FCPN – A Fuzzy Continuous Petri Net Modeling and Simulation Tool

Fei Liu, Wujie Sun
School of Software Engineering,
South China University of Technology

January 21, 2019

Contents

1 Introduction **3**
1.1 Petri nets . . . 3
1.2 Continuous Petri nets . . . 3
1.3 Fuzzy logic . . . 4
1.3.1 Mamdani Fuzzy Inference . . . 5
1.3.2 T-S Fuzzy Inference . . . 7
1.4 Fuzzy continuous Petri nets . . . 8
1.5 Our tool . . . 8
1.6 Features - overview . . . 9
1.6.1 Features for modeling . . . 9
1.6.2 Features for simulation . . . 9

2 Modeling **10**
2.1 An overview of the main interface . . . 10
2.2 Draw a net . . . 10
2.2.1 Edit a place . . . 11
2.2.2 Edit a transition . . . 11
2.2.3 Edit an arc . . . 12
2.2.4 Use FIS . . . 14

3 Simulation **21**
3.1 Run simulation . . . 21
3.2 Show simulation results . . . 21
3.3 Export simulation results . . . 23
3.4 Rich functions . . . 23

4 Examples **25**
4.1 1D Diffusion Reaction . . . 25
4.1.1 Introduction . . . 25
4.1.2 Model . . . 25
4.1.3 Simulation result . . . 30
4.2 Enzyme . . . 33
4.2.1 Introduction . . . 33
4.2.2 Model . . . 33
4.2.3 Simulation result . . . 33
4.3 RKIP . . . 36
4.3.1 Introduction . . . 36
4.3.2 Model . . . 36
4.3.3 Simulation result . . . 36
4.4 6-MP metabolism . . . 40
4.4.1 Introduction . . . 40
4.4.2 Model . . . 40
4.4.3 Simulation result . . . 41

1 Introduction

Systems biology aims to study the interactions between the components of a biological system and how these interactions cause the behaviour of the system as a whole. Modelling and simulation plays an essential role in the study of systems biology by employing the data to build mathematical or computational models of a biological system, which help biologists to better understand and predict the system. However, the complexity of biological systems gives rise to a number of modelling challenges, one of which is uncertainty.

The uncertainty further constitutes the structural uncertainty and parametric uncertainty for a biological model.

On the other hand, continuous Petri nets (CPNs) offer a graphical way to represent systems of ordinary differential equations (ODEs). In a CPN model, places take real-valued tokens to represent the concentration of species, while transitions represent continuous changes of concentrations.

By adding fuzzy rules to CPNs, we further obtain fuzzy continuous Petri nets (FCPNs). Therefore an FCPN model can be divided into two parts: the certain ODE part and the uncertain fuzzy logic part, thus achieving the modeling of a biological system where we have sufficient data for some components but insufficient information for others.

1.1 Petri nets

Petri nets provide a formal and graphical representation of systems based on their firm mathematical foundation for the analysis of system properties.

A Petri net, also known as a place/transition (PT) net, is one of several mathematical modeling languages for the description of distributed systems. It is a class of discrete event dynamic system. A Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, represented by bars) and places (i.e. conditions, represented by circles). The directed arcs describe which places are pre- and/or postconditions for which transitions (signified by arrows). [1] No arc is allowed between the two places or transitions. Each arc has its own weight, and the default value is 1.

Let's take the reaction $2H_2 + O_2 \rightarrow 2H_2O$ for example. The Petri nets of the it is shown as Figure 1. Circles named H_2, O_2, and H_2O are places while the square named t_0 is the transition. As we can see, H_2 and O_2 are preconditions for t_0 while H_2O is the postcondition for t_0. The tokens of H_2, O_2, and H_2O are 30, 20, and 0, respectively. So this Petri net can be interpreted as: 2 parts H_2 and 1 part O_2 participate in the reaction, which produces 2 parts H_2O. And the initial parts of H_2, O_2, and H_2O are 30, 20, and 0, respectively.

1.2 Continuous Petri nets

A continuous Petri net is a variant of Petri nets. In this model, the values identified in each place are non-negative real values (no longer required to be non-negative integer values), and the conditions and results of the transitions are also changed accordingly: the occurrences are positive real values.

ODE is the abbreviation for ordinary differential equations. In FCPN, we convert the model to ODEs to calculate the next step value of each input according to its current step value.

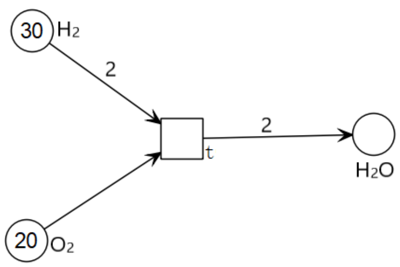


Figure 1: Example of Petri nets.

Let us continue to use the water example. Since the continuous Petri net is a variant of Petri nets, the net seems exactly the same. However, since we can convert the model to ODEs, the calculation become easier.

MassAction(0.1) is used in $t0$. MassAction() is a macro that creates the rate function for a transition out of its preplaces and takes a parameter as argument. For example, assume that t0 has two preplaces p0 and p1, and the parameter k is defined. Now you can specify the rate function of t0 to obey mass/action kinetics as k*p0*p1 or MassAction(k).

Each arc has a formula associated with it. The formula is obtained by multiplying the arc weight and the transition function. If the place is the precondition of one arc, then the formula is multiplied by 1 to get the adding formula, otherwise -1. And the ODE of each place is obtained by adding the formula of arcs connected with it. Therefore, the result is shown as below. For example, the ODE of H_2 is $\frac{dH_2}{dt} = -2 * (0.1 * H_2^2 * O_2)$. After setting the time step, it is easy to get the values of next step by using Runge-Kutta methods.

Table 1: ODEs of the water example

Species	ODE
H_2	$-2 * (0.1 * H_2^2 * O_2)$
O_2	$-1 * (0.1 * H_2^2 * O_2)$
H_2O	$2 * (0.1 * H_2^2 * O_2)$

1.3 Fuzzy logic

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. By contrast, in Boolean logic, the truth values of variables may only be the integer values 0 or 1. It is based on the observation that people make decisions based on imprecise and non-numerical information, fuzzy models or sets are mathematical means of representing vagueness and imprecise information, hence the term fuzzy. These models have the capability of recognising, representing, manipulating, interpreting, and utilising data and information that are vague and lack certainty. [2]

The process of fuzzy logic can be divided into four parts: fuzzification, fuzzy rule base, fuzzy inference engine, and defuzzification.

First, let me introduce the fuzzy inference. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all the pieces that are described in Membership Functions, Logical Operations, and If-Then Rules. [3] In our system, we provide two kinds of fuzzy inference - Mamdani and T-S.

1.3.1 Mamdani Fuzzy Inference

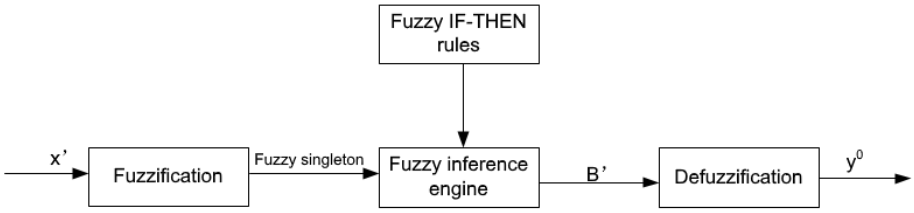


Figure 2: Block diagram of Mamdani.

Mamdani fuzzy inference is the most commonly seen fuzzy methodology and was among the first control systems built using fuzzy set theory. It expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. [4]

(1)Fuzzification

Fuzzification is about to fuzzify all input values into fuzzy membership functions.

For example, in Figure 3, p0 and p1 are inputs while p2 is the output. And Low, Medium, and High are three fuzzy sets. We assume that the minimum value and the maximum value of p0, p1, and p2 are both 0 and 2, respectively. After fuzzification, the fuzzy membership functions are shown as Figure 3 where X indicates the value of the input and μ indicates the membership grade of the fuzzy set. And we have three fuzzy membership functions Low, Medium, and High in each variable.

(2)Fuzzy rule base

The rule base is a collection of all rules. As we can see, the uniform format of the rules is

$$r^r\text{: IF } x_1 \text{ IS } A_1^r \text{ AND ... AND IF } x_n \text{ IS } A_n^r \text{ THEN } y^r \text{ IS } B'^r$$

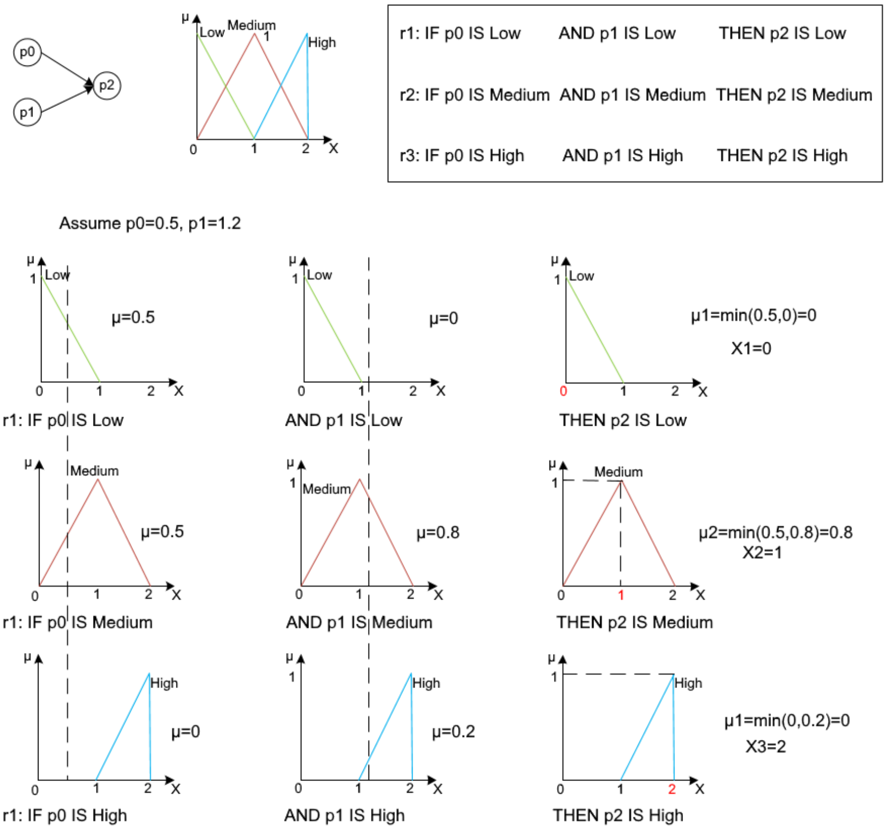
(3)Fuzzy logic operators

In fuzzy logic operators, we execute all applicable rules in the rulebase to compute the fuzzy output functions as is shown in Figure 3.

(4)Defuzzification

Defuzzification is about to de-fuzzify the fuzzy output functions to get “crisp” output values.

Figure 3 shows the process of how to calculate the output value of p2.



1. Calculate the μ of each input in each rule.
2. Find the minimum μn of each rule.
3. Multiply the μn with Xn, where Xn is the value of X when μ=1 of rule n.
4. Divide the sum of μn*Xn by the sum of μn.

p2=(μ1*X1+μ2*X2+μ3*X3)/(μ1+μ2+μ3)
=(0*0+0.8*1+0*2)/(0+0.8+0)
=1

Figure 3: An example of fuzzy logic (Mamdani).

1.3.2 T-S Fuzzy Inference

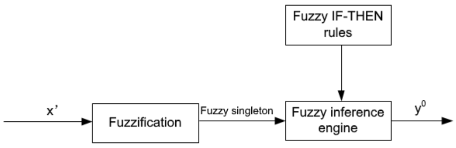


Figure 4: Block diagram of T-S.

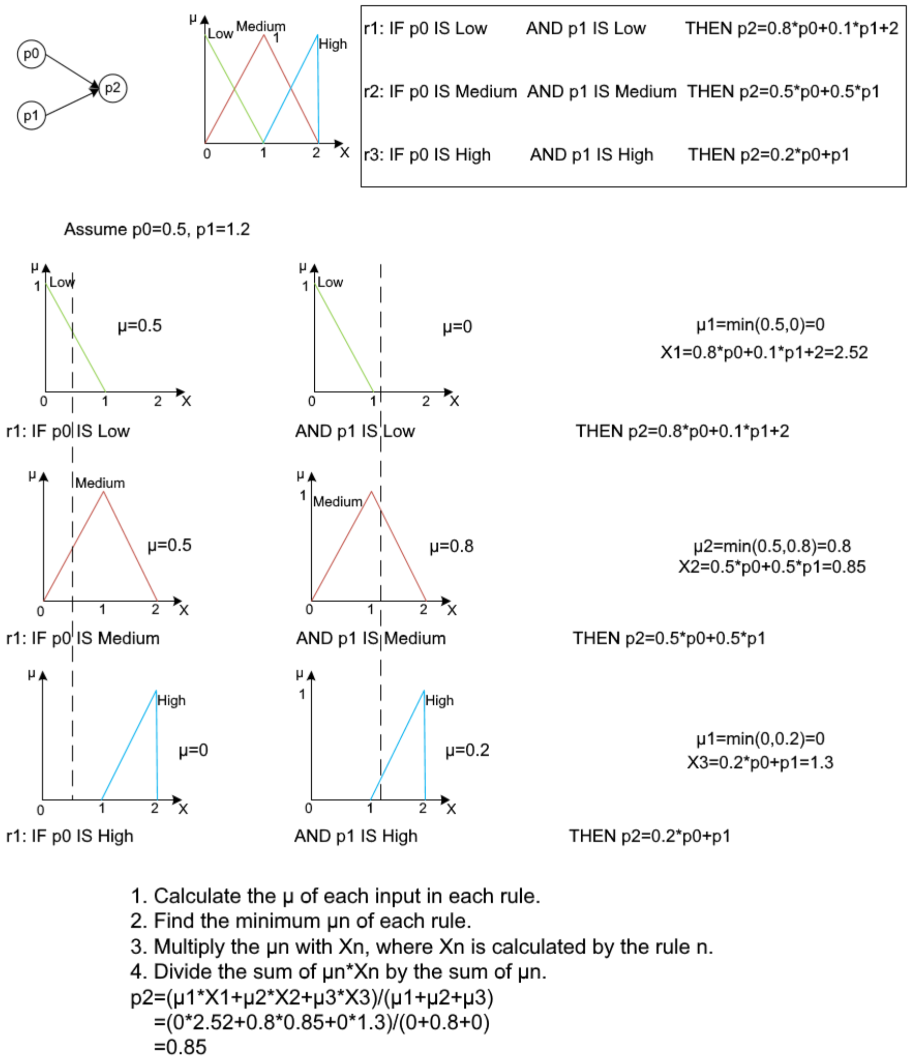


Figure 5: An example of fuzzy logic (T-S).

