2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

JuzzyOnline: An Online Toolkit for the Design, Implementation, Execution and Sharing of Type-1 and Type-2 Fuzzy Logic Systems

Christian Wagner *Senior Member, IEEE*, Mathieu Pierfitt and Josie McCulloch *Student Member, IEEE*

***Abstract*—In this paper we present an online fuzzy logic toolkit for the design, implementation, execution and sharing of type-1 (T1), interval type-2 (T2) and (zSlices based) general T2 fuzzy logic system (FLSs). The motivation to develop the toolkit stems from the desire to provide a free-to-use fuzzy logic toolkit available which is platform-independent, easily accessible and which does not require any background knowledge of programming. This toolkit aims to help expand the accessibility of FLSs, in particular of T2 FLSs, to both research and industrial applications outside of the fuzzy logic community and computer science more generally. We review the features currently available through the JuzzyOnline toolkit (including a complete, previously unseen visualisation of the inference steps for zSlices based general T2 FLSs) and demonstrate a sample Fuzzy Logic System implementation of the toolkit. Finally, we conclude with some future developments and a call for feedback and contributions to aid in further development.**

I. Introduction

FUZZY logic, first introduced in 1965 [1], has since become a popular field which has been successfully applied to a wide variety of applications from medical to industrial. The majority of research and applications have been based on T1 fuzzy logic, though recent years have shown an increasing interest in T2 fuzzy logic, introduced in 1975 [2], and T2 FLSs [3]. Most T2 applications are research-led, however, and T2 fuzzy logic has not been widely adopted outside of the fuzzy logic community.

The application of new methods can be difficult where there is a lack of software to implement them. Particularly to those without a programming background, the access to new methodologies and/or tools is often made very challenging, highlighting the need for tools that provide straightforward access to new methods. Likewise, the reproducibility of research is aided by the availability of software with which enables the recreation of research results to those beyond the research community. Though, currently there are a number of (T2) fuzzy logic toolkits available, they require users to both have a prior knowledge of FLSs and a good understanding of programming. To address these points, we have developed a free, openly accessible, cloud-based online toolkit, which can be used to design, execute and share both T1 and T2 FLSs.

The developed platform provides online access (through a web-based GUI) to many of the features of the Java based fuzzy logic toolkit we developed previously [4]. The proposed toolkit does not require any prior programming knowledge and both enables the rapid design and development of T1, interval and general T2 FLSs from the convenient environment of a web-browser, as well as the straightforward sharing and execution of the resulting systems. By developing a toolkit which is web-based, its access is available to anyone via a web browser. The application is platform independent and we have eliminated the need to install any additional software or code to be able to use the application. This aids in making the toolkit more accessible than previous toolkits.

We expect that the JuzzyOnline toolkit will provide more direct and easy access to both standard and more advanced (i.e. T2) fuzzy systems to both amateurs (including students) and professionals from a variety of disciplines and are already employing the system in our own teaching. This paper is intended as an introduction to the JuzzyOnline toolkit, as well as an invitation to provide feedback and recommendations.

In Section II we provide background on fuzzy logic toolkits currently available, followed by an overview of the JuzzyOnline toolkit in Section III. Finally, some conclusions are presented in Section IV.

Christian Wagner and Josie McCulloch are with the School of Computer Science, University of Nottingham, Nottingham, United Kingdom (email: {christian.wagner, psxjm5} @nottingham.ac.uk)
Mathieu Pierfitt is within Institut National Polytechnique de Toulouse, Toulouse, France (email: mpierfitte@me.com)
This work was partially funded by the EPSRCs Towards Data-Driven Environmental Policy Design grant, EP/K012479/1 and the RCUKs Horizon Digital Economy Research Hub grant, EP/G065802/1.

II. Background

Toolkits to aid the implementation of T1 FLSs have proven successful in the widespread adoption of fuzzy logic. The most well known T1 fuzzy logic toolkit is provided with MATLAB®, allowing users to design T1 FLSs using command-line functions or a GUI [5].

Toolkits for T2 FLSs have also been developed, however their adoption has been limited. Initially, researchers from within the fuzzy logic community released free-to-use source code. For example, J. Mendel released open-source MATLAB® software to aid in the development of interval T2 FLSs. However, using and modifying the source-code requires in-depth knowledge of the MATLAB® programming language and a significant amount of insight into fuzzy logic algorithms, making it challenging to use, in particular for those not within Computer Science.

