HP-41 EE Filters Module

Overview

This module comprises a selection of programs and functions loosely grouped around the EE Filters theme. It's meant to complement the HP EE Solutions Book, notably the hp-41 version of the Butterworth & Chebyshev's Filters programs published by T. Mickelson in PPCCJ V7N6. Other programs have been taken from the PRISMA magazine, the official publication of the CCD German user's Club.

Amongst the new contents, MCODE functions to calculate the Elliptic Integrals of first and second order are also included, together with one example of usage (Mutual inductance between 2 coaxial coils). Note that these functions are based on the Arithmetic-Geometric Mean and its extension, resulting in a much faster execution than the traditional approach based on Hypergeometric Functions.

The table below shows the function names in alphabetical order with a brief description. The Authors and sources are listed below for a complete program description and user instructions in case you're interested.

XROM	Function	Description	Author	Source
17,00	-OPAMP 41	*Section Header*	*Ángel Martin*	*This project*
17,01	AGM	*Arithmetic-Geometric Mean*	*Ángel Martin*	*SandMath project*
17,02	AGM2	*Extended AGM*	*Ángel Martin*	*Paper by Semjon Adlaj*
17,03	ELIPE	*Elliptic Integral 2nd Order*	*Ángel Martin*	*This project – see below*
17,04	ELIPK	*Elliptic Integral 1st Order*	*Ángel Martin*	*This project – see below*
17,05	"EK"	*Elliptic Integral 2nd Order*	*Ángel Martin*	*This project – see below*
17,06	"KK"	*Elliptic Integral 1st Order*	*Ángel Martin*	*This project – see below*
17,07	"MIND"	*Mutual Inductance*	*Ángel Martin*	*NASA SP-42 document*
17,08	"NOIZ"	*Noise Factor*	*Unknown?*	*Best of Prisma*
17,09	"OPGOFF"	*OpAmp Gain and Offset*	*Stefan Vorkoetter*	*Author's Web pages*
17,10	"OPOSC"	*OpAmp Oscillations*	*Stefan Vorkoetter*	*Author's Web pages*
17,11	"SLNKEY"	*Sullen-Key Filter Design*	*Stefan Vorkoetter*	*Author's Web pages*
17,12	"="	*Builds Prompt in ALPHA*	*Ángel Martin*	*This project*
17,13	-41 FILTERS	*Section Header*	*Ángel Martin*	*This project*
17,14	ACOSH	*Inverse Hyperbolic COS*	*Ángel Martin*	*SandMath project*
17,15	ASINH	*Inverse Hyperbolic SIN*	*Ángel Martin*	*SandMath project*
17,16	"APPROX"	*Digital Filter Approximation*	*Thomas Wegmann*	*Prisma 1985.V3.p31*
17,17	"BC"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,18	"C3"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,19	"C?"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,20	"FB"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,21	"N"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,22	"O"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,23	"T"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,24	"Z"	*BW & Chebyshev Filters*	*Terry Mickelson*	*PPCCJ V9N2p20*
17,25	"HP2"	*High-Pass 2nd order*	*Heinrich Henze*	*Best of Prisma*
17,26	"LP2"	*Low-Pass 2nd order*	*Heinrich Henze*	*Best of Prisma*
17,27	dB-	*Decibel Subtraction*	*Ángel Martin*	*ETSII Project*
17,28	dB +	*Decibel Addition*	*Ángel Martin*	*ETSII Project*

Below is a brief description for some of the functions of the module for your convenience – You're encouraged to check the documentation provided in the linked documents for all the other functions and programs; in particular Stefan Vorkoetter's pages are world-class documentation worth checking out.

ACOSH and ASINH

These functions calculate the main values of the argument of the hyperbolic sine and cosine, as defined by the well-known relationships:

asinh (x) = Ln [x + SQRT(x^2 +1)]

acosh (x) = Ln [x + SQRT(x^2 – 1)]

They are used in the Chebyshev's filter design by approximation (program "APPROX")

AGM and AGM2

These functions calculate the standard and extended arithmetic-geometric means of two values, stored in the X and Y registers.

The result is left in the X register, and the second argument (originally in X) is saved in LastX.

STACK	INPUT/OUTPUT	OUTPUT/INPUT
Y	y	z
X	x	AGM(y,x) / AGM2(y,x)
L	-	x

Example1: Calculate the AGM of 17 and 23:

17, ENTER^, 23, XEQ "AGM" => 19.88669905

Example 2: Calculate the extended AGM of 1 and 2:

1, ENTER^, 2, XEQ "AGM2" => 1,456946581

ELIPK and ELIPE

These functions calculate the Elliptic Integrals of first and second orders; K(x) and E(x) respectively.

The algorithms are based on the mathematical relationships involving the arithmetic-geometric means, which provide a much faster and slightly more accurate approach than those using hyper-geometric functions – a double-bonus for a 41 implementation. No data registers are used but contents of Alpha registers {O,P} are lost.

STACK	INPUT/OUTPUT	OUTPUT/INPUT
Y	y	y
X	x	E(x) / K(x)
L	-	x

Note that the argument x must be ***positive and less than or equal to 1*** – otherwise a DATA ERROR condition will occur.

Example: calculate the difference between the values of both integrals for the argument x=0.5

0.5, XEQ "ELIPK" => 1,854074677
LASTX, XEQ "ELIPE" => 1,350643881
[-] => 0,503430796

EK and KK

These two are FOCAL counterparts of the MCODE functions described above. Besides relative comparison purposes, they are included for you to see the actual coding of the functions, showing the usage of the AGM functions. The program is very short, and is listed below:

Line	Instruction	Line	Instruction	Line	Instruction
01	**LBL "KK"**	08	**AGM**	15	**XROM "KK"**
02	CHS	09	ST+ X	16	RCL O
03	E	10	1/X	17	X^2
04	+	11	PI	18	E
05	SQRT	12	*	19	**AGM2**
06	STO O	13	RTN	20	*
07	E	14	**LBL "EK"**	21	END

Contrary to their MCODE counterparts, these functions will not preserve the stack nor will save the original argument into the LastX register.

The formulas used are as follows – note the (pesky) convention on the argument notation, which is *not* squared:

$$K(x) = \pi / \{2\ AGM\ [1,(1-x)]\}$$

$$E(x) = K(x) * AGM2\ [1, (1-x)]$$

MIND: Mutual inductance of coaxial coils

This short program perfectly showcases the usefulness of the elliptic integral functions described above. Use it to calculate the mutual inductance between two coaxial coils of radius R1 and R2, separated a distance “d”. As it’s known, the formula for the mutual induction (in Henries) can be expressed as:

$$M = (8\pi / k\ 10^{\wedge}7)\ \ SQRT\ (r1 * r2)\ [\ (1-m/2)\ K(m) - E(m)]$$

Where: $m = k^{\wedge}2 = 4\ R1\ R2 / [(R1+R2)^{\wedge}2 + d^{\wedge}2]$

Example: Calculate the mutual inductance for the test case r1=0.2; r2=0,25; and d=0.1 (all distances in meters)

MIND = 2,48787 E-07 Henry