# A Fuzzy Rule-Based Control System for Fast Line-Following Robots

G. Eleftheriou
*Department of Computer Science*
*Intelligent Systems Laboratory*
*Neapolis University Pafos, Cyprus*
i.eleftheriou@nup.ac.cy

L. Doitsidis
*School of Production Engineering and Management*
*Technical University of Crete, Greece*
ldoitsidis@dpem.tuc.gr

Z. Zinonos, S. A. Chatzichristofis
*Department of Computer Science*
*Intelligent Systems Laboratory*
*Neapolis University Pafos, Cyprus*
zinon.zinonos, s.chatzichristofis@nup.ac.cy

***Abstract*—This paper focuses on the optimization of the line-tracking and following control technique used in high speed autonomous movable IoT devices such as line-following robotic vehicles. A fusion of a new pro-active / feed-forward control system, based on the simultaneous use of computer vision and the use of an array of analog infrared reflective phototransistors with an optimised PID based feed-back implementation is proposed. Experimental evaluations demonstrate that the proposed approach aids and improves the performance of line-following robots compared to the traditional PID technique.**

***Index Terms*—IoT, robot, control system, Fuzzy Logic, PID, Computer Vision**

## I. Introduction

An autonomous robot is an intelligent unmanned vehicle which covers a set of frontier research fields including environment perception, pattern recognition, navigation and positioning, intelligent decision and control and computer science [1]. The last decades, an increasing interest has been recorded on the exploitation of unmanned vehicles in fields such as environmental monitoring [2], commercial air surveillance [3], domestic policing, geophysical surveys, disaster relief, scientific research, civilian casualties search and rescue operations, maritime patrol, traffic management, etc. Regardless the domain that they belong to, the key element that distinguish them as the leading edge of their technology is the provided degree of autonomy.

In general, autonomous operation of mobile vehicles is of crucial importance in day to day activities and an increasing set of them is depended on robots operating autonomously in unknown cluttered environments. Designing systems that are able to operate in the aforementioned manner isn't trivial and requires significant effort.

Most recent forms of such automata are based on differential drive autonomous mobile robotic vehicles that usually implement line-tracking and following using a navigation technique that relies on feedback from an array of photosensitive reflective photo-interrupters that detect a non-reflective (black) line on a reflective (white) surface. It is worth noting that in real-world applications, complex line structures (i.e. crossovers, junctions acute and right-angles) may exist on the path, which make the challenge even greater as far as it concerns the design of robust controllers [4].

The vehicle motor speed settings and control include the Proportional Integral Derivative (PID) control methods [5], predictive control methods [6], fuzzy control methods [7], model reference adaptive methods [8], neural-network control approaches [9], fractional-order control methods [10], etc. Most of the recently proposed methods adopt the PID control mechanism. PID control exhibits significant advantages including its simple structure, good control effect and robust and easy implementation [1], [11]. A pure line tracking algorithm is highly likely to produce unstable results considering the robot has to turn when faced rotation points or the endpoint of the lines. Moreover, surface wear, ambient lighting variation, noise in the readings, incorrect pose of the robot due to bouncing of the wheels etc. can inhibit the robot's behaviour.

Fuzzy logic can provide a viable alternative to the aforementioned approaches, since it is a tool from the computational intelligence domain that can provide robust controllers for complex and non-linear systems based on human expert's knowledge. The wide applicability of fuzzy logic in autonomous navigation is mainly based on suitable knowledge representation of inherently vague notions achieved through fuzzy IF-THEN rules. These rules typically contain linguistic information, which describes the problem at hand very simple and fast. Further, in the majority of fuzzy logic application in navigation, a mathematical model of the dynamics of the vehicle is not needed in the design process of the motion controller. Only the problem-specific heuristic control knowledge is needed for the inference engine design. From a more practical point of view, fuzzy logic is the most appropriate modelling tool for representing imprecision and uncertainty of the sensor readings. Another reason that explains the popularity of fuzzy logic in autonomous navigation is the low computation time of the hardware implementations of fuzzy controllers which favors real-time applications [12].

Apart from the straight-forward purely fuzzy logic controllers, different schemes have been proposed in the literature, based on hybrid fuzzy logic approaches. In [13] the authors have proposed a scheme based on Fuzzy-PI controllers that can adjust their parameters $(K_p, K_I)$ to reduce the error caused by the dynamic changes and navigation challenges of an omnidirectional robot combined with an additional fuzzy controller which is responsible for the obstacle avoidance

2325-2944/20/$31.00 ©2020 IEEE
DOI 10.1109/DCOSS49796.2020.00068

Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

system.

This paper proposes the employment of a PID closed-loop control system fused in parallel to a Fuzzy-Ruled Based Computer-Vision system. This system detects and recognizes the line ahead of the robot and reports it to the on-board micro-controller so that the robot's next move can be evaluated ahead of time and a speed variation parameter can be manipulated to slow down or accelerate the robot in pre-decided situations, thus greatly increasing its performance.

In brief, the proposed method adopts the following methodology: an onboard IR sensor array collects data for each IR sensor in regards to the robot's position towards the line. In parallel, the robot's camera takes a snapshot of the area ahead of the robot and the camera's microcontroller identifies the length of the line which is in sight. In sequel, the outputs of the two data collecting devices are used as inputs to a Fuzzy Rule-based system. The Fuzzy Rule-Based system generates a crisp output regarding the current status of the robot towards the line. Based on the crisp output an optimal $K_p$, $K_i$ and $K_d$ set of parameters and an appropriate speed setting are retrieved.