T-S can be called as Sugeno or Takagi-Sugeno-Kang. This method is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. [5] And the uniform format of the rules is

$$r^r\text{: IF } x_1 \text{ IS } A_1^r \text{ AND ... AND IF } x_n \text{ IS } A_n^r \text{ THEN } y^r = a_1^r x_1 + ... + a_n^r x_n + c^r$$

where a_i^r is ith coefficient in rth rule and c^r is a constant.

Therefore, there is no need to use defuzzification in T-S. The process of calculation is shown as Figure 5.

1.4 Fuzzy continuous Petri nets

Adding fuzzy logic to continuous Petri nets makes the Petri nets more powerful. For example, if we want to get the value of H_2O using fuzzy logic, ODE no longer works for H_2O. The value of H_2O is obtained by fuzzy logic.

Please note that when using fuzzy logic in fuzzy continuous Petri nets, the value we obtained is not the value of the variable, but the amount of change within the step. The change value of H_2O of current time step is obtained by the values of H_2 and O_2 of previous time step.

The result might be a little different from ODE. If the values of H_2 and O_2 are down to 18 and 14, the ODE result of H_2O is 12, while in fuzzy logic, the result of H_2O might be 11.8.

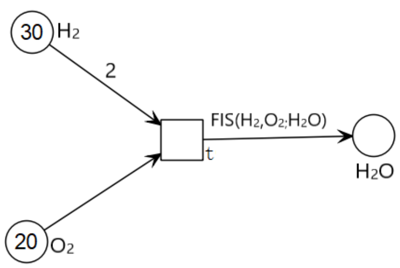


Figure 6: Example of fuzzy continuous Petri nets.

1.5 Our tool

This software provides modeling and simulation functions of fuzzy continuous Petri nets for researchers in the field of systems biology.

It includes three main functions: continuous Petri net modeling, fuzzy (Mamdani & T-S) modeling, and hybrid simulation. The aim of this software is to give an easy-to-use tool for the construction and simulation of an FCPN model.

We offer the Windows, Linux, and macOS(beta) versions of the software. To download the software, please follow the instruction in README.md and download the corresponding version of FCPN.

1.6 Features - overview

Before exploring all features in detail in the following sections, we will give a brief overview for the expected features here.

1.6.1 Features for modeling

- Concise and efficient interface design.
- Drawing of the Petri net graph as usual.
- Flexible user-defined functions.
- Multiple fuzzy logic choices.
- Simple and fast fuzzy logic settings.
- Rich shortcut settings, such as undo, redo, save, print, etc.

1.6.2 Features for simulation

- Highly automated simulation process.
- Diversified export of simulation results.
- Custom simulation result drawing.

2 Modeling

This section will present a general step-by-step procedure for how to construct an FCPN model. A simple example will be used for the illustration of the procedure.

2.1 An overview of the main interface

After opening the application, what we should do first is to draw an FCPN. Figure 7 shows the main interface of the software.

The menu bar is on the top and the tool bar is on the left, which facilitates us to create the net conveniently. The tools from top to bottom include “new”, “open”, “save”, “save as”, “print”, “undo”, “redo”, “delete a node”, “normal cursor”, “create a new place”, “create a new transition”, “create a new arc”, and “start simulation”.

The top window is used to draw the net and the bottom window to output a log. The slider on the bottom right can be used to zoom in or zoom out a net.

Figure 7: Main interface of the software.

2.2 Draw a net

We can draw a net using the following steps taking Figure 8 as an example:

- Left click the place button (the circle on the tool bar).
- Put three places on the palette where you want by clicking the left mouse.
- Left click the transition button(the square on the tool bar).
- Put one transition on the palette where you want by clicking the left mouse.

- Left click the arc button (the arrow on the tool bar).
- Left click a place, press the mouse and hold, move the mouse to a transition, then release the mouse. In this way you can draw an arrow from a place to a transition.
- Repeating these steps, an FCPN can be constructed; see Figure 8.

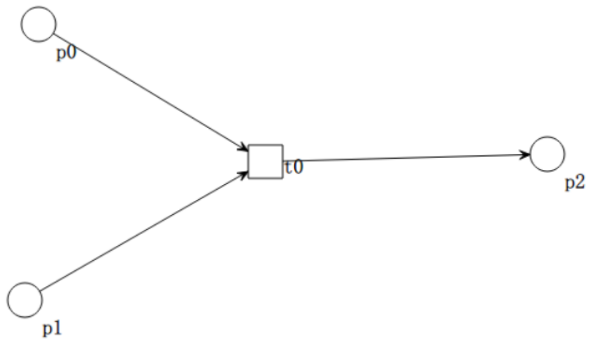


Figure 8: Draw the net.

2.2.1 Edit a place

We can edit a place in the following way:

- Double click the place on the palette.
- A window named Place Attributes (Figure 9) will show.
- Change the Name and Marking.
- After editing, left click OK button.

For example, in Figure 8, we change the names of p0, p1, p2 to H_2, O_2, H_2O, respectively, and change the markings of p0, p1, p2 to 30, 20, 0, respectively. Please note that the name of each place should be unique and a marking should be a non-negative real number.

2.2.2 Edit a transition

We can edit a transition in the following way:

- Double click the transition on the palette.
- A window named Transition Attributes (Figure 10) will show.

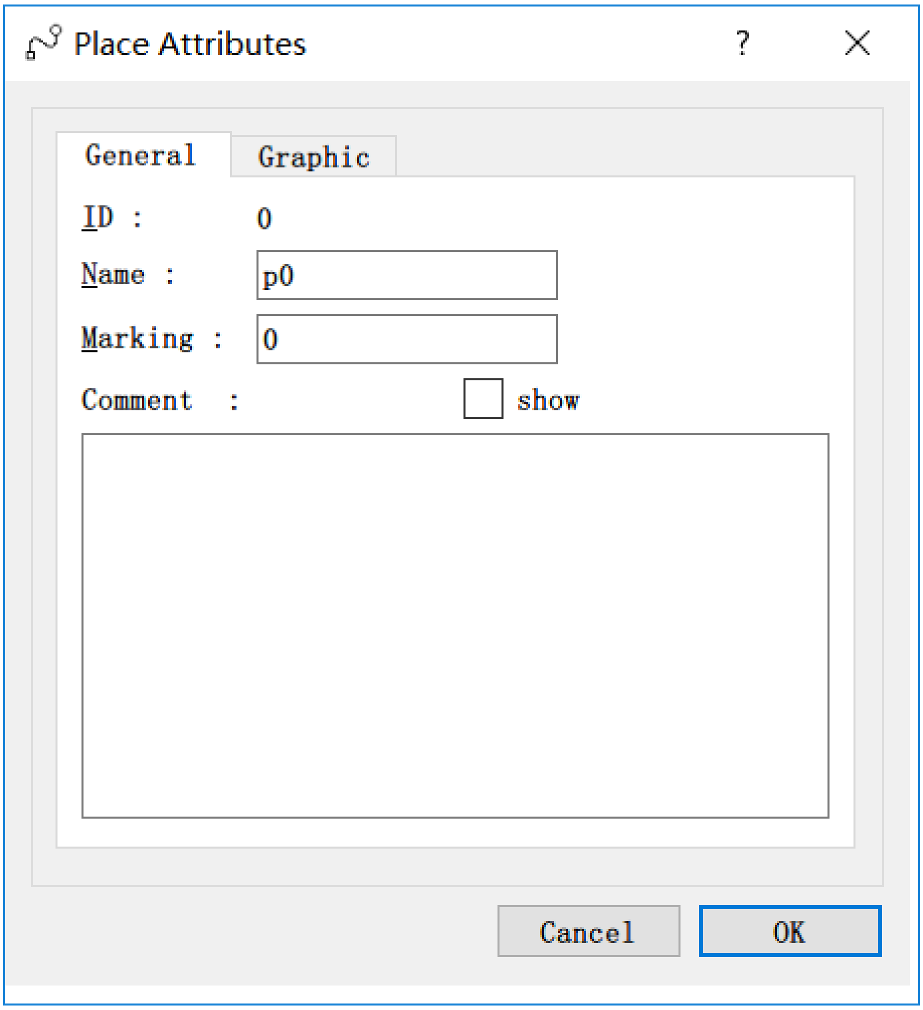


Figure 9: Edit a place.

- Change the Name and Function.
- After editing, left click OK button.

For example, in Figure 8, we change the function of t0 from MassAction(1) to MassAction(0.1). Please note that the name of each transition should be different and the function should be like “1”, “H2”, “MassAction(0)”, “MassAction(O2)”, “MassAction(H2+O2)”. Remember not to change the “MassAction(1)” to “massaction(1)” or something like that. And do not leave the function empty. Otherwise complie error might occur.

2.2.3 Edit an arc

We can edit an arc in the following way:

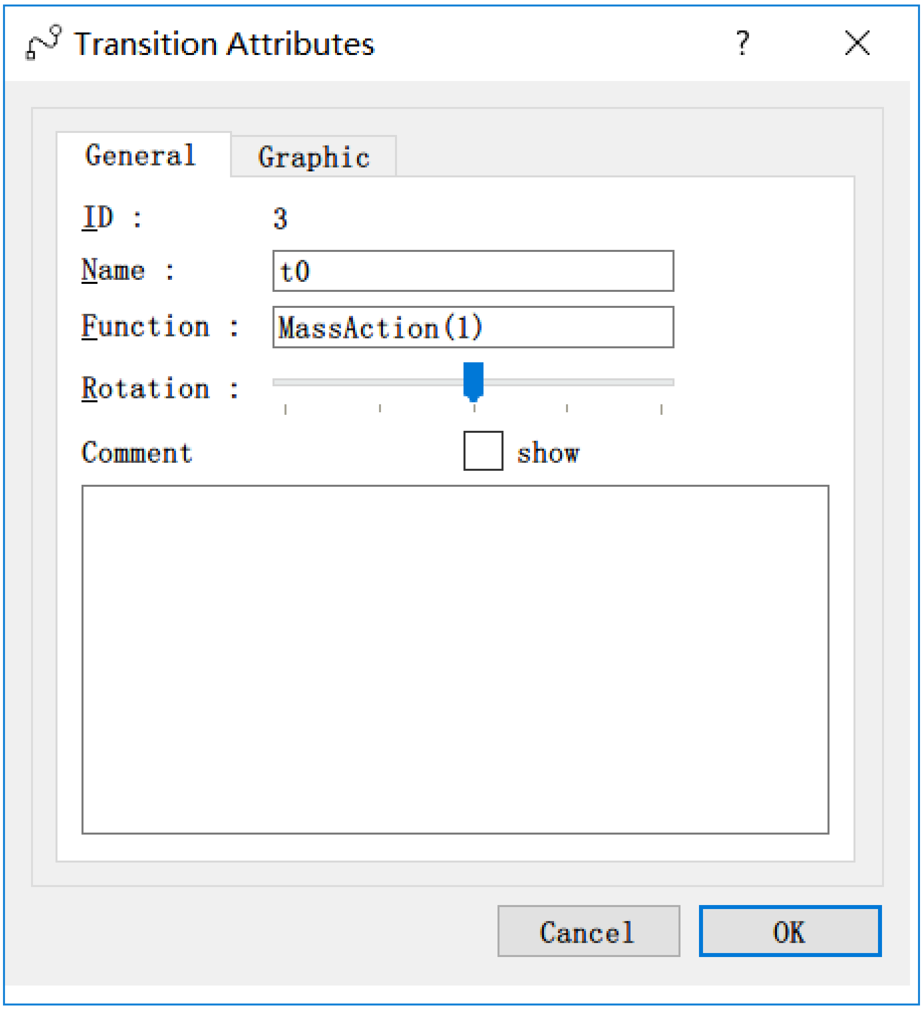


Figure 10: Edit a ransition.

- Double click the arc on the palette.
- A window named Arc Attributes (Figure 11) will show.
- Change the Expression.
- After editing, left click OK button.

For example, in Figure 8, we change the expression of arc from p0 (H2) to t0 to 2. For the expression of arc from t0 to p2 (H2O), see next step. Please note that the expression should be like “1”, “H2” if you don’t want use fuzzy logic, and do not leave the expression empty. Otherwise complie error might occur.

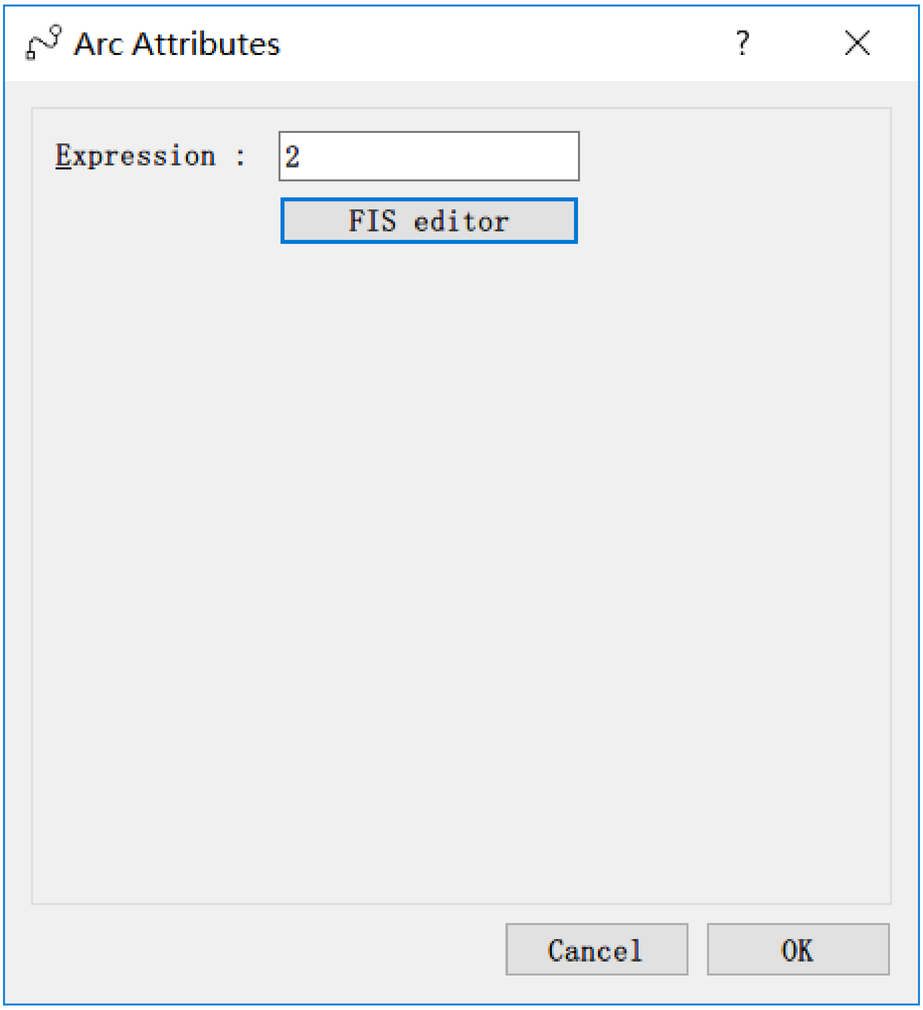


Figure 11: Edit an arc.

