MATLAB™ and MATHEMATICA™ Programs to Compute the Coefficients Associated to Certain Incomplete Lipschitz-Hankel Integrals with Applications to Wireless Communication Theory

José F. Paris and David Morales-Jiménez

Dept. of Ingeniería de Comunicaciones, E.T.S.I. de Telecomunicación,
Universidad de Málaga, Málaga,
E-29071, Spain
paris@ic.uma.es, morales@ic.uma.es

Abstract

This brief technical note contains MATLAB™ and MATHEMATICA™ recursive programs which are useful to compute certain incomplete Lipschitz-Hankel integrals. Such integrals have applications within the context of wireless communication theory.

Index Terms

MATLAB™, MATHEMATICA™, Incomplete Lipschitz-Hankel Integrals, Wireless Communications Theory.

I. Introduction

Certain class of performance analysis problems in the scope of wireless communications theory [1]-[4] require to evaluate the following incomplete Lipschitz-Hankel integrals (ILHI)

$$I_{e_{m,n}}(x;\alpha) \triangleq \int_0^x t^m e^{-\alpha t} I_n(t)\, dt, \tag{1}$$

where $\alpha \in \mathbb{R}$, $m, n \in \mathbb{N}$; $\alpha > 1$ and $x \in [0, \infty)$.

Closed-forms expressions for the integrals defined in (1) were derived in [2] in terms of the Marcum Q function and the modified Bessel functions. Such results are contained in the following Lemma and Proposition.

*This work is partially supported by the Spanish Government under project TEC2007-67289/TCM and by AT4 wireless.

Lemma 1: The ILHIs for low degrees and orders are given by

(i) $I_{e_{0,0}}(x;\alpha) = \bar{\alpha} - 2\bar{\alpha}Q_1\left(\frac{\sqrt{x}}{\sqrt{\alpha+\sqrt{\alpha^2-1}}}, \sqrt{x}\sqrt{\alpha+\sqrt{\alpha^2-1}}\right) + \bar{\alpha}e^{-\alpha x}I_0(x)$

(ii) $I_{e_{1,0}}(x;\alpha) = \alpha\bar{\alpha}^3 - 2\alpha\bar{\alpha}^3 Q_1\left(\frac{\sqrt{x}}{\sqrt{\alpha+\sqrt{\alpha^2-1}}}, \sqrt{x}\sqrt{\alpha+\sqrt{\alpha^2-1}}\right) + \alpha\bar{\alpha}^3 e^{-\alpha x}I_0(x) - \alpha\bar{\alpha}^2 e^{-\alpha x}xI_0(x)$
$-\bar{\alpha}^2 e^{-\alpha x}xI_1(x)$

(iii) $I_{e_{0,1}}(x;\alpha) = \alpha\bar{\alpha} - 1 - 2\alpha\bar{\alpha}Q_1\left(\frac{\sqrt{x}}{\sqrt{\alpha+\sqrt{\alpha^2-1}}}, \sqrt{x}\sqrt{\alpha+\sqrt{\alpha^2-1}}\right) + (1+\alpha\bar{\alpha})\,e^{-\alpha x}I_0(x)$

(iv) $I_{e_{1,1}}(x;\alpha) = \bar{\alpha}^3 - 2\bar{\alpha}^3 Q_1\left(\frac{\sqrt{x}}{\sqrt{\alpha+\sqrt{\alpha^2-1}}}, \sqrt{x}\sqrt{\alpha+\sqrt{\alpha^2-1}}\right) + \bar{\alpha}^3 e^{-\alpha x}I_0(x) - \bar{\alpha}^2 e^{-\alpha x}xI_0(x)$
$-\alpha\bar{\alpha}^2 e^{-\alpha x}xI_1(x)$

where $\bar{\alpha} \triangleq \frac{1}{\sqrt{\alpha^2-1}}$.

Proof: See Appendix II in [2]. ■

Proposition 1: The mth-degree nth-order ILHI is represented by an expression involving a first-order Marcum Q function and a finite number of Bessel functions, specifically

$$I_{e_{m,n}}(x;\alpha) = \mathcal{A}^0_{m,n}(\alpha) + \mathcal{A}^1_{m,n}(\alpha)Q_1\left(\frac{\sqrt{x}}{\sqrt{\alpha+\sqrt{\alpha^2-1}}}, \sqrt{x}\sqrt{\alpha+\sqrt{\alpha^2-1}}\right) + \mathrm{e}^{-\alpha x}\sum_{i=0}^{m}\sum_{j=0}^{n+1}\mathcal{B}^{i,j}_{m,n}(\alpha)x^i I_j(x). \tag{2}$$

where the set of coefficients $\mathcal{A}^l_{m,n}(\alpha), \mathcal{B}^{i,j}_{m,n}(\alpha)$ can be obtained recursively.

Proof: See Appendix III in [2]. ■

The ILHI coefficients $\mathcal{A}^l_{m,n}(\alpha), \mathcal{B}^{i,j}_{m,n}(\alpha)$ are computed in exact manner by a set of linear recursive equations defined in the Appendix III of [2].

