*International Journal of Engineering Applied Sciences and Technology, 2021*
**Vol. 6, Issue 3, ISSN No. 2455-2143, Pages 180-189**
Published Online July 2021 in IJEAST (http://www.ijeast.com)

# HEADING CONTROL OF AUTOMATED GUIDED VEHICLES

Abhishek Bhardwaj, Kshitij Anand, Murtaza Khasamwala

**ABSTRACT - With rapid developments in industry wide automation control systems, there have been numerous proposed design and control strategies as alternatives to conventional methods. This term paper deals with the same in the context of Automated Guided Vehicles (AGV's) with a review of the current literature for proposed design improvements as well as trajectory control systems that can improve productivity and reduce conflicts compared to current methods. The results of these proposed methods are also compared in contrast to each other to obtain holistic view of the merits and demerits of the same. It is found that a fuzzy logic based PID control is a definite upgrade over conventional PID control systems and that the application of LIDAR and Camera based technologies can help enable free roaming AGV's lending to increased robustness and flexibility. For a control system consisting of a single AGV and multiple processing stations, a genetic algorithm enhanced centralized fuzzy logic control system is the most optimal path planning algorithm whereas in the cases of multiple robots, a dynamic priority allocated free roaming AGV system with semi-decentralized control is found to be much more efficient at simultaneously solving paths and reducingconflicts with other robots in real time compared to the purely centralized systems.**

Keywords: AGV control, centralized control, decentralized control, fuzzy logic, genetic algorithm.

## I. INTRODUCTION

Automated guided vehicles or AGV‟s as they are more commonly known as are by definition unmanned mobile robots-usually wheel based-used in intralogistics and manufacturing logistics, usually to transfer raw materials, semi-finished as well as final products from one point of the factory or warehouse to another. While the stationary robots such as robotic arms or CNC machines are responsible for the manufacturing and material/product handling operations, they are limited by their workspace and hence one can use mobile robots such as AGV to fill in those gaps in automation.

With the rise of e-commerce giants all around the world from the likes of Amazon to Alibaba, there has been a huge uptick in demand for optimal automated solutions to warehouse logistics problems. It has been estimated that more than 13,000 AGV systems have been installed globally (Bechtsis et al., 2017). These consist mainly of finding and moving required consumer products from stacked shelves to transport vehicles with minimum delay, a well suited job for the mobile AGV system. Simultaneously an industry wide push towards the current trend of Industry 4.0 has incited numerous research and redesign on the current automation systems including both the stationary and mobile kind.

Although AGV‟s have been around since 1955, the onboard mechatronic systems as well as the overall trajectory planning control systems are mostly still of yesteryear, not having been upgraded in line with more recent technological advancements (Fragapane et al., 2021). Hence the inefficiencies and problems that were tolerated beforehand when they were a novelty are posing major issues for current needs and hence have to be redesigned. In a conventional AGV system, there are marked paths around the factory floor that are followed by the robots when moving around to simplify control and ensure worker safety. There is a central control system that takes in data from every workstation, such as inventory, process time, and down time to assign loads and paths to individual AGV‟s based on some predefined conditions and algorithms and so the required tasks are automated.

This type of archaic implementation of control system leads to inefficiencies and conflicts that hinder further progress in effective as well as total automation as promised by Industry 4.0. This term paper will explore the latest research advancements in this field that attempts to remedy the current issues as well as improve desired parameters such as efficiency and latency.

## II. LITERATURE REVIEW

This section is further divided into two subjections namely Micro and Macro control. Micro control refers to motion control of the AGV‟s in an individual sense; that is design and control of how sensing and movement of each AGV will be carried out. Whereas the Macro control section will be focus on planning of the overall trajectory for the motion of the AGV from pickup to destination, as well as resolution of conflicts that may arise when more than one AGV‟s are in simultaneous use. The micro control section directly affects the macro control, as they flexibility of movement of the individual AGV are required to plan optimized paths, without which calculated paths may not be physically possible in practice or available AGV motion may be underutilized, leading to sub optimal solutions.

### 2.1. Micro Control

As discussed before, micro control refers to an individual look into AGV control, which involves studying how it is receiving surrounding information as well as how motion is mechanically executed. Conventionally AGV‟s are programmed to move along predefined paths marked by special tape around the factory floor, whereby using infrared sensors and line detecting algorithms, they can keep moving in that path as ordered through central controller. Motion is usually carried out by set of wheels that are controlled by DC motors connected to the central onboard Programmable Logic Controller (PLC). Along with sensors for line following, they are also equipped with proximity sensors all around the body such that when a human or another AGV comes in front of the robot, it can immediately stop to prevent a crash. The usual type of control employed in basic Proportional Integral Derivative (PID) control, which is a control loop system that takes in an error signal and produces an appropriate output signal. The gain values for conventional PID systems are preset and unalterable leading to key issues in response time and steady state errors.

Zhou et al., (2018) propose a more intelligent AGV control system that incorporates a fuzzy logic enabled variable PID control for the individual AGV"s for a more dynamic control system.

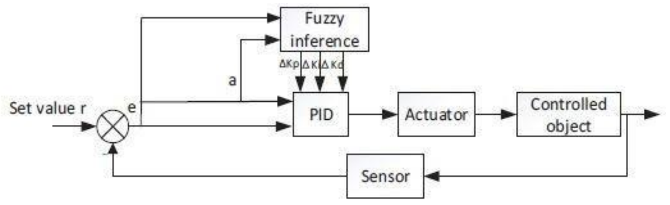

Figure 1: Fuzzy logic PID control system (Zhou et al., 2018)

Figure 1 shows an overview of the working of the proposed fuzzy PID based control system, which is similar to a conventional closed loop feedback based control system, except for the addition of a fuzzy inference. Fuzzy logic implies that instead of a set inputs and outputs, there is a predefined range of inputs that are used to obtain a more desirable output compared to standard direct approach. In this case, the inputs to the fuzzy logic system are as shown „e" and „a" which are linear and angular errors respectively. The predefined range for „e" is [-15mm, +15mm] and for „a" is [-15°, +15°]. A membership function is created based on the defined ranges with sufficient parameters such that the values of „e" and „a" are converted to corresponding $\Delta K_p$, $\Delta K_d$, and $\Delta K_i$ which are the change in proportional gain, differential gain, and integral gain respectively. This fuzzy PID Control logic is simulated in MATLAB Simulink and the results are compared to the conventional PID Control system.

