Modified Extremum Seeking Control for Target Tracking and Formation Control in Pursuit-Evasion Game

Fachruddin Ari Setiawan
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ddinfachruddin00@gmail.com

Trihastuti Agustinah
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
trihastuti@ee.its.ac.id

Muhammad Fuad
Faculty of Electrical Engineering
Universitas Trunojoyo Madura
Bangkalan, Indonesia
fuad@trunojoyo.ac.id

***Abstract*— In a pursuit-evasion game, the mobile robot pursuer's ability to navigate from its initial position to the evader while maintaining a safe distance from other objects requires a good obstacle avoidance system. This study aims to perform target tracking in evader sieges and obstacle avoidance against other pursuer robots and static obstacles by proposing a modified extreme seeking controller (ESC). A modified backstepping control (BC) was used as an autopilot control for a nonholonomic mobile robot to execute the modified ESC command. The modified BC based on the modified ESC requires the positions of the targeted evader, pursuers, and obstacles. The pursuer uses this information to capture an evader by arranging the desired formation without colliding with static obstacles or other robots. The results of the simulations show that the pursuers successfully surround the evader and construct the formation without colliding with obstacles. The proposed method resulted in the closest distance of 2.071 m between the pursuers, 1.954 m between each pursuer and the evader, and 2.425 m between the pursuers and static obstacles.**

Keywords*— *extremum seeking control, formation control, obstacle avoidance, pursuit-evasion game, target tracking

I. INTRODUCTION

Currently, research on multi-agent systems has advanced significantly, with the findings of studies demonstrating the benefits of multi-agent systems over single agent systems [1]-[4]. This multi-agent system has been widely used in a variety of fields, including target tracking [5], [6], formation control [7], [8], obstacle avoidance [9], [10], and other interesting research areas involving multi-agent systems, such as pursuit-evasion games or differential games [11]-[14].

The pursuit-evasion game discusses methods that can be used in a multi-agent system, where the pursuer agents are able to capture or surround the evader agents with the least amount of cost. In a pursuit-evasion game, several processes such as target tracking, formation control, and obstacle avoidance interact to form an interconnected system. Extreme seeking control (ESC) is control method that can be used in a pursuit-evasion game to perform target tracking and then form the desired formation, as investigated in [15]. This method uses the scientific basis of finding the optimal point to control the formation of a multi-agent system, but the method in this study can still be developed because the dynamic model of the controlled agent is very simple, that is, by assuming that the controlled agent is a moving point. This study used an unobstructed environment. If the modified ESC is applied to a more complex plant and environment with obstacles, the dynamic model of the system will become more general and realistic, allowing the method's success in target tracking, formation control, and obstacle avoidance to be tested.

A nonholonomic mobile robot is a plant that can be used in pursuit-evasion games. Currently, many controllers can be applied to nonholonomic mobile robots. One of them is the backstepping control (BC). The BC was used to control the mobile robot to follow a predetermined target point [16]. This study modifies the BC to complement the modified ESC as a mobile robot autopilot control and decomposer of the problems raised.

Modified ESC and BC can be realized in pursuit-evasion games for pursuit, encirclement, and surveillance of the evader when it is under siege. Modification of the ESC also intends to control the agent so that it can avoid static and moving obstacles. In the condition that only one evader is the target of pursuit, no task allocation method needs to be added in the combination of the modified ESC and modified BC. However, if there are multiple evaders as pursuit targets, a task allocation method is required to determine one evader among several evaders as the priority target in each pursuer group. This study aims to develop a method for target tracking in evader sieges and obstacle avoidance against other pursuer robots and static obstacles by modifying the ESC and combining it with the modified BC in a pursuit-evasion game with one target.

The main contributions of this research are:

- Develop a modified ESC that can be applied to nonholonomic mobile robot systems for target tracking, formation control, and obstacle avoidance in a simple Pursuit-Evasion Game system. Modifications are made to allow the pursuer agent to surround the evader and form the desired formation in an

 How to cite: Setiawan, F. A., Agustinah, T., & Fuad, M. (2022). Modified Extremum Seeking Control for Target Tracking and Formation Control in Pursuit-Evasion Game. *JAREE (Journal on Advanced Research in Electrical Engineering), 6(2).*

environment that contains static and dynamic obstacles.

- The backstepping control is modified to simplify the nonholonomic mobile robot system as a moving mass point, so that the modified ESC only requires information about the moving target position.

The remainder of this paper is organized as follows. Section II describes the materials and methods, including the dynamics and kinematics of a nonholonomic mobile robot, backstepping control formulation, modified BC for moving-point input, ESC formulation, modified ESC for target tracking and formation control, and modified ESC for obstacle avoidance. The simulation results are explained in Section III. Finally, Section IV concludes the study.

II. Material and Methods

A. Dynamics and Kinematics Nonholonomic Mobile Robot

This study used nonholonomic mobile robots as pursuers or evader agents. The robot used was a two-wheeled differential drive robot. The direction of motion of the mobile robot in the Cartesian plane is shown in Fig. 1. The center of mass of the robot is denoted as C, and the width of the robot is $2L$. The diameter of the wheel is $2R_a$ and the distance between the center of mass C and wheel axis A has a magnitude of a.

