Automated Guided Vehicle Simulation Software Development using Parallel Cascade Fuzzy Method for Reaching a Target

Benediktus Wijayanto
Department of Informatics, Faculty of Science and Mathematics
Diponegoro University
Semarang, Indonesia
benedict.and.w@gmail.com

Adi Wibowo
Department of Informatics, Faculty of Science and Mathematics
Diponegoro University
Semarang, Indonesia
bowo.adi@gmail.com

Abstract – Environment change is considered problem in a development process of Auto Guided Vehicle Robots. Parallel cascade fuzzy is a specialized method that developed for volatile operating environment. In the previous study, Parallel cascade fuzzy haven't implemented for reach a goal area. In this study, parallel cascade fuzzy being used by Auto Guided Vehicle robot to avoiding obstacle and reaching goal area. Another fuzzy algorithm involved in the test stage to compare the performance with Parallel Cascade Fuzzy. In the development of robot movement control system, simulation software that able to reveal a working pattern of such method can be used to facilitate the testing process.

Keywords— robot, Auto Guided Vehicle, simulation, fuzzy, parallel cascade fuzzy

I. INTRODUCTION

Robotics development have a lot improvement since it was founded, one of its kind is Autonomous Guided Vehicle (AGV). Since it was founded, AGV had been made for industrial use purpose [1][2][3]. As the time passed, AGV application had penetrated into many fields. Now, AGV devices and transportation units that can be operated by end-user are already exist.

The successful application of fuzzy logic in complex non-linear control systems with complex mathematical models has inspiring researchers to apply fuzzy logic to artificial intelligence in robots, such as navigation, avoiding obstacles and finding goals. The problem encountered in the application of robots in the real world is the input uncertainty caused by environmental changes [4].

One of many methods that used for controlling AGV movement was fuzzy logic system. Fuzzy has applicated into many kinds of usage since it developed into many forms that use for special purpose. Some fuzzy method that implemented in AGV development such as multi-mode control method based on fuzzy selector [5] and fuzzy logic robust controller based on Sugeno fuzzy model [6]. A method called Parallel Cascade Fuzzy Inference System, published in article on the International Conference on Advanced Computer Science and Information Systems (ICACSIS) proceedings entitled "Parallel Cascade Fuzzy Inference System at The Environment Changes, Case Study: Automated Guide Vehicle Robot Using Ultrasonic Sensor" was made as fuzzy improvement for AGV robot control system to avoiding obstacles [7]. AForge.net fuzzy library was used and Fuzzy Auto Guided Sample software (made by Fabio L. Caversan) [8] was developed into Fuzzy Auto Guided Vehicle 1.2 to test parallel cascade fuzzy as AGV robot movement control system.

Based on that background, the main goal of this study is to know the performance of parallel cascade fuzzy in its application on the robot device to reach a target through simulation software by comparing it with Serial Hierarchical Fuzzy System (SHFS) and Multiple Input Multiple Output (MIMO) Fuzzy.

II. METHODOLOGY

This research did by comparing parallel cascade fuzzy, Serial Hierarchical Fuzzy System (SHFS) and Multiple Input Multiple Output (MIMO) Fuzzy using the Fuzzy Auto Guided Vehicle 1.3 simulation software, developed from the predecessor software, Fuzzy Auto Guided Sample [8] and Fuzzy Auto Guided Vehicle 1.2 [7]. The fuzzy logic system was built using fuzzy library from Aforge.net. This research using fuzzy Mamdani with centroid defuzzification [9] as a base for each compared fuzzy methods.

A. Fuzzy with MIMO models.

This method is based on standard fuzzy logic system with multiple inputs and multiple output [10]. Fuzzy with MIMO models is a fuzzy logic system that contain two or more different input value (X_N) used in the fuzzy system process. The rule base is containing two or more different kind of rule base that produce two or more different output value (Y_N). Based Fig. 1 shows the module of fuzzy with MIMO models. Fuzzy system containing fuzzification, followed by fuzzy inferencing, fuzzy rule bases, and the value reinterpreted with defuzzification process.

