

## نرم افزار تحلیل مدار با استفاده از تئوری گراف ها

امیرعلی بنائی کاشانی ۹۳۲۳۰۱۰ - بهاره شیرمحمدی ۹۳۲۳۰۴۳

۲۶ تیر ۱۳۹۷

### ۱ مقدمه

پروژه ما قادر است یک مدار RLC و منبع جریان را با استفاده از تئوری گراف تحلیل کرده و ماتریس های مربوطه را در صفحه ای جداگانه نمایش دهد. با کلیک کاربر روی هر کدام از درایه های هر ماتریس، المان مربوط به آن روی مدار با رنگ جداگانه ای مشخص می گردد.

### ۲ طرح GUI

#### ۱.۲ وارد کردن المان های مدار

در گام اول چند PushButton در بخش گرافیکی قرار می دهیم تا کاربر با کلیک بر روی هر کدام از آنها، المان موردنظرش را انتخاب نماید. هر کدام از المان های مدار، R - L - C دارای قابلیت های - Rotate Change value - Delete می باشند.

#### ۲.۲ کلاس المان های مدار

برای هر المان یک کلاس در نظر میگیریم که از QLabel ارث می برد و شامل متغیرهای: گره های اولیه و ثانویه، یک عکس از المان مربوطه که به آن المان map می شود، یک Text Edit که مقدار المان را نمایش می دهد و تابع های: MousePressEvent, MouseMoveEvent برای کلیک بر روی المان ها و جا به جا کردن آن ها، hide-detail, show-detail, delete-me, Rotate که به ترتیب برای چرخاندن جهت المان، حذف کردن المان، مشخص کردن المان در مدار در حالت انتخاب و خروج از حالت انتخاب.

برای Node هم، یک کلاس جداگانه داریم که آن هم از QLabel ارث می برد و یک مربع آبی رنگ به سائز ۱۰ \* ۱۰ می سازد و آن را نمایش می دهد. برای کلاس Main window که circuit gui نام دارد، یک دیکشنری داریم که key آن id گره است و value آن یک set است از گره های متصل به این گره (که در بخش اتصال گره ها به هم توضیح داده می شود) هم چنین یک set دیگر هم داریم به نام Connected-elements

که هر المانی که ساخته می شود به گره های اولیه و ثانویه آن ، این المان را اضافه می کنیم. برای اتصال گره ها به هم ، تابع Connect-Nodes را داریم که اگر کاربر بر روی دکمه wire کلیک کند و به صورت متوالی ، دو گره را انتخاب کند ، برنامه به این تابع می رود و اختلاف  $x,y$  دو گره را حساب می کند و بین آنها دو سیم افقی و عمودی رسم می کند.

### ۳.۲ id دادن به المان ها

در کلاس اصلی Main window یعنی circuit gui توابعی داریم به نام های : add-res, add-cap- add-ind که با کلیک کردن کاربر بر روی دکمه های مقاومت – خازن – سلف ، این توابع فراخوانی شده و یک آبیجکت از این کلاس ها ساخته می شود و به آن ها نام داده می شود و سپس در لیستی به نام elements اضافه می گردند. در مورد گره ، در بخش اتصال گره ها به هم ، عمل id دادن انجام می شود (بسته به اینکه از قبل id داشته اند یا خیر) و دیکشنری node-dict تکمیل می گردد.