The rest of the paper is organized as follows. In section II a detailed literature review of different line-following approaches is presented, while in section III the custom robotic device is described in detail. In section IV the proposed approach is formally analysed, while in V the experiments that highlight the advantages of the approach are presented. Finally, section VI draws the concluding remarks.

## II. Related Work

A large number of existing studies in the broader literature have examined the immensely increasing need in today's world for the involvement of assistive robots in many aspects of day to day life. In [14], a line following robot which can securely transfer materials inside an office autonomously as per user direction, e.g. by calling the robot by the push of a button, was developed. The researchers experimented with a simple ON-OFF line following algorithm and the classical PID algorithm. The classical PID algorithm proved smoother and faster in following the line compared to the ON-OFF algorithm, but the line following track had to be simple and right-angled and acute-angled free for the robot to follow the line robustly.

To solve the various challenges that are specific to the mobile robot line tracking system due to the inability of the PID controller to successfully control such a highly non-linear and unstable system, the authors in [15] proposed a customized PID control implementation scheme where: a) a Modified Integral Control is implemented via an integral control effort that strives to counteract steady state errors in curved paths by summing up only the last ten error values instead of adding all the previous values, and b) a Speed Variation Parameter is added to improve performance by reducing the motor speed at turns and curves so that stability is maintained. The proposed implementation adds a heavy computational effort to the microcontroller and provides only minor improvement to the overall track time of the robot.

In [16], a driver assistance system to achieve a driver workload reduction is presented. The system supports the driver with two driving tasks - keeping in lane and maintaining vehicle headway. The system consists of a radar-equipped adaptive cruise control system that can detect a forward vehicle and a lane keeping assistance system which consists of a camera-equipped lane recognition unit and an Electric Power Steering. The system calculates the optimum steering torque to assist with lane keeping based on the relationship between the vehicle's position and the recognized lane by the camera, or the target path. The lane keeping assistance algorithm consists of a feedforward path torque using the road radius and a feedback path torque using the vehicle heading angle and the lateral deviation within the lane. The torque transmitted to the electric power steering is the result of the fusion of 2 separate inputs, the calculated torque and the driver's steering torque. The proposed system proved to be so robust that was introduced on the Japanese market as the Honda Intelligent Driver Support System (HIDS).

The authors in [17] discuss the plethora of sensors that could be used in line tracking cars that compete in robotic competitions such as camera sensors, colour sensors, and laser sensors. It also covers the necessity of capturing the track line with high speed and the need for the data from sensors to be as simple as possible. Since the more elaborate sensors can make the hardware more complicated, in this paper the infrared sensor is promoted for this work and both the digital and analog variants are evaluated. The researcher's conclusion is that the use of analog sensors, while costing a bit more time to compute the input data, can reveal more details about the position of the robot towards the line and by carefully tuning a PID controller the autonomous line tracing car could move more smoothly than with the use of digital sensors. To make the robot move faster than the competition they propose using 3 microcontrollers and a combination of analog sensors, digital sensors, an acceleration sensor and marks on the right side of the track line to indicate incoming acute turns so the robot can decrease its speed to be able to navigate the track smoothly. This approach not only complicates the build of line tracking robots, but it also reduces the capability of the robot to move autonomously since it will depend on line marks to navigate difficult paths robustly.

Another study [18] presents a controller design of line following together with distance keeping based on fuzzy logic for application of automatic guided vehicle, using an array of infrared light sensors and an ultrasonic range finder for distance detection. The implementation is based on a proportional only controller with a table of reasoning rules for the fuzzy controller and a Fuzzy set of controller output. The design of the fuzzy control is utilized in line following together with distance keeping control. The results show that the two controls can be easily combined directly because they can be almost independent via the design and a crisp output can drive two PWM capable servo motors to complete the tasks at hand.

Additionally, the authors in [19], proposed an algorithm



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

for line tracking and object recognition with a camera, that takes central points by a forward-facing Pixy camera, in order to perform proportional only tracking and following of a bright red line. Bright red colour is used, so the Pixy camera can utilize the build in colour tracking algorithms to identify which path to follow, distinguishing the bright hue over the contrast colour of the floor, which avoids confusions. Labelling the objects with a specific colour allows Pixy camera to understand which object is the target one. The line-detection process is a simple method performed by taking the central point value of the red line according to the sight of the Pixy camera and steering the robot in either left or right. In this work the researchers added barcodes on the path that reveal the incoming turns to help the robot navigate more robustly, making the algorithm not suitable for autonomous driving.

In another approach [4], a Finite State Machine is introduced that detects irregularities in the measurements from an array of IR sensors during wheeled robot line following. Based on the estimated state, the FSM switches between a proposed variable-gain PD controller for line following and an open loop controller for handling special cases. The contribution of this work lies in the combination of the introduced FSM for smart filtering of IR measurements, with a motion controller that adjusts the robot's response to turns and its linear velocity based on the complexity of the path. In this proposed methodology, that relies vastly in the time consuming task of continuously saving the sensor measurements and constantly comparing them to known possible measurements of identifiable line types, the line following robot can accurately track the path by achieving high speeds at straight segments and lower speeds in sharp turns. Moreover in this work the assumption of the robot makers and competitors that the robot is continuously well-aligned with the line is discussed and it is proven false and several examples are demonstrated where the robot may confront a right-angle under a wide variety of different poses, because of sliding, drifting, vibration and oscillations during its movements, which may result to misleading measurements from the IR sensors.