This technical note provides two simple recursive programs for MATLAB™ and MATHEMATICA™ which allow us to efficiently compute the ILHI coefficients needed to evaluate (1).

II. MATLAB™ Program to Compute the ILHI Coefficients

The following code has been successfully tested to compute the ILHI coefficients using MATLAB™ 7.5

```
function [A0,A1,B]=coeff_ILHI(m,n,a)
b=1/sqrt(a^2-1);
A0=0; A1=0;
B=zeros(m+1,n+2);
```

```
if      [m,n]==[0,0],
         A0= b;
         A1= -2*b;
         B=[b 0];
elseif  [m,n]==[1,0],
         A0= a*b^3;
         A1= -2*a*b^3;
         B=[a*b^3 0; -a*b^2 -b^2];
elseif  [m,n]==[0,1],
         A0= a*b-1;
         A1= -2*a*b;
         B=[1+a*b 0 0];
elseif  [m,n]==[1,1],
         A0= b^3;
         A1= -2*b^3;
         B=[b^3 0 0 ; -b^2 -a*b^2 0];
elseif  [m,n]==[1,2],
         A0= a*b^3    -2*(a*b-1);
         A1= -2*a*b^3 -2*(-2*a*b);
         B=[a*b^3-2*(1+a*b) 0 0 0; -a*b^2 -b^2 0 0];
else
         if (m==0)
             [A0_n_2, A1_n_2, B_n_2]=coeff_ILHI(0,n-2,a);
             [A0_n_1, A1_n_1, B_n_1]=coeff_ILHI(0,n-1,a);
             A0= 2*a*A0_n_1 -A0_n_2;
             A1= 2*a*A1_n_1 -A1_n_2;
             for j=0:n-2,
                 B(1+ 0, 1+ j  )= 2*a*B_n_1(1+ 0, 1+ j) -B_n_2(1+ 0, 1+ j);
             end
                 B(1+ 0, 1+ n-1  )= 2*a*B_n_1(1+ 0,1+ n-1) + 2;
         elseif (m==1)
             [A0_n_2, A1_n_2, B_n_2]=coeff_ILHI(1,n-2,a);
             [A0_n_1, A1_n_1, B_n_1]=coeff_ILHI(1,n-1,a);
             A0= 2*a*(n-1)/(n-2)*A0_n_1 -(n)/(n-2)*A0_n_2;
             A1= 2*a*(n-1)/(n-2)*A1_n_1 -(n)/(n-2)*A1_n_2;
             for i=0:1,
             for j=0:n-2,
                 B(1+ i, 1+ j  )= 2*a*(n-1)/(n-2)*B_n_1(1+ i, 1+ j) ...
                        -(n)/(n-2)*B_n_2(1+ i, 1+ j);
             end
```

```
            end
                B(1+ 0, 1+ n-1  )= 2*a*(n-1)/(n-2)*B_n_1(1+ 0, 1+ n-1) ;
                B(1+ 1, 1+ n-1  )= 2*a*(n-1)/(n-2)*B_n_1(1+ 1, 1+ n-1) +2*(n-1)/(n-2);
        else
            [A0_m_2, A1_m_2, B_m_2]=coeff_ILHI(m-2,n,a);
            [A0_m_1, A1_m_1, B_m_1]=coeff_ILHI(m-1,n,a);
            A0= a*(2*m-1)/(a^2-1)*A0_m_1 +(n^2-(m-1)^2)/(a^2-1)*A0_m_2;
            A1= a*(2*m-1)/(a^2-1)*A1_m_1 +(n^2-(m-1)^2)/(a^2-1)*A1_m_2;
            for i=0:m-2,
                for j=0:n+1,
                    B(1+ i, 1+ j  )= a*(2*m-1)/(a^2-1)*B_m_1(1+ i, 1+ j) ...
                        +(n^2-(m-1)^2)/(a^2-1)*B_m_2(1+ i, 1+ j);
                end
            end
            for j=0:n+1,
                B(1+ m-1, 1+ j  )= a*(2*m-1)/(a^2-1)*B_m_1(1+ m-1, 1+ j) ...
                    +(m+n-1)/(a^2-1)*delta(j-n);
            end
            if (n==0)
                for j=0:n+1,
                    B(1+ m, 1+ j  )= -a/(a^2-1)*delta(j-0)-1/(a^2-1)*delta(j-1);
                end
            else
                for j=0:n,
                    B(1+ m, 1+ j  )= -a/(a^2-1)*delta(j-n)-1/(a^2-1)*delta(j-(n-1));
                end
            end
        end
 end
function y=delta(k)
if k==0,
    y=1;
else
    y=0;
end
```