2.2.4 Use FIS

FIS is the abbreviation for fuzzy inference system. We provide two kinds of FIS type: Mamdani and T-S. The steps are different between Mamdani and T-S. For example, in Figure 8, here are steps for using Mamdani:

- Double click the arc from t0 to p2 (H2O).
- Click the FIS editor button.
- Choose Mamdani as the FIS Type.
- Enter the time step and select the number of input variables.
- Click the Apply button.
- Choose each variable and its type in the table.

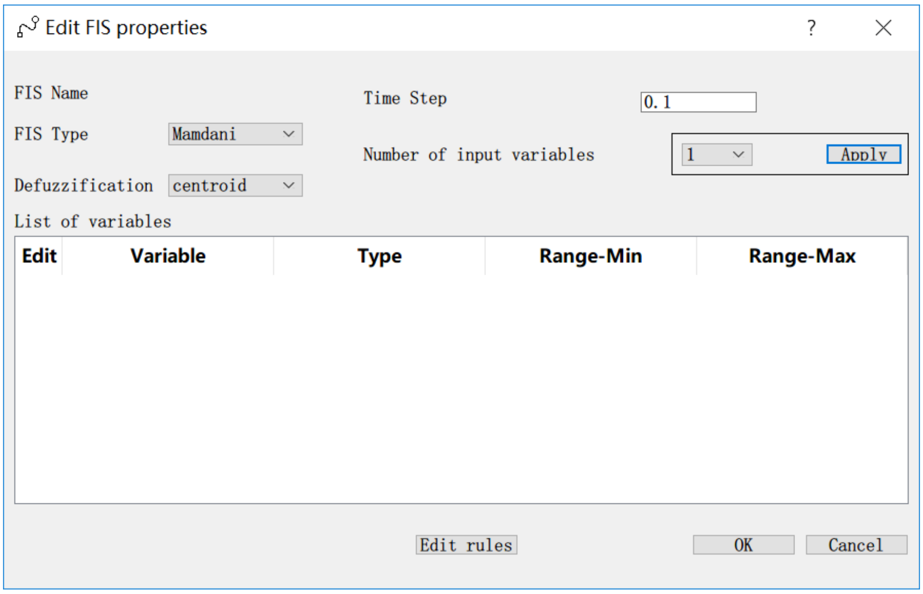


Figure 12: FIS editor.

- Enter the minimum change and maximum change in one step, which are corresponding to the Range-Min and Range-Max of the table.

Make sure there is only one variable whose type is PN_OUTPUT, and it should be the variable connected by the arc you choosed. In this example, that's p2 (H2O). The values are shown in Figure 13.

Now we need to edit each variable's membership functions. Here are the steps:

- Click the Edit button on the first column.
- Choose the number of linguistic variables.
- Click the Apply button and the according label will show.
- For each function, enter the value of left, center and right which means the x values when y is equal to 0, 1, 0, respectively.
- Click the Show Plot button to see whether it is the graph you want.
- Click OK button to save the change and close the window.
- Repeat these steps to edit each variable.

we provide 2, 3, 5, 7 for the user to choose the number of linguistic varibles. The values are shown in Figure 14, 15, 16.

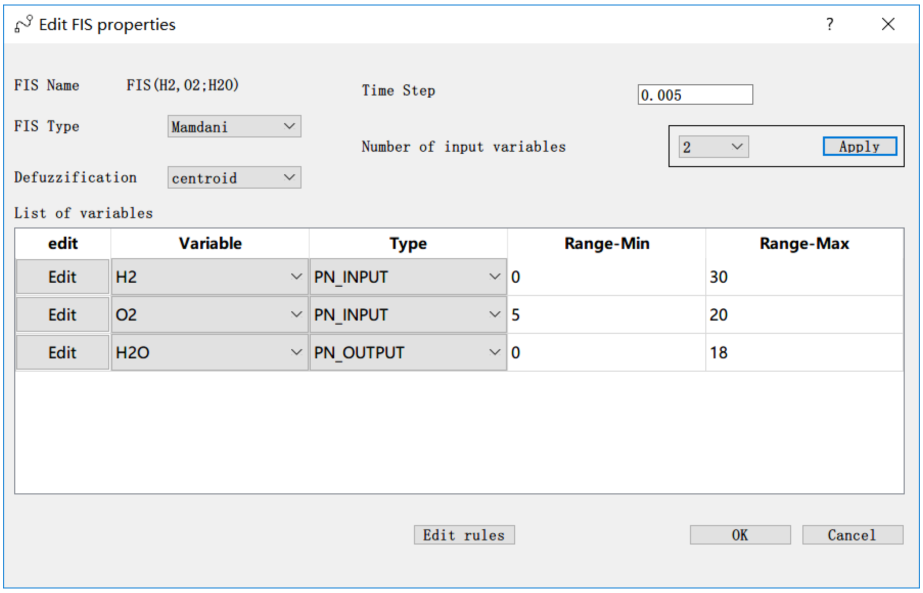


Figure 13: Edit variables.

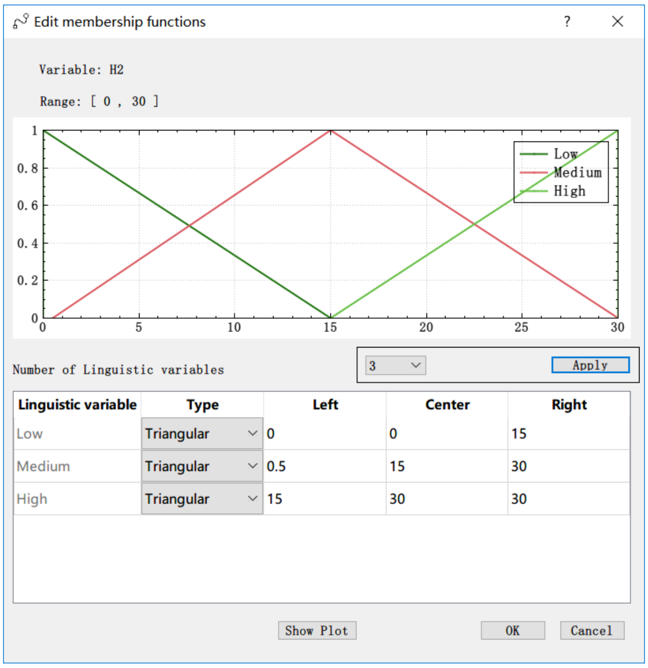


Figure 14: Membership functions of H2.

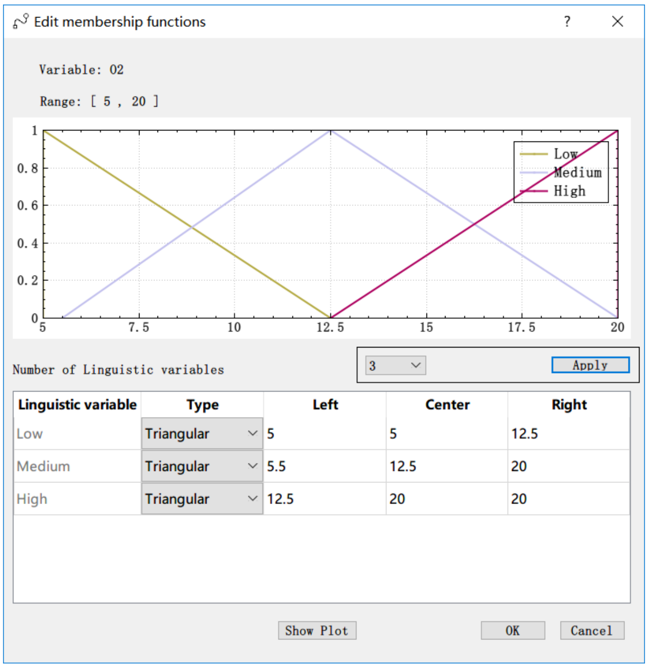


Figure 15: Membership functions of O2.

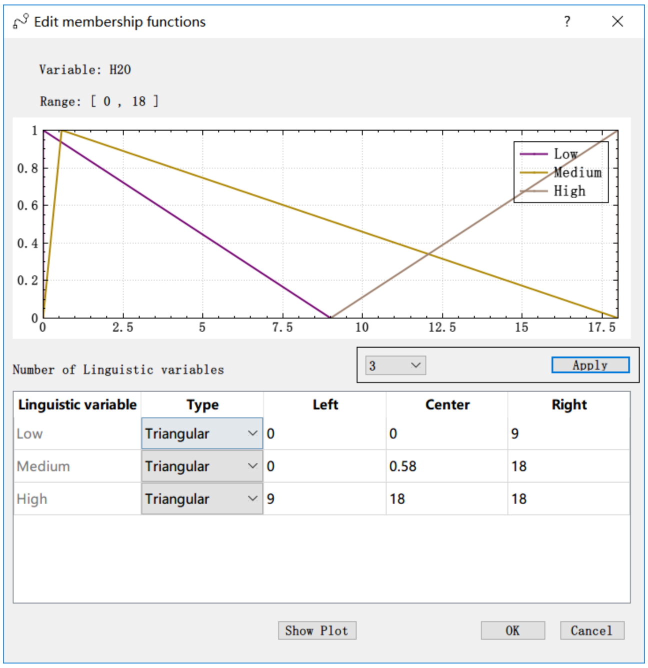


Figure 16: Membership functions of H2O.

Now we should edit rules.

- Click the Edit rules button on the center of bottom.
- Select the items of a rule, and left click Add rule button to add it.
- Repeat the previous step to add rules.
- Click the OK button to save the changes and close the window.

We can select a row and left click Change rule button to change it to the rule you selected in the below. Choosing a row and left clicking the Delete rule button can be used to delete the rule.

Make sure that the meanings of input's linguistic variables and output's linguistic variables are different. For example, as is shown in Figure 17, one of the rules is - IF H2 IS Low AND O2 IS Low Then H20 IS Low. This means that the low values of H2 and O2 will cause low value change of H2O, not meaning that the value of H2O will be low.

The rules are shown in Figure 17.

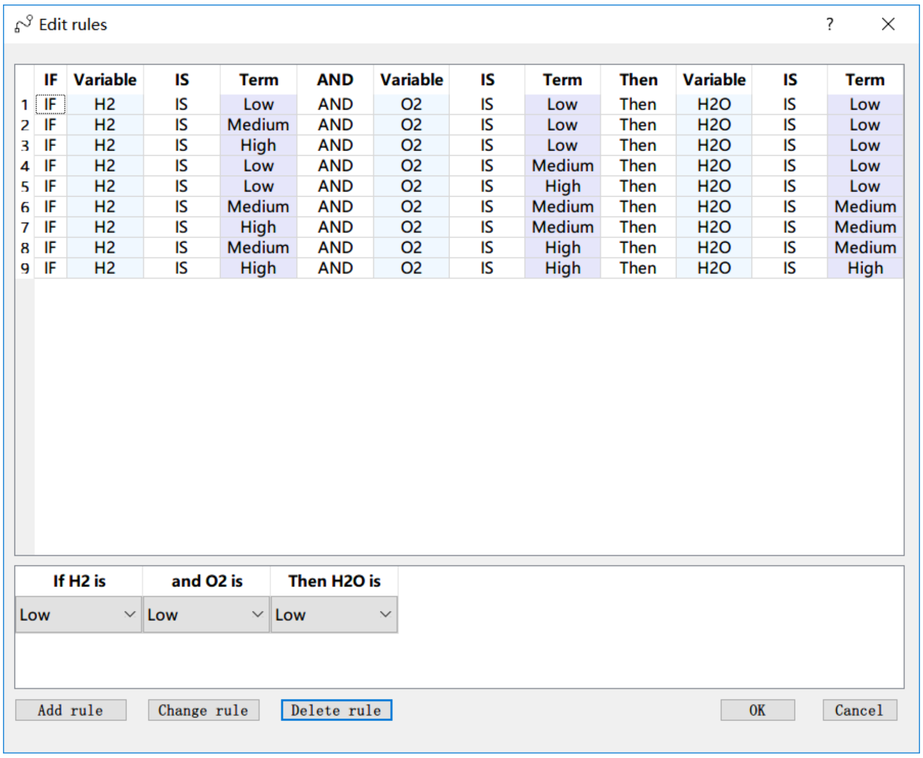


Figure 17: Edit rules.

Now left click the OK button of FIS editor and left click the OK button of arc to return to the main interface. The net should be like Figure 18.

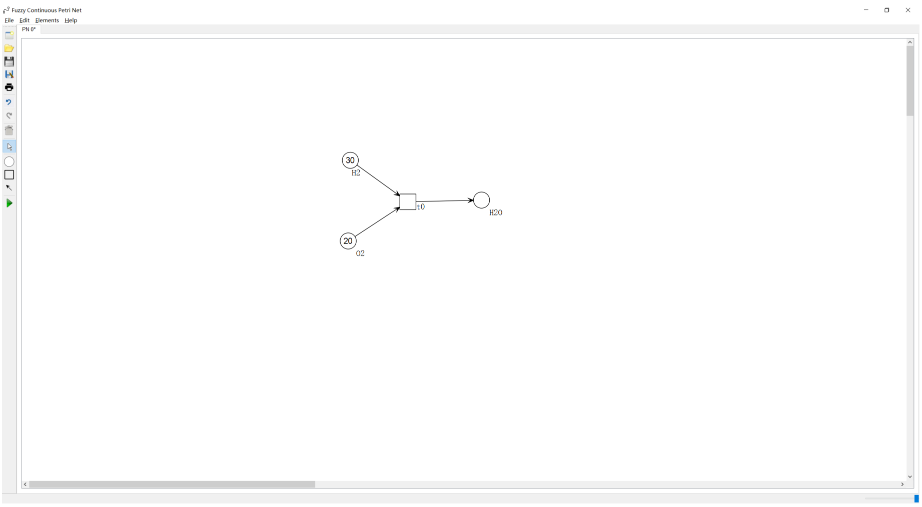


Figure 18: After FIS.

There are some differences between Mamdani and T-S. First, in T-S, when left clicking the Edit button of variable whose type is PN_OUTPUT, there is no need to edit its membership functions.

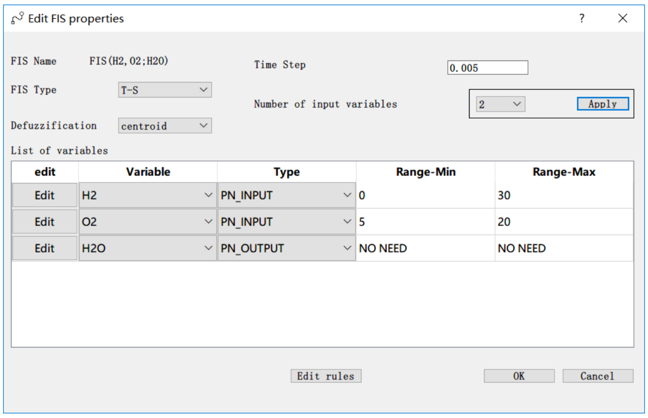


Figure 19: T-S FIS.

