**An open source Matlab/SIMULINK Toolbox For Interval Type-2 Fuzzy Logic Systems**

Tufan Kumbasar

Department of Control and Automation Engineering

Istanbul Technical University

Maslak, TR-34469, Istanbul, Turkey

kumbasart@itu.edu.tr

Ahmet Taskin

Department of Software and Electronics

AVL Research and Engineering

Sancaktepe, TR-34885, Istanbul, Turkey

ahmet.taskin@avl.com

*Abstract*—In the last two decades, we have witnessed that interval type-2 fuzzy logic systems have been successfully implemented in various engineering areas. The main advantages of the interval type-2 fuzzy logic systems are the better handling of the uncertainties than type-1 fuzzy logic systems. However, it is quite complicated needs many calculations to implement all phases of an interval type-2 fuzzy logic system for an application. This paper introduced an open source toolbox for interval type-2 fuzzy logic systems to cover all phases of an interval type-2 fuzzy logic systems from the beginning to the end. The introduced interval type-2 fuzzy logic toolbox makes possible to design an interval type-2 fuzzy logic system by using the user interfaces of the toolbox. In addition, type reduction phase is one of the important part of an interval type-2 fuzzy logic systems, and most known type reduction methods are embedded to the toolbox and it is possible to choose desired. Also, one of the main advantage of the introduced toolbox is automatic connection to Matlab/SIMULINK. After finishing the interval type-2 fuzzy logic system design, it is possible to export it to the Matlab/SIMULINK environment automatically. The proposed toolbox is created in the same format of the Matlab Fuzzy Logic Toolbox so it is easy to use it for the users of the Matlab Fuzzy Logic Toolbox

Keywords—Interval type-2 fuzzy logic system, type-reduction, user interface, toolbox

# Introduction

Type-2 fuzzy sets (T2-FSs) were first introduced by Zadeh as an extension of the type-1 fuzzy sets (T1-FSs) in 1975 (Zadeh, 1975). The memberships in a type-1 fuzzy sets (T1-FSs) T1-FS are crisp values; however, the memberships in a T2-FS are T1-FS (Wagner and Hagras, 2010). Recently, type-2 fuzzy sets (T2-FSs) and type-2 fuzzy logic systems (T2-FLSs) are used in many areas. However, it is not so easy to describe mathematically the T2-FSs due to their additional dimension (Mendel, 2007). Therefore, people are interested in interval type-2 fuzzy sets (IT2-FSs), whose memberships are interval instead of T1-FSs in a general T2-FS (Mendel, 2000). This brings many advantages like simplicity and reduced computational cost.

Type-2 fuzzy sets (T2-FSs) were first introduced by Zadeh as an extension of the type-1 fuzzy sets (T1-FSs) in 1975 (Zadeh, 1975). The memberships in a type-1 fuzzy sets (T1-FSs) T1-FS are crisp values; however, the memberships in a T2-FS are T1-FS (Wagner and Hagras, 2010). Recently, type-2 fuzzy sets (T2-FSs) and type-2 fuzzy logic systems (T2-FLSs) are used in many areas. However, it is not so easy to describe mathematically the T2-FSs due to their additional dimension (Mendel, 2007). Therefore, people are interested in interval type-2 fuzzy sets (IT2-FSs), whose memberships are interval instead of T1-FSs in a general T2-FS (Mendel, 2000). This brings many advantages like simplicity and reduced computational cost.

Type-2 fuzzy sets (T2-FSs) were first introduced by Zadeh as an extension of the type-1 fuzzy sets (T1-FSs) in 1975 (Zadeh, 1975). The memberships in a type-1 fuzzy sets (T1-FSs) T1-FS are crisp values; however, the memberships in a T2-FS are T1-FS (Wagner and Hagras, 2010). Recently, type-2 fuzzy sets (T2-FSs) and type-2 fuzzy logic systems (T2-FLSs) are used in many areas. However, it is not so easy to describe mathematically the T2-FSs due to their additional dimension (Mendel, 2007). Therefore, people are interested in interval type-2 fuzzy sets (IT2-FSs), whose memberships are interval instead of T1-FSs in a general T2-FS (Mendel, 2000). This brings many advantages like simplicity and reduced computational cost.

