

Esta documentação descreve sumariamente o uso do programa HK e de seus programas auxiliares APRA, APRS, APOL e LADDER, que compõem um pacote para a síntese de aproximações para filtros analógicos e para a obtenção de realizações passivas para elas. Inicialmente é feita uma descrição dos programas, seguindo-se um exemplo completo de síntese.

1 - O programa HK

O programa HK obtém aproximações para filtros analógicos contínuos de vários tipos, usando o método clássico de aproximação usando a função de transdução $H(s)$ e a função característica $K(s)$.

Na notação adotada, a função característica é definida por $K(s)=F(s)/P(s)$, e a função de transdução por $H(s)=E(s)/P(s)$. onde $F(s)$, $E(s)$ e $P(s)$ são polinômios com coeficientes reais. $K(s)$ e $H(s)$ são associadas pela equação de FeldtKeller: $H(s)H(-s)=1+K(s)K(-s)$, ou $E(s)E(-s)=F(s)F(-s)+P(s)P(-s)$. Desta forma, $|H(j\omega)|$ é a atenuação do filtro, e a função de transferência desejada vale $T(s)=k/H(s)$, onde k depende da forma de realização. Os pólos da função de transferência $T(s)$ são as raízes de $E(s)$, e seus zeros, os zeros de transmissão, são as raízes de $P(s)$. As raízes de $F(s)$ são os zeros de atenuação.

Apenas aproximações tipo passa-baixas podem ser diretamente construídas pelo programa, mas é possível transformá-las depois para outras formas. Em todos os cálculos, o programa assume aproximações normalizadas, com corte ou frequência central em 1 rad/s.

Dada $K(s)$, o programa calcula $H(s)$, ou dada $H(s)$, o programa calcula $K(s)$. Os polinômios necessários, $F(s)$ e $P(s)$ ou $E(s)$ e $P(s)$, podem ser fornecidos ao programa usando-se a opção de ler polinômio, ou calculados pelo próprio programa, nos casos de alguns tipos particulares de aproximação. $K(s)$ é calculada diretamente para quatro tipos de aproximação em que $K(j\omega)$ é puramente real, e é obtida a partir de um polinômio característico $Q(w)$. $H(s)$ é calculada diretamente para a aproximação de Bessel.

Aproximações polinomiais, onde $K(j\omega)=\varepsilon Q(w)$, incluem as aproximações de Butterworth e Chebyshev. O parâmetro ε define a atenuação na banda passante. A aproximação é obtida a partir do polinômio característico $Q(w)$, e o

programa gera os polinômios adequados para aproximações de Butterworth ($Q(w)=wn$), Chebyshev ($Q(w)=Cn(w)$) e Chebyshev modificada ($Q(w)=CMn(w)$), que é $Cn(w)$ com menores raízes movidas para a origem por uma transformação de Moebius) polinômios, que levam a aproximações intermediárias entre estes extremos, podem ser gerados pelo programa APOL, e lidos pelo programa.

Aproximações polinomiais inversas, onde $K(jw)=\varepsilon\alpha^2wn/Qr(w)$ e $Qr(w)=wnQ(1/w)$ é o polinômio $Q(w)$ com os coeficientes em ordem reversa, incluem as aproximações de Chebyshev inversas ($Q(w)=Cn(w)$). O parâmetro α define a relação entre a atenuação nas bandas passante e de rejeição. Todos os polinômios geráveis pelo programa APOL podem também ser usados para estas aproximações.

Aproximações racionais simétricas onde $K(jw)=\varepsilon\alpha Q(w)/Qr(w)$ incluem as aproximações elípticas. Polinômios $Q(w)$ adequados, mesmo para o caso das aproximações elípticas, devem ser gerados numericamente pelos programas APRS ou APRA.

Aproximações racionais simétricas modificadas, onde $K(jw)=\varepsilon(2\alpha Q(w)-Qr(w))/Qr(w)$ incluem aproximações com "ripple" uniforme na banda passante e zeros de transmissão duplos, de ordem par (um caso algo particular). Polinômios $Q(w)$ adequados devem ser gerados pelo programa APRS.

Outras aproximações podem ser geradas, fornecendo-se ao programa $H(s)$ ou $K(s)$, e calculando a outra função.

A opção de elevar um polinômio ao quadrado serve para a obtenção de aproximações cujas funções $K(s)$ ou $H(s)$ sejam o quadrado de alguma outra, por exemplo.