This work [20] presents a model of autonomous control of a line following robot with the capability to know its position based on continuous speed and length measurements based on the readings from both an 8 digital IR optical sensor board and two rotary encoders attached to the rear driving wheel axles, filtered through a histogram filter and compared to a predetermined map. The current position is determined by a markov process histogram filter and the error in measurements is embedded in calculating the location probability. The model accounts for measurement errors and battery depletion as influencing factors and has high processing demands. Even though the researchers were able to achieve some decrease in elapsed time over the classical methods, the results show higher relative errors of the measured distance when there is curvature. This is due to the different paths of the inner and outer wheels, when robot follows curvature paths. In the current implementation there is a mechanism for averaging the path travelled from both inner and outer curvature, but simple averaging does not perform well enough and the relative errors have high values.

Finally, in [21] the process of design, implementation and testing a small line follower robot designed for line following robot competitions is illustrated and the technical and mechanical issues and problems are investigated. The authors highlight the most difficult aspects of such a task such as the line sensing process that requires high resolution and high robustness due to the often poor reflection of the line to be tracked, the difficulty to accurately control the speed according to the lane condition, the need for the speed to be limited during passing a curve due to the friction of the tire and the floor, the difficulty to design the correct and capable electrical circuit, the process of selecting the appropriate processor/microcontroller, type of motors and motor driver, wheel and tire combination, chassis type, weight and dimensions, the complexity of the programming. Overall, the paper discusses the plethora of the variables that can affect the design and implementation of line following robots but does not begin to describe the lengthy effort of the actual tuning of the robot and the various methodologies that are available including the selection of the correct algorithm and the level of the robustness of the control system that accompanies it.

## III. The Custom Line-Following Robot

To highlight the proposed approach, which will be described in detail in Section IV, a custom robotic vehicle was developed. We have used off-the-shelf components aiming on developing a low cost, very low computationally demanding, agile lightweight vehicle (less than 0.5 kg), capable of performing autonomous navigation by fusing the information from different sensors and processing it on board, using a lightweight algorithm that can be used in any kind of similar or up-scaled platform. An overview of the different components and modules and how these are distributed and interact with each other in our navigation framework is illustrated in Figure 1.

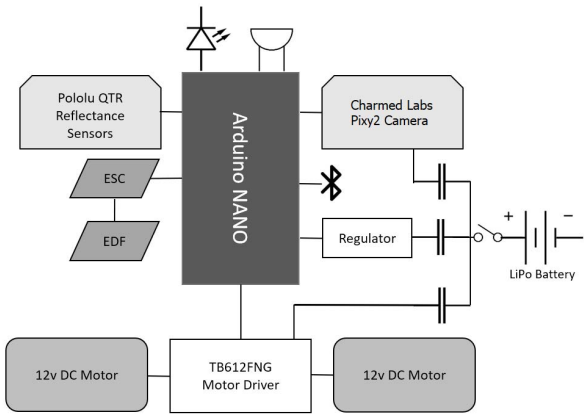


Fig. 1. An Overview of the Components.

The core of the system is an Arduino NANO board with a low end 16MHz ATmega328P microcontroller. It supports



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

an Input Voltage of 7-12 V, it boasts 22 I/O Pins, 6 of which have PWM output and 8 of them are Analog. It has a compact size of 18 × 45 mm and weights 7g.

For the data acquisition a Pololu QTR-8A Analog Infrared-reflective phototransistor 8 sensor array and a Charmed Labs Pixy2 camera were used. The Pololu QTR-8A sensors feature a linear array of IR emitter/phototransistor pair modules that are well suited for applications that require detection of changes in reflectivity. This change, can be due to a color change at a fixed distance, such as when sensing a black line on a white background, as well as due to a change in the distance to or presence of an object in front of the sensor. These sensors are offered both in digital and analog outputs.

Pixy2 vision sensors are small, relatively light (10g), inexpensive, fast and capable cameras that can be easily connected and interact with microcontrollers using UART serial, SPI, I2C, USB, digital and analog outputs. Pixy2 uses the NXP LPC4330 204 MHz, dual core Processor and the Aptina MT9M114 1296 X 976 resolution Image sensor with integrated image flow processor. It has a 60 degrees horizontal, 40 degrees vertical Lens field-of-view and approximately 20 lumens integrated light source.

For the differential drive system 2 × 16mm 12 volt 5000 RPM DC motors, with a stall current of 2.5 Amps each were integrated. They were specifically chosen for their high speed and great torque. To drive the motors 2 × 100KHz PWM capable TB6612FNG Dual Motor Drivers where connected together, thus boosting their capabilities to our desired amperage ( 5-6 Amp at full power). The TB6612FNG is a dual-channel full-bridge drive chip. The maximum continuous drive current of single channel can reach 1.2A, peak value 2A/3.2A (continuous pulse / single pulse), which can drive DC motors.