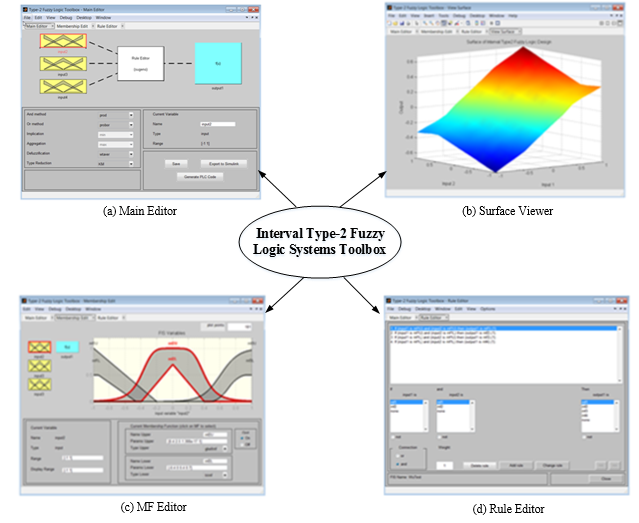
Type-2 fuzzy sets (T2-FSs) were first introduced by Zadeh as an extension of the type-1 fuzzy sets (T1-FSs) in 1975 (Zadeh, 1975). The memberships in a type-1 fuzzy sets (T1-FSs) T1-FS are crisp values; however, the memberships in a T2-FS are T1-FS (Wagner and Hagras, 2010). Recently, type-2 fuzzy sets (T2-FSs) and type-2 fuzzy logic systems (T2-FLSs) are used in many areas. However, it is not so easy to describe mathematically the T2-FSs due to their additional dimension (Mendel, 2007). Therefore, people are interested in interval type-2 fuzzy sets (IT2-FSs), whose memberships are interval instead of T1-FSs in a general T2-FS (Mendel, 2000). This brings many advantages like simplicity and reduced computational cost.

# Interval Type-2 Fuzzy Sets and Fuzzy Logic Systems

## Interval Type-2 Fuzzy Sets

Interval Type-2 Fuzzy Sets (IT2-FSs) are special case of the T2-FSs. Due to their reduced computational cost, it has attracted interest of the researchers. In this section, background materials on the IT2-FSs are given. An IT2-FS can be expressed as in the equation 1.

Fig 1. Overview of the user interfaces of the IT2-FLS toolbox



  

In this equation, *x* is the primary variablehas domain, *u* is the secondary variable has domain and is the support of the secondary membership function. In addition, amplitude of is a secondary grade of and equail to 1 for and (Wu, 2013).

## Interval Type-2 Fuzzy Logic Systems

An interval type-2 fuzzy logic system (IT2-FLS) is a FLS that contains at least one IT2-FS. IT2-FLSs (or T2FLSs) consist of five main parts:

• Fuzzification

• Rule base

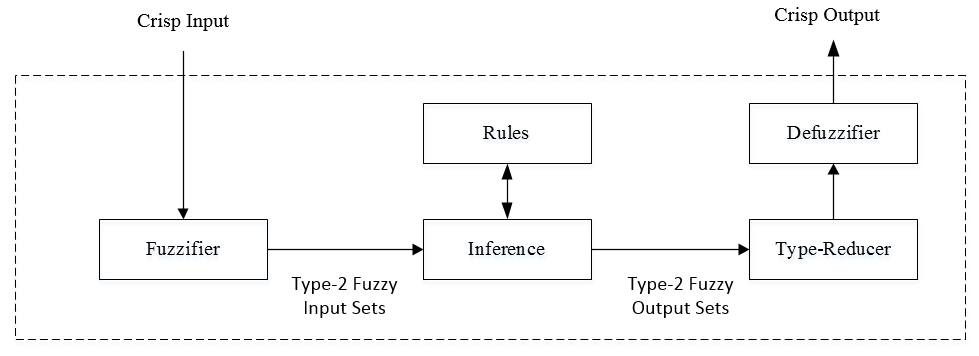
• Inference mechanism

• Type Reducer

• Defuzzification

IT2-FLSs have an additional part that is called as type-reducer, while T1-FLSs do not have type-reducer. Type-reducer is necessary for the IT2-FLSs because defuzzifier block can operates by using the type-1 fuzzy sets, so type-2 fuzzy sets have to be converted to type-1 fuzzy sets before the defuzzifier block. The output of the type-reducer block is called as type-reduced set and then defuzzifier uses the type reduced set for defuzzification process. A simple block diagram of an IT2-FLS can be seen in Figure 2.

