



Sharif University of Technology

Scientia Iranica

Transactions B: Mechanical Engineering

www.sciencedirect.com



Control of car-like (wheeled) multi robots for following and hunting a moving target

H. Sayyaadi^{*}, H. Kouhi, H. Salarieh

School of Mechanical Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-9567, Iran

Received 22 December 2010; revised 8 May 2011; accepted 12 June 2011

KEYWORDS

Multi robot;
Multi agents;
Following and hunting target;
Car-like;
Nonholonomic;
Kalman filter.

Abstract The main purpose of this paper is to design a decentralized controller for some car-like (wheeled) multi robots to follow and hunt a moving target. Considering geometric dimensions, mass and moment of inertia, robots are very similar to actual cars in which the outputs of the controller are steering and driving wheel torques. All robots are equipped with range and bearing sensors along with antenna, to communicate radio wave signals. A Kalman filter is implemented to estimate relative position, state variables of the target and state variables of other robots. The controller is designed to carry out the group maneuver of the system, based on the system dynamics analysis of inertial agents, as well as minimizing the norm of the error between desired and actual acceleration. A simulation study considering the group maneuver of four robots has been carried out, and the achieved results in positions, velocities and other relevant state variables of agents are depicted in the paper. The control torques of the steering system and driving wheels are derived and depicted appropriately. These results guarantee the performance of the addressed controller coupled with the Kalman filter, despite the existence of nonholonomics in dynamics, inertial behaviors, etc.

© 2011 Sharif University of Technology. Production and hosting by Elsevier B.V.

Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Over the last few years, multi robot control has attracted a great deal of attention from among engineers and researchers [1], the main reason being the fact that multi robots are able to perform tasks that a single robot is unable to do. To be more specific, they are robust to single agent failures, they need much less information than does a single robot, and they are all capable of covering a region, flocking, group searching, following and hunting moving targets, etc. For example, some previous research work, including fundamentals in Multi Agents and Networking, carried out by researchers, are:

1. The proportional task allocation problem; a combination of deployment and consensus problems was undertaken by Sayyaadi and Moarref [2];
2. Finite time consensus in directed switching network topologies, based on a time-delayed algorithm and communications, was undertaken by Sayyaadi and Dostmohammadian [3].

Considering the desired mission of a group operation, a proper robot should be chosen for each task. Among different types of robot, the predominant role of wheeled robots is undeniable, due to the fact that they can travel long distances while carrying equipment needed for group operations. Considering practical applications in industry, communications and transportation, car-like robots are known to be the most predominant among wheeled robots. Easy manufacturing, low maintenance costs, the ability to carry loads and heavy equipment, having adequate energy on board, better efficiency, the ability to maneuver at high speed, proper suspension systems, etc. are the main reasons for the superiority of car-like robots.

The importance of car-like robots essentially corresponds to the features of following and hunting moving targets. These features have a great deal of importance in military applications, security implementations and other scientific and practical fields. For example, they can be considered as security coverage operations for the leader of a group of agents.

^{*} Corresponding author.

E-mail address: sayyaadi@sharif.edu (H. Sayyaadi).



Nomenclature

Φ	Steering angle of robot
v	Front wheel velocity
I_C	Moment of inertia relative to geometric center of robot
L	Robot's length
r_w	Radius of robots wheel
τ_1	Steering torque of robot
Index d	Desired
k_ϕ	Constant of steering controller
$q = [x \ y]^T$	Symbol of geometric center of robot
$C(\theta), S(\theta)$	Symbol of sine and cosine
τ_{\max}, τ_{\min}	Maximum and minimum of robot's actuator torque
Index t	Symbol of moving target
k_t	Coefficient of target escape force
F_t	Escape force of moving target
B_t	Input matrix of target dynamic
X_i	Vector of robot i 's variables
X_{ij}	Vector of relative variables between two robots
X_{it}	Vector of relative variables between robot and target
R_{ij}	Relative distance between two agents
p_0	Signal strength attenuation in one meter distance
EKF	Extended Kalman Filter
w_p^i	Added noise to the signal strength sensor of robot i
$w_{R_{it}}^i$	Added noise to the distance sensor of robot i
w_{yr}	Added noise to y coordinate of the sensor of tracer
e_i	Gaussian white noise with characteristic
\hat{F}_t^i	Estimation of target escape force by robot i
k_p	Proportional control coefficient between two robots
k_p^t	Proportional control coefficient between a robot and target
R_t	Equilibrium distance between robot and target
Z^{ij}	Output of robot i sensor containing agent j data
Θ^{ij}	Output matrix of the sensor of robot i containing agent j data
R_1^{ij}	Covariance matrix of exerted noise on agent j which is used in the estimation algorithm of robot i
$k^{ij}(t)$	Kalman filter gain which is used to estimate state variables of agent j by robot i
θ	Body angle of robot
m	Mass of robot
d	Geometric distance of robot center from front and rear
β	Angle between steering and x axis
c_s	Constant of steering damping
τ_2	Driving torque of robot
Φ_d	Desired steering angle
dt	Step time for discretization
\ddot{q}_d, \dot{q}_d	Desired velocity and acceleration of robot's center
J	Cost function of robot performance
ρ	Coefficient of actuator filter
m_t	Mass of moving target
C_t	Damping of target dynamic
x_t, y_t	Coordinates of target center

A_t	State matrix of target dynamic
X_{Ci}	Vector of robot i center variable
X_t	Vector of target center variables
x_{ij}, y_{ij}	Relative position between robot i and j
ψ_{ij}	Viewing angle of agent j by robot i
α	Coefficient of signal strength attenuation
E	Expected value
w_ψ^i	Added noise to the view angle sensor of robot i
w_{xr}	Added noise to x coordinate of the sensor of tracer
σ	Standard deviation of noise w
$\phi_j(t)$	State transient matrix of robot j from inertia coordinates system point of view
F_{error}^i	Estimation error of target escape force
k_d^i	Derivative control coefficient between two robots
k_d^t	Derivative control coefficient between a robot and target
R_R	Equilibrium distance between two robots
\hat{Z}^{ij}	Output Estimation of the sensor of robot i containing agent j data
$\phi_{ij}(t)$	State transient matrix of agent j in coordinates systems attached to robot i
R_2^{ij}	Covariance matrix of the sensor of robot i containing agent j data
$p^{ij}(t)$	Covariance matrix of the state variables of agent j which are used in robot i estimation algorithm

In this paper, the robots are considered to behave non-holonomically in dynamics and require vigorous control, while considering their geometries and inertial effects too. Also, steering and driving torques are considered to be the output of controllers. The saturation and filtering effects of the actuators are also considered here. The robots in the group are controlled in such a way that each robot responds with an appropriate reaction, based on the control algorithm and the information passed down from other robots and the target (decentralized control). The force exerted on the moving target keeps the target away from the group. Each robot, in order to estimate the relative position and state variables of the other robots, is equipped with an antenna, which measures the direction and strength of the transmitted signals from other robots, and uses it for estimating the state variables and relative positions of moving targets from the range and bearing sensors. Using the output of these sensors and invoking the Kalman filter, each robot estimates the needed state variables of other agents and uses them in its control algorithm to operate the group maneuver. In the following, some work related to following and hunting a moving target is presented.

A control algorithm is designed for a unicycle-type robot in order to hunt a moving target, supposing that all information is at hand [4]. A task allocation method for multi-robot cooperative hunting behavior is proposed based on an improved auction algorithm [5]. Consider the problem of the navigation and guidance of a unicycle-type mobile robot towards a maneuvering target, based on measurements concerning only the distance from the robot to the target [6]. A hybrid control approach to the problem of steering a group of unicycle-type mobile robots to reach desired relative positions and orientations, with respect to a specific target and other group-mates, is referred to as the cooperative target enclosing problem [7]. By using a hybrid control approach, some unicycle-type robots have been controlled to follow and hunt a moving target. In this approach,

basic assumptions are that all information, such as speed and relative variables, are accessible [8]. Unicycle robots are controlled to enclose and capture a target. That work only uses local information gathered by on-board relative-position sensors assumed to be noisy, anisotropic, and unable to detect the identity of the measured object. Communication between robots is provided by limited-range transceivers [9]. The approach of co-operative hunting for multiple mobile targets by a multi-robot is addressed in [10], which divides the pursuing process into forming pursuing groups and capturing the targets. Some drawbacks of these works are as follows.

Robots are modeled as unicycle or as point masses in the plane. The kinematics of the robot is used for designing the controller. The target moves in a specific path and does not escape the robots. Positioning by image processing is not applicable at far distances, and some of them suppose all information is accessible.

Some studies related to car-like robot group maneuvers are addressed here. The formation of car-like multi robots and their change to another formation have been studied in [11]. The most important work regarding car-like robots was described in a doctoral dissertation [12]. In this research study, the main goal was to control multiple car-like robots following a leader; all the system moving in a certain geometrical shape. It was presumed that each robot had a leader and this configuration continues chain-like to the main leader. Therefore, the control algorithm for each robot was defined such that they follow the leader at a certain relative distance and angle [13]. The main leader, ahead of the group, was responsible for guidance. It was assumed that the main leader had a constant velocity; meanwhile, the design of the controller for some accelerated inertial agents had its own specific rules. The control algorithm was based on an optimization procedure in which the cost function, using numerical methods, uses a lot of CPU time, because of the huge amount of calculations. Moving through the obstacles and changing the formation is another research topic [14].

Integration of the wireless network signal strength data with vehicle sensor information, by means of a Kalman filter, is proposed, to estimate the relative position of each vehicle in a robot formation [15]. The WiFi (Wireless Fidelity) data consist of the Reception Signal Strength (RSS) and the angle of the maximal RSS with respect to robot orientation [16]. This estimator was not incorporated into the group movement algorithm and was defined for just one robot. In order to represent the control algorithm and design the estimator, only kinematic equations were used, and mass, moment of inertia and main inputs, which are torques, were neglected.

Based on the type of operation, each car-like robot needs a specific controller. The work undertaken in regard to controlling a robot is: parking a robot in a specific place, finding the best path for movement [17,18], and robot movement on a path with obstacles [19] etc.

There are few studies about hunting a moving target, and performing this operation using car-like robots is the goal of this present article. The target escapes from the robots and does not have any geometric and kinematic constraints; it moves freely and it is assumed that all necessary information is not available. Therefore, sensors and estimators are used. The design of an estimator is based on the dynamics equations of the robot. The control algorithm is designed in such a way that a series of robots or obstacles can be added to the group without interrupting the group maneuver. Because the behavior of a group of inertial agents is different from a group of non-inertial agents, in order to design the controller, the dynamics of a group

of inertial agents is explicated for the group of robots and the controller is designed based upon it. In multi agent systems and multi robot applications, it is very difficult to predict the path each one takes. The control algorithm used in this article is easy to apply and, compared to other methods, a huge amount of calculation is not needed. Using a decentralized controller using only local information, compared to previous work, the assumptions in this paper are more compatible with real ones. It is not necessary to design a controller for the robot body angle directly; it will be done indirectly and the algorithm is scalable. In this paper, the agent is used for both robot and target. This article includes an explanation of kinematics and the kinetics of the robot, the design of the controller for the robot, an explanation of the dynamics of the moving target, an explanation of the sensors, the design of the estimator for the robot and moving target, the design of the controller for the group of robots, an explanation of the indirect control of the body angle of the robot, simulation results, nomenclature and, finally, concluding remarks.

