک بار محاسبه کرد و فقط بردار b را تغییر داد.

۶.۵ راهنمای کلی پیاده‌سازی در C++

برای پیاده‌سازی تحلیل مدار در زبان C++ به شیوه‌ی شی‌گرا:

- کلاس‌هایی برای گره‌ها، المان‌ها (مقاومت، منبع ولتاژ، منبع جریان) و ماتریس مدار طراحی کنید.
- یک کلاس مرکزی برای تشکیل دستگاه معادلات MNA داشته باشید.

- کلاس های جداگانه ای برای الگوریتم های حل (Gaussian LU) تعریف کنید.
- برای تحلیل های زمانی و پارامتری، از حلقه های مناسب استفاده و در هر تکرار دستگاه به روز رسانی و حل شود.

۶ کمی ریاضی

در شبیه سازی مدارن مدار مانند LTspice، حل معادلات یک وظیفه حیاتی است. مدارها به دستگاه هایی از معادلات جبری و دیفرانسیلی تبدیل می شوند و روش های عددی برای حل سریع و دقیق آن ها به کار می رود. این درسنامه معرفی می کند:

- چگونگی مدل سازی ریاضی مدارها،
- روش های حل معادلات جبری خطی و غیرخطی،
- روش های عددی برای حل معادلات دیفرانسیلی،
- نحوه ارتباط بین ساخت ماتریس و حل عددی،
- مثال هایی برای توضیح هر مفهوم.

۷ معادلات دیفرانسیلی در مدارها

وقتی خازن و سلف وارد مدار می شوند، معادلات دیفرانسیلی داریم:

• خازن: $i_C = C \frac{dv}{dt}$

• سلف: $v_L = L \frac{di}{dt}$

در نتیجه، دستگاه MNA به معادلات دیفرانسیلی-جبری (DAE) تبدیل می شود.

۱.۷ مثال: مدار RC

$$V_1 \rightarrow R \rightarrow \text{گره ۱} \rightarrow C \rightarrow GND$$

اعمال قانون جریان کیرشهف در گره ۱:

$$\frac{V_1 - V_{\text{گره ۱}}}{R} = C \frac{dV_{\text{گره ۱}}}{dt}$$

این یک معادله دیفرانسیل خطی مرتبه اول است.

۸ روش های عددی برای حل معادلات

۱.۸ معادلات جبری: تحلیل DC

برای حل $A \cdot x = b$:

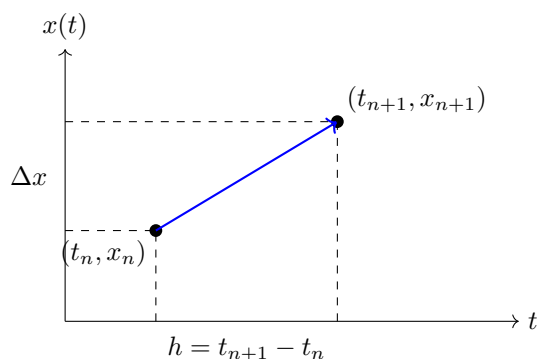
- روش های مستقیم: تجزیه LU، حذف گاوسی،
- روش های تکراری: گرادیان مزدوج، GMRES (برای دستگاه های بسیار بزرگ).

۲.۸ معادلات دیفرانسیلی (تحلیل گذرا)

در تحلیل عددی مسائل زمانی، مشتق ها با استفاده از تقریب های اختلاف محدود جایگزین می شوند. به عنوان مثال، مشتق تابع $x(t)$ در زمان t_n به صورت زیر تقریب زده می شود:

$$\left. \frac{dx}{dt} \right|_{t_n} \approx \frac{x_{n+1} - x_n}{h}$$

که در آن h فاصله زمانی بین دو گام متوالی است.



در این نمودار، شیب خط بین دو نقطه تقریب مشتق تابع در بازه زمانی h را نشان می دهد

۱.۲.۸ روش اویلر معکوس (مرتبه اول)

با داشتن:

$$\frac{dy}{dt} = f(t, y)$$

تقریب اویلر معکوس:

$$\frac{y_{n+1} - y_n}{h} = f(t_{n+1}, y_{n+1})$$

که نیازمند حل معادله برای y_{n+1} است.

۲.۲.۸ مثال: مدار RC با اویلر معکوس

بازنویسی:

$$\frac{V_1 - V_{\text{گره},n+1}}{R} = C \frac{V_{\text{گره},n+1} - V_{\text{گره},n}}{h}$$

و حل برای $V_{\text{گره},n+1}$ در هر گام زمانی.

۳.۸ معادلات غیرخطی: روش نیوتن-رافسون

وقتی معادلات غیرخطی هستند (مثل مدار دیودی)، از روش نیوتن-رافسون استفاده می شود:

۱.۳.۸ الگوریتم:

توجه: در روش نیوتن-رافسون، معادله $f(x)$ معادله ای بر حسب مجهول ما است که طبق قوانین مدار باید برقرار شود. با داشتن $f(x) = 0$:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

و تکرار تا همگرایی.

۲.۳.۸ مثال: مدار دیودی

مدار به صورت زیر است:

$$V_1 \rightarrow R \rightarrow \text{گره ۱} \rightarrow \text{دیود} \rightarrow (\text{GND}) \text{ زمین}$$

مدل جریان دیود:

$$i = I_S \left(e^{\frac{v_D}{nV_T}} - 1 \right)$$

اعمال قانون جریان کیرشهف (KCL) در گره ۱:

$$\frac{V_1 - V_{\text{گره ۱}}}{R} = I_S \left(e^{\frac{V_{\text{گره ۱}}}{nV_T}} - 1 \right)$$

این معادله غیرخطی است و باید با استفاده از روش نیوتن-رافسون حل شود.

روش حل نیوتن-رافسون:

فرض کنیم:

$$f(V_{\text{گره ۱}}) = \frac{V_1 - V_{\text{گره ۱}}}{R} - I_S \left(e^{\frac{V_{\text{گره ۱}}}{nV_T}} - 1 \right)$$

نیوتن-رافسون می گوید:

$$V_{\text{گره ۱}}^{(k+1)} = V_{\text{گره ۱}}^{(k)} - \frac{f(V_{\text{گره ۱}}^{(k)})}{f'(V_{\text{گره ۱}}^{(k)})}$$

که در آن:

$$f'(V_{\text{گره ۱}}) = -\frac{1}{R} - \frac{I_S}{nV_T} e^{\frac{V_{\text{گره ۱}}}{nV_T}}$$

۳.۳.۸ مقادیر عددی برای حل

فرض می کنیم:

$$V_1 = 5V, \quad R = 1000 \Omega, \quad I_S = 10^{-14} A, \quad V_T = 26 mV, \quad n = 1$$

ابتدا یک مقدار اولیه برای گره V_1 انتخاب می کنیم، مثلاً:

$$V_{\text{گره ۱}}^{(0)} = 0.7V$$

گام اول نیوتن-رافسون:

محاسبه $f(V_{\text{گره ۱}}^{(0)})$ و $f'(V_{\text{گره ۱}}^{(0)})$:

ابتدا:

$$f(0.7) = \frac{5 - 0.7}{1000} - 10^{-14} \left(e^{\frac{0.7}{0.026}} - 1 \right)$$

چون:

$$\frac{0.7}{0.026} \approx 26.92 \Rightarrow e^{26.92} \approx 5.27 \times 10^{11}$$

در نتیجه:

$$I_S(e^{26.92} - 1) \approx 10^{-14} \times 5.27 \times 10^{11} \approx 0.00527 A$$

پس:

$$f(0.7) = \frac{4.3}{1000} - 0.00527 = 0.0043 - 0.00527 = -0.00097 A$$

اکنون مشتق:

$$\begin{aligned} f'(0.7) &= -\frac{1}{1000} - \frac{10^{-14}}{0.026} e^{26.92} \\ f'(0.7) &= -0.001 - \left(\frac{10^{-14}}{0.026} \times 5.27 \times 10^{11} \right) \\ f'(0.7) &\approx -0.001 - 0.2027 = -0.2037 \end{aligned}$$

بنابراین:

$$V_{\text{گره}}^{(1)} = 0.7 - \frac{-0.00097}{-0.2037} \approx 0.7 - 0.00476 = 0.69524 V$$

۴.۳.۸ ادامه روند تکرار

روند تکرار با مقدار جدید $V_{\text{گره}}^{(1)}$ ادامه می یابد تا مقدار $f(V_{\text{گره}})$ به اندازه کافی کوچک شود (مثلاً کمتر از 10^{-6}).