2 - Procedimento normal de síntese:

O processo normal para a geração de uma aproximação, a partir de $K(s)$, e obter uma realização para o filtro é:

- Obter o polinômio $Q(w)$ e usar um dos quatro tipos de aproximação para obter $K(s)$, ou então ler diretamente os polinômios $F(s)$ e $P(s)$ gerados externamente. O programa APRA pode ser usado para gerar estes polinômios para qualquer aproximação passa-baixas com $K(jw)$ real. Os programas APOL e APRS são especializados (e algo otimizados) para aproximações polinomiais diretas ou inversas, e racionais

simétricas, respectivamente. Eles geram $Q(w)$, mas podem também gerar $F(s)$ e $P(s)$.

- Calcular $E(s)$. O programa resolve a eq. de FeldtKeller por fatoração polinomial, calculando primeiramente as raízes de $E(s)E(-s)$, e montando o polinômio $E(s)$ com as que estiverem no SPLE. Após este cálculo, as raízes de $E(s)$ podem ser listadas, salvas, ou plotadas no plano complexo. Gráficos de $|K(jw)|$, $|H(jw)|$ e $|T(jw)|$ (este em dB) podem ser plotados tão logo os polinômios necessários estejam disponíveis.

- Gerar imitâncias para síntese. O programa gera imitâncias para síntese por rede LC simplesmente terminada ou duplamente terminada. As impedâncias LC da realização em "lattice" também podem ser calculadas. Terminações unitárias são usadas, sempre que possível. As redes que realizam estas imitâncias podem ser obtidas pelo programa LADDER. No caso de realizações em forma fisicamente simétrica ou antissimétrica, os programas LADDER, SIMET e LATTICE podem ser usados para a obtenção das redes, diretamente a partir das raízes de $E(s)$ no caso de aproximações com $K(s)$ polinomial (SIMET), ou a partir das raízes de $E(s)$ e de $P(s)$ no caso geral (LADDER e LATTICE).

No caso de geração de aproximação a partir de $H(s)$, tendo-se $E(s)$ e $P(s)$, $F(s)$ deve ser calculado, também pela equação de FeldtKeller. Como não existe necessidade das raízes de $F(s)$ estarem no SPLE, o programa pede o sinal das raízes de $F(s)F(-s)$ que devem compor $F(s)$. Atenção deve ser dada ao sinal das partes imaginárias, que devem estar em pares conjugados, e ao ajuste das tolerâncias para cálculo de raízes no caso de aproximações cujas funções características possuam zeros de atenuação sobre o eixo jw . Especialmente nestes casos, recomenda-se usar a síntese a partir de $K(s)$ sempre que possível.

Obtidos os polinômios $F(s)$, $P(s)$ e $E(s)$, transformações tipo PB-PA, PB-PF e PB-RF podem ser aplicadas a eles. Isto é útil para a verificação do efeito das transformações nas curvas de resposta em frequência e nos pólos e zeros. As imitâncias para síntese são geradas corretamente, mas o programa LADDER não as realiza por necessitarem de extrações que não são no infinito. Para realização de aproximações convencionais não passa-baixas é melhor utilizar as transformações na rede que o programa LADDER realiza ao fim da síntese (Pode também ser usado o programa LC, não

descrito aqui, que realiza expansões arbitrárias de redes LC em "ladder". Ele necessita dos pólos e zeros da admitância a expandir, que podem ser calculados pelo HK a partir dos numeradores e denominadores salvos em disco).

3 - Observações:

O programa HK opera internamente como uma calculadora RPN polinomial. Acessíveis externamente são os polinômios temporários X e Y, e os registradores polinomiais Q, F, P e E.