## ۳ تحلیل مدار

### ۱.۳ استخراج و محاسبه ماتریس های اساسی

در این بخش به نحوه تحلیل و استخراج ماتریس های اولیه مربوط به روش تحلیل گره می پردازیم. برای این منظور قدم اول استخراج ماتریس تلافی  $A$  می باشد. ماتریس تلافی به تعداد یکی کمتر از گره ها سطر و به تعداد شاخه ها ستون دارد. در تحلیل ما به ازای هر المان یک شاخه در نظر گرفته شده است لذا تعداد ستون های ماتریس تلافی به تعداد المان های مدار خواهد بود. در تابع مربوط به سیم کشی و اتصال نود ها (node) به گونه ای عمل شده که نودهایی که دارای id یکسان اند یک گره مدار را تشکیل می دهند. همچنین در کلاس نود، یک مجموعه به نام connected-elements تعریف شده که مشخص می کند به هر نود چه المان هایی از مدار متصل است. در اتصال نود ها به هم connected-elements آن ها با هم اجتماع گرفته می شود. از طرفی برای هر المان یک نود اولیه و یک نود ثانویه قرار داده شده است. طبق قرارداد، تحلیل به گونه ای صورت می گیرد که جهت جریان در هر شاخه از نود اولیه به نود ثانویه باشد. حال با توجه به توضیحات فوق می توان ماتریس تلافی را به این صورت محاسبه نمود. با استفاده از دو حلقه ی for تو در تو که یکی روی المان ها و دیگری روی گره ها می گردد چک می کنیم که اولاً آیا در مجموعه ی connected-elements نود فعلی، المان فعلی موجود است یا خیر. در صورت موجود نبودن به درایه مربوطه صفر نسبت داده می شود. در صورت موجود بودن چک می شود که نود فعلی برابر نود اولیه یا ثانویه المان فعلی است. در صورتی که اولیه باشد درایه مربوطه در ماتریس برابر ۱ و در صورت ثانویه درایه مربوطه برابر -۱ قرار داده می شود. تا به اینجا ماتریس تلافی محاسبه گشت. لازم به ذکر است که در محاسبه این ماتریس گره مبنا که زمین مدار است دخالتی ندارد. مرحله بعدی محاسبه ماتریس  $J_s$  است. این ماترسی منابع جریان مستقل مدار را مشخص می کند. نحوه محاسبه این ماتریس به این صورت است که تمام المان های مدار را چک می کند و آن هایی که منبع اند را با استفاده از

اسمشان تشخیص می دهد. سپس مقدار جریان آن ها (که یک ویژگی از کلاس منابع است) را در درایه مربوطه داخل ماتریس قرار می دهد. حال به محاسبه ماتریس ادمیتانس قطری که نمادش  $Y_b$  است می پردازیم. این ماتریس، یک ماتریس مربعی و قطری است که ابعاد آن به تعداد شاخه های مدار می باشد. درایه های روی قطر اصلی این ماتریس برابر ادمیتانس شاخه مربوطه می باشد. تا به اینجا ماتریس هایی که باید از روی توپولوژی مدار به دست می آمدند تشکیل شدند. حال برای محاسبه سایر ماتریس ها (که مربوط به ولتاژ و جریان گره ها و شاخه ها می شوند) باید از فرمول های مربوطه استفاده کنیم. فرمول های اساسی به قرار زیر می باشند:

$$Y = A.Y_b.A^T \quad (۱)$$

$$I_s = -A.J_s \quad (۲)$$

$$e = Y^{-1}.I_s \quad (۳)$$

$$V = A^T.e \quad (۴)$$

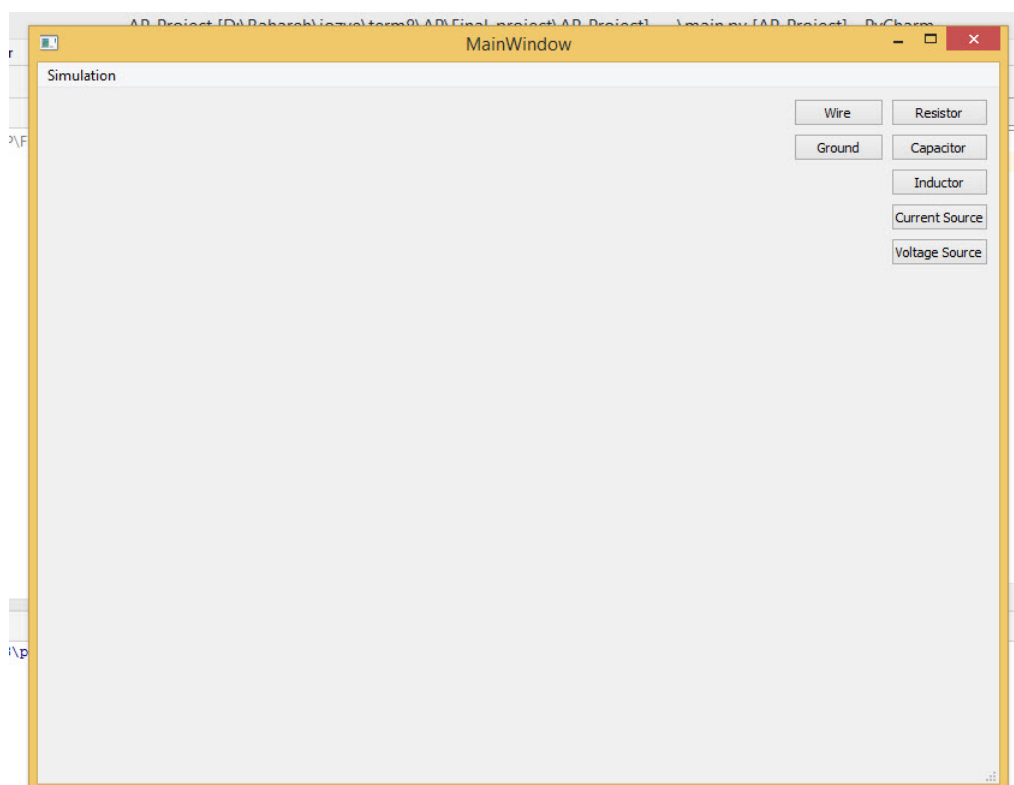
$$J = Y_b.V + J_s \quad (۵)$$

با استفاده از فرمول های فوق و ماژول numpy به راحتی تمامی ماتریس های مدار محاسبه می گردند.

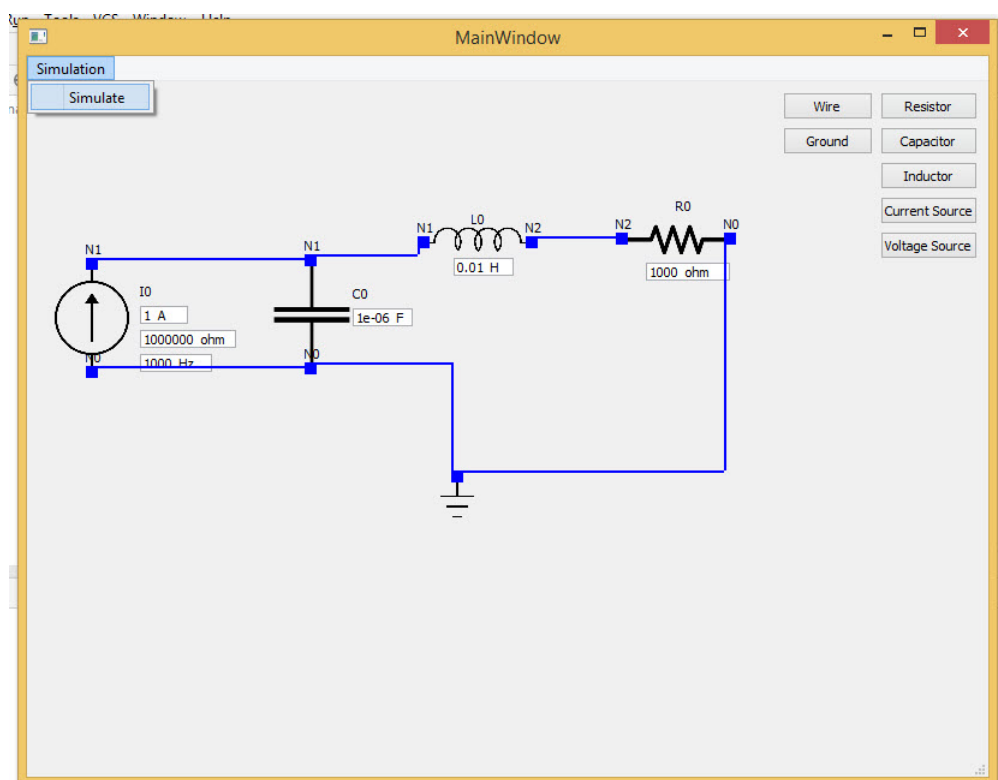
## ۲.۳ نمایش ماتریس ها

تا این قسمت تمامی متغیر های مدار با استفاده از ماتریس های بیان شده در بخش قبل محاسبه گشتند. اما این هدف پروژه نیست. حال باید ترتیبی داده شود تا مشخص گردد هر درایه از هر ماتریس مربوط به کدام قسمت از مدار است. برای تحلیل مدار باید از نوار بالای صفحه گزینه simulate انتخاب شود. با کلیک بر این گزینه صفحه جدیدی باز می شود که سمت راست آن کلید هایی قرار داده شده برای نمایش هر ماتریس. با کلیک کردن بر هریک از این ماتریس ها، ماتریس مربوطه در صفحه باز شده نمایش داده می شود. اتفاقی که در کد می افتد این است که به ازای هر درایه از ماتریس ها یک Qlabel ساخته می شود. سپس مقدار درایه داخل آن نوشته می شود و برای آن style sheet مناسب قرار داده می شود. برای صفحه جدیدی که باز می شود یک کلاس تعریف شده است. در این کلاس تابع MousePressEvent به گونه ای تعریف شده که مختصات نقطه ای از صفحه که بر آن جا کلیک می شود به دست بیاید. حال با استفاده از این مختصات و با توجه به این که در حال نمایش کدام ماتریس هستیم (با استفاده از یک فرمول) می توان به دست آورد که کدام درایه از ماتریس انتخاب شده است. پس فهمیدن موضوع فوق رنگ درایه مربوطه به آبی تغییر می کند و رنگ سایر