With recent advances in sensor technology, along with breakthroughs in machine learning algorithms, computer vision has come a long way since the implementation of the first AGV. Hence Puppim de Oliveira et al., (2019) undertook a qualitative and quantitative analysis of the use of a USB enabled Camera for AGV control. They propose the replacement of traditional magnetic tape based line detection sensors with a RGB Camera enabled with Python implemented OpenCV for computer vision enabled line detection. The robot used for experimentally testing the proposed method uses 4 omnidirectional mecanum wheels to give a greater degree of flexibility during operation and control. The control system deployed was a simple PID feedback control loop, which takes in the positional and angular error and produces an appropriate motor speed at each wheel to correct the same. This is fed into a microcontroller chip onboard to calculate the inverse kinematics required to reproduce the PID Control output in the robot, which in turn controls the mecanum based wheel motors to move the AGV.

Machine learning based noise reduction and line detection algorithms are employed to detect the edges of the given path. By calculating the position and angular orientation of the path, the linear and angular position of the AGV can be estimated and hence can be controlled. The response and accuracy of this camera based line detection is recorded.

De Silva et al., (2018) proposed a completely free roaming version of the AGV, usually called as an Autonomous Mobile Robot (AMR) by the application of fusion of a Light Detection and Ranging sensor (LiDAR) and a wide angle camera for frontal area detection. LiDAR is a depth sensing instrument that works in a principle similar to radar. Here a beam of light is sent outwards and a timer is set to measure the total time taken for the beam to return back to the sensor. The speed of light is used to calculate the total distance between the sensor and the object that the beam of light reflected off from. This process is repeated in the entire range of the sensor and hence a virtual depth based map of the surroundings is created. In tandem with a wide angle camera that can capture images a much higher angle, a very detailed view of the surroundings of the AGV can be obtained, which can be further processed to calculate optimal paths while avoiding obstacles.

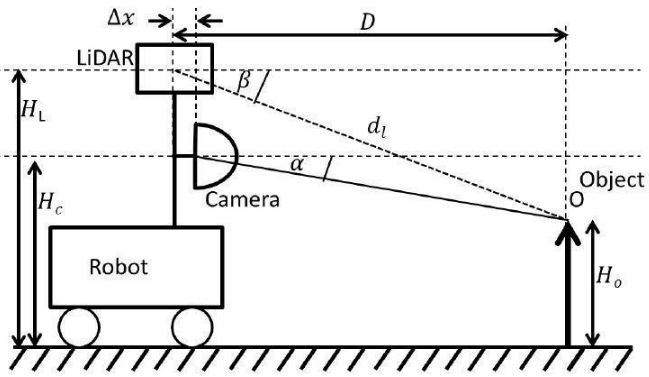

Figure 2: Schematic diagram of AGV (De Silva et al., 2018)

Figure 2 gives a schematic representation of the proposed AGV sensor configuration, with both front facing LiDAR and wide angle camera are used to detect the distance and height of the nearby object. The model is validated experimentally by creating a prototype mobile robot with the designed wide angle camera and LiDAR sensor, and after applying suitable visual detection networks, the resulting free space maps as detected by the robot are evaluated for accuracy.

## 2.2. Macro Control

Macro control refers to the overall trajectory control of one or more AGV‟s from pickup to delivery and back. Conventionally consisting of a centralized controller, this section of control takes in inputs such as immediate workstation requirements, their relative distance from starting position, lead and delay time during delivery and hence comes up with an optimum path and load for maximum delivery with minimum delays. This type of control problem is similar to the commonly known "travelling salesman" problem where there are a number of different locations that need to be visited with minimum total distance covered, with increasing number of destination points and different load conditions substantially increasing the complexity of the problem.

Klosowski et al., (2018) proposed a hybrid information system employing fuzzy logic and genetic algorithms. They approached the problem via the multimodal method of loading, which is essentially the loading of more than one transport module on a single AGV during its round trip to maximize efficiency. Genetic algorithms are heuristic optimization algorithms that mimic the biological evolution and natural selection processes to arrive upon an optimal solution to a given problem. This algorithm uses an iterative approach, wherein increasing the number of solution iterations decreases the total error from the optimal solution. The fuzzy controller takes in a 3-element input vector $\boldsymbol{W}_x = [x_1; x_2; x_3]$ consisting of the following elements,

$x_1$ – Machining Progress [%]
$x_2$ – Waiting-for-delivery [%]
$x_3$ – Risk [1, 2…10]

And the output vector, $\boldsymbol{W}_y = [y_1; y_2]$, where:

$y_1$ = [-1:1], if $y_1>0$ then delivery is needed
$y_2$ = [-1:1], if $y_2>0$ then pickup is needed

While delivery and pickup are only executed when the output values are greater than 0, the actual value above that are used to assign priority to individual delivery/pickup requests, helping the controller organize workstation requests effectively. The Risk factor is continually estimated on deviation from ideal service times at any given moment in time. The formula describing the risk is specified to indicate the average risk value ($R$=5) when during delivery, the number of units to be machined for a given transportation load at the moment of delivery is equal to the number of units in the transported load. When the delivery is carried out at an increased buffer stock, that is when the transportation load > number of units needing transport, the risk decreases. Conversely, when the size of the container load is lower than the total number of parts that need to be transported, the risk increases. Hence the most optimal pick up point with minimal risk is when the machining of the last unit is just finishing as the delivery vehicle approaches.

Because the AGV starts and stops at the same point by design, the problem of ambiguity with different sources and sinks is completely removes. Once the required $\boldsymbol{W}_y$ are calculated by the fuzzy controller, an initial path is drawn connecting the workstation, and using the y values as weights, a genetic algorithm is developed and deployed to find the optimal path. In the end the proposed method in *MATLAB* using *Simulink* and *Stateflow* and the final results are compared against traditional centralized control systems.