Second, when editing rules, the last column should be edited by the user as is shown in Figure 20.

Add rules first and then change the content of last column by double clicking the content of last column of each row and entering the formulas. Then left click OK button to save changes.

Other steps are silimar to Mamdani.

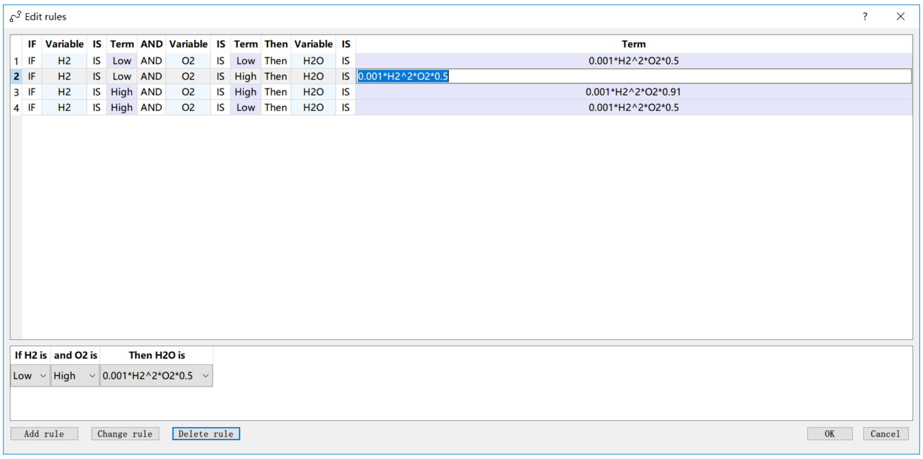


Figure 20: Enter rules when using T-S.

3 Simulation

After modeling, left click the "start simulation" button on the tool bar to use the simulation function.

3.1 Run simulation

In the simulation shown as Figure 21, the user can first set simulation parameters, then left click the Start Simulation button to start simulation. The settings include:

- Setting interval start.
- Setting interval start.
- Choosing ODE solver.
- Setting time step.
- Choosing to check negative value or not.

Please make sure the time step is smaller than FISs' step size. Otherwise the result might be incorrect.

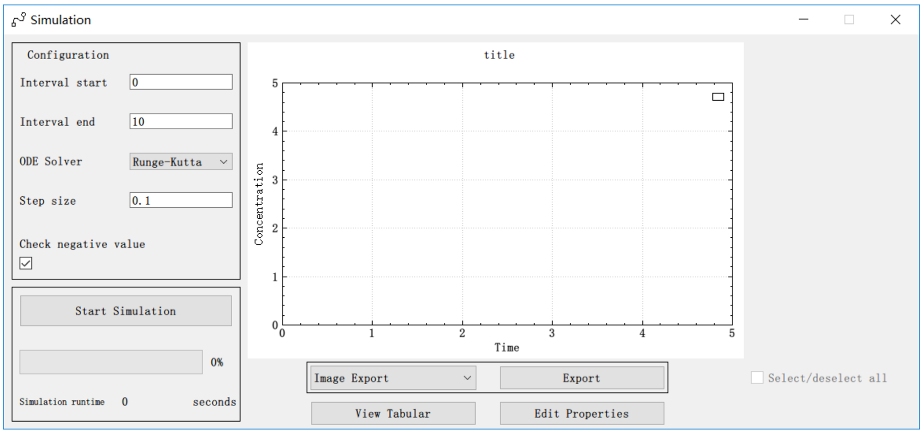


Figure 21: Simulation.

3.2 Show simulation results

During simulation, the progress bar shows the progess. And the time is updated. After simulation, the graph will show. The user can zoom in or zoom out to see the details and check or uncheck the boxes on the right to show or hide the according variables.

To avoid misoperation, after entering the simulation, most of the functions in main interface such like adding items, can not be used. So if you want to be back to main interface, remember to left click "normal cursor" on the left tool bar. Otherwise you will find that most of the functions can't use.

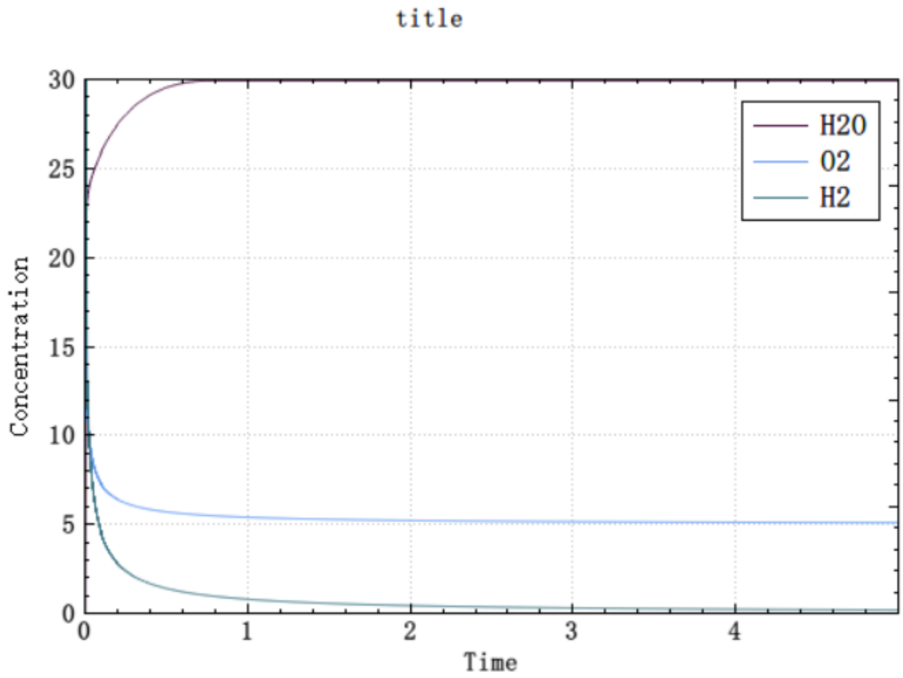


Figure 22: Simulation result (Mamdani).

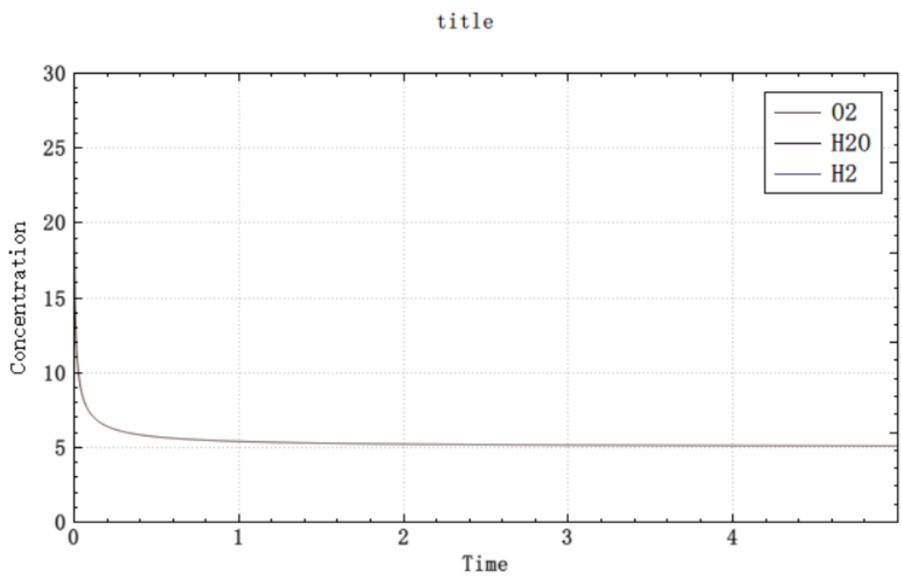


Figure 23: Only show the result of O2.

Actually, the user can choose to use no FIS, Mamdani, T-S, hybrid FIS to run the simulation. Therefore, let's try to use no FIS to run the simulation.

- Close the simulation window.
- Left click “normal cursor”.
- Delete the FIS arc and replace it with a new arc whose expression is 2.
- Start simulation again to see the result.

The result can be a little bit different, but still shows the general trend.

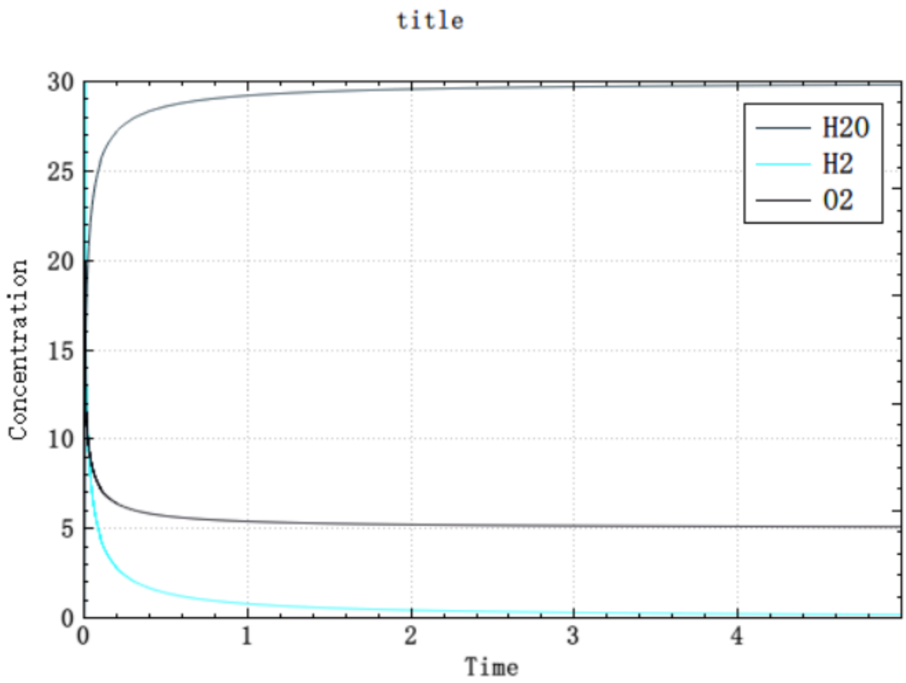


Figure 24: Simulation result (No FIS).

3.3 Export simulation results

The user can choose the export method and left click the Export button to export.

3.4 Rich functions

Different tools are provided where located at the center of bottom. The user can view the Tabular to see the values of variables by left clicking the View Tabular button or edit properties by left clicking the Edit Properties.

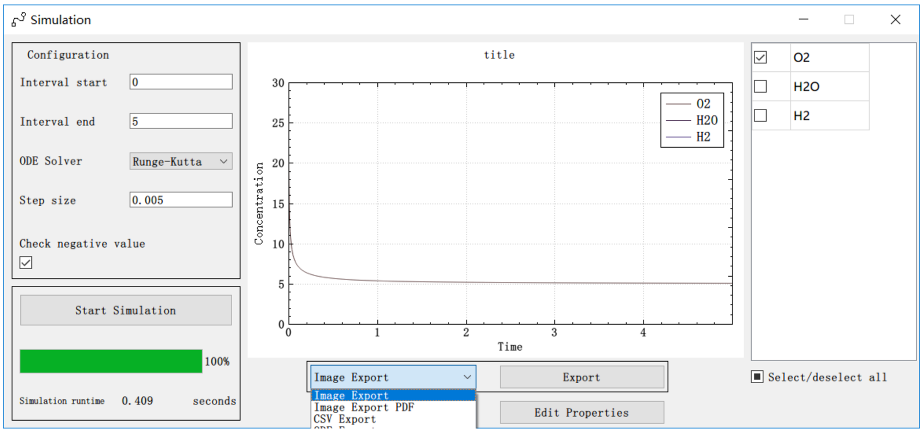


Figure 25: Export result.

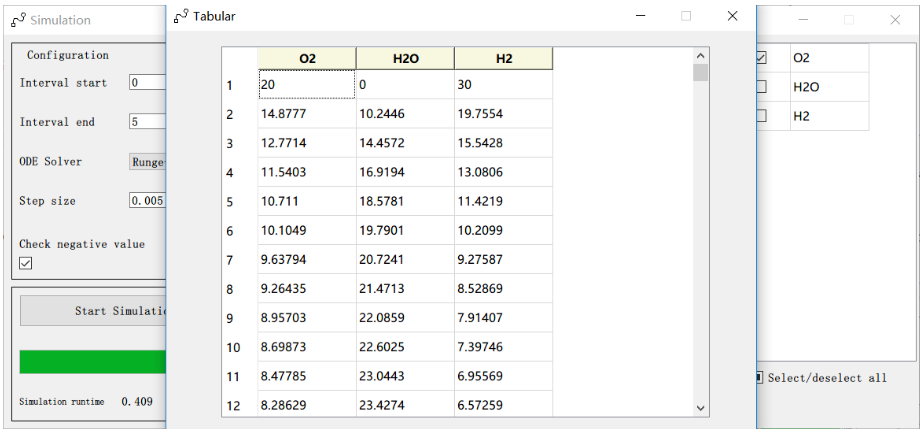


Figure 26: View Tabular.

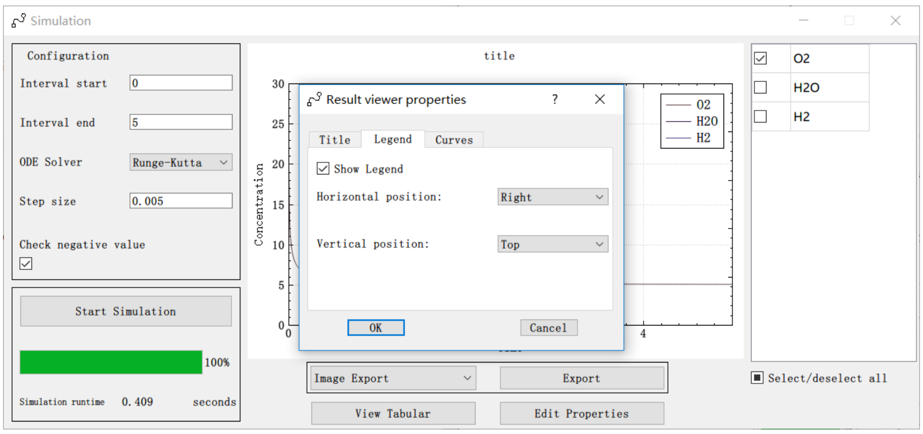


Figure 27: Edit properties.

4 Examples

In this section, we will show four examples. And we provide the file of the examples named A_B_C_D where A is the name of the example, and B is the FIS type we used, and C is the interval end and D is the step size. The default interval start is 0.