Castro et al. [6] developed a MATLAB® based toolkit which provides a GUI for the development of interval T2 FLSs. While this greatly facilitates the access to T2 fuzzy logic algorithms, the fact that MATLAB® software is subject to license fees, while also mainly being used in engineering and computer science, still imposes a considerable hurdle for their wider adoption.

Wagner et al. [7] developed a fuzzy logic toolkit using the R programming language. R is a freely available, platform-free language which is popular within research communities from Psychology to Computer Science. The toolkit uses a command-line interface, similar to that of the MATLAB® Fuzzy Logic Toolbox. Although the toolkit does not provide a GUI, it does contain functions with which to plot visualisations of fuzzy sets (FSs) and control surfaces.

Other toolkits available (with a focus on T1 fuzzy toolkits) include the open-source native applications KBCT [8] and GUAJE (an upgraded version of KBCT) [9], which can be used to extract features from data and generate fuzzy knowledge bases with a linguistic rule base. Also available is FisPro [10], an open-source toolkit created for generating fuzzy inference systems and rules from data. FisPro also provides visualisations of the system with a graphical user interface. Xfuzzy [11], based on the specification language XFL, is another available toolkit. Xfuzzy provides tools to aid in the design process of fuzzy logic-based inference system and the evaluation of different fuzzy operators. The toolkit also provides a graphical user interface to aid in the development of FLSs.

Recently, the Juzzy toolkit [4] was developed for T1, and interval and general T2 FLSs. The toolkit was developed using the freely-available, platform-independent programming language Java which enables the resulting applications to run on a variety of hardware platforms and operating systems. Juzzy is the first toolkit to enable the implementation of (zSlices based [12]) general T2 FLSs and supports the use of multiple processors for the parallel processing of general T2 FSs. The toolkit can dynamically allocate processing to multiple cores, speeding up the process of applications with no additional effort from the developer. Additionally, although no GUI is provided for the Juzzy toolkit, it contains functions for plotting FSs and control surfaces.

JuzzyOnline, the toolkit introduced in this paper, has been developed on top of the Juzzy toolkit to provide an easily accessible toolkit with a GUI. It has been developed to address the remaining existing hurdles to the experimentation with T1/T2 FLSs for experts and non-experts alike, namely: the need for programming skills, the need for installing specific software, and the difficulty in collaborating and sharing FLSs with colleagues, students, etc. The toolkit is free to use online and does not require any prerequisites to be installed other than a web browser. Its adoption is also not limited by prior knowledge of programming languages, it enables straightforward sharing of FLSs and it does not require an in-depth knowledge of FLSs.

In the following section, we briefly highlight some of the features of the JuzzyOnline toolkit, before using a practical example to demonstrate its functionality.

III. Juzzy Online Features

JuzzyOnline is available to use at http://juzzyonline.wagnerweb.net. It is free of charge for non-commercial use and we only ask authors/developers to reference this paper when using it.

All of the features required to construct a Mamdani FLS are included for T1, interval T2 and (zSlices based [12]) general T2 fuzzy systems. We present an example of a FLS created using the online toolkit, and highlight similarities in the development of different types of FLSs. The examples are solely intended to demonstrate the features of the toolkit, rather than to demonstrate or compare the performance of different FLSs. The website offers a brief tutorial to those unfamiliar with either the website or Mamdani FLSs, and provides example FLSs of T1, interval T2 and general T2 which apply the tipping problem, introduced in Section III-B.

All of the images within this paper have been taken from the toolkit and the ability to export images of FSs and other components of the FLS is integrated (accessible via left-clicking figures) into the system. Additionally, the toolkit offers the ability to save FLSs, which can be accessed and shared amongst others via a URL. Further details are provided in Section III-C.

A. Membership Functions

In this section, we give an overview of membership functions (MFs) supported by JuzzyOnline. The system supports Gaussian, triangular, trapezoidal and Gau-Angle MFs.

1) Type-1 Fuzzy Sets : We first give an overview of T1 FSs that can be Fig. 1 shows a triangular MF copied from JuzzyOnline. The triangle is defined by its left-most, right-most and centre coordinates. Fig. 2 shows how the parameters of the MF are chosen using JuzzyOnline. Fig. 3 shows a trapezoidal FS whose coordinates are determined by the two left and two right coordinates which define the MF. Fig. 4 shows a Gaussian MF for which the mean and standard deviation are chosen. Fig. 5 shows a Gau-Angle MF, for which the left, right and centre coordinates are chosen.