Fig 2. Block diagram of an IT2-FLs



Let’s consider a rule base of an IT2-FLS that contains N rules as below:

If is and … and is , then is (2)

Where are IT2-FSs and is the interval output. For an input vector classical steps of the IT2-FLS can be summarized as below:

1. Membership interval of each on each is computed
2. Firing interval of the nth rule is computed:

(3)

(4)

1. Type reduction (TR) method is performed to combine and corresponding rule consequents.

(5)

(6)

(7)

1. Defuzzified (crisp) output is computed:

(8)

# Interval Type-2 Fuzzy Logic Toolbox

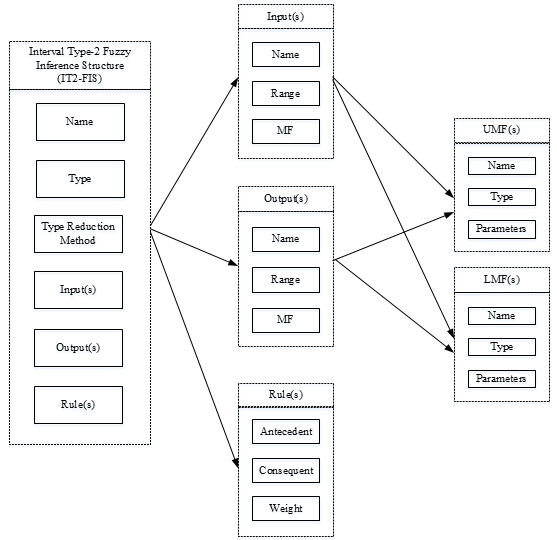
The proposed interval type-2 fuzzy logic system (IT2-FLS) toolbox is implemented by reusing the Matlab® commercial Fuzzy Logic Toolbox and adding new functions for type reduction operations, improving the toolbox user interfaces, creating a Simulink library, connecting the toolbox to Simulink.

The IT2FLS toolbox design starts by creating a structure in the Matlab workspace. This structure is named as interval type-2 fuzzy inference structure (IT2-FIS). The IT2-FIS structure includes all information of the user’s design. When new membership functions (MFs) added or current MFs modified or some rules added, actually, at each action this structure is updated by the IT2FLS toolbox automatically. An example overview of the it2fis is given in the figure 3.

## Main Editor

Main Editor page is the entry point of the IT2-FLS Toolbox. When the IT2FLS Toolbox started, main editor is opened first. The other pages can be accessed from the Main Editor. An overview of the main editor is given in the figure 1 a. There are some menus in the top of the screen in the Main Editor. The IT2-FLS toolbox saves the designs to a file with extension ‘it2fis’ or to Matlab workspace as a structure. Saving and loading operations are available under the file tab. In addition, it is possible to open a new toolbox from the file menu.

Fig.3 The interval type-2 structure of the toolbox



Adding inputs and outputs is possible from the edit menu. Default is one input and one output. However, users can add more inputs and outputs as desired by using the edit menu in the main editor.

Another opportunity of IT2-FLS toolbox is to export the current design to Matlab/Simulink automatically. There is button in the main editor for exporting. This topic is explained in detail in the next section.

Type reduction (TR) is one of the important operation for the IT2-FLS. The IT2-FLS toolbox includes the codes of the following type reduction methods and users are able to choose one of these TR methods as desired.

• Karnik-Mendel Algorithm (KM)

• Enhanced Karnik-Mendel Algorithm (EKM)

• Iterative Algorithm with Stop Condition (IASC)

• Enhanced Iterative Algorithm with Stop Condition (EIASC)

• Enhanced Opposite Direction Searching Algorithm (EODS)

• Wu-Mendel Uncertainty Bound Method (WM)

• Nie-Tan Method (NT)

• Begain-Melek-Mendel Method (BMM)

• Custom (User defined function for type reduction)

The menu to choose the desired TR method is given in the figure 4.

Fig. 4.Type Reduction Methods for choosing

In addition, it is possible to use a custom function for Type Reduction for the users who would like to develop their own type reduction function. To do this, ‘custom’ must be selected from the pop-up menu then the desired function can be selected for type reduction.

## Membership Function Editor

Membership functions (MFs) of the IT2-FLS design can be edited from the membership function editor page. An overview of the membership function editor page is given in the figure 1 b.

In this page, basically, range, types and values of the upper and lower membership functions are designed. Membership functions types are original functions of Matlab Fuzzy Logic Toolbox for UMFs and LMFs separately. The only difference, there is an additional parameters for each type of membership functions to define the height. For instance, triangle membership function is defined with four parameters as follows:

y = trimf(l,c,r,h) (9)

First parameter defines the left point (*l*), second point defines the center point (*c*) and third parameter defines the right point (*r*) of the triangular membership function. The last parameter (*h*) defines the height of the triangle membership function.