Open the application and left click the "Open" button on the tool bar or use Cltr+O to open the file. Then you can run the simulation.

4.1 1D Diffusion Reaction

4.1.1 Introduction

The first example is 1D diffusion reaction.

4.1.2 Model

The model is shown as Figure 28.

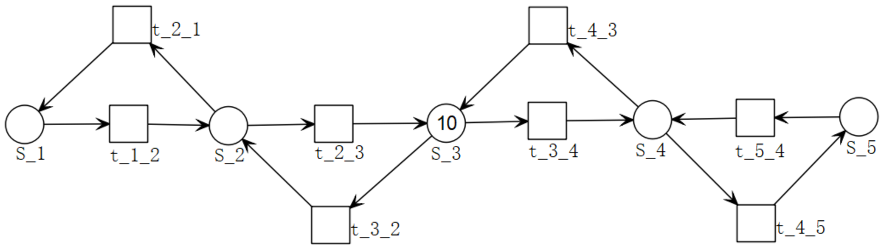


Figure 28: The model of 1D diffusion reaction.

Table 2: Transition functions of 1D Diffusion Reaction.

Transition	Function
t_1_2	MassAction(1)
t_2_1	MassAction(1)
t_2_3	MassAction(1)
t_3_2	MassAction(1)
t_3_4	MassAction(1)
t_4_3	MassAction(1)
t_4_5	MassAction(1)
t_5_4	MassAction(1)

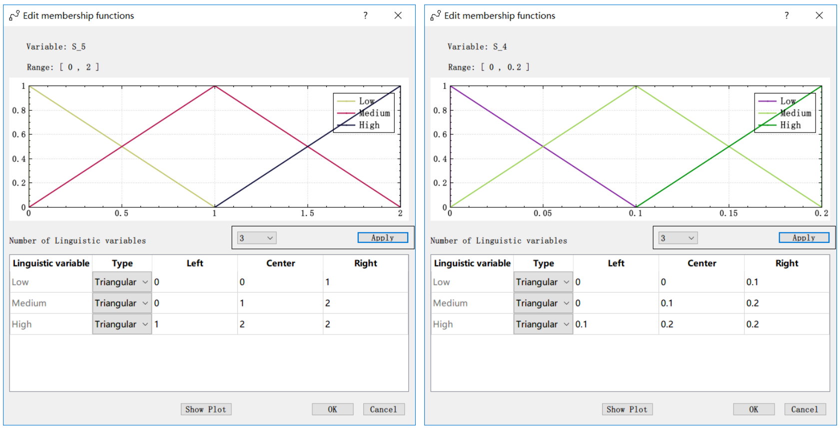


Figure 29: Membership functions of 1D Diffusion Reaction using Mamdani.

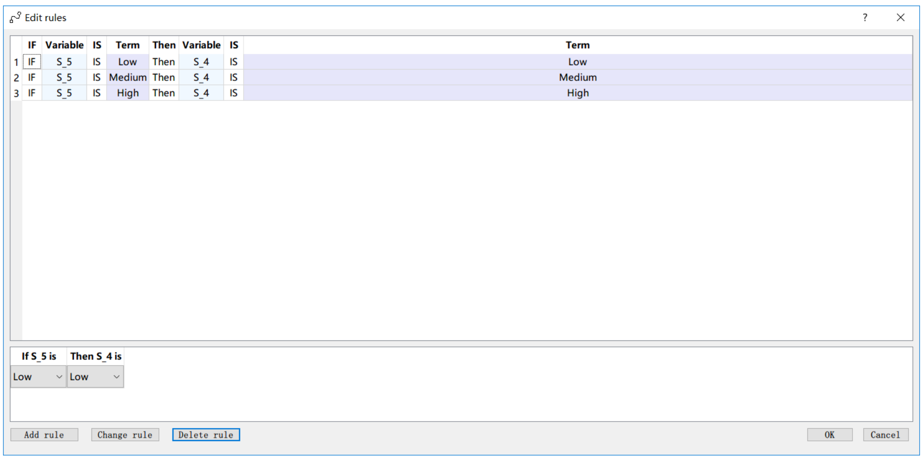


Figure 30: Rules of 1D Diffusion Reaction using Mamdani.

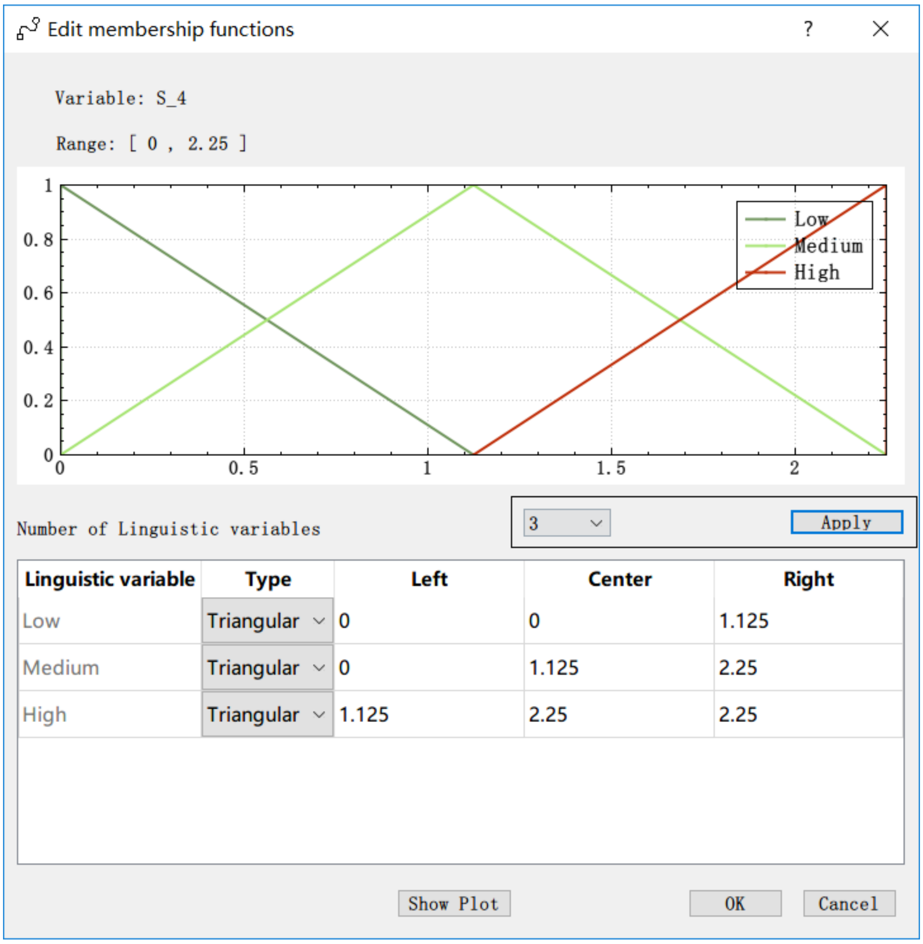


Figure 31: Membership functions of 1D Diffusion Reaction using T-S.

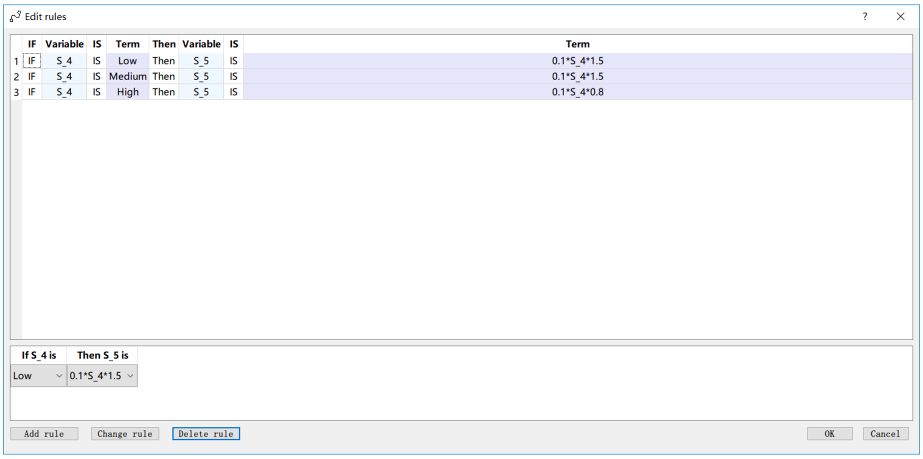


Figure 32: Rules of 1D Diffusion Reaction using T-S.

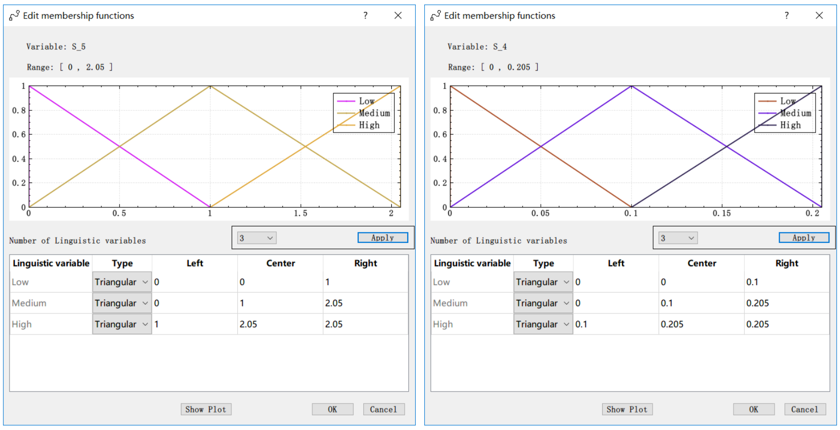


Figure 33: Membership functions of 1D Diffusion Reaction using hybrid FIS (Mamdani).

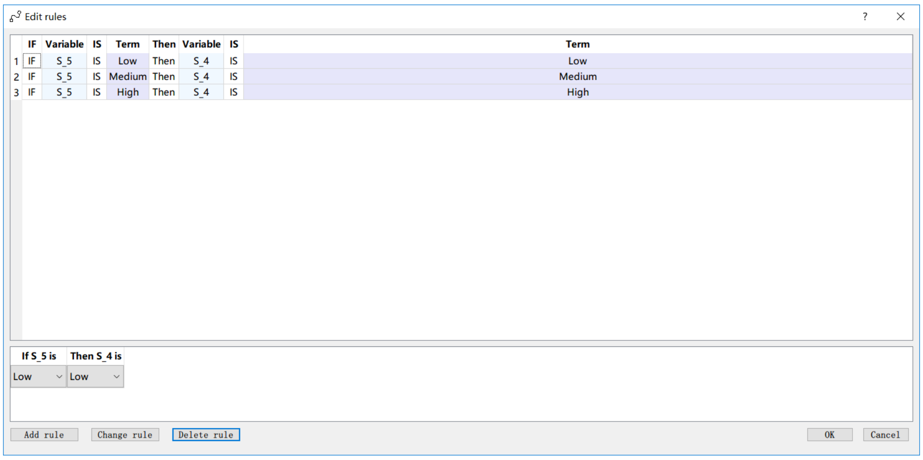


Figure 34: Rules of 1D Diffusion Reaction using hybrid FIS (Mamdani).

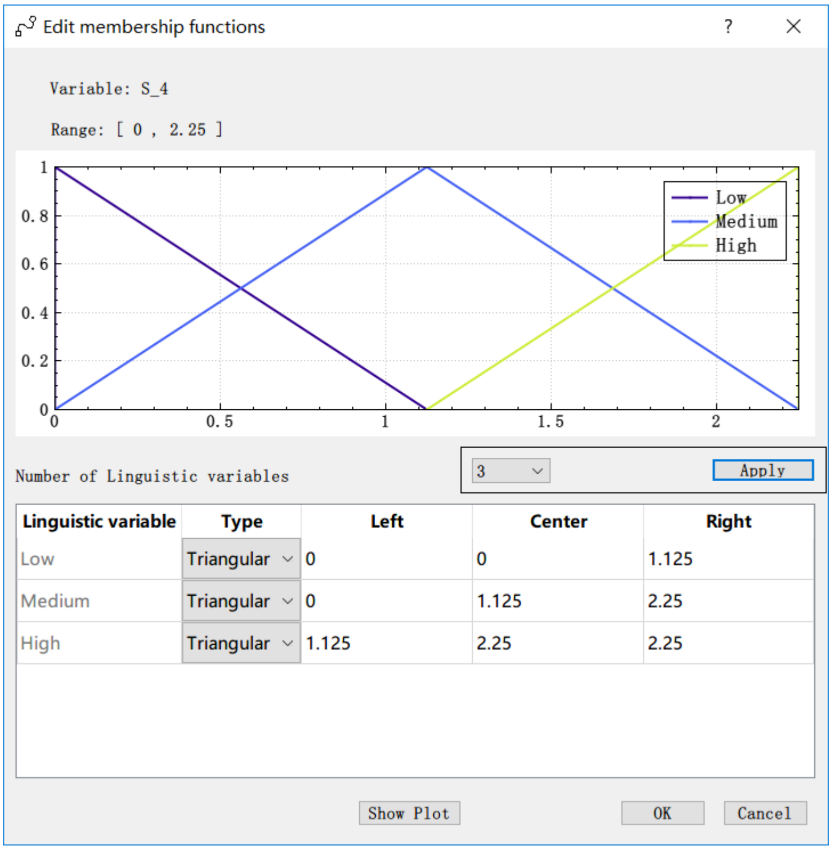


Figure 35: Membership functions of 1D Diffusion Reaction using hybrid FIS (T-S).

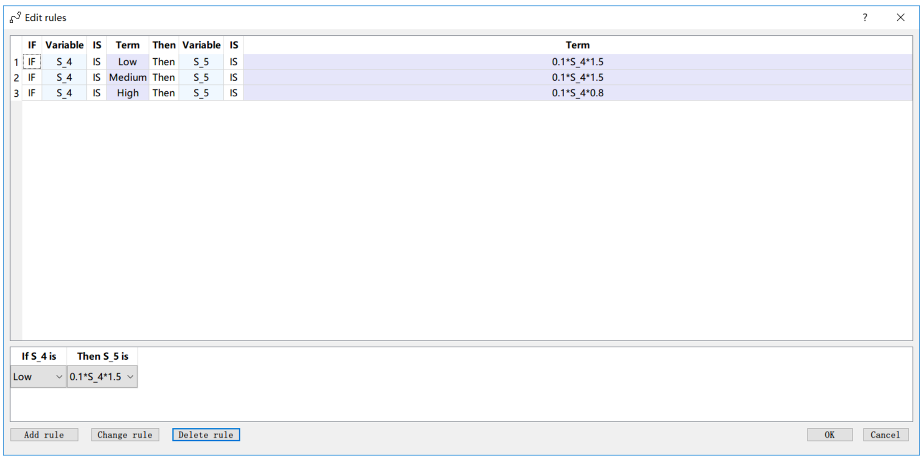


Figure 36: Rules of 1D Diffusion Reaction using hybrid FIS (T-S).

