APPLIED DOMINANT TRANSFER AND FUZZY LOGIC FOR NONLINEAR WAVE-WAVE INTERACTIONS

Adhi Susilo, Will Perrie, Matiur Rahman

ABSTRACT: Third generation wave modeling demands an accurate computation of the nonlinear wave-wave interactions. The latter play an important role in the evolution of wind waves, representing a mechanism for shifting wave energy to lower and higher frequencies within the spectrum. The original formulation for wave-wave interactions was proposed by Hasselmann (1962) more than four decades ago. However, the implementation of this method is time consuming and not practical. Using estimated dominant transfer as representative of the whole spectrum, Susilo and Perrie (2006) developed a new method called *Dominant Transfer Approximation*, DTA. This method does not need the entire spectrum but uses only a specific wave number which gives the dominant transfer and a scaling factor, F_d, which is denoted the 'dominant factor'. Some experiments must be performed to determine the F_d. To install the DTA method in a wave model a reliable program to compute F_d must be established. In this study, we try to address this problem using fuzzy logic.

***AMS Mathematical Subject Classification*:** 03B52, 68T37, 76B15, 76M15

***Keywords and phrases*:** wind waves, nonlinear interactions, quadruplets, dominant transfer, dominant factor, fuzzy logic.

1. INTRODUCTION

The energy conservation relation for wind-generated waves in the deep water usually is stated as

$$\frac{dE}{dt} = S_{inp} + S_{nl4} + S_{wcap} \tag{1}$$

where S_{inp} is the energy input by wind, S_{nl4} is the non-linear quadruplet wave-wave interactions, and S_{wcap} represents the energy dissipation by white-capping and wave breaking.

The basic equation describing S_{nl4} is the Boltzmann integral or kinetic equation, proposed by Hasselmann (1962). Although Hasselmann and Hasselmann (1981) developed their pioneering systematic computation of S_{nl4}, the so-called the exact method is not practical for operational applications, because it is hampered by the complexity of the functional form. Its computation is several orders of magnitude more expensive than all other terms in equation (1).

More recently, Hasselmann *et al.* (1985) developed the Discrete Interaction Approximation, DIA, which dramatically increased the computational speed. DIA enabled the development of third generation wave prediction models such as WAM and SWAN. However, DIA has a number of shortcomings. For example it compares poorly with full integrations of S_{nl4} for many types of spectra (Van Vledder, 2000).

Improvements to the full nonlinear computation method have been attempted, for example by Lin and Perrie (1999), and the DTA method by Susilo and Perrie (2006). The latter combines the full S_{nl4} formulation with a simple scaling algorithm. However this method relies on finding an accurate dominant factor F_d. The present study attempts to compute F_d with fuzzy logic (FL).

2. DOMINANT FACTOR

The S_{nl4} term is the rate of change of action density at a particular wave number due to resonant interactions between quadruplets of wave numbers. This is given by the following six-fold Boltzmann integral:

$$\frac{dn_1}{dt} = \iiint C(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4)[n_1 n_3 (n_4 - n_2) + n_2 n_4 (n_3 - n_1)]$$

$$\delta(\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 - \mathbf{k}_4)\ \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4) d\mathbf{k}_2 d\mathbf{k}_3 d\mathbf{k}_4\ . \tag{2}$$

Equation (2) describes the rate of change of n_1 at wave number $\mathbf{k}_1$ due to all quadruplet interactions involving $\mathbf{k}_1$. Webb (1978) introduced a transfer function $T(\mathbf{k}_1, \mathbf{k}_3)$

$$\frac{dn_1}{dt} = \int T(\mathbf{k}_1, \mathbf{k}_3)\ d\mathbf{k}_3 \tag{3}$$

where

$$T(\mathbf{k}_1, \mathbf{k}_3) = \iint C(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4 [n_1 n_3 (n_4 - n_2) + n_2 n_4 (n_3 - n_i)]$$

$$\delta(\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 - \mathbf{k}_4)\ \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4)$$

$$\Theta(|\mathbf{k}_1 - \mathbf{k}_4| - |\mathbf{k}_1 - \mathbf{k}_3|)\ d\mathbf{k}_2 d\mathbf{k}_4 \tag{4}$$

where n_i is the action density $n(k_i)$ at wave number k_i, ω_i is the angular frequency at k_i, the $\delta(. . .)$ is the Dirac delta function and the term C is the coupling coeficient (Webb, 1978; Tracy and Resio, 1982) and

$$\Theta(x) = 1 \text{ if } x > 0$$

$$\Theta(x) = 0 \text{ if } x \leq 0$$

$$x = |\mathbf{k}_1 - \mathbf{k}_4| - |\mathbf{k}_1 - \mathbf{k}_3|. \tag{5}$$

Applying resonance conditions, $\omega_1 + \omega_2 = \omega_3 + \omega_4$ and $\mathbf{k}_1 + \mathbf{k}_2 = \mathbf{k}_3 + \mathbf{k}_4$, equation (4) can be solved. Finally, in polar coordinates system the nonlinear energy transfer, equation (3), can be computed from the following equation,

$$\frac{dn_1}{dt} = \int_0^\infty \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_3)\ k_3 d\theta_3 dk_3. \tag{6}$$

This implies that the nonlinear energy transfer must be computed over the entire 2-dimensional spectrum. However, the DTA formulation selects a set of (k_1, k_3) which gives a maximum transfer and uses a multiplication factor or a scaling factor to approximate the integral over the 2-dimensional spectrum. Equation (6) can be rewritten as:

$$\frac{dn_1}{dt} \approx F_d \int_0^{2\pi} T(\mathbf{k}_1, \mathbf{k}_{3_d})\ \Delta k_3 d\theta_{3_d} d\theta_3 \tag{7}$$

where F_d is a scaling factor, the so-called dominant factor and $(\mathbf{k}_1, \mathbf{k}_{3_d})$ is a set of k_1 and k_3 where maximum transfer occurs.

3. FINDING DOMINANT FACTOR, F_d

As shown by Susilo and Perrie (2006), F_d is a function of peakedness (γ) and spectral spreading factor, for a given spectrum. Thus, there are two inputs to get one output. Because peakedness and spectral spreading are usually only defined once at the initialization of a simulation, we need to specify these variables so they can be determined at any time.