2. Kinematics and kinetics of the car-like robot

As seen in Figure 1(a), the robot used here is a car-like robot equipped with four wheels, while considering its geometries, mass and moment of inertia. The steering and driving torques on the front wheels are the outputs of the controller. In addition to operating control algorithms and defining torques, robots are sensory equipped and capable of measuring the steering angle, the body angle of the robot, wheel speed and, as a result, the velocity and acceleration of each point of themselves. Because each robot measures its body angle, it is able to be aware of the inertial coordinate in all positions. Measuring the body angle is done by using an INS [20], compass based sensory system.

Based on the theorems developed in the kinetics and kinematics of car vehicles, the steering angles of the two front wheels have a slight difference, in order to allow an instantaneous axis of rotation. These angle differences do not cause noticeable discrepancies between two and four wheel robot model equations. In order to obtain its motion equations, it is required to assume that the robot is two wheeled [19,21]. In Figure 1, θ is the body angle, Φ is the steering angle, v is the front wheel speed, m is the robot mass, c is the geometrical center of the robot, d is the distance between the geometrical center and the two wheels, I_c is the moment of inertia, with respect to the geometrical center, xy is the symbol of the inertial coordinate system, F_1 is the robot driving force (the reaction of the force that the wheels exert on the ground surface), F_2 is the force exerted on the robot (from the robot inertia and velocity constraint of the front wheel) and F_3 is the force exerted on the robot (from the robot inertia and velocity constraint of the rear wheel). It is also assumed that the steering angle constraint is $-\pi/2 \leq \Phi \leq \pi/2$. The mass and inertia of every wheel are neglected, and the vehicle moves on the horizontal plane. There is also no slippage between the wheels and the surface. Kinematic equations are derived, based on the above presumptions, and also assuming the robot to be rigid. Eqs. (1) and (2) define velocity constraints on front and rear wheels; this means that the linear and angular velocities of the robot have to be satisfied in these two:

$$\dot{x}_c \sin \beta - \dot{y}_c \cos \beta - d\dot{\theta} \cos \Phi = 0, \quad \beta = \theta + \Phi, \quad (1)$$

$$\dot{x}_c \sin \theta - \dot{y}_c \cos \theta + d\dot{\theta} = 0. \quad (2)$$

x_c and y_c are the Cartesian coordinates of the robot center. In this article, x , y , C and S are abbreviations used instead of x_c , y_c ,

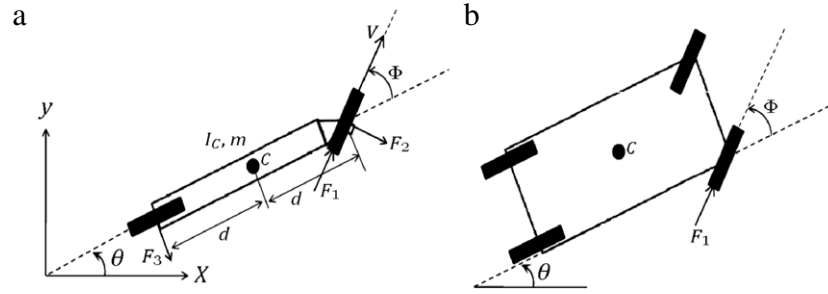


Figure 1: Schematic diagram of a four wheeled car-like robot (a), and simplified to two wheeled robot (b).

cos, sin. Using Eqs. (1) and (2), and using the newly mentioned abbreviations, the following equations are derived [22,23]:

$$a_1 = C\Phi C\theta - \frac{S\theta S\Phi}{2}, \quad a_2 = S\theta C\Phi + \frac{C\theta S\Phi}{2},$$

$$a_3 = \frac{S\Phi}{2d}, \quad (3)$$

$$\dot{x} = va_1, \quad (4)$$

$$\dot{y} = va_2, \quad (5)$$

$$\dot{\theta} = va_3. \quad (6)$$

Eqs. (4) and (5) are the velocity components of the robot center and Eq. (6) is the angular velocity of the robot, expressed as a function of front wheel speed (v) and angles, Φ , θ .

v and Φ are kinematic inputs and \dot{x} , \dot{y} and $\dot{\theta}$ are kinematic outputs. Based on these equations, one cannot find any value for v that simultaneously satisfies the three equations, $\dot{x}_d = va_1$, $\dot{y}_d = va_2$, $\dot{\theta}_d = va_3$, where d stands for the desired condition. In Eqs. (7) and (8), τ'_1 and τ'_2 are the main torques of steering and driving, r_w is the radius of the wheel and c_s is the damping factor of the steering system.

$$\dot{\Phi} = c_s \tau'_1 = \tau_1, \quad (7)$$

$$F_1 = \frac{\tau'_2}{r_w} = \tau_2. \quad (8)$$

In order to simplify these equations, the normalizing of parameters had been carried out. The inputs are $\dot{\Phi} = \tau_1$ and $F_1 = \tau_2$. Finally, by multiplying and dividing the results by r_w and c_s , the main torques are derived. The damping effects of the steering system are more than the inertia effects and therefore in Eq. (7), only the damping coefficient is included.

Using the above defined variables, putting them into Eq. (9) and making derivatives of Eqs. (4)–(6), acceleration equations are derived, as given by Eqs. (10)–(12). It is important to notice that because Eqs. (4)–(6) are valid in all time domains, it is also possible to differentiate them again in time.

$$a_4 = S\Phi C\theta + \frac{S\theta C\Phi}{2}, \quad a_5 = S\theta S\Phi - \frac{C\theta C\Phi}{2}, \quad (9)$$

$$\ddot{x} = \dot{v}a_1 - \tau_1 va_4 - v^2 a_3 a_2, \quad (10)$$

$$\ddot{y} = \dot{v}a_2 - \tau_1 va_5 + v^2 a_3 a_1, \quad (11)$$

$$\ddot{\theta} = \dot{v}a_3 + \tau_1 v \frac{C\Phi}{2d}. \quad (12)$$

Eqs. (10)–(12) express the relation between the derivative of the wheel speed and the torque of the steering with robot acceleration. These equations are used to derive kinetic motion equations. The Newton–Euler method is used in order to derive

the dynamics equations of the robot [23,20]. By defining some new parameters in Eq. (13), using Newton–Euler equations, Eq. (14) is obtained.

$$k_1 = \frac{I}{dm}, \quad a_6 = a_1^2 + a_2^2 + k_1 da_3^2,$$

$$a_7 = \frac{k_1 C\Phi}{2} a_3 - a_5 a_2 - a_4 a_1, \quad (13)$$

$$\begin{bmatrix} C\beta & S\beta & S\theta \\ S\beta & -C\beta & -C\theta \\ S\Phi & -C\Phi & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ k_1 \ddot{\theta} \end{bmatrix}. \quad (14)$$

Eq. (14) expresses the relation between forces and the acceleration of the robot. By replacing Eqs. (10)–(13) into Eq. (14), solving the system of equations and deriving F_1 , Eq. (15) is derived. Eq. (15) expresses the relation between driving torque exerted on the wheel, steering torque and derivative of the wheel speed.

$$\tau_2 = F_1 = m(\dot{v}a_6 + \tau_1 va_7). \quad (15)$$

Robot governing equations can be written in state space format as follows:

$$\begin{aligned} x_1 &= x, & x_2 &= \dot{x}, & x_3 &= y, & x_4 &= \dot{y}, \\ x_5 &= \theta, & x_6 &= \dot{\theta}, & x_7 &= v, & x_8 &= \Phi. \end{aligned} \quad (16)$$

And finally it is summarized as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_7^2 a_2 a_3 \\ x_4 \\ x_7^2 a_1 a_3 \\ x_6 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -x_7 \left(\frac{a_1 a_7}{a_6} + a_4 \right) & \frac{a_1}{ma_6} \\ 0 & 0 \\ -x_7 \left(\frac{a_2 a_7}{a_6} + a_5 \right) & \frac{a_2}{ma_6} \\ 0 & 0 \\ x_7 \left(\frac{C\Phi}{2d} - \frac{a_3 a_7}{a_6} \right) & \frac{a_3}{ma_6} \\ \frac{x_7 a_7}{a_6} & \frac{1}{ma_6} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \quad (17)$$

And finally it is summarized as:

$$\dot{X} = f(X, \tau),$$

$$X = [x_1 \ x_2 \ \dots \ x_8]^T, \quad \tau = [\tau_1 \ \tau_2]^T. \quad (18)$$

Eq. (18) is a continuous equation in the time domain. After discretization, the state space variables are derived, which are shown in (19).

$$X(t+1) = f(X(t), \tau(t))dt + X(t) = F(X(t), \tau(t)). \quad (19)$$

The main reason why Eq. (19) is defined is that it can be used in order to use the Kalman filter that estimates state variables. The discretization method used in Eq. (19) is in Euler format; it is acceptable to use the Runge–Kutta discretization method instead, which has better precision. Although it seems that the discretization error of the Euler method is greater because an estimator is used, its error is corrected continuously. Eq. (19) is only used in the discretized Kalman filter, and the equations of the real model are solved using a better method.

3. Representing a control algorithm for the robot

A nonholonomic system can achieve any configuration. Compared to controllable systems, there is no continuous and derivable state feedback to change the configuration of a nonholonomic system.

It means that it is required to have a feedback that depends on time and state variables. Feedback linearization of a nonholonomic system is almost impossible. Robot kinematics equations can transform into a “chain form”, which is controllable. But, reverse mapping that transforms the “chain form” equation to robot kinematics equations is not generally an easy task [24]. The “chain form” brings a very complex control algorithm, involving a huge amount of calculation. In this paper, a simple and obvious control strategy has been addressed in such a way that there is no need to use the “chain form”.

Considering the fact that this article aims to control a number of robots in order to hunt and loop around a target, the control of the center coordinates of each robot, $q = [x \ y]^T$, is adequate. The robot body angle controller is not designed directly and it is indirect and spontaneous. It is observed, as follows, that the body angle of the robot converges to a desired value that is physically acceptable, and it is because of the type of maneuver and the governing control equations.