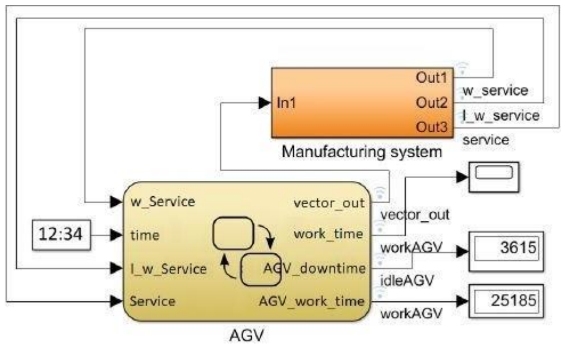

Figure 3: Matlab Simulink Model (Klosowski et al., 2018)

Drotos et al., (2020) proposed a method to balance the problem of optimizing route efficiency and minimize conflicts when more than few AGV‟s are in simultaneous operation by introducing few constraints in the setup of the workspace. Here the workstation and guided path placements are carefully planned and designed beforehand so that even in the case of multiple load carrying AGV‟s, the system would not be overwhelmed by requests.

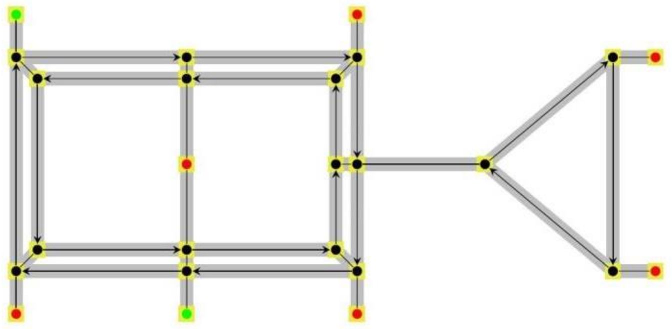

Figure 4: Example factory layout (Drotos et al., 2020)

Figure 4 is a representation of a simple factory layout under the proposed control scheme, where the green nodes are parking spaces, red nodes represent stations, and arrows indicate directed edges.

There are a few basic rules that need to be compulsorily followed by the individual vehicles during motion as follows,

- As the vehicles move, they respect the maximum speed (i.e., they cannot go faster), the maximum acceleration, and the safety distance.

- There can be at most one vehicle in any intersection or in any station simultaneously.
- The vehicles can neither overtake one another, nor go in the opposite direction simultaneously in the same lane.
- Changing direction in lanes is not allowed.

This method also considers online requests, which are external requests that come in as operations are being carried out. To make this feasible, the schedule or path planned for the individual AGV‟s are being dynamically solved and executed. The henceforth developed solution algorithm is simulated using computational methods and compared to a method described in Malopolski (2018) which deals with a similarly structured factory layout.

The main drawback with centralized systems is that due to the way the system is designed to control every AGV, the control system doesn‟t scale well with increasing number of mobile robots, having to take into account not only their individual paths but also collisions with other robots. Hence another proposed method to remedy these problems of centralized control is a distributed version where the AGV‟s are given individual pickup and delivery points and its upto them to decide on the optimal collision free path. Draganjac et al., (2016) proposed a similarly decentralized system of algorithmic control which utilizes autonomous mobile robots to complete tasks of their own discretion without a central controller. This proposed control system assumes the AGV‟s are free roaming, that is they do not follow a guided path network but actually can move almost anywhere in their nearby surroundings. While such a configuration can be a major drawback in centralized control systems to optimize, in the case of decentralized control, it introduces a degree of flexibility that is much needed to effectively implement the same. Each vehicle is assigned a private zone around its physical real world dimensions to help in conflict resolving.

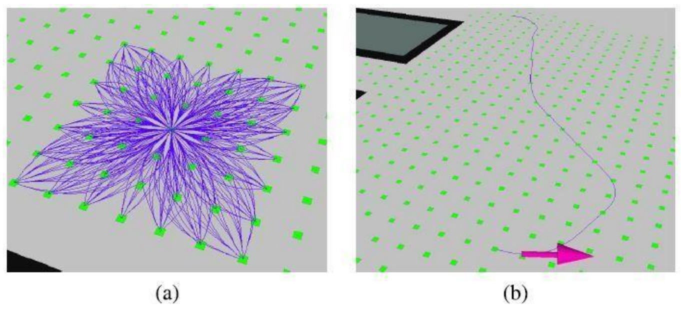

Figure 5: (a) Possible paths from given center point (b) Example of computed path based on lattice points (Draganjac et al., 2016)

For the case of path planning, to help efficiently solve free roaming path planning algorithms, the whole ground area is divided into a set of lattice points through which the AGV can imprint curves and hence calculate the optimal path to a given destination.

$$\kappa(s) = a + bs + cs^2 + ds^3 \qquad (1)$$

Equation (1) represents the cubic polynomial that is used to calculate individual path segments between the state lattice points. Figure 5 (a) displays all of the possible paths around a central lattice points about a 3 lattice point grid around the center. Figure 5 (b) shows an example for a path generated based on the lattice points, taking into account external parameters such as obstacles and the like. Due to the nature of cubic polynomial that is employed to resolve the path for the individual robots, the curves are of a distinct petal like shape that goes around the lattice structure through the intermediary points.

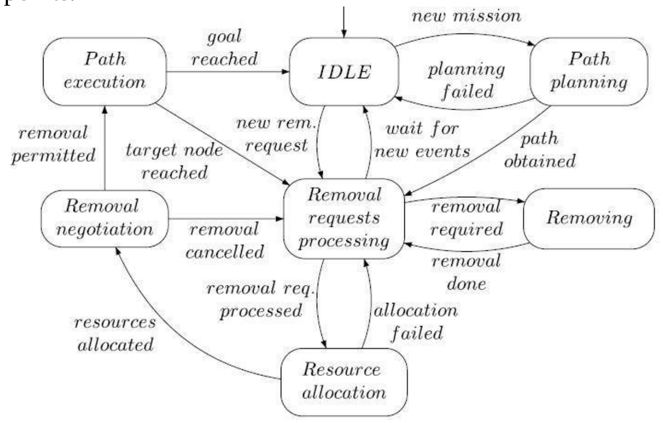

Figure 6: Overall machine logic flowchart (Draganjac et al., 2016)

Figure 6 shows the complete control loop running in each of the AGV‟s during operation. As seen a majority of the elements are concerned with removal application, which is nothing but the conflict resolution mechanism of the control system. Here removal refers to the removal of the lower priority AGV in between a conflict from the path of the higher priority one to ensure smooth flow of traffic during operation.