We define new variables, $slope_g$ to represent peakedness and $slope_s$ to represent the spectral spreading,

$$slope_g = \frac{1 - \frac{n(k_{\max+1}, \theta_{\max})}{n(k_{\max}, \theta_{\max})}}{(\lambda - 1) k_{\max}} \tag{8}$$

$$slope_s = \frac{1 - \frac{n(k_{\max}, \theta_{\max+1})}{n(k_{\max}, \theta_{\max})}}{\Delta\theta k_{\max}} \tag{9}$$

where n is the action density, λ is the incremental factor, $(k_{\max}, \theta_{\max})$ is the location of (k, θ) where n is maximum. See Figure 1 for detail.

The next step is to find a method to compute the $F_d(slope_g, slope_s)$. We choose the fuzzy logic (FL) method as a tool to solve this problem. Using if-then rules, FL provides a simple way to determine F_d based on some numerical experiments that have to be done. On the other hand, a number, such as F_d, also can be defined by FL (Kosko, 1993). For instance, a number 0.9 can be defined as 90% of unity or 10% of zero (see schematic model suggested in Figure 2). Thus, we apply FL as a methodology to determine F_d.

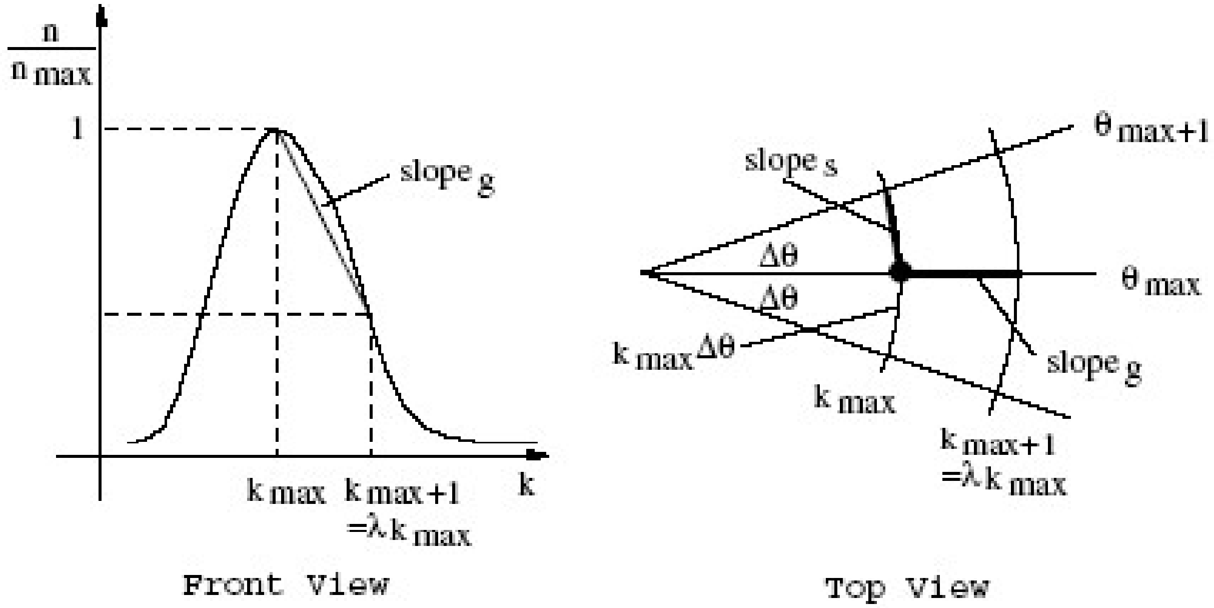


Figure 1: The *slope*$_g$ and *slope*$_s$.

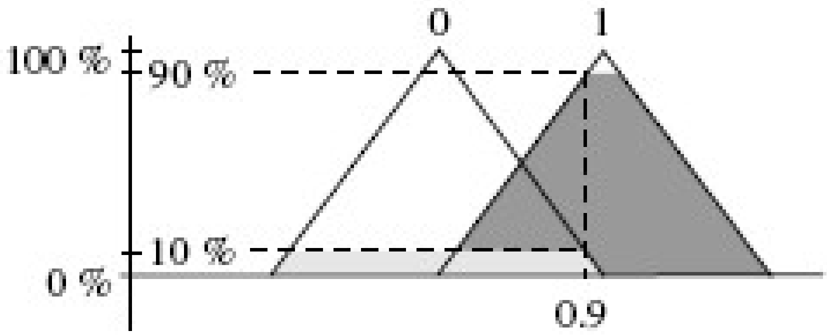


Figure 2: A Number Defined by FL.

Before we discuss this method further, we would like to review the fuzzy logic method. Going back a couple of millennia, Aristotle formulated the idea of a bivalent value (two crisps value or crisp set), for example *A* or not-*A*, alternately, an apple or not an apple. Aristotle's teaching is basic to the digital era, consisting of 0 or 1. However, not all cases can be analyzed by bivalent values. Let's eat the apple bit by bit. Finally there is no apple anymore. You see, we have gone from apple to nothing. When you have half an apple, is it an apple or nothing (not-apple) ? Given a glass that is half full of water, is it half full or half empty ? In this case, we can not say '*A* or not-*A*' but we have to say '*A* and not-*A*'. Now we are entering a new concept of multivalence, commonly called fuzzy logic. This is a system of logic in which a statement can be stated as a continuum of values in between 'false or true' or '0 or 1'. In the 1960s, Prof. Lotfi A. Zadeh at the University of California Berkeley, introduced the fuzzy set. But this idea originally received a huge amount of criticism. However, in the next decade he continually broadened the fuzzy set theory. In 1974, Mamdani developed a controller for the steam engine based on a fuzzy algorithm from Zadeh's paper. Since then, applications of the fuzzy set or fuzzy logic in industrial settings have boomed, especially in Japan and Europe. More history about fuzzy logic can be found in Yen and Langari (1999).

Next, we are going to review the crisp and fuzzy set in more detail. Table 1 shows some differences between bivalence and multivalence as listed by Kosko (1993).

Table 1
Bivalence vs Multivalence

BIVALENCE	*MULTIVALENCE*
A or not-*A*	*A* and not-*A*
all or none	some degree
0 or 1	continuum between 0 and 1

Figure 3 depicts a diagram of bivalence (or crisp set) and multivalence (fuzzy set) and its membership function, μ, or it is called grade of belonging. In crisp set, a membership of element *x* of set *A* or not - *A* is

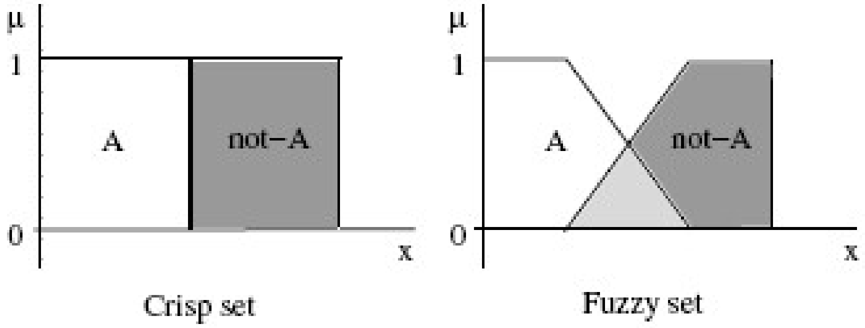


Figure 3: Bivalence and multivalence.

