Visual servoing based control methods for nonholonomic mobile robot

Mahmut Dirik, Adnan Fatih Kocamaz and Emrah Dönmez*

**Department of Computer Engineering, Faculty of Engineering, Inönü University, 4400, Malatya, Turkey*
**Corresponding Author: mahmut.dirik@inonu.edu.tr*

Submitted: 05/02/2018
Revised: 19/09/2018
Accepted: 23/09/2019

ABSTRACT

In this paper, we utilized two different vision-based go-to-goal robot control approaches on indoor nonholonomic mobile robot systems. In the proposed methods, eye-out-device configured camera (overhead camera) image data are used as the input parameters to determine the speeds of robot wheels. The main purpose of this system is to minimize the complexity of conventional robot control kinematics and to provide an efficient control approach to manage the wheel speeds and the direction angle of the mobile robot. In addition to reducing the complexity of robot control kinematics, it is also intended to reduce systematic and nonsystematic errors. The proposed method is divided into three stages: the first stage consists of the overhead camera calibration and the configuration of the robot motion environment. At this stage, the labels placed on the robot and target position were identified and the position information of the robot was obtained. In the second stage, control inputs such as position and orientation based on robot motion tracking and visual feature information were obtained. In the third stage, Graph-based Gaussian and Angle-based Decision tree control approaches were performed. We have briefly described these control approaches as follows: Graph-based Decision Tree Control (GDTC), Graph-based Gaussian Control (GGC), Angle-based Decision Tree Control (ADTC), and Angle-based Gaussian Control (AGC). Using these control approaches, many real-time experimental studies with eye-catching device configuration have been performed. The efficacy and usability of the methods have been demonstrated by experimental results.

Keywords: decision tree control; gaussian control; non-holonomic mobile robot; overhead camera; vision-based control.

HIGHLIGHTS

- With this study, new control approaches to traditional mobile robot control algorithms have been developed.
- It is intended to minimize the required hardware equipment or additional components in real-time applications and intended to reduce systematic and nonsystematic errors.
- To control the motion of a nonholonomic mobile robot, vision-based robot control platform using GDTC, GGC, ADTC, and AGC controllers has been developed.
- A new framework for the evolutionary algorithm.
- Finally, we combine the advantage of the traditional mobile robot control algorithm with the proposed technique to improve computational efficiency.

INTRODUCTION

During the last two decades, mobile robots have been widely used in a number of sectors like industry, service, military, and so on. Visual-based control strategy has significant doctrines in order to accomplish tasks such as mobile robot tracking (Keshmiri, Xie, and Mohebbi 2014) (Li et al. 2016) (Farah 2017) (Zou et al. 2016), stabilization (Hu et al. 2010) (Parikh et al. 2015), path planning (Dao, Pan, and Pan 2016) (Jiang et al. 2016) (Ziaei, Oftadeh, and Mattila 2015a), and navigation control (Ziaei, Oftadeh, and Mattila 2014) (Engel, Sturm, and Cremers 2012) (Chugo et al. 2012) (Lippiello et al. 2016). In robot control systems, accuracy, speed, sensitivity, efficiency, and cost are the most significant factors (Ziaei, Oftadeh, and Mattila 2014) (Siciliano and Khatib 2007). The visual-based control method provides accurate position information compared to conventional methods that use mainly onboard sensors such as wheel encoders and other measurement units. The conventional methods cannot accurately measure robot positions and directions due to the limited sensors capacities and external environments defects.

Different methods have been used to control the robot motion by utilizing the different type of sensor data (Tsai and Song 2009). Recently, the visual-based control (VBC) for robot motion control draws attention in real-time applications. There are a significant number of works focusing on VBC for the wheeled mobile robot (WMR) control (Tsai and Song 2009) (Cherubini and Chaumette 2013) (Dirik, Kocamaz, and Dönmez 2016a) (Dönmez, E., Kocamaz and Dirik 2017). Multi-directional visual-based feedback control scheme with Jacobean Matrix Estimator (JME) control application on robot manipulators has been recommended in several studies (Hutchinson, Hager, and Corke 1996). In (Mahon et al. 2008), the visual feedback control loop is performed and emerged error rate of trajectories by tracking robot from a single camera has been calculated. It was specified that the system has been operated with potential field method and the general parameters are the quality of knowing shape and measurements of an object and quality of camera calibration. A fish-eye lens camera was used to detect physically labeled objects by the threshold method in (Chen and Lee 2011). A multi-agent system was developed to provide location detection of mini-mobile robots simultaneously by tracking colored LEDs placed on robots with cameras in (Mahon et al. 2008). It has been claimed that a new Gaussian-based controller method with visual servoing has been developed in (Dönmez, Kocamaz, and Dirik 2016). They used both color thresholding and Hough transform to improve the object detection process in sequentially acquired images. Brightness level (or light intensity) has been also specified as a major problem for object detection process in this study. In the context of VBC, mobile robot control simulations were carried out using different methods. The mobile robot motion control was carried out by detecting all of the objects extracted from the background and foreground in (Zhang et al. 2017). A global control model has been implemented using a CAD-based method of motion control and tracking of the mobile robot with a single omnidirectional overhead camera in (Ziaei, Oftadeh, and Mattila 2015a) (Ziaei, Oftadeh, and Mattila 2014). Robot control method based on visual servoing is used as an alternative and supportive method in (Elsheikh, El-Bardini, and Fkirin 2016) (Ziaei, Oftadeh, and Mattila 2015b) (Zhang et al. 2017). It is claimed that this control method is robust and powerful. The VBC simply controls a robot by processing detected object position images taken from the camera as guide points for the control process. In the utilized VBC, feature extraction has been implemented for detection and tracking of the objects initially. Then, the robot and target positions have been determined, the robot steering has been implemented, and the feedback control loop beenis kept up-to-date. The main contribution of this work is that it combines the visual servoing control loop based on the angle and graph-based Gaussian and decision tree method for controlling a nonholonomic mobile robot. Several experiments have been implemented and the results have shown the efficiency of the proposed methods.

This paper is organized as follows. First, we introduce the problem definition and preliminaries information, and then we present the details of the vision-based control approach and mobile robot navigation process. The following section introduces the proposed kinematic models for the nonholonomic mobile robot navigation system, and the experiments and observations are reported in the next section. In the section that follows, a discussion of experimental results is presented and in the last section, the results of the study and future works are presented.

PROBLEM DEFINITION

The general problems in robot navigation are motion control, localization, and path planning. There are remarkable numbers of methods and algorithms used to address such problems. In this study, we have especially focused on localization and motion control tasks. We used the new control approaches GDTC, GGC, ADTC, and AGC, which use visual information as input parameters. These are the basic algorithms used to determine and update the desired input parameters according to the robot and target positions. In this work, the imaging sensor tracks the labeled mark sets on the robot and target by using color-based thresholding method on acquired image frames. The control is performed on an external computer and only commands for the speed of the wheels are processed and sent. Therefore, the internal processing unit is not required. So, both the cost and the energy gain are achieved. To increase the detection robustness, camera calibration and color-based quantization are utilized. These processes are necessary to get better performance in real-time tracking of the robot, target, and other objects (Dönmez, Kocamaz, and Dirik 2016) (Bernard Espiau and Radu Horaud, 1998) (Dirik, Kocamaz, and Dönmez 2016b) (Dirik, Kocamaz, and Dönmez 2017a). Robot position and direction information are obtained from the robot motion environment using the overhead camera. The controller input values are acquired by using this information. The obtained inputs are transferred to Gauss and Decision tree controller and desired output values (wheel speeds, direction angle) of the robot are acquired. Both angle and graph information are extracted from the input image frames as shown in Figure 1. By using these values, the robot performs the go-to-goal task. This control process continues synchronously until the robot reaches the target position. The general structure of the proposed system is given in Figure 1.