نتیجه:

با چند گام نیوتن-رافسون مقدار $V_{\text{گره}}$ تقریباً به ولتاژ بایاس مستقیم دیود ($\sim 0.7V$) همگرا می شود.

۹ مثال کار شده: ساخت ماتریس و حل مدار RC

در این بخش، روند کامل زیر را گام به گام طی می کنیم:

- طراحی یک مدار ساده، RC،

– ماتریس معرفی شده در این بخش در صورت نبود منابع وابسته کارایی کاملی دارد و نیازی به پیاده سازی ماتریس ۲.۵ ندارد

- ساخت ماتریس MNA و طرف راست دستگاه،
- حل دستگاه برای یک گام زمانی،
- نمایش نتایج عددی صریح.

۱.۹ بیان مسئله

یک مدار ساده RC را در نظر بگیرید:

$$V_1 \rightarrow R \rightarrow \text{گره} \rightarrow C \rightarrow \text{زمین}$$

با پارامترهای زیر:

- مقاومت: $R = 1\text{ k}\Omega = 1000\ \Omega$,

- خازن: $C = 1\ \mu F = 1 \times 10^{-6}\text{ F}$,

- منبع ولتاژ: $V_1 = 5\text{ V}$,

- ولتاژ اولیه خازن: $V_{\text{گره},n} = 0\text{ V}$,

- گام زمانی: $h = 1\ \mu s = 1 \times 10^{-6}\text{ s}$.

از روش اویلر معکوس برای گسسته سازی زمانی استفاده می کنیم.

۲.۹ ساخت ماتریس

ابتدا با اعمال قانون جریان کیرشهف (KCL) در گره ۱ داریم:

$$\frac{V_1 - V_{\text{گره}}}{R} = i_C$$

که جریان خازن به صورت زیر تقریب زده می شود:

$$i_C \approx C \frac{V_{\text{گره},n+1} - V_{\text{گره},n}}{h}$$

در نتیجه معادله گسسته شده به صورت زیر خواهد بود:

$$\frac{V_1 - V_{\text{گره},n+1}}{R} = C \frac{V_{\text{گره},n+1} - V_{\text{گره},n}}{h}$$

با بسط آن داریم:

$$\left(\frac{1}{R} + \frac{C}{h}\right) V_{\text{گره},n+1} = \frac{V_1}{R} + \frac{C}{h} V_{\text{گره},n}$$

۱.۲.۹ فرم ماتریسی

این را می توان به صورت دستگاه ماتریسی نوشت:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

که:

$$\mathbf{x} = V_{\text{گره},n+1}$$

ماتریس \mathbf{A} و بردار \mathbf{b} عبارتند از:

$$\mathbf{A} = \left(\frac{1}{R} + \frac{C}{h} \right)$$

$$\mathbf{b} = \left(\frac{V_1}{R} + \frac{C}{h} V_{\text{گر},n} \right)$$

با جایگذاری مقادیر عددی:

$$\mathbf{A} = \left(\frac{1}{1000} + \frac{1 \times 10^{-6}}{1 \times 10^{-6}} \right) = 1.001 \, S$$

$$\mathbf{b} = \left(\frac{5}{1000} + 1 \times 0 \right) = 0.005 \, A$$

بنابراین:

$$\mathbf{A} = [1.001] \quad , \quad \mathbf{b} = [0.005]$$

۳.۹ حل دستگاه ماتریسی

اکنون دستگاه را حل می‌کنیم:

$$1.001 \times V_{\text{گر},n+1} = 0.005$$

در نتیجه:

$$V_{\text{گر},n+1} = \frac{0.005}{1.001}$$

$$V_{\text{گر},n+1} \approx 4.995 \, mV$$

پس از یک گام زمانی، ولتاژ خازن تقریباً به $4.995 \, mV$ می‌رسد.

۴.۹ خلاصه

در هر گام زمانی:

- ماتریس \mathbf{A} بر اساس اجزای مدار و تقریب‌های عددی ساخته می‌شود،
- بردار سمت راست \mathbf{b} به ولتاژ منابع و مقادیر قبلی بستگی دارد،
- حل $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ ولتاژ جدید گره‌ها را به دست می‌دهد.

۱۰ مثال کار شده: ساخت ماتریس و حل مدار RLC

در این بخش، مراحل زیر را طی می‌کنیم:

- طراحی مدار سری RLC،

– ماتریس معرفی شده در این بخش در صورت نبود منابع وابسته کارایی کاملی دارد و نیازی به پیاده سازی ماتریس ۲.۵ ندارد

- ساخت ماتریس MNA و طرف راست دستگاه،
- حل دستگاه برای یک گام زمانی،
- نمایش نتایج عددی صریح.

۱.۱۰ بیان مسئله

یک مدار سری ساده RLC را در نظر بگیرید:

$$V_1 \rightarrow R \rightarrow L \rightarrow \text{گره} \rightarrow C \rightarrow \text{زمین}$$

با پارامترهای زیر:

• مقاومت: $R = 1\text{ k}\Omega = 1000\ \Omega$,

• سلف: $L = 1\text{ mH} = 1 \times 10^{-3}\text{ H}$,

• خازن: $C = 1\ \mu\text{F} = 1 \times 10^{-6}\text{ F}$,

• منبع ولتاژ: $V_1 = 5\text{ V}$,

• شرایط اولیه:

– ولتاژ خازن: $V_{\text{گره},n} = 0\text{ V}$,

– جریان سلف: $I_{L,n} = 0\text{ A}$.

• گام زمانی: $h = 1\ \mu\text{s} = 1 \times 10^{-6}\text{ s}$.

از روش اویلر معکوس استفاده می‌کنیم.

۲.۱۰ ساخت ماتریس

تعاریف زیر را معرفی می‌کنیم:

• $V_{\text{گره},n+1}$: ولتاژ گره ۱ در زمان $n+1$,

• $I_{L,n+1}$: جریان عبوری از سلف در زمان $n+1$.