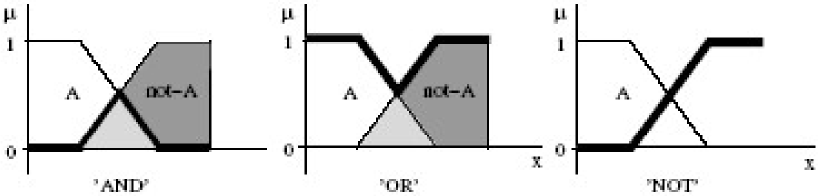


Figure 4: Logic of fuzy set shown in bold line.

defined by:

$$\mu_A(x) = \begin{cases} 0, & \text{if } x \notin A \\ 1, & \text{if } x \in A \end{cases}$$

$$\mu_{not-A}(x) = \begin{cases} 0, & \text{if } x \notin not-A \\ 1, & \text{if } x \in not-A \end{cases}$$

However, in fuzzy set theory, the membership μ is defined by $\mu_A(x) \in [0, 1]$ or $\mu_{not\text{-}A}(x) \in [0, 1]$, that expresses the degree to which *x* belongs to *A* or *not* – A.

As a membership function differs between two sets, the logic of sets is also different.

Crisp logic defines:

- AND : $A \cap B$
- OR : $A \cup B$
- NOT : A'.

Fuzzy logic defines (the diagrams are shown in Figure 4):

- AND : $\mu_{A \cap B} = \min[\mu_A(x), \mu_B(x)]$
- OR : $\mu_{A \cap B} = \max[\mu_{\mathrm{A}}(x), \mu_B(x)]$
- NOT : $\mu_{A'} = 1 - \mu_A(x)$.

An example showing how the fuzzy logic formulation works can be depicted as shown in Figures 5-6. The *Fuzzifier* will compute the membership of inputs, then the *Interencer* will denote the membership of outputs based on the logic of the fuzzy set. The *Interencer* also aggregate all outputs. Of course, in the end we need a

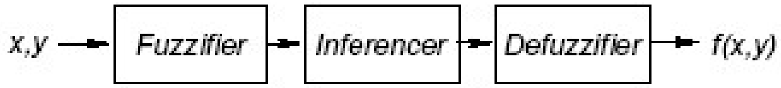


Figure 5: Fuzzy controller.

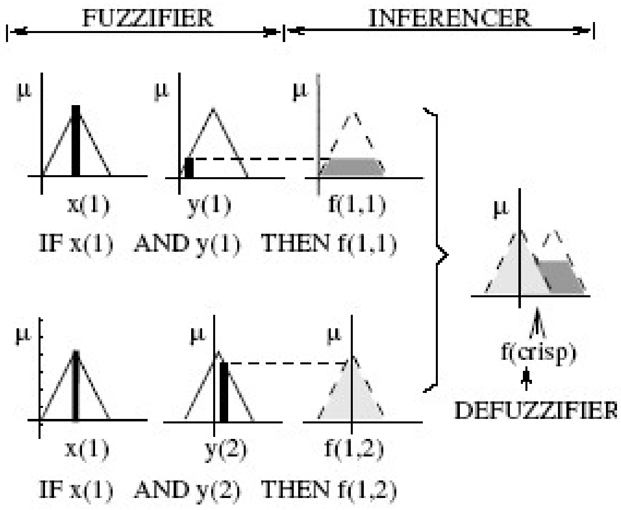


Figure 6: Example of fuzzy controller

single value or crisp value, not a multivalence value, and this is a job that is specified by the *Defuzzifier* when it determines the crisp value.

In this paper, a fuzzy controller gives one output from two inputs. The x and y represent $slope_g$ and $slope_s$ while $f(x, y)$ represents $F_d(slope_g, slope_s)$ respectively. Case examples are conducted to collect the FL required data, then sets of rules are determined. Samples are taken, representing windsea as well as swell cases, for peakedness varying as = 1, 3, 5, and 7 and spectral spreading varying as $\cos^{2p}\theta$ where $p = 1, 2$, and 6.

Table 2 shows data from computed samples. Before establishing FL rules, it is convenient to relabel each case with a simple naming convention. We classify as '*Tiny*', '*Small*', '*Big*' and '*Large*' representing small $slope_g$ to large $slope_g$ and we also classify varying spreading factor as '*Low*', '*Mid*', and '*High*' for small $slope_s$ to large $slope_s$ conditions, respectively, in Table 3. From Table 3, we define the rules as shown on Table 4. Membership functions of Table 4 can be depicted as shown in Figure 7.

Table 2
F_d from computed samples.

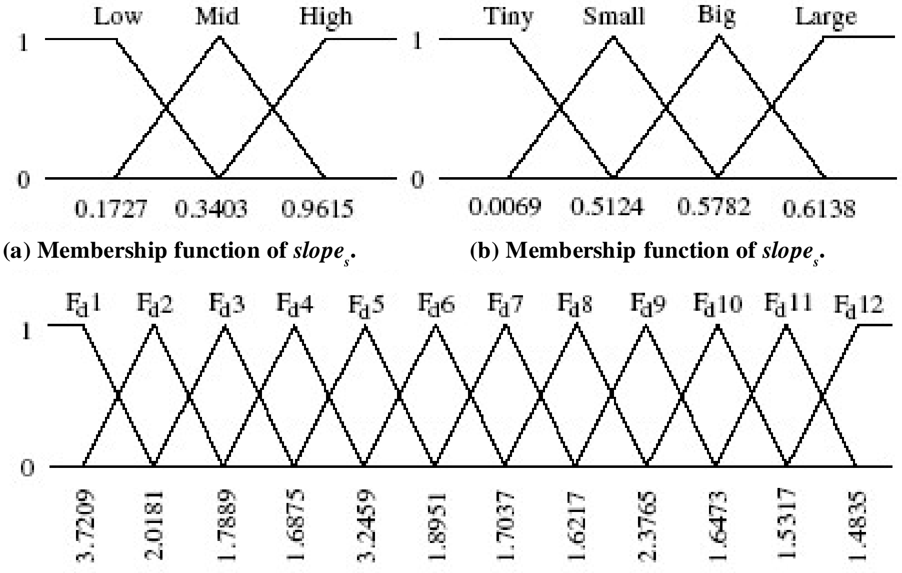
F_d		$slope_g$			
		0.16144	*8.14315*	*9.18954*	*9.75586*
$slope_s$	0.69768	3.7209	2.0181	1.7889	1.6875
	1.37433	3.2459	1.8951	1.7037	1.6217
	3.88293	2.3765	1.6473	1.5317	1.4835

Table 3
Rewriting matrix of F_d in simplified form.