To optimally align the chosen components we‘ve designed and manufactured a custom Printed Circuit Board (PCB) and placed our parts on it. The PCB served also as the robot's chassis, therefore minimizing the need for additional structural enhancements.

The robot's PCB features the appropriate sockets to plug in an Arduino NANO, a dual TB6612FNG Motor Driver, an Infrared Remote Control Start/Stop Module to remotely drive a digital I/O Pin High or Low, a HC05/06 Bluetooth module that can easily achieve serial wireless data transmission for an easy IoT integration and an Electronic Speed Controller (ESC). The ESC is needed to drive the electric ducted fan which is a brushless DC motor with a propeller attached to its axle, enclosed in a plastic tubular enclosure. It is used to add downforce to the robot to add traction at high speeds.

The PCB also has precise location soldering pads for a Pololu QTR8A IR Sensor Array at the center of the front end, a 5v regulator to give stable voltage to all the 5v devices, capacitors for the motor drivers, a JST power connector, LEDs and resistors, 3× Double Pole Double Throw Slide switches for the main power circuit, the ESC power side and ESC signal side lines, a 3-way-8 combination Dual Inline Pole tactics switch, a tactile Button, 2× 16mm motors, a Buzzer and 4× direction LEDs. There is also a hole to accommodate a 27-30mm Electric Ducted Fan and provision for its mounting bracket. There are also provisions for the Camera mounting bracket, the motors mounting brackets and the slider mechanism. The dimensions were carefully selected at H 205mm × W 100mm and 2 × 5mm × 50mm extensions were incorporated, one on each side, in front of the wheels, to house floor swiping mechanisms to help remove dirt and dust before they are caught by the tires and cause traction loss .

Custom wheels were 3D printed, giving the advantage to control their weight and radius for optimum acceleration and their width for optimal traction at high speeds. The tires were molded using a silicon-polyurethane compound that yielded a Shore A17 hardness.

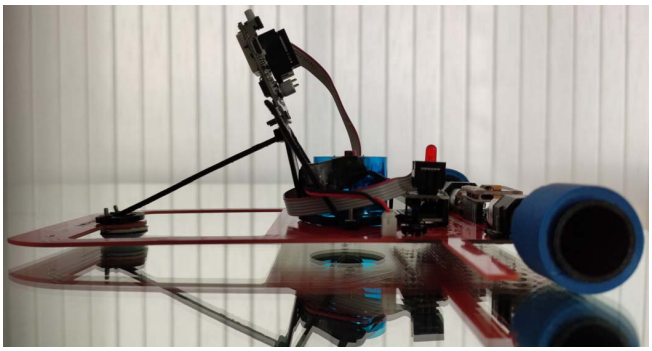

Fig. 2. Side-View of the Robot Constructed for our Experiments.

A 5V regulator took care of the power supply to both, the Arduino, the QTR-8A and the Pixy2. Figure 2 depicts the robot used in our experiments.

## IV. Proposed Navigation Algorithm

In the case of autonomous line-following a PID-only line tracking controller can be used with great success up to certain speeds only and on fairly easy to navigate paths.

The overall PID control function can be expressed mathematically as :

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e\,(t')\,dt' + K_{\mathrm{d}} \frac{de(t)}{dt} \qquad (1)$$

where $K_p$, $K_i$ and $K_d$, all non-negative, $u(t)$ is the output of the controller, $e(t)$ is the error, i.e. the difference between the setpoint and the process variable and $de(t)/d(t)$ is the rate of change in time.

A PID-only controller can be used to robustly track and follow simple paths up to speeds of around 1 meter per second, but when the speed is increased or the path consists of acute turns, the nature of the feedback system inhibits fast reactions since the correction value is always dependent on a past measurement of the deviation from the setpoint. Moreover, a single set of PID tuning parameters is never optimal for autonomous line tracking and following in all types of paths, but is only optimal for the type of path that it is optimally tuned.



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

One can easily validate the above limitations of the classical PID-only control method implementing the following scenario: Construct a pure line tracking robot and a simple line following path, consisting of only wide angled curves. Optimally tune the PID parameters of said robot for fast line tracking and following the constructed path using the widely acceptable Ziegler-Nichols method [22], or any other tuning method [23] [24]. Test the performance of the robot and document the results over a number of attempts. In sequel add a single acute turn or right angle turn and test the performance of the robot again. Doing so, one can easily conclude that the robot, though optimally tuned, will always fail to follow the acute or wide angle turn, with the result always being line loss under the sensor area. The same unorthodox results can be observed if the scenario is reversed, i.e, a complex path with acute and right angles is constructed first, the robot is optimally tuned to track and follow the path as fast as possible and it's performance is recorded. When the path changes to a simpler one, the performance will be significantly reduced compared to a tuned robot for a simple path.

From the above, we can safely infer that the PID-only controller, when it comes to controlling a robot towards autonomous tracking and following a line/path, has profound limitations due to the extreme changes of the path curvature. An extreme change in the path, such as an acute or right angle, can lead to large deviations from the set-point and consequently large errors that can't be controlled adequately, resulting in sudden loss of the line under the sensor and uncontrollable behaviour.

Consider the case presented in Figure 3 where the classical PID controller leads to line loss due to extreme changes in the path curvature. One can visualize how the PID-only controller, due to it's feedback dependency, can suffer from accidental line loss.