A rotina de cálculo de raízes de polinômio usada pelo programa é bastante sofisticada, podendo encontrar com grande precisão todas raízes de qualquer multiplicidade de polinômios de ordem elevada. Acontece frequentemente, entretanto, que a rotina não percebe que uma raiz é múltipla, devido à tolerância inicial para valor de polinômio usada, propositalmente pequena demais (raízes múltiplas são detectadas pelo módulo da derivada do polinômio, mas se forem detectadas indevidamente as demais raízes estarão erradas). Nestes casos ela deve ser aumentada até que as raízes múltiplas sejam corretamente calculadas. Este problema é comum no cálculo das raízes de $P(s)$ (para usar no programa LADDER) quando os zeros de transmissão são múltiplos, e no cálculo de $F(s)$, quando a aproximação possui zeros de atenuação sobre o eixo imaginário. Pode ocorrer também que a convergência para uma raiz não seja atingida após um número limite de iterações. Neste caso, o programa para e pede novos valores para tolerâncias e aproximação da raiz. Pode-se aumentar a tolerância para valor de polinômio, caso em que o programa tentará encontrar raízes múltiplas, aumentar a tolerância para raízes, caso em que a aproximação atual será aceita como raiz, ou não mudar nada e tentar mais um ciclo de iterações. Raramente é necessário alterar a aproximação para a raiz. A rotina pode ser usada separadamente, lendo-se o polinômio como qualquer dos polinômios usados pelo programa e pedindo-se suas raízes, que podem ser listadas/salvas ou plotadas. Observe-se que as raízes calculadas são sempre as do polinômio X, e o número de raízes listadas ou plotadas é sempre o grau de X. Se alguma operação alterar X após um cálculo de raízes, as raízes mostradas poderão não ser válidas.

No gráfico de raízes, o cursor pode ser movimentado com Return e Backspace, o gráfico movido com o cursor, e a

ampliação mudada com + e -. Espaço centra o gráfico no cursor e Esc termina.

Nos outros gráficos, o cursor é ativado e desativado com " ". O gráfico pode ser movimentado com as teclas do cursor, "A", "R", "-" e "+". Esc termina. O cursor se move mais rapidamente com a tecla Ctrl pressionada. Apertando outra tecla, o programa permite mudar as escalas ou salvar uma tabela com os dados do gráfico.

Os programas procuram os "drivers" gráficos (do Turbo Pascal, arquivos *.BGI) no diretório local ou no diretório indicado pela variável do DOS "TPBGI". Use o comando abaixo (no autoexec.bat, por exemplo) para setar a variável:

```
set tpbgi=<diretório onde estão os arquivos *.BGI>
```

4 - O programa APOL

Este programa destina-se ao projeto de polinômios $M(w)$, utilizáveis em aproximações polinomiais e polinomiais inversas (como $Q(w)$ no HK). Os polinômios característicos gerados são pares ou ímpares, e oscilam de forma semelhante a polinômios de Chebyshev para $-1 \leq w \leq 1$. Podem ser diretamente gerados polinômios de Chebyshev e Chebyshev modificados por transformação de Moebius (isto o HK também faz). Outros polinômios necessitam de um processo de otimização. O programa pede então o número de raízes na origem de $M(w)$ e uma série de parâmetros f , que são os valores limites para as oscilações de $K(jw)/\varepsilon = M(w)$ para $0 < w \leq 1$. No caso de ordem par sem zeros na origem, o primeiro valor corresponde a $w=0$. Os valores "default" correspondem a um polinômio de Chebyshev. Outras opções permitem diversas manipulações e um gráfico de $M(w)$.

5 - O programa APRS

Este programa destina-se ao projeto de polinômios $Q(w)$, utilizáveis em aproximações racionais simétricas. As funções características geradas, tipo $K(jw)/\varepsilon = \alpha Q(w)/Qr(w)$, são semelhantes às das aproximações elípticas, ou de Cauer, apresentando oscilações nas bandas passante e de rejeição, com zeros de transmissão e atenuação em frequências inversas uma da outra. Todos os casos são gerados por otimização numérica. O programa pede as especificações de ordem e atenuação da aproximação desejada, o número de zeros de atenuação na origem e uma série de parâmetros f ,

que são os valores de $K(j\omega)/\varepsilon$ para $0 < \omega < 1$ (1º valor em $\omega=0$ se não existir raiz na origem). Os valores "default" resultam em uma aproximação elíptica normal. O programa pode também gerar $K(s)$ diretamente. Neste caso, é conveniente normalizar antes a aproximação para corte em 1 rad/s, o que o programa pode fazer. Note-se que o programa HK necessita apenas do polinômio $\alpha Q(\omega)$ (em que a cte. multiplicativa do polinômio $Q(\omega)$ vale α). A função característica normalizada, $K(j\omega)/\varepsilon$, pode ser plotada para verificação.