	Tiny	*Small*	*Big*	*Large*
Low	F_d1	F_d2	F_d3	F_d4
Mid	F_d5	F_d6	F_d7	F_d8
High	F_d9	F_d10	F_d11	F_d12

Table 4
If-then FL rules from study cases.

1.	If *Low*	AND	*Tiny*	then F_d1
2.	If *Low*	AND	*Small*	then F_d2
3.	If *Low*	AND	*Big*	then F_d3
4.	If *Low*	AND	*Large*	then F_d4
5.	If *Mid*	AND	*Tiny*	then *Fd*5
6.	If *Mid*	AND	*Small*	then F_d6
7.	If *Mid*	AND	*Big*	then F_d7
8.	If *Mid*	AND	*Large*	then F_d8
9.	If *High*	AND	*Tiny*	then F_d9
10.	If *High*	AND	*Small*	then F_d10
11.	If *High*	AND	*Big*	then F_d11
12.	If *High*	AND	*Large*	then F_d12



(a) Membership function of *slope*$_s$. **(b) Membership function of *slope*$_s$.**

(c) Membership function of *Fd*.

Figure 7: FL membership function of inputs and output.

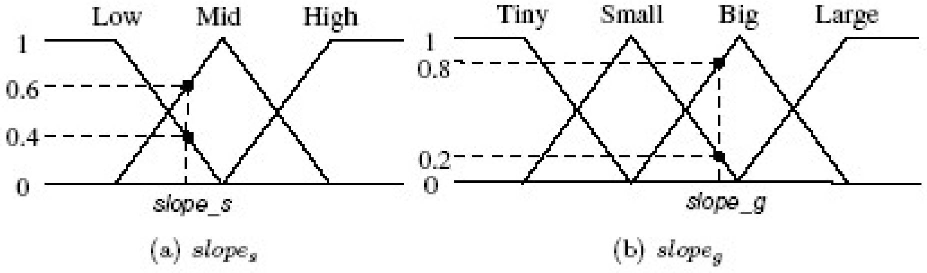


Figure 8: Inputs.

The next example will show how the FL method works. Suppose we have inputs *slope*$_s$ and *slope*$_g$ as shown in Figure 8(a) and Figure 8(b).

The fuzzifier gives

$$\mu_{Low}(slope_s) = 0.4$$
$$\mu_{Mid}(slope_s) = 0.6$$
$$\mu_{Small}(slope_g) = 0.2$$
$$\mu_{Big}(slope_g) = 0.8.$$

With outputs from the fuzzifier, the inferencer defines 4 rules which are 'fired' or activated because of input combinations. Condition '*Low* **AND** *Small*' will turn on rule number 2. '*Low* **AND** *Big*' will turn on rule number 3. Other combination pairs, '*Mid* - *Small*' and '*Mid* - *Big*' will fire rules number 6 and 7 respectively. Following is a summary of 'fired' rules and the corresponding value of $F_d(\mu_A \cap \mu_B) = \min(\mu A, \mu B)$.

Rule 2. If *Low* AND *Small* then F_d2

$$\mu_{Low}(slope_s) \cap \mu_{Small}(slope_g) = \min[\mu_{Low}(slope_s), \mu_{Small}(slope_g)]$$
$$= \min[0.4, 0.2] \rightarrow 0.2\ F_d2$$

Rule 3. If *Low* **AND** *Big then* F_d3

$$\mu_{Low}(slope_s) \cap \mu_{Big}(slope_g) = \min[\mu_{Low}(slope_s), \mu Big(slope_g)]$$
$$= \min[0.4, 0.8] \rightarrow 0.4\ F_d3$$

Rule 6. If *Mid* **AND** *Small* then F_d6

$$\mu_{Mid}(slope_s) \cap \mu_{Small}(slope_g) = \min[\mu_{Mid}(slope_s), \mu_{Small}(slope_g)]$$
$$= \min[0.6, 0.2] \rightarrow 0.2\ F_d6$$

Rule 7. If *Mid* **AND** *Big* then F_d7

$$\mu_{Mid}(slope_s) \cap \mu_{Big}(slope_g) = \min[\mu_{Mid}(slope_s), \mu_{Big}(slope_g)]$$
$$= \min[0.6, 0.8] \rightarrow 0.6\ F_d7.$$

The final step is the defuzzifier, which maps the fuzzy set from inputs to the output. There are many approaches to defuzzification, as mention by Patyra (1996), i.e.

- Centre of area (COA)
- Centre of gravity (COG)
- Height defuzzification (HD)
- Centre of largest area (COLA)
- Mean of maxima (MOM).

Two methods which are common are centre of gravity (COG) method and centre of average (COA) method. The COG method calculates the centroid of the area of all membership outputs, and in most cases the centre of gravity is the same as the centre of area. Therefore these names often refer to the same method. More detailed