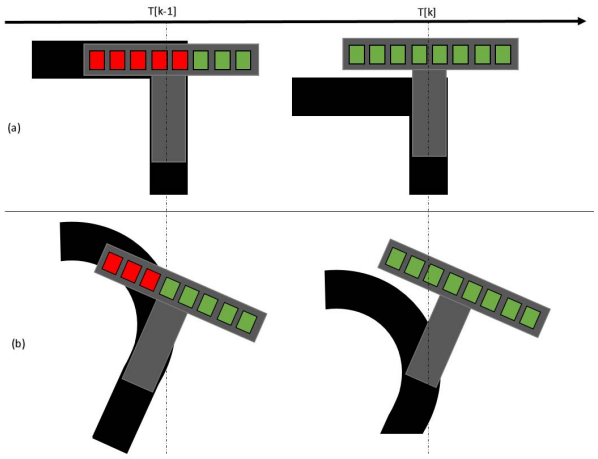


Fig. 3. Different cases of line loss.

Moreover, PID also suffers from overshot and parameters setting in tracking performance [25], characteristics that create the need for a new control system, more suitable and more robust for line following and tracking.

Our proposed method strives to correct these limitations by adding an anticipatory control, via the addition of an input from a new feed-forward system that can greatly enhance the ability of the PID controller to react timely in fast non-linear deviations from the set-point.

The ability to be pro-active can aid in the manipulation of the speed of the line-following vehicle (e.g decelerating when an acute turn is in sight, accelerating when a simpler path is present etc.etc), thus greatly enhancing the ability to navigate complicated paths faster. Several researchers have proposed applying a visual system to autonomous vehicles to capture and analyze the visual images to overcome the limitations of other sensors [26].

We propose a new algorithm that combines the advantages of the PID with the benefits of a visual system using a Fuzzy Ruled-Based approach.

Consider a robot towards achieving the objective of line-following. For ease of understanding, it is assumed the augmented decision vector:

$$\mathbf{x}(k) \equiv \left[x_k^1, x_k^2, \ldots, x_k^N\right]^\tau$$

where $N$ denotes the number of the decision variables of the robot at $k$ iteration. These decision variables represent the controllable parameters of the robot, in our case the speed of each motor. The augmented vector which contains the available exteroceptive measurements takes the form:

$$\mathbf{y}(k) \equiv \left[y_k^1, y_k^2, \ldots, y_k^M\right]^\tau$$

where $M$ denotes the number of sensor measurement of the robot at $k$ iteration. In our case, $\mathbf{y}(k)$ originates from two different sensors by repeated measurements over time, a set of infrared reflective phototransistors and a 120 frame-per-second rate camera.

The optimal $\mathbf{x}(k)$ vector at $k$ iteration in the proposed control strategy can be straightforwardly derived from:

$$\mathbf{x}(k) = \mathscr{F}_1(\hat{\mathbf{y}}(k)) + \mathscr{F}_2(\dot{\mathbf{y}}(k)) \tag{2}$$

Let $\hat{\mathbf{y}}(k)$ be the vector consisting of the entries of $\mathbf{y}(k)$ related to the infrared reflective sensor and $\dot{\mathbf{y}}(k)$ be the vector consisting of the entries related to camera sensor. In other words, $\mathscr{F}_1$ takes into account only the IR array measurements while $\mathscr{F}_2$ considers only the camera's measurements. In general, the proposed method follows $\mathscr{F}_1$ to manipulate the steering parameters, while $\mathscr{F}2$ manipulates the speed parameters of the robot and selects an optimal set of $K_p$, $K_i$, $K_d$ parameters for the desired speed.

$\mathscr{F}_1$ uses as input the infrared reflective phototransistors' measurements and in the case of absence of input from the camera, based on a traditional PID controller, calculates safe steering and speed parameters. In general a PID controller continuously calculates an error value as the difference between a desired setpoint and a measured process variable. When fine-tuned, it produces an output that can be used as a correction



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

towards the desired set-point. PID is typically accurate and practical in many real life scenarios where the deviation from the set-point is linear.
In the presence of input from the camera $\mathscr{F}_2$ selects a more suitable set of PID parameters and speed for various cases and lengths of line.

The proposed navigation mechanism is summarized in Algorithm 1.

**Algorithm 1:** Proposed Navigation Algorithm

**Result:** Augmented Decision Vector $x(k)$

```
initialization;
while True do
    Capture IR measurements;
    Capture an Image Frame;
    Detect Lines;
    if Only one Line Detected then
        Calculate Length of the Line;
    else
        Calculate the Intersection Point;
        Retrieve Position of the Intersection Point;
        Calculate Length of the Line;
    end
    Based on the current speed of the motors, calculate
     X_1;
    Based on the Length of the Line, calculate X_2;
    Use the Fuzzy Ruled-Based System to Calculate F_2;
    Retrieve the optimal K_p, K_i and K_d values for F_2;
    Calculate F_1 based on the IR sensor's measurements;
    Calculate x(k);
end
```

In our implementation, the output of the $\mathscr{F}_2$ is based on a fuzzy interface. In general, a fuzzy inference is used for the process of determining the response of a system to a given input by using fuzzy logic and fuzzy linguistic rules for expressing the system's I/O relation. Its main characteristic is that it inherently performs an approximate interpolation between '*neighboring*' input and output situations. The proposed system uses a Mamdani fuzzy rule-based system to provide a personalized feedback for each participant and consists of two fuzzy inputs and one fuzzy output.