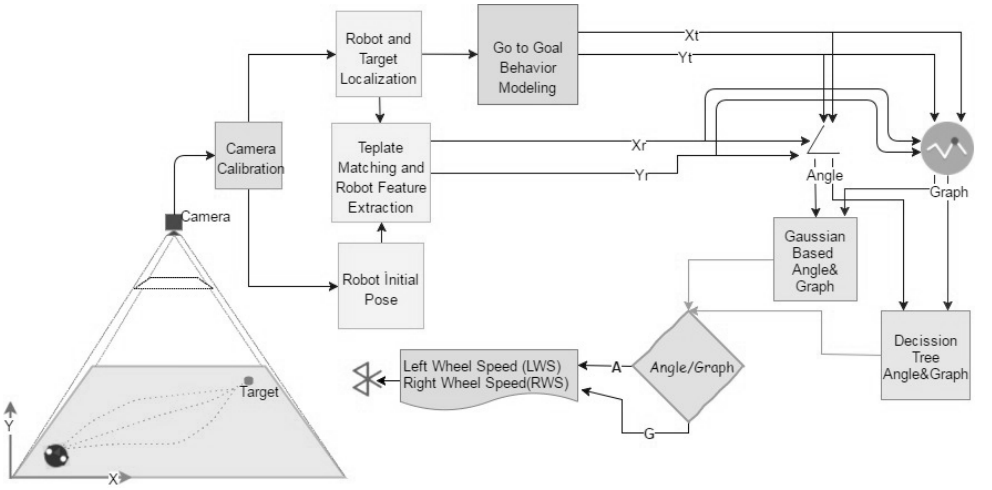


Figure 1. Overhead camera-based visual system. Xt, Yt: target position, Xr, Yr: Robot current position, Angle: left and right virtual triangle base angles, Graph: Virtual triangle graph edge length value.

VISION-BASED NAVIGATION PROCESS

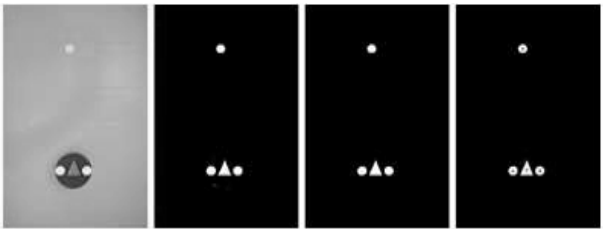
In this section, the details of the vision-based control approach for a nonholonomic mobile robot have been given. The used overhead camera has been placed vertical to the ceiling and its height is about 180 cm. The light intensity can be a major problem in the configuration space. The solutions related to the problems faced by the system in case of change of light intensity, brightness change or more details related to the brightness issue have been discussed in previously implemented studies (Dönmez, Kocamaz, and Dirik 2017a) (Dirik, Kocamaz, and Dönmez 2016b). The communication between the mobile robot and the computer has been performed with Bluetooth interface. We have used the Gaussian and decision tree methods as controllers.

Detecting and Tracking Process

Object detection is performed with color thresholding in image space. This task has been done under variable light conditions and it was found that the HSV (Hue-Saturation-Value) color space is better than RGB (Red-Green-Blue) color space for filtering. HSV color filter is operated to detect the boundaries of the objects (mobile robot and target) in image space. The object position coordinates have been obtained. Then, the control parameters have been calculated and the robot motion control has been triggered. Equation 1 was utilized for this purpose.

$$f(p_k) = \frac{1}{M}\sum_{i=1}^{M} h(p_i)K_\sigma(p_k - p_i), \qquad p\epsilon\{H,S,V\} \tag{1}$$

where $h(p_i)$ corresponds to the histogram of color channels. p_k indicates k^{th} level of channel, M represents the total level of the channel, and K_σ represents the standard deviation factor. The detection process is illustrated in Figure 2. This process is executed to detect centroids of objects; coordinates are marked on these objects centroids (see Figure 2).



a) b) c) d)

Figure 2. Detection process: a) Real time image frame; b) object extraction; c) filtered image; d) object centroids illustration.

The algorithm uses color-based masks to distinguish the robot and target objects based on the predefined threshold. The environment luminance can be varying in the daytime; for this reason, encountering errors are possible while color filtering is applied. To overcome this problem, additional methods are needed. This supportive method is color quantization in which image colors are separated by measuring levels in Q as $q_j, j = 1,\ldots,n$, quantization cells. Equation 2 and Equation 3 are used for this purpose.

$$E(d,q) = \int_c d\left(x, q(x)\right) d\mu = \sum \int_{c_j} d\left(x, q(x)\right) d\mu \geq \sum \int_{c_j} d(x, q_j) d\mu \tag{2}$$

$$q_j = \arg\min_{c\in C, x\in C_j} E(d(x,c)) = \arg\min_{c\in C} \int_{C_j} d(x,c) d\mu \tag{3}$$

In these equations, 'd' and 'q' are distortion and quantizer parameters, respectively.

Positioning Models

In order to demonstrate that the proposed control approaches can be applied to position tracking and motion control, an overhead camera has been used. It has provided smooth feedbacks for closed-loop control. The control system continuously scans the working environment for object detection. The center coordinates of each object are obtained from the input image frame. These coordinates are connected to each other to obtain a positioning scheme. They are used to create nodes and edges for graph-based model and to create angles for triangle model (see Figure 3 and Figure 4). This approach is used to control the robot position and orientation. Detailed description can be seen in previously performed studies (Dönmez, Kocamaz, and Dirik 2017c) (Dirik, Kocamaz, and Dönmez 2016c)

Positioning Model with Graph Form

In order to control the mobile robot with a graph-based model, it is first necessary to find the coordinates of the nodal points. A triangle graph structure is created between the nodes. The input parameters of the proposed control approach have been obtained by using the edge values of this triangle. The positioning scheme of graph-based model is shown graphically in Figure 3(a). The detected objects and calculated parameters are shown as a sample in Figure 3(b). This scheme is performed for graph-based vision control structure.

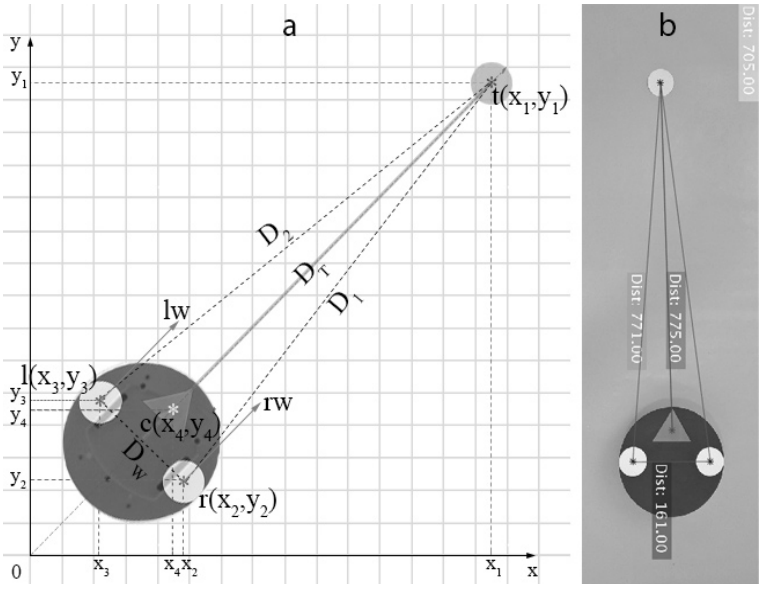


Figure 3 (a). Distance-based positioning scheme; (b) A real image frame.

The "c, l, r, t" points are the graph node centers; 'c' represents the red triangle (robot center) label, 'l' and 'r' represent the left and right wheel labels, and 't' corresponds to the center coordinates of the target position. D_1, D_2, D_W and D_T indicate the weight of the edges between nodes. The distance between the robot and the target position is indicated by D_T. The distance between the nodes is obtained by the Euclidean distance calculation given in Equation 4

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \qquad (4)$$

The output distance values acquired from this equation are used as input to mobile robot control. D_1 and D_2 are used as input parameters in the graph-based model. The orientation of the robot depends on the differentiation of D_1 and D_2. The node coordinates change according to the robot position in sequential image frames. The right and left wheel speeds are determined in real-time for each update task by using Equation 4.

Positioning Model with Triangle Form

To extract robot position information, 'c' label (red) has been utilized. The other two 'l' and 'r' (white) labels have been used to locate the left and right wheels of the WMR. A trigonometric triangle is obtained using coordinate values of all these labels. The input parameters of the proposed control approach have been obtained by using the inner angle values of this triangle. The triangular-based positioning model diagram with robot, target, and other parameters is illustrated in Figure 4(a). This scheme is performed for angle-based vision control approaches.