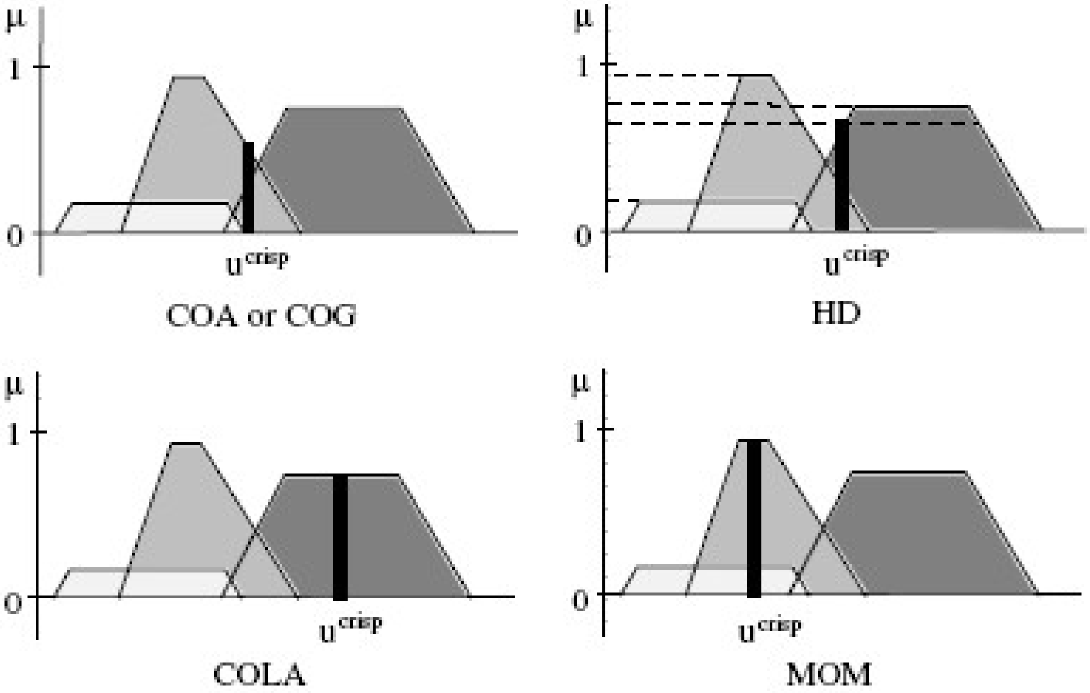


Figure 9: Methods of defuzzification.

definitions of the defuzzification method can be found in Patyra (1996). The methods are illustrated in Figure 9.

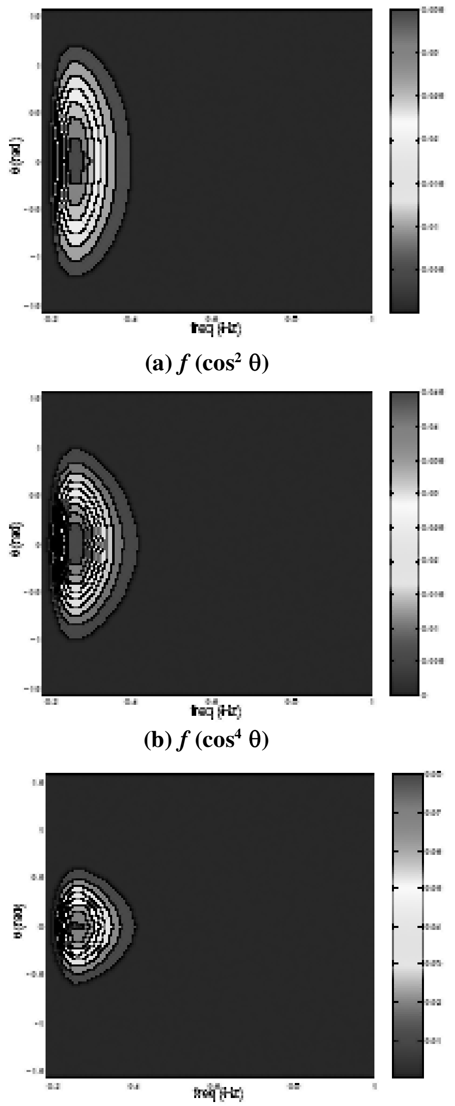
We denote the final output as u^{crisp} and we compute the output with the COA method as defined by the following relation (see Passino, 1998),

$$u^{crisp} = \frac{\sum_i \mu_{premise_{(i)}} f_i}{\sum_i \mu_{premise_{(i)}}}. \tag{10}$$

Finally, we obtain defuzzification results from the inferencer, which yields

$$u^{crisp} = \frac{0.2F_d 2 + 0.4F_d 3 + 0.2F_d 6 + 0.6F_d 7}{0.2 + 0.4 + 0.2 + 0.6}.$$

In this case, $\mu_{premise_{(i)}}$ is $\mu_{premise_{(i)}}$ $(slope_s)$ and $\mu_{premise_{(i)}}$ $(slope_g)$. The final output, u^{crisp}, is the average of all F_d'*s* from four relevant results that follow from the inputs.



(a) ***f*** **(cos^2 θ)**

(b) ***f*** **(cos^4 θ)**

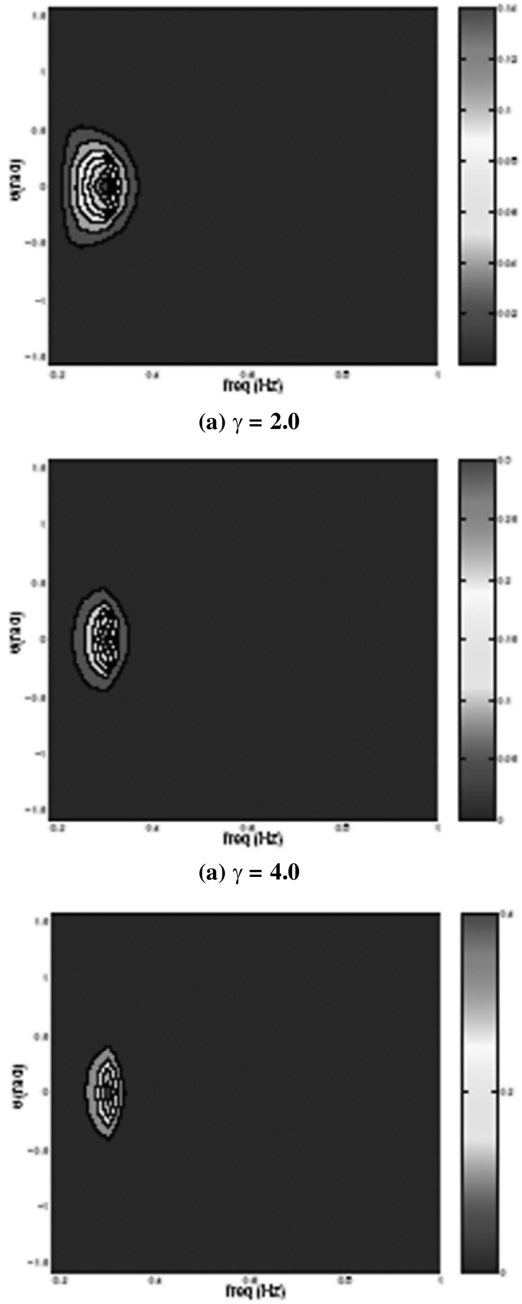
(c) ***f*** **(cos^{12} θ)**

Figure 10: Spectrum energy for different spreading factors at γ = 1.0.