The first input of the fuzzy system $\mathcal{X}_1$ corresponds to the current maximum speed of the motors:

$$\mathcal{X}_1 = \frac{\max\{\mathbf{x}_k^j\} \times 100}{|X'|},\ j \in [1, \ldots, N] \tag{3}$$

where $k$ refers to the current iteration and $|X'|$ denotes the maximum operation speed of the motors (rpm).

To visually detect the position and the length of the line, the proposed architecture uses a Pixy2 vision sensor. To detect vectors of black lines on white surfaces and calculate and return the length of the detected line that we need to feed $\mathscr{F}_2$, we use an algorithm that operates on the Pixy2's microcontroller to detect both lines and line segments.

Since our robot is always facing a path and its objective is to follow said path in a forward motion, a primary line is considered to be the detected line that has its start point at the bottom edge of the image captured by the Pixy2 camera in every timestep. Our algorithm identifies all the other lines in the image and calculates their intersection point with the primary line. In most of the cases, only 1 (the primary) or 2 lines are detected at each timestep, $L_1$ and $L_2$. In case of more lines, we select the intersection point of the primary line with the other lines using a voting system, i.e we select the intersection point as the point where most of the other lines cross the primary line. Doing so, we can easily calculate the length of the detected line or line segment, since this will be either the difference between the starting point and the ending point of the primary line or the $y$ coordinate of the intersection point.

Hereafter, the length of the detected line at $k$ iteration will be refereed as $L_k$. The intersection between 2 lines $L_1$ and $L_2$ can be defined using the following equation:

$$L_k = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} - \frac{(y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

with line $L_1$, being defined by two distinct points $(x_1, y_1), (x_2, y_2)$, and line $L_2$, also being defined by two distinct points $(x_3, y_3), (x_4, y_4)$.

In Figure 4(a), (b) and (c) we demonstrate common cases where the length of line needs to be calculated. In Figure 4(a) the length of the line is easily calculated since the primary line has clear starting and end points, whereas in Figure 4(b) and Figure 4(c) the length of the detected line is the $y$ point of the intersection point between the primary line and the second detected line(s). The Pixy2 camera is mounted on the center middle of the robot and the lens mounted on the Pixy2 have 0° horizontal (yaw) and 45° vertical (pitch) FoV.

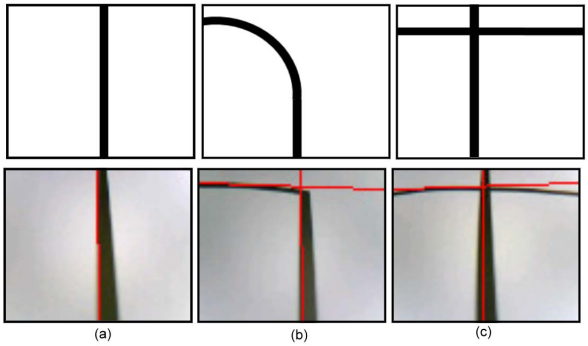


Fig. 4. Various cases of detected lines (a) and intersections (b), (c).

The second input of the fuzzy system $\mathcal{X}_2$ is related with the length of the detected line. $\mathcal{X}_2$ can be calculated as follows:

$$\mathcal{X}_2 = \frac{L_k \times 100}{|L'|} \tag{4}$$



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

where $|L'|$ refers to the maximum length of a line the camera can recognize.

A set of triangular membership functions (as illustrated in Figure 5(a) and (b)) fuzzifies each $\mathcal{X}$ score. Fuzzification is the procedure of mapping a set of crisp values into a fuzzy set, using a collection of membership functions. Sequentially, the fuzzy inputs are combined using a list of 6 rules for producing the fuzzy output of the system. The fuzzy rules are in the form of *IF-AND-THEN* and listed in Table I.

TABLE I
THE FUZZY INFERENCE RULES

| **IF** $\mathcal{X}_1$ | **AND** $\mathcal{X}_2$ | **THEN** | **IF** $\mathcal{X}_1$ | **AND** $\mathcal{X}_2$ | **THEN** |
|---|---|---|---|---|---|
| Low | Close | LC | Medium | Far | MF |
| Low | Far | LF | High | Close | HC |
| Medium | Close | MC | High | Far | HF |

The *AND* operator is implemented on the fuzzy input variables using the *MINIMUM* operation, and the results of various rules are combined producing a set of fuzzy output variables. The six triangular membership functions that form the fuzzy output are illustrated in Figure 5(c).

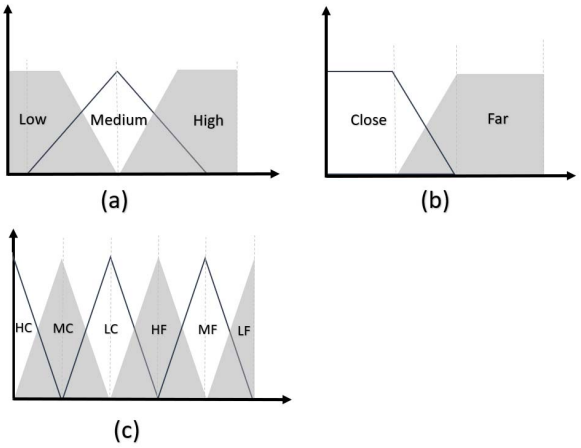


Fig. 5. (a) and (b) The fuzzy inputs of the proposed system, (c) the fuzzy output.