6 - O programa APRA

Este programa projeta o caso geral das aproximações passa-baixas em que $K(j\omega)$ é real, podendo encontrar as mesmas aproximações que os programas APOL e APRS encontram (um pouco mais lentamente), e outras mais. Nestas aproximações, $K(j\omega)/\varepsilon = \alpha X(\omega)/Y_r(\omega)$, onde $X(\omega)$ e $Y(\omega)$ são polinômios independentes, de mesma ordem. O cálculo é feito também por otimização. O programa pede as especificações de ordem e atenuação (A_{max} e A_{min}) da aproximação a obter, o número de zeros de atenuação na origem, o número de zeros de transmissão no infinito (zeros na origem de $X(\omega)$ e $Y(\omega)$), e os valores extremos f_x e f_y , de $\alpha X(\omega)/Y_r(\omega)$ e de $\alpha Y(\omega)/X_r(\omega)$ para $0 < \omega < 1$ (1º valor em $\omega=0$ se não existir raiz na origem). Cada valor extremo de $\alpha X(\omega)/Y_r(\omega)$, $f_x = \pm 1$, equivale a um ponto de atenuação A_{max} na banda passante, e cada passagem por zero a um zero de atenuação. Um zero duplo de atenuação pode ser criado especificando-se $f_x = 0$. Para $\alpha Y(\omega)/X_r(\omega)$, cada valor extremo $f_y = \pm 1$ equivale a um ponto de atenuação A_{min} na banda de rejeição, e cada passagem por zero a um zero de transmissão. Um zero duplo de transmissão pode ser criado especificando-se $f_y = 0$.

O programa permite também a especificação dos extremos pela atenuação desejada em cada um deles. Para o uso dos resultados deste programa no programa HK, a função característica $K(s)$ deve ser salva, normalmente antes normalizada para corte em 1 rad/s. A função característica normalizada, $K(j\omega)/\varepsilon$, pode também ser plotada para verificação.

Nos três programas, o processo de otimização pode ser interrompido pelo toque de uma tecla, se necessário (quando a convergência com a tolerância especificada não é atingida). Existe também um cálculo de raízes polinomiais no processo de otimização, e pode também ser necessário

reduzir a tolerância nas raízes quando a convergência não é atingida (O programa para após um número de iterações, sugerindo como nova tolerância o valor atualmente atingido. Pode-se aceitá-lo ou especificar outro valor, que deve ser mantido tão baixo quanto possível).

Ester três últimos programas foram desenvolvidos com o objetivo principal de projetar aproximações com "ripple" uniforme nas bandas passante e de rejeição, mas com zeros de atenuação e transmissão podendo ser duplos (Nos três programas um caso extremo é obtido nos casos de ordem ímpar especificando-se um zero na origem, os f ímpares com o valor "default" e os pares como 0. Para ordens pares, o número de zeros na origem deve ser de 2 ou 4, dependendo da ordem, e a regra para os f é a mesma). Aproximações assim, embora algo menos seletivas que aquelas em que todos os zeros são simples, são realizáveis em formas com menor sensibilidade e apresentam melhores características de atraso de grupo. Se os zeros de transmissão forem duplos, realizações fisicamente simétricas ou antissimétricas são possíveis, com vantagens para simulações analógicas integradas, em que a simetria permite melhor imunidade a gradientes de processo ou temperatura, e para simulações digitais, com menor sensibilidade a arredondamento de parâmetros e possíveis simplificações de "hardware".

7 - O programa LADDER

Este programa destina-se a realizar os cálculos necessários à expansão de uma imitância de porta dada em "ladder" RLC, com zeros de transmissão criados pela técnica de deslocamento de zeros.

O programa pede inicialmente o tipo de imitância a considerar, e a seguir dois conjuntos opcionais de raízes, para serem usados como zeros de transmissão a realizar e pólos da transferência. Os pólos são úteis somente quando da síntese da imitância de meia-rede de uma rede simétrica ou antissimétrica. Estas raízes são lidas de arquivos, no formato em que são salvas pelo programa HK. Sempre que o programa pedir uma raiz a usar em alguma operação, estas raízes serão listadas, e seus valores poderão ser referenciados pelos nomes dados (ou outro valor poderá ser escrito).