MIMO architecture was used in some case to reach the efficiency and the robustness of the system.[11][12][13]

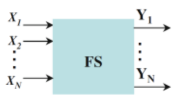


Fig. 1. MIMO Module [10]

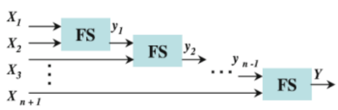


Fig. 2. Serial Hierarchical Fuzzy System. [10]

B. SHFS

One of problems in fuzzy is how to handle the many rules used in computing. Fuzzy rules number grows along with inputs number of a fuzzy system. For example, a fuzzy logic

978-1-5386-7440-6/18/$31.00 ©2018 IEEE

system that have n input variables m membership variables need m^n fuzzy rules. [9]

Hierarchical fuzzy system built to overcome a lot number of fuzzy rules that needed by a fuzzy logic system with many input variables. One type of hierarchical fuzzy system models used in this research is serial hierarchical fuzzy system. This system has fuzzy system unit that giving outputs stated as y_n, from inputs stated as X_{n-1} and y_{n-1} from fuzzy system unit with inputs stated as X_{n-1} and X_n. The architecture of serial hierarchical fuzzy systems is shown in Fig. 2. Serial fuzzy in "Modeling of Hierarchical Fuzzy Systems" journal state that $(n + 1)^{th}$ fuzzy logic unit has input from the first fuzzy logic unit and the external input value (input value that obtained from outside the fuzzy system). However, the serial hierarchical fuzzy system that used in this research only uses one input for the second fuzzy logic unit.

C. *Parallel Cascade Fuzzy*

Parallel cascade fuzzy built to handle dynamic condition. Dynamic condition interpreted as a condition of a system that handle inputs from fuzzy system can be change, depend on its operation environment. Parallel cascade fuzzy was claimed as a suitable and robust system to handle a dynamic condition [6].

Fig. 3 shows the architecture of parallel cascade fuzzy. This system has input and processed through control system without change the input values. Inputs are processed on the control system according to the operating environment conditions. Inputs is used by one of control systems in fuzzification process. Output from fuzzification will translated based on rule base in fuzzy inferencing process. After that, output from defuzzification process can be reprocessed as input and fuzzy set modifier for the next fuzzy module.

III. Development Specification

In this research, Fuzzy Auto Guided Vehicle 1.3 Simulation Software is modified into three kinds of software to compare performance of each movement control system mentioned before. First one is using the Fuzzy with MIMO model, the second is using SHFS, and the last method used is Parallel Cascade Fuzzy method. Following is how each method implemented as a control system of Auto Guided Vehicle to reach a target. Every notation used in fuzzy rule base specification explained in Table 1.

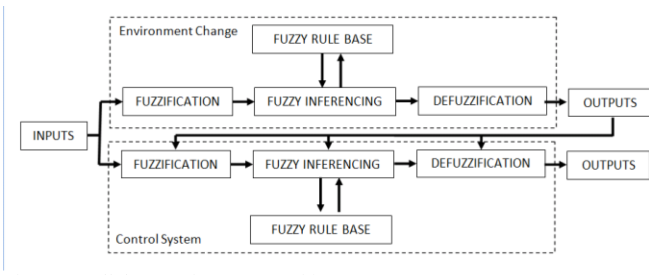


Fig. 3. Parallel Cascade Fuzzy Architecture.[7]

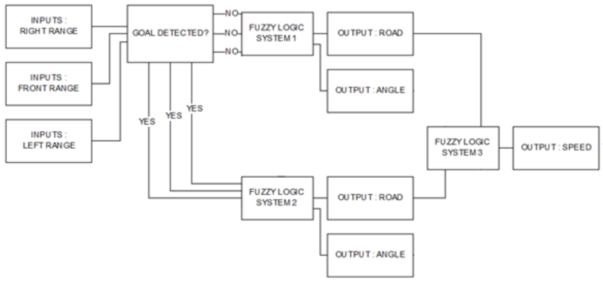


Fig. 4. Simulated SHFS Architecture.