The dynamics and kinematics equations of the robot are formulated as follows [16]:

$$M(q)\ddot{q} + V(q,\dot{q}) = B(q)\tau - A^T(q)\lambda \quad (1)$$

$$\begin{bmatrix} m & 0 & ma\sin\theta \\ 0 & m & -ma\cos\theta \\ ma\sin\theta & -ma\cos\theta & I_C + 2ma^2 \end{bmatrix}\ddot{q} + \begin{bmatrix} ma\dot{\theta}^2\cos\theta \\ ma\dot{\theta}^2\sin\theta \\ 0 \end{bmatrix} = \frac{1}{R_a}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} - \begin{bmatrix} -\sin\theta \\ cos\,\theta \\ -a \end{bmatrix}\left[-m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}\right] \quad (2)$$

with

$$q = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$$

$$\lambda = \left[-m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}\right]$$

$$M(q) = \begin{bmatrix} m & 0 & ma\sin\theta \\ 0 & m & -ma\cos\theta \\ ma\sin\theta & -ma\cos\theta & I_C + 2ma^2 \end{bmatrix}$$

$$V(q,\dot{q}) = \begin{bmatrix} ma\dot{\theta}^2\cos\theta \\ ma\dot{\theta}^2\sin\theta \\ 0 \end{bmatrix}$$

$$B(q) = \frac{1}{R_a}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ cos\,\theta \\ -a \end{bmatrix}$$

where $q \in \Re^{3\times1}$ is the position and orientation vector, $M(q) \in \Re^{3\times3}$ is the inertia matrix, $V(q,\dot{q}) \in \Re^{3\times1}$ is the centripetal and Coriolis matrix, $B(q) \in \Re^{3\times2}$ is the input transformation matrix, $A(q) \in \Re^{1\times3}$ is the matrix associated with the constraints, $\lambda \in \Re^{1\times1}$ is the vector of constraint forces, x_c is the position of the robot along x-axis, y_c is the robot's position along y-axis, θ is the orientation of the robot, $\dot{q}$ is the robot's velocity, $\ddot{q}$ is the robot's acceleration, m is the robot's mass, and I_C is the robot's moment of inertia. The input vector of the robot is represented by $\tau \in \Re^{2\times1}$ is the torque applied to the wheel, consisting of τ_R for the torque on the right wheel and τ_L for the torque on the left wheel.

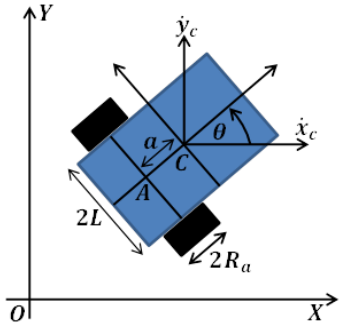


Fig. 1. Nonholonomic Differential Mobile Robot.

B. Modified Backstepping Control

Backstepping control (BC) is a method that can be used as autopilot control for a nonholonomic mobile robot. In this study, the BC is used to connect the modified ESC and the nonholonomic mobile robot so that the commands can be carried out by the mobile robot and provide the desired output. A block diagram of BC is shown in Fig. 2.

1) BC Formulation

In the BC formulation, transformation matrix *S* is used to transform the differential from the robot position state into a new state containing the linear and angular velocities of the robot in the body frame, which is formulated as follows [16]:

$$\dot{q} = S\mathrm{v} \quad (3)$$

$$S^T M S\dot{\mathrm{v}} + S^T\left(M\dot{S} + V_m S\right)\mathrm{v} = S^T B\tau \quad (4)$$

$$\bar{M}\dot{\mathrm{v}} + \bar{V}_m\mathrm{v} = \bar{B}\tau \quad (5)$$

With

$$S = \begin{bmatrix} cos\,\theta & -a\,sin\,\theta \\ sin\,\theta & a\,cos\,\theta \\ 0 & 1 \end{bmatrix} \quad (6)$$

$$q = [x_c \quad y_c \quad \theta]^T \quad (7)$$

$$\mathrm{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (8)$$

$$\bar{M} = S^T M S$$

$$V(q,\dot{q}) = V_m\dot{q}$$

$$V_m = \begin{bmatrix} 0 & 0 & ma\dot{\theta}\cos\theta \\ 0 & 0 & ma\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix}$$

$$\bar{V}_m = S^T V_m S$$

$$\bar{B} = S^T B$$

where $S(q) \in \Re^{3\times2}$ is a Jacobian matrix that transforms velocity in base coordinates/body frame (v) to velocity in Cartesian coordinates/local frame $(\dot{q})$, $\mathrm{v} \in \Re^{2\times1}$ is a velocity vector in base coordinates consisting of linear velocity $(v = v_1)$ and angular velocity $(\omega = v_2)$, $\bar{M}(q) \in \Re^{2\times2}$ is the inertia matrix, $\bar{V}_m(q,\dot{q}) \in \Re^{2\times2}$ is the centripetal and Coriolis matrix, $\bar{B}(q) \in \Re^{2\times2}$ is the input transformation matrix.

Assuming that u is the auxiliary input, using nonlinear feedback in (9) will change the dynamic control problem into a kinematic control problem, as in (10-11).

$$\tau = \bar{B}^{-1}[\bar{M}u + \bar{V}_m \mathrm{v}] \quad (9)$$

$$\dot{q} = S\mathrm{v} \quad (10)$$

$$\dot{\mathrm{v}} = u \quad (11)$$

To satisfy the tracking conditions in (10-11), the tracking error vector is expressed in (12-13) and derived in (14-15) as follows:

$$e_p = T_e(q_r - q) \quad (12)$$

$$e_p = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (13)$$

$$\dot{e}_p = \begin{bmatrix} v_2 e_2 - v_1 + v_r \cos e_3 \\ -v_2 e_1 + v_r \sin e_3 \\ \omega_r - v_2 \end{bmatrix} \quad (14)$$

$$v_c = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \quad (15)$$

where, $K = [k_1 \quad k_2 \quad k_3]$ is a vector with positive constants.

The derivative of v_c is formulated as follows:

$$\dot{v}_c = \begin{bmatrix} \dot{v}_r \cos e_3 \\ \dot{\omega}_r + k_2 \dot{v}_r e_2 + k_3 \dot{v}_r \sin e_3 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{e}_p \quad (16)$$

Assuming that the linear and angular velocity references are constant, the derivative of the auxiliary velocity control input v_c and auxiliary input u can be formulated as follows:

$$\dot{v}_c = \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{e}_p \quad (17)$$

$$u = \dot{v}_c + K_4(v_c - v) \quad (18)$$

where $K_4 = k_4 I$, k_4 is a positive constant.