O programa pede então o que será expandido: Uma imitância dada por uma razão de polinômios ou uma imitância de

meia-rede de uma rede simétrica ou antimétrica. Nos últimos casos, as seleções de pólos e zeros da $f. de t.$ a realizar que serão usados para montar a imitância são pedidos. No primeiro caso (o mais usual), são pedidos o numerador e o denominador da imitância a expandir, que podem ser lidos de arquivo (como salvos pelo HK) ou do teclado. No caso de leitura do teclado, os polinômios podem ser especificados por seus coeficientes ou por suas raízes, caso em que os arquivos de pólos e zeros anteriormente lidos podem ser úteis.

A seguir, entra-se no processo de expansão da rede. Cada passo da síntese deve ser especificado. O programa apenas realiza os cálculos necessários, após verificar a possibilidade das operações pedidas. Caso uma operação seja impossível, nada é feito e pode-se tentar outra operação na mesma imitância. A qualquer momento, a síntese pode ser abandonada ou reiniciada, com outra imitância ou com a mesma.

São permitidas extrações parciais e totais de constante e pólo no infinito, o que é suficiente para a realização de aproximações passa-baixas. O programa admite extrações em imitâncias RLC, inclusive para a realização de zeros de transmissão complexos por tanques RLC.

Ao fim da síntese, a rede obtida pode ser salva em arquivo, ou modificada por escalamentos em frequência e impedância e transformações tipo PB-PA, PB-PF e PB-RF. No caso de surgimento de tanques LC compostos, estes podem ser convertidos para a forma de associação de tanques simples.

Note-se finalmente que, embora os programas verifiquem a consistência de diversas operações, o conjunto foi implementado como ferramenta de investigação, e não como sistema especialista. É necessário que o usuário saiba o que está fazendo. "Garbage in - garbage out" vale...

8 - Exemplo completo de síntese:

Seja a síntese de um filtro passa-baixas de 7^a ordem, com as especificações:

- "Ripple" uniforme na banda passante de 1 dB, com zeros de atenuação simples.
- "Ripple" uniforme na banda de rejeição, com $A_{min}=50$

dB, sendo dois pares de zeros duplos e um par de zeros simples de transmissão, estes em frequência mais alta.

- Realização em "ladder" LC duplamente terminada fisicamente simétrica, com terminações de 50 Ohms e corte de 1 dB em 20 kHz.

A função característica é gerada pelo programa APRA, onde as perguntas sobre especificações para a otimização devem ser respondidas como:

Grau da aproximação a obter: 7
No. de zeros de atenuação na origem: 1
No. de zeros de transmissão no infinito: 1
Atenuação mínima na banda de rejeição: 50
Atenuação máxima na banda passante: 1

ε : 0.508847139909587396
 α : 24.929022257639580100

Aproximação para $X(w)$:
Aproximação inicial: $w^{0.7}$
Extremos de $\alpha X(w)/Y_r(w)$ em $0 \leq w < 1$:
f(1): -1.0000000 Aten(1): 1.0000000 dB
f(2): 1.0000000 Aten(2): 1.0000000 dB
f(3): -1.0000000 Aten(3): 1.0000000 dB
Mudar f, Atenuação, Sinal de f ou Continuar: C

Aproximação para $Y(w)$:
Aproximação inicial: $w^{0.7}$
Extremos de $\alpha Y(w)/X_r(w)$ em $0 \leq w < 1$:
f(1): -1.0000000 Aten(1): 50.0000000 dB
f(2): 1.0000000 Aten(2): 50.0000000 dB
f(3): -1.0000000 Aten(3): 50.0000000 dB
Mudar f, Atenuação, Sinal de f ou Continuar: F

Indice: 3
Extremo f(3): 0
Extremos de $\alpha Y(w)/X_r(w)$ em $0 \leq w < 1$:
f(1): -1.0000000 Aten(1): 50.0000000 dB
f(2): 1.0000000 Aten(2): 50.0000000 dB
f(3): 0.0000000 Aten(3): 1000.0000000 dB (inf)
Mudar f, Atenuação, Sinal de f ou Continuar: C

Observe-se que na banda passante, os extremos de $K(jw)/\varepsilon = \alpha X(w)/Y_r(w)$ são os valores f dados nas especificações para $X(w)$, em ordem crescente de frequência, e na banda de rejeição os valores de $K(jw)/\varepsilon$ nos extremos entre

os zeros de transmissão valem α^2/f , com os f dados na especificação de $Y(w)$, em ordem decrescente de frequência. Daí o valor 0 no último f , colocando zeros duplos de transmissão no início da banda de rejeição. O programa trata atenuações de mais de 1000 dB como infinitas, daí a atenuação correspondente ao último extremo.