4. RESULTS

Figures 10*a*-10*c* show the reference spectra for spreading exponents p = 1, 2 and 6, and peakedness γ = 1. Similar results can be plotted for peakedness γ = 3, 5, and 7 and are not shown here. With these spectra, FL rules are determined as shown in Table 4. These cases assume a JONSWAP spectrum with parameters (unless otherwise stated) $\alpha = 0.01$, peak spectral width parameters $\sigma_a = 0.07$ and $\sigma_b = 0.09$, and $f_p = 0.3$ Hz.

Tested spectra are JONSWAP spectra with the same parameters as determined by the experiments but having different peakedness and spreading factors. The spectra use γ = 2, 4, and 6 with the Hasselmann-Mitsuyasu spreading factor defined by Hasselmann and Hasselmann (1981). These spectra are depicted on Figure 11 and

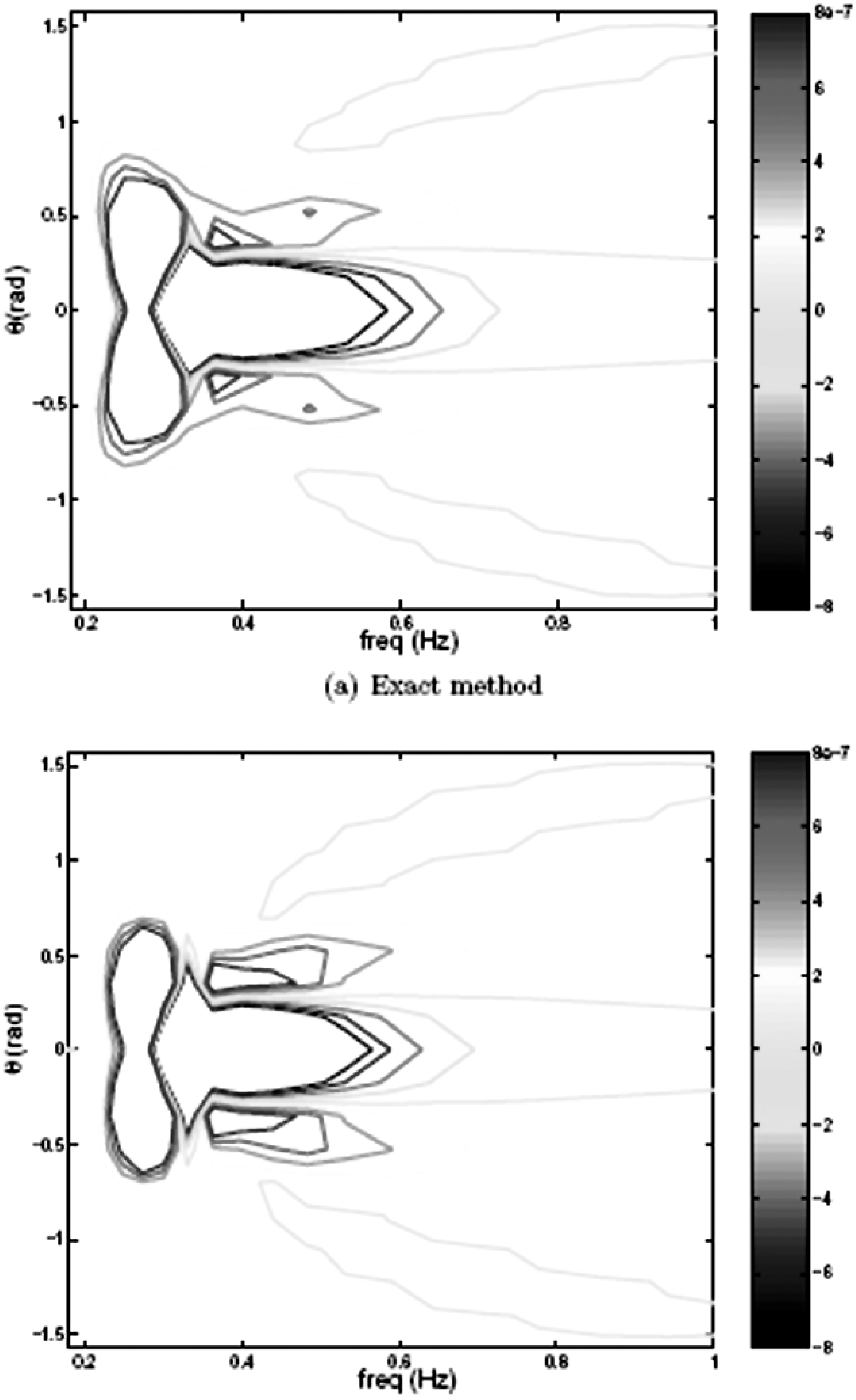


(a) γ = 2.0

(a) γ = 4.0

(a) γ = 6.0

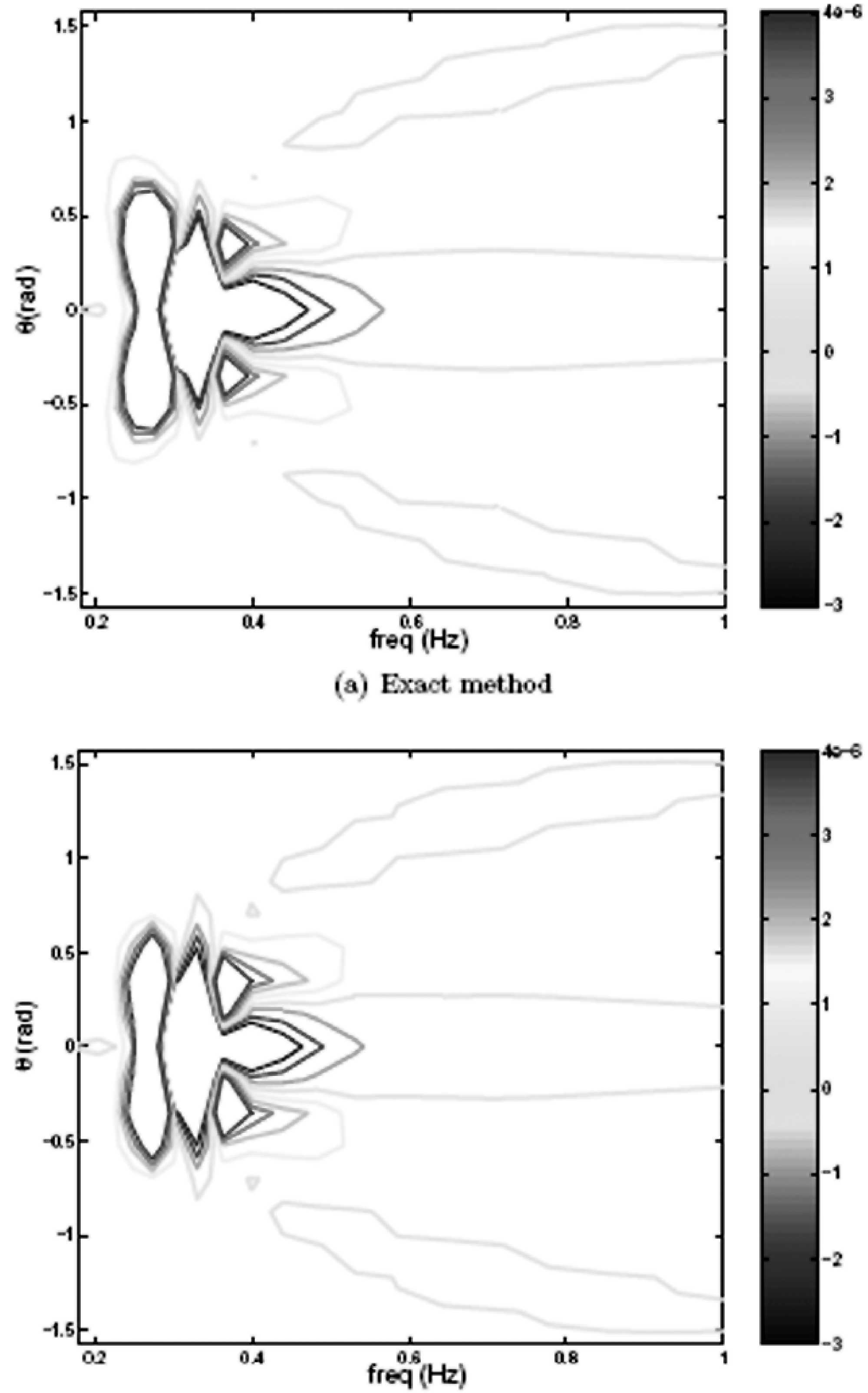
Figure 11: Spectrum energy for different γ = 2, 4, 6 with Hasselmann-Mitsuyasu spreading factor (Hasselmann and Hasselmann, 1981).