The final crisp output is produced using the centroid defuzzification method. Centroid defuzzification method is also known as center of gravity or center of area defuzzification and can be expressed as:

$$\mathcal{X}^* = \frac{\int \mu_i(x) x \partial x}{\int \mu_i(x) \partial x} \tag{5}$$

where, $\mathcal{X}^*$ is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and $x$ is the output variable. In general, the proposed system combines $\mathcal{X}_1$ and $\mathcal{X}_2$ values to produce a single value that affects the basic speed of the motors for each iteration. In the sequel, the calculated $\mathcal{X}^*$ value is quantized into one of the $D$, $D \in [1-100]$ unit-norm $\hat{\mathcal{X}}^*$ values: $\mathcal{X}^* = \hat{\mathcal{X}}_{\vartheta}{}^*$, where:

$$\vartheta = \underset{i}{\operatorname{argmin}} \left\{ \|\mathcal{X}^* - \hat{\mathcal{X}}_i^*\| \; \forall \; i \in [1, \ldots, D] \right\} \tag{6}$$

A priori, for each $\mathcal{X}^*$, through extensive experimentation and optimization procedures, a set of optimal $K_p$, $K_i$ and $K_d$ parameters that balance controller response time and stability have been calculated. Finally:

$$\mathscr{F}_2(\dot{\mathbf{y}}(k)) = e_1 \cdot \mathcal{X}^* \text{ while } e_1 = [1, 1, \ldots, 1]^\tau \tag{7}$$

where $|e_1| = N$.

In brief, $\mathscr{F}_2$ shapes on the one hand the basic speed of the robot at each iteration while on the other hand provides the optimal $K_p$, $K_i$ and $K_d$ parameters for $\mathscr{F}_1$. It is worth noting that, in order to avoid the problem of motor saturation, the output of the $\mathscr{F}_2$ also defines the maximum motor's speed that $\mathscr{F}_1$ can provide.

## V. EXPERIMENTAL SETUP

Experiments were conducted on a track of a black line on a white surface. The track was designed and printed on a 300 $\times$ 300 cm glossy PVC material, so it could be as slippery as possible. The total black line (path) length was 20m long. The path was selected to be as close to real-world applications, consisting of complex line structures such as acute turns, intersections, a loop and at places single or consecutive 90 degree turns.

TABLE II
EXPERIMENTAL RESULTS

| **Setup** | **Avg Speed** | **Success** |
|---|---|---|
| Classical PID Controller | 1.25 m/s | 35/100 |
| PID Controller with EDF | 1.65 m/s | 43/100 |
| Proposed Method with PID Controller | 1.55 m/s | 92/100 |
| Proposed Method with PID Controller and EDF | 1.95 m/s | 90/100 |

To evaluate the proposed methodology we constructed 4 identical systems: A traditional PID line-following algorithm navigation robot, a PID with an EDF line-following robot, the proposed method navigation robot and the proposed method navigation robot with an EDF. Table II summarizes the average results over 100 attempts for each system to track and follow the line.

Moreover, several online videos[1] demonstrate the robot in action.

The robot competed and won the Line-Following Classic (without EDF) challenge during the ROBOTEX-CY 2019 competition in the Universities category, on a 25m-long track with an ET of 15.2s and an average speed of 1.65 m/s, took a 3rd place in the Line-Following challenge (with EDF) during the International Robotics Competition ROBOTEX INTERNATIONAL 2019 in Estonia on a 35m-long track with an ET of 16.25s and an average speed of 2.15 m/s and took a 1st and 2nd place in the Line-Following Turbo challenge (with EDF)

[1] tinyurl.com/ISLRobot, youtu.be/fhhQtYSF1-U, youtu.be/6wJtP8r6enc, youtu.be/VKZHEr93wuc



Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.

during the 2020 XII INTERNATIONAL ROBOTIC ARENA in Poland on a 50m-long track with an ET of 22.2s and an average speed of 2.25 m/s.

## VI. Conclusion

This paper proposed the employment of an optimised PID control system fused in parallel to a Computer Vision system for line-following robots. The visual information greatly enhanced the ability of the PID controller to react timely in fast non-linear deviations from the setpoint. The proposed architecture orchestrates a flexible, low computationally cost and easily adaptable and IoT integratable algorithm. Applying an extensive set of experiments we demonstrate significant evidence of the effectiveness and the efficiency of the methodology. Through participation in International Robotic Competitions it is proven that the proposed controller, outperforms classical PID approaches as well as other line following control methods from the recent literature.

## References


[1] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu and T. Mei, Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID, *International Journal of Advanced Robotic Systems* **9**(2) (2012), 44.

[2] M. Dunbabin and L. Marques, Robots for environmental monitoring: Significant advancements and applications, *IEEE Robotics & Automation Magazine* **19**(1) (2012), 24–39.

[3] B. Grocholsky, J. Keller, V. Kumar and G. Pappas, Cooperative air and ground surveillance, *IEEE Robotics & Automation Magazine* **13**(3) (2006), 16–25.

