Nome: Jônatas Negri Grandini
Disciplina: Tópicos Especiais
Professor: Tiago Weber
Universidade Federal de Santa Catarina – Campus Araranguá – Engenharia da Computação

Otimização fonte modo comum

PARTE 1.

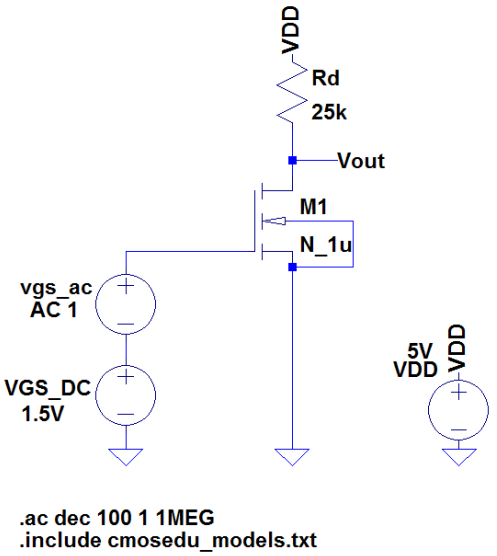

Figura 1: Circuito do exercício

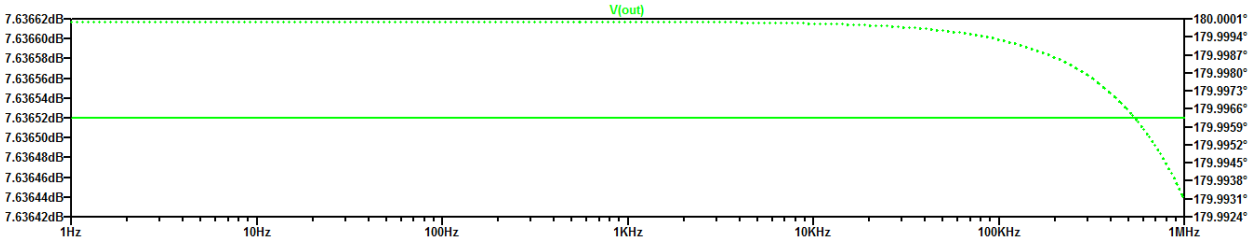

Figura 2: Ganho do circuito da figura 1 em dB

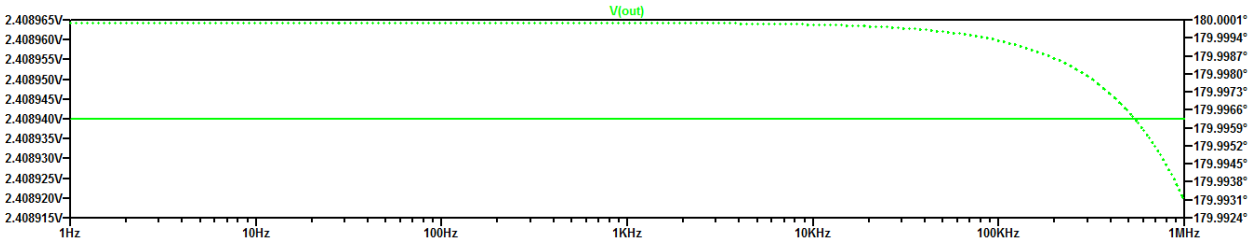

Figura 3: Ganho do circuito da figura 1 linear

Ou seja, deve-se, a partir de agora, melhorar o ganho inicial do circuito que é de 7,64 dB mudando sua resistencia, largura e cumprimento do transistor e V_{GS} em nível DC.

PARTE 2.

Para otimizar o circuito através do algoritmo hill climbing, antes é necessário propor métodos automáticos de obter os valores. Em que deve-se encontrar o ganho e a excursão média de saida.

Para encontrar o ganho, no LTSpice, basta utilizar o método .measure com a seguinte expressão ".meas AC Gain FIND V(out) AT 1". E isso irá resultar, após o circuito ser executado pelo LTSpice, um arquivo .log que irá informar o ganho em dB na frequencia de 1Hz.

Para encontrar a excursão máxima de saída basta calcula-la com a seguinte fórmula: $V_{DD} - (V_{GS} - V_{th})$. E como as funções de gerar o arquivo ja sabem qual o valor os valores de todas essas variáveis, não é necessario um comando no LTSpice para isso.

PARTE 3.

Para otimizar os circuitos utiliza-se da técnica de *Black Box*, em que o otimizador apenas altera os valores de entrada e a *Black Box* que retorna o valor da função custo, fazendo todos os passo necessários para obte-lo.

Obter a função custo, para simulações elétricar em LTSpice consiste em 3 etapas principais: criar um arquivo *NetList* que represente o circuito conforme os valores de entrada no *black box,* executar o simulador LTSpice utilizando o circuito feito no passo anterior e coletar o resultado no arquivo .log gerado pelo LTSpice.