اعمال قانون جریان کیرشهف (KCL) در گره ۱:

$$I_L = I_C$$

که:

$$I_C = C \frac{dV_{\text{گره}}}{dt} \approx C \frac{V_{\text{گره},n+1} - V_{\text{گره},n}}{h}$$

اعمال قانون ولتاژ کیرشهف (KVL) دور حلقه:

$$V_1 - RI_L - L \frac{dI_L}{dt} - V_{\text{گره}} = 0$$

تقریب مشتق جریان سلف با اویلر معکوس:

$$\frac{dI_L}{dt} \approx \frac{I_{L,n+1} - I_{L,n}}{h}$$

در نتیجه:

$$V_1 - RI_{L,n+1} - L \frac{I_{L,n+1} - I_{L,n}}{h} - V_{\text{گره},n+1} = 0$$

۱.۲.۱۰ فرم ماتریسی

اکنون سیستم به صورت زیر نوشته می شود:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

که در آن:

$$\mathbf{x} = \begin{bmatrix} V_{\text{گر},n+1} \\ I_{L,n+1} \end{bmatrix}$$

و دستگاه معادلات به صورت:

$$\begin{cases} \frac{C}{h} V_{\text{گر},n+1} - I_{L,n+1} = \frac{C}{h} V_{\text{گر},n} & (KCL) \\ -V_{\text{گر},n+1} + \left(-R - \frac{L}{h}\right) I_{L,n+1} = -V_1 - \frac{L}{h} I_{L,n} & (KVL) \end{cases}$$

در نتیجه:

$$\mathbf{A} = \begin{bmatrix} \frac{C}{h} & -1 \\ -1 & -R - \frac{L}{h} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \frac{C}{h} V_{\text{گر},n} \\ -V_1 - \frac{L}{h} I_{L,n} \end{bmatrix}$$

۳.۱۰ جایگذاری مقادیر عددی

با داشتن:

$$R = 1000 \Omega, \quad L = 1 \times 10^{-3} H, \quad C = 1 \times 10^{-6} F, \quad h = 1 \times 10^{-6} s$$

محاسبه می کنیم:

$$\frac{C}{h} = 1, \quad \frac{L}{h} = 1000$$

در نتیجه:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & -2000 \end{bmatrix}$$

شرایط اولیه:

$$V_{\text{گر},n} = 0 V, \quad I_{L,n} = 0 A, \quad V_1 = 5 V$$

پس:

$$\mathbf{b} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$$

۴.۱۰ حل دستگاه ماتریسی

دستگاه معادلات:

$$\begin{bmatrix} 1 & -1 \\ -1 & -2000 \end{bmatrix} \begin{bmatrix} V_{\text{گر},n+1} \\ I_{L,n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \end{bmatrix}$$

از معادله اول:

$$V_{\text{گر},n+1} = I_{L,n+1}$$

جایگذاری در معادله دوم:

$$-V_{\text{گر},n+1} - 2000V_{\text{گر},n+1} = -5$$

$$-2001V_{\text{گر},n+1} = -5$$

$$V_{\text{گر},n+1} = \frac{5}{2001} \approx 0.00249875 \text{ V}$$

و:

$$I_{L,n+1} = V_{\text{گر},n+1} \approx 0.00249875 \text{ A}$$

۵.۱۰ اختیاری: کد پایتون برای حل مسئله

```

1 import numpy as np
2 R = 1000
3 L = 1e-3
4 C = 1e-6
5 h = 1e-6
6 V1 = 5
7 V_node1_old = 0
8 I_L_old = 0
9 A = np.array([[C/h, -1],
10               [-1, -R - L/h]])
11 b = np.array([C/h * V_node1_old,
12               -V1 - (L/h)*I_L_old])
13 x = np.linalg.solve(A, b)
14 V_node1_new = x[0]
15 I_L_new = x[1]
16 print(f"V_node1 at next timestep: {V_node1_new:.6f} V")
17 print(f"I_L at next timestep: {I_L_new:.6f} A")

```

اجرای این کد خروجی می دهد:

V_node1 at next timestep: 0.002499 V

I_L at next timestep: 0.002499 A

که مطابق با محاسبات دستی ماست.

۱۱ گام زمانی تطبیقی

در شبیه سازها:

- هنگام تغییرات سریع، گام زمانی کاهش می یابد،
- هنگام تغییرات کند، گام زمانی افزایش می یابد.

۱۲ جدول خلاصه

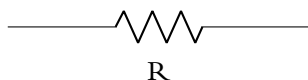
نوع مسئله	فرم معادله	روش حل
معادلات جبری خطی	$A \cdot x = b$	تجزیه LU، حل کننده های پراکنده
معادلات دیفرانسیلی مرتبه اول	$\frac{dy}{dt} = f(t, y)$	اویلر معکوس، ذوزنقه ای
معادلات غیر خطی	$f(x) = 0$	نیوتن-رافسون

۱۳ معرفی المانهای مداری

۱.۱۳ ورودی گیری المانهای مداری: مقاومت (Resistor)

۱.۱.۱۳ فرمت دستور در LTSpice

برای تعریف یک مقاومت در LTSpice از ساختار زیر استفاده می شود:



$$V = I \cdot R$$

```
add R<name> <node1> <node2> <value>
```

در این دستور، مقاومت با نام '<name>' به دو نود '<node1>' و '<node2>' متصل می شود و مقدار آن با '<value>' مشخص می گردد.

۲.۱.۱۳ دستور حذف مقاومت Delete Resistor

برای حذف یک مقاومت از شبیه سازی در LTSpice از دستور زیر استفاده می شود:

```
delete R<name>
```

این دستور باعث می شود که مقاومت با نام '<name>' از شبیه سازی حذف گردد.

۳.۱.۱۳ پارامترها

• **name:** نام یکتا برای مقاومت مثال: R_{load}, R_1

• **node1 node2:** نام نودهای متصل به دو سر مقاومت که باید به مانند LTSpice باشد مثلاً نود 9 میشود: N009

• **value:** مقدار مقاومت با واحد مناسب: $\Omega, k\Omega, M\Omega$

۴.۱.۱۳ مثال صحیح

تعریف یک مقاومت ۲.۲ کیلو اهمی بین نودهای VIN و GND

```
add R1 VIN GND 2.2k
```

۵.۱.۱۳ خطاهای رایج در تعریف مقاومت و رفع آنها

در هنگام تعریف مقاومت در LTSpice ممکن است با خطاهای متنوعی مواجه شوید. در این بخش برخی از مهم ترین خطاها به همراه نمونه کد اشتباه و متن خطا آورده شده است.

۱. **مقادیر نامعتبر** اگر مقدار مقاومت صفر یا منفی باشد، یا به شکلی نادرست نوشته شود، خطای مقدار نامعتبر ایجاد می شود.

خطای نمایش داده شده:

Error: Resistance cannot be zero or negative

نمونه کد اشتباه:

```
add Rshort N001 N002 0
```

راه حل: مقدار مقاومت باید عددی مثبت و غیر صفر باشد. از مقادیر معتبر مانند $1k, 4.7k, 10k$... استفاده کنید.

۲. فرمت نادرست اگر فرمت تعریف مقاومت اشتباه باشد (مثلاً فاصله ها رعایت نشود یا ترتیب پارامترها غلط باشد)، خطای فرمت داده می شود.
خطای نمایش داده شده:

Error: Syntax error

نمونه کد اشتباه:

```
aDdR1N001N0021k
```

راه حل: در تعریف مقاومت باید فاصله مناسب بین نام، نود اول، نود دوم و مقدار رعایت شود. ساختار صحیح به این صورت است:

```
R1 N001 N002 1k
```

۳. عدم وجود المان اگر اشتباهی در نامگذاری یا تعریف المان ها رخ دهد و نرم افزار نتواند المان مورد نظر را شناسایی کند، این خطا رخ می دهد.
البته برای مقاومت که یک المان پایه ای در LTSpice است این خطا کمتر پیش می آید، اما در صورت تایپ اشتباه، ممکن است رخ دهد.
خطای نمایش داده شده:

Error: Element <name> not found in library

نمونه کد اشتباه:

```
add r1 N001 N002 1k
```

راه حل: مطمئن شوید که نام المان با حرف بزرگ شروع شده باشد 'R' برای مقاومت و نامگذاری المان مطابق استاندارد SPICE باشد.

۴. حذف مقاومت ناموجود اگر مقاومت مورد نظر در مدار وجود نداشته باشد و بخواهید آن را حذف کنید، خطای زیر نمایش داده می شود:
خطای نمایش داده شده:

Error: Cannot delete resistor; component not found

نمونه کد اشتباه:

```
delete Runknown
```

راه حل: اطمینان حاصل کنید که مقاومت با نام صحیح در مدار تعریف شده باشد و سپس مجدداً دستور حذف را اجرا کنید.