Another major issue that needs to be addressed is deadlock and livelock. Deadlock refers to a conflict condition where the traffic is such that none of the involving AGV‟s are in motion, as there isn‟t a sufficient conflict resolution in place to handle the issue. On the other hand livelock is an issue in which one or more of the AGV‟s are stuck in a loop continuously in motion, typically in a circular manner but not in a desirable fashion, as in not delivering or picking up required materials as commanded. As major path recalculation is explicitly forbidden, the given vehicles are constantly moving on the most optimum paths towards their destination. Although the motion of one or more vehicles can be temporarily interrupted by the higher priority vehicle requesting removal action, due to very fact that the higher priority AGV in each case proceeds with their motion towards their destination, there are no major blocks, and hence the given algorithm for conflict resolution can be classified and deadlock and livelock free.

Demesure et al., (2017) proposed a mixed architecture of central and decentralized control to achieve the best of both worlds. Initially a central supervisor processes and assigns a preliminary trajectory to help in easier decentralized trajectory control in the individual AGV‟s. Next the presumed trajectories are treated as a sort of intention for each AGV, which is then shared with all the physically nearby robots, to detect impending collisions if any and resolve the same according to a variable priority system which is dynamically computed.

During initial conflict detection and resolution the priority of the individual AGV‟s are calculated by a factor IP, consisting of a value between 0-1, with 0 being least priority and 1 being highest. The value of IP is calculated as the ratio of estimated travel time assuming a linear path and the maximum permissible travel time specified before the start of the operation. In case of each and every conflict, this priority factor is calculated for every AGV involved in the conflict, and the corresponding paths are calculated accordingly. The main conflict resolution schema is split into 3 major parts.

1) Scheduling Conflict: This conflict arises when two or more nearby robots select the same destination for the same time period. As such a conflict cannot be easily resolved with decentralized methods, a central supervisor checks the time period since last recalculation of path for each robot as well as their individual priority scores to assign an order by which the conflict can be resolved.

2) Sequential Resolution: In order to resolve conflicts involving deadlocks, the agents are sorted by their respective priority scores and so the conflict can be resolved by starting with the highest priority and end with the lowest one. Hence during conflict each agent only has to consider other agents with a higher priority in their calculations.

3) Collision Conflict Resolution: The trajectories that have been initially designed are checked for conflicts with all other trajectories that are of a higher priority value to detect collisions pre-emptively.

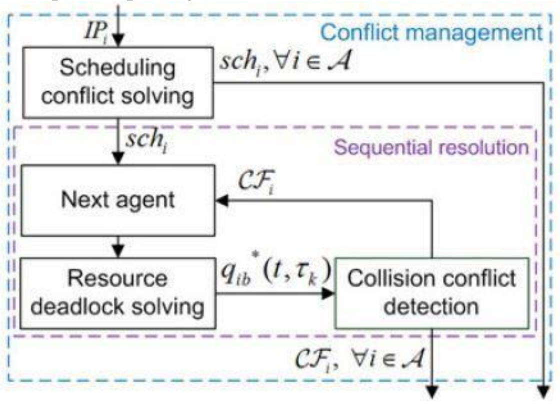
$IP_i$
Conflict management
Scheduling conflict solving
$sch_i, \forall i \in \mathcal{A}$
$sch_i$
Sequential resolution
Next agent
$\mathcal{CF}_i$
Resource deadlock solving
$q_{ib}^*(t, \tau_k)$
Collision conflict detection
$\mathcal{CF}_i, \forall i \in \mathcal{A}$

Figure 7: Conflict resolution mechanism (Demesure et al., 2017)

The resulting algorithms are tested numerically as well as experimentally with model robots and their respective paths and performance indicators are evaluated.

## III. RESULTS AND DISCUSSION

In the case of the proposed fuzzy logic based PID control as in Zhou et al., (2018), the resulting solution is validated by simulating the same in Matlab and comparing the response with a traditional PID Controller. The input given is a step response with a final output signal to be „1‟ for zero error.

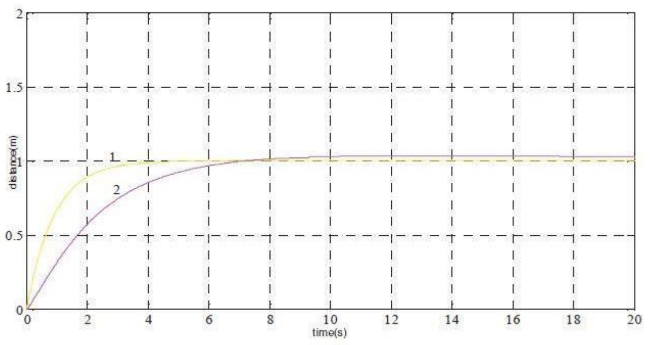

Figure 8: Step Response Graph (Zhou et al., 2018)

Figure 8 show the control scheme response to given step input. Here curve 1 represents the fuzzy logic control system whereas curve 2 is the conventional control system. As seen, the beginning curve slope is much higher for the new method, hence giving a faster response time. Also the traditional method of curve 2 gives a steady state error after reaching required goal, which is absent in curve 1 thereby giving a higher accuracy.