(a) Exact method

(b) DTA-FL method

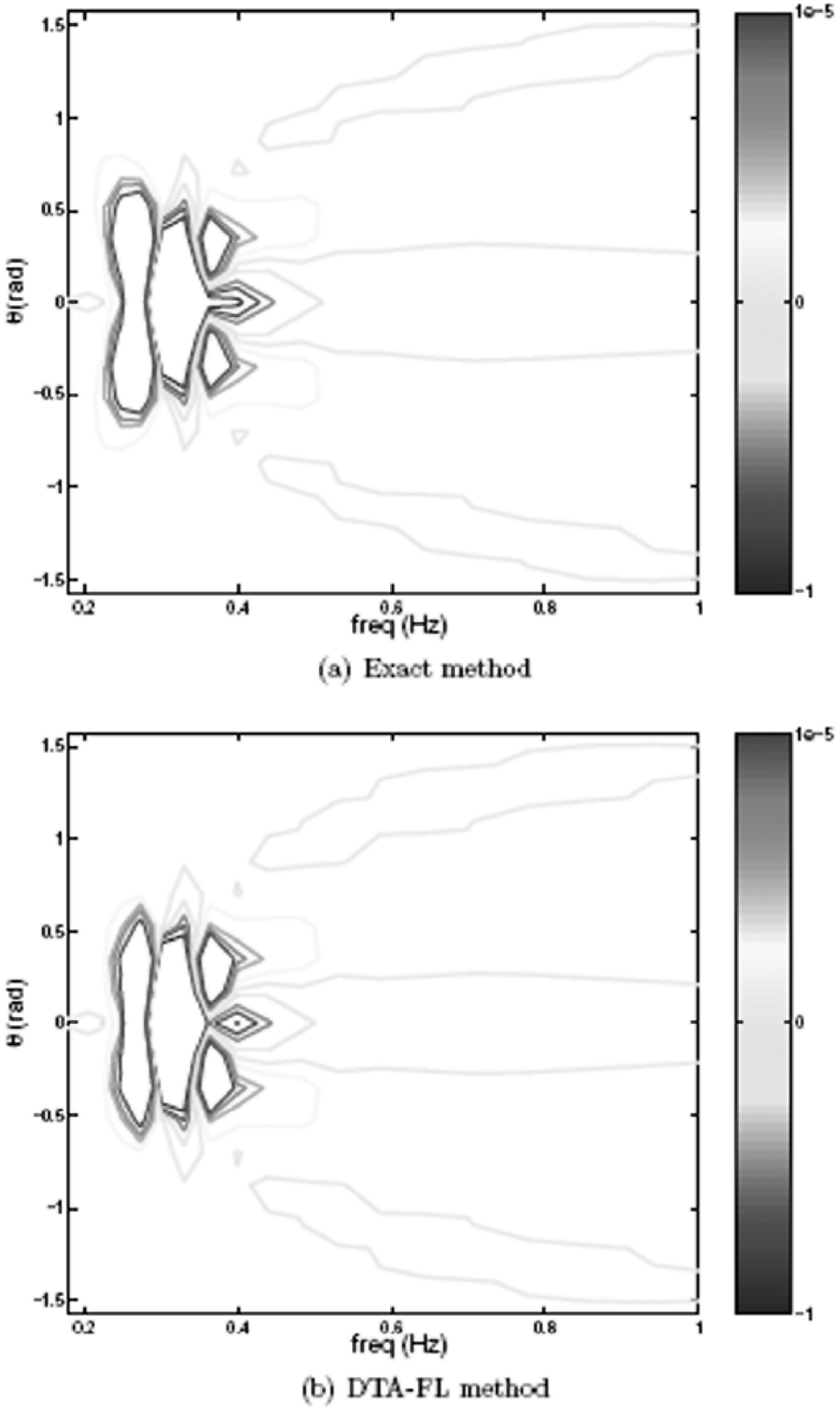
Figure 12: Two-dimensional S_{nl}, with γ = 2.0.