In [25], the control of nonholonomic systems is explicitly explained. Consider the fact that robot constraints are not geometric, which means that there is no position limitation. It means that each point of the robot can travel to any arbitrary position (the system is reachable), but because of the velocity constraint, no point can travel in any arbitrary direction (the system is uncontrollable). It is not allowed to move in any arbitrary direction, except in cases where the desired movement is in agreement with the kinematic constraint equation. These special cases rarely happen.

For nonholonomic robots, it is not possible either to define the velocity of the robot in such a way that $\dot{q} = \dot{q}_d$, nor to define the torques in such a way that $\ddot{q} = \ddot{q}_d$.

\ddot{q}_d is the desired acceleration. As an example, in order to track q_t and \dot{q}_t by q and \dot{q} , the desired acceleration is designed in the form of $\ddot{q}_d = -k_d(\dot{q} - \dot{q}_t) - k_p(q - q_t) + \ddot{q}_t$ and the control signal is a function of \ddot{q}_d , \dot{q} and q . The design of the desired acceleration is explained in Section 8 of this article.

In [19], the Euclidean norm of real and desired velocity differences is optimized with a special control signal. In this article, the Euclidean norm of the real and desired accelerations is to be optimized to obtain the control action. There is another point of view in which the Euclidean norms of the desired and real coordinates are optimized. Considering the fact that the control torque at any time interval is effective on the next coordinates, optimizing just one term of the Euclidean norm coordinates does not lead to a satisfactory answer. A complete form of this algorithm will be described in the control part as MPC (Model Prediction Control). For a non-linear model,

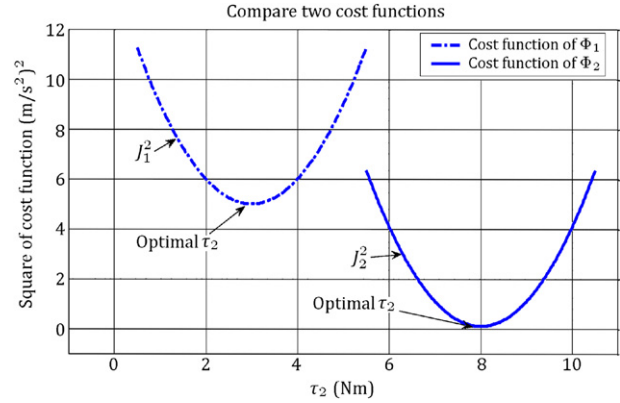


Figure 2: Qualitative view of cost function at two different steering angles.

in order to operate the MPC algorithm, future prediction, numerical optimization and a huge amount of calculation are needed. On the other hand, because the torque, at any time interval, defines the acceleration at that time, it is only possible to design a suitable controller by optimizing the Euclidean norm of difference between the desired and real acceleration. Generally, it cannot be deduced that this method is suitable for all systems. There is also a “parameter optimization” in this algorithm, which is easy, and there exists an analytical answer. Eq. (20) defines the cost function that needs to be optimized:

$$J = \|\ddot{q} - \ddot{q}_d\| \quad (20)$$

Figure 2 qualitatively shows the cost function (Eq. (20)) at two different steering angles. Meanwhile, steering angle Φ is in a proper position, Φ_2 , a suitable driving torque, τ_2 , can be found in order to even make the cost function zero. Therefore, an intuitive approach enables this algorithm to be used in the best possible way.

The design of the controller in this section is done under the presumption that variables \ddot{x}_d , \ddot{y}_d , \dot{x}_d and \dot{y}_d are known. The first two terms are the desired acceleration of the robot center. The method used to derive these two terms is explained in the multi robot control strategy, which will come later. The next two terms are the desired velocities, which are derived by integrating accelerations. By replacing Eqs. (10) and (11) into Eq. (20), Eqs. (21) and (22) are obtained:

$$J^2 = (\ddot{x} - \ddot{x}_d)^2 + (\ddot{y} - \ddot{y}_d)^2, \quad (21)$$

$$J^2 = (\dot{v}a_1 + v\dot{a}_1 - \ddot{x}_d)^2 + (\dot{v}a_2 + v\dot{a}_2 - \ddot{y}_d)^2. \quad (22)$$

Finally, the cost function will be in a form wherein two input variables, τ_1 and τ_2 , are observed; τ_1 will appear in the equation describing \dot{a}_j 's, and τ_2 is derived from \dot{v} and τ_1 (Eq. (15)). At first, \dot{v} is derived from Eq. (23) and then τ_2 by presuming τ_1 is known.

$$\frac{\partial J^2}{\partial \dot{v}} = 0, \quad (23)$$

$$\frac{\partial J^2}{\partial \tau_1} = 0. \quad (24)$$

τ_1 is obtained by using an intuitive method. An equation such as Eq. (24) can be used to evaluate τ_1 . By using such an equation for τ_1 , there is no control over angle Φ , because $\dot{\Phi} = \tau_1$. The steering angle should not exceed upper and lower angle limits, which are predefined by certain values. Therefore, τ_1 is obtained by a different method in order not to violate the steering angle

constraint. Using Eq. (22) and solving Eq. (23), the optimized value for \dot{v} is obtained as follows:

$$\dot{v} = \frac{\ddot{x}_d a_1 + \ddot{y}_d a_2}{a_8} + \frac{v(a_4 a_1 + a_5 a_2)}{a_8} \tau_1, \quad (25)$$

$$a_8 = a_1^2 + a_2^2.$$

Using Eqs. (15) and (25), the optimized value of τ_2 is also obtained.

$$\tau_2 = m a_6 \left(\frac{\ddot{x}_d a_1 + \ddot{y}_d a_2}{a_8} \right) + \tau_1 m v \left(\frac{a_6}{a_8} (a_4 a_1 + a_5 a_2) + a_7 \right). \quad (26)$$

In Eq. (26), τ_2 is defined in an optimized manner, as a function of τ_1 . In order to obtain τ_1 , it is just needed to define the desired $\dot{\Phi}$. As a result, τ_1 is obtained from equation $\tau_1 = \dot{\Phi}$. Eq. (27) shows the desired dynamics for $\dot{\Phi}$, in which the terms Φ_d (desired angle) and $\dot{\Phi}_d$ will be calculated. k_ϕ is constant gain and should be tuned. In order to calculate τ_1 , it is needed to determine the desired dynamic of Φ . As a result, τ_1 is calculated from equation $\tau_1 = \dot{\Phi}$.

$$\dot{\Phi} = -k_\phi (\Phi - \Phi_d) + \dot{\Phi}_d. \quad (27)$$

In order to calculate Φ_d , some efforts are made to find the best steering angle. The best angle is the angle by which the center of the robot is moved toward an arbitrary direction, \dot{x}_d and \dot{y}_d . By solving Eqs. (28) and (29) simultaneously and defining the new variables into Eq. (30), the best steering angle is obtained.

$$\dot{x}_d = v a_1, \quad (28)$$

$$\dot{y}_d = v a_2, \quad (29)$$

$$a_9 = \dot{y}_d C \theta - \dot{x}_d S \theta, \quad a_{10} = \dot{y}_d S \theta + \dot{x}_d C \theta. \quad (30)$$

Eq. (31) shows the desired steering angle, and the current one should converge to it.

$$\Phi_d = \tan^{-1} \left(2 \frac{a_9}{a_{10}} \right). \quad (31)$$

As seen, because \tan^{-1} function is used, the angle varies between $-\pi/2, \pi/2$ and satisfies the steering constraint. The steering angle should correctly follow the desired angle and therefore $\dot{\Phi}_d$ should appear in the dynamics angle equation, as stated in Eq. (27). By defining new variables into Eq. (32) and taking derivatives from Eqs. (30) and (31), $\dot{\Phi}_d$ is calculated by Eq. (34).

$$b_1 = \ddot{y}_d C \theta - \ddot{x}_d S \theta, \quad b_2 = -\dot{y}_d S \theta - \dot{x}_d C \theta, \quad (32)$$

$$b_3 = \ddot{y}_d S \theta + \ddot{x}_d C \theta, \quad b_4 = \dot{y}_d C \theta - \dot{x}_d S \theta, \quad (33)$$

$$\dot{a}_9 = b_1 + v a_3 b_2, \quad \dot{a}_{10} = b_3 + v a_3 b_4, \quad (33)$$

$$\dot{\Phi}_d = 2 \frac{\dot{a}_9 a_{10} - \dot{a}_{10} a_9}{a_{10}^2 + 4 a_9^2}. \quad (34)$$

Eq. (34) shows the changing rate of the desired angle.

$$\tau_1 = \dot{\Phi} = -k_\phi (\Phi - \Phi_d) + \dot{\Phi}_d. \quad (35)$$

Eq. (35) shows the value of the steering torque and Eq. (36) is the closed loop equation of error angle in which by choosing a positive k_ϕ , the error converges to zero.

$$\dot{e} + k_\phi e = 0, \quad e = \Phi - \Phi_d. \quad (36)$$

Because the differential equation of the steering angle is not of the second order, there is no overshoot. As a result, Φ

tracks Φ_d without overshoot and the steering angle constraint is not violated. Because of the properties of \tan^{-1} , angle Φ_d discontinuously changes from $\pm\pi/2$ to $\mp\pi/2$ and due to the filter defined by Eq. (35), angle Φ converges to angle Φ_d mildly. Sudden changes of Φ_d do not lead to sudden changes in $\dot{\Phi}_d$, and $\dot{\Phi}_d$ is still a suitable input for Eq. (35).

Considering the fact that the goal of this article is to control a group of robots, this algorithm is applied to a group of robots. One of the advantages of this algorithm is that it performs group robot control easily and the results are appropriate. Therefore, complicated algorithms are not required to achieve this goal.

There is a question that needs to be answered here, and that is, “why it is endeavored to make a control signal written in the form of a torque instead of $\dot{\Phi}$ and \dot{v} , which are kinematics parameters that can be used as the control signal?”

There are two main answers to this question, which are:

1. Because of robot inertia, it is adequate to analyze torque, and a more clear meaning is observed from this method rather than using $\dot{\Phi}$ and \dot{v} .
2. The saturation and filtering effects of the actuators on ideal torques should be considered and then the actual torque is applied to the model; this makes the problem look more realistic.

In much work done earlier using kinematic controllers, the abilities of actuators to supply control signals were not investigated; this can be a drawback.