Another advantage of the IT2-FLS toolbox is possibility of designing upper and lower membership functions in either same type or different type. For instance, LMF can be triangle while UMF is Gaussian. The only prerequisite is that all the membership degree values of the lower membership function have to be smaller than or equal to the upper membership function’s ones. Otherwise, the toolbox would give a warning message.

Designing output membership functions by using the membership function editor page has some opportunities. It is possible to choose type of the output membership functions either constant or linear. In addition, it is possible to define parameter values either as a crisp or as an interval.

## Rule Editor

Rule editor page allows users to defining the rules of the type-2 fuzzy logic design. Rule editor is implemented by reusing the Matlab® commercial Fuzzy Logic Toolbox. An overview of rule editor page is in the figure 1 d.

As seen in the figure, input and output variables are available in the rule editor. Here, to add a new rule, desired input and output variables are chosen by clicking and then the new rule is added by clicking the add rule button. Then added new rule would be appear in the top. All desired rules added for the IT2-FLS design from the rule editor page. All added new rule information are written to the IT2-FIS structure at the Matlab workspace automatically, and after the rule editor page closed, these information are hold by this structure.

## Surface Viewer

After completing the IT2-FLS design, it is possible to see the surface of the output at each point. To see the surface, surface button from the view menu in the main editor is clicked. It is possible to edit or save the surface opened as desired. An example surface is given in the figure 1 b.

# Working With Matlab/Simulink

The IT2-FLS toolbox is designed to be working with Matlab/Simulink. The one of the main features of the IT2-FLS toolbox is the automatic connection to Simulink. It is possible to use this feature by just clicking the button ‘Export to Simulink’ in the Main Menu. Then, current design would be exported to Simulink automatically. The toolbox uses its own Simulink library blocks that is explained in the next section to work on the Simulink environment.

## Simulink Library of the IT2-FLS Toolbox

A Simulink library for the IT2-FLS toolbox is created first to ensure the connection between toolbox and Simulink. The IT2-FLSs toolbox library consists of 2 blocks. One of them can be used to embed the design to the block and the other also provides opportunity to choose the desired type reduction method.

The first block is the default library block of the IT2-FLSs toolbox. The designed IT2-FIS is defined to this block to allow make simulations. The second library block has an additional opportunity to choose a type reduction method. Desired type reduction method can be chosen easily by using the second block of the IT2-FLS toolbox Simulink library. An overview of the second block is given in the figure 6.

Fig. 5. Simulink Library of the IT2-FLS Toolbox

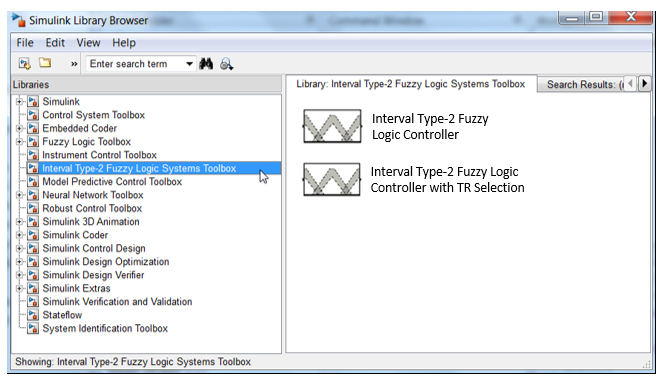
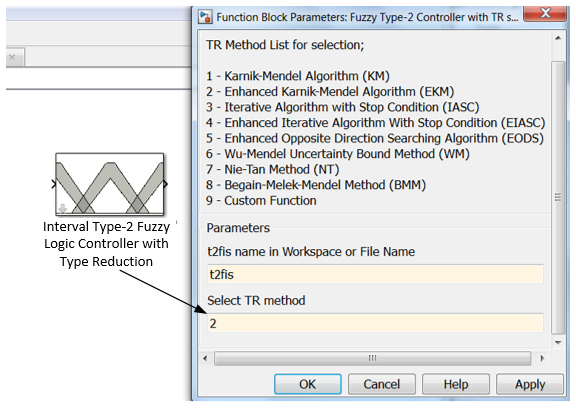