4.1.3 Simulation result

In this example, we use only ODEs, Mamdani, T-S, and hybrid FIS. The result of only ODEs is shown as Figure 37.

When using Mamdani, we choose to change the arc from t_5_4 to S_4 to FIS. The result is shown as Figure 38.

When using T-S, we choose to change the arc from t_4_5 to S_5 to FIS. The result is shown as Figure 39.

When using hybrid FIS, we choose to change the arc from t_5_4 to S_4 to Mamdani and change the arc from t_4_5 to S_5 to T-S. The result is shown as Figure 40.

In method of only ODEs, the value of all variables eventually becomes 2 while in other three methods, the final values of the variables are not 2, but are all very close to 2. And the trends of curves are consistent.

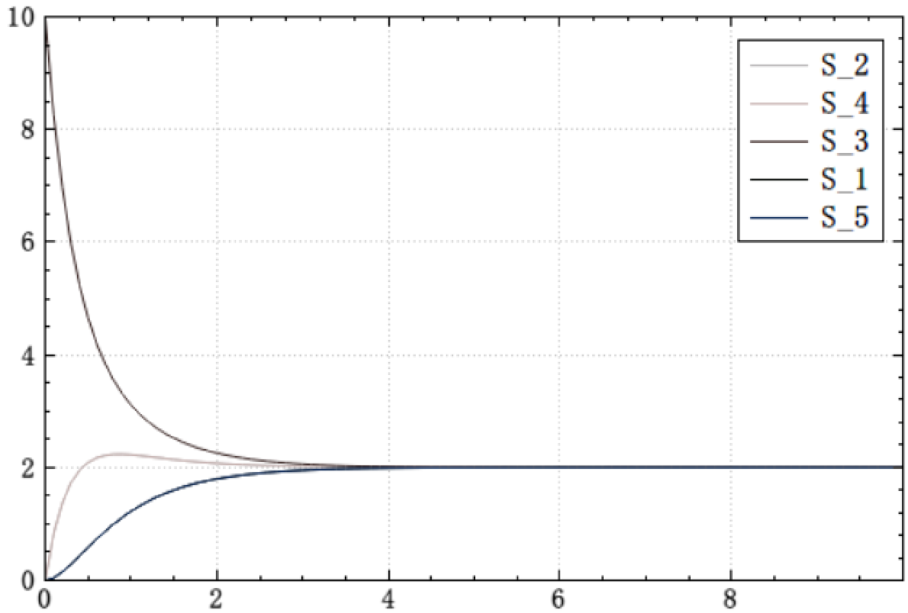


Figure 37: Simulation result of 1D diffusion reaction using only ODEs.

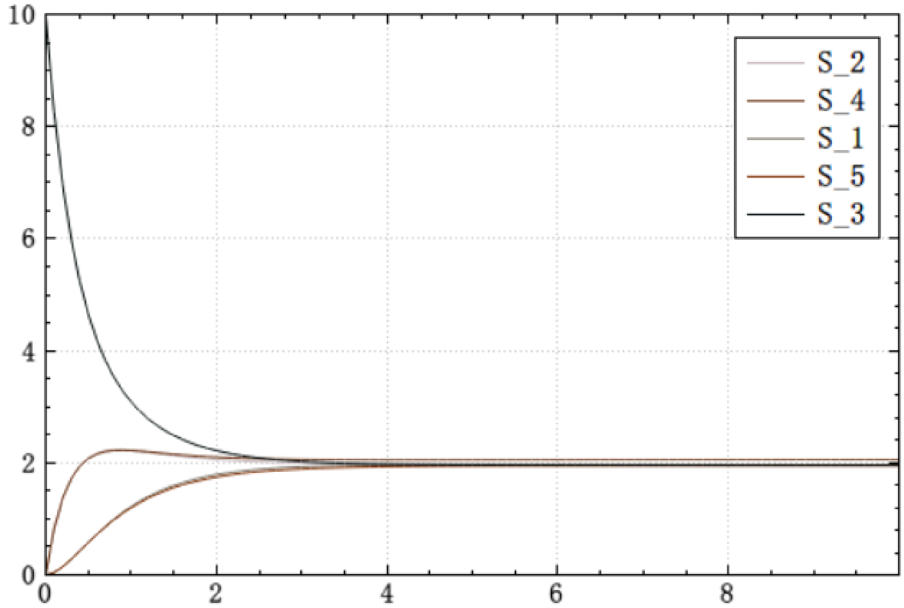


Figure 38: Simulation result of 1D diffusion reaction using Mamdani.

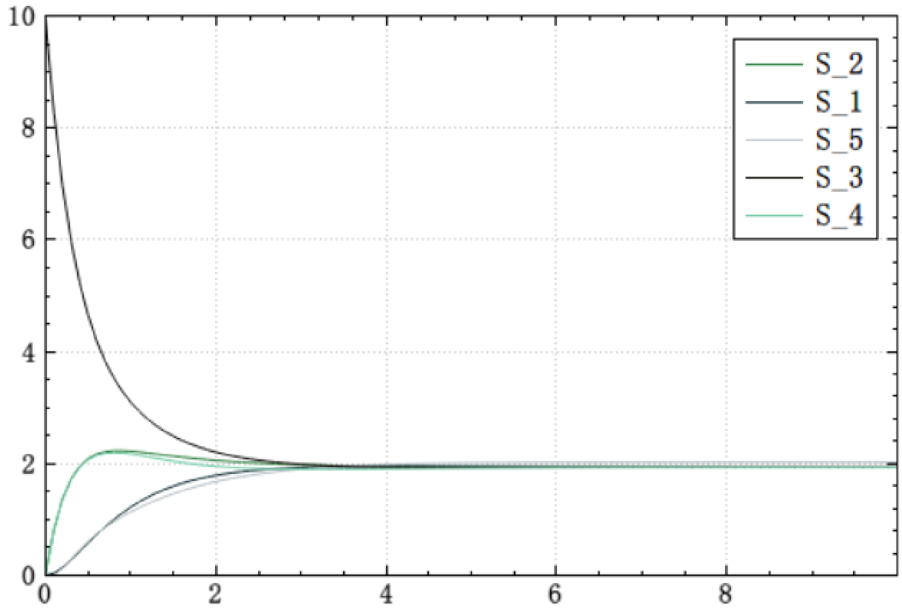


Figure 39: Simulation result of 1D diffusion reaction using T-S.

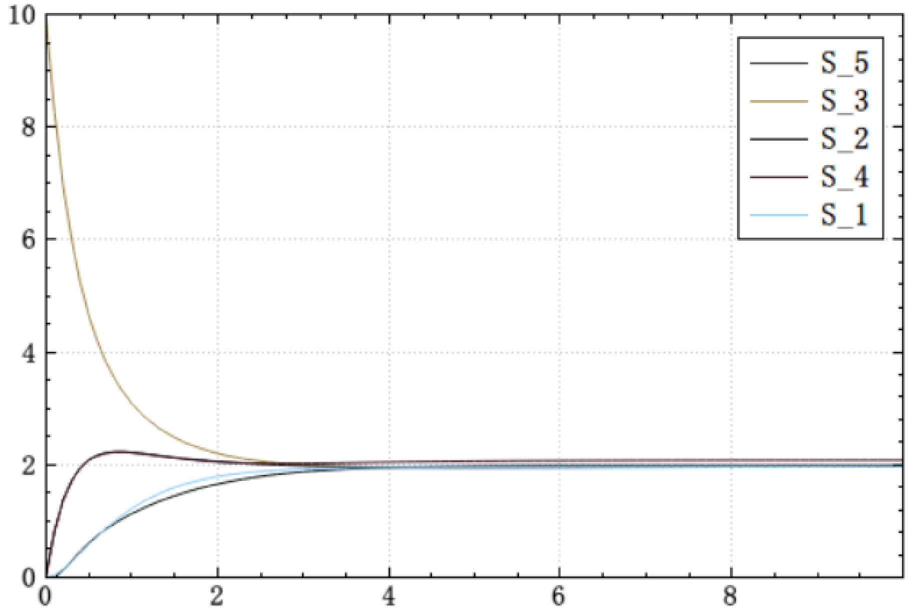


Figure 40: Simulation result of 1D diffusion reaction using hybrid FIS.

4.2 Enzyme

4.2.1 Introduction

The second example is enzyme. Enzymes are macromolecular biological catalysts. Enzymes accelerate chemical reactions. The molecules upon which enzymes may act are called substrates and the enzyme converts the substrates into different molecules known as products. Almost all metabolic processes in the cell need enzyme catalysis in order to occur at rates fast enough to sustain life. Metabolic pathways depend upon enzymes to catalyze individual steps. [6]

4.2.2 Model

The model is shown as Figure 41.

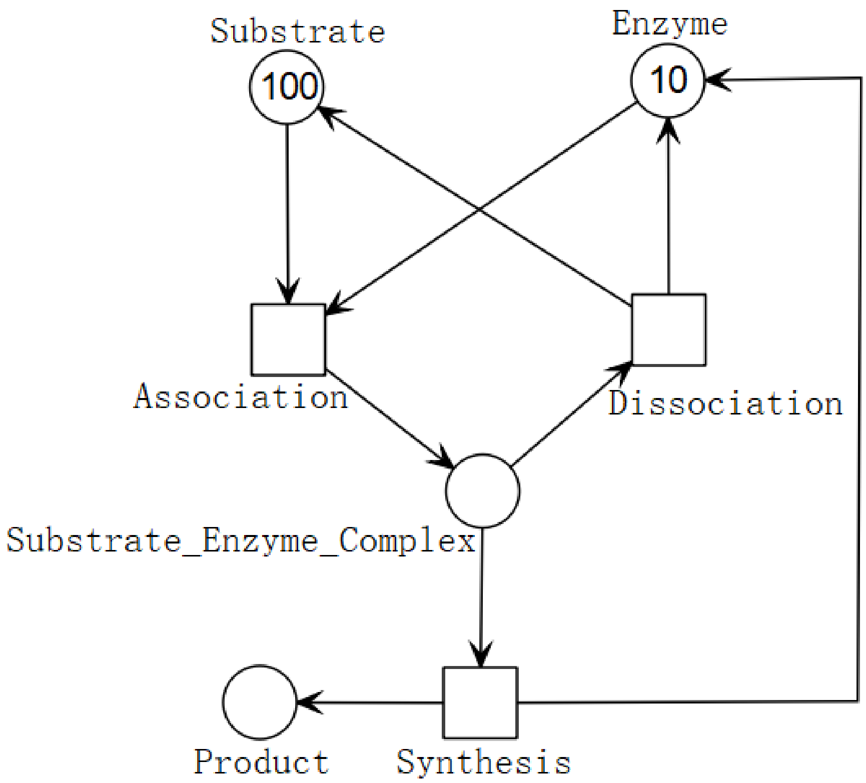


Figure 41: The model of enzyme.

4.2.3 Simulation result

In this example, we use only ODEs and Mamdani. The result of only ODEs is shown as Figure 44.

When using Mamdani, we choose to change the arc from Synthesis to Product to FIS. The result is shown as Figure 45.

As we can see, the results are almost exactly the same.

Table 3: Transition functions of enzyme.

Transition	Function
Assoication	MassAction(0.1)
Dissociation	MassAction(0.1)
Synthesis	MassAction(0.1)

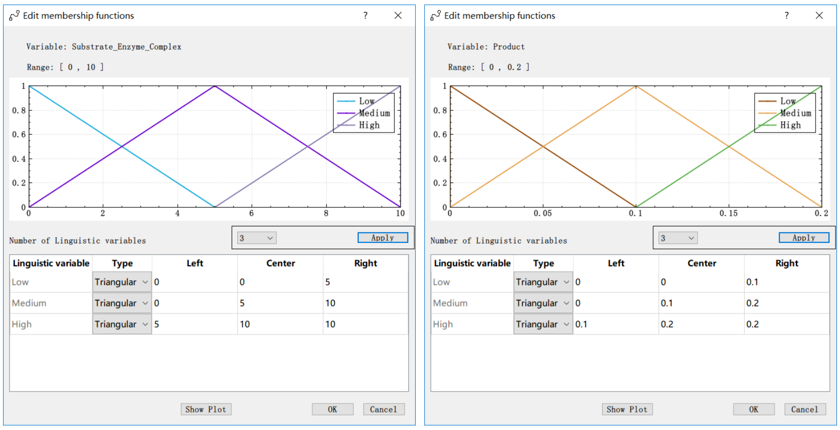


Figure 42: Membership functions of enzyme using Mamdani.

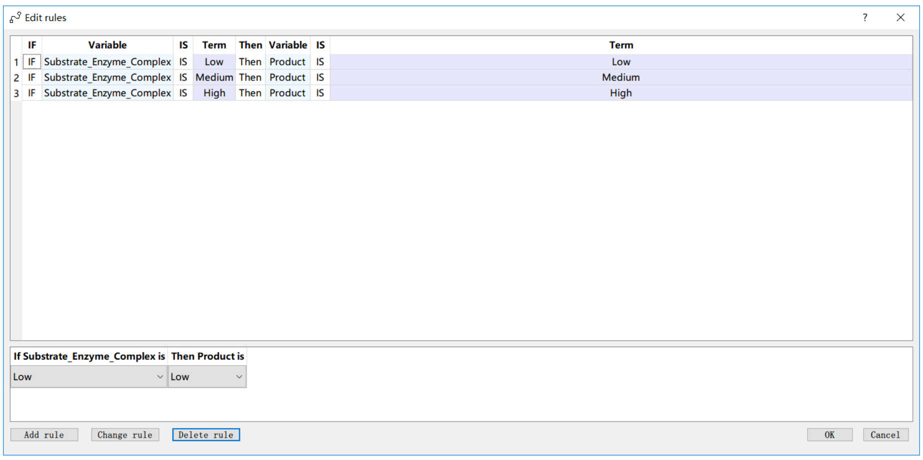


Figure 43: Rules of enzyme using Mamdani.