When the desired torque is very large, the real actuator can only apply its maximum output, or when the input signal has many oscillations, a mechanical system has a limited capability to follow these commands because of its inertia. In this article, the effects of filtering and saturation of the actuator are investigated to show that although the actuator is not ideal, it can be controlled properly. Eq. (37) explains the saturation effects in actuators. Eq. (38) shows the inability of a non-ideal actuator to create control signals with high frequency.

$$\tau = \begin{cases} u & |u| \leq \tau_{\max} \\ \tau_{\max} \text{sign}(u) & |u| \geq \tau_{\max} \end{cases} \quad (37)$$

$$\tau(s) = \frac{u(s)}{\rho s + 1} \quad (38)$$

u is an ideal signal, τ is the output of a non-ideal actuator and ρ in Eq. (38) is the reverse of the filter pole, and when it is increased, the output ability to follow step input decreases. Eq. (39) shows the discrete form of the filter. dt is the step time for discretization.

$$\tau(t+1) = \tau(t) e^{-\frac{dt}{\rho}} + \left(u(t+1) + u(t) e^{-\frac{dt}{\rho}} \right) \frac{dt}{2\rho}. \quad (39)$$

4. Kinematics and kinetics of the moving target

Figure 3 shows that the moving target has mass and size and it is assumed that it can move to any direction without any geometric constraint.

Eq. (40) shows the dynamics of the target, in which m_t is the mass, k_t is the escaping force coefficient and C_t is the damping constant of the target in which the t index stands for the target.

$$m_t \begin{bmatrix} \ddot{x}_t & \ddot{y}_t \end{bmatrix}^T = k_t F_t - C_t \begin{bmatrix} \dot{x}_t & \dot{y}_t \end{bmatrix}^T. \quad (40)$$

The escaping force increases, such that the distance between the target and the robot decreases, making the target escape with a stronger force as the robot approaches. When a robot

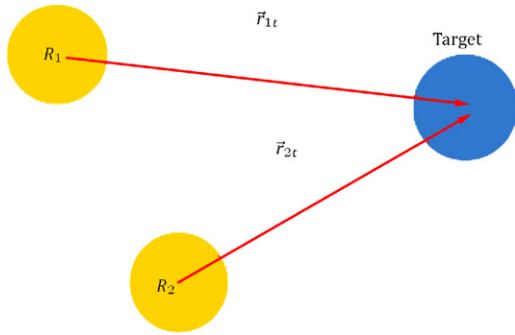


Figure 3: Target position relative to the two following robots.

is closer to the target than other robots, it exerts a stronger repulsive force on the target. The damping coefficient is a factor that does not allow the velocity to increase; this property is usually observed in natural and artificial systems. Eq. (41) shows how to evaluate the target escape force:

$$F_t = \sum_{i=1}^n \frac{\vec{e}_{it}}{R_{it}} = \sum_{i=1}^n \frac{\vec{r}_{it}}{\|\vec{r}_{it}\|^2}. \quad (41)$$

In Eq. (42), r_{it} is the relative position vector of robot i compared to the target. In this article, it is presumed that the target's physical properties are pre-identified.

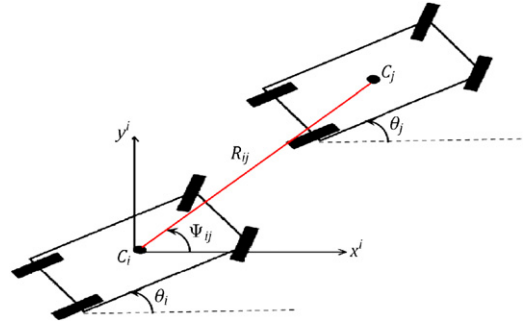
$$\vec{r}_{it} = [x_t - x_i \quad y_t - y_i]^T = [x_{it} \quad y_{it}]^T. \quad (42)$$

5. Governing presumptions about sensors

The moving target can identify its position relative to the center of the robots and calculate the escape force from the group based on it. Robots include sensors that can measure steering angle, body angle and front wheel speeds precisely. The robot torque values can also be known using the control algorithm. Robots transmit their own data, such as torques, steering angles, robot angles and front wheel speeds, to other robots using radio transmission facilities [16].

In order to determine the large relative distances, radio signals are used instead of image processing. The relative positions of the robots are obtained using radio signals. The transmitted signals of each robot specifically belong to that robot, and each has its own frequency different from other robots [15,26]. The receiving robot receives the direction of the signal and signal strength with some added noise. It also identifies the robot that has sent the signal. Positioning using the strength and signal direction is explained in [27]. The receiving robot receives information, such as torques, steering angles, robot angles and front wheel speeds, including some small noise, but most of this noise is from the signal strength rather than the other transmitted information, such as body angle, etc. The further the two robots are away from each other, the further the RSS (reception signal strength) decreases between them; this can be a good reference to calculate their relative positions [28].

Robots can measure the view angle and relative distance of the moving target using image processing from some noisy captured images [29,30]. But, this estimation would suffer from two major difficulties, including far distance and a disrupted image; for example, when another robot is in-between the target and the observer robot. It is possible to fuse the information obtained by the other sensors to solve such

Figure 4: Schematic diagram showing signal transmission between transmitter robot j and receiving robot i .

problems, or use the proposed method in this article. In this method, it is presumed that a tracer, such as flying equipment, is used to find out the relative distance of the target and the robots. The added noise on tracer information is presumed to be more dominant than the noise entered into the image processing algorithm used by the robots themselves; however, both of them are used to find the best states estimation.

6. Design of estimators to estimate robot state variables

In this section, relative position and state variables estimation by the estimator is investigated. In Figure 4, an arbitrary robot, j , and an estimator robot, i , are shown, respectively. An estimator robot should estimate state variables and the relative position of the center of robot j , using the received signal from robot j 's antenna. In this article, the Kalman filter is used as an optimum estimator.

In Figure 4, the distance between the center of two robots, R_{ij} , and the receiving angle of the signal, Ψ_{ij} , are shown. Robot j transmits its own information to robot i , including some noise. Robot i distinguishes the coming signal from robot j by distinguishing the frequency of that signal. Then, using this information, it will operate the estimation algorithm for robot j . The velocity and position variables of robot j are measured with respect to the $x^i y^i$ coordinate system. In Eq. (43), the first term expresses the attenuation of the signal strength (dB) transmitted by robot j [16,28]. p_0 is attenuation of the signal strength one meter away from the source and α is the strength attenuation coefficient. Robot i receives the transmitted signal mixed by noise w_p^i . This noise in the received signal depends on many factors which are explained in [31]. w_p^i is white Gaussian noise with $(0, \sigma_p^2)$ characteristics. The second parameter in Eq. (43) defines the angle of the received signal, which is mixed by white Gaussian noise, with $(0, \sigma_\psi^2)$.

$$z_1^{ij} = \begin{bmatrix} p_0 + \alpha \ln(R_{ij}) + w_p^i \\ \Psi_{ij} + w_\psi^i \end{bmatrix}. \quad (43)$$

The second input data in Eq. (43) is the angle, since its partial derivations are needed in the Extended Kalman Filter (EKF) algorithm. If used as stated above, there would be many difficulties in numerical calculation. In such a situation, it is possible to transform the measured data, using the proposed equation given by Eq. (44).

$$z_1^{ij} = \begin{bmatrix} \cos(\Psi_{ij} + w_\psi^i)(p_0 + \alpha \ln(R_{ij}) + w_p^i) \\ \sin(\Psi_{ij} + w_\psi^i)(p_0 + \alpha \ln(R_{ij}) + w_p^i) \end{bmatrix}. \quad (44)$$

By these changes, undoubtedly, the physical concept of the problem is unaltered, which does not mean that the

characteristics of the sensor are changed; just that the mathematical representation of the measurement is changed. Eq. (44) is the final output of the sensor, which is used in the EKF algorithm. But in order to calculate the output matrix and sensor covariance, it is assumed that the exerted noise on angle is so small that its sine function can be approximated linearly by the angle of its argument (Eq. (45)):

$$\begin{aligned}\cos(\Psi_{ij} + w_{\psi}^i) &= \cos(\Psi_{ij}) - \sin(\Psi_{ij}) w_{\psi}^i, \\ \sin(\Psi_{ij} + w_{\psi}^i) &= \sin(\Psi_{ij}) + \cos(\Psi_{ij}) w_{\psi}^i.\end{aligned}\quad (45)$$

These procedures are only done to calculate the output matrix and covariance. By presuming that angle noise is sometimes not small enough, the calculated noise covariance and output matrix may have a little error and in this case, the Kalman filter is not optimal anymore, but is still a proper estimator. The covariance of the noise which comes in the actual situation of the sensor is different from the calculated covariance in theory. Efforts have been made to relate the sine and cosine values to relative position coordinates. In Eq. (46), x_{ij} and y_{ij} are relative position coordinates between robots i and j :

$$\begin{aligned}\cos(\Psi_{ij}) &= \frac{x_{ij}}{R_{ij}}, \quad \sin(\Psi_{ij}) = \frac{y_{ij}}{R_{ij}}, \\ R_{ij} &= \text{norm}(x_{ij}, y_{ij}).\end{aligned}\quad (46)$$

By defining new variables into Eq. (47) and using them, Eq. (44) is expressed as Eq. (48).

$$\begin{aligned}H^{ij} &= \begin{bmatrix} \frac{x_{ij}}{R_{ij}}(p_0 + \alpha \ln(R_{ij})) \\ \frac{y_{ij}}{R_{ij}}(p_0 + \alpha \ln(R_{ij})) \end{bmatrix}, \\ Q_1^{ij} &= \begin{bmatrix} -\frac{y_{ij}}{R_{ij}}(p_0 + \alpha \ln(R_{ij})) & \frac{x_{ij}}{R_{ij}} \\ \frac{x_{ij}}{R_{ij}}(p_0 + \alpha \ln(R_{ij})) & \frac{y_{ij}}{R_{ij}} \end{bmatrix} \begin{bmatrix} \sigma_{\psi} & 0 \\ 0 & \sigma_p \end{bmatrix}, \\ Q_2^{ij} &= \begin{bmatrix} -\frac{y_{ij}}{R_{ij}} \\ \frac{x_{ij}}{R_{ij}} \end{bmatrix} \sigma_{\psi} \sigma_p,\end{aligned}\quad (47)$$

$$z_1^{ij} = H^{ij} + Q_1^{ij} [e_1 \quad e_2]^T + Q_2^{ij} e_1 e_2. \quad (48)$$

In Eq. (48), e_1 and e_2 are white Gaussian noise with (0, 1) or $w_{\psi}^i = \sigma_{\psi} e_1$, $w_p^i = \sigma_p e_2$. In order to calculate the output matrix and use it in the EKF, the relative variables between two robots should be identified. State variables of robot j are in the following equation:

$$X_j = [x_j \quad \dot{x}_j \quad y_j \quad \dot{y}_j \quad \theta_j \quad \dot{\theta}_j \quad v_j \quad \Phi_j]^T. \quad (49)$$

In this article, relative variables are the position and velocity of the geometrical centers of two robots, with respect to each other. It is not needed to consider the relative state of the last four terms in Eq. (49). The last four terms are used in an absolute way (it means that robot j is observed from the attached coordinates system to robot i). As a result, using Eqs. (49) and (50), relative variables of robot j to robot i are defined as Eq. (51). The relative variables are state variables of the estimator ($X_{\text{estimator}} = X_{ij}$).