TABLE I. Fuzzy Rule Base Notation Abbreviation Explanation

FD =	FrontalDistance	F =	Far
RD =	RightDistance	M =	Medium
LD =	LeftDistance	N =	Near
A =	Angle	VN =	VeryNegative
		NT =	Negative
		LN =	LittleNegative
		Z =	Zero
		LP =	LittlePositive
		PT =	Positive
		VP =	VeryPositive
S =	Speed	FS =	FastSpeed
		MS =	MediumSpeed
		LS =	LowSpeed
		SS =	Stopingspeed
R =	Road	FG =	FarGoal
		MG =	MediumGoal
		NG =	NearGoal
		GR =	GangRoad
		VR =	VillageRoad
		CR =	CityRoad
		TR =	TollRoad
RT =	RoadT	G =	Gang
		V =	Village
		C =	City
		T =	Toll

A. *Fuzzy with MIMO Model Specification.*

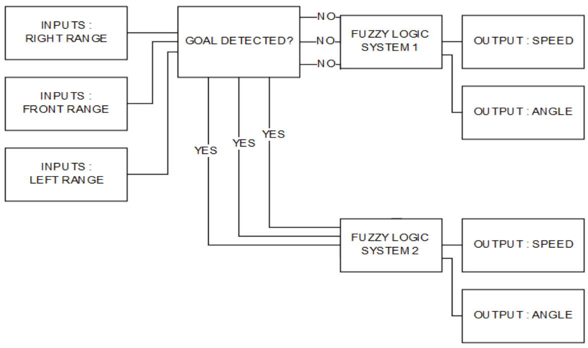


Fig. 5. Simulated Fuzzy System with MIMO Model Architecture

Fuzzy with MIMO implemented with basic fuzzy architecture that use two fuzzy logic systems. Each of them runs in separate state. First logic system built to handle a condition when goal area is undetected, the others activated when goal area is detected. Every fuzzy logic system produces two outputs, speed and angle. Inputs from this system are obtained from three proximity sensors. The architecture of this methods is shown in Fig. 4.

The written specification in this paper is the fuzzy logic system that handle a target reaching control system specification only. These specifications are:

1) Input variables: ‘LeftDistance’, ‘RightDistance’, dan ‘FrontalDistance’ with membership function value NEAR[0 15 50], MEDIUM[15 50 60 100], and FAR[60 100 120].

2) Output variables: ‘Speed’ with membership function value FASTSPEED[-10 -7.5 -6.5], MEDIUMSPEED[-7.5 -6.5 -4.5 -3.5], LOWSPEED[-4.5 -3.5 -1.5 -0.5], STOPINGSPEED[-1.5 -0.5 0] and ‘Angle’ with membership function value VERYNEGATIVE[-50 -40 -35], NEGATIVE[-40 -35 -25 -20], LITTLENEGATIVE[-25 -20 -10 -5], ZERO[-10 -5 5 10], LITTLEPOSITIVE[5 10 20 25], POSITIVE[20 25 35 40], VERYPOSITIVE[35 40 50].

3) Fuzzy rule base: In fuzzy logic system 2 consist of fourteen rule bases, i.e.:

[R1]	FD = F ⇒ A = Z	[R8]	FD = F ⇒ S = FS
[R2]	FD = M ⇒ A = Z	[R9]	FD = M ⇒ S = MS
[R3]	RD = F ⇒ A = PT	[R10]	RD = F ⇒ S = MS
[R4]	LD = F ⇒ A = NT	[R11]	LD = F ⇒ S = MS
[R5]	RD = M ⇒ A = VP	[R12]	RD = M ⇒ S = LS
[R6]	LD = M ⇒ A = VN	[R13]	LD = M ⇒ S = LS
[R7]	FD = N ⇒ A = Z	[R14]	FD = N ⇒ S = SS

B. *SHFS Specification*

In this research, SHFS implemented as a system that have three fuzzy logic systems. The first fuzzy logic system active when goal area undetected by sensor. This fuzzy logic has turn angle and road type outputs. Second fuzzy logic has same inputs and outputs with the first fuzzy logic, but it active when sensor detect a goal area. The third fuzzy generate speed output that obtained from turn angle and road type inputs. Architecture of the implemented SHFS shown in Fig. 5.

This system was built with these specifications:

1) Input variables: ‘LeftDistance’, ‘RightDistance’, ‘FrontalDistance’ (fuzzy logic system 2) with membership function value NEAR[0 5 20], MEDIUM[15 25 35 45], FAR[40 50 120] and ‘RoadT’ (fuzzy logic system 3) with fuzzy membership value GANG[0 0.5 0.75], VILLAGE[0.5 0.75 1.25 1.75], CITY[1.25 1.75 2.25 2.75], TOLL[2.25 2.75 3].