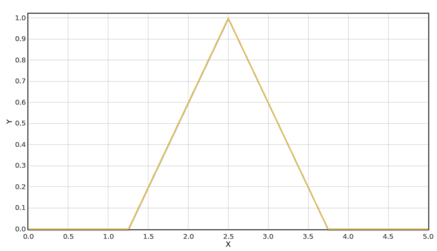


Fig. 1. An example of a T1 FS with a triangular MF

Fig. 2. Selecting the parameters of a triangular MF for a T1 FS.

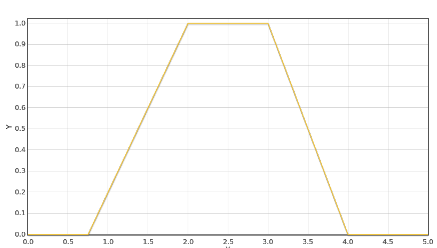


Fig. 3. An example of a T1 FS with a trapezoidal MF

2) Interval Type-2 Fuzzy Sets : Interval T2 FSs offer the same MF types as T1 FSs and the coordinates of the lower and upper MFs are chosen independently as

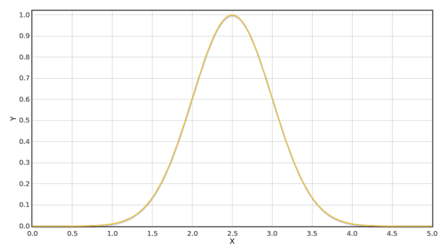


Fig. 4. An example of a T1 FS with a Gaussian MF

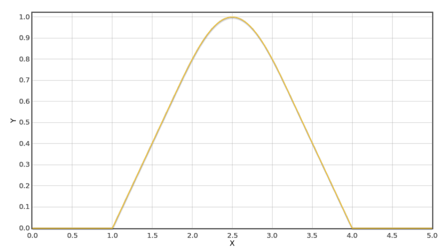


Fig. 5. An example of a T1 FS with a Gau-Angle MF

shown in Fig. 7. Fig. 6 shows a triangular interval T2 FS, Fig. 8 shows a trapezoidal FS, Fig. 9 shows a Gaussian FS and Fig. 10 shows a Gau-Angle FS.

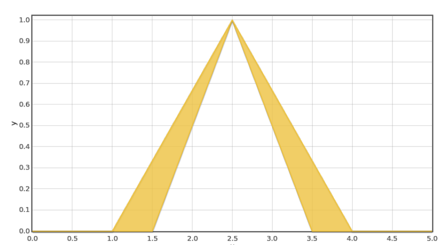


Fig. 6. An example of an interval T2 FS with a triangular MF

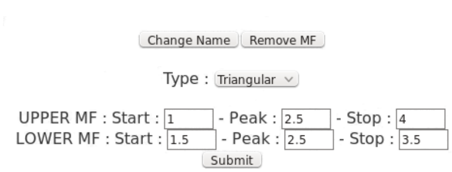


Fig. 7. An example of a selecting the parameters of a triangular MF for a T2 FS.

3) General Type-2 Fuzzy Sets : In JuzzyOnline, each general T2 FS is created using zSlices [12]. zSlices-based general T2 FSs are constructed using a series of zSlices where each zSlice is an interval T2 FS with a secondary membership value z_i for the i^{th} zSlice. This is unlike regular interval T2 FSs for which the secondary membership value is always 1. Each zSlice is distributed evenly throughout the footprint of uncer-