۵. بررسی تکراری بودن نام مقاومت قبل از تعریف مقاومت، لازم است که مطمئن شوید نام مقاومت تکراری نیست. در صورتی که یک مقاومت با همین نام قبلاً در مدار وجود داشته باشد، باید خطای تکراری بودن نام نمایش داده شود.
خطای نمایش داده شده:

Error: Resistor <name> already exists in the circuit

راه حل: قبل از تعریف هر مقاومت جدید، باید بررسی کنید که آیا مقاومتی با همان نام در مدار وجود دارد یا خیر. اگر مقاومت با همین نام وجود داشته باشد، دستور تعریف مقاومت جدید اجرا نشود و پیامی مبنی بر تکراری بودن نام نمایش داده شود.

۶.۱.۱۳ نکات مهم

• واحدها: حتماً از واحدهای معتبر استفاده کنید:

— Ω یا بدون واحد (پیش فرض)

— k یا kΩ (کیلوهم)

— MΩ یا Meg (مگا هم)

- **نماد علمی:** استفاده از نمادگذاری علمی مجاز است مثال: 2e3 برای ۲۰۰۰Ω

۷.۱.۱۳ نمونه مدار کامل

مدار تقسیم ولتاژ با جزئیات کامل:

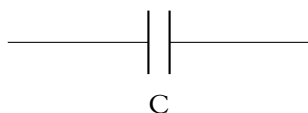


```
add Rupper N001 N002 10k
add Rlower N002 N003 5k
```

۲.۱۳ ورودی گیری المان های مداری: خازن (Capacitor)

۱.۲.۱۳ فرمت دستور در LTSpice

برای تعریف یک خازن در LTSpice از ساختار زیر استفاده می شود:



$$I(t) = C \frac{dV(t)}{dt}$$

```
add C<name> <node1> <node2> <value>
```

در این دستور، خازن با نام '<name>' به دو نود '<node1>' و '<node2>' متصل می شود و مقدار آن با '<value>' مشخص می گردد.

۲.۲.۱۳ دستور حذف خازن Delete Capacitor

برای حذف یک خازن از شبیه سازی در LTSpice از دستور زیر استفاده می شود:

```
delete C<name>
```

این دستور باعث می شود که خازن با نام '<name>' از شبیه سازی حذف گردد.

۳.۲.۱۳ پارامترها

- **name:** نام یکتا برای خازن مثال: C1, Cfilter

- **node1, node2:** نام نودهای متصل به دو سر خازن که باید مشابه با LTSpice باشند. به عنوان مثال، نود 3 می شود: N003

- **value:** مقدار خازن با واحد مناسب: F, μF, nF

۴.۲.۱۳ مثال صحیح

تعریف یک خازن ۱ میکروفارادی بین نودهای VIN و GND

```
add C1 VIN GND 1u
```

۵.۲.۱۳ خطاهای رایج در تعریف خازن و رفع آنها

در هنگام تعریف خازن در LTSpice ممکن است با خطاهای متنوعی مواجه شوید. در این بخش برخی از مهم ترین خطاها به همراه نمونه کد اشتباه و متن خطا آورده شده است.

۱. مقادیر نامعتبر اگر مقدار خازن صفر یا منفی باشد، یا به شکلی نادرست نوشته شود، خطای مقدار نامعتبر ایجاد می شود.

خطای نمایش داده شده:

Error: Capacitance cannot be zero or negative

نمونه کد اشتباه:

```
add C1 N001 N002 0
```

راه حل: مقدار خازن باید عددی مثبت و غیر صفر باشد. از مقادیر معتبر مانند $1u$ ، $4.7u$ ، $10u$ و ... استفاده کنید.

۲. فرمت نادرست اگر فرمت تعریف خازن اشتباه باشد (مثلاً فاصله ها رعایت نشود یا ترتیب پارامترها غلط باشد)، خطای فرمت داده می شود.

خطای نمایش داده شده:

Error: Syntax error

نمونه کد اشتباه:

```
addC1N001N0021u
```

راه حل: در تعریف خازن باید فاصله مناسب بین نام، نود اول، نود دوم و مقدار رعایت شود. ساختار صحیح به این صورت است:

```
add C1 N001 N002 1u
```

۳. عدم وجود المان اگر اشتباهی در نامگذاری یا تعریف المان ها رخ دهد و نرم افزار نتواند المان مورد نظر را شناسایی کند، این خطا رخ می دهد. البته برای خازن که یک المان پایه ای در LTSpice است این خطا کمتر پیش می آید، اما در صورت تایپ اشتباه، ممکن است رخ دهد.

خطای نمایش داده شده:

Error: Element <name> not found in library

نمونه کد اشتباه:

```
add c1 N001 N002 10u
```

راه حل: مطمئن شوید که نام المان با حرف بزرگ شروع شده باشد 'C' برای خازن و نامگذاری المان مطابق استاندارد SPICE باشد.

۴. حذف خازن ناموجود اگر خازن مورد نظر در مدار وجود نداشته باشد و بخواهید آن را حذف کنید، خطای زیر نمایش داده می شود:

خطای نمایش داده شده:

Error: Cannot delete capacitor; component not found

نمونه کد اشتباه:

```
delete Cunknown
```

راه حل: اطمینان حاصل کنید که خازن با نام صحیح در مدار تعریف شده باشد و سپس مجدداً دستور حذف را اجرا کنید.

۵. بررسی تکراری بودن نام خازن قبل از تعریف خازن، لازم است که مطمئن شوید نام خازن تکراری نیست. در صورتی که یک خازن با همین نام قبلاً در مدار وجود داشته باشد، باید خطای تکراری بودن نام نمایش داده شود.

خطای نمایش داده شده:

Error: Capacitor <name> already exists in the circuit

راه حل: قبل از تعریف هر خازن جدید، باید بررسی کنید که آیا خازنی با همان نام در مدار وجود دارد یا خیر. اگر خازن با همین نام وجود داشته باشد، دستور تعریف خازن جدید اجرا نشود و پیامی مبنی بر تکراری بودن نام نمایش داده شود.

۶.۲.۱۳ نکات مهم

• واحدها: حتماً از واحدهای معتبر استفاده کنید:

– F یا بدون واحد (پیش فرض)

– μF یا uF (میکروفاراد)

– nF (نانو فاراد)

• نماد علمی: استفاده از نمادگذاری علمی مجاز است مثال: $1e-6$ برای ۱ میکروفاراد

۷.۲.۱۳ نمونه مدار کامل

مدار فیلتر RC با جزئیات کامل:



```
add C2 N001 N002 10u
```

```
add C1 N002 N003 1u
```

۳.۱۳ ورودی گیری المانهای مداری: سلف (Inductor)

۱.۳.۱۳ فرمت دستور در LTSpice

برای تعریف یک سلف در LTSpice از ساختار زیر استفاده می شود:



$$V(t) = L \frac{dI(t)}{dt}$$

```
add L<name> <node1> <node2> <value>
```

در این دستور، سلف با نام <name>، به دو نود <node1> و <node2> متصل می شود و مقدار آن با <value> مشخص می گردد.

۲.۳.۱۳ دستور حذف سلف Delete Inductor

برای حذف یک سلف از شبیه سازی در LTSpice از دستور زیر استفاده می شود:


```
delete L<name>
```

این دستور باعث می شود که سلف با نام '<name>' از شبیه سازی حذف گردد.

۳.۳.۱۳ پارامترها

• **name**: نام یکتا برای سلف مثال: $Lfilter, L1$

• **node1 node2**: نام نودهای متصل به دو سر سلف که باید مشابه با LTSpice باشند. به عنوان مثال، نود 3 می شود: N003

• **value**: مقدار سلف با واحد مناسب: $mH, \mu H, H$

۴.۳.۱۳ مثال صحیح

تعریف یک سلف ۱ میلی هنری بین نودهای VIN و GND

```
add L1 VIN GND 1m
```

۵.۳.۱۳ خطاهای رایج در تعریف سلف و رفع آنها

در هنگام تعریف سلف در LTSpice ممکن است با خطاهای متنوعی مواجه شوید. در این بخش برخی از مهم ترین خطاها به همراه نمونه کد اشتباه و متن خطا آورده شده است.