$$X_{Ci} = [x_i \quad \dot{x}_i \quad y_i \quad \dot{y}_i \quad 0 \quad 0 \quad 0 \quad 0]^T, \quad (50)$$

$$X_{ij} = X_j - X_{Ci}. \quad (51)$$

After defining the relative variables between two robots, the output matrix of the first sensor is calculated (Eq. (52)):

$$\begin{aligned}\Theta_1^{ij} &= \frac{\partial(H^{ij})}{\partial(X_{ij})} \Big|_{\hat{x}_{ij}} = \frac{1}{R_{ij}^3} \\ &\times \begin{bmatrix} y_{ij}^2(p_0 + \alpha \ln(R_{ij})) + \alpha x_{ij}^2 & 0 \\ -x_{ij}y_{ij}(p_0 + \alpha(\ln(R_{ij}) - 1)) & 0 \\ -x_{ij}y_{ij}(p_0 + \alpha(\ln(R_{ij}) - 1)) & 0_{1 \times 5} \\ x_{ij}^2(p_0 + \alpha \ln(R_{ij})) + \alpha y_{ij}^2 & 0_{1 \times 5} \end{bmatrix} \Big|_{\hat{x}_{ij}}.\end{aligned}\quad (52)$$

Considering the fact that the body angle, steering angle and wheel speed of robot j is transmitted to robot i , the output of the second sensor is in the form of Eq. (53) in which e_i is white Gaussian noise with (0, 1) characteristics. This information is still mixed by some little noise, but efforts have been made in this article to filter this noise away in an optimized manner.

$$\begin{aligned}z_2^{ij} &= \begin{bmatrix} \theta_j \\ v_j \\ \Phi_j \end{bmatrix} + \sigma_{\theta v \Phi} \begin{bmatrix} e_3 \\ e_4 \\ e_5 \end{bmatrix}, \\ \sigma_{\theta v \Phi} &= \begin{bmatrix} \sigma_{\theta} & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_{\Phi} \end{bmatrix}.\end{aligned}\quad (53)$$

The noise in transmitted data is presumed to be very small, compared to the noise in the signal strength and received signal direction. By using Eq. (53), the output matrix for the EKF is obtained by Eq. (54):

$$\begin{aligned}\Theta_{\theta v \Phi} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \Theta_2^{ij} &= [0]_{3 \times 4} \quad \Theta_{\theta v \Phi}.\end{aligned}\quad (54)$$

In Eq. (55), the outputs of all sensors are expressed simultaneously to calculate the covariance of the sensor.

$$Z^{ij} = \begin{bmatrix} H^{ij} \\ \Theta_2^{ij} X_{ij} \end{bmatrix} + \begin{bmatrix} Q_1^{ij} & 0_{2 \times 3} \\ 0_{3 \times 2} & \sigma_{\theta v \Phi} \end{bmatrix} [e_{5 \times 1}] + \begin{bmatrix} Q_2^{ij} \\ 0_{3 \times 1} \end{bmatrix} e_1 e_2. \quad (55)$$

Eq. (56) is the final output matrix.

$$\Theta^{ij} = [\Theta_1^{ij} \quad \Theta_2^{ij}]^T. \quad (56)$$

Whenever it is multiplied by X_{ij} , the calculated output in the Kalman algorithm is a linearized form. Eq. (57) expresses the governing rules for random variables used in this paper.

$$\begin{aligned}e_i &\sim (0, 1), \quad E(e_i^2) = \sigma_i^2 = 1, \\ E(e_i^2 e_j^2) &= \sigma_i^2 \sigma_j^2 = 1, \quad E(e_i e_j) = 0, \quad i \neq j.\end{aligned}\quad (57)$$

By using Eq. (57), the matrix of covariance of the sensor noise is transformed into Eq. (58):

$$R_2^{ij} = \begin{bmatrix} \hat{Q}_1^{ij} \hat{Q}_1^{ijT} & 0 \\ 0 & \sigma_{\theta v \Phi} \sigma_{\theta v \Phi}^T \end{bmatrix} + \begin{bmatrix} \hat{Q}_2^{ij} \hat{Q}_2^{ijT} & 0 \\ 0 & 0 \end{bmatrix}. \quad (58)$$

It should also be noticed that robot i uses the estimated values to calculate covariance. Considering the fact that the existing disturbance on the robot dynamic is presumed to be zero, the noise covariance matrix of the model is zero (Eq. (59)):

$$R_1^{ij} = [0_{8 \times 8}]. \quad (59)$$

In order to perform the EKF algorithm, the estimation of the sensor output is needed; Eq. (60) is the estimation of the output in which the estimated relative variables are used:

$$\hat{Z}^{ij} = [\hat{H}^{ij} \quad \Theta_2^{ij} \hat{X}_{ij}]^T. \quad (60)$$

Using Eq. (19), the Jacobian matrix of robot j is obtained.

$$\begin{aligned} \phi_j(t) &= \left. \frac{\partial F(X_j(t), \tau(t))}{\partial (X_j(t))} \right|_{X_j(t)} \\ &= \left. \frac{\partial f(X_j(t), \tau(t))}{\partial (X_j(t))} \right|_{X_j(t)} dt + I_{8 \times 8}. \end{aligned} \quad (61)$$

This matrix is based on the assumption that robot j is observed from an inertia coordinate system. But the point that should be noted here is that the Jacobian matrix of robot j observed from the local coordinate system attached to robot i and the global coordinate system, differ from each other. But there is a relation between these two which needs to be defined. From Eqs. (19), (50) and (51), Eq. (62) is derived:

$$\begin{aligned} X(t+1) &= X_{ij}(t+1) + X_{Ci}(t+1), \\ X_{ij}(t+1) &= f(X_{ij}(t) + X_{Ci}(t), \tau(t)) dt + X_{ij}(t) \\ &\quad + X_{Ci}(t) - X_{Ci}(t+1). \end{aligned} \quad (62)$$

The above equation reveals the governing dynamics of relative variables (Eq. (51)). Eq. (63) is obtained from Eqs. (19) and (51).

$$f(X_{ij}(t) + X_{Ci}(t), \tau(t)) = f(X_{ij}(t), \tau(t)) + f(X_{Ci}(t), \tau(t)). \quad (63)$$

From Eqs. (19) and (51) and using the Eulerian discretization, Eq. (64) is derived:

$$\begin{aligned} f(X_{Ci}(t), \tau(t)) &= [\dot{x}_i \quad 0 \quad \dot{y}_i \quad 0 \quad 0_{4 \times 1}]^T, \\ f(X_{Ci}(t), \tau(t)) dt &= [x_i(t+1) - x_i(t) \quad 0 \quad y_i(t+1) - y_i(t) \quad 0 \quad 0_{4 \times 1}]^T. \end{aligned} \quad (64)$$

By substituting Eq. (63) into Eq. (62) and using Eqs. (64), (65) is derived:

$$\begin{aligned} f(X_{Ci}(t), \tau(t)) dt + X_{Ci}(t) - X_{Ci}(t+1) &= -[0 \quad \dot{x}_i(t+1) - \dot{x}_i(t) \quad 0 \quad \dot{y}_i(t+1) - \dot{y}_i(t) \quad 0_{4 \times 1}]^T \\ &= -[0 \quad \ddot{x}_i(t) \quad 0 \quad \ddot{y}_i(t) \quad 0_{4 \times 1}]^T dt \triangleq G(\ddot{x}_i(t), \ddot{y}_i(t)). \end{aligned} \quad (65)$$

Using Eqs. (62) and (65), Eq. (66), which is another form of Eq. (62), is derived, which shows that the dynamic of relative variables is not a function of the absolute position of any robots.

$$X_{ij}(t+1) = f(X_{ij}(t), \tau(t)) dt + X_{ij}(t) + G(\ddot{x}_i(t), \ddot{y}_i(t)). \quad (66)$$

By knowing the dynamics of relative variables, the Jacobian matrix of robot j , from the viewpoint of robot i , is calculated by the following equation:

$$\begin{aligned} \phi_{ij}(t) &= \left. \frac{\partial (X_{ij}(t+1))}{\partial X_{ij}(t)} \right|_{\hat{X}_{ij}} \\ &= \left. \frac{\partial (f(X_{ij}(t), \tau(t)) dt + X_{ij}(t) + G(\ddot{x}_i(t), \ddot{y}_i(t)))}{\partial X_{ij}(t)} \right|_{\hat{X}_{ij}} \\ &= \left. \frac{\partial f(X_{ij}(t), \tau(t))}{\partial X_{ij}(t)} \right|_{\hat{X}_{ij}} dt + I_{8 \times 8} \\ &= \left. \frac{\partial f(X_j(t), \tau(t))}{\partial X_j(t)} \right|_{\hat{X}_{ij}} dt + I_{8 \times 8} = \phi_j(t)|_{X_j = \hat{X}_{ij}}. \end{aligned} \quad (67)$$

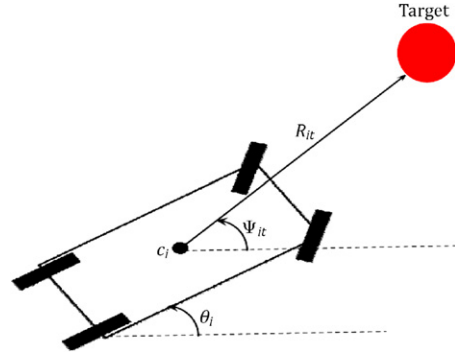


Figure 5: Schematic diagram showing range and bearing between robot i and the target.

As a result, in order to calculate the Jacobian matrix, it is enough to replace relative variables (Eq. (51)) into absolute variables of the absolute Jacobian matrix (Eq. (61)).

Eq. (68) calculates the noise on the received torque signal. This noise is small, like the noise of the steering angle, body angle and front wheel speed of robot j .

$$[\tilde{\tau}_1 \quad \tilde{\tau}_2]^T = [\tau_1 \quad \tau_2]^T + v_\tau. \quad (68)$$

Since that noise is small, whenever the torque noise passes from the actuator filtering, it is completely filtered, and the output of the actuator for such a situation is like the way in which the torque has entered the actuator without any noise. Eqs. (69)–(72) are Kalman filter equations for estimating the relative vector defined in Eq. (51). Robot i implements this algorithm to estimate relative variables between robot j and itself.

$$k^{ij}(t) = \phi_{ij}(t) p^{ij}(t) \Theta^{ijT}(t) \left[R_2^{ij}(t) + \Theta^{ij}(t) p^{ij}(t) \Theta^{ijT}(t) \right]^{-1}, \quad (69)$$

$$p^{ij}(t+1) = [\phi_{ij}(t) - k^{ij}(t) \Theta^{ij}(t)] p^{ij}(t) \phi_{ij}^T(t) + R_1^{ij}(t), \quad (70)$$

$$\begin{aligned} \hat{X}_{ij}(t+1) &= \left(f(\hat{X}_{ij}(t) + X_{Ci}(t), \tau(t)) - \dot{X}_{Ci}(t) \right) dt \\ &\quad + \hat{X}_{ij}(t) + k^{ij}(t) \left(Z^{ij}(t) - \hat{Z}^{ij}(t) \right), \end{aligned} \quad (71)$$

$$Z^{ij} = [z_1^{ij} \quad z_2^{ij}]^T. \quad (72)$$

As seen earlier, it is not needed to bring X_{Ci} position terms into Eq. (71). f in the above equation is the same as f in Eq. (18). The outputs which are used in Eq. (72) are real sensor outputs or those introduced in Eqs. (44) and (53).