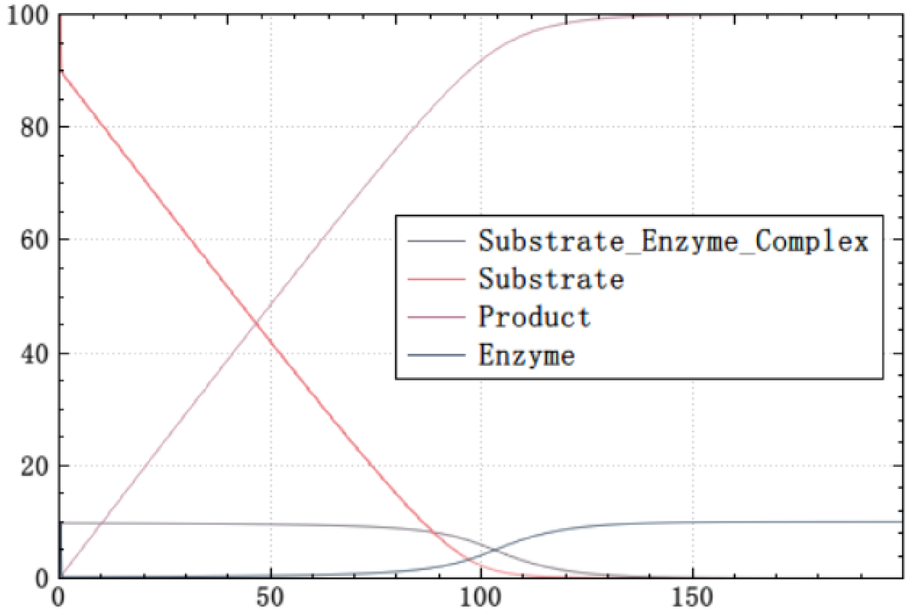


Figure 44: Simulation result of enzyme using only ODEs.

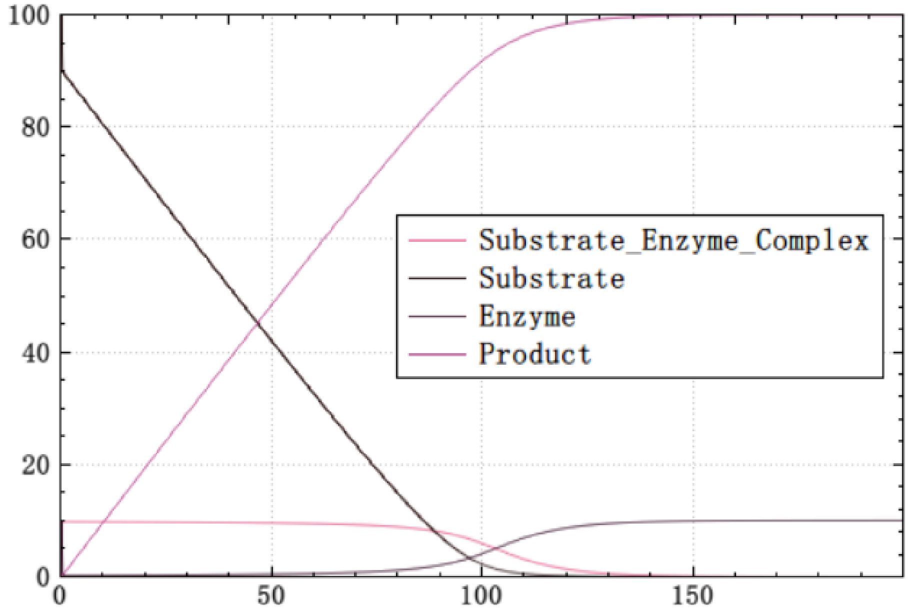


Figure 45: Simulation result of enzyme using Mamdani.

4.3 RKIP

4.3.1 Introduction

The third example is RKIP. The Raf kinase inhibitor protein (RKIP) is a kinase inhibitor protein, that regulates many signaling pathways within the cell. RKIP is a member of the phosphatidylethanolamine-binding protein family and has displayed disruptive regulation on the Raf-1-MEK1/2, ERK1/2 and NF-kappaB signalling pathways, by interaction with the Raf-1 kinase. [7]

4.3.2 Model

The model is shown as Figure 46.

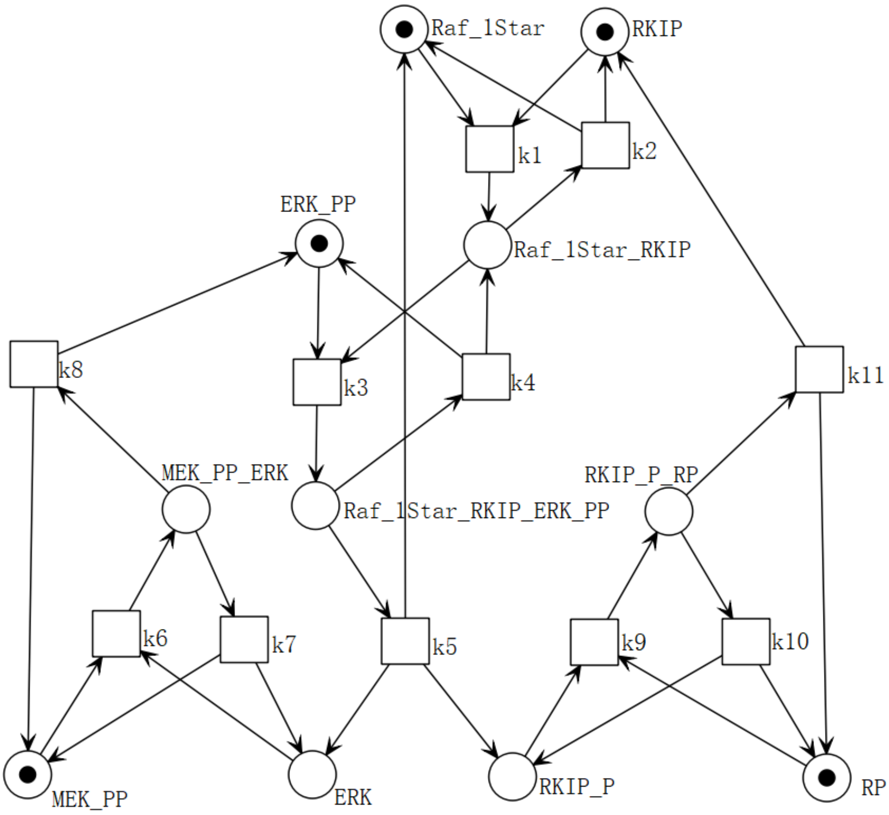


Figure 46: The model of RKIP.

4.3.3 Simulation result

In this example, we use only ODEs and T-S. The result of only ODEs is shown as Figure 49.

When using T-S, we choose to change the arc from k3 to Raf_1Star_RKIP_ERK_PP to FIS. The result is shown as Figure 50.

Table 4: Transition functions of RKIP.

Transition	Function
k1	MassAction(0.53)
k2	MassAction(0.0072)
k3	MassAction(0.625)
k4	MassAction(0.00245)
k5	MassAction(0.0315)
k6	MassAction(0.6)
k7	MassAction(0.0075)
k8	MassAction(0.071)
k9	MassAction(0.92)
k10	MassAction(0.00122)
k11	MassAction(0.87)

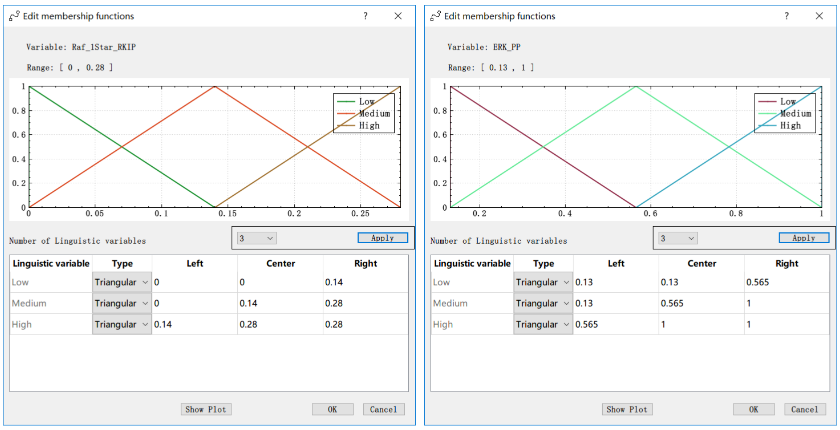
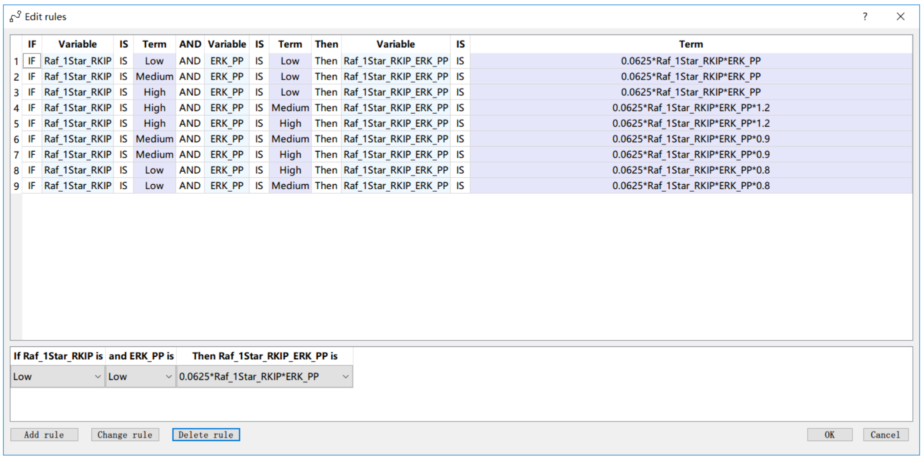


Figure 47: Membership functions of RKIP using T-S.

As we can see, the results are almost consisitent. Pay attention to the bold blue curves of Figure 49 and Figure 50. The final result is a little different. But we can adjust the rules to get a better result.



	IF	Variable	IS	Term	AND	Variable	IS	Term	Then	Variable	IS	Term
1	IF	Raf_1Star_RKIP	IS	Low	AND	ERK_PP	IS	Low	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP
2	IF	Raf_1Star_RKIP	IS	Medium	AND	ERK_PP	IS	Low	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP
3	IF	Raf_1Star_RKIP	IS	High	AND	ERK_PP	IS	Low	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP
4	IF	Raf_1Star_RKIP	IS	High	AND	ERK_PP	IS	Medium	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*1.2
5	IF	Raf_1Star_RKIP	IS	High	AND	ERK_PP	IS	High	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*1.2
6	IF	Raf_1Star_RKIP	IS	Medium	AND	ERK_PP	IS	Medium	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*0.9
7	IF	Raf_1Star_RKIP	IS	Medium	AND	ERK_PP	IS	High	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*0.9
8	IF	Raf_1Star_RKIP	IS	Low	AND	ERK_PP	IS	High	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*0.8
9	IF	Raf_1Star_RKIP	IS	Low	AND	ERK_PP	IS	Medium	Then	Raf_1Star_RKIP_ERK_PP	IS	0.0625*Raf_1Star_RKIP*ERK_PP*0.8

Figure 48: Rules of RKIP using T-S.

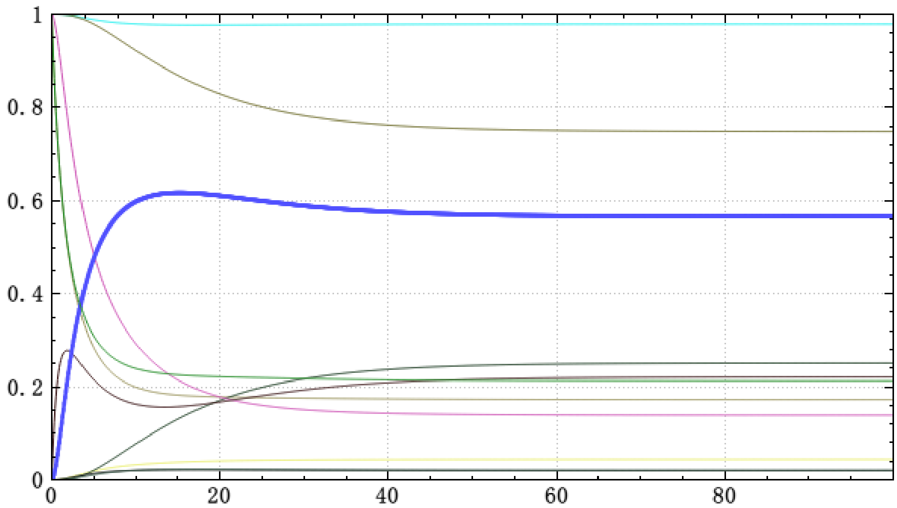


Figure 49: Simulation result of RKIP using only ODEs.

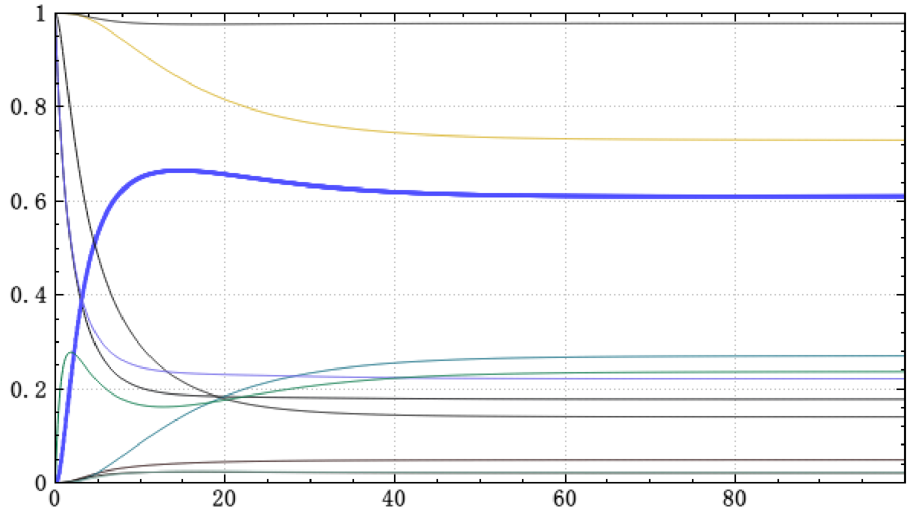


Figure 50: Simulation result of RKIP using T-S.

4.4 6-MP metabolism

4.4.1 Introduction

The last example is 6-MP metabolism. 6-Mercaptopurine (6-MP) is one of the important chemotherapy drugs used for treating acute lymphocytic leukaemia (ALL). According to the model given in [8][9], we build the FCPN.

4.4.2 Model

The model is shown as Figure 51.

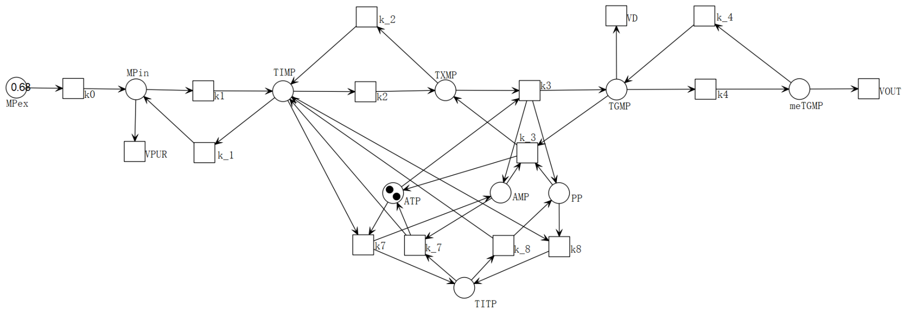


Figure 51: The model of 6-MP metabolism.