2) Output variables: ‘Angle’ (fuzzy logic system 2) with membership function value VERYNEGATIVE[-50 -40 -35], NEGATIVE[-40 -30 -20], LITTLENEGATIVE[-25 -15 -5], ZERO[-10 0 10], LITTLEPOSITIVE[5 15 25], POSITIVE[20 30 40], VERYPOSITIVE[35 40 50]. ‘Road’(fuzzy logic system 2)with membership function value NEARGOAL[0.5 0.55 0.75], MEDIUMGOAL[0.5 0.75 .75 2], FARGOAL[1.75 2 3]. ‘Speed’ (fuzzy logic system 3) FASTSPEED[-10 -7.5 -6.5], MEDIUMSPEED[-7.5 -6.5 -4.5 -3.5], LOWSPEED[-4.5 -3.5 -1.5 -0.5], STOPINGSPEED[-1.5 -0.5 0].

3) Fuzzy rule base: In fuzzy logic system 2 consist of eighteen rule bases, i.e.:

[R1]	FD = F ⇒ R = FG	[R10]	FD = F ⇒ A = Z
[R2]	FD = M ⇒ R = MG	[R11]	FD = M ⇒ A = Z
[R3]	FD = N ⇒ R = NG	[R12]	FD = N ⇒ A = Z
[R4]	RD = F ⇒ R = FG	[R13]	RD = F ⇒ A = NT
[R5]	RD = M ⇒ R = MG	[R14]	LD = F ⇒ A = PT
[R6]	RD = N ⇒ R = NG	[R15]	RD = M ⇒ A = NT
[R7]	LD = F ⇒ R = FG	[R16]	LD = M ⇒ A = PT
[R8]	LD = M ⇒ R = MG	[R17]	RD = N ⇒ A = VN
[R9]	LD = N ⇒ R = NG	[R18]	LD = N ⇒ A = VP

In fuzzy logic system 3 consist of four rule bases, i.e.:

[R1] RT = T ⇒ S = FS
[R2] RT = C ⇒ S = MS
[R3] RT = V ⇒ S = LS
[R4] RT = G ⇒ S = SS

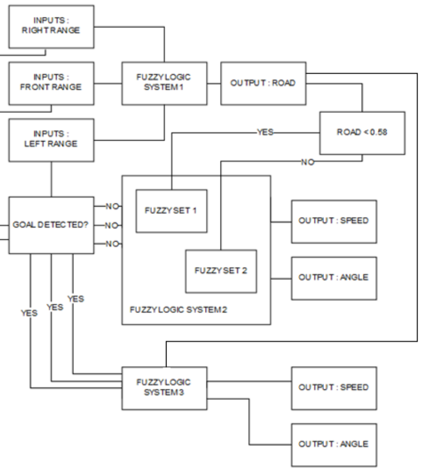


Fig. 6. Simulated Parallel Cascade Fuzzy Architecture.

C. *Parallel Cascade Fuzzy Specification*

Parallel cascade fuzzy method implemented to process inputs from proximity sensors that detect range from robot to obstacle in front of it. Those inputs processed into outputs i.e. speed and turn angle. Fuzzy logic system 1 has output road type that used by the fuzzy logic system 2 as a fuzzy set modifier. Fuzzy set modifier is a variable that changes the domain of fuzzy set. Road type value used to determine which fuzzy set will be use in of fuzzy logic system 2. If the road type value is greater than or equal to 0.58, then fuzzy set 1 used, otherwise if less than 0.58, fuzzy set 2 used. Fuzzy logic system 3 added to processing speed and turn angle outputs when goal area detected. This system is affected by road type value as a fuzzy set modifier too. Architecture of the implemented parallel cascade fuzzy is shown in Fig. 6.

This system was built with these specifications:

1) Input variables: ‘LeftDistance’, ‘RightDistance’, ‘FrontalDistance’ (in all fuzzy logic systems) with membership function value NEAR[0 5 20], MEDIUM[15 25 35 45], FAR[40 50 120].