[4] A. Toumpa, A. Kouris, F. Dimeas and N. Aspragathos, Control of a line following robot based on FSM estimation, *IFAC-PapersOnLine* **51**(22) (2018), 542–547.

[5] F. Cheein, C. De La Cruz, T. Bastos and R. Carelli, Slam-based cross-a-door solution approach for a robotic wheelchair, *International Journal of Advanced Robotic Systems* **7**(2) (2010), 155–164.

[6] R. Lenain, B. Thuilot, C. Cariou and P. Martinet, Model predictive control for vehicle guidance in presence of sliding: application to farm vehicles path tracking, in: *Proceedings of the 2005 IEEE international conference on robotics and automation*, IEEE, 2005, pp. 885–890.

[7] R. Choomuang and N. Afzulpurkar, Hybrid Kalman filter/fuzzy logic based position control of autonomous mobile robot, *International Journal of Advanced Robotic Systems* **2**(3) (2005), 20.

[8] W. Wang, K. Nonami and Y. Ohira, Model reference sliding mode control of small helicopter XRB based on vision, *International Journal of Advanced Robotic Systems* **5**(3) (2008), 26.

[9] Z. Li, W. Chen and H. Liu, Robust control of wheeled mobile manipulators using hybird joints, *International Journal of Advanced Robotic Systems* **5**(1) (2008), 9.

[10] D. Zhuang, F. Yu and Y. Lin, The Vehicle Directional Control Based on Fractional Order PDˆ mˆ u Controller, *JOURNAL-SHANGHAI JIAOTONG UNIVERSITY-CHINESE EDITION-* **41**(2) (2007), 0278.

[11] A. Visioli, *Practical PID control*, Springer Science & Business Media, 2006.

[12] N.C. Tsourveloudis, L. Doitsidis and K.P. Valavanis, Autonomous Navigation of Unmanned Vehicles: A Fuzzy Logic Perspective, in: *Cutting Edge Robotics*, V. Kordic, A. Lazinica and M. Merdan, eds, IntechOpen, 2005, pp. 291–310. doi:10.5772/4654.

[13] S.M. Masmoudi, N. Krichen, M. Masmoudi and D. Nabil, Fuzzy logic controllers design for omnidirectional mobile robot navigation, *Applied Soft Computing* **49** (2016), 901–919.

[14] M.A. Kader, M.Z. Islam, J. Al Rafi, M.R. Islam and F.S. Hossain, Line Following Autonomous Office Assistant Robot with PID Algorithm, in: *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, IEEE, 2018, pp. 109–114.

[15] V. Balaji, M. Balaji, M. Chandrasekaran, I. Elamvazuthi et al., Optimization of PID control for high speed line tracking robots, *Procedia Computer Science* **76** (2015), 147–154.

[16] S. Ishida and J.E. Gayko, Development, evaluation and introduction of a lane keeping assistance system, in: *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, 2004, pp. 943–944.

[17] F.-h. Jen and B.T. Mai, Building an autonomous line tracing car with PID algorithm, in: *Proceedings of the 10th World Congress on Intelligent Control and Automation*, IEEE, 2012, pp. 4478–4483.

[18] Y.-Y. Lee, Y.-T. Chung, Y.-N. Lin and S.-C. Chung, A design of fuzzy control for line following together with distance keeping on wheel robot, in: *2015 International Conference on Orange Technologies (ICOT)*, IEEE, 2015, pp. 135–138.

[19] J.-H. Li, Y.-S. Ho and J.-J. Huang, Line Tracking with Pixy Cameras on a Wheeled Robot Prototype, in: *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, IEEE, 2018, pp. 1–2.

[20] D. Nikolov, G. Zafirov, I. Stefanov, K. Nikov and S. Stefanova, Autonomous navigation and speed control for line following robot, in: *2018 IEEE XXVII International Scientific Conference Electronics-ET*, IEEE, 2018, pp. 1–4.

[21] M. Pakdaman, M.M. Sanaatiyan and M.R. Ghahroudi, A line follower robot from design to implementation: Technical issues and problems, in: *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Vol. 1, IEEE, 2010, pp. 5–9.

[22] J.G. Ziegler and N.B. Nichols, Optimum settings for automatic controllers, *trans. ASME* **64**(11) (1942).

[23] S. Bharat, A. Ganguly, R. Chatterjee, B. Basak, D.K. Sheet and A. Ganguly, A Review on Tuning Methods for PID Controller, *Asian Journal For Convergence In Technology (AJCT)* (2019).

[24] M. Trafczynski, M. Markowski, S. Alabrudzinski and K. Urbaniec, Tuning parameters of PID controllers for the operation of heat exchangers under fouling conditions, *Chemical Engineering Transactions* **52** (2016), 1237–1242.

[25] L. Zhu, W. Wang, W. Yang, Z. Pan and A. Chen, Visual Path Tracking Control for Park Scene, *CoRR* **abs/1805.01184** (2018). http://arxiv.org/abs/1805.01184.

[26] L. Juang and J. Zhang, Visual Tracking Control of Humanoid Robot, *IEEE Access* **7** (2019), 29213–29222. doi:10.1109/ACCESS.2019.2901009.




Authorized licensed use limited to: MANIPAL INSTITUTE OF TECHNOLOGY. Downloaded on February 01,2024 at 14:37:44 UTC from IEEE Xplore. Restrictions apply.