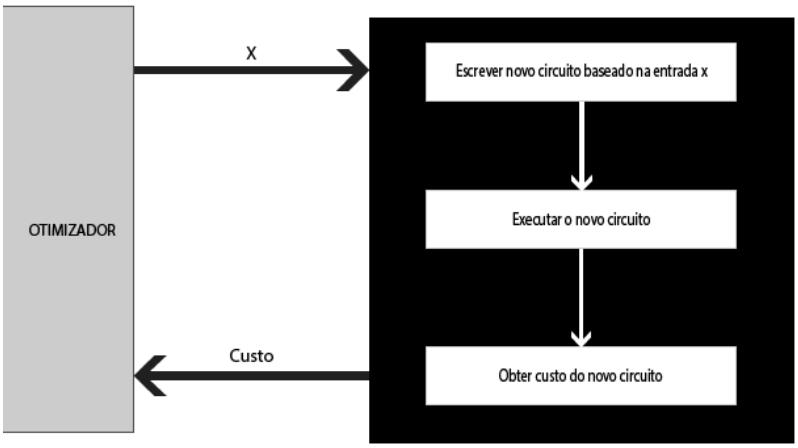

Figura 4: Funcionamento da implementação Black Box

Dos arquivos fornecidos pelo professor, era necessário alterar a função do MATLAB de escrever o novo circuito para que fosse possível alterar mais de um parametro no novo circuito a ser criado, como é o caso desse exercício, em que buscava-se alterar o valor de $R_d, V_{GS}, W \ e \ L$.

A função de executar o circuito se mantém a mesma, visto que é apenas uma chamada do programa LTSpice. Já a função de obter custo do novo circuito se manteve praticamente a mesma, visto que é apenas ler um arquivo .log, foi alterado apenas qual o parametro a se ler da simulação, que no caso era o ganho.

Porém a função custo em si também foi alterada para ser um valor mais plausível, já que a função custo que havia antes era de aproximação quadrática. A função custo utilizada foi “y = 1-ganho/35” isso pois, imagindo que o ganho máximo possível é de 35 dB, a função custo ótima será igual a 0, e empiricamente o valor máximo de ganho para esse circuito encontrado foi de 32,7. Valores menores que 35 gerarão (inclusive negativos), necessariamente valores maiores que zero, cumprindo com o que deve ser uma função custo.

A função hill climbing fornecida se manteve praticamente a mesma, com a diferença agora que os valores a serem alterados tem limites inferiores e superiores e são mais do que um valor. O princípio do algoritmo ainda se mantém o mesmo, que é, gera-se valores de entradas aleatório próximos dos valores que geraram o menor custo, se esses novos valores gerarem um novo custo ainda menor, estes valores alterados se tornam o novo ótimo e repete-se a iteração até que ocorra o número máximo de iterações.

A função antiga de gerar novos valores era apenas $0.1 * rand * valorÓtimo$, ou seja, o novo valor será uma mudança de até $\pm 10\%$ do valor antigo desde que respeite os limites superiores e anteriores. A nova função proposta é a seguinte:

```
novo = otimo + e^(-5*i/iterações)*randn*(limiteSuperior- limiteInferior)
```

Sendo *novo* o novo valor da variável, *ótimo* o valor que atualmente é considerado ótimo, *i* a iteração desse momento, *iterações* o número máximo de iterações, *rand* um valor de média 0 e desvio padrão 1, *limiteSuperior e limiteInferior* os valores máximos e mínimos respectivamente da variável.

Essa expressão é aparentemente complexa, mas sua idéia é simples. A idéia é que, no começo das iterações a mudança do valor ótimo seja muito grande (para fugir de possíveis mínimos locais) e conforme as iterações vão chegando próximas ao fim suas alterações sejão mais cirurgicas em relação ao valor atual. Isso corresponde a parte exponencial da equação a cima. Ja a subtração do limite inferior pelo limite superior é para que a mudança seja proporcional aos possíveis valores que a variável pode admitir.

Esse algoritmo de otimização obteve, com 500 iterações, um ganho de **32,7 dB com** $\mathbf{R_D = 98,09049k\Omega}$ **,** $\mathbf{V_{GS} = 1V, L = 4,71um}$ **e** $\mathbf{W = 99,76um}$. E a alteração da função custo ótima e a cada iteração são representados na *figura 5.*