۱. **مقادیر نامعتبر** اگر مقدار سلف صفر یا منفی باشد، یا به شکلی نادرست نوشته شود، خطای مقدار نامعتبر ایجاد می شود.

خطای نمایش داده شده:

Error: Inductance cannot be zero or negative

نمونه کد اشتباه:

```
add L1 N001 N002 0
```

راه حل: مقدار سلف باید عددی مثبت و غیر صفر باشد. از مقادیر معتبر مانند $1m, 4.7m, 10m$ و ... استفاده کنید.

۲. **فرمت نادرست** اگر فرمت تعریف سلف اشتباه باشد (مثلاً فاصله ها رعایت نشود یا ترتیب پارامترها غلط باشد)، خطای فرمت داده می شود.

خطای نمایش داده شده:

Error: Syntax error

نمونه کد اشتباه:

```
addL1N001N0021m
```

راه حل: در تعریف سلف باید فاصله مناسب بین نام، نود اول، نود دوم و مقدار رعایت شود. ساختار صحیح به این صورت است:

```
add L1 N001 N002 1m
```

۳. عدم وجود المان اگر اشتباهی در نامگذاری یا تعریف المان‌ها رخ دهد و نرم‌افزار نتواند المان موردنظر را شناسایی کند، این خطا رخ می‌دهد. البته برای سلف که یک المان پایه‌ای در LTSpice است این خطا کمتر پیش می‌آید، اما در صورت تایپ اشتباه، ممکن است رخ دهد. خطای نمایش داده شده:

Error: Element <name> not found in library

نمونه کد اشتباه:

```
add l1 N001 N002 10m
```

راه‌حل: مطمئن شوید که نام المان با حرف بزرگ شروع شده باشد 'L' برای سلف و نامگذاری المان مطابق استاندارد SPICE باشد.

۴. حذف سلف ناموجود اگر سلف مورد نظر در مدار وجود نداشته باشد و بخواهید آن را حذف کنید، خطای زیر نمایش داده می‌شود: خطای نمایش داده شده:

Error: Cannot delete inductor; component not found

نمونه کد اشتباه:

```
delete Lunknown
```

راه‌حل: اطمینان حاصل کنید که سلف با نام صحیح در مدار تعریف شده باشد و سپس مجدداً دستور حذف را اجرا کنید.

۵. بررسی تکراری بودن نام سلف قبل از تعریف سلف، لازم است که مطمئن شوید نام خازن تکراری نیست. در صورتی که یک سلف با همین نام قبلاً در مدار وجود داشته باشد، باید خطای تکراری بودن نام نمایش داده شود. خطای نمایش داده شده:

Error: inductor <name> already exists in the circuit

راه‌حل: قبل از تعریف هر سلف جدید، باید بررسی کنید که آیا سلفی با همان نام در مدار وجود دارد یا خیر. اگر سلف با همین نام وجود داشته باشد، دستور تعریف سلف جدید اجرا نشود و پیامی مبنی بر تکراری بودن نام نمایش داده شود.

۶.۳.۱۳ نکات مهم

• واحدها: حتماً از واحدهای معتبر استفاده کنید:

— H یا بدون واحد (پیش فرض)

— μH (میکروهنری)

— mH (میلی هنری)

• نماد علمی: استفاده از نمادگذاری علمی مجاز است مثال: $1\text{e-}6$ برای ۱ میکروهنری

۷.۳.۱۳ نمونه مدار کامل



```
add L2 N001 N002 10m
```

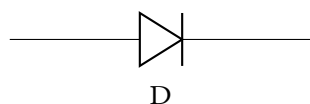
```
add L1 N002 N003 1m
```

۴.۱۳ ورودی گیری المان های مداری: دیود (Diode)

* بخش دیود کاملاً امتیازی است

۱.۴.۱۳ فرمت دستور در LTSpice

برای تعریف یک دیود در LTSpice از ساختار زیر استفاده می شود:



```
add D<name> <node1> <node2> <model>
```

در این دستور، دیود با نام '<name>' به دو نود '<node1>' و '<node2>' متصل می شود و مدل دیود با '<model>' مشخص می گردد.

۲.۴.۱۳ دستور حذف دیود Delete Diode

برای حذف یک دیود از شبیه سازی در LTSpice از دستور زیر استفاده می شود:

```
delete D<name>
```

این دستور باعث می شود که دیود با نام '<name>' از شبیه سازی حذف گردد.

۳.۴.۱۳ پارامترها

• **name:** نام یکتا برای دیود مثال: *Doutput*, *D1*• **node1 node2:** نام نودهای متصل به دو سر دیود که باید مشابه با LTSpice باشند. به عنوان مثال، نود 3 می شود: *N003*• **model:** مدل دیود که معمولاً از پیش در کتابخانه LTSpice تعریف شده است. می تواند به شکل پیش فرض 'D' یا مدل دیود زئر 'Z' باشد.

۴.۴.۱۳ مثال صحیح

تعریف یک دیود استاندارد و دیود زئر:

```
add D1 VIN GND D
add D2 VIN GND Z
```

۵.۴.۱۳ خطاهای رایج در تعریف دیود و رفع آنها

در هنگام تعریف دیود در LTSpice ممکن است با خطاهای متنوعی مواجه شوید. در این بخش برخی از مهم ترین خطاها به همراه نمونه کد اشتباه و متن خطا آورده شده است.

۱. مدل نامعتبر اگر مدل دیود اشتباه یا غیرقابل شناسایی باشد، خطای مدل نامعتبر رخ می دهد.

خطای نمایش داده شده:

Error: Model <model> not found in library

نمونه کد اشتباه:

```
add D1 N001 N002 XYZ
```

۲. فرمت نادرست اگر فرمت تعریف دیود اشتباه باشد (مثلاً فاصله ها رعایت نشود یا ترتیب پارامترها غلط باشد)، خطای فرمت داده می شود.

خطای نمایش داده شده:

Error: Syntax error

نمونه کد اشتباه:

```
add D1N001N002D
```

راه حل: در تعریف دیود باید فاصله مناسب بین نام، نود اول، نود دوم و مدل رعایت شود. ساختار صحیح به این صورت است:

```
add D1 N001 N002 D
```

۳. عدم وجود المان اگر اشتباهی در نامگذاری یا تعریف المان ها رخ دهد و نرم افزار نتواند المان مورد نظر را شناسایی کند، این خطا رخ می دهد.

خطای نمایش داده شده:

Error: Element <name> not found in library

نمونه کد اشتباه:

```
add d1 N001 N002 D
```

راه حل: مطمئن شوید که نام المان با حرف بزرگ شروع شده باشد 'D' برای دیود و نامگذاری المان مطابق استاندارد SPICE باشد.

۴. حذف دیود ناموجود اگر دیود مورد نظر در مدار وجود نداشته باشد و بخواهید آن را حذف کنید، خطای زیر نمایش داده می شود:

خطای نمایش داده شده:

Error: Cannot delete diode; component not found

نمونه کد اشتباه:

```
delete Dunknown
```

راه حل: اطمینان حاصل کنید که دیود با نام صحیح در مدار تعریف شده باشد و سپس مجدداً دستور حذف را اجرا کنید.

۵. بررسی تکراری بودن نام دیود قبل از تعریف دیود، لازم است که مطمئن شوید نام دیود تکراری نیست. در صورتی که یک دیود با همین نام قبلاً در مدار وجود داشته باشد، باید خطای تکراری بودن نام نمایش داده شود.