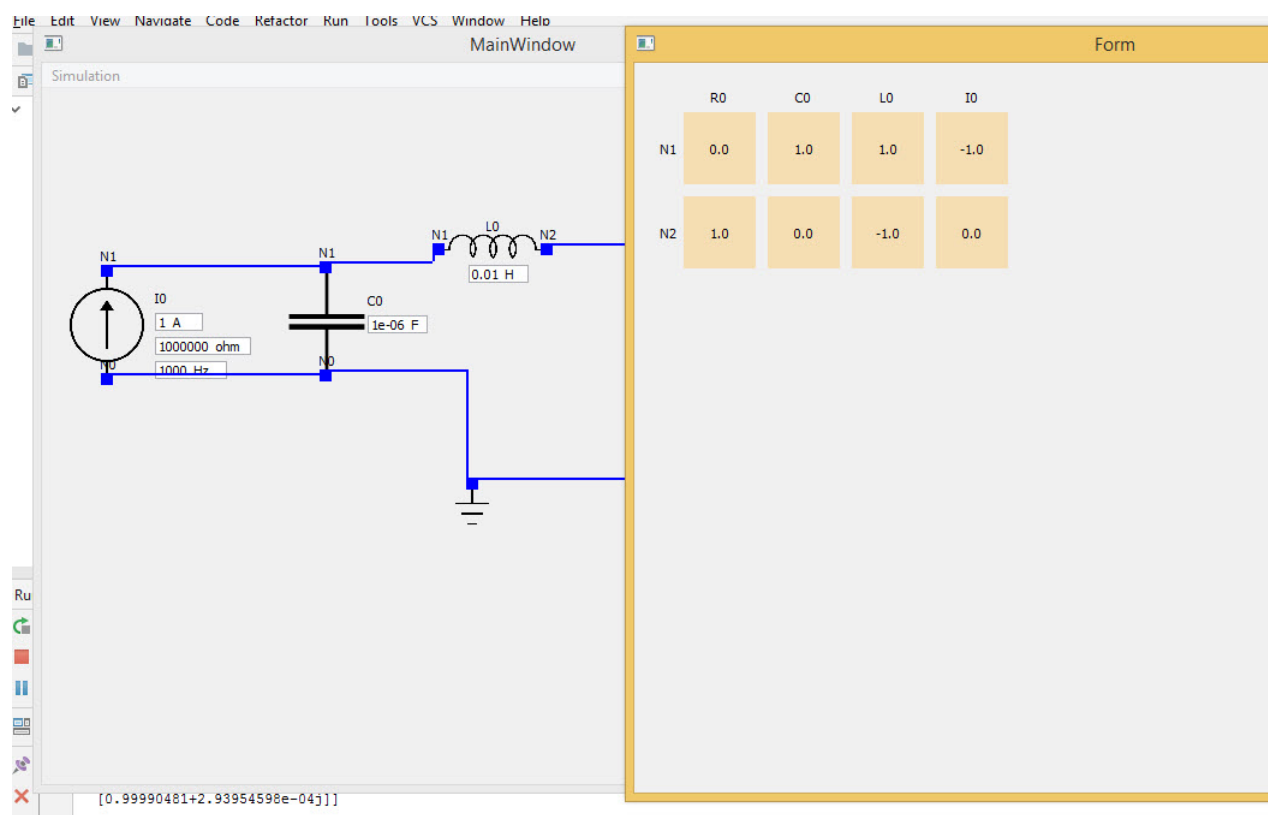
درایه ها به حالت قبلی باز می گردد. حال باید بخش مربوطه در مدار با توجه به این که کدام درایه از کدام ماتریس انتخاب شده است تغییر کند. برای این منظور اگر درایه انتخاب شده نیاز باشد تا گره ای را مشخص کند، تمامی نود ها با id مربوطه به رنگ قرمز در می آیند و سایر نود ها رنگشان آبی (که همان رنگ پیش فرض نود ها است) می شود. حال اگر نیاز باشد المانی در مدار مشخص گردد، تابع show-detail برای المان مربوطه فراخوانی می شود و سایر المان هایی که نیاز به مشخص شدن ندارند تابع hide-detail شان فراخوانی می شود. در اینجا ذکر توضیحاتی در رابطه با توابع فوق خالی از لطف نیست. کاری که تابع show-detail می کند این است که اولاً پس زمینه المان را به رنگ زرد در می آورد. ثانیاً یک فلش از نود اولیه به نود ثانویه آن رسم می کند تا جهت جریان آن مشخص شود. تابع hide-detail عکس این اعمال را انجام می دهد. یعنی رنگ پس زمینه المان و فلش آن را حذف می کند. لازم به ذکر است که چون نماد منبع جریان حاکی از جهت جریان آن است، هنگام فراخوانی show-detail آن، دیگر فلش مجزایی برای مشخص کردن منبع جریان قرار داده نمی شود. نکته دیگری که حائز اهمیت است این است که برای نمایش هر ماتریس باید ماتریس قبلی از روی صفحه پاک شود. برای این مقصود یک تابع clear برای کلاس نمایش ماتریس ها قرار داده شده که قبل از نمایش هر ماتریس فراخوانی می شود و صفحه را پاک می کند