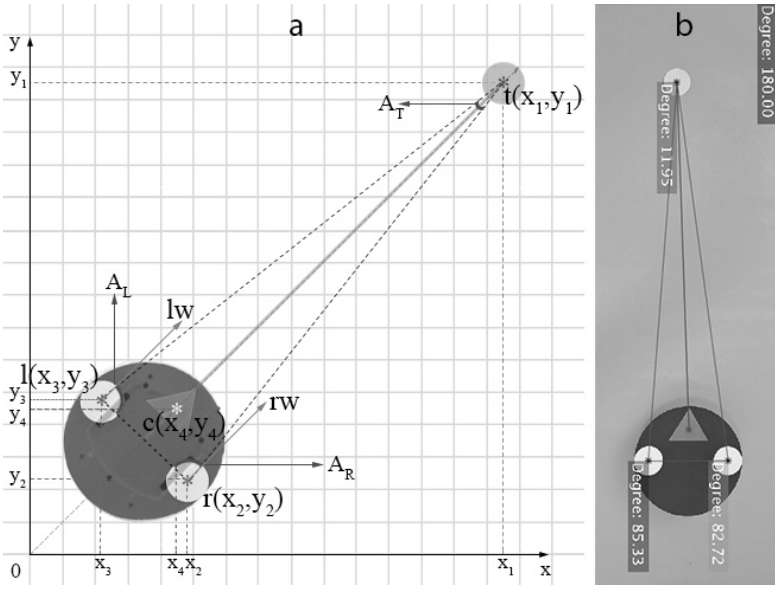


Figure 4 (a). Triangle-based positioning model system; (b) A real image frame

The triangle is obtained by joining the created edges as seen in the figure. Interior angles are the input parameters of the control model approach. 'A_L, $A_R A_T$' denoted the left wheel, right wheel, and target angles, respectively. To regulate input parameters for the controller, acquired angle data are compared dynamically. Whenever the robot moves, the angle values of tracked objects are updated periodically by using the coordinate values in the 2D (x-y) space. Equation 5 is used to calculate interior angle values and Equation 6 is used for transmitting the $\mathrm{ang_r}$ parameter to the degree value.

$$\mathrm{ang_r} = \mathrm{acos}\left(\frac{\Delta_x^2 - \Delta_y^2 - \Delta_z^2}{-2 * \Delta_y * \Delta_z}\right) \tag{5}$$

$$\mathrm{A_{360}} = \mathrm{ang_r} * \frac{180}{\mathrm{pi}} \tag{6}$$

The Δ_x, Δ_y and Δ_z are distance vectors between centroid coordinates of detected objects. The '$\mathbf{ang_r}$' is the radian value of the related angle and '$\mathbf{A_{360}}$' is the conversion of this radian value to the degree value. Velocity values for both wheels are computed according to these angle values for the triangle-based approach.

KINEMATIC MODELS AND MOBILE ROBOT NAVIGATION

In this study, an efficient control approach based on angle/graph in the closed-loop control for mobile robot kinematic model is emphasized. The control parameters and robot positioning scheme deal with the mobile robot kinematics. The proposed kinematic model contains different calculations, unlike traditional mobile robot kinematics. For adjustment of wheel speed of the robot that we used in this study and to convert the PWM (Pulse Width Modulation) units to the metric units, Equation 7 is used.

$$Sp\left[\frac{mm}{s}\right] = \frac{v_p}{t_R} * \frac{\emptyset_w * \pi}{Nb_p} \tag{7}$$

The Sp is left/right wheel velocity of the robot. v_p represents pulse value. t_R represents refresh time. $\emptyset_w$ represents the diameter of the wheel. Nb_p represents the resolution of a cycle.

Kinematic Model of Gaussian Method

The Gaussian (normal) distribution that has a number of mathematically advantageous characteristics is used across engineering disciplines. The single-dimensional general Gaussian function is given in Equation 8. This equation is distributed according to the parameters of the mean (μ) and standard deviation (σ). The input parameter (**x**) in this equation is the difference between input parameters ($\mathbf{D_1 - D_2}$ or $\mathbf{A_L - A_R}$). The input parameters are the difference between the angle values or the graph weight (edge length) values as previously described.

$$f_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{8}$$

$$x_D = \left|\frac{D_1 - D_2}{\lambda}\right| \tag{9}$$

$$x_A = \left|\frac{A_L - A_R}{\varphi}\right| \tag{10}$$

$$S_C = S_{max} * (1 - f_G) \tag{11}$$

$$G_C = V_{max} * (1 - f_G) \tag{12}$$

The control parameters x_D and x_A represent the absolute difference value for graph and angle approaches given in Equations 9 and 10. The λ and φ values are used as smoothing factors for these difference values. To calculate the effect of the Gaussian function on wheel speeds, Equation 11 was used. The S_C is speed coefficient, S_{max} is the maximum speed of wheel, and f_G is Gaussian function. The x_D and x_A are affected on S_C as seen in the equations. Left and right wheel speeds are determined by using the values obtained from the described equations (8-12); the following equations characterize the motion parameters.

Graph-Based Gaussian Control (GGC)	**Angle-Based Gaussian Control (AGC)**	
$\mathbf{S_L} = \begin{cases} \mathbf{S_{max} * \tau + S_C}, & \mathbf{D_1 < D_2} \\ \mathbf{S_{max} * \tau - S_C}, & \mathbf{D_1 > D_2} \end{cases}$	$V_R = \begin{cases} V_{max} * \gamma + G_C, & A_R < A_L \\ V_{max} * \gamma - G_C, & A_R > A_L \end{cases}$	(13)
$\mathbf{S_R} = \begin{cases} \mathbf{S_{max} * \tau + S_C}, & \mathbf{D_1 > D_2} \\ \mathbf{S_{max} * \tau - S_C}, & \mathbf{D_1 < D_2} \end{cases}$	$V_L = \begin{cases} V_{max} * \gamma + G_C, & A_R < A_L \\ V_{max} * \gamma - G_C, & A_R > A_L \end{cases}$	(14)
$\mathbf{S_{L,R} = S_{max} * \tau + S_C}, \ \mathbf{D_1 \cong D_2}$	$V_{R,L} = V_{max} * \gamma + G_C, \quad Thr \leq A_T$ $V_L = 0 \ \& \ V_R = 0 \text{ iff } A_T \geq 60$	(15)

In these equations, S_L, S_R, V_L and V_R are Current speed of left/right wheels of the mobile robot. The 'τ' and γ are used as a constant scaling factors. The calculation given in these equations is carried out according to the edge weight of the virtual triangle formed with the graph base. In this way, each speed of the wheels is weighted according to the model.

Kinematic Model of Decision Tree Method

A decision tree is an important tool for determining the most suitable output from a range of candidate possibility. This technique is an appropriate solution to automate a decision procedure. It can be applied to many engineering

fields. In the proposed method, we used the decision tree algorithm as a control function to adjust the mobile robot wheel speeds. The controller uses both angle and distance input parameters. This structure is shown in Figure 5.

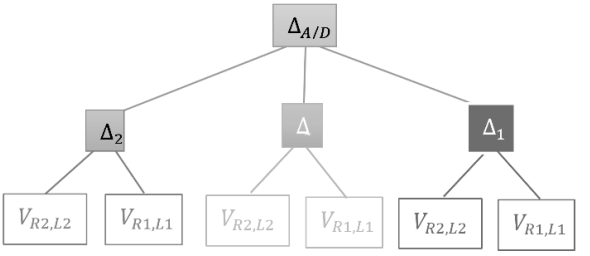


Figure 5. Decision tree structure designed for robot control.

The $\Delta_{A/D}$, given in this structure, has represented the input value for both angle and graph-based control approaches. There are several equations that have been used for these approaches. These equations are given in (16-20). In these equations, $\Delta_{A/D}$ symbolize the $(D_1 - D_2)$ or $(A_R - A_L)$ difference value as an input parameter. WMR speed (velocity parameters of the wheels) parameters are determined according to the decision tree by using Equation 16.