خطای نمایش داده شده:

Error: diode <name> already exists in the circuit

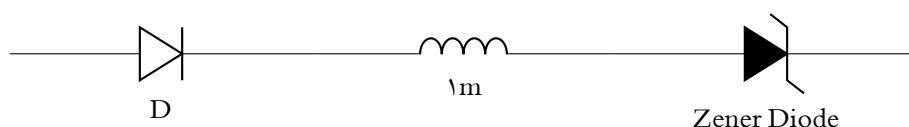
راه حل: قبل از تعریف هر دیود جدید، باید بررسی کنید که آیا دیودی با همان نام در مدار وجود دارد یا خیر. اگر دیود با همین نام وجود داشته باشد، دستور تعریف دیود جدید اجرا نشود و پیامی مبنی بر تکراری بودن نام نمایش داده شود.

۶.۴.۱۳ نکات مهم

- واحدها: نیازی به واحد برای دیود نیست، زیرا دیود یک المان غیر خطی است و تنها به مشخصات مدل وابسته است.

۷.۴.۱۳ نمونه مدار کامل

مدار ساده با دیودهای عادی و زنر:



```
add D1 N001 N002 D
add L1 N002 N003 1m
add D2 N003 N004 Z
```

۵.۱۳ ورودی گیری المان های مداری: زمین (Ground)

۱.۵.۱۳ فرمت دستور در LTSpice

برای تعریف یک زمین در LTSpice از ساختار زیر استفاده می شود:

```
add GND <node>
```

در این دستور، زمین به نود '<node>' متصل می شود. در LTSpice معمولاً زمین به عنوان مرجع ولتاژ استفاده می شود.

۲.۵.۱۳ دستور حذف زمین Delete Ground

برای حذف زمین از شبیه سازی در LTSpice از دستور زیر استفاده می شود:

```
delete GND <node>
```



این دستور باعث می شود که اگر نود '<node>' وجود داشته باشد، زمین حذف شود، اما اگر نود وجود نداشته باشد، پیامی با عنوان روبرو چاپ می شود.

Node does not exist

۳.۵.۱۳ پارامترها

• **node**: نود متصل به زمین. زمین تنها به یک نود از مدار متصل می شود و به عنوان مرجع ولتاژ عمل می کند.

۴.۵.۱۳ مثال صحیح

تعریف یک زمین برای نود N001:

```
add GND N001
```

۵.۵.۱۳ خطاهای رایج در تعریف زمین و رفع آنها

در هنگام تعریف زمین در LTSpice ممکن است با خطاهای متنوعی مواجه شوید. در این بخش برخی از مهم ترین خطاها به همراه نمونه کد اشتباه و متن خطا آورده شده است.

۲. فرمت نادرست اگر فرمت تعریف زمین اشتباه باشد (مثلاً فاصله ها رعایت نشود یا ترتیب پارامترها غلط باشد)، خطای فرمت داده می شود.

خطای نمایش داده شده:

Error: Syntax error

نمونه کد اشتباه:

```
addGNDN001
```

راه حل: در تعریف زمین باید فاصله مناسب بین نود رعایت شود. ساختار صحیح به این صورت است:

```
add GND N001
```

۳. عدم وجود المان اگر اشتباهی در نامگذاری یا تعریف المان ها رخ دهد و نرم افزار نتواند المان مورد نظر را شناسایی کند، این خطا رخ می دهد. خطای نمایش داده شده:

Error: Element <name> not found in library

نمونه کد اشتباه:

```
add gnd N001
```

راه حل: مطمئن شوید که نام المان با حرف بزرگ شروع شده باشد 'GND' برای زمین و نامگذاری المان مطابق استاندارد SPICE باشد.

۶.۵.۱۳ نکات مهم

- ضرورت وجود زمین: در هر مدار الکتریکی باید یک نقطه زمین (مرجع ولتاژ) وجود داشته باشد. بدون زمین، هیچ مرجعی برای اندازه گیری ولتاژ نخواهید داشت و شبیه سازی به درستی انجام نخواهد شد.

۱۴ تغییر نام نودها و برخی دستورات دیگر

۱.۱۴ نمایش لیست نود های موجود که با دستور روبرو انجام میشود

```
.nodes
```

- مثالی از خروجی نمونه:

Available nodes:

n001, n002, VDD, GND, Vout, Vin

۲.۱۴ دستور نمایش المان های خاص مانند خازن و مقاومت

```
.list [component_type]
```

- نحوه خروجی المان ها به ترتیب زمان ورود آن ها است

۱.۲.۱۴ نمایش تمام المان ها

```
.list
```

- نحوه خروجی المان ها به ترتیب زمان ورود آن ها است

۳.۱۴ ساختار دستور تغییر نام نود

```
.rename node <old_name> <new_name>
```

۱.۳.۱۴ پارامترهای دستور

- old_name: نام فعلی نود در مدار
- new_name: نام جدید مورد نظر برای نود

۲.۳.۱۴ مثال اجرا

```
.rename node N001 Vout
```

۳.۳.۱۴ پیام موفقیت آمیز

```
SUCCESS: Node renamed from <old_name> to <new_name>
```

۴.۳.۱۴ خطاهای احتمالی

۵.۳.۱۴ خطای نود ناموجود

```
ERROR: Node <old_name> does not exist in the circuit
```

اقدامات اصلاحی

- بررسی نام نود با دستور .nodes
- اطمینان از صحت حروف بزرگ و کوچک
- بررسی اتصالات مدار

۶.۳.۱۴ خطای نام تکراری

```
ERROR: Node name <new_name> already exists
```

اقدامات اصلاحی

- انتخاب نام جدید برای نود
- بررسی نام‌های موجود با دستور .nodes

۷.۳.۱۴ خطای سینتکسی

```
ERROR: Invalid syntax - correct format:
```

```
.rename node <old_name> <new_name>
```

۱۵ دستورات دریافت خروجی در LTSpice

۱.۱۵ ساختار کلی دستورات خروجی

- لطفا در ابتدا به بخش ۲۰ توجه کنید

ساختار اصلی دستورات نمایش نتایج به صورت زیر است:

```
.print <analysis_type> <variable1> <variable2> ...
```

۲.۱۵ نمایش ولتاژ

برای مشاهده ولتاژ در نقاط مختلف مدار:

```
.print TRAN <Tstep> <Tstop> [<Tstart>] [<Tmaxstep>] V(n001)
.print DC <SourceName> <StartValue> <EndValue> <Increment> V(out)
```

۳.۱۵ نمایش جریان

برای مشاهده جریان المان های مختلف مدار:

```
.print TRAN <Tstep> <Tstop> [<Tstart>] [<Tmaxstep>] I(R1)
.print DC <SourceName> <StartValue> <EndValue> <Increment> I(C1)
```

۴.۱۵ نمایش همزمان چند متغیر

می توانید چندین متغیر را همزمان نمایش دهید:

```
.print TRAN <Tstep> <Tstop> [<Tstart>] [<Tmaxstep>] V(in) V(out) I(R1) I(D1)
```

۵.۱۵ ساختار کلی پیام های خطا

۱.۵.۱۵ خطای نود ناموجود

```
Node <node_name> not found in circuit
```

شرح خطا این خطا زمانی رخ می دهد که نود مورد نظر در شماتیک مدار وجود نداشته باشد.

اقدامات اصلاحی

- بررسی صحت نام نود وارد شده
- بررسی اتصالات مدار
- استفاده از دستور nodes. برای مشاهده لیست نودهای موجود

۲.۵.۱۵ خطای المان ناموجود

```
Component <component_name> not found in circuit
```

شرح خطا این خطا زمانی ظاهر می شود که المان مشخص شده در مدار وجود نداشته باشد.