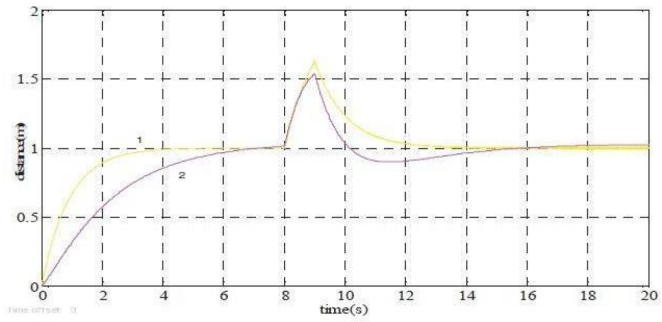

Figure 9: Step disturbance Response Graph (Zhou et al., 2018)

Now the given input has a disturbance or an obstacle given after reaching steady state, and the reaction of the two methods is recorded in Fig 9. As seen the oscillation of conventional PID system is much more severe than that of the fuzzy curve, along with which the time taken to reach steady state after disturbance is also higher, further cementing that the fuzzy logic based control is a more dynamic and robust system.

For the USB Camera based line detection and following algorithm proposed in Puppim de Oliveira et al., (2019) the resulting deviations in angular as well as linear dimension are plotted versus time.

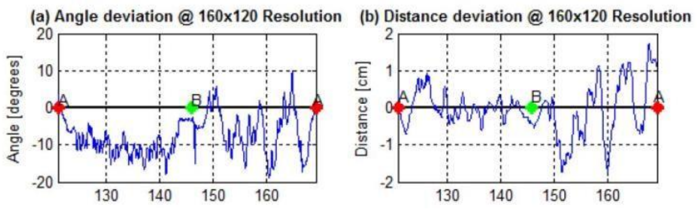

Figure 10: Angular and linear error (Oliveira et al., 2019)

As seen in Fig. 10, the total angle and distance deviation from the reference line/guided path across the recorded time period are quickly corrected with an appropriate PID activated output signals on the mecanum wheels. For given lower resolution the solution offers distance measurement resolution of about 0.5 mm with increasing the number of pixels decreasing resolution upto 0.125 mm for higher image resolution. Similarly for angular deviation, a minimum resolution of 0.6° is achieved at lower image resolution and at higher image resolutions upto 0.16° of angular resolution can be achieved, which even surpasses the current available commercial sensors for this application. Frame rates of above 100 frames per seconds are achieved from the camera leading to low latency and a quick dynamic response to external disturbances even when comparing them to traditional magnetic or infrared based sensors. Even though the proposed design is more accurate and robust than current sensors used, it the design still hinges on a Guided Path Network model for effective usage and hence is still limited by the inflexibility of the same.

The fusion of LiDAR and a wide angle camera lens as proposed in De Silva et al., (2018) is experimentally tested by actually building a model prototype of an Autonomous Mobile Robot fitted with a LiDAR sensors equipped with 16 lasers for frontal detection as well as a 360° wide angle lens camera and an onboard computer to process the images. Different image detection algorithms are tested as shown below.

| Algorithm | Accuracy | Precision |
|---|---|---|
| Proposed Gaussian Process Framework | 0.933 | 0.908 |
| Tensors factorization for missing values [36] | 0.8 | 0.827 |
| Robust DCT smoothing for grid data [37] | 0.923 | 0.921 |

Table 1: Comparison of resolution matching algorithms (De Silva et al., 2018)

As shown in Table 1, the robust DCT based smoothing method gives the best results in terms of precision, it falters in accuracy as compared to the proposed Gaussian process framework and hence to minimize uncertainty the Gaussian framework is used for further processing.

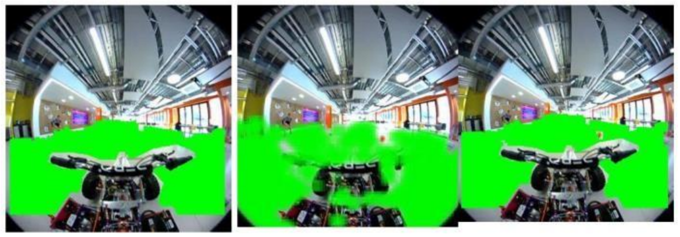

Figure 11: Comparison of final state space maps (De Silva et al., 2018)

Figure 11 shows the comparison of final state space models of the AGV surroundings with (a) Only LiDAR (b) Only Image (c) Fusion of both LiDAR and Image classifiers. While the LiDAR detects the entire area is front as a free range, it ignores few obstacles in its path which are picked up by the image classifier. Hence after fusion the final state space map not only tracks the entire surrounding area till the final edges, it also accurately detects and marks obstacles in the same.

For the centralized multimodal control system approach proposed in Klosowski et al., (2018) as both starting and ending point for the given AGV is the same, it completes a loop as shown below,

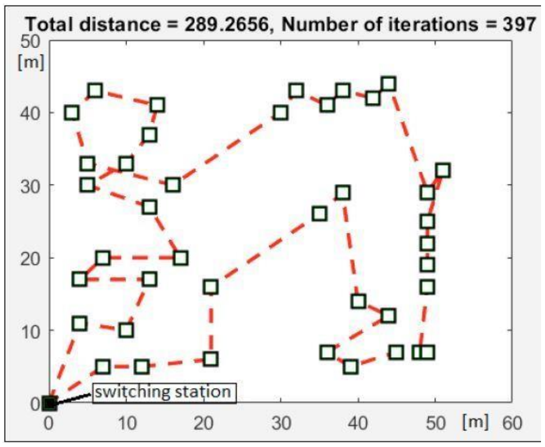

Figure 12: Workstations with optimized route (Klosowski et al., 2018)

Figure 12 shows the optimal path for all the given workstations such as to minimize travel distance. This path is found out by the iterative process of the implemented genetic algorithm.