7. Design of the estimator to estimate velocity and relative position of the target

Estimation of the relative position and velocity of the moving target by the robot will be explained here. As expressed in the previous section, using image processing, robots are able to measure the distance and viewing angle of the target mixed with noise [29,30]. It is also possible to measure the relative distance of the robot and the target with greater noise using another tracer, such as flying equipment. This method is useful when the view of the target is obstructed. In normal situations, these two data are mixed to find an optimized estimation by using a Kalman filter (Figure 5).

Eq. (73) is the real sensor output, which is transformed into Eq. (74) in order to solve computational problems. The same procedure was undertaken in the previous section of the article.

$$z_1^{it} = \begin{bmatrix} \Psi_{it} + w_{\Psi}^i \\ R_{it} + w_{R_{it}}^i \end{bmatrix}, \quad (73)$$

$$z_1^{it} = \begin{bmatrix} (R_{it} + w_{R_{it}}^i) \cos(\Psi_{it} + w_{\Psi}^i) \\ (R_{it} + w_{R_{it}}^i) \sin(\Psi_{it} + w_{\Psi}^i) \end{bmatrix}. \quad (74)$$

Eq. (74) is just a mathematical representation of the sensor output and it does not change the nature of the sensor. Eq. (75) is for the state variables of the moving target and Eq. (76) is for relative variables between the target and robot i .

$$X_t = [x_t \quad \dot{x}_t \quad y_t \quad \dot{y}_t]^T, \quad (75)$$

$$X_{it} = X_t - X_{Ci}^{1:4} = [x_{it} \quad \dot{x}_{it} \quad y_{it} \quad \dot{y}_{it}]^T. \quad (76)$$

Eq. (77) is for the tracer sensor output. It is assumed that this information is sent to the robots by the flying equipment. The noise in this data is greater than the noise of the distance and view angle of the robot sensor. As discussed in the previous section of this article, when the view of the target is obstructed, these robots can use the information obtained from robots that can view the target. In this case, it is not needed to use robot communication; the case studied here is different and it is assumed based on the tracer data.

$$z_2^{it} = \begin{bmatrix} x_{it} + w_{xr} \\ y_{it} + w_{yr} \end{bmatrix}. \quad (77)$$

In Eq. (78), noise characteristics and their relation are explained. These equations are used to calculate the covariance matrix.

$$\begin{aligned} w_{\Psi}^i &\equiv (0, \sigma_{\Psi}^2), & w_R^i &\equiv (0, \sigma_R^2), \\ w_{xr} &\equiv (0, \sigma_{xr}^2), & w_{yr} &\equiv (0, \sigma_{yr}^2), \\ w_s^i &= \sigma_s e_s, & e_s &\equiv (0, 1), \\ Ee_1e_2 &= 0, & Ee_1^2e_2^2 &= 1. \end{aligned} \quad (78)$$

By defining new variables in Eq. (79), the output of the sensors transform into Eq. (80).

$$\begin{aligned} H^{it} &= \begin{bmatrix} x_{it} \\ y_{it} \\ x_{it} \\ y_{it} \end{bmatrix}, \\ Q_1^{it} &= \begin{bmatrix} -y_{it} & \frac{x_{it}}{R} & 0 & 0 \\ x_{it} & \frac{y_{it}}{R_{it}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{\Psi} & 0 & 0 & 0 \\ 0 & \sigma_R & 0 & 0 \\ 0 & 0 & \sigma_x & 0 \\ 0 & 0 & 0 & \sigma_y \end{bmatrix}, \\ Q_2^{it} &= \frac{\sigma_{\Psi}\sigma_R}{R_{it}} \begin{bmatrix} -y_{it} \\ x_{it} \\ 0 \\ 0 \end{bmatrix}, & e &= \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}, \end{aligned} \quad (79)$$

$$Z^{it} = [z_1^{it} \quad z_2^{it}]^T = H^{it} + Q_1^{it}e + Q_2^{it}e_1e_2. \quad (80)$$

Eq. (81) is for the output matrix to be used in the EKF and Eq. (82) is the noise covariance matrix.

$$\Theta_1^{it} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \Theta^{it} = \begin{bmatrix} \Theta_1^{it} \\ \Theta_1^{it} \end{bmatrix} \quad (81)$$

$$R_2^{it} = \hat{Q}_1^{it}\hat{Q}_1^{iT} + \hat{Q}_2^{it}\hat{Q}_2^{iT}. \quad (82)$$

Using Eqs. (40) and (75) and defining matrix (Eq. (83)), the state space variables of the moving target transform into Eq. (84).

$$\begin{aligned} A_t &= \frac{1}{m_t} \begin{bmatrix} 0 & m_t & 0 & 0 \\ 0 & -C_t & 0 & 0 \\ 0 & 0 & 0 & m_t \\ 0 & 0 & 0 & -C_t \end{bmatrix}, \\ B_t &= \frac{1}{m_t} \begin{bmatrix} 0 & 0 \\ k_t & 0 \\ 0 & 0 \\ 0 & k_t \end{bmatrix}, \end{aligned} \quad (83)$$

$$\dot{X}_t = A_t X_t + B_t F_t. \quad (84)$$

In Eq. (85), the transient state matrix is observed.

$$\phi_t = A_t dt + I_{4 \times 4}. \quad (85)$$

Eq. (86) is the discretized format of the moving target dynamics.

$$X_t(t+1) = \phi_t X_t(t) + B_t F_t dt. \quad (86)$$

Eq. (87) is the dynamics of the relative variables of the moving target to robot i .

$$X_{it}(t+1) = \phi_t (X_{it}(t) + X_{Ci}(t)) - X_{Ci}(t+1) + B_t F_t dt. \quad (87)$$

In Eq. (87), it seems that the dynamics of the relative variables are a function of position variables, X_{Ci} . But, because of the nature of matrix A_t (having two zero columns), the relative variables are not a function of the absolute position of the robot. Eq. (88) shows how it is proven.

$$\begin{aligned} \phi_t (X_{Ci}(t)) - X_{Ci}(t+1) &= A_t X_{Ci}(t) dt + X_{Ci}(t) - X_{Ci}(t+1) \\ &= A_t X_{Ci}(t) dt - \dot{X}_{Ci}(t) dt \\ &\triangleq (h(\dot{x}_i, \dot{y}_i) - \dot{X}_{Ci}(t)) dt \\ &\Rightarrow X_{it}(t+1) = \phi_t X_{it}(t) + B_t F_t dt \\ &\quad + (h(\dot{x}_i, \dot{y}_i) - \dot{X}_{Ci}(t)) dt. \end{aligned} \quad (88)$$

Eq. (87) changed to Eq. (88). Considering Eq. (88), it is deduced that ϕ_t matrix is the same as Eq. (85), no matter what the coordinate system is. Eq. (89) shows the output estimation:

$$\hat{Z}^{it} = \hat{H}^{it}. \quad (89)$$

By presuming that the moving target knows the exact relative distance, it is able to calculate the escape force. But robots use relative position estimation, hence F_t is different from \hat{F}_t^i . Eq. (90) shows how \hat{F}_t^i is calculated by robot i in which the estimation values which have errors are used.

$$\begin{aligned} \hat{F}_t^i &= \sum_{j=1}^n \frac{\vec{e}_{jt}}{\hat{R}_{jt}} = \sum_{j=1}^n \frac{\vec{r}_{jt}}{\|\vec{r}_{jt}\|^2}, & \hat{r}_{it} &= \begin{bmatrix} \hat{x}_{it} \\ \hat{y}_{it} \end{bmatrix}, \\ F_t &= \hat{F}_t^i + F_{\text{error}}^i. \end{aligned} \quad (90)$$

The difference between F_t and \hat{F}_t^i is F_{error}^i . On the other hand, F_{error}^i is completely unknown and there is no specific model applicable to it. According to Eq. (91) a white Gaussian noise can be considered in such a way wherein its effect on state variables is like the F_{error}^i effect. (N means normal.)

$$F_{\text{error}}^i dt \sim N(0, \sigma_t \sigma_t^T dt). \quad (91)$$

If it is possible to replace disturbance with a white Gaussian noise with characteristics given in Relation (91), then it is possible to use Kalman filter equations to estimate position; the simulation results confirm this. After this presumption,

it is important to have a correct estimation of the standard deviation of noise σ_t in an intuitive and empirical method, which is dependent on the amplitude and frequency of the disturbance force (or it can be said that it is dependent on errors of relative position estimation). Eq. (92) explains the dynamic of the moving target from the viewpoint of robot i .

$$m_t [\hat{\ddot{x}}_t \ \hat{\ddot{y}}_t]^T = k_t (\hat{F}_t^i + F_{\text{error}}^i) - C_t [\hat{\dot{x}}_t \ \hat{\dot{y}}_t]^T. \quad (92)$$

In Eq. (93), the process noise covariance matrix is calculated. As a result, the Kalman algorithm operates using only \hat{F}_t^i and R_1^{it} without need of F_t .

$$R_1^{it} = B_t \sigma_t \sigma_t^T B_t^T. \quad (93)$$

Eqs. (94)–(96) are related to the Kalman filter that robot i used to estimate the relative position and velocity of the moving target.

$$k^{it}(t) = \phi_t(t) p^{it}(t) \Theta^{itT}(t) \times [R_2^{it}(t) 7 + \Theta^{it}(t) p^{it}(t) \Theta^{itT}(t)]^{-1}, \quad (94)$$

$$p^{it}(t+1) = [\phi_t(t) - k^{it}(t) \Theta^{it}(t)] p^{it}(t) \phi_t^T(t) + R_1^{it}(t), \quad (95)$$

$$\hat{X}_{it}(t+1) = (A_t (\hat{X}_{it}(t) + X_{Ci}(t)) - \dot{X}_{Ci}(t)) dt + \hat{X}_{it}(t) + B_t \hat{F}_t^i dt + k^{it}(t) (Z^{it}(t) - \hat{Z}^{it}(t)),$$

$$Z^{it} = [z_1^{it} \ z_2^{it}]^T. \quad (96)$$

As mentioned earlier, it is not necessary to enter position terms X_{Ci} into Eq. (96). The outputs of Z^{it} used in Eq. (96) are the real outputs of the sensor (Eqs. (74) and (77)). By having adequate information about the moving target and its escape force, it is possible to calculate its absolute acceleration to use in the next section.