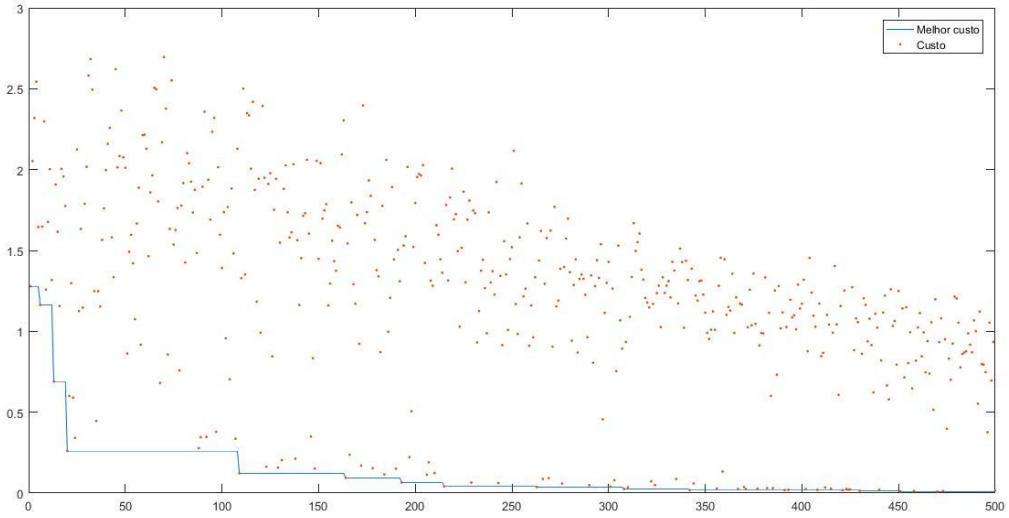

Figura 5: Variação do melhor custo e do custo com o passar das iterações

No gráfico é possível observar que conforme a iteração vai chegando próxima ao fim, os pontos começam a ficar menos dispersos, indicando que o algoritmo de reduzir as mudanças com o passar das iterações está realmente funcionando. Observa-se que o eixo y está representado pelo valor da função custo, enquanto o eixo x representa as iterações.

PARTE 4

Semelhante ao passo anterior, porém agora com uma função custo mais elaborada, que é de multiplas variáveis, é necessário elaborar uma nova função custo. Essa nova função custo é

$$custo\ = 1 - 0{,}5 * \frac{ganho}{35} - 0{,}5 * \frac{\text{excursão}}{5.8}$$

Isso pois o valor máximo da excursão de saída sera a tensão máxima de alimentação V_{DD} que é 5V somado V_{th} que é 0.8V ou seja 5,8V, conforme a fórmula de excursão de saída. E os valore estão multiplicados por 0,5 para que sua soma seja unitária.

Como para esse circuito, o ganho foi máximo quando V_{GS} era mínimo (1V), consequentemente uma tensão de excursão maior, espera-se que o resultado da simulação seja semelhante ao anterior. Isso pois o algoritmo não precisa tentar balancear

entre qual dos dois parametros diminui influencia mais o custo pois ambos influenciam de forma muito semelhante.

Essa idéia foi comprovado ao rodar a simulação em que, com 500 iterações, obteve-se **Ganho:31.66 dB e Excursão:4.80V com RD=67621.95Ω VGS=1.00V W=73.69um L=2.60um**. Sendo que essa diferença de aproximadamente 1 dB é oriunda da aleatoriedade da forma como o ganho é encontrado, em que, a cada vez que roda-se a simulação o ganho fica em torno de 30,5 e 32,7dB.

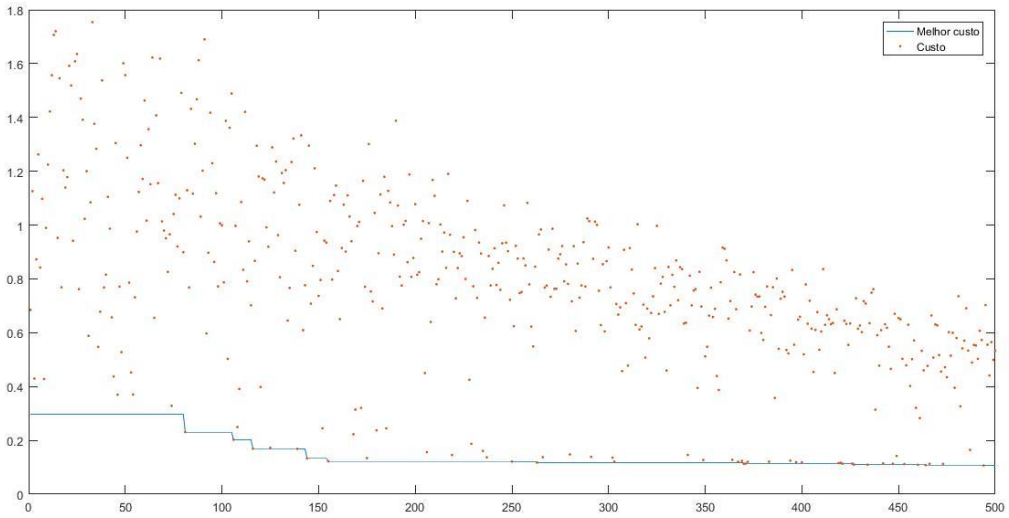

Figura 6: *Variação do melhor custo e do custo com o passar das iterações para função custo com pesos multiplos*