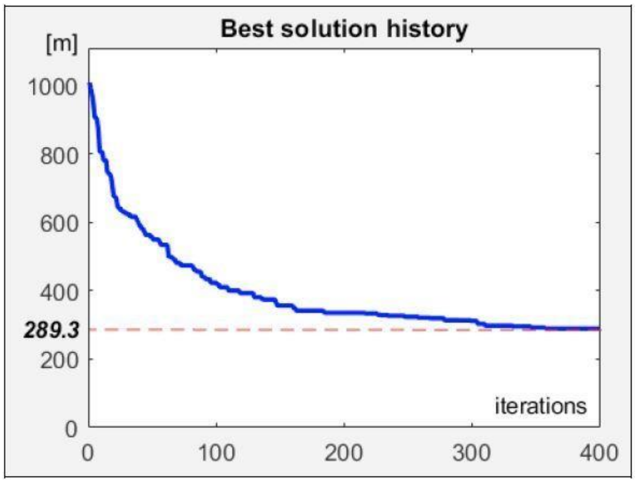

Figure 13: Route length vs. number of iterations (Klosowski et al., 2018)

Figure 13 shows the actual working of the genetic algorithm that instead of randomly choosing paths, over several iterations lands upon the most optimal path to minimize distance and hence the total travel lead time. After around 400 iterations a sufficiently optimal solution is found and the next part of the part planning is convened.

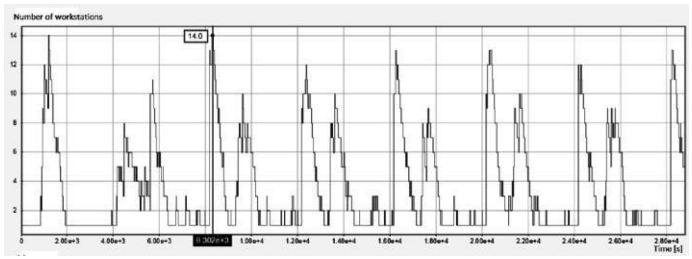

Figure 14: Number of workstations in need of servicing (Klosowski et al., 2018)

The simulation is set to run for an 8 hour shift and the graph of number of incoming service requests by the individual workstations is recorded as shown in Fig 14. At most there were 14 workstations that needed servicing simultaneously whereas at some times only one station needed servicing. A significant variation or high slope of decreasing indicates that the AGV is performing exceptionally well, as it shows the servicing needs of the workstations are being met in a timely manner. Although at the start of the simulation the risk factor was set at R=5, after the initial load run, it reduced to 0.8 as there wasn‟t much waiting time after servicing, and over the entire 8 hour shift the risk factor never went above 2.7, meaning there was never a need to deliver machined parts from a workstation in more than one load, meaning low inventory was kept throughout production, and hence the production efficiency was relatively high through the shift. Also the repeating cycling nature of servicing needs and in effect machined and un-machined part loads implies continuous efficient production without any buffer inventory shortages or overloads.

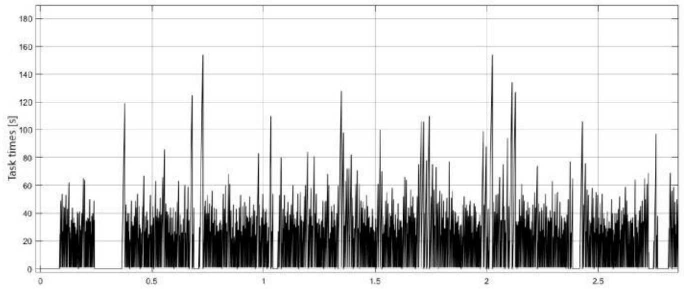

Figure 15: Duration of logistic tasks vs. time (Klosowski et al., 2018)

Another important factor to consider is how long the AGV was running during the entire 8 hour shift and how efficient was each run with the genetic algorithm optimized paths. As shown in Fig 15, during the total course of the shift, the AGV was delivering machined products and raw materials more often than not, hence leading to the low buffer inventory indicated by Fig 14. Also the maximum travel time for any one individual task was never more than 160 seconds, owing to the optimized path planning done prior to start of the shift, with the average task time being far lower around 40-50 seconds.

The results of proposed algorithm in Drotos et al., (2020) are obtained by computational methods, implemented in Java programming language.

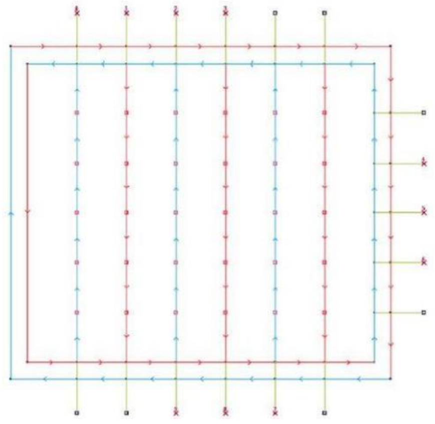

Figure 16: The initial layout used for simulation (Drotos et al., 2020)

As shown in Fig 16, the initial chosen layout is a cube where the initial location of the vehicle is denoted by an „x‟, a pickup node or point by a black square, a delivery node or point by a red square. The directed edges are red or blue while undirected edges are green. The total number of pickup stations is 7, number of delivery stations are 30 and total number of vehicles are 10. The effectiveness of this method is validated by comparing it to Malopolski (2018) in the results table. The improvements proposed to the traditional methods are implemented in both and represented separately. Also the case of loop elimination has been removed and replaced with a simple local search procedure; this method is also separately recorded. All of these methods are compared by defining a parameter called tardiness which is essentially a sum of ratios of time required to complete each request to the number of requests.

| Layouts | # req | Malopolski | Malopolski + impr | Our | Our +LS | Our +impr |
|---|---|---|---|---|---|---|
| *A* | 98 | 1556.57 | 8.23 | 24.16 | 0.51 | 0.03 |
| | | (967.72) | (12.50) | (26.27) | (1.77) | (0.26) |
| | 147 | 2585.60 | 137.16 | 231.10 | 78.07 | 20.61 |
| | | (1580.83) | (104.11) | (157.17) | (67.05) | (23.99) |
| *B* | 85 | 1641.88 | 77.74 | 45.88 | 2.47 | 0.19 |
| | | (953.36) | (64.89) | (41.28) | (5.16) | (0.69) |
| | 127 | 2628.42 | 328.24 | 264.49 | 115.21 | 51.33 |
| | | (1530.00) | (222.35) | (176.78) | (90.22) | (48.21) |
| *C* | 79 | 1039.87 | 47.69 | 2.47 | 0.05 | 0.01 |
| | | (659.43) | (45.81) | (4.53) | (0.21) | (0.03) |
| | 119 | 1874.22 | 299.88 | 126.54 | 34.78 | 14.54 |
| | | (1153.18) | (203.52) | (96.96) | (34.54) | (18.09) |

Table 2: Average tardiness (Drotos et al., 2020)

Here in Table 2, the „+impr" denotes the usage of procedure improvement schedule, while „+LS" denotes the local search heuristic that replaces the loop elimination module. The results show that the new proposed algorithm clearly outperforms the Maloposki methods which don"t show any comparable results without the improvements of the current method. Although the tardiness and hence performance factor of this method is well below previous methods, the overall design by its nature is dependent on a structured factory or warehouse layout which cannot be even slightly delayed from along with a set of stringent rules that severely affect the overall usability of this control system.