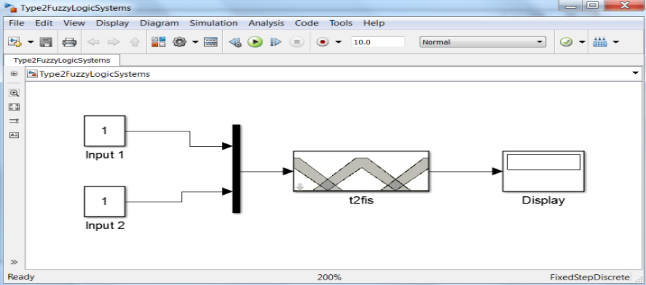
Fig. 6. IT2-FLS Toolbox Library Block with TR Selection



## Exporting the IT2-FLS Design to Simulink

After completing the IT2-FLS design, it is possible to export it to Simulink automatically from the IT2-FLS toolbox. To do this, there is a button named ‘Export to Simulink’ in the main editor. After finishing the IT2-FLS design, it is possible to export it to Simulink automatically by clicking this button. Then, a Simulink model would be created and opened automatically by the IT2-FLS toolbox. An example Simulink model created by the toolbox is given in the figure 7.

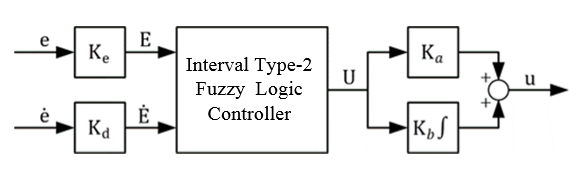
Fig. 7.An example Simulink Model that is created by the IT2-FLS Toolbox



# Illustratıve Examples

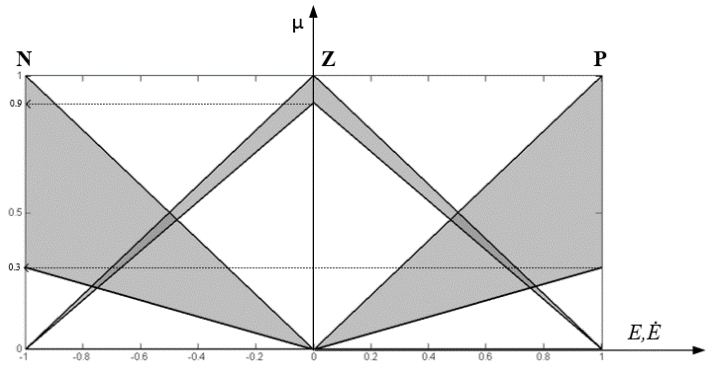
In this section, several simulations have been done to examine and compare the control performance of different type reduction by using the toolbox. For this purpose, firstly, an interval type-2 fuzzy logic controller (IT2-FLC) is created by using the IT2-FLS toolbox. Then, simulations are done to compare the control performance of the different type reduction methods for the same closed loop control system. The IT2-FLC is designed with two input variables and one output variable. The block diagram of the interval type-2 fuzzy logic controller is given in figure 8.

Fig. 8. IT2-FPID controller



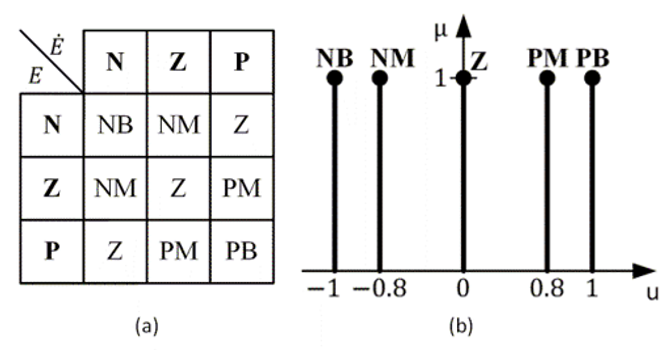
The first input of the IT2-FLC is the error signal the second input of the IT2-FLC is the derivative of the error signal. The MFs of the input variables are chosen same and as triangle. The height of the LMFs are found by using the genetic algorithm. The designed MFs for the IT2-FPID are given in the Figure 9.

Fig. 9. : The Inputs MFs used for the IT2-FPID controller



The output MFs and the rule base of the IT2-FPID is given in the figure 10.

Fig. 10. : (a) The fuzzy rule base, (b) the consequent MFs used for the IT2-FPID controllers.



The process to be controlled is chosen as a first order plus dead time (FOPDT) system. The wide use of FOPDT is due both to its simplicity as well as its ability to capture the essential dynamics of several industrial processes. The transfer function of the FOPDT is as follows:

|  |  |  |
| --- | --- | --- |
|  |  | **(6.1)** |

where *K* is the process gain, *T* is the time constant and finally *L* is the time delay. In the simulations, four different type reduction methods are used. In order to compare the performance of transient responses of the different type reduction methods, overshoot, settling time and the integral time absolute error (ITAE) which is defined in the following equation are considered

*ITAE =*

where *y* represents the system output and *r* represents the reference value.







