

# 1 Resistor tool

Este programa tiene como objetivo obtener la mejor aproximación posible a un valor de resistencia dado, mediante una combinación serie o paralelo y teniendo en cuenta los valores comerciales existentes para cada tolerancia.

```
function [ r1, r2, error, PorS ] = resistortool( r, tolerance, topology )
% RESISTOR TOOL devuelve la mejor aproximacion al valor de r con los
% valores comerciales existentes para la tolerancia indicada (5, 10 o 20)
% y con la topologia indicada ('s' para serie o 'p' para paralelo)
%
% si la tolerancia no se indica o no es valida, se asume 5%
% si la topologia no se indica o no es valida se devuelve la que
% aproxime mejor el valor pedido
% si el valor de r no es valido (<=0, o no es un unico numero) se
% devuelve -1 en todos los resultados
```

Figure 1: interfaz y descripción de *resistortool* en Matlab

Otros recaudos que se tomaron en la implementación fueron no excederse de los valores máximos y mínimos de los componentes que se encuentran en el pañol de electrónica de la Universidad, a saber:  $0.1\Omega$  y  $10M\Omega$ .

Para encontrar la mejor aproximación al valor de  $r$  en serie, se le asigna sucesivamente a  $r1$  los valores comerciales de la tolerancia que corresponda, en el mismo orden de magnitud que  $r$  y en un orden menos. Esto se debe a que para obtener un número de orden  $N$  sumando dos números, al menos uno de ellos debe ser del mismo orden de magnitud o uno menos que el de la suma que se desea obtener (por ejemplo:  $99+99=198 < 1000$ ,  $95+6=101 \geq 100$ ). Para cada valor de  $r1$  que se prueba,  $r2$  tomará valores en el orden de magnitud de  $r - r1$ . Este proceso continúa hasta que  $r1 \geq r$  (en cuyo caso  $r2 = 0$ ) o  $r1 + r2 > r$ , y se van guardando los valores que resultan en menor error relativo.

En la combinación en paralelo, se pasan todos los parámetros a admitancia, se busca la mejor combinación en serie, y se vuelve a resistencia.

Debido a los errores de aproximación que se producen al realizar las cuentas (sobre todo para el paralelo), luego de obtener  $r1$  y  $r2$  mediante el algoritmo anterior, se verifica que no se haya pasado por alto que  $r$  ya era un valor comercial, en cuyo caso se reemplaza  $r1$  por  $r$  y  $r2$  por 0 o infinito según la topología que se esté usando.

Por último, se aprovechó el código que ya existía hasta el momento para implementar un *capacitortool*, que realiza el mismo proceso pero con otros valores máximos y mínimos ( $1pF$  y  $4700\mu F$ ), e intercambiando serie con paralelo.