In the case of the decentralized control system as designed in Draganjac et al., (2016), the results were obtained by implementing the proposed algorithm in the robot operating system (ROS) framework. ROS is an open source platform that provides tools and libraries for building robot applications.

During implementation a complexity analysis was carried out and it turned out that the complexity of the proposed algorithm is of the order of O (KN) where K represents number of nodes or destinations in a given path and N represents number of vehicles currently under use. Hence this problem is that of linear complexity, which is a vast improvement over traditional centralized applications where the complexity is exponential, that is with the addition of every new vehicle, the time taken to solve trajectories increases exponentially. 10 similar robots of ascending priority were simulated, with all being given 30 random missions.

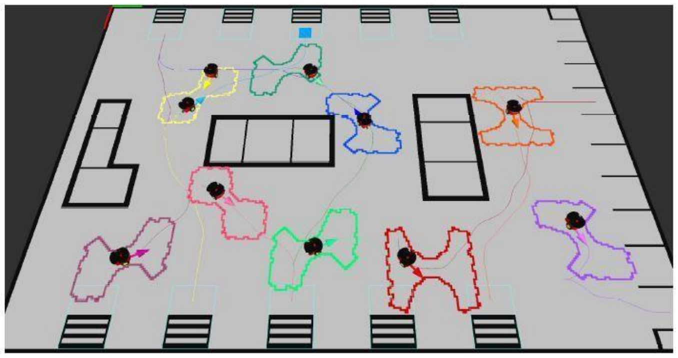

Figure 17: ROS Simulation (Draganjac et al., 2016)

Figure 17 shows a ROS package based visualization of the 10 autonomously controlled robots along with their private zones marked for help in resolving conflicts. There are also static obstacles around the workspace to simulate real life conditions.

Performance Metrics Obtained by Half Hour Simulation of a System Composed of Ten Vehicles

| Vehicle priority | Completed missions | Nbr. of yields | Nbr. of removals |
|---|---|---|---|
| 1 | 14 | 1 | 0 |
| 2 | 12 | 2 | 1 |
| 3 | 12 | 2 | 2 |
| 4 | 10 | 5 | 4 |
| 5 | 7 | 7 | 6 |
| 6 | 9 | 6 | 4 |
| 7 | 7 | 9 | 4 |
| 8 | 6 | 11 | 6 |
| 9 | 4 | 12 | 8 |
| 10 | 4 | 12 | 9 |

Table 3: Performance metrics (Draganjac et al., 2016)

After a 30 minute simulation of the above mentioned ROS implemented packages, the total number of deliveries, number of yields and removals for each robot is tabulated above. Here removal means the robot either stopped or backtracked away from its current position to make way for the higher priority opponent in a conflict whereas yield means the robot simply found a different path around the opponent to continue moving without conflict. As expected the higher priority vehicles have minimal number of yields and removals, as they usually have the right of way in encounters, and hence they have achieved the highest number of completed missions. This degrades as one goes down the priority list with the last 2 vehicles having only completed 4 missions each with a relatively large number of yields and removals.

Overall although this entire simulation was carried out in a single PC and not in distributed microcontrollers onboard each vehicle as in a real life scenario, the simulation was almost real time with minimal delay. Whereas in comparison to previous centralized methods the total time taken to simulate even one set of safe trajectories for each and every robot would be atleast a couple of hours ($10^7$ milliseconds). This method is limited by the fact that priorities of individual AGV"s are set in stone at the start of the system and in cases of conflict the most usual method of resolution is the lower priority AGV completely stopping or maybe even backtracking to make way for the higher priority one, leading to inefficiencies overall high task times for lower priority AGV"s.

In the decentralized path planning and conflict resolution algorithm proposed in Demesure et al., (2017), the numerical results were obtained by simulating the derived equations in MATLAB and graphing the same.

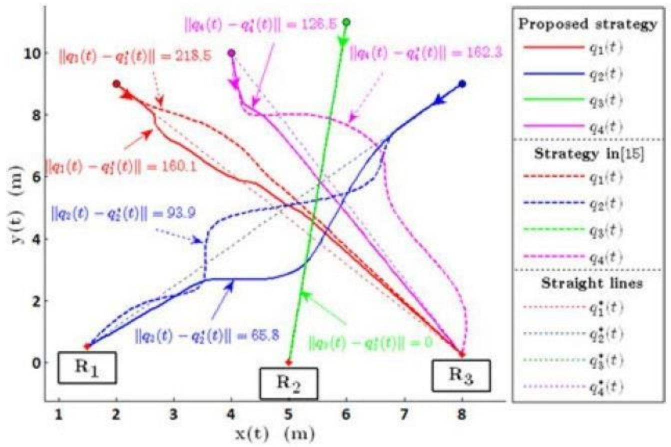

Figure 18: Graph of path planning simulation (Demesure et al., 2017)

The graph shown in Fig. 18 shows the comparison of the proposed control strategy to similar previous methods. The full dark lines are traces of paths as per the new strategy, whereas the dark dotted lines are as per a previous decentralized path planning algorithm. The light dotted straight lines are the ideal paths for each robot with minimal transportation time. As seen the new proposed strategy is much better at recalculating paths during conflicts so as to minimize total travel time.