A partir deste ponto o programa entra no processo de otimização, que pode ser acompanhado pela listagem na tela da tolerância atual. Se necessário o processo pode ser interrompido com o toque de uma tecla. Ao fim da otimização, é recomendada a observação do gráfico da função característica normalizada obtida [G]. Recomenda-se tentar outras especificações e observar o resultado.

Após a otimização, a função característica deve ser escalada para corte em 1 rad/s, e salva: [E][K].wp é a frequência de corte da aproximação antes do escalamento. Examine-se o efeito no gráfico [G].

```
wp=      0.948736682747717275
* K(jw)/ε escalado *
#
Polinômio P(s):
- a( 0):      6.558110987467952850
- a( 1):      -0.000000000000000000
- a( 2):      11.415608221834105500
- a( 3):      0.000000000000000000
- a( 4):      6.208520159528418960
- a( 5):      -0.000000000000000000
- a( 6):      1.000000000000000000
- Cte:       0.016361146471550049
Salvar? (Cr ou nome) ex.p
Polinômio F(s):
- a( 0):      0.000000000000000000
- a( 1):      0.261052862675191122
- a( 2):      -0.000000000000000000
- a( 3):      1.369003151627380620
- a( 4):      0.000000000000000000
- a( 5):      2.105027909864539030
- a( 6):      -0.000000000000000000
- a( 7):      1.000000000000000000
- Cte:       1.000000000000000000
Salvar? (Cr ou nome) ex.f
```

Passa-se agora ao programa HK, primeiramente fornecendo a função característica calculada, pela leitura de arqui-

vos com os coeficientes dos polinômios $F(s)$ e $P(s)$:
[O][F][C], [O][P][C]:

Ler $Q(w), F(s), P(s)$ ou $E(s)$ (ou $X(s), Y(s)$)? F
Coeficientes, Raízes ou w e Q ? C
Nome do arquivo ou Cr: ex.f

Ler $Q(w), F(s), P(s)$ ou $E(s)$ (ou $X(s), Y(s)$)? P
Coeficientes, Raízes ou w e Q ? C
Nome do arquivo ou Cr: ex.p

Pede-se então o cálculo de $E(s)$ [E], e a geração de imitâncias da rede LC para a síntese em "ladder" LC duplamente terminada [M][D]. Seja a geração de $z11/rg$ [1]:

Numerador da imitância:

- a(0): 0.117331602261899717
- a(1): 0.000000000000000000
- a(2): 1.024433127346216610
- a(3): 0.000000000000000000
- a(4): 1.907482372005927740
- a(5): 0.000000000000000000
- a(6): 1.000000000000000000
- Cte: 0.914486910381909324

Salvar? (Cr ou nome) ex.n11

Denominador da imitância:

- a(0): 0.000000000000000000
- a(1): 0.369852787202412714
- a(2): 0.000000000000000000
- a(3): 1.682129298405598660
- a(4): 0.000000000000000000
- a(5): 2.314032565401035860
- a(6): 0.000000000000000000
- a(7): 1.000000000000000000
- Cte: 2.000000000000000000

Salvar? (Cr ou nome) ex.d11

Os zeros de transmissão, raízes do polinômio $P(s)$, devem ser salvos também. $P(s)$ possui raízes duplas no caso, e a tolerância para valor de polinômio deve ser alterada antes do cálculo, para maior precisão [T]:

Mínimo valor de polinômio não nulo: $1e-5$

A seguir, as raízes de $P(s)$ são calculadas e listadas [A][P][Z]:

Raízes de $Q(w), F(s), P(s)$ ou $E(s)$ (ou $X(s), Y(s)$)? P
Calculando...+6 5 +4 3 2 1
- x(1): 0.0000000000000000 -1.857891001386838j
- x(2): -0.0000000000000000 1.857891001386838j
- x(3): -0.0000000000000000 -1.174044544830865j
- x(4): -0.0000000000000000 -1.174044544830865j
- x(5): 0.0000000000000000 1.174044544830865j
- x(6): 0.0000000000000000 1.174044544830865j
- Cte: 0.016361146471550049
Salvar? (Cr ou nome) ex.z

E recomendada a observação dos gráficos de resposta em frequência e das raízes dos polinômios: [G][T], [G][K], [G][H], [G][F], [G][A], [A][E][X], [A][P][X], [A][F][X].