2) Modified BC for Moving Point Input

This study proposes a modified BC that can be used as an input in the form of robot speed on the x and y axes. The speed on the two axes is the output of the ESC, which must be carried out by the robot so that it can perform target encirclement and target tracking.

The input used in this test is still the destination point (x_r, y_r), the difference is that the values of x_r and y_r are obtained from a function that causes these values to change continuously over time. The values of x_r and y_r are determined by the new variables v_x and v_y as new inputs.

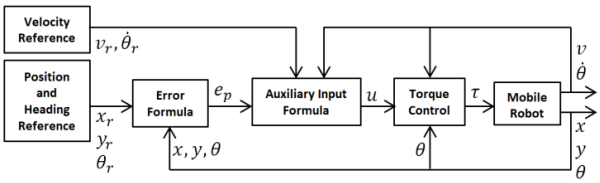


Fig. 2. Backstepping Control Block Diagram

$$x_r = x + k_{way}.T.v_x \quad (19)$$

$$y_r = y + k_{way}.T.v_y \quad (20)$$

where

x : robot position on the x-axis
y : robot position on the y-axis
x_r : destination point on the x-axis
y_r : destination point on the y-axis
v_x : waypoint speed on x-axis
v_y : waypoint speed on y-axis
k_{way} : waypoint constant (value: 197)
T : sampling time (value: 0.005 s)

The modified BC method that was applied and combined with the modified ESC method is shown in Fig. 4.

C. Modified Extremum Seeking Control

This extremum-seeking control (ESC) is used on each robot so that the robot can maintain a formation and follow a moving target by defining the appropriate potential function. Position information from other agents and target positions were used as input for the ESC. From this information, an input in the form of a two-dimensional speed is issued, which is used as input for the pursuer agent. A flowchart of ESC is shown in Fig. 3.

1) ESC Formulation

The potential function between the number of *N* agents and target is defined as follows [15]:

$$J_{at}(x, x_t) = \sum_{i=1}^{N} J_{it}\left(\left\|x^i - x_t\right\|\right) \quad (21)$$

where J_{it} is a potential function between the i-th agent and the target, x^i is the center position of the i-th pursuer agent, and x_t is the center position of the target. It is assumed that J_{it} results in a minimum value at $\left\|x^i - x_t\right\| = \delta_{it}$. δ_{it} is the defined distance between the agent and target. For collision avoidance purposes, δ_{it} was set at more than 0.

The potential function between the agent and other agents is formulated as follows:

$$J_{aa}(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}\left(\left\|x^i - x^j\right\|\right) \quad (22)$$

where x^j is the center position of the j-th pursuer agent, and J_{ij} is the potential function between the i-th and the j-th agents.

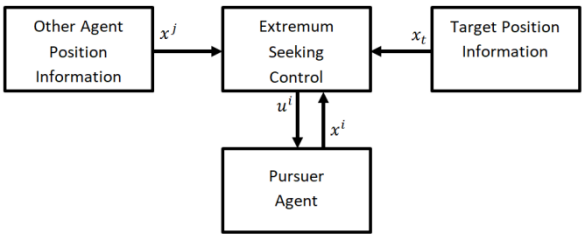


Fig. 3. Extremum Seeking Control Flowchart.

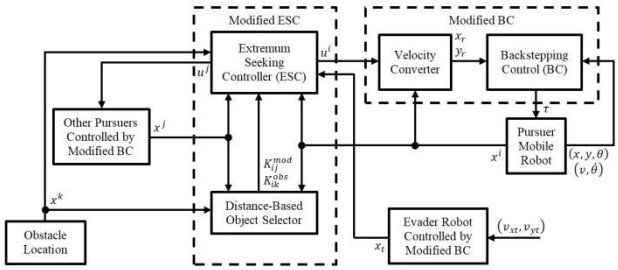


Fig. 4. Modified Extremum Seeking Control Block Diagram

From the two potential functions shown in (21) and (22), the performance function can be obtained as follows:

$$y = J(x, x_t) = J\left(\|x^i - x_t\|, \|x^i - x^j\|\right) = K_{at}J_{at}(x, x_t) + K_{aa}J_{aa}(x) \quad (23)$$

$$y = K_{at}\sum_{i=1}^{N} J_{it}\left(\|x^i - x_t\|\right) + K_{aa}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} J_{ij}\left(\|x^i - x^j\|\right) \quad (24)$$

where y denotes the overall potential function. K_{at} is the weight of the potential function between the agent and target, and K_{aa} is the weight of the potential function between the agent and other agents.

The attraction-repulsion relationship between the agent and other agents is shown in (25), while (26) represents the attraction-repulsion relationship between the agent and the target.

$$g_{ar}^{ij}(\|x\|)\begin{cases} > 0, & \|x\| > \delta_{ij}, \\ = 0, & \|x\| = \delta_{ij}, \\ < 0, & \|x\| < \delta_{ij}. \end{cases} \quad (25)$$

$$h^{it}(\|x\|)\begin{cases} > 0, & \|x\| > \delta_{it}, \\ = 0, & \|x\| = \delta_{it}, \\ < 0, & \|x\| < \delta_{it}. \end{cases} \quad (26)$$

Equation (24) is derived so that the following equation is obtained:

$$\nabla_{x^i}J(x, x_t) = K_{at}(x^i - x_t)h^{it}\left(\|x^i - x_t\|\right) + K_{aa}\sum_{j=1, j\neq i}^{N}(x^i - x^j)g_{ar}^{ij}\left(\|x^i - x^j\|\right)$$