III. MATHEMATICA™ PROGRAM TO COMPUTE THE ILHI COEFFICIENTS

The following code has been successfully tested to compute the ILHI coefficients using MATHEMATICA™ 7.0.

```
coeffILHI[mm_,nn_,aa_]:= Module[{m=mm,n=nn,a=aa,\[Delta],b,
a0,a1,bb,a0n1,a0n2,
a1n1,a1n2,bbn1,bbn2,i,j,a0m1,a1m1,bbm1,a0m2,a1m2,bbm2},
\[Delta][k_]:=KroneckerDelta[k];
b=1/Sqrt[a^2-1];a0=0;a1=0; bb=ConstantArray[0,{m+1,n+2}]; Which[
{m,n}=={0,0}, a0=b;a1=-2 b;bb={{b,0}}, {m,n}=={1,0}, a0=a
b^3;a1=-2 a b^3;
              bb={{a b^3,0},{-a b^2,-b^2}},
{m,n}=={0,1}, a0=a b-1;a1=-2 a b;
              bb={{1+a b,0,0}},
{m,n}=={1,1}, a0=b^3;a1=-2 b^3;
              bb={{b^3,0,0},{-b^2,-a b^2,0}},
{m,n}=={1,2}, a0=a b^3-2 (a b-1);a1=-2 a b^3-2 (-2 a b);
              bb={{a b^3-2 (1+a b),0,0,0},{-a b^2,-b^2,0,0}},
m==0,
    {a0n1,a1n1,bbn1}=coeffILHI[0,n-1,a];{a0n2,a1n2,bbn2}=coeffILHI[0,n-2,a];
    a0=2 a a0n1-a0n2;a1=2 a a1n1-a1n2;
    Do[bb[[1+0,1+j]]=
    2 a bbn1[[1+0,1+j]]-bbn2[[1+0,1+j]],{j,0,n-2}];
    bb[[1+0,1+n-1]]=2 a bbn1[[1+0,1+n-1]]+2,
m==1,
    {a0n1,a1n1,bbn1}=coeffILHI[1,n-1,a];{a0n2,a1n2,bbn2}=coeffILHI[1,n-2,a];
    a0=2 a (n-1 )/(n-2) a0n1-n/(n-2) a0n2;
    a1=2 a (n-1 )/(n-2) a1n1-n/(n-2) a1n2;
    Do[bb[[1+i,1+j]]=
    2 a (n-1 )/(n-2) bbn1[[1+i,1+j]]-n/(n-2) bbn2[[1+i,1+j]],{i,0,1},{j,0,n-2}];
    bb[[1+0,1+n-1]]=2 a (n-1 )/(n-2) bbn1[[1+0,1+n-1]];
    bb[[1+1,1+n-1]]=2 a (n-1 )/(n-2) bbn1[[1+1,1+n-1]]+2 (n-1 )/(n-2),
m>=2,
    {a0m1,a1m1,bbm1}=coeffILHI[m-1,n,a];{a0m2,a1m2,bbm2}=coeffILHI[m-2,n,a];
    a0=a b^2 (2m-1)a0m1+b^2 (n^2-(m-1)^2)a0m2;
    a1=a b^2 (2m-1)a1m1+b^2 (n^2-(m-1)^2)a1m2;
    Do[bb[[1+i,1+j]]=
    a b^2 (2m-1) bbm1[[1+i,1+j]]+b^2 (n^2-(m-1)^2)bbm2[[1+i,1+j]],{i,0,m-2},{j,0,n+1}];
    Do[bb[[1+m-1,1+j]]=
    a b^2 (2m-1) bbm1[[1+m-1,1+j]]+b^2 (m+n-1)\[Delta][j-n],{j,0,n+1}];
```

```
    If[n==0,
    Do[bb[[1+m,1+j]]=
    -a b^2 \[Delta][j-0]-b^2 \[Delta][j-1],{j,0,n+1}],
    Do[bb[[1+m,1+j]]=
    -a b^2 \[Delta][j-n]-b^2 \[Delta][j-(n-1)],{j,0,n}]]];
{a0,a1,bb}];}
```

IV. Conclusions

In this technical note, computer programs are provided to evaluate the coefficients associated to certain incomplete Lipschitz-Hankel integrals which appear in wireless communication theory.

References

[1] J. F. Paris, "Nakagami-*q* (Hoyt) distribution function with applications," *Electron. Lett.*, vol. 45, no. 4, pp. 210–211, Feb. 2009.

[2] J. F. Paris, E. Martos-Naya, U. Fernández-Plazaola, J. López-Fernández, "Analysis of Adaptive MIMO Beamforming under Channel Prediction Errors based on Incomplete Lipschitz-Hankel Integrals," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2815–2824, July 2009.

[3] D. Morales-Jiménez, J. F. Paris, "Closed-form analysis of dual-branch switched diversity with binary nonorthogonal signalling," *Electronics Letters*, vol. 45, no. 23, pp. 1179–1180, November 2009.

[4] D. Morales-Jiménez, J. F. Paris, "Outage Probability Analysis for Nakagami-q (Hoyt) Fading Channels under Rayleigh Interference," *IEEE Transactions on Wireless Communications*, pp. 1272-1276 , April 2010.