The fuzzy logic system 3 has unfixed fuzzy membership function due to its fuzzy set modifier (*i* variable). The membership function of ‘NEAR’ linguistic variable shown by (1), ‘MEDIUM’ shown by (2), and ‘FAR’ shown by (3).

$$\mu NEAR[r] = \begin{cases} 1; r < 5i \\ \frac{10i-r}{10i-5i}; 5i \le r \le 10i \\ 0; r > 10i \end{cases} \quad (1)$$

$$\mu MEDIUM[r] = \begin{cases} 1; 10i < r < 60i \\ \dfrac{r - 5i}{10i - 5i}; 5i \le r \le 10i \\ \dfrac{80i - r}{80i - 60i}; 60i \le r \le 80i \\ 0; r > 5i \vee r < 80i \end{cases} \quad (2)$$

$$\mu FAR[r] = \begin{cases} 1; 80i < r < 120i \\ \frac{r-60i}{80i-60i}; 60i \le r \le 80i \\ 0; r < 60i \end{cases} \quad (3)$$

1) Output variables: 'Road' (fuzzy logic system 1) with membership function value GANGROAD[0.5 0.55 0.6 0.65], VILLAGEROAD[0.6 0.7 0.8 0.9], CITYROAD[0.8 1 1.2 1.5], TOLLROAD[1.4 1.8 2.99 3]. 'Angle' (fuzzy logic system 3) with membership function value VERYNEGATIVE[-50 -40 -35], NEGATIVE[-40 -30 -20], LITTLENEGATIVE[-25 -15 -5], ZERO[-10 0 10], LITTLEPOSITIVE[5 15 25], POSITIVE[20 30 40], VERYPOSITIVE[35 40 50]. The 'Speed' (fuzzy logic system 3) output has unfixed fuzzy membership function due to its fuzzy set modifier. The membership function of 'STOPSPEED' linguistic variable shown by (4), 'LOWSPEED' shown by (5), 'MEDIUMSPEED' shown by (6), and 'FASTSPEED' shown by (7).

$$\mu STOPSPEED[p] = \begin{cases} 1; p > 0 \\ \dfrac{p - (-0{,}1i)}{0 - (-0{,}1)}; -0.1i \le p \le 0 \\ 0;\ p < -0.1i \end{cases} \quad (4)$$

$$\mu LOWSPEED[p] = \begin{cases} 1; -1i < p < -0.5i \\ \dfrac{p - (-1.5i)}{-1i - (-1.5i)}; -1.5i \le p \le -1i \\ \dfrac{-0.1i - p}{-0.1i - (-0.5i)}; -0.5i \le p \le -0.1i \\ 0; p > -0.1i \vee o < -1.5i \end{cases} \quad (5)$$

$$\mu MEDIUMSPEED[p] = \begin{cases} 1; -3.5i < p < -2i \\ \dfrac{p - (-4i)}{-3.5i - (-4i)}; -4i \le p \le - - 3.5i \\ \dfrac{-1i - p}{-1i - (-2i)}; -2i \le p \le -1i \\ 0; p > -4i \bigvee p < -1i \end{cases} \quad (6)$$

$$FASTSPEED[p] = \begin{cases} 1; p < -4i \\ \frac{-3.5i-p}{-3.5i-(-4i)}; -3.5i \le p \le -4i \\ 0; p > -3.5i \end{cases} \quad (7)$$

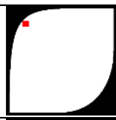
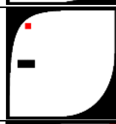
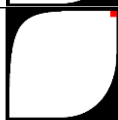
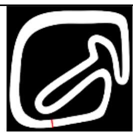
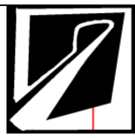
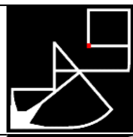
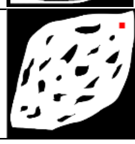
2) Fuzzy rule base: In fuzzy logic system 1 consist of 27 rule bases, i.e.:

[R1] RD = N ∧ LD = N ∧ FD = F ⇒ R = GR
[R2] RD = N ∧ LD = N ∧ FD = M ⇒ R = GR
[R3] RD = N ∧ LD = N ∧ FD = N ⇒ R = GR
[R4] RD = N ∧ LD = M ∧ FD = F ⇒ R = VR
[R5] RD = N ∧ LD = M ∧ FD = M ⇒ R = VR
[R6] RD = N ∧ LD = M ∧ FD = N ⇒ R = GR
[R7] RD = N ∧ LD = F ∧ FD = F ⇒ R = CR
[R8] RD = N ∧ LD = F ∧ FD = M ⇒ R = CR
[R9] RD = N ∧ LD = F ∧ FD = N ⇒ R = GR
[R10] RD = M ∧ LD = N ∧ FD = F ⇒ R = VR
[R11] RD = M ∧ LD = N ∧ FD = M ⇒ R = VR
[R12] RD = M ∧ LD = N ∧ FD = N ⇒ R = GR
[R13] RD = M ∧ LD = M ∧ FD = F ⇒ R = CR
[R14] RD = M ∧ LD = M ∧ FD = M ⇒ R = CR
[R15] RD = M ∧ LD = M ∧ FD = N ⇒ R = GR
[R16] RD = M ∧ LD = F ∧ FD = F ⇒ R = TR
[R17] RD = M ∧ LD = F ∧ FD = M ⇒ R = TR
[R18] RD = M ∧ LD = F ∧ FD = N ⇒ R = GR
[R19] RD = F ∧ LD = N ∧ FD = F ⇒ R = CR
[R20] RD = F ∧ LD = N ∧ FD = M ⇒ R = CR
[R21] RD = F ∧ LD = N ∧ FD = N ⇒ R = GR
[R22] RD = F ∧ LD = M ∧ FD = F ⇒ R = TR
[R23] RD = F ∧ LD = M ∧ FD = M ⇒ R = VR
[R24] RD = F ∧ LD = M ∧ FD = N ⇒ R = GR
[R25] RD = F ∧ LD = F ∧ FD = F ⇒ R = TR
[R26] RD = F ∧ LD = F ∧ FD = M ⇒ R = VR
[R27] RD = F ∧ LD = F ∧ FD = N ⇒ R = GR

In fuzzy logic system 1 consist of fourteen rule bases, i.e.:

[R1]	FD = F ⇒ A = Z	[R8]	LD = F ⇒ S = FS
[R2]	FD = F ⇒ S = FS	[R9]	RD = M ⇒ A = LP
[R3]	FD = M ⇒ A = Z	[R10]	RD = M ⇒ S = LS
[R4]	FD = M ⇒ S = FS	[R11]	LD = M ⇒ A = LN
[R5]	RD = F ⇒ A = P	[R12]	LD = M ⇒ S = LS
[R6]	RD = F ⇒ S = FS	[R13]	FD = N ⇒ A = Z
[R7]	LD = F ⇒ A = NT	[R14]	FD = N ⇒ S = SS

TABLE II. TEST AREAS

No.	Area Name	Area Image	Information
1.	goalA		An area to test the robot's ability to reach a goal positioned in front of the robot.
2.	goalB		An area to test the robot's ability to reach a goal positioned in front of the robot with one obstacle.
3.	goalC		An area to test the robot's ability to reach a goal positioned on the front right position of the robot.
4.	circA		An area to test the ability of the robot controls the moving direction and the speed on a circuit with a homogeneous trajectory width.
5.	circB		An area to test the ability of the robot to control the direction of motion and adjust the speed at the circuit with a not significance changes of the trajectory width.
6.	circC		An area to test the ability of the robot to control the direction of motion and adjust the speed on the circuit with a significant change of the width of the trajectory.
7.	laneA		An area to test the ability of the robot to operate in narrow trajectory.
8.	obsA		An area to test robot's ability to operate in area with many obstacles and various turn angle.

IV. TEST RESULT

The test was done by comparing of each method implemented as AGV robot moving control system in areas listed in Table 2. The compared parameters are the elapsed time of the AGV robot completing each area (from the starting point to goal area) and the success rate. If the robot collides with an obstacle, then the test is failed and time not recorded in result table. Each architecture tested ten times on each operational area with interval time input 10 ms.

Based on test result of ‘goalA’ shown in Table 3, parallel cascade fuzzy have the best average time to reach a target. A relatively wide area causes the fuzzy set modifier to have a large value. This impacts on the highest speed that can be achieved by robots that use parallel cascade fuzzy is high enough (recorded 10 pixels / iteration). In the SHFS method, the highest speed that a robot can achieve is only 6 pixels / iteration. In the MIMO fuzzy method, the highest speed the robot can reach is 8 pixels / iteration.

Based on the test on the 'goalB' area, the parallel cascade fuzzy method has the shortest average travel time.