8. Design of the controller for the group of robots

In this section of the article, the desired acceleration, \ddot{x}_d and \ddot{y}_d , for the robot is calculated. The aim of this study is to control a group of robots to follow a moving target and loop around it. Looping can be done both in a manner where the moving target is completely confined by a group of robots, or in a way where robots make coverage around the target while moving. The aim is to follow the moving target by the robots and make them arrange a relative distance to the moving target in such a way that each robot has a certain relative distance to the other robots. By dynamical analysis of the inertial agents, the desired acceleration for each robot is calculated.

The governing dynamics for massless agents are different from inertial agent dynamics. Each component of the inertial agents tends to keep its velocity. The extracted control rules for the massless group of robots cannot be applied to an inertial group of agents; it means they need to be combined with other algorithms to achieve that goal. If the group of inertial agents does not control correctly, it may lead to chaos in the group. As a result, it is important to know the dynamical behavior of the agents. There is a suitable algorithm that is simple and compatible with an intuitive point of view and dynamical analysis. In [32], the desired accelerations are calculated using that method and in this section of the article, a more perfect approach is used to determine the desired acceleration.

Figure 6 shows a group of inertial agents. This group includes a moving target and two tracking robots. Each of the agents has mass and their controller output is force. It is necessary to

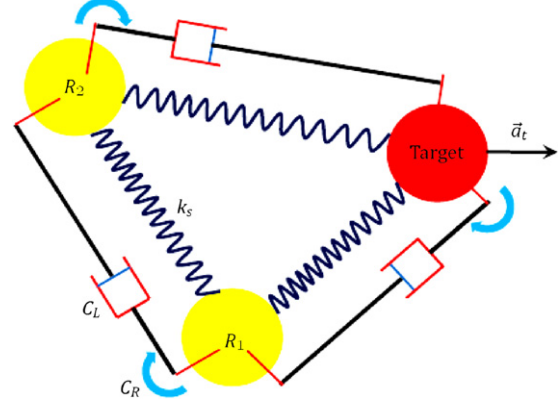


Figure 6: Physical descriptions of the control algorithm stated by mechanical elements.

calculate the desired acceleration for each robot to calculate the exerted forces or torques on each robot. In Section 3 of the article, the relationship between the desired acceleration and torque is defined. In this section, the appropriate desired acceleration would be calculated. In Figure 6, k_s or the elasticity coefficient, is the symbol of proportional control, and C_L and C_R are linear and rotational damping coefficients, respectively, which are symbols of derivative control. This point of view can be useful in order to design a controller and tune the coefficients. The controller for one car-like robot is explained in Section 3. In Figure 6, the goal for the robots is to follow the target using a group maneuver and to be at specified relative distances from the target. The desired acceleration for each robot is the summation of the desired accelerations, which are applied by other robots and the target.

Eq. (97) shows the moving target acceleration and Eq. (98) shows the desired acceleration of robot i .

$$a_t = [\ddot{x}_t \ \ddot{y}_t]^T, \quad (97)$$

$$\ddot{q}_{di} = [\ddot{x}_{di} \ \ddot{y}_{di}]^T. \quad (98)$$

Eq. (99) expressed the relative vector and distance, and the direction between two agents.

$$\vec{r}_{ij} = [x_j - x_i \ y_j - y_i]^T, \quad R_{ij} = \|\vec{r}_{ij}\|, \quad \vec{e}_{ij} = \frac{\vec{r}_{ij}}{R_{ij}}. \quad (99)$$

Eq. (100) expresses the desired acceleration of robot i in which n is the number of robots, R_{ij} is the distance between two agents, R_R is the equilibrium distance between robots, R_t is the equilibrium distance between the robots and the target, k_p is the proportional control coefficient for adjusting distances, k_d is the derivative coefficient to reduce the relative speed between robots, k_p^t is the proportional coefficient, and k_d^t is the derivative coefficient between the robots and the target.

$$\ddot{q}_{di} = \sum_{\substack{j=1 \\ j \neq i}}^n \left(k_p (R_{ij} - R_R) \vec{e}_{ij} + k_d \begin{bmatrix} \dot{x}_j - \dot{x}_i \\ \dot{y}_j - \dot{y}_i \end{bmatrix} \right) + k_p^t (R_{it} - R_t) \vec{e}_{it} + k_d^t \begin{bmatrix} \dot{x}_t - \dot{x}_i \\ \dot{y}_t - \dot{y}_i \end{bmatrix} + \vec{a}_t. \quad (100)$$

Equilibrium distances are adjusted based on the type of maneuver and safety margins to prevent any collision. As an example, in order to hunt a moving target, the equilibrium

distance between the robots and the moving target should be in such a way that in an equilibrium situation, the moving target is placed at the geometrical center of the group of robots. In order to make coverage around the moving target, the equilibrium distance between the robots should be chosen in such a manner that the target should be able to move freely and the robots move around it harmonically.

According to Eq. (100), the moving target acceleration is the only acceleration term that appears in the desired acceleration of each robot. It should be noticed that for a group of agents, it is meaningless to have two different accelerations in their equation of the desired value, and it is dynamically wrong. Here, the moving target provides acceleration and the proportional and derivative control terms of the robots. According to this design, the group would have equilibrium amongst itself, while following the moving target properly.

By installing a coordinate system on the moving target and looking at the other robots (omitting the moving target acceleration term), only proportional and derivative control rules remain, in which the aim would be to reach a static stability with no tracking.

According to the article goal, the equilibrium distance between the robots and the target is important and the proportional control coefficient needs to be large enough. This choice leads to higher acceleration when the relative distance between the target and the robots is far, which causes the robots to have high speeds and makes the system unstable and chaotic. Hence, using constant control coefficients is not suggested. In order to improve response quality, two pairs of different coefficients are used for short and far distances, and the suitable coefficient is chosen when necessary. In order to solve this problem, for far distances, the proportional coefficient should be small enough and the derivative coefficient should be large; for short distances, the opposite approach is needed. This is done by a continuous mathematical equation automatically.

$$D = R_{it} - (R_t + \delta), \quad (101)$$

$$\beta = \frac{(1 + \tanh(D * a^2))}{2}, \quad (102)$$

$$[k_p^t \quad k_d^t]^T = \beta [k_{p1}^t \quad k_{d1}^t] + (1 - \beta) [k_{p2}^t \quad k_{d2}^t]. \quad (103)$$

In Eq. (101), δ decides at which distance from the equilibrium distance the change of the coefficient starts. In Eq. (102), β is a coefficient that varies between zero and one. When the difference between the relative distance and the equilibrium distance is greater than δ , β is approximately 1, and when it is smaller than δ , β tends to zero. It means that this coefficient decides which coefficient should be used. Finally, a shows the β changing rate.

In order to achieve the final goal, the derivative control of the robots should always operate in order to control their relative speed, but it is not necessary for the proportional control to work permanently. Robots move towards the target, and when the relative distance between two robots is less than their equilibrium distance, the proportional control operates and prevents their collision. If the robots try to achieve an equilibrium distance between themselves and the moving target, the final goal, which is to follow the moving target and hunt, will be achieved.

9. Indirect control of robot body angle

Considering the fact that the control equations are just for the center of the robot, the body angles that change during this

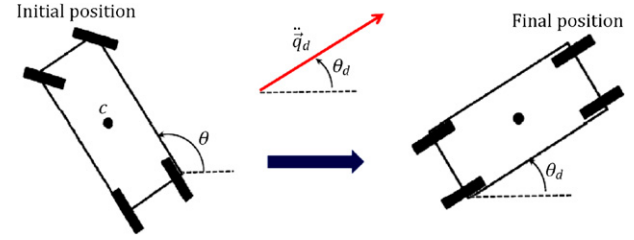


Figure 7: Robot tendency to a specified orientation.

control are needed to be known precisely. Different simulations show an interesting phenomenon, which will be described here. When the control of the robot group is done, each robot angle tends to reach the desired acceleration angle, as shown in Figure 7.

Figure 7 shows the situation in which the robot angle converges to the desired acceleration angle of the robot center. This acceleration is a summation of the different desired accelerations explained earlier. The desired acceleration angle is the direction towards which the agents tend to move. This phenomenon will be explained in the section of the simulation of the coverage around the target. As seen in Figure 7, the steering angle converges to zero too. So, the robots always follow the target where their front sides are towards the tracking direction. In order to get to this goal, a limited time span is consumed.

10. Simulation results

In this section, a group of four robots are controlled to hunt or follow a target by using control algorithms and estimations. Simulation results include: following the target by a robot and the target reaction, hunting the moving target by four robots, following and making coverage around the target, the distance between the robots and the target during hunting, real escape force and its estimation error, relative distance between two robots and its estimation error, relative distance of robots compared to the target and their estimation error, velocity of the center of the robots and moving target, driving and steering torque for a robot, and the simulations results for indirect control of the robot body angle throughout.

The parameters used in this section are tabulated in Table 1. The equilibrium distance between robots and the target is $R_t = 5$ (m). Therefore, $R_R = 5\sqrt{2}$ (m). Since there are four robots and during hunting the target should be placed in the geometrical center of these four robots, in order to follow and make coverage around the target, the distance between the robots is achieved shorter or equal to $R_R = 0.7 \times 5\sqrt{2}$ (m).

In the simulation figures, the triangle is the symbol of the car-like robot and the star is the symbol of the moving target.

Figure 8 shows following a moving target by a robot and the way that it escapes. The diagrams related to hunting a moving target and its simulation results are shown in Figures 9–16.

In Figure 9, four car-like robots are used to follow and hunt a moving target. Although the moving target escapes the group, the robots reach and circle around it. The simulation is done for 120 s and the group is treading steadily from 60 to 110 s. But after 110 s, due to a small asymmetry in group formation and the existence of nonholonomic constraints in robot dynamics, the moving target decides to escape from the enclosed region. However, the group configuration changes slightly to react to the targets new behavior (Figure 10).

Table 1: Geometric, physical and control parameters and properties used in simulations.