If the Lyapunov function is taken as written in (27), then (28) can be obtained as follows:

$$V = J(x, x_t) \quad (27)$$

$$\dot{V} = \sum_{i=1}^{N}\left[\nabla_{x^i}J(x, x_t)\right]^T\dot{x}^i + \left[\nabla_{x^t}J(x, x_t)\right]^T\dot{x}_t = \sum_{i=1}^{N}\left[\nabla_{x^i}J(x, x_t)\right]^T(u^i - \dot{x}_t) \quad (28)$$

The control input u for the i-th agent is given as follows:

$$u^i = \dot{x}_t - k^i\nabla_{x^i}J(x, x_t); \qquad k^i > 0 \quad (29)$$

2) Modified ESC for Target and Formation Control

This study presents a modified ESC for target-tracking and formation control. These modifications were used to select the objects included in the ESC formula. Only objects whose distance to the robot is less than a predetermined minimum distance are included in the ESC formulation. Therefore, the ESC formulation was modified as follows:

$$\nabla_{x^i}J(x, x_t) = K_{at}(x^i - x_t)h^{it}\left(\|x^i - x_t\|\right) + K_{aa}\sum_{j=1, j\neq i}^{N} K_{ij}^{mod}(x^i - x^j)g_{ar}^{ij}\left(\|x^i - x^j\|\right) \quad (30)$$

with

$$K_{ij}^{mod} = \begin{cases} 1, & if\ \|x^i - x^j\| < d_{range}, \\ 0, & otherwise. \end{cases}$$

where K_{ij}^{mod} is the modification constant of the ESC, and d_{range} is the maximum distance that the robot can detect its surroundings. If the distance of the i-th robot with the j-th subject is less than d_{range}, then K_{ij}^{mod} will be worth 1. If the distance is greater than d_{range}, then K_{ij}^{mod} is zero, and the subject is ignored in the calculation of the ESC formula.

3) Modified ESC for Obstacle Avoidance

This section discusses the modification of ESC for obstacle avoidance. The modification is made by adding the obstacle variable in the ESC formula, as follows:

$$\nabla_{x^i}J(x, x_t) = K_{at}(x^i - x_t)h^{it}\left(\|x^i - x_t\|\right) + K_{aa}\sum_{j=1, j\neq i}^{N} K_{ij}^{mod}(x^i - x^j)g_{ar}^{ij}\left(\|x^i - x^j\|\right) + K_{ao}\sum_{k=1}^{N} K_{ik}^{obs}(x^i - x^k)g_{ar}^{ik}\left(\|x^i - x^k\|\right) \quad (31)$$

where K_{ao} is the weight of the potential function between the agent and obstacle, K_{ik}^{obs} is an object selection constant similar to K_{ij}^{mod}, x^k is the center position of the k-th obstacle, and $g_{ar}^{ik}\left(\|x^i - x^k\|\right)$ is the attraction-repulsion relationship between the agent and obstacle. The formula for determining the values of K_{ik}^{obs} and $g_{ar}^{ik}\left(\|x^i - x^k\|\right)$ values is as follows:

$$K_{ik}^{obs} = \begin{cases} 1, & if\ \|x^i - x^k\| < r_{obs}, \\ 0, & otherwise. \end{cases}$$

$$g_{ar}^{ik}\left(\|x^i - x^k\|\right) = \|x^i - x^k\| - \delta_{ik}$$

where r_{obs} is the maximum distance of obstacles that can be detected by the robot, and δ_{ik} is the safe distance between the i-th robot and the center of the k-th obstacle.

The block diagram of the modified ESC method for evader tracking and obstacle avoidance using the modified BC is shown in Fig. 4.

III. Results and Discussion

The parameters of the nonholonomic mobile robot were obtained from the datasheet for a two-wheeled differential drive, as listed in Table I [16], [17]. The parameters used in the BC are listed in Table II.

TABLE I. PARAMETER OF NONHOLONOMIC MOBILE ROBOT

Symbol	Parameter	Value
m	Robot mass (kg)	62
a	Distance between wheel axis and rotation axis (m)	0.00001
I	Total Inertia $(kg.m^2)$	1.24
R_a	Wheel radius (m)	0.1
L	Half of the robot's width (m)	0.2

TABLE II. PARAMETER OF BACKSTEPPING CONTROL

Parameter	Value
k_1	1
k_2	8
k_3	1
k_4	1.48

TABLE III. PARAMETER OF VELOCITY REFERENCES

Time (s)	$v_x(m/s)$	$v_y(m/s)$
0 – 20	1.8	0
20 – 40	1.273	-1.273
40 – 60	1.8	0
60 – 80	1.273	1.273
80 – 100	0	1.5
100 – 120	-1.061	1.061
120 – 140	-1.8	0
140 – 160	-1.273	-1.273
160 – 180	0	-1.5
180 – 190	-1.273	-1.273
190 – 200	-1.8	0

A. *Simulation of Modified BC for Moving Target*

The following simulation aimed to test the modified BC in directing the robot to follow a moving destination point. The parameters used are k_{way}, v_x, and v_y where k_{way} is assigned a value of 238, and the two parameters v_x and v_y are the target speed on the x-axis and y-axis, respectively, whose values vary, as shown in Table III.

The robot trajectory and resulting waypoint trajectory of the moving target are shown in Fig. 5. From this figure, it can be concluded that the robot succeeded in following the movement of the moving target point, as indicated by the overlap of the two lines.

The speed of the robot on the x-axis in Fig. 6 is compared with the speed input (v_x) shown in Table III. The robot speed on the y-axis in Fig. 7 is compared against the velocity input of the target point (v_y). From Fig. 6 and Fig. 7, the characteristics of the speed response of the robot include the x-axis speed with a mean settling time of 2.0696 s and a mean square steady-state errors of 3.1375×10^{-4}, while the Y-axis speed has a mean settling time of 2.1823 s and a mean square steady-state errors 6.4739×10^{-4}. It can be concluded from the response characteristics that the speed of the robot on the x-axis or y-axis of the earth frame successfully follows the given speed input with relatively small steady-state errors and a relatively short settling time.