Graph-Based Decision Tree Control (GDTC)	**Angle-Based Decision Tree Control (ADTC)**	
$\mathbf{V_R, V_L} = \begin{cases} \Delta_1, & \mathbf{D_1 > D_2} \\ \Delta_2, & \mathbf{D_1 < D_2} \\ \Delta\ , & \mathbf{D_1 \cong D_2} \end{cases}$	$V_R, V_L = \begin{cases} \Delta_1, & A_R > A_L \\ \Delta_2, & A_R < A_L \\ \Delta\ , & A_R \cong A_L \end{cases}$	(16)
$\Delta_{Vm} < \lvert\Delta_{A/D}\rvert < \Delta_{Vn}\ ?\ \boldsymbol{\delta = \delta_n}\ :\ \boldsymbol{\delta = NIC}$	$V_{R,L} = V_{max} * \tau + G_C, \quad Thr \leq A_T$	(17)
$\mathbf{V_L = V_{max} * \frac{\Delta_t}{\Delta_1} * \delta}, \quad \Delta_w < \Delta_t$	$V_L = (V_{max} + A_R) * \gamma \mp \sqrt{(D_i + 1) * \frac{1}{A_T + 1}}$	(18)
$\mathbf{V_R = V_{max} * \frac{\Delta_t}{\Delta_2} * \beta}, \quad \Delta_w < \Delta_t$	$V_R = (V_{max} + A_L) * \varepsilon \mp \sqrt{(D_i + 1) * \frac{1}{A_T + 1}}$	(19)
$\mathbf{V_L = 0\ \&\ V_R = 0\ iff\ \Delta_w - 30 < \Delta_t}$	$V_L = 0\ \&\ V_R = 0\ \text{iff}\ A_T \geq 60$	(20)

In Graph-based approach, Δ_{Vm} and Δ_{Vn} are the lower and upper pre-defined limit values used to determine δ smoothing factor value. If the $\Delta_{A/D}$ difference value is found between predefined interval values, then δ takes a new δ_n value; if $\Delta_{A/D}$ difference value is not in these intervals, then next interval condition (NIC) is controlled whether δ is in the checked interval. Δ_t is target node, V_{max} is maximum robot wheel speed (mm/s), Δ_W is threshold value, and V_L and V_R are denoted as the left and right speeds.

In the angle-based approach, the Thr is the specified threshold value. It is the stable parameter value found for the kinematic method after several experiments. The parameters used in the equations developed for the angle-based control approach are briefly described as follows. γ and ε are used as smoothing factors. γ affects V_L negatively and ε affects V_R positively if the sign of D_i (that is, difference value of $A_L - A_R$) is positive. If the sign of D_i is negative, then ε affects V_L and γ affects V_R, and vice versa. In the equation, the expression D_i is actually $\Delta_{A/D}$. The A_T value is the threshold for stopping the process of robot. When this condition is not fulfilled, the wheel velocity parameters are calculated and sent to the wheels as previously stated. The real-time implementation of the graph-based and angle-based decision tree kinematic control approaches is discussed and the results are given in the next section.

EXPERIMENTS AND OBSERVATIONS

In this section, experimental results are clarified. The vision-based nonholonomic mobile robot control has been presented. The experimental environment consists of a fixed overhead camera, mobile robot, and Bluetooth communication module. The markers on both the robot and target are used to extract concerned position information in the reference image. The distance and angle between the mobile robot and the target have been obtained to calculate controller inputs. Based on these control inputs, the velocities of the left and right wheels of the mobile robot have been produced and GDTC, GGC, ADTC, and AGC control algorithms are used to compute the actual velocities of the wheels. The initial velocities and positions of the mobile robot are given in Table 1 and Table 2. The overall control algorithms have been physically implemented and tested. The efficiency and robustness of the proposed methods are demonstrated.

Table 1. Reference parameters and position parameters of angle-based vision control approaches.

		Initial velocity of wheels (PWM)		**Initial Angles of Virtual Triangle (Deg^o)**			**Initial Robot Posture (m, m, Deg^o)**		
Methods	Angles	V_L	V_R	A_L	A_R	A_T	x	y	Ø
Angle-Based Gaussian Control (AGC).	***0^o***	85.93	94.06	87.32	84.19	8.4	654	316	0
	45^o	73.76	106.23	123.70	44.07	6.23	646	303	45
	90^o	39.26	140.73	180	0	0	671	313	90
	180^o	-80	80	86.6	85.02	8.36	710	324	180
Angle-Based Decision Tree Control (ADTC)	***0^o***	82.98	82.74	84.93	86.52	8.5	627	316	0
	45^o	42.8	139.82	130.82	43.02	6.15	637	295	45
	90^o	-80	80	180	0	0	659	287	90
	180^o	-180	80	87.23	84.01	8.6	684	318	180

Table 2. Reference parameters and position parameters of graph-based vision control approaches.

		Initial velocity of wheels (PWM)		**Initial lengths of the virtual triangle edge (px)**				**Initial Robot Posture (m, m, Deg^o)**		
Methods	Angles	V_L	V_R	D_1	D_2	D_T	D_W	x	y	Ø
Graph-Based Gaussian Control (GGC)	***0^o***	74.85	75.17	482.0	480.0	451.0	72.0	627	316	0
	45^o	69.25	125.36	507.0	457.0	462.0	71.0	637	297	45
	90^o	-80	80	515.0	445.0	482.0	70.0	657	288	90
	180^o	-80	80	481.0	481.0	508.0	72.0	682	320	180
Graph-Based Decision Tree Control (GDTC)	***0^o***	85.9	94.05	494.0	493.0	463.0	73.0	652	323	0
	45^o	70.79	109.2	514.0	466.0	469.0	71.0	659	302	45
	90^o	-80	80	520.0	450.0	486.0	70.0	677	294	90
	180^o	-80	80	491.0	489.0	517.0	73.0	707	324	180

Experiments have been implemented in various scenarios as indicated in Table 1 and Table 2. The experiment results that are frames per second, elapsed time, the emerged path cost, and path cost rate have been given in Table 3 and Table 4. It was observed that Gaussian-based proposed method is faster than the decision tree method.

Table 3. Path cost information with angle positioning.

Methods	Angles	FPS	Elapsed Time(s)	Path Cost (px)	Path Cost Rate
Angle-Based Gaussian Control (AGC).	***0º***	14.23	8.127	426.183	4.457%
	45º	14.47	8.422	429.479	5.264%
	90º	14.66	8.955	438.112	7.380%
	180º	14.12	9.423	507.214	24.317%
Angle-Based Decision Tree Control (ADTC)	***0º***	13.99	7.312	420.635	3.097%
	45º	14.08	7.556	422.710	3.605%
	90º	14.22	7.854	434.303	6.450%
	180º	14.10	8.660	486.401	19.216%

Table 4. Path cost information with graph positioning.

Methods	Angles	FPS	Elapsed Time(s)	Path Cost (px)	Path Cost Rate
Graph-Based Gaussian Control (GGC)	***0º***	15.88	7.459	422.071	3.449%
	45º	15.80	7.662	424.282	3.991%
	90º	15.35	8.003	431.112	5.665%
	180º	15.64	8.965	484.810	18.830%
Graph-Based Decision Tree Control (GDTC)	***0º***	15.12	6.508	409.426	0.349%
	45º	15.89	6.822	411.544	0.869%
	90º	15.54	7.210	418.610	2.600%
	180º	15.36	7.772	474.568	13.533%

The real working experiment is shown in Figure 6 and Figure 7. These are example frames from the experiments of robot initial position in '0º' and '180º' angles under simple and extreme conditions, respectively. The marks colored in bold red paths indicate the robot path trajectories and the black one is the power cord of the robot. It can be said that the performance of the methods is satisfactory.

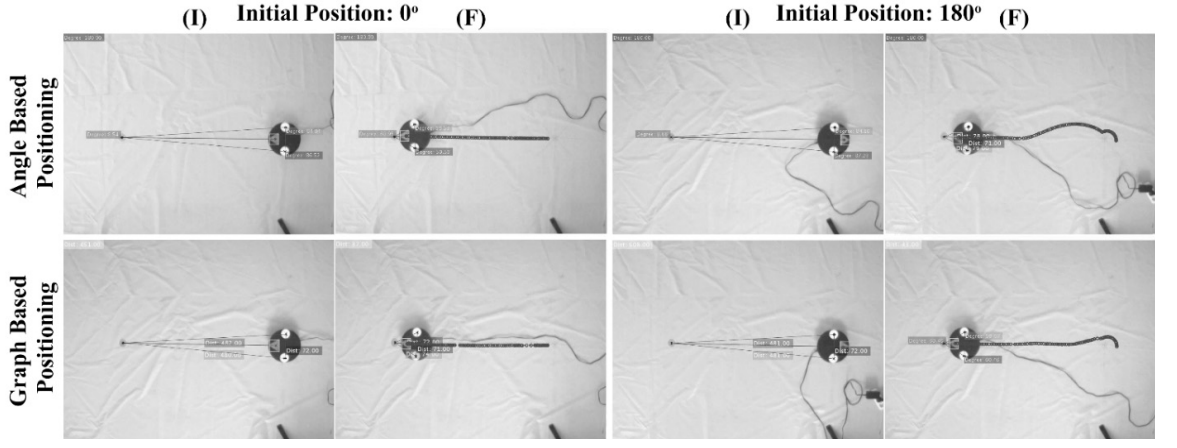


Figure 6. Extracted path with Decision Tree control for both positioning models, (I): Initial state - (F): Final state.