اقدامات اصلاحی

- بررسی نام المان وارد شده
- استفاده از دستور list. برای مشاهده المان های موجود
- بررسی صحت اتصالات و نام گذاری المان ها

۳.۵.۱۵ خطای سینتکسی

Syntax error in command

شرح خطا این خطا مربوط به اشتباهات ساختاری در وارد کردن دستور است.

۱۶ فایل ورودی مدار

برای وارد کردن مدار، می توان از یک فایل متنی با فرمت ساده استفاده کرد که در هر خط یک المان مدار را تعریف کند. قالب کلی هر خط:

<نوع> <نام> <گره ۱> <گره ۲> <مقدار>

مثال:

```
V V1 1 0 5
R R1 1 2 1000
R R2 2 0 2000
```

۱۷ منوی فایل

یکی از گزینه های موجود در منوی اصلی، دستور زیر است:

```
-show existing schematics
```

با استفاده از این دستور، کاربر می تواند شماتیک های قبلی موجود خود را باز کرده و مرور کند.

هنگامی که کاربر گزینه `-show schematics existing` را انتخاب می کند، یک منوی جدید با دستور زیر ظاهر می شود:

```
-choose existing schematic:
1-draft1
2-draft2
3-draft3
4-elecphase1
```

همان طور که در بالا نشان داده شده است، این منو تمام شماتیک های موجود را به کاربر نمایش می دهد. سپس کاربر باید یکی از این شماتیک های موجود را بر اساس شماره ی آن انتخاب کند.

در این مرحله، کاربر باید از میان فایل های شماتیک موجود، یکی را انتخاب کند. توجه داشته باشید که کاربر باید شماتیک را با استفاده از شماره ی موجود در فهرست انتخاب نماید.

2

واضح است که باید ورودی های نامناسب مانند ورودی های تعریف نشده، غیر عدد صحیح یا اعداد خارج از محدوده را مدیریت کنیم. در اینجا چند نمونه از ورودی های نامناسب آورده شده است:

```
(بزرگ تر از بیشینه شماره موجود باشد) 25
rreturn
draft1
```

خروجی مربوط به ورودی‌های نامناسب در زیر نشان داده شده است: توجه داشته باشید که برنامه در حالت انتظار باقی می‌ماند تا زمانی که ورودی مناسب از سوی کاربر داده شود.

```
-Error : Inappropriate input
```

نمایش شماتیک به کاربر به صورت **Netlist: Spice** پس از مدیریت ورودی‌های نامناسب، باید Netlist Spice شماتیک انتخاب‌شده را به کاربر نشان دهیم. در اینجا یک نمونه آورده شده است:

```
draft2:
V1 in 0 DC 0 AC 1 SIN(0 1 1k)
R1 in out 1k
C1 out 0 1uF
tran 1ms.0 10ms
.ac dec 10 10 100k
.end
```

پس از ارائه‌ی نت‌لیست، برنامه به مرحله‌ی انتخاب شماتیک بازخواهد گشت.

```
-choose existing schematic:
1-draft1
2-draft2
3-draft3
4-elecphase1
```

دستور وارد کردن فایل جدید

```
NewFile <file_path>
```

پارامترها

• file_path: آدرس کامل فایل شماتیک مورد نظر (با پسوند مناسب)

مثال اجرا

```
>>> NewFile /path/to/design.txt
```

خروج از منو: وقتی منوی فوق در کنسول نمایش داده می‌شود، کاربر علاوه بر انتخاب شماتیک مدنظر خود، قادر خواهد بود به منوی اصلی نیز بازگردد.

```
return
```

۱۸ تعریف منابع مستقل و وابسته

۱.۱۸ منابع مستقل

منبع مستقل، عنصری است که مقدار آن (ولتاژ یا جریان) کاملاً مشخص است و به هیچ متغیر دیگری در مدار وابسته نیست. مقدار این منابع توسط طراح یا کاربر تعیین می‌شود و در تحلیل مدار به عنوان داده‌ی اولیه استفاده می‌شود.

- منبع ولتاژ مستقل **Source): Voltage (Independent** ولتاژی ثابت که مستقل از مدار در دو سر خود ایجاد می‌کند. مثال: یک باتری.

– منبع ولتاژ مستقل:

```
add VoltageSource<Name> <Node1> <Node2> <Value>
```

که در آن:

* Name: نام منبع

* Node1: گره مثبت

* Node2: گره منفی (مثلاً زمین)

* Value: مقدار ولتاژ بر حسب ولت (V)

- منبع جریان مستقل: **Source Current Independent** جریانی ثابت که مستقل از مدار در شاخه‌ی خود برقرار می‌کند.

- منبع جریان مستقل:

```
add CurrentSource<Name> <Node1> <Node2> <Value>
```

که در آن:

– Name: نام منبع

– Node1: گره جریان دهنده

– Node2: گره جریان کشنده

– Value: مقدار جریان بر حسب آمپر (A)

در برنامه: برای مدل‌سازی منابع مستقل می‌توانید کلاس‌هایی مانند VoltageSource و CurrentSource تعریف کنید که مقدار آن‌ها در لحظه‌ی تعریف مشخص می‌شود و بدون توجه به شرایط مدار، ثابت می‌ماند.

۲.۱۸ منبع سینوسی **Source: Sinusoidal**

سیگنالی به فرم $A \sin(2\pi ft)$ که در آن:

- A دامنه‌ی موج (Amplitude)

- f فرکانس موج (Frequency)

```
add V<name> <node+> <node-> SIN(<Voffset> <Vamplitude> <Frequency>)
```

که در آن:

- *Voffset: مقدار DC شیفت

- Vamplitude: دامنه موج

- Frequency: فرکانس موج (Hz)

۳.۱۸ ورودی‌های ویژه و بخش امتیازی

در شبیه‌سازهای واقعی مانند SPICE، عنصری به نام VPULSE تعریف می‌شود که قابلیت ایجاد انواع سیگنال‌های زمانی را با تنظیم پارامترهای مختلف دارد. در این پروژه:

- در صورتی که دانشجو منبعی مشابه VPULSE طراحی کند که بتواند:

- مقادیر اولیه و نهایی ولتاژ/جریان را تعیین کند،
- زمان شروع پالس، زمان شیب صعودی، زمان شیب نزولی و زمان پایداری را مشخص کند،
- تناوب سیگنال را مدیریت کند،

و خروجی‌هایی نظیر موج مربعی، مثلثی، پله‌ای یا حتی ترکیبی تولید نماید، امتیاز ویژه و اضافی برای پروژه وی در نظر گرفته خواهد شد.

add V<name> <node+> <node-> PULSE <type> (<Value>)

که در آن می‌توان انواع موج‌های پله‌ای، مربعی، مثلثی و ترکیبی را تولید کرد.

- Name: نام منبع ولتاژ یا جریان
- Node+: گره مثبت اتصال منبع
- Node-: گره منفی اتصال منبع (مثلاً زمین)
- type: برای اینکه مشخص شود دقیقاً کدام نوع ورودی است
- Value: مقدار ولتاژ اولیه (قبل از شروع پالس) این قسمت قابلیت شخصی سازی برای هرگروه را دارد

۴.۱۸ *منابع وابسته

منبع وابسته، عنصری است که مقدار آن (ولتاژ یا جریان) به یک متغیر دیگر در مدار بستگی دارد. این منابع معمولاً به یکی از ولتاژها یا جریان‌های دیگر مدار متکی هستند.

- منبع ولتاژ وابسته به ولتاژ VCVS: Source Voltage Controlled Voltage – ولتاژی ایجاد می‌کند که متناسب با ولتاژ بین دو گرهی دیگر در مدار است.
- منبع ولتاژ وابسته به جریان CCVS: Source Voltage Controlled Current – ولتاژی ایجاد می‌کند که متناسب با جریان یک شاخه‌ی دیگر در مدار است.
- منبع جریان وابسته به ولتاژ VCCS: Source Current Controlled Voltage – جریانی ایجاد می‌کند که متناسب با ولتاژ بین دو گرهی دیگر است.
- منبع جریان وابسته به جریان CCCS: Source Current Controlled Current – جریانی ایجاد می‌کند که متناسب با جریان یک شاخه‌ی دیگر است.