Passa-se agora ao programa LADDER. A imitância a expandir é uma impedância [I], e os zeros de transmissão devem ser lidos. Não há pólos a ler:

Zeros de transmissão a realizar (opcional)
Nome do arquivo: (cr p/ignorar) ex.z
6 frequências:
x1: 0.0000000000000000 -1.857891001386838j
x2: -0.0000000000000000 1.857891001386838j
x3: -0.0000000000000000 -1.174044544830866j
x4: -0.0000000000000000 -1.174044544830866j
x5: 0.0000000000000000 1.174044544830866j
x6: 0.0000000000000000 1.174044544830866j

Pólos da f. de t. a realizar (opcional)
Nome do arquivo: (cr p/ignorar)

A impedância é dada por uma razão de polinômios [P], lidos a seguir:

Numerador:
Nome do arquivo: (cr p/teclado) ex.n11
Denominador:
Nome do arquivo: (cr p/teclado) ex.d11

Para a realização por estrutura simétrica, os zeros simples serão realizados por um tanque LC no centro da rede, e os duplos por dois tanques iguais, um em cada lado. A função característica puramente ímpar garante a simetria do resto da rede. A sequência de comandos é: [I][Z], [Z], [Z], [P]. Uma inversão para criar pólo no infinito, três extrações parciais do pólo no infinito para

criar os três pares de zeros de transmissão, e a extração completa do pólo no infinito.

```
# Admitância:
z1: 0.0000000000000000 -1.857891001386838j
z2: -0.0000000000000000 1.857891001386838j
z3: -0.0000000000000000 -1.174044544830866j
z4: -0.0000000000000000 -1.174044544830866j
z5: 0.0000000000000000 1.174044544830866j
z6: 0.0000000000000000 1.174044544830866j
Frequência: (Re Im ou nome) z3
C1: 1.292597534873804 em //
L2: 0.510799618095358 em tanque // em série
C2: 1.420300651398807 em tanque // em série
# Admitância:
Frequência: (Re Im ou nome) z1
C3: 2.154363713489590 em //
L4: 1.006392097912381 em tanque // em série
C4: 0.287867290054310 em tanque // em série
# Admitância:
Frequência: (Re Im ou nome) z3
C5: 2.154363713489691 em //
L6: 0.510799618095558 em tanque // em série
C6: 1.420300651398251 em tanque // em série
# Admitância:
C7: 1.292597534874345 em //
* Síntese completa *
```

Os escalamentos em impedância e frequência são feitos agora: [N][I], [N][F].

Fator de escalamento em impedância: 50
Fator de escalamento em frequência: 125663.7062

É aconselhável salvar redes não normalizadas com valores na notação científica, para maior precisão: [M].

Número de decimais: -1

A rede final gerada é então listada [E] (opcional) e salva [S]:

```
C1: 2.05723287011E-0007 em //
L2: 2.03240710282E-0004 em tanque // em série
C2: 2.26047869245E-0007 em tanque // em série
C3: 3.42877634066E-0007 em //
L4: 4.00430692499E-0004 em tanque // em série
```

C4: 4.58155021460E-0008 em tanque // em série
C5: 3.42877634066E-0007 em //
L6: 2.03240710282E-0004 em tanque // em série
C6: 2.26047869245E-0007 em tanque // em série
C7: 2.05723287011E-0007 em //

#

Arquivo onde salvar: ex.val

Faltam apenas as terminações, que são ambas de 50 Ohms. Para simulação da rede, o arquivo gerado pode ser usado como lista de valores no programa EDFIL. A estrutura "mid-series" da rede é óbvia pelos comentários.

Obs: Os valores numéricos no exemplo podem não ser exatamente os mesmos obtidos com versões mais recentes dos programas, onde alguns métodos numéricos foram modificados para maior precisão. O formato das listagens pode ser diferente também.

Sobre as versões para Windows: Os programas vem sendo convertidos para Windows, pois os antigos programas de DOS não rodam mais diretamente a partir do Windows 7. Os novos programas retêm a mesma funcionalidade, com algumas alterações na "interface" com o usuário, e algumas novas funções.

Antônio Carlos Moreirão de Queiroz - 8/8/2015

E-mail: acmq@ufrj.br