(a) Exact method

(b) DTA-FL method

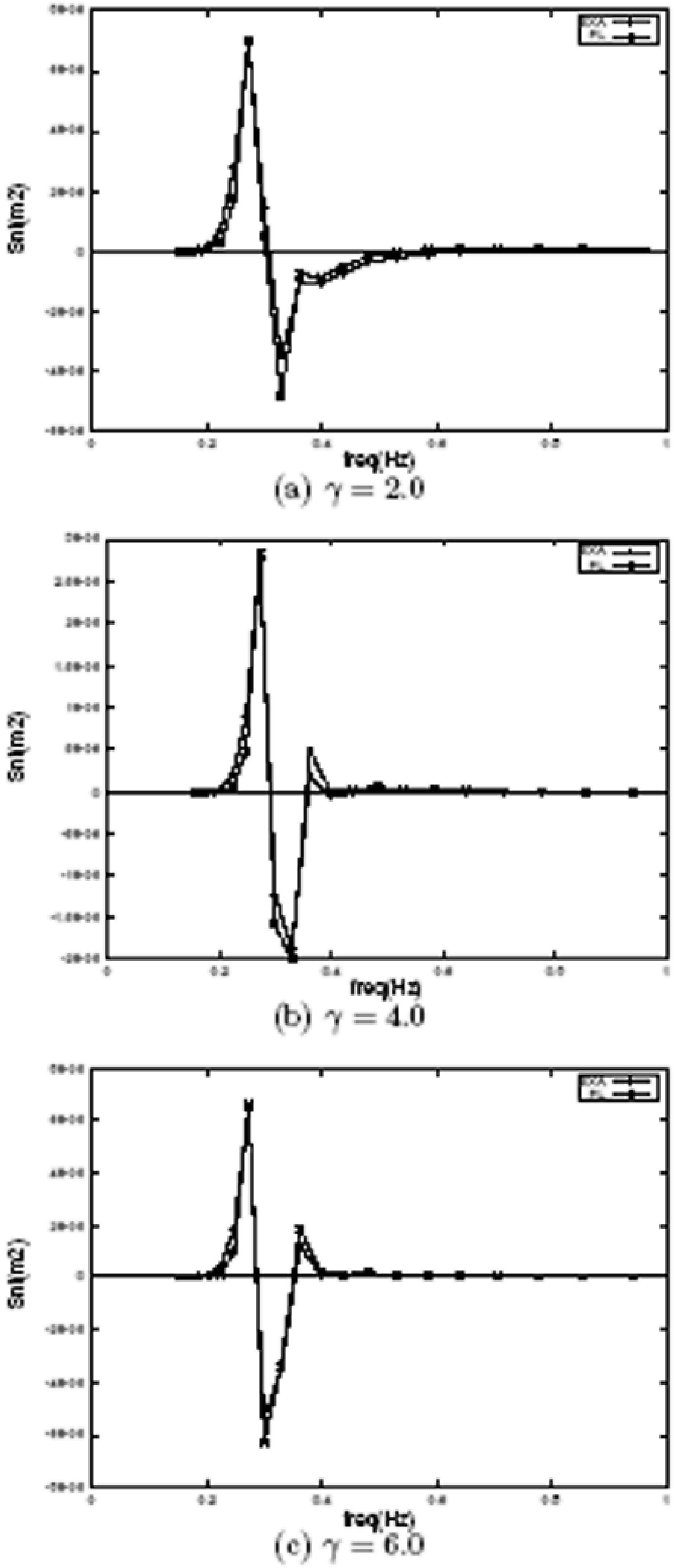
Figure 13: Two-dimensional S_{nl}, with γ = 4.0.



(a) Exact method

(b) DTA-FL method

Figure 14: Two-dimensional S_{nl}, with γ = 6.0.



(a) $\gamma = 2.0$

(b) $\gamma = 4.0$

(c) $\gamma = 6.0$

Figure 15: One-dimensional S_{nl} for different γ = 2, 4, 6.

computed S_{nl} are shown in Figures 12-15. These figures show the comparisons between the full Boltzmann method and the DTA-FL method, in both 1-dimension and 2-dimensions.

5 CONCLUSION

Based on several numerical cases and experiments, as presented in this paper, fuzzy logic can determine the required dominant factor for given spectra. As shown in Figures 12-15, DTA-FL gives good agreement with results from the full Boltzmann formulation. Based on simple rules, the fuzzy logic (FL) method requires straightforward instructions for computer programming. By finding an accurate F_d, the FL method makes DTA a possible candidate to be installed in an operational wave forecast model.

ACKNOWLEDGEMENTS

The authors are grateful to the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Climate and Atmospherics Studies, the Panel of Energy Research and Development, and SURA SCOOP for financial support leading to this paper.

REFERENCES

[1] Hasselmann, K., 1962: *On the Non-linear Energy Transfer in Gravity-wave Spectrum, Part 1. General Theory,* J. Fluid Mech., 12, 481-500.

[2] Hasselmann, S., and Hasselmann, K., 1981: *A Symmetrical Method of Computing the Nonlinear Transfer in a Gravity Wave Spectrum,* Max Planck Inst. Tech Report, Hamburg.

[3] Hasselmann, S., Hasselmann, K., Allender, J.H., and Barnett, T.P., 1985: *Computations and Parameterizations of the Nonlinear Energy Transfer in a Gravity-Wave Spectrum. Part II: Parameterizations of the Nonlinear Energy Transfer for Application in Wave Models,* J. Physical Oceanography, 13, 1378-1391.

[4] Kosko, B., 1993: *Fuzzy Thinking - The New Science of Fuzzy Logic,* Hyperion, New York.

[5] Lin, R.Q., and Perrie, W., 1999: *Wave-wave Interactions in Finite Depth Water,* J. Geophys. Res. Vol. 104, no. c5, pages 11,193-11, 213.

[6] Passino, K.M., and Yurkovich, S., 1998: *Fuzzy Control,* Addison-Wesley Longman, Inc., USA.

[7] Patyra, M.J., and Mlynek, D.M., 1996: *Fuzzy Logic, Implementation and Applications,* John Wiley & Sons, Ltd., England.

[8] Susilo, A. and Perrie, W., 2006: *A Dominant Transfer Approximation for The Nonlinear Wave-wave Interactions in Wind Wave Spectra,* submitted to IJAMES.

[9] Tracy, B.A., and Resio, D.T., 1982: *Theory and Calculation of the Non-linear Energy Transfer between Sea Waves in Deep Water,* WIS Report 11, 47 pages.

[10] Van Vledder, G., 2000: *Improved Method for Obtaining the Integration Space for the Computation of Nonlinear Quadruplet Wave-wave Interactions,* Alkyon Hydraulic Consultancy & Research, Emmeloord, The Netherland, 14 pages.

[11] Webb, D.J., 1978: *Non-Linear Transfers between Sea Waves,* Deep Sea Research, vol. 25, pp 279-298.

[12] Yen, J., and Langari, R., 1999: *Fuzzy Logic: Intelligence, Control, and Information,* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Adhi Susilo and Matiur Rahman
Dept. of Engineering Mathematics, Dalhousie University
Halifax, Canada

Will Perrie
Fisheries & Oceans Canada, Bedford Institute of Oceanography
Dartmouth, Canada