Table 5: Transition functions of 6-MP metabolism.

Transition	Function
k0	MassAction(5)
k1	MassAction(10)
k_1	MassAction(0.01)
k2	MassAction(10)
k_2	MassAction(4)
k3	MassAction(5)
k_3	MassAction(0.01)
k4	MassAction(0.00001)
k_4	MassAction(0.1)
k7	MassAction(0.01)
k_7	MassAction(1)
k8	MassAction(0.5)
k_8	MassAction(0.01)
VPUR	MassAction(0.01)
VD	MassAction(0.9)
VOUT	MassAction(0.0001)

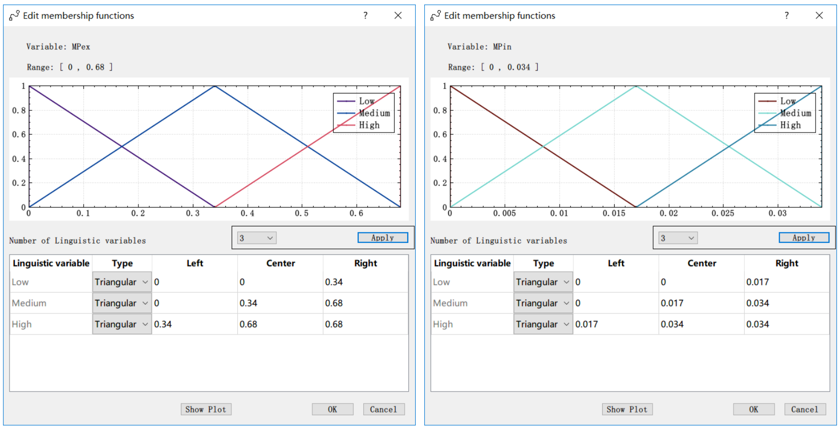


Figure 52: Membership functions of 6-MP metabolism using Mamdani.

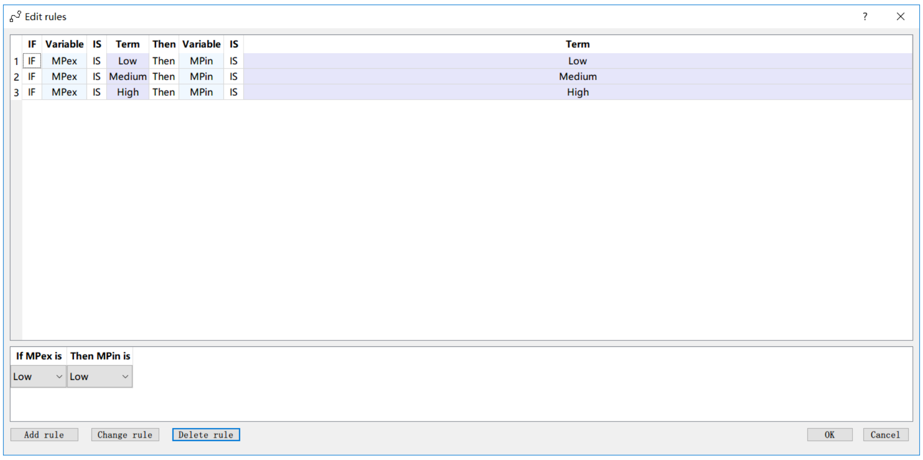


Figure 53: Rules of 6-MP metabolism using Mamdani.

4.4.3 Simulation result

In this example, we use only ODEs, Mamdani, and T-S. The result of only ODEs is shown as Figure 56.

When using Mamdani, we choose to change the arc from k0 to MPin to FIS. The result is shown as Figure 57.

When using T-S, we choose to change the arc from k0 to MPin to FIS. The result is shown as Figure 58.

As we can see, the results are almost exactly the same.

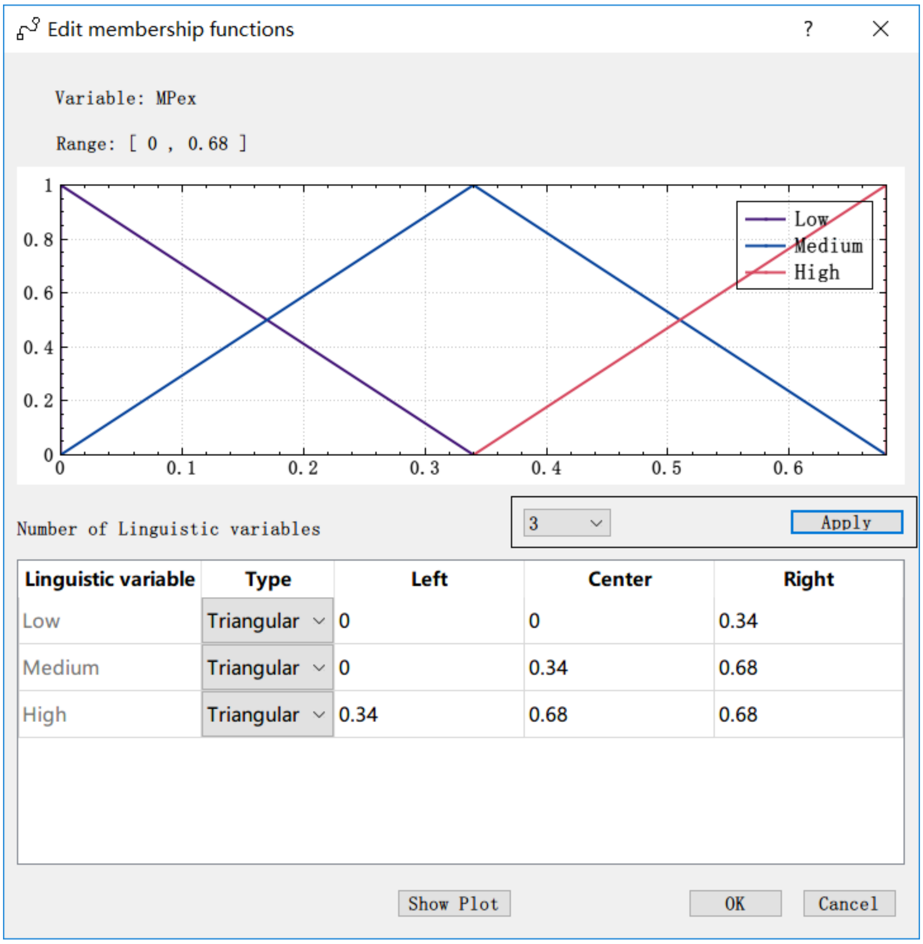


Figure 54: Membership functions of 6-MP metabolism using T-S.

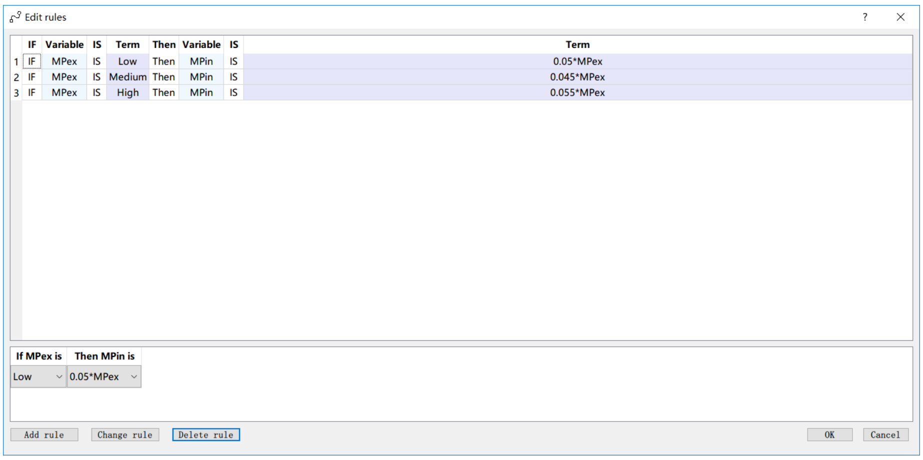


Figure 55: Rules of 6-MP metabolism using T-S.

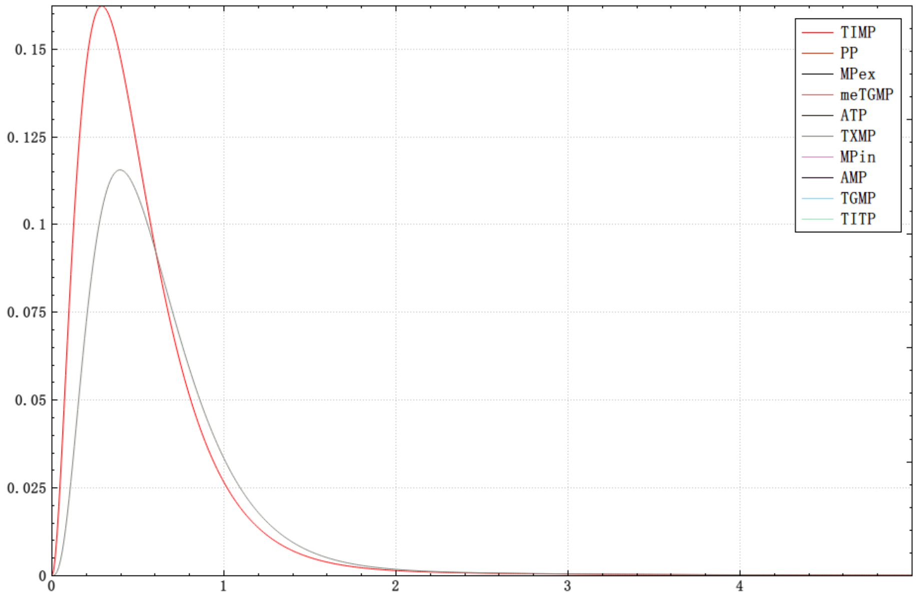


Figure 56: Simulation result of 6-MP metabolism using only ODEs.

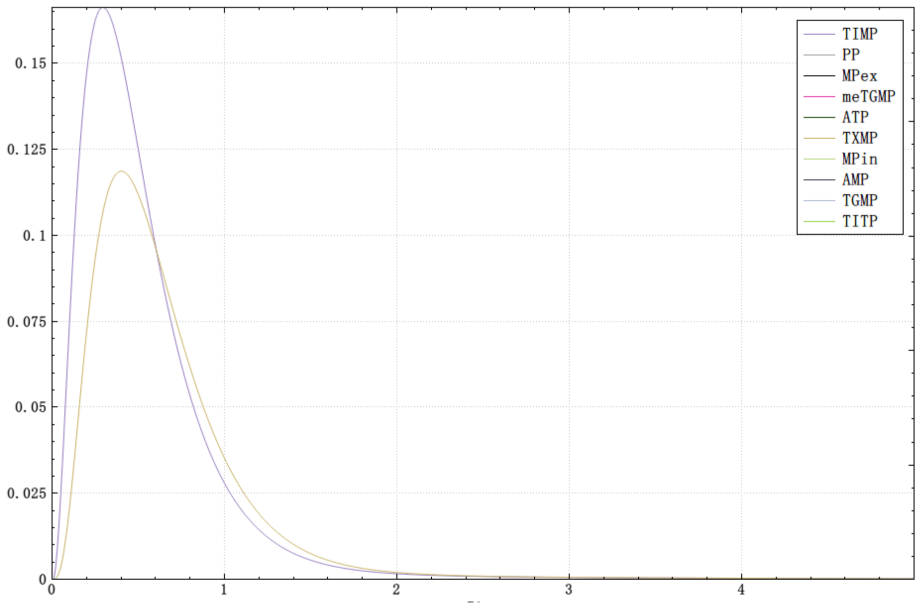


Figure 57: Simulation result of 6-MP metabolism using Mamdani.

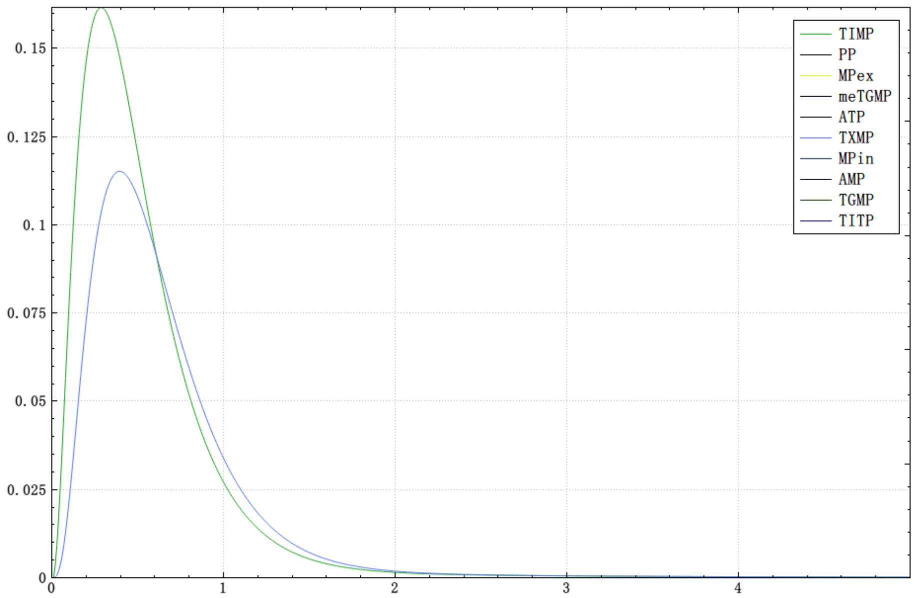


Figure 58: Simulation result of 6-MP metabolism using T-S.

References

[1] Petri net. https://en.wikipedia.org/wiki/Petri_net. Accessed: 2018-09-29.

[2] Fuzzy logic. https://en.wikipedia.org/wiki/Fuzzy_logic. Accessed: 2018-09-29.

[3] Fuzzy inference process. https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html. Accessed: 2018-09-29.

[4] Mamdani-type fuzzy inference. https://www.mathworks.com/help/fuzzy/what-is-mamdani-type-fuzzy-inference.html. Accessed: 2018-09-29.

[5] Sugeno-type fuzzy inference. https://www.mathworks.com/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html. Accessed: 2018-09-29.

[6] Enzyme. https://en.wikipedia.org/wiki/Enzyme. Accessed: 2018-09-29.

[7] Raf kinase inhibitor protein. https://en.wikipedia.org/wiki/Raf_kinase_inhibitor_protein. Accessed: 2018-09-29.

[8] Anastasia I Lavrova, Eugene B Postnikov, Andrey Yu Zyubin, and Svetlana V Babak. Ordinary differential equations and boolean networks in application to modelling of 6-mercaptopurine metabolism. *Royal Society open science*, 4(4):160872, 2017.

[9] Anastasia I Lavrova, Eugene B Postnikov, Andrey Yu Zyubin, and Svetlana V Babak. Ode and random boolean networks in application to modelling of 6-mercaptopurine metabolism. *arXiv preprint arXiv:1611.00054*, 2016.