$dt = 0.05$ (s)	Discretization step time
$d = 1$ (m), $L = 2d$, $m = 5$ (kg)	Robot mass and geometries (length and mid-length)
$\rho = 0.2$, $\tau_{1\max} = 1$ (N m), $\tau_{2\max} = 5$ (N m)	Actuator parameters
$k_p = 6$ (N/m), $k_d = 1$ (Ns/m)	Damping and spring(repulsive) coefficient between robots
$k_p^t = 0.5$ (N/m), $k_d^t = 4$ (Ns/m)	Damping and spring coefficient between robots and target at long distances
$k_p^s = 3$ (N/m), $k_d^s = 3$ (Ns/m)	Damping and spring coefficient between robots and target at small distances
$\delta = 2$ m, $a = 2$	Parameters of controller coefficient variations
$k_\phi = 0.5$ (1/s)	Coefficient of steering controller
$m_t = 1$ (kg), $c_t = 1$ (Ns/m), $k_t = 1$	Mass, damping and coefficient of target escape force
$\sigma_t = 0.05$ (N \sqrt{s})	Applied noise to target dynamics
$R_t = 5$ (m)	Equilibrium distance between robot and target
$R_R = 5\sqrt{2}$ (m)	Equilibrium distance between two robots (minimum allowable distance) during hunting the target
$R_R = 0.7 \times 5\sqrt{2}$ (m)	Equilibrium distance between two robots during covering around the target
$p_0 = 1$ (dB), $\alpha = 5/\ln(10)$ (dB)	Signal strength attenuation parameters
$\sigma_p = 0.2$ (dB), $\sigma_\psi = 0.05$ (rad)	Added noise to signal strength and direction
$\sigma_{\psi\theta\phi} = 0.01I_{3 \times 3}$	Added noise to received data sent by robot
$\sigma_R = 0.1$ (m), $\sigma_\psi = 0.05$ (rad)	Added noise to target distance and view angle sensor
$\sigma_x = 1$ (m), $\sigma_y = 1$ (m)	Added noise to tracer (flying equipment) data

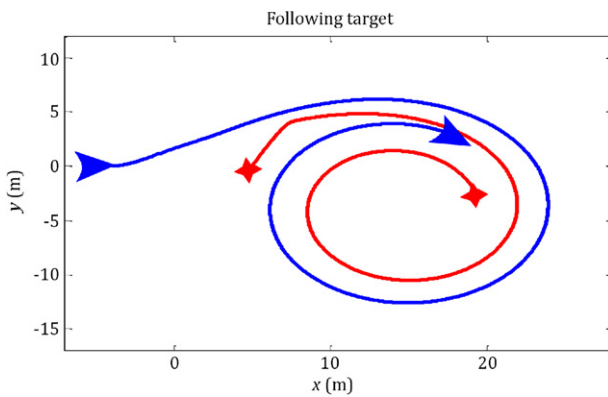


Figure 8: Following target with a robot.

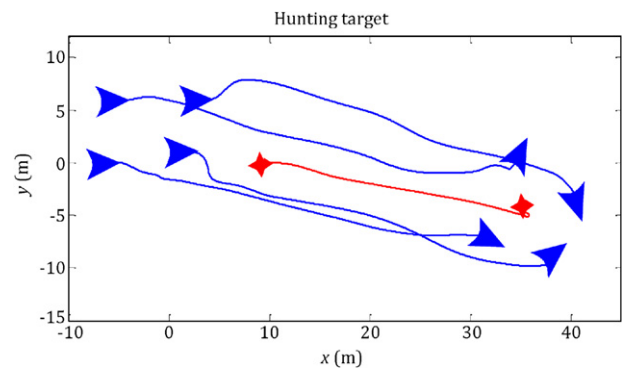


Figure 10: Robots try to stop target from escaping after it is hunted.

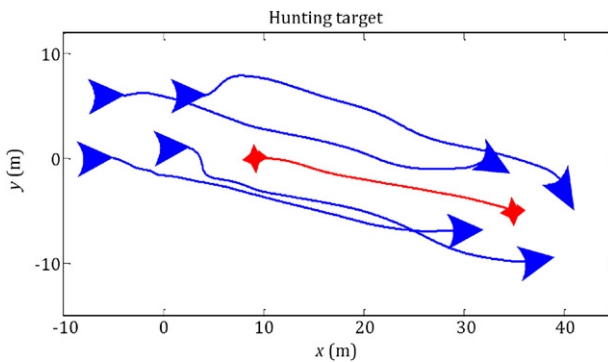


Figure 9: Following and hunting the target with four robots.

The target dynamics are holonomic and move in any arbitrary direction: meanwhile, car-like robots have movement limitations and nonholonomic constraints. The phenomenon shown in Figure 10 happens only when the group reaches a steady state behavior. Whenever the group moves in a determined direction, the robots hunt the target and the robots velocity converges to zero. However, it is practically impossible for a group of car-like robots to hunt a target in a steady situation. Because, whenever the target decides to change its direction after being steady for a while, robots follow it with a delay, which is ordinary (because of asymmetry and nonholonomic constraints). The problem can be assumed in another way. In other words, it can be presumed that whenever

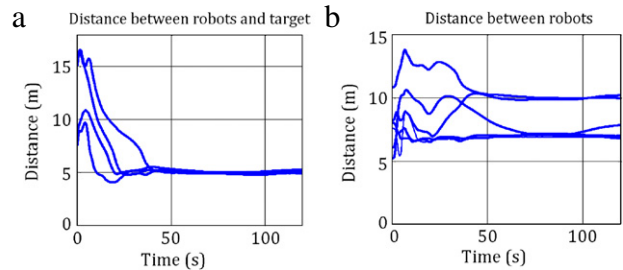
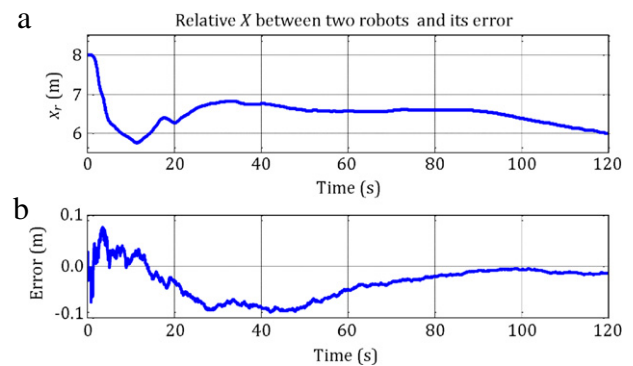


Figure 11: Distances between the target and robots (a), and distances between robots (b).

Figure 12: Relative position (x) between two robots and its estimation error.

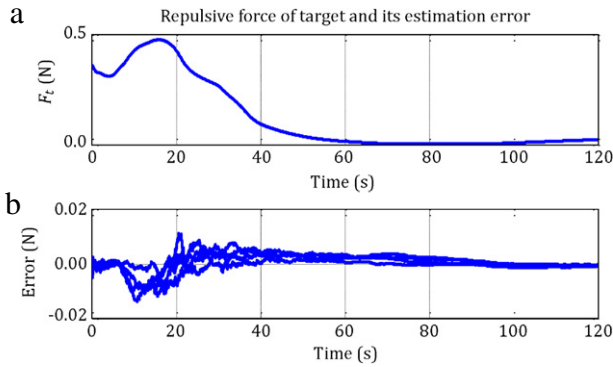


Figure 13: Real escape force of target and its estimation error.

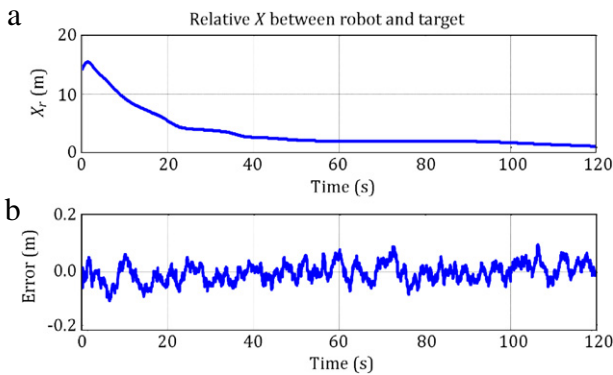
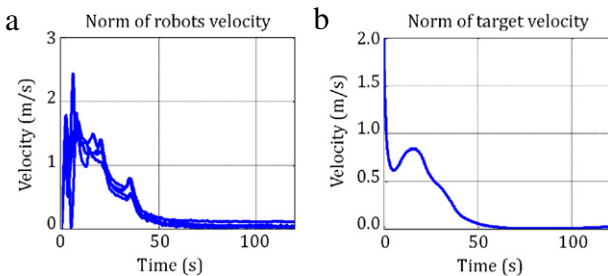
Figure 14: Relative position (x) between robot and target with its estimation error.

Figure 15: Velocity of robots center (a), and velocity of target (b).

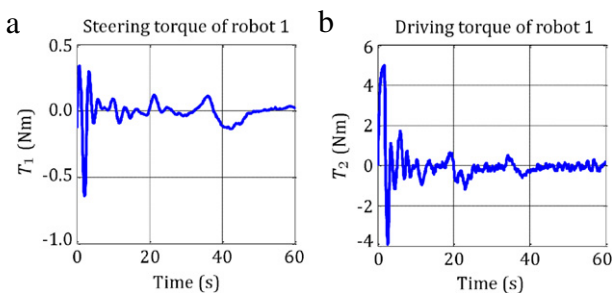


Figure 16: Driving torque (a), and steering torque (b).

the target is hunted, the cars velocities become zero and the target no longer tends to escape. The system behavior under these conditions is observed in Figure 9.

Figure 11(a) shows that the distance between the robots and the moving target converges to the equilibrium distance, $R_t =$

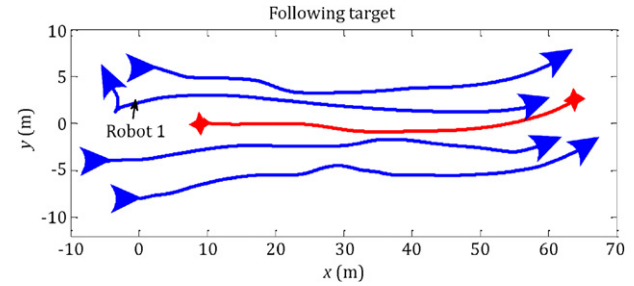


Figure 17: How the target is followed and how coverage is made around it.

5 (m), and Figure 11(b) shows the distances between robots greater than the minimum allowable distance, $L = 2d = 2$. Figure 11 guarantees that no collision will happen. This distance is two times greater than the minimum distance between the centers of two robots. The minimum equilibrium distance between robots in the simulation is $R_R = 5\sqrt{2}$ (m).

Figure 12 shows the relative distance between two robots and its estimation error. The results show that the estimator works properly. As shown, the estimation error is quite small.

Figure 13 shows the real target escape force and its estimation error (done by four robots). The escaping force estimation is done by all of the robots. At the beginning, the increase in the escape force means that the robots are approaching the target, and correspondingly a decrease in the escape force means that they are circling