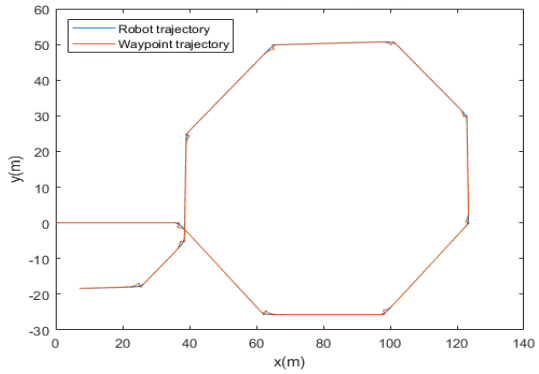


Fig. 5. BC experiment result with moving-point waypoint.

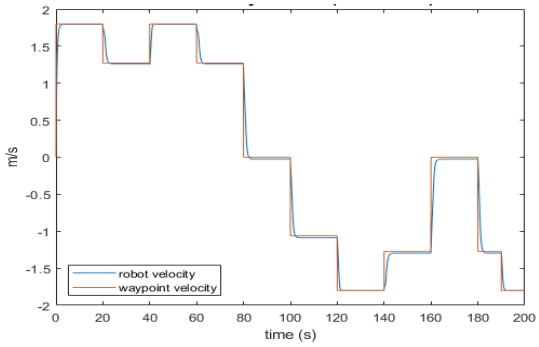


Fig. 6. Robot velocity in x-axis (earth frame).

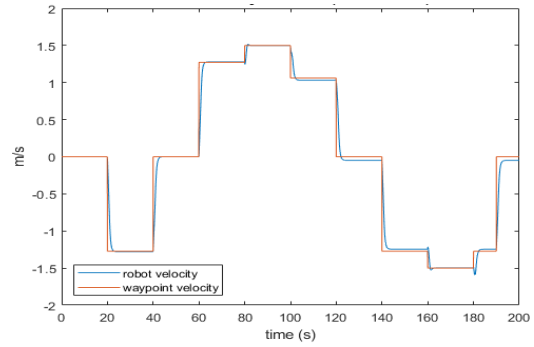


Fig. 7. Robot velocity in y-axis (earth frame).

TABLE IV. PARAMETER OF MODIFIED ESC

Symbol	Definition	Value
K_{at}	Weight of the potential function between the agent and the target	0.0002
K_{aa}	Weight of the potential function between each pursuer	0.01
δ_{it}	Desired distance between the agent and the target (m)	3
δ_{ij}	Desired distance between each pursuer (m)	5
k	Input constant of ESC controller	10
v_{xt}	Target velocity in x-axis (m/s)	1.5
v_{yt}	Target velocity in y-axis (m/s)	0
d_{range}	Maximum distance of robot detection (m)	7.2

B. *Simulation of Modified ESC for Target Tracking and Formation Control*

This simulation aimed to test the modification of the ESC method and determine the effect of these modifications on the movement of the pursuer robot. In addition, this simulation also aims to determine whether modifications to the ESC will make the pursuer robot successfully chase and surround a moving evader. The modified ESC parameters are presented in Table IV.

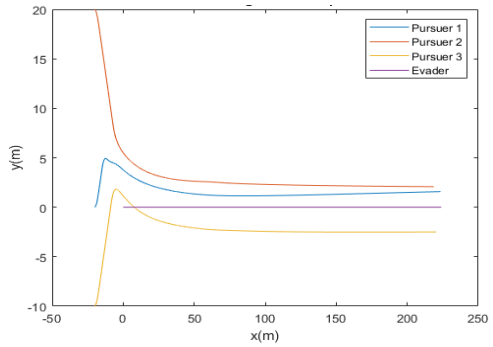


Fig. 8. Robot trajectory using standard ESC

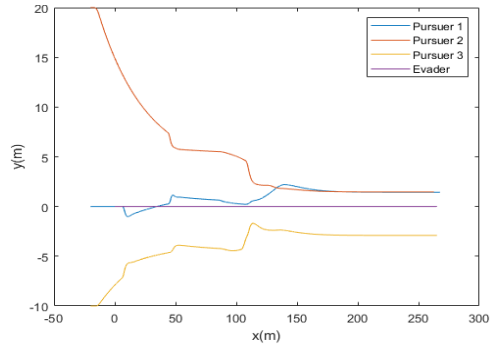


Fig. 9. Robot trajectory using MESC

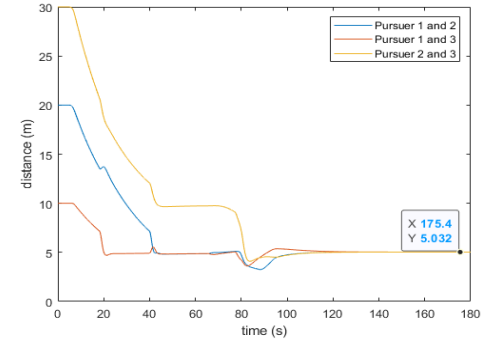


Fig. 10. Distance between pursuers

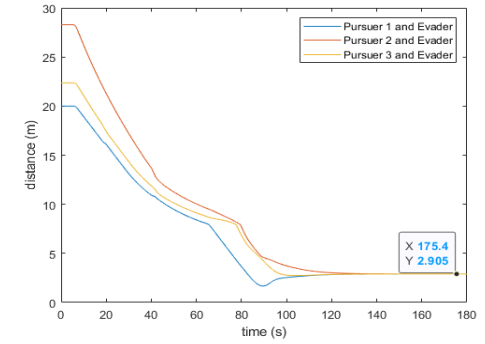


Fig. 11. Distance between pursuer and evader

The simulation results obtained using the standard ESC and modified ESC are shown in Fig. 8 and Fig. 9, respectively. The performance of the modified ESC in terms of distances is shown in Fig. 10 and Fig. 11.