As the first step, the nominal process parameters are set to K=1, T=1 and L=0.2. The sampling time of the simulation is set to 0.05 s. Since the unit step changes are studied for the simulations is set to 1, then the other scaling factors are determined as , , after optimization. The step responses of the closed loop control system with different type reduction methods are illustrated in Figure 11.

Fig. 11**.** The step responses of the closed loop system for various TR methods for the nominal process.

In order to examine the performance of the different type reduction methods, the system parameters are perturbed. The step responses of the closed loop control system with different type reduction methods for the first perturbed process, which has the parameters as K=1.3, T=1.9, L=0.4, is given in Figure 12 and for the second perturbed process, which has the parameters as K=1.1, T=1.3, L=0.45, is given in Figure 13.

Fig 12.The step responses of the closed loop system for various TR methods for the perturbed process (K=1.3, T=1.9, L=0.4).

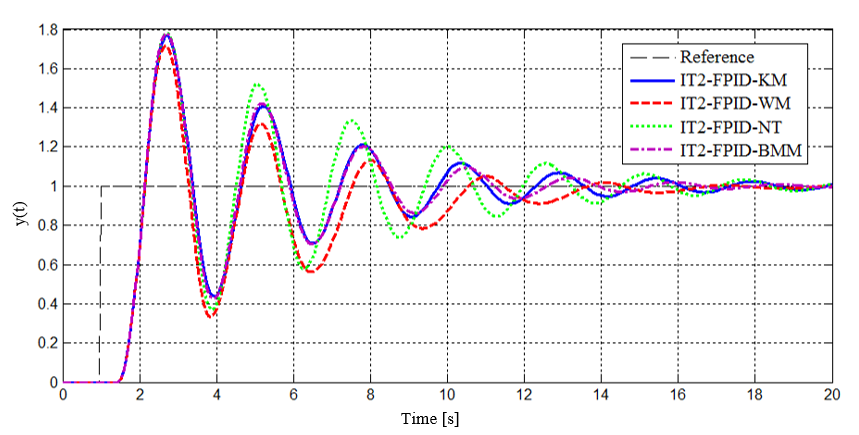


Fig 12.The step responses of the closed loop system for various TR methods for the perturbed process (K=1.1, T=1.3, L=0.45).

The performance results of the closed loop control system with different type reduction methods are tabulated in Table 1. The settling time, overshoot, IAE and ITAE values for each processes are given in this table.

Table 1.Simulation comparison using different performance measures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process | TR | Ts | OS(%) | ITAE |
| Process-1 | KM | 1.15 | 2.6 | 13.938 |
| WM | 1.15 | 0.0 | 14.059 |
| NT | 1.15 | 2.9 | 13.840 |
| BMM | 1.15 | 4.3 | 14.035 |
| Process-2 | KM | 8.05 | 61.3 | 44.737 |
| WM | 10.5 | 57.0 | 51.875 |
| NT | 7.95 | 61.4 | 45.578 |
| BMM | 8.20 | 61.7 | 44.379 |
| Process-3 | KM | 13.30 | 76.0 | 63.358 |
| WM | 12.10 | 71.4 | 65.594 |
| NT | 14.45 | 76.3 | 74.995 |
| BMM | 11.15 | 79.2 | 61.389 |



The other comparison has been made to compare the computational time of the different type reduction methods. For this purposes, the each function that has been created for the each type reduction method is run 500 times and the maximum, minimum and average values have been measured and given in the Table 2.

Table 2.Comparison of computational time of the different TR methods

|  |  |  |  |
| --- | --- | --- | --- |
| TR Method | MEAN (s) | MIN (s) | MAX (s) |
| KM | 0.0058 | 0.0052 | 0.0116 |
| EKM | 0.0056 | 0.0053 | 0.0099 |
| IASC | 0.0055 | 0.0052 | 0.0101 |
| EIASC | 0.0055 | 0.0052 | 0.0113 |
| EODS | 0.0053 | 0.0051 | 0.0122 |
| WM | 0.0048 | 0.0045 | 0.0092 |
| NT | 0.0046 | 0.0044 | 0.0087 |
| BMM | 0.0049 | 0.0046 | 0.0096 |

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.