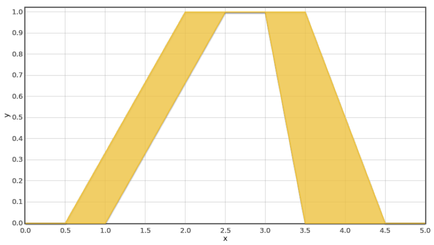


Fig. 8. An example of an interval T2 FS with a trapezoidal MF

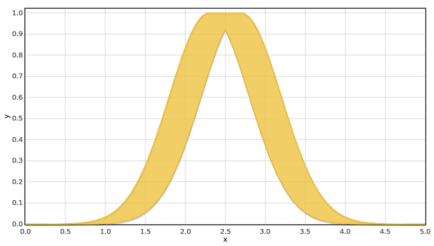


Fig. 9. An example of an interval T2 FS with a Gaussian MF

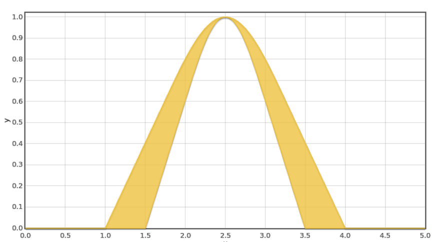


Fig. 10. An example of an interval T2 FS with a Gau-Angle MF

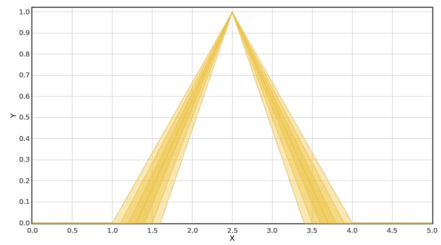


Fig. 11. An example of a general T2 FS with a triangular MF

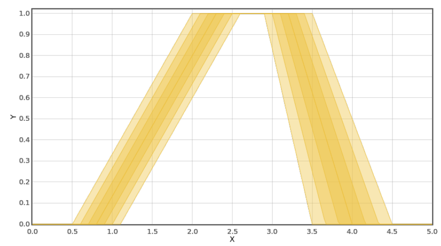


Fig. 12. An example of a general T2 FS with a trapezoidal MF

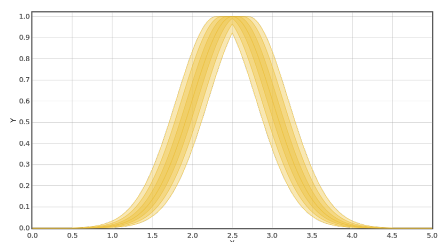


Fig. 13. An example of a general T2 FS with a Gaussian MF

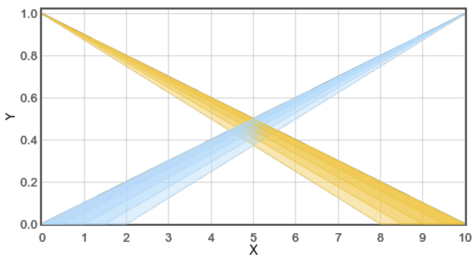
tainty and any number of zSlices may be chosen. For the following examples we have chosen to represent the FSs using four zSlices, making them easily visualisable.

JuzzyOnline provides the following general T2 FS types: triangular, trapezoidal and Gaussian. Membership functions are defined using the same method as interval T2 FSs, as shown in Fig. 7. In Fig. 11 a triangular general T2 FS is presented, Fig. 12 shows a trapezoidal FS and Fig. 13 shows a Gaussian FS. Note the shading across the footprint of uncertainty within the general T2 FSs (e.g., in Fig. 11 to 13) is an indication of the secondary membership value. Darker shading indicates a higher value. Comparing this to the images of interval T2 FSs (e.g., in Fig. 6 to Fig. 10), we see a uniform colour because the secondary membership value of an interval T2 FS is always 1.

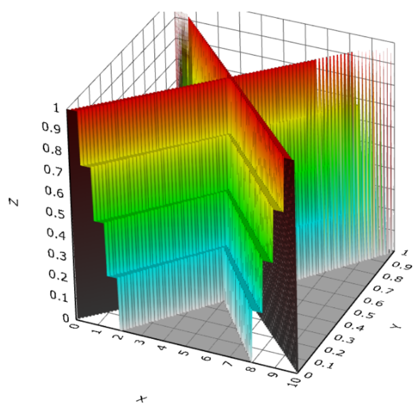
For general T2 FSs, a basic "full 3D" view is also available. For example, Fig. 14 shows two fuzzy general T2 FSs in 2D and 3D as generated by JuzzyOnline. The "full 3D" view is mainly available to provide an intuitive view which highlights the 3D character of the sets - this is expected to be particularly useful to those not yet familiar with general T2 FSs.

B. Application

We give an example of a FLS created using JuzzyOnline, applied to the waiter-tipping problem. The waiter-tipping problem involves determining how much to tip a waiter given the quality of the food and the quality of the service. The FLS will have two inputs (food and service) and one output (tip). The inputs are given on a scale [0,10] and the output is given as a percentage in the range [0, 30]. We shall give an example of a T1 FLS used to implement the waiter-tipping problem. Note that the sequence of steps required for implementing T1, interval T2 and general T2 FLSs are identical, i.e. only the MFs are different, thus we only provide the example case for a T1 FLS. The example application which we show here is available to view and use, as a T1, an interval T2 and a general T2 implementation on the website at http://juzzyonline.wagnerweb.net.



(a)



(b)

Fig. 14. An example of two general T2 FSs shown in 2D (a) and 3D (b).