These figures show that using the standard ESC affected the pursuers' attraction to each other and caused movements that did not prioritize the pursuit of the target. Using the modified ESC, the pursuer robots prioritized pursuing the targets and avoiding the other pursuers when they were too close.

There were no mutual attractions when the two pursuers were too far apart at the beginning of the simulation. The distances between the robots are shown in Fig. 10. It has a value of 5.032 m as the equilibrium point, corresponding to the given input (δ_{ij}) in the modified ESC. The one shown in Fig. 11 is the distance between all pursuer and evader, an equilibrium point occurs at 2.905 m, almost following the input value of (δ_{it}) used.

TABLE V. PARAMETER OF MESC FOR PURSUIT-EVASION GAME

Symbol	Definition	Value
K_{at}	Weight of the potential function between the agent and the target	0.0009
K_{aa}	Weight of the potential function between each pursuer	0.012
K_{ao}	Weight of the potential function between the agent and obstacles	0.048
δ_{it}	Desired distance between the agent and the target (m)	3
δ_{ij}	Desired distance between each pursuer (m)	5
δ_{ik}	Desired distance between the agent and obstacles (m)	8
k	Input constant of ESC controller	10
v_{xt}	Target velocity in x-axis (m/s)	0.8485
v_{yt}	Target velocity in y-axis (m/s)	0.8485
r_{obs}	Maximum distance of robot's obstacle detection (m)	7.55

C. Simulation of MESC for Pursuit-Evasion Game (PEG)

This simulation aims to test whether the modified ESC can be used to control the pursuer such that the evader can be surrounded while avoiding all existing obstacles. The simulation was carried out using three pursuer robots, one evader robot, and 10 static obstacles. The initial position of the first pursuer robot was always at the point (0,0), whereas the initial positions of the other pursuer robots and the static obstacle were randomized using a uniform distribution. Several assumptions were made for the PEG simulation. The evader robot moves straight by ignoring obstacles and the pursuer robot, and all pursuers must chase the evader while avoiding obstacles around the pursuer. The parameters used in all PEG scenarios are listed in Table V.

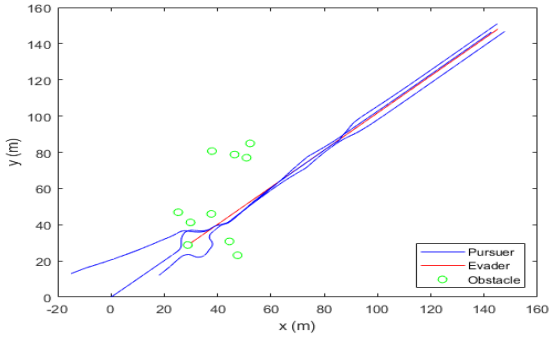
1) Scenario 1

Each scenario had a difference in the initial positions of the second and third robots and the positions of all obstacles. This is because of the use of a uniform random function in the selection of the position of the variable to test the method used. The parameters of the initial location of the robot and the location of the obstacles used in Scenario 1 are listed in Table VI.

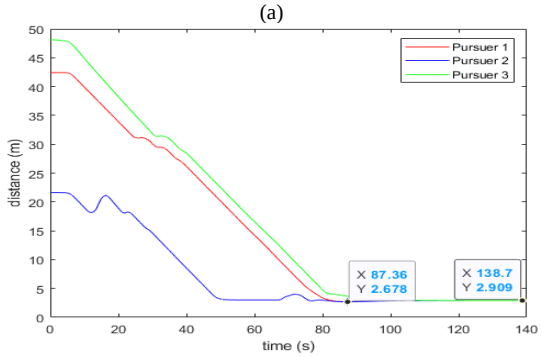
The results of the simulation using Scenario 1 are shown in Fig. 12.(a-d). The trajectories of the pursuer and evader are shown in Fig. 12.(a). Information about the distance between the pursuer and evader is shown in Fig. 12.(b). Fig. 12.(c) contains information regarding the distance between the pursuers. Information about the minimum value of the distance between the pursuers and obstacles is shown in Fig. 12.(d).

TABLE VI. OBJECT INITIAL LOCATIONS FOR SCENARIO 1

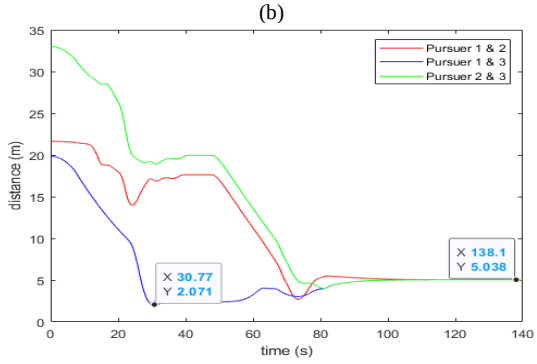
Object	x(m)	y(m)	Object	x(m)	y(m)
Pursuer 1	0	0	Obstacle 4	29.8094	41.2628
Pursuer 2	18	12	Obstacle 5	52.1977	84.9309
Pursuer 3	-15	13	Obstacle 6	47.4591	23.1198
Evader	30	30	Obstacle 7	50.8481	76.9867
Obstacle 1	44.4202	30.7941	Obstacle 8	46.3653	78.7713
Obstacle 2	37.6363	45.9828	Obstacle 9	37.8518	80.6826
Obstacle 3	28.7868	28.7868	Obstacle 10	25.1971	46.8667