منبع ولتاژ وابسته به ولتاژ: (VCVS)

add E<Name> <Node1> <Node2> <CtrlNode1> <CtrlNode2> <Gain>

که در آن:

- Name: نام منبع
- Node1: گره مثبت
- Node2: گره منفی (مثلاً زمین)

- CtrlNode1: گره مثبت وابسته
- CtrlNode2: گره منفی وابسته
- Gain: ضریب منبع وابسته

• VCCS:

```
add G<name> <node+> <node-> <control+> <control-> <Gain>
```

که در آن:

- Name: نام منبع شروع با G
- Node+: گره مثبت جریان خروجی
- Node-: گره منفی جریان خروجی
- Control+: گره مثبت ولتاژ کنترل
- Control-: گره منفی ولتاژ کنترل
- Gain: ضریب منبع وابسته

• CCVS:

```
add H<name> <node+> <node-> <vname> <gain>
```

که در آن:

- Name: نام منبع شروع با H
- Node+: گره مثبت ولتاژ خروجی
- Node-: گره منفی ولتاژ خروجی
- Vname: نام منبع ولتاژی که جریانش کنترل کننده است (مثل یک V یا VSOURCE موجود)
- Gain: ضریب تقویت ولتاژ وابسته به جریان

• CCCS:

```
add F<name> <node+> <node-> <vname> <gain>
```

که در آن:

- Name: نام منبع شروع با F
- Node+: گره مثبت جریان خروجی
- Node-: گره منفی جریان خروجی
- Vname: نام منبع ولتاژی که جریانش کنترل کننده است
- Gain: ضریب تقویت جریان خروجی نسبت به جریان کنترلی

در برنامه: برای مدل سازی منابع وابسته، کلاس هایی می توانید تعریف کنید که در آن ها مقدار ولتاژ یا جریان در زمان تحلیل مدار و با توجه به سایر متغیرهای مدار محاسبه می شود. مثلاً کلاس DependentVoltageSource که یک پارامتر کنترل کننده دریافت کند.

۱۹ مدیریت خطاها و پیغام‌های مناسب

در فرآیند تحلیل مدار، ممکن است ورودی‌هایی نامعتبر یا مدارهایی غیرقابل تحلیل دریافت شوند. در چنین مواردی، برنامه باید به درستی خطا را شناسایی کرده و پیغام مناسب و قابل فهمی برای کاربر نمایش دهد. این عمل، به بهبود تجربه‌ی کاربری و افزایش پایداری برنامه کمک می‌کند.

۱.۱۹ انواع خطاهای احتمالی در مدار

- نبودن اتصال زمین (گراند): مدار باید حداقل شامل یک گره زمین باشد. در صورت عدم وجود زمین، مدار قابل تحلیل نخواهد بود.
- مدار ناپیوسته یا چند بخشه: اگر مدار به چند قسمت جداگانه تقسیم شده باشد و همه‌ی گره‌ها به هم متصل نباشند، تحلیل کلی مدار امکان‌پذیر نیست.
- مقداردهی نامعتبر به عناصر: مثلاً مقاومت یا سلف یا خازن دارای مقدار منفی یا صفر غیرمجاز باشند.
- تعریف اشتباه منابع وابسته: منبع وابسته بدون مرجع کنترلی معتبر تعریف شده باشد.
- ورودی ناقص برای منابع متغیر با زمان: همه‌ی پارامترهای لازم برای تعریف منابع سینوسی، کسینوسی، مثلثی یا پالس ارائه نشده باشند.
- تکرار نامناسب نام گره‌ها یا عناصر: دو یا چند عنصر یا گره با نام‌های تکراری که منجر به سردرگمی در تحلیل مدار شوند.

نمونه‌هایی از پیغام‌های خطای پیشنهادی

در صورت بروز خطا، برنامه می‌تواند پیغام‌های زیر را به کاربر نمایش دهد:

Error: No ground node detected in the circuit.
 Error: Negative or zero value for a component is invalid.
 Error: Dependent source has an undefined control element.
 Error: Missing parameters for time-dependent source.
 Error: Duplicate node or component name detected.

نکته مهم: در صورت بروز هرگونه خطا، اجرای برنامه باید به گونه‌ای مدیریت شود که به جای قطع ناگهانی، پیغام مناسب نمایش داده شده و از کاربر درخواست اصلاح ورودی گردد.

۲۰ راهنمایی برای پیاده‌سازی مدیریت خطا در برنامه

در برنامه‌نویسی به زبان سی‌پلاس‌پلاس، مدیریت خطا معمولاً به دو روش انجام می‌شود:

- بررسی شرطها (کنترل منطقی): پیش از انجام هر عملیات حساس مانند اضافه کردن یک عنصر به مدار یا محاسبه‌ی ویژگی‌های آن، باید مقداردهی‌ها و ورودی‌ها بررسی شوند. در صورت غیرمعتبر بودن داده‌ها، برنامه باید با استفاده از دستورات شرطی مناسب، پیغام خطا تولید کند و از ادامه‌ی اجرای بخش نادرست جلوگیری کند.
- استفاده از ساختار پرتاب و دریافت خطا (استثناها): می‌توان با تعریف استثنای (خطاهای) خاص، هنگام مواجهه با شرایط غیرقابل قبول، خطایی را پرتاب کرده و در بخش مناسبی از برنامه آن را دریافت و مدیریت نمود. استفاده از استثناها باعث جداسازی بهتر منطق برنامه‌ی اصلی از مدیریت خطا می‌شود.

منوی شبیه‌سازی و تحلیل مدار

در این پروژه، پس از پیاده‌سازی مدار، کاربران باید بتوانند نتایج تحلیل مدار را مشاهده کنند. انواع تحلیل‌هایی که در این برنامه مورد نیاز است در دو بخش اجباری و اختیاری تقسیم‌بندی شده‌اند.

تحلیل های اجباری

- تحلیل گذرا **Analysis Transient** در این تحلیل، رفتار مدار نسبت به زمان بررسی می شود. کاربران باید بتوانند تغییرات ولتاژ و جریان در گره ها و شاخه های مختلف مدار را در طول زمان مشاهده کنند. این تحلیل برای مدارهایی که منابع متغیر با زمان (مانند سینوسی یا پالسی) دارند اهمیت ویژه ای دارد.

```
.TRAN <Tstep> <Tstop> [<Tstart>] [<Tmaxstep>]
```

- .TRAN : تحلیل گذرا؛ شامل پارامترهای:

– Tstep: گام زمانی نمونه برداری Step Time Sampling

– Tstop: زمان پایان تحلیل Time Stop Simulation

– Tstart (اختیاری): زمان شروع ثبت داده Time Data Saving Start

– Tmaxstep (اختیاری): بیشینه گام زمانی Step Time Maximum

۱.۲۰ تحلیل های اختیاری (امتیازی)

- تحلیل **DC Sweep** در این نوع تحلیل، مقدار یک منبع ولتاژ یا جریان به صورت خطی تغییر داده می شود و پاسخ مدار در طول این تغییرات بررسی می شود. این تحلیل به ویژه برای مشاهده رفتار غیرخطی مدارها یا تعیین نقاط کاری مهم است.

```
.DC <SourceName> <StartValue> <EndValue> <Increment>
```

نکته مهم: پیاده سازی تحلیل های اجباری برای همه ی دانشجویان الزامی است. پیاده سازی تحلیل های اختیاری برای دریافت نمره ی اضافه توصیه می شود.