Based on test results obtained in the 'goalA' and 'goalB' areas, re-testing for the same area with the destination area moved to the right side of the robot operational area.

Test results in the 'goalC' area show that the fuzzy method has the shortest average travel time, with little gap compared to parallel cascade fuzzy method. In this test proves that the orientation of the right and left of the destination location against the initial position of the robot affect the travel time. The trend factor of robot turn direction needs to be adjusted again to get more valid results. However, based on the average time obtained by parallel cascade fuzzy compared with the fuzzy method, the difference that is not too significant can be a benchmark. When the destination area is in the opposite direction of the robot turning tendency, the time to reach that goal still shows good results.

In the next test, each method tested in circuit operating area. The area 'circA' has a characteristic trajectory with a homogeneous width.

In comparison to the circA area shown in Table 3 the fastest route time result is a fuzzy method with a time difference of 2.055 seconds with parallel cascade fuzzy, while the SHFS method is far adrift with time difference 19.177. Further testing is done on the area 'circB' with a change in the width of the trajectory that is not too significant.

In the test of the 'circB' area, the fuzzy method has the best average time record of 16.751 seconds. Parallel cascade fuzzy is under MIMO fuzzy method with average time of 22.394 seconds followed by SHFS method with average travel time 42.323. Furthermore, to reinforce the performance results of each method on the condition of the width of the passage, test was done in the area of 'circC' which has a very significant change in trajectory width.

Based on the results of the test on the 'circC' area, in case of handling very narrow width trajectories, the parallel cascade fuzzy method can overcome this. Other methods can’t handle trajectories less than one and a half times the width of the robot body and collide with the obstacle area, so the results of testing the SHFS and Fuzzy methods with the MIMO model are considered failed. Testing the travel time is only done five times because the results of travel time can’t be compared on testing in the area 'circC'.

Test on the 'laneA' area is a test to confirm that the parallel cascade fuzzy method can handle trajectories of width less than one and a half times the width of the robot body. 'laneA' was made with a 12 pixels line width.

The result obtained in the test of the 'obsA' area show that the parallel cascade fuzzy method reached the destination area with an average travel time of 3.195 seconds. This shows that the robot with the parallel cascade fuzzy motion method can reach a destination area well. The parallel cascade fuzzy method can handle more diverse turn angles so the robot does not have to turning around to find the position with the right angle to take the turn.

In this case, the parallel fuzzy cascade method has the advantage of including the fuzzy set modifier in speed and turn angle outputs processing, so that the robot has relativity on a given type of road. The membership range for each distance set variable will change its membership value based on the fuzzy set modifier value obtained from the value in the first fuzzy logic system. This causes various the maximum speed for each type of road. The area conditions in the tests of the ‘circA’ and the ‘circB’ areas tend to be more in medium-width road conditions, which have an adverse effect on the parallel cascade fuzzy test results. However, when examined with other area test results, robots with parallel cascade fuzzy motion method have good results and can handle trajectories less than one and a half times the width of the robot body. It shows that this method can be applied in a robot motion control system that has the ability to handle various types of operational area conditions.

Similar to parallel cascade fuzzy, SHFS can also add a condition that considers the type of trajectory as an input parameter in the system. At SHFS, speed is only determined based on the value obtained from the first fuzzy logic system that produces the type of trajectory. If the trajectory is narrow, the resulting speed will have a low maximum limit, and so on for the type of road that is getting wide is proportional to the

TABLE III. TEST RESULT

Area	Parallel Cascade Fuzzy		SHFS		MIMO Fuzzy	
	Average Time (seconds)	**Result**	**Average Time (seconds)**	**Result**	**Average Time (seconds)**	**Result**
goalA	1.299	Success	2.459	Success	1.906	Success
goalB	1.677	Success	3.013	Success	8.332	Success
goalC	2.495	Success	3.356	Success	2.395	Success
circA	17.976	Success	35.098	Success	15.921	Success
circB	22.394	Success	42.323	Success	16.751	Success
circC	29.317	Success	-	Failed	-	Failed
laneA	17.567	Success	-	Failed	-	Failed
obsA	3.195	Success	15.100	Success	24.775	Success

maximum speed of the AGV robot. However, the handling of the SHFS method is not as good as parallel cascade fuzzy.