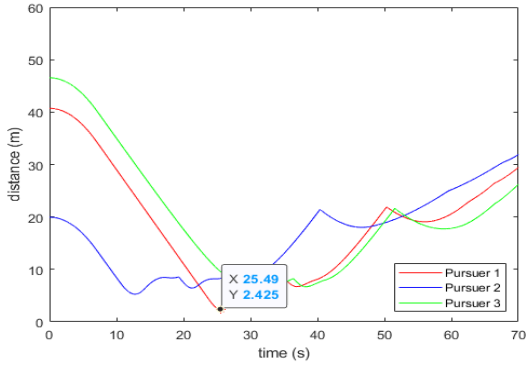
(a)



(b)



(c)

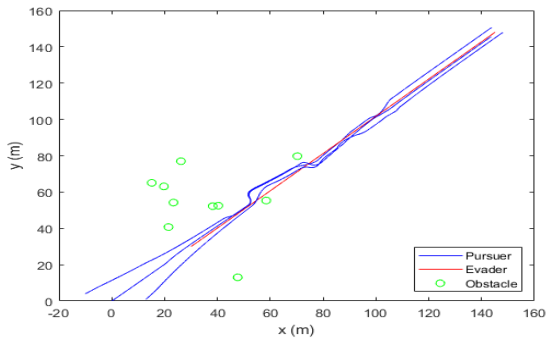


(d)

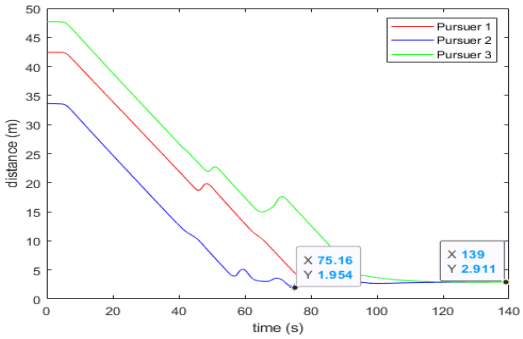
Fig. 12. Simulation result of PEG scenario 1. (a) Robot Tracjectory. (b) Distance between pursuers and evader. (c) Distance between pursuers. (d) Distance between pursuers and nearest obstacle (enlarged view).

TABLE VII. OBJECT INITIAL LOCATIONS FOR SCENARIO 2

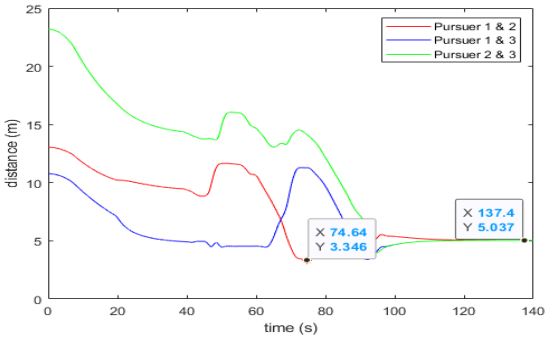
Object	x(m)	y(m)	Object	x(m)	y(m)
Pursuer 1	0	0	Obstacle 4	19.7141	63.1059
Pursuer 2	13	1	Obstacle 5	15.1253	65.0916
Pursuer 3	-10	4	Obstacle 6	58.4433	55.3583
Evader	30	30	Obstacle 7	26.1928	77.0040
Obstacle 1	38.2126	52.2486	Obstacle 8	70.2350	79.7749
Obstacle 2	21.4683	40.7295	Obstacle 9	23.3324	54.2237
Obstacle 3	47.6768	13.0730	Obstacle 10	40.3142	52.4869



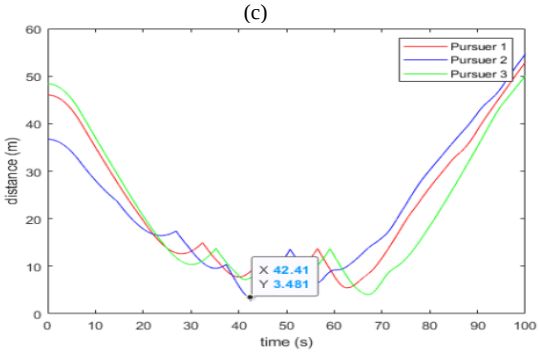
(a)



(b)



(c)



(d)

Fig. 13. Simulation result of PEG scenario 2. (a) Robot Tracjectory. (b) Distance between pursuers and evader. (c) Distance between pursuers. (d) Distance between pursuers and nearest obstacle (enlarged view).

2) Scenario 2

The parameters of the initial location of the robot and the locations of the obstacles used in scenario 2 are listed in Table VII. The results of the simulation using scenario 2 are shown in Fig. 13.(a-d).

3) Results Analysis

The trajectories of the robot are shown in Figs. 12.(a) and 13.(a). From these figures, it can be observed that the pursuer managed to avoid obstacles. However, to further ensure that there is no collision between each robot or between the robots and obstacles, it is necessary to pay attention to the distance, as shown in Figs. 12.(b-d) and Figs. 13.(b-d). The distances between all pursuers and evaders are shown in Figs. 12.(b) and 13.(b). From these figures, it can be concluded that none of the pursuers hit the evader, with the minimum distance between the pursuer and the evader being 2.678 m in scenario 1 at 87.36 s and 1.954 m in scenario 2 at 75.16 s.

Information regarding the distance between the pursuers is shown in Figs. 12.(c) and 13.(c). From these figures, it can be concluded that there is no collision between pursuers. The minimum distance between the pursuers was 2.071 m in Scenario 1 at 30.77 s and 3.346 m in Scenario 2 at 74.64 s. The Information on the minimum value of the pursuer distance against all obstacles at all times is illustrated in Figs. 12.(d) and 13.(d). From these figures, it can be concluded that no pursuer hits an obstacle. The minimum distance between the pursuer and all obstacles was 2.425 m in Scenario 1 at 25.49 s and 3.481 m in Scenario 2 at 42.4 s.

