This repository includes a physics-based COMSOL Multiphysics model of a TaOx-based memristor (RRAM) device. The model was developed and tested using COMSOL Multiphysics version 6.3 and serves as the core device simulator for the MATLAB-based optimization framework.

Optimization Methodology

1. **Dataset.initial.m**

The optimization workflow starts by running the file "**Dataset.initial.m**". This file contains the reference simulation data for the baseline memristor device. Specifically, the dataset is provided in three columns corresponding to time, applied voltage, and device current, respectively. These data represent the initial current–voltage–time response obtained from the reference COMSOL simulation. In addition, "**Dataset.initial.m**" defines the voltage points at which the device current is evaluated during post-processing. For example, a read voltage of V = −0.1 V is specified for the calculation of the HRS/LRS resistance ratio.
The same file also includes the procedures used to extract key performance metrics, including:

- the forming voltage $(V_f)_{Ref}$, and
- the $(\mathrm{HRS/LRS}_{\ f})_{Ref}$,

which are computed consistently based on the simulated current–voltage characteristics.

2. **Dataset.m**

After computing the $(V_f)_{Ref}$ and $(\mathrm{HRS/LRS}_{\ f})_{Ref}$, the workflow proceeds to the file **"Dataset.m"**. In this file, the extracted reference values are first defined and stored for later comparison during the optimization process. Subsequently, a total of 13 design parameters associated with the RRAM device are defined. For each parameter, the number of members considered in the optimization process is specified. Finally, the initial dataset required for the optimization is generated and stored, providing the starting point for the genetic algorithm.

3. **Cmodel.m**

In the file "**Cmodel.m**", the COMSOL model **"Memristor_COMLAB.mph"** is loaded and all optimization variables are substituted with the corresponding candidate values. This file serves as the interface between MATLAB and the COMSOL device model during the optimization process. Within "**Cmodel.m**", two objective functions are explicitly evaluated:

- Objective 1: the normalized forming voltage, defined as

$$\frac{V_f}{(V_f)_{Ref}}$$

- Objective 2: the normalized resistance ratio, defined as

$$\frac{(\mathrm{HRS/LRS})}{(\mathrm{HRS/LRS})_{Ref}}$$

These normalized objectives enable a consistent comparison between candidate solutions and the reference device.

4. Try.m

In the subsequent file, **"Try.m"**, four solution methods are implemented to improve the robustness of the COMSOL simulation. The script sequentially attempts these methods, starting from Method 1. If the solver fails to converge or the solution diverges, the script automatically proceeds to the next method. This fallback strategy ensures that the optimization process can continue even when certain parameter combinations lead to numerical instability.

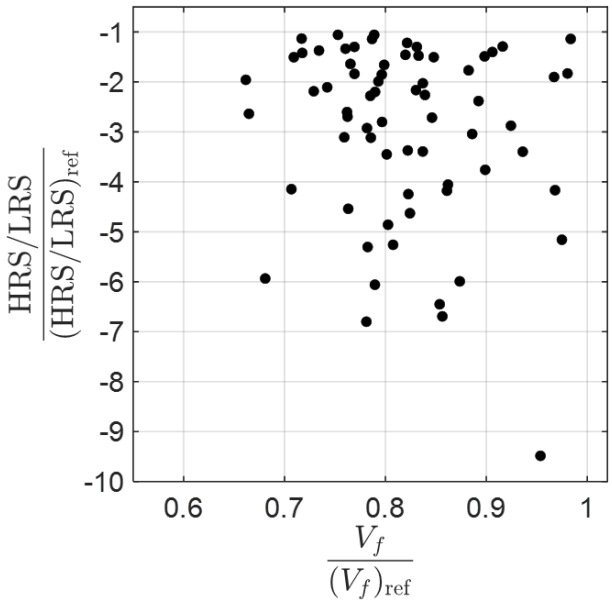
5. Initial_Run.m

The workflow then proceeds to the file **"Initial_run.m"**, which performs the initial run of the optimization framework. This step is required to obtain a valid initial solution and to establish the objective function behavior before launching the full optimization. Within **"Initial_run.m"**, the MATLAB genetic algorithm (`ga`) is invoked to generate the initial random population. In this study, a population size of 200 individuals is used.

During the objective function evaluation, each candidate solution is passed to the function **"Try.m"**, which manages solver robustness, and subsequently to **"Cmodel.m"**, where the COMSOL simulation is executed and the objective values are extracted.

6. Fitfun.m and fitnesscheck.m

After the simulation data are stored, the fitness evaluation is handled in **"Fitfun.m"**. The fitness function is designed such that, as verified in **"fitnesscheck.m"**, the optimization process consistently progresses toward the minimization of both objective functions. If the distribution of the 200 evaluated individuals indicates an unfavorable optimization trend, the fitness formulation is systematically adjusted to ensure simultaneous minimization of the two objectives.



7. **Final_Run.m**

After the initial run is completed, Generation 0 (Gen 0) is created. This initial population and its evaluated objective/fitness values are then used as the starting point for the full optimization. Next, the MATLAB script **"Final_Run.m"** is executed. This script loads the saved initial (Gen 0) dataset and launches the main genetic algorithm loop. In this work, the number of generations is set to 20; therefore, the GA iteration is repeated 20 times (from Gen 1 to Gen 20). At each generation, the GA creates a new population by applying selection, crossover, and mutation based on the fitness values defined in **"Fitfun.m"**. In other words, candidate solutions with better fitness (i.e., lower values of both objectives) are more likely to be selected as parents, and the next generation is formed by recombining and perturbing these parents. Each newly generated individual is then evaluated again through the COMSOL–MATLAB pipeline ("Try.m" → "Cmodel.m") to compute the corresponding objective values, and the results are stored.

After completing 20 generations, the algorithm produces a set of optimized candidates. These final solutions correspond to RRAM device parameter sets that outperform the reference device, as indicated by improved (minimized) normalized objectives relative to the reference forming voltage and reference HRS/LRS ratio.