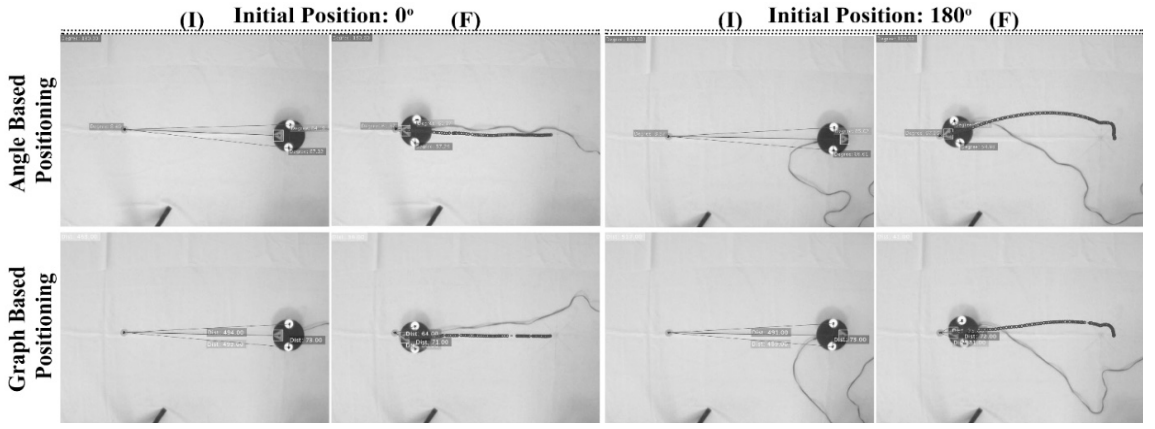


Figure 7. Extracted path with Gaussian control for both positioning models, (I): Initial state - (F): Final state.

The previously introduced equations are used to create the output values given in the tables. The magnitude of the wheel speeds of the mobile robot varies with the angle direction of the robot w.r.t the target. The graphical representations of the changes in angle values, distance values, and robot positions according to their initial values are illustrated. The comparative results prove the efficiency of the methods.

ANGLE-BASED CONTROL APPROACHES

In this section, experimental results obtained with AGC and ADTC are demonstrated (Figure 8 and Figure 9). The rows of the table show initial direction values (as angle) according to the robot position. We have performed four different experiments with distinct initial directions. However, only two directions are given in the figures. These two directions represent simple and difficult position conditions. The columns of the table show the input, output, and position parameter changes against the FPS. The controller input (A_L, A_T, A_R), output (V_R , V_L), and position (x, y) parameter changes have been given against the elapsed time in 2D configuration space. The controller input parameters have converged to each other at the end of the given task. The controller output parameters have been generated by control functions according to inputs to manage the driving of the mobile robot. The position of the mobile robot (Cartesian coordinates of the robot in 2D configuration space) has changed according to the position of the target.

Angle-Based Gaussian Control (AGC)

In this section, experimental results obtained with AGC are demonstrated in Figure 8. The controller inputs (A_L, A_T, A_R), outputs (V_R , V_L), and positions (x, y) parameter changes have been given against the elapsed time in 2D configuration space. The abbreviations used herein are as follows. In column a), angles that vary depending on the processed image frames are represented as Left (A_L), Right (A_R), and Target (A_T). In column b), the changes in robot wheel speeds depending on the processed image frames are represented as Right speed (V_R), Left speed (V_L). Column c) represents the changes in robot coordinates (x, y) depending on processed image frames.

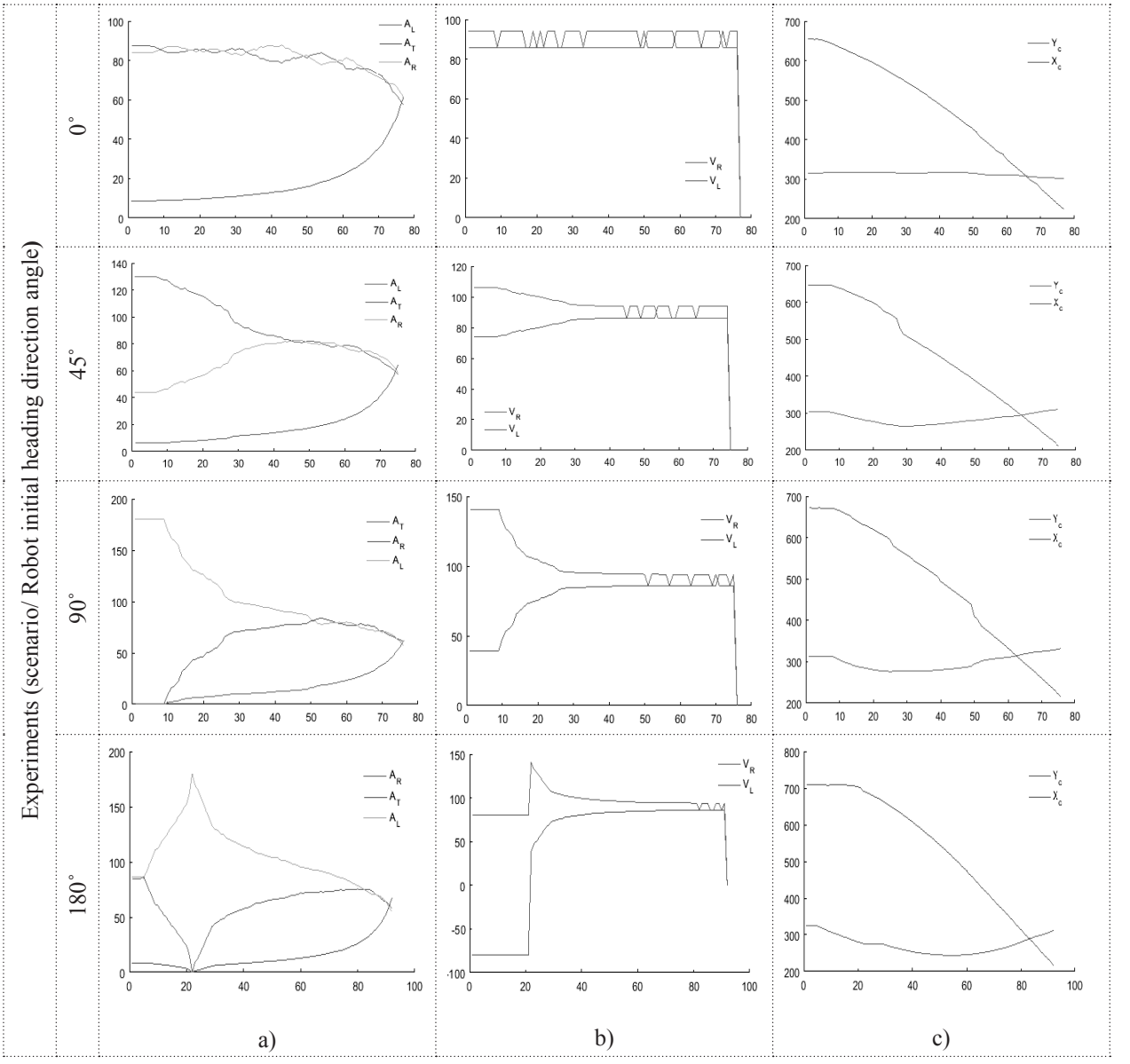


Figure 8. The performance of experimental results using the Angle-Based Gaussian controller (AGC). X-axis of all graphs shows the number of frames processed. Legend: A_L, left angle; A_R, right angle; A_T, target angle; V_L, left speed; V_R, right speed; X_C, robot center x coordinate; Y_C, robot center y coordinate.