1) Inputs and Outputs: The FSs we shall use to describe the quality of food, quality of service and amount of tip are shown in Fig. 15, Fig. 16 and Fig. 17, respectively. We have chosen to use 2 to 3 FSs for each variable for simplicity, however for real applications any number of FSs can be created. Note that it is when creating a new variable (e.g., food) that one chooses if it is to be an input or output, and when designing the FSs their names may be assigned which will later be used by the rule base.

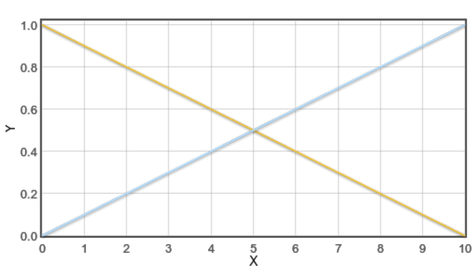


Fig. 15. Fuzzy sets used the describe the quality of food; yellow = bad, blue = great

2) Rule Base Creation: Having set up the input and output FSs we can next create the rule base. Each previously created FS is identified as an input or output in the rule base. Dropdown boxes are used to choose the FSs of each variable within the rule base. Fig. 18 shows an example of a rule base which we shall use in our example application. We can see that dropdown boxes are visible for the fifth rule, which is being edited. The rule editor allows the creation, editing and deletion of rules. Note that only AND connections within rules are currently supported within JuzzyOnline and the t-norm may be selected as minimum or product.

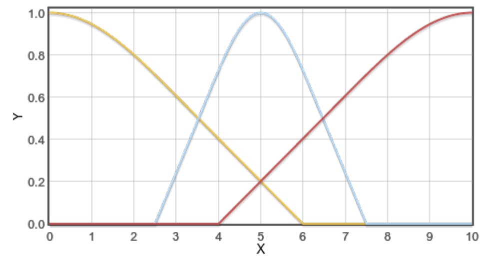


Fig. 16. Fuzzy sets used the describe the quality of service; yellow = unfriendly, blue = OK, red = friendly

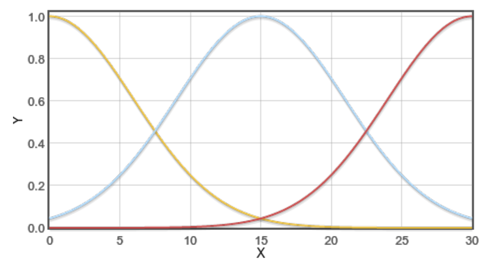


Fig. 17. Fuzzy sets used the describe the amount of tip to give; yellow = low, blue = medium, red = generous

Though the sample application system is MISO, multiple-input multiple-output (MIMO) systems are supported, and when created the output variable(s) for each rule may be chosen. Fig. 19 shows a screenshot of a rule base creating with JuzzyOnline containing two input and two output variables.

After creating a rule base the FLS may be executed. The next section covers the execution of the application.

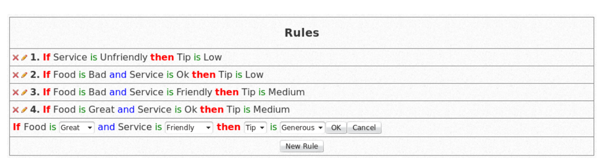


Fig. 18. An example of a rule-base for a MISO system

3) Inference and Defuzzification: In JuzzyOnline, a clear, detailed visualisation of the logical inference is provided at the level of each rule. An example screenshot is shown in Fig. 20 for the waiter tipping

Fig. 19. An example of a rule-base for a MIMO system

problem. The FSs for each rule are listed to aid in the visualisation of the system. The rules as shown in Fig. 18, with the corresponding FSs in Fig. 15, 16 and 17, are listed in Fig. 20.

For execution input, the user can provide a specific input value for each input after which the inference visualisation is updated to show the points at which the FSs are fired. Fig. 20 shows an example of the value 7 for *food* and 8 for *service*. A vertical line is plotted on the graphs to indicate where the value intersects with the FS. The FSs are then coloured to show the section with which each FS is fired; note the blue lines for the inputs and the red lines for the outputs in Fig. 20.

Fig. 21 provides the same view as Fig. 20, however for the case that zSlices based general T2 FSs with 4 zSlices are used. It provides the first visualisations of this kind for zSlices based general T2 FLSs.