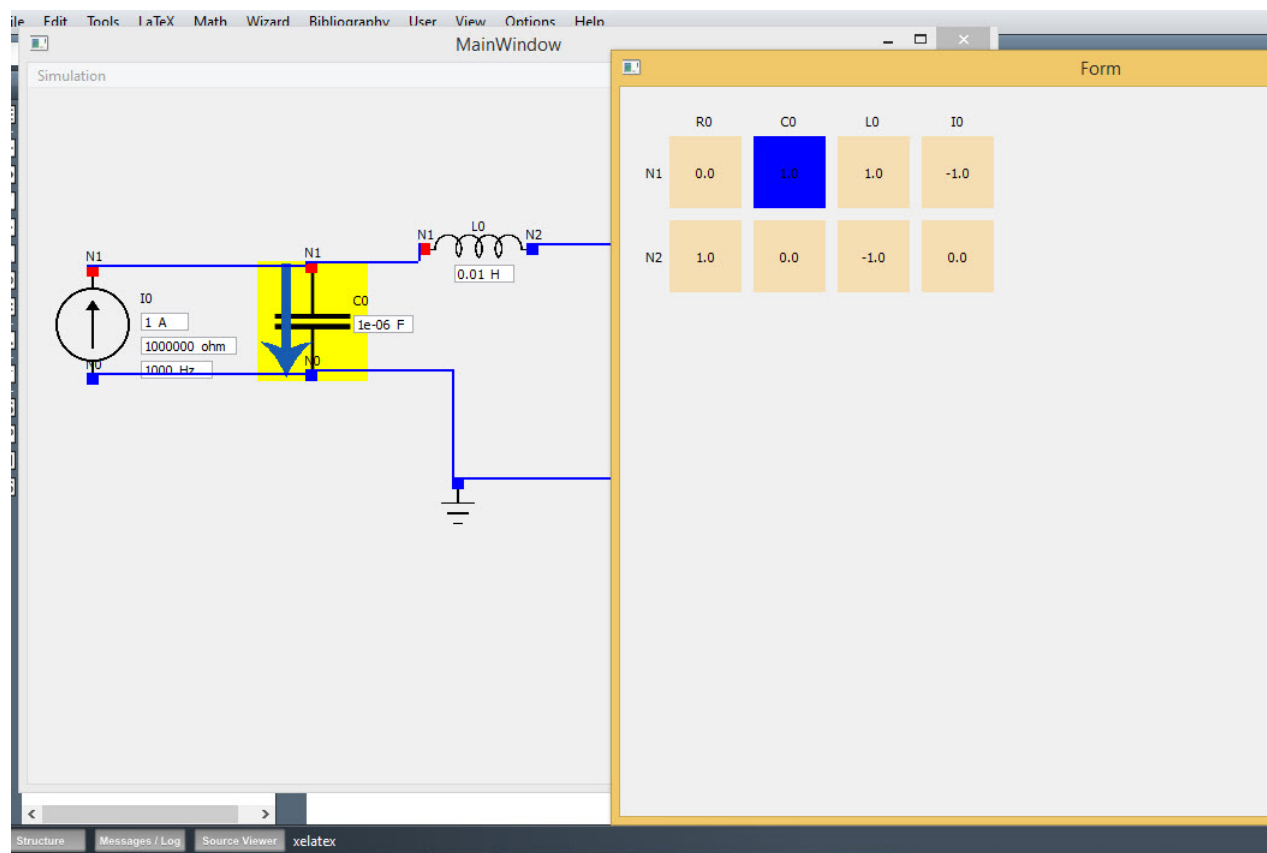
شکل ۱: صفحه gui که در ابتدای باز شدن نرم افزار نمایش داده شده و کاربر در این محیط مدار را رسم میکند:



شکل ۲: رسم مدار توسط کاربر

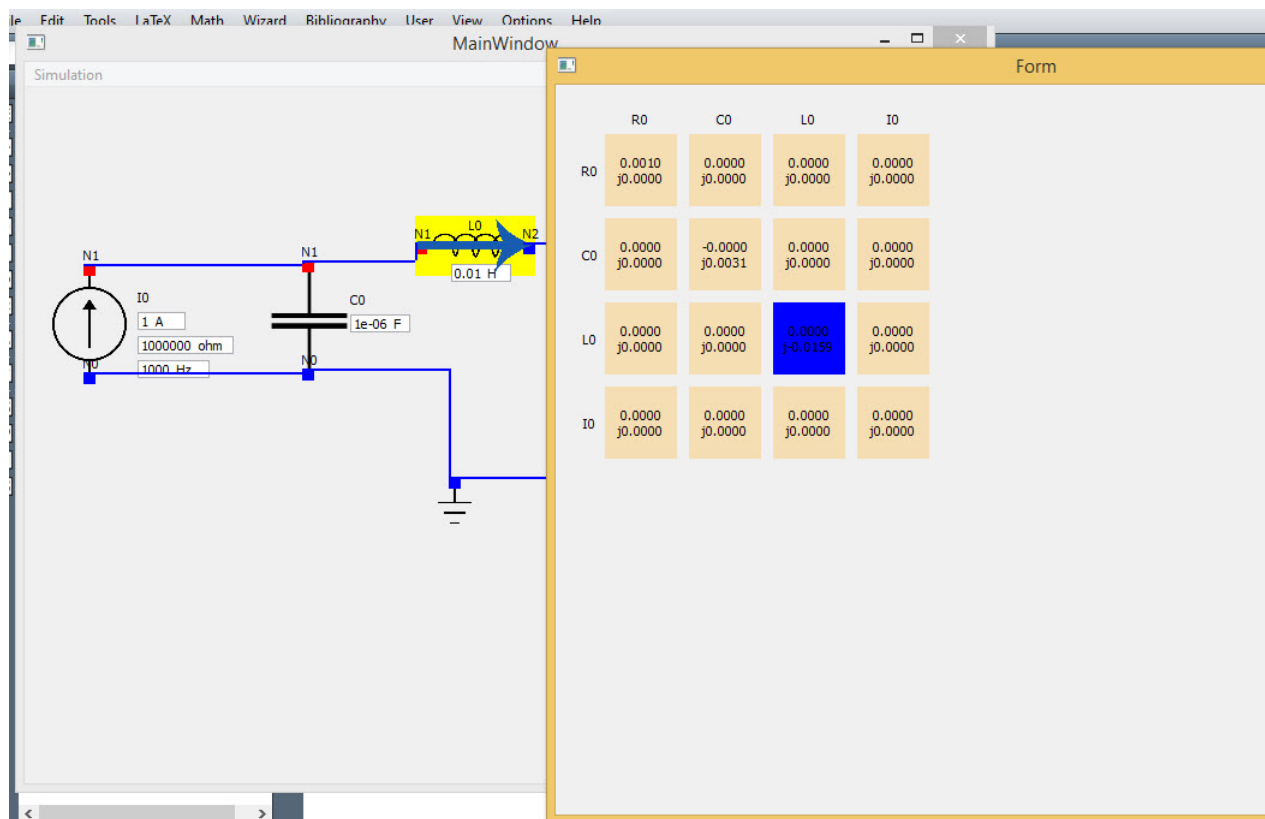


شکل ۳: آنالیز مدار و نمایش ماتریس های اساسی



شکل ۴: انتخاب یک المان از ماتریس A توسط کاربر و نمایش داده شدن آن بر روی مدار





شکل ۵: انتخاب یک المان از ماتریس  $Y$  توسط کابر و نمایش داده شدن آن بر روی مدار