Angle-Based Decision Tree Control (ADTC)

Experimental results obtained with ADTC are demonstrated in Figure 9. The controller inputs (A_L, A_T, A_R), outputs (V_R, V_L), and position (x, y) parameter changes have been given against the elapsed time in 2D configuration space. The abbreviations used herein are as follows. In column a), angles that vary depending on the processed image frames are represented as Left (A_L), Right (A_R), and Target (A_T). In column b), the changes in robot wheel speeds depending on the processed image frames are represented as Right speed (V_R), Left speed (V_L). Column c) represents the changes in robot coordinates (x, y) depending on processed image frames.

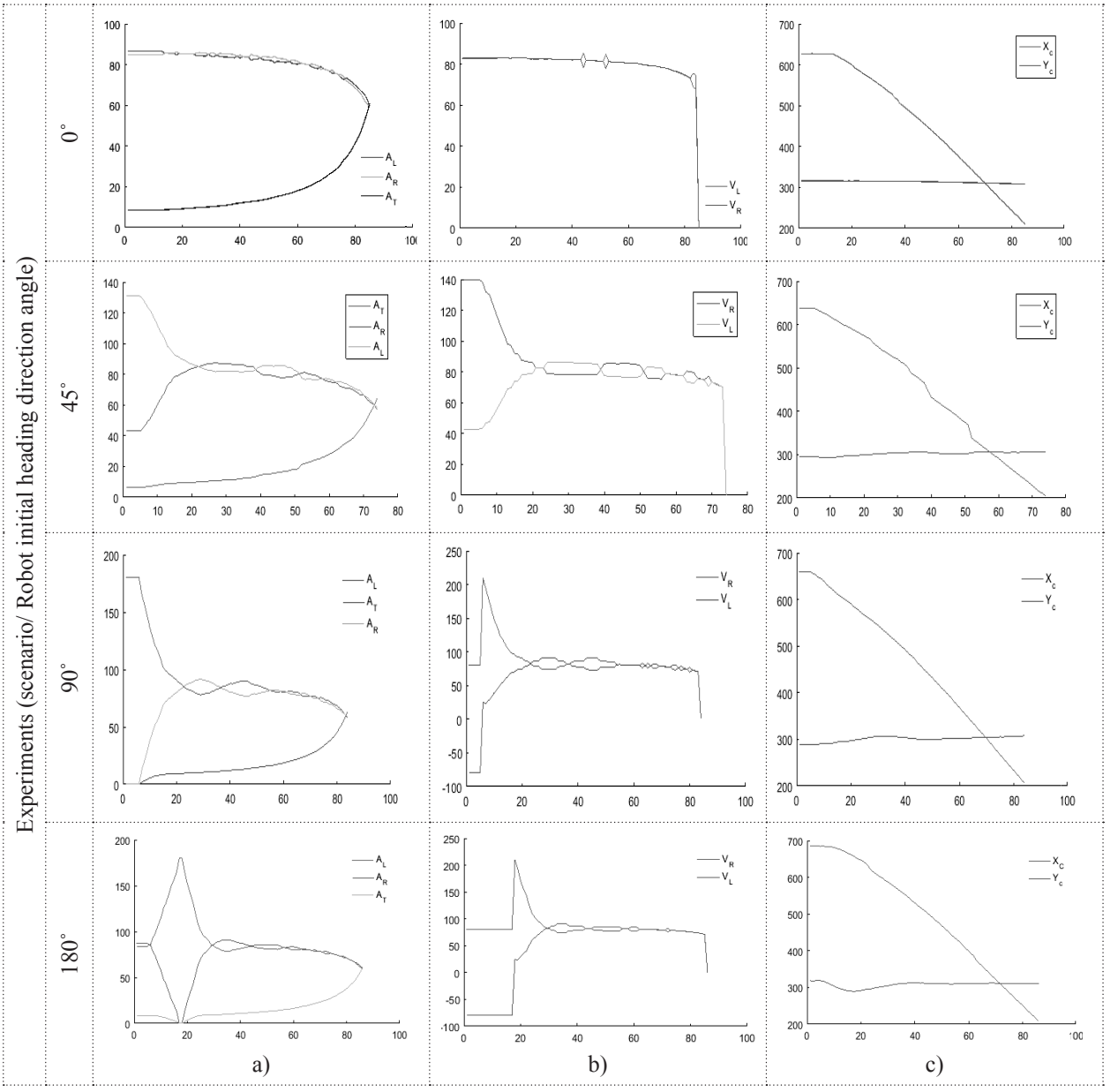


Figure 9. The performance of experimental results using the Angle-Based Decision Tree controller (ADTC). X-axis of all graphs shows the number of frames processed. Legend: A_L, left angle; A_R, right angle; A_T, target angle; VL, left speed; V_R, right speed; X_C, robot center x coordinate; Y_C, robot center y coordinate.

It is possible to see that the proposed method is successful from the experimental result of the graphical representation given in Figure 8 and Figure 9. As the robot approaches the target, the shape of the equilibrium is formed between the centers of the labels above the robot and the center coordinates of the target label. This means that the robot has reached the target. The change in the speed of the robot wheel in the process of the formation of this equilateral triangle depending on the processed image frames is also measured for the accuracy of the proposed method.

GRAPH-BASED CONTROL APPROACHES

The performances of GDTC and GGC experimental results are demonstrated in Figure 10 and Figure 11. The rows of the table show initial direction values (as the angle) according to the robot position. We have performed four different experiments with distinct initial directions. However, only two directions are given in the figures. These two directions represent simple and difficult position conditions. The columns of the table show the input, output, and position parameter changes against the FPS. The changes of the input controls (D_1, D_2, D_T, D_W), outputs (V_R, V_L), and position (x, y) parameters have been given against the elapsed time in 2D space configuration. The controller input parameters have converged to the D_W threshold value at the end of the given task. The controller output parameters have been generated by control functions according to inputs to manage the driving of the mobile robot. The position of the mobile robot (Cartesian coordinates of the robot in 2D configuration space) has changed according to the position of the target.

Graph-Based Gaussian Control (GGC)

The performances of GGC experimental results are demonstrated in Figure 10. The changes of the input controls (D_1, D_2, D_T, D_W), outputs (V_R, V_L), and position (x, y) parameters have been given against the elapsed time in 2D space configuration. The abbreviations used herein are as follows. In column a), edge lengths that change in between the robot and the target are represented as Right (D_1) and Left (D_2). (D_W) represents the base distances between label centers on the robot. Column b) shows the changes in robot wheel speeds Right speed (V_R), Left speed (V_L) depending on the processed image frames. Column c) represents the changes in robot coordinates (x, y) depending on processed image frames.