V. Conclusion

In its application as motion control robot Automated Guided Vehicle equipped with three proximity sensors and three color sensors, the parallel cascade fuzzy method, simulated in Fuzzy Auto Guided Vehicle 1.3 software, has a level of control capability to achieve targets faster than the Serial method Fuzzy Hierarchical System (SHFS) and fuzzy with Multiple Input Multiple Output (MIMO) model. Application of parallel cascade fuzzy on AGV robot operated in area with circuit type has automatic speed setting based on trajectory type without having to readjusting the domain in fuzzy set. The cascade fuzzy parallel method has a more diverse range of area types than the SHFS and fuzzy MIMO methods because it can handle a relativity parameter based on the width of the trajectory and the distance the obstacle is read by the sensor in the robot operational area. The results obtained are not included with the determination of the fuzzy rule base and the domain value in each fuzzy variable used.

The creation of the rule base and the determination of the domain value of each fuzzy variable need a deeper analysis to determine the effect on the parallel cascade fuzzy method. The application of methods on AGV robot motion control can be performed to test the validity of test results related to other factors that may affect the performance of robots such as contours of operational areas, types of robot drive motors, and others.

References

[1] Bob Trebilcock, 2010. *Let's Remember Mac Barrett, Father of the AGV.* [Online] Available at: https://www.mmh.com/article/lets_remember_mac_barrett_father_of_the_agv [Accessed April 16th 2018].

[2] G. Arun Kumar & A. Sivasubramaniam. 2017. *Automated Guided Vehicle for Physically Handicapped People – A Cost Effective Approach.* IOP Conf. Series: Materials Science and Engineering 282.

[3] Virendra Patil & Bhatwadekar S.G. 2018. *Automated Guided Vehicle System.* International Research Journal of Engineering and Technology Volume 05 Isuue 04, p. 4038-4043.

[4] Ren C. Luo, Chung T. Liao, Su L. Kuo & Kuei C. Lin, 2005. *Automatic Docking and Recharging System for Autonomous Security Robot.* Edmonton, IEEE.

[5] Zhigang Yang, Zhijing Wang, Wentao Su and Jian Zhang, 2010. *Multi-Mode Control Method Based on Fuzzy Selector in the Four Wheel Steering Control System.* Proceedings of the 8th IEEE International Con ference on Control and Automation, Xiamen, China, pp. 1221-1226.

[6] A. Shaout, M.A. Jarrah, H. Al-Araji, K. Al-Tell. 2000. *A nonlinear optimal four-wheels steering controller*. Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Lansing, Michigan, USA, pp. 1426-1429.

[7] Adi Wibowo, Sutikno, Kushartantya, Helmie Arif Wibawa, Ari Wibisono, 2012. *Parallel Cascade Fuzzy Inference System at The Environment Changes, Case Study: Automated Guide Vehicle Robot Using Ultrasonic Sensor.* Jakarta, Faculty of Computer Science Universitas Indonesia, p. 319 - 324.

[8] Fabio L. Caversan, 2009. *Fuzzy Computing: Basic Concepts.* [Online] Available at: http://www.aforgenet.com/articles/fuzzy_computing_basics/ [Acessed March 19th 2018].

[9] Arifin Saiful, 2015. *Implementasi Logika Fuzzy Mamdani untuk Mendeteksi Kerentanan Daerah Banjir Di Semarang Utara*, Semarang: Universitas Negeri Semarang.

[10] Alicia D. Benitez & Jorge Casillas, 2013. *Multi-Objective Genetic Learning of Serial Hierarchical Fuzzy Systems for Large-Scale Problems.* Soft Computing, 17(1), p. 165–194.

[11] Fatima Zahra Baghli, Larbi El bakkali, & Tassine Lakhal. 2015. *Multi-input multi-output fuzzy logic controller for complex system: Application on two-links manipulator.* Procedia Technology 19, p. 607 – 614.

[12] Paulo Salgado. 2008. *Rule generation for hierarchical collaborative fuzzy system.* Applied Mathematical Modelling 32, p. 1159-1178.

[13] S. Vijayachitra, A. Tamilarasi, & M. Pravin Kumar. 2009. *Multiple Input Multiple Output (MIMO) Process Optimization using Fuzzy GA Clustering*. International Journal of Recent Trends in Engineering Vol. 2 No. 2, p. 16-18.