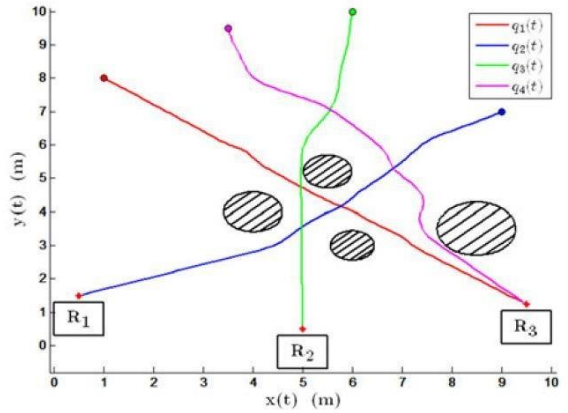

Figure 19: Path planning with obstacles (Demesure et al., 2017)

Figure 19 shows the final simulated paths of the agents when there are certain obstacles in between. Here the obstacle detected and avoidance are carried out locally and not supervised by any central controller.

| Number of agents | Global performance $P\%$ | | | Computational cost | |
|---|---|---|---|---|---|
| | Min | Mean | Max | Step 1 | Step 2 |
| 5 | 96.5 | 97.9 | 99.3 | 1.49 s (3) | 1.12 s (2) |
| 10 | 86.5 | 91.3 | 95.4 | 1.82 s (3) | 1.18 s (2) |
| 15 | 76.8 | 81.6 | 85.3 | 2.56 s (5) | 1.30 s (3) |
| 20 | 71.9 | 79.6 | 81.9 | 2.64 s (6) | 1.38 s (4) |
| 25 | 58.6 | 64.2 | 69.8 | 2.66 s (6) | 1.41 s (4) |

Table 4: Simulated performance indicators (Demesure et al., 2017)

The global performance value is calculated as a ratio of the initial completion time as estimated by the first trajectory and the final completion time for the task after the simulation is over. As seen in Table 4: The performance indicators are better for lower number of agents meaning the ideal travel time is close to the actual one. Even in the case of a large number of agents, the computational cost is exponentially lower than those using centralized methods. Here majority of computational cost is in the initial step of path planning, and can be reduced with better implementation in lower level programing languages such as C++. Moreover the robots during navigation never stop or retreat, they just recalculate their path and move accordingly. Similarly the proposed method was implemented in 4 real life model robots and their performance was recorded.

| | Final time (simulation) | Final time (real) | Completion time | Operation due date |
|---|---|---|---|---|
| AGV 1 | 25.44 s | 25.92 s | 31.6 s | 76.1 s |
| AGV 2 | 17.76 s | 18.07 s | 22.38 s | 30.8 s |
| AGV 3 | 13.14 s | 13.33 s | 18.33 s | 23.8 s |
| AGV 4 | 21.02 s | 21.68 s | 26.36 s | 60.2 |

Table 5: Experimental results (Demesure et al., 2017)

Table 5 shows the final results of the model robots as compared to simulated test runs. There are only minute differences between the simulated and actual travel times, and during the entire operation there was not a single collision that had to be resolved with external intervention. Hence this proposed method is validated experimentally.

## IV. CONCLUSIONS

In the case of micro control the conclusions are as follows,

- The implementation of a fuzzy logic based variable PID controller feedback loop is a definite upgrade over traditional PID controls resulting in quicker response time period as well as a more robust control system.
- Fusion of LiDAR and wide angle camera lens in AGV provides a very detailed state space map of its surroundings and is necessary for the implementation of a robust free roaming control setup not confined to conventional Guided Paths Networks.

For the overall trajectory control of AGV"s, the conclusions are as follows,

- In cases of a single AGV with multiple destinations such as the ones in manufacturing, a genetic algorithm enhanced fuzzy logic based path planning algorithm is the optimal solution to ensure efficient production.
- Furthermore, the addition of a Risk Factor not only helps evaluate current status of production but also helps limit buffer inventory and hence optimize cost efficiency of the overall production cycle.
- In scenarios involving multiple AGV"s as well as multiple destinations such as warehouse distribution centers, a semi-decentralized control system with an initial centrally calculated trajectory and dynamic priority allocation is ideal for real-time conflict resolution as well as minimizing total travel time for the individual AGV"s.

**ACKNOWLEDGEMENT**
We would like to thank our seniors for encouraging us to explore different research avenues and providing us the opportunity to present the same in the form of this paper. We would also like to thank the college libraries for granting easy access to various research.

## V. REFERENCES

Bechtsis, D. , Tsolakis, N. , Vlachos, D. , & Iakovou, E. (2017). Sustainable supply chain management in the digitalisation era: The impact of Automated Guided Vehicles. *Journal of Cleaner Production, 142* , 3970–3984 .

Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., & Djemai, M. (2017). Decentralized motion planning and scheduling of AGVs in an FMS. *IEEE Transactions on Industrial Informatics*, *14*(4), 1744-1752.

De Silva, V., Roche, J., & Kondoz, A. (2018). Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots. *Sensors*, *18*(8), 2730.

Draganjac, I., Miklić, D., Kovačić, Z., Vasiljević, G., & Bogdan, S. (2016). Decentralized control of multi-AGV systems in autonomous warehousing applications. *IEEE Transactions on Automation Science and Engineering*, *13*(4), 1433-1447.

Drótos, M., Györgyi, P., Horváth, M., & Kis, T. (2020). „Suboptimal and conflict-free control of a fleet of AGVs to serve online requests." *Computers & Industrial Engineering*, 152, 106999.

Fragapane, G., de Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research.*

Kłosowski, G., Gola, A., & Amila, T. (2018). „Computational intelligence in control of AGV multimodal systems.*' Ifac-papersonline*, 51(11), 1421-1427.

Małopolski, W. (2018). A sustainable and conflict-free operation of AGVs in a square topology. *Computers & Industrial Engineering*, *126*, 472-481.

Puppim de Oliveira, D., Pereira Neves dos Reis, W., & Morandin Junior, O. (2019)." A qualitative analysis of a USB camera for AGV control." *Sensors*, 19(19), 4111.

Zhou, X., Chen, T., & Zhang, Y. (2018, November). „Research on intelligent AGV control system." *In 2018 Chinese Automation Congress (CAC)* (pp. 58-61). IEEE.