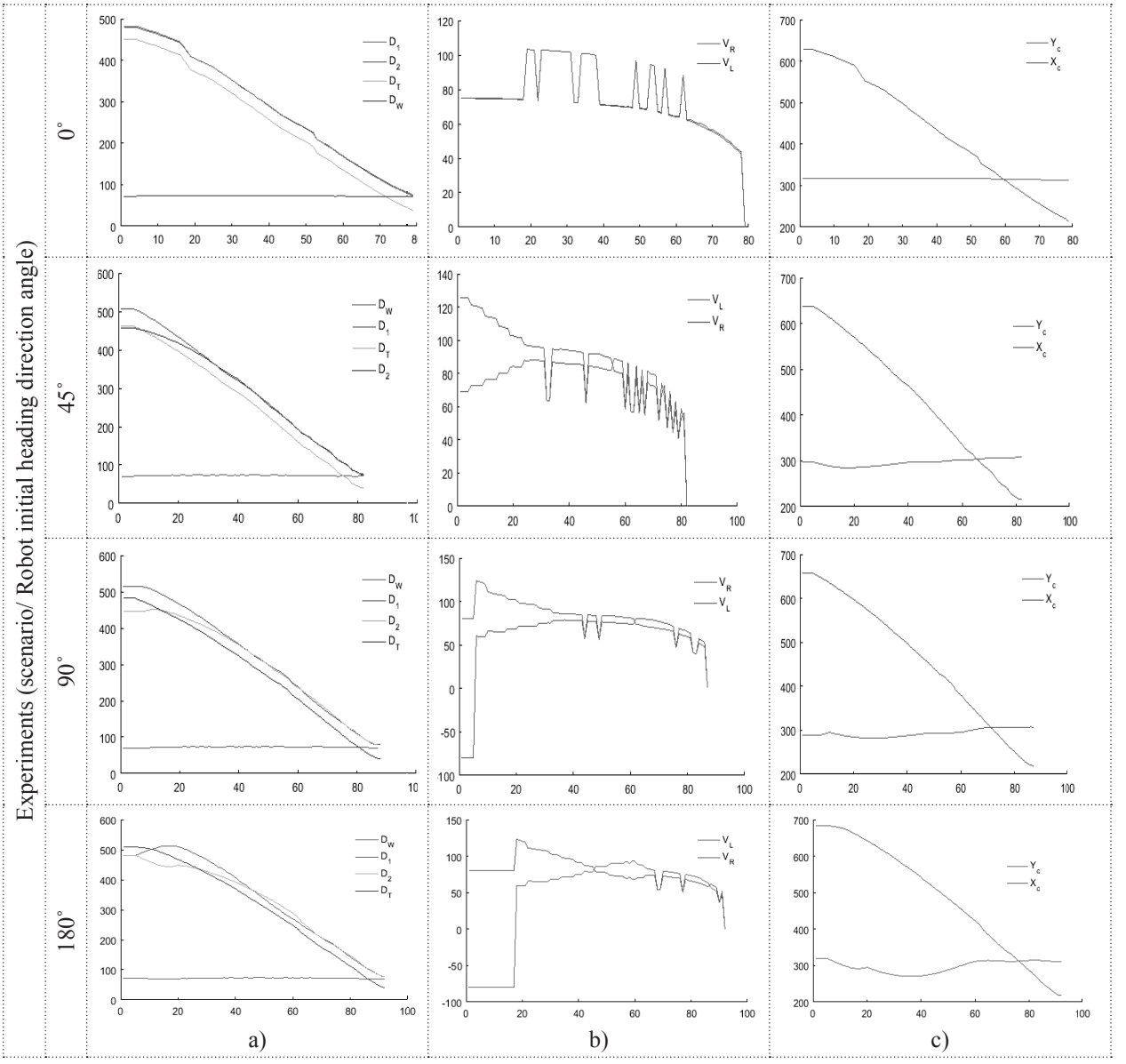


Figure 10. The performance of experimental results using the Graph-Based Gaussian controller (ADTC). X-axis of all graphs shows the number of frames processed. Legend: D_1, left distance; D_2, right distance; D_W, base distance of label on robot; D_T, distance to target; V_L, left speed; V_R, right speed; X_C, robot center x coordinate; Y_C, robot center y coordinate.

Graph-Based Decision Tree Control (GDTC)

The performances of GDTC experimental results are demonstrated in Figure 11. The changes of the input controls (D_1, D_2, D_T, D_W), outputs (V_R , V_L), and position (x, y) parameters have been given against the elapsed time in 2D space configuration. The abbreviations used herein are as follows. In column a), edge lengths that change in between the robot and the target are represented as Right (D_1) and Left (D_2). (D_W) represents the base distances between label centers on the robot. Column b) shows the changes in robot wheel speeds Right speed (V_R), Left speed (V_L) depending on the processed image frames. Column c) represents the changes in robot coordinates (x, y) depending on processed image frames.

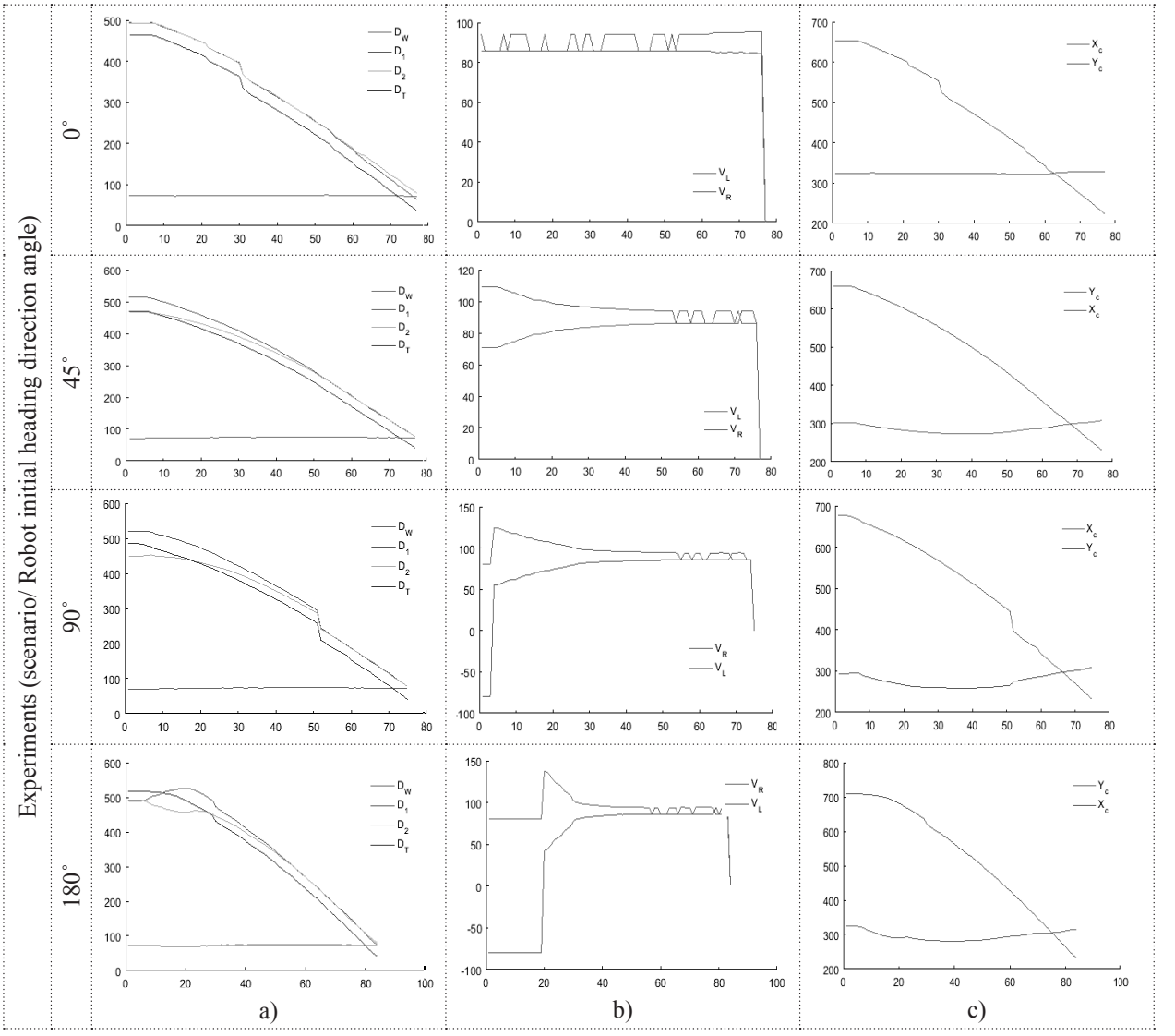


Figure 11. The performance of experimental results using the Graph-Based Decision Tree controller (GDTC). X-axis of all graphs shows the number of frames processed. Legend: D_1, left distance; D_2, right distance; D_W, base distance of label on robot; D_T, distance to target; V_L,left speed; V_R, right speed; X_C, robot center x coordinate; Y_C, robot center y coordinate.

The success and accuracy of the proposed GDTC method are presented in Figure 10 and Figure 11 in a graphical representation of the data obtained from the experimental results. These graphs show that the distance values are gradually approaching the target depending on the processed image frames, but the distance between the two labels given on the robot does not change, and the response of the robot wheel speeds to this distance is shown.