By looking at the distance between the pursuer and evader in Figs. 12.(b) and 13.(b), it can be concluded that at the end of the simulation, all pursuers can chase and surround the evader at a distance that is very close to the desired value of 3 m. With regard to information from the distance between the pursuers in Figs. 12.(c) and 13.(c), it can be concluded that at the end of the simulation, all pursuers can surround the evader and form a formation with a distance between pursuers approaching the desired number of 5 m.

From all the explanations of the above results, it is concluded that the modified ESC succeeded in directing the pursuer to surround the evader and then construct a formation. At the same time, the modified ESC directs the pursuer to avoid other pursuers, evaders, and all obstacles so that there is no collision.

IV. Conclusion

This paper proposes a modified BC that can be used to control a mobile robot with input in the form of a moving target speed such that the speed of the mobile robot on the local frame can match the desired speed.

The next contribution is the modified ESC, which successfully reduces mutual attraction between the non-close pursuers to prioritize pursuing the targets and avoiding other pursuers that are quite close. Moreover, the distance between each pursuer or the distance between the pursuer and target can be controlled according to the given input.

In the PEG simulation with one evader, the MESC succeeded in controlling the pursuer to surround the evader. This was indicated by the distance data at the end of the simulation. The distance between all pursuers was approximately 5 m. According to the predetermined parameters, the distance between all pursuers and evaders was close to 3 m. In addition, the modified ESC manages to control the pursuer to avoid other pursuers, evaders, and obstacles. This can be revealed by the minimum distance between the pursuers of 2.071 m, the minimum distance between the pursuer and the evader of 1.954 m, and the minimum distance between the pursuer and all obstacles of 2.425 m. Thus, the pursuer never hits the evader, obstacles, or other pursuers.

Acknowledgment

The authors would like to express their gratitude to the Department of Electrical Engineering at the Institut Teknologi Sepuluh Nopember for allowing us to use the research facility.

References

[1] M. Aydın, E. Bostancı, M. Serdar Güzel, and N. Kanwal, “Multiagent Systems for 3D Reconstruction Applications,” in *Multi Agent Systems - Strategies and Applications*, R. López – Ruiz E d., IntechOpen, 2020, doi: 10.5772/intechopen.88460.

[2] S. Peng, S. Mukhopadhyay, R. Raje, M. Palakal, and J. Mostafa, “A comparison between single-agent and multi-agent classification of documents,” in *Proc. 15th Parallel and Distributed Processing Symposium*, pp. 935–944, 2001, doi: 10.1109/IPDPS.2001.925048.

[3] R. Ahmad, S. Ali and D. H. Kim, "A Multi-Agent system for documents classification," *2012 International Conference on Open Source Systems and Technologies*, pp. 28-32, 2012.

[4] A. Dorri, S. S. Kanhere, and R. Jurdak, “Multi-Agent Systems: A Survey,” *IEEE Access*, vol. 6, pp. 28573-28593, Dec. 2018.

[5] P. Benavidez and M. Jamshidi, “Mobile robot navigation and target tracking system,” in. *Proc. the 2011 6th International Conference on System of Systems Engineering*, pp. 299-304, Jun. 2011.

[6] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: taxonomy and survey,” *Autonomous Robots*, vol. 40, no. 4, pp. 729-760, 2016, doi: 10.1007/s10514-015-9491-7.

[7] R. Toyota and T. Namerikawa, “Formation control of multi-agent system considering obstacle avoidance,” in *Proc. 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan* (SICE), pp. 446-451, Sep. 2017.

[8] Z. Yang, Q. Zhang, and Z. Chen, “Formation Control of Multi-Agent Systems with Region Constraint,” *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/8481060.

[9] M. Lu, Y. Zou and S. Li, "Multi-agent formation control with obstacle avoidance based on receding horizon strategy," in *Proc. 2019 IEEE 15th International Conference on Control and Automation* (ICCA), pp. 1361-1366. Jun. 2019.

[10] Y. Chen, A. Singletary, and A. D. Ames, “Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach,” *IEEE Contr. Syst. Lett.*, vol. 5, no. 1, 2021.

[11] J. Y. Kuo, H. F. Yu, K. F. R. Liu, and F. W. Lee, “Multiagent Cooperative Learning Strategies for Pursuit-Evasion Games,” *Mathematical Problems in Engineering*, vol. 2015, 2015, doi: 10.1155/2015/964871.

[12] C. Yu, Y. Dong, Y. Li, and Y. Chen, “Distributed multi-agent deep reinforcement learning for cooperative multi-robot pursuit,” *The Journal of Engineering*, vol. 2020, no. 13, 2020, doi: 10.1049/joe.2019.1200.

[13] I. Ahmed and P. Kumam, “An Optimal Pursuit Differential Game Problem with One Evader and Many Pursuers,” *Mathematics*, vol. 7 (9), pp. 1–11, 2019, doi: 10.3390/math7090842.

[14] X. Fang, C. Wang, L. Xie, and J. Chen, “Cooperative Pursuit With Multi-Pursuer and One Faster Free-Moving Evader,” *IEEE Trans. Cybern.*, vol. 52, no. 3, 2022, doi: 10.1109/TCYB.2019.2958548.

[15] L. Liu, C. Luo, and F. Shen, “Multi-agent Formation Control with Target Tracking and Navigation,” in *Proc. IEEE International Conference on Information and Automation* (ICIA), pp. 98-103, Jul. 2017.

[16] R. Fierro and F. L. Lewis, “Control of a Nonholonomic Mobile Robot: Backstepping Kinematics Into Dynamics,” in *Proc. IEEE Conference on Decision and Control*, pp. 3805-3810, Dec. 1995.

[17] Omron, “Mobile Robots LD Series - Autonomous Mobile Robots (AMRs), self-mapping, self-navigating.,” *Data Sheets*, 2020. https://assets.omron.eu/downloads/datasheet/en/v10/i828_ld-series_mobile_robot_datasheet_en.pdf (accessed Jun. 02, 2021).