The resulting FS is inferred from the outputs in Fig. 20 as shown in Fig. 22. For this example, we have used minimum t-norm in both connective and inference t-norms, as shown in Fig. 20 and Fig. 22, to derive the final FS. However, it is also possible to switch either of these to product t-norms; see Section III-C.

To obtain a single value from the output FS, a choice of defuzzification methods is available. For T1 FSs one may choose between centroid and height defuzzification, and for T2 FSs the choice between centroid and centre-of-sets may be made. Regarding T2 FSs, type reduction is calculated using the Karnik-Mendel algorithm [13]. A visualisation of the type-reduced set and the coordinates of the resulting interval is given by JuzzyOnline. An example of an interval T2 FS and its type-reduced set are shown in Fig. 23. As the general T2 FSs are constructed from zSlices, which are interval T2 FSs, type-reduction is computed by type reducing each of the zSlices before combining the resulting output sets. An example of a general T2 FS and its type-reduced set constructed from the application shown in Fig. 21 is shown in Fig. 24. JuzzyOnline also provides the full numeric coordinates for each type-reduced zSlice as well as the overall set, enabling users to include them in their own applications or publications.

The control surface of the FLS can be generated using

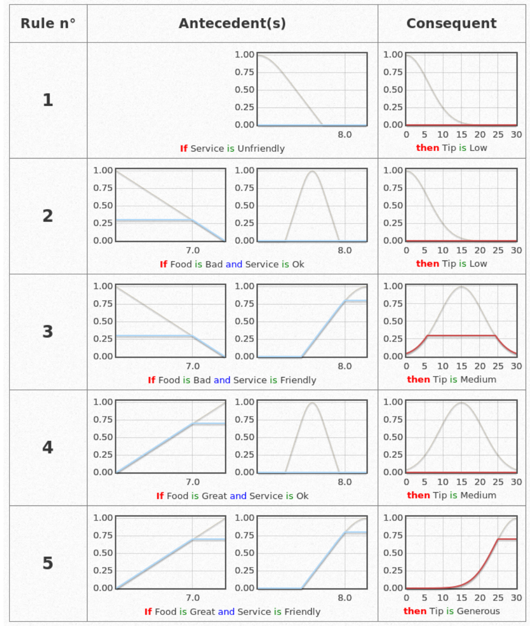


Fig. 20. An example of inference using JuzzyOnline applied to the waiter-tipping problem using T1 FSs.

JuzzyOnline. The control surface for the waiter-tipping problem is shown in Fig. 25. The discretisation steps of the x and y axis may be chosen before rendering, and the choice of which input is assigned the the x and y axes can be chosen by the user. Note that the output for a 2-input, single output system is rendered on the z-axis. In the case of a MIMO system, the user may choose which output is to be rendered on the z-axis.

C. Administration Features

Beyond the FLS specific features, JuzzyOnline provides a number of features that facilitate the design, saving, documentation and sharing of FLSs. We briefly highlight these features, together with advanced configuration features for FLSs below.

1) Saving and Sharing Fuzzy Logic Systems: At any point a FLS may be saved from the "System View" page. Upon saving, a URL is given which encodes the application. For example, the following URL encodes a very basic T1 FLS. The URL contains the name of the system (in this case it is called "Example"), details of input and output FSs and their MFs, and finally the link lists the rules used by the system. http://ritweb.cloudapp.net:8080/JuzzyOnline/gensys?type=1&name=Example&input=in&lb=0.0&ub=10.0&mfnb=1&mf=MF1&fun=triangular&p=0.0_2.0_10.0&output=out&lb=0.0&ub=10.0&mfnb=1&mf=MF1&fun=gaussian&p=10.0_5.0&if=0_0&then=0_0.

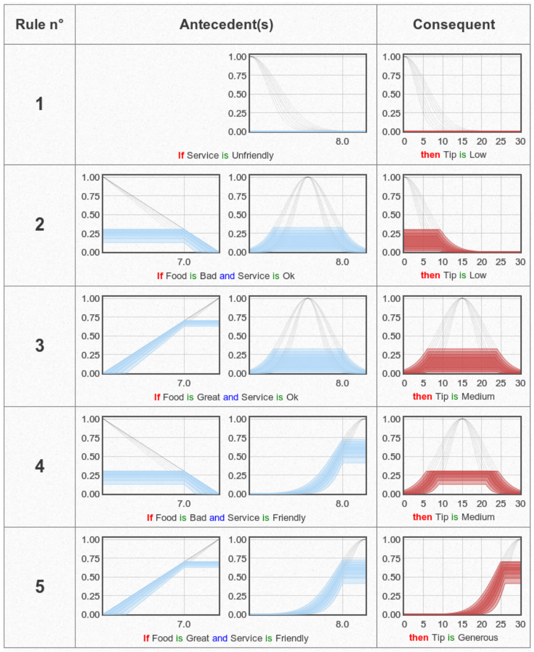


Fig. 21. An example of inference using JuzzyOnline applied to the waiter-tipping problem using general T2 FSs.

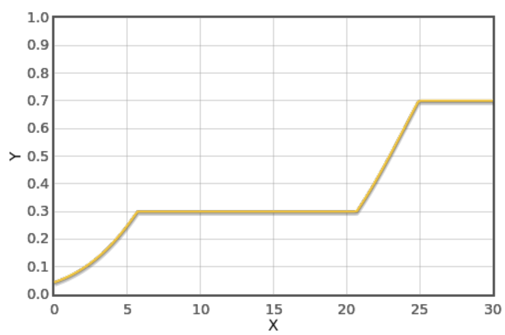
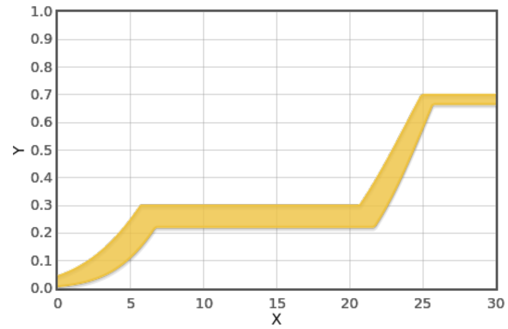


Fig. 22. An example output FS from a T1 FLS on JuzzyOnline

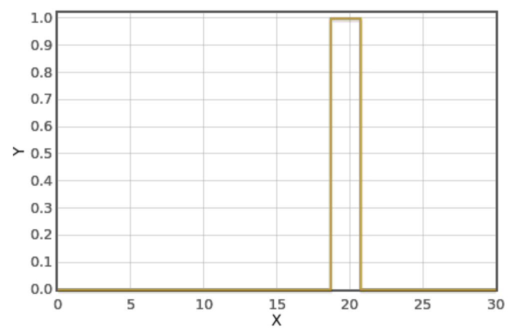
Using the details within the URL, JuzzyOnline constructs the FLS. JuzzyOnline can also automatically generate the URL of your system and provide a link which can be copied and shared via email or any other means.

2) Generating image files for figures within JuzzyOnline: All of the figures in this paper have been generated by JuzzyOnline. Users may right-click any image of FSs on the website to save it. It is also possible to configure the images throughout the website via the preferences menu. For example, the labels of the axes may by configured to display (x, y, z), (x, u, μ), or the text (x, primary membership, secondary membership).

Another configurable feature of the images is the ability to choose where the legend labelling the the

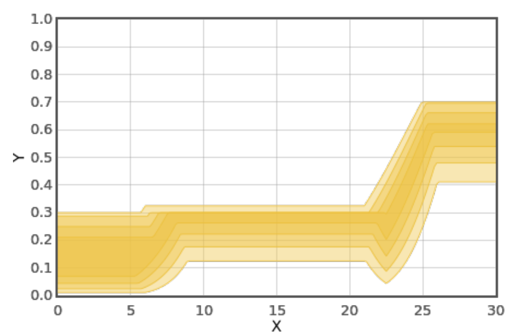


(a)

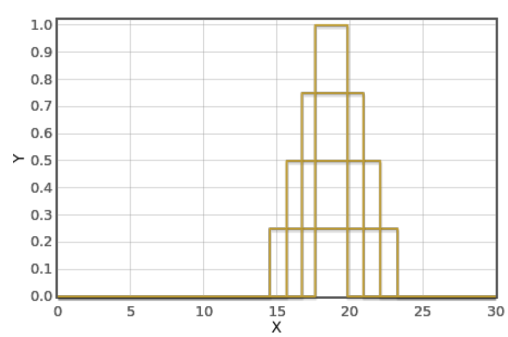


(b)

Fig. 23. (a) An interval T2 output FS created by Juzzy Online. (b) The type-reduced set of (a).



(a)



(b)

Fig. 24. (a) A general T2 output FS created by JuzzyOnline. (b) The type-reduced set of (a).

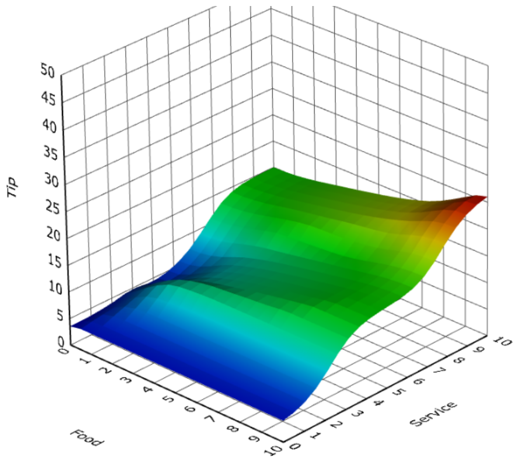


Fig. 25. Control surface of FLS generated by JuzzyOnline.

FSs is displayed. Note that this can only be configured on the website and is not available for saved copies of images. Users may choose to move the legend to any corner of the image, or the legend may be hidden. Finally, the user is also able to switch between displaying coloured or black and white images.

3) Advanced Fuzzy Logic Systems Configuration: With JuzzyOnline, a user is able to configure t-norms as either minimum or product. This can be independently configured for the connective t-norm, which is used by the "AND" clause in rules, and the inference t-norm, which is used to infer the final output FS.

Once the final FS has been derived, multiple methods of defuzzification are available. For T1 FSs a user may choose between centroid and height defuzzification, and for T2 FSs (both interval and general) centroid and centre-of-sets defuzzification are available.

IV. Conclusions

We have presented an overview of the JuzzyOnline toolkit which is available to use at http://juzzyonline.wagnerweb.net. It is free of charge for non-commercial use and we only ask authors/developers to reference this paper when using it. The main purpose of the JuzzyOnline toolkit is to enable a variety of communities to develop FLSs and to share these systems with others. Further, we expect that the detailed inference provided in JuzzyOnline will help to facilitate the understanding of in particular general T2 FLSs and thus make these advanced tools available to a wider community.

We have provided a brief overview of the features available within the toolkit, and have given examples of the design, implementation, execution and visualisations that are possible with the toolkit. We have indicated that the browser-based JuzzyOnline does not require any additional software but is easily accessible online from a variety of devices from computers to smart phones. We have demonstrated the complete end-to-end design and execution of FLSs in JuzzyOnline and have shown examples of first-of-their-kind visualisations of (zSlices based) general T2 inference, type-reduction and defuzzification. Future developments include potential extensions of the platform such as support for non-singleton fuzzy systems and the ability to save fuzzy systems in the XML format to aid in sharing and modifying the models with other toolkits.

The purpose of this paper is to disseminate Juzzy-Online and to provide an "anchor" for those wishing to use it. We hope it serves to expand the availability of T2 fuzzy logic toolkits both inside and outside of the fuzzy logic community, and to encourage feedback and suggestions on the functionality and applicability of the toolkit.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[3] N. Karnik and J. Mendel, "Introduction to type-2 fuzzy logic systems," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, vol. 2, may 1998, pp. 915–920 vol.2.

[4] C. Wagner, "Juzzy-A Java based toolkit for Type-2 Fuzzy Logic," in *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2013 IEEE Symposium on*. IEEE, 2013, pp. 45–52.

[5] "MATLAB® Fuzzy Logic Toolbox™," The MathWorks, Inc., Mar. 2010.

[6] J. R. Castro, O. Castillo, and P. Melin, "An interval type-2 fuzzy logic toolbox for control applications," in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*. IEEE, 2007, pp. 1–6.

[7] C. Wagner, S. Miller, and J. M. Garibaldi, "A fuzzy toolbox for the R programming language," in *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1185–1192.

[8] J. Alonso, S. Guillaume, and L. Magdalena, "KBCT: a knowledge management tool for fuzzy inference systems," 2003.

[9] J. M. Alonso and L. Magdalena, "Generating understandable and accurate fuzzy rule-based systems in a java environment," in *Fuzzy Logic and Applications*. Springer, 2011, pp. 212–219.

[10] S. Guillaume and B. Charnomordic, "Learning interpretable fuzzy inference systems with fispro," *Information Sciences*, vol. 181, no. 20, pp. 4409 – 4427, 2011, special Issue on Interpretable Fuzzy Systems.

[11] D. Lopez, C. Jimenez, I. Baturone, A. Barriga, and S. Sanchez-Solano, "Xfuzzy: a design environment for fuzzy systems," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, vol. 2, May 1998, pp. 1060–1065 vol.2.

[12] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zslices," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 637–660, 2010.

[13] J. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR, 2001.