DISCUSSION OF EXPERIMENTAL RESULTS

In this paper, two efficient visual-based go-to-goal robot control approaches have been designed and compared by using the nonholonomic mobile robot. The proposed methods reduce the required parameter for the kinematic complexity of the robot's onboard equipment and eliminate costly and power-consuming tasks such as information

gathering and processing. For the nonholonomic mobile robot control, the input parameters of the kinematic control were obtained from the vision-based configuration and the robot kinematic control task has been implemented with the designed kinematic methods. The visual control approach aims to provide a system with robust behavior control, high accuracy, and low cost. The proposed method is divided into three stages; at the first stage, the test environment was observed with the overhead camera and the calibration of the camera was performed. The location information of the objects in the camera field of view is tracked by specific filtering operations. In the second step, the kinematic input parameters are obtained for robot motion control from position information. These kinematic approaches are angle- and graph-based approaches. In the third phase, Decision Trees and Gaussian algorithms, which determine the mobile robot wheel speeds using the input parameters, are used and a control loop is triggered. We have briefly described these control approaches as follows: Graph-based decision tree control (GDTC), Graph-based Gaussian control (GGC), angle-based decision tree (ADTC), and angle-based Gaussian control (AGC). Real-time experiments were carried out with this proposed system. The designed methods have demonstrated remarkable performance in experiments. The system reduces both the complexity of the mobile robot control kinematics and the robot control cost by using only an overhead camera (without using onboard sensors). It also minimizes systematic and nonsystematic errors. Tracking performance of the proposed method seems to work robustly as long as the center coordinates of the labels can be retained. As graphically illustrated previously, the triangle angles gradually approach each other. In other words, as the robot moves towards the target, the angles approach each other and eventually form a structure similar to an equilateral triangle. This is the same for edge lengths. As the triangular base angle approaches, the wheel speeds approach each other. In the graph-based approach, the wheel speeds are adjusted according to the size of the edge weights/distances. In the angle-based approach, the wheel speeds are set according to the size of the base angles of the virtual triangle being created. In both graph-based and decision-tree control approaches, the robot has successfully achieved its goal. It should be stated that the theoretical substructure of the control approach has formed and its practical aspect can be tested and applied. We would like to say that the proposed control methodologies are suitable and efficient for indoor mobile robotic systems.

CONCLUSION AND FUTURE WORKS

In this work, two control approaches have been developed and deeply investigated for nonholonomic mobile robots. By using these proposed control approaches, vision-based robot motion control has been performed. These control approaches have been employed by utilizing the overhead camera configuration. They can be used as an alternative control method to traditional robot control kinematics. A number of experiments have been carried out to measure the efficiency and performance of the system. The proposed control method provides both more flexibility and simple robot control, minimizing systematic and nonsystematic errors. It is independent of the internal sensors and only utilizes an overhead camera as sensors. Since internal or external range sensors are not necessary, the system cost is decreased by using the proposed method. In this work, it is focused especially on developing robot control approaches and using them to model go-to-goal behavior between the robot and the target. Several real-time experiments have been conducted using a nonholonomic robot in real-world configuration space for confirmation and advantages of the proposed methods. It has been understood that the system has great advantages in the control process. In future work, this system will be integrated and experimented with path planning. In addition, it is planned to design a multi-overhead camera system and using different control algorithms in the environment where the obstacles are hosted.

REFERENCES

Bernard Espiau & Radu Horaud. 1998. Visual Servoing with Calibrated Cameras: A Review. *Vigor, Esprit-IV Reactive Ltr Project, Number 26247*.

Chen, C.H.L. & M.F.R. Lee. 2011. Global Path Planning in Mobile Robot Using Omnidirectional Camera. In *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 4986-89.

Cherubini, Andrea & François Chaumette. 2013. Visual Navigation of a Mobile Robot with Laser-Based Collision Avoidance. *The International Journal of Robotics Research* **32**(2): 189-205.

Chugo, D., K. Hirose, K. Nakashima, S. Yokota, H. Kobayashi & H. Hashimoto. 2012. Camera-Based Navigation for Service Robots Using Pictographs on the Crossing Point. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 4154-59.

Dao, T.K., T.S. Pan & J.S. Pan. 2016. A Multi-Objective Optimal Mobile Robot Path Planning Based on Whale Optimization Algorithm. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, 337-42.

Dirik, M., Kocamaz, A. F. & Dönmez, E. 2016a. Vision-Based Decision Tree Controller Design Method Sensorless Application by Using Angle Knowledge. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, 1849-52.

Dirik, M., Kocamaz, A.F. & Dönmez, E. 2017a. Static Path Planning Based on Visual Servoing via Fuzzy Logic. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Dönmez, E., Kocamaz, A.F. & Dirik, M. 2016. Robot Control with Graph Based Edge Measure in Real Time Image Frames. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, 1789-92.

Dönmez, E., Kocamaz, A.F. & Dirik, M. 2017a. Visual Based Path Planning with Adaptive Artificial Potential Field. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Dönmez, E., Kocamaz, A.F. & Dirik, M. 2017c. Bi-RRT Path Extraction and Curve Fitting Smooth with Visual Based Configuration Space Mapping. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1-5.

Dönmez, E., Kocamaz, A.F. & Dirik, M. 2017. A Vision-Based Real-Time Mobile Robot Controller Design Based on Gaussian Function for Indoor Environment. *Arabian Journal for Science and Engineering*, December, 1-16.

Elsheikh, E.A., M.A. El-Bardini & M.A. Fkirin. 2016. Dynamic Path Planning and Decentralized FLC Path Following Implementation for WMR Based on Visual Servoing. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, 1-7.

Engel, J., J. Sturm & D. Cremers. 2012. Camera-Based Navigation of a Low-Cost Quadrocopter. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2815-21.

Farah, E. 2017. "Fuzzy PID Based Path Tracking Control of a 5-DOF Needle-Holding Robot." In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, 1-5.

Hu, G., N. Gans, N. Fitz-Coy & W. Dixon. 2010. Adaptive Homography-Based Visual Servo Tracking Control via a Quaternion Formulation. *IEEE Transactions on Control Systems Technology,* **18**(1): 128-35.

Hutchinson, S., G.D. Hager & P.I. Corke. 1996. A Tutorial on Visual Servo Control. *IEEE Transactions on Robotics and Automation,* **12**(5): 651-70.

Jiang, M., Y. Chen, W. Zheng, H. Wu & L. Cheng. 2016. Mobile Robot Path Planning Based on Dynamic Movement Primitives. In *2016 IEEE International Conference on Information and Automation (ICIA)*, 980-85.

Keshmiri, M., W.F. Xie & A. Mohebbi. 2014. Augmented Image-Based Visual Servoing of a Manipulator Using Acceleration Command. *IEEE Transactions on Industrial Electronics,* **61**(10): 5444-52.

Li, B., Y. Fang, G. Hu & X. Zhang. 2016. Model-Free Unified Tracking and Regulation Visual Servoing of Wheeled Mobile Robots. *IEEE Transactions on Control Systems Technology,* **24**(4): 1328-39.

Lippiello, V., J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. Á Trujillo, Y. R. Esteves & A. Viguria. 2016. Hybrid Visual Servoing With Hierarchical Task Composition for Aerial Manipulation. *IEEE Robotics and Automation Letters,* **1**(1): 259–66.

Mahon, I., S.B. Williams, O. Pizarro & M. Johnson-Roberson. 2008. Efficient View-Based SLAM Using Visual Loop Closures. *IEEE Transactions on Robotics,* **24**(5): 1002-14.

Parikh, A., R. Kamalapurkar, H.Y. Chen & W.E. Dixon. 2015. Homography Based Visual Servo Control with Scene Reconstruction. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 6972-77.

Siciliano, Bruno & Oussama Khatib. 2007. *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Tsai, C.Y., & K.T. Song. 2009. Visual Tracking Control of a Wheeled Mobile Robot With System Model and Velocity Quantization Robustness. *IEEE Transactions on Control Systems Technology,* **17**(3): 520-27.

Zhang, X., Y. Fang, B. Li & J. Wang. 2017. Visual Servoing of Nonholonomic Mobile Robots With Uncalibrated Camera-to-Robot Parameters. *IEEE Transactions on Industrial Electronics,* **64**(1): 390-400.

Ziaei, Z., R. Oftadeh & J. Mattila. 2014. Global Path Planning with Obstacle Avoidance for Omnidirectional Mobile Robot Using Overhead Camera. In *2014 IEEE International Conference on Mechatronics and Automation (ICMA)*, 697-704.

Ziaei, Oftadeh & Mattila 2015a. Vision-Based Path Coordination for Multiple Mobile Robots with Four Steering Wheels Using an Overhead Camera. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 261-68.

Zou, Y., C. Wen, M. Shan & M. Guan. 2016. Image-Based Visual Tracking Adaptive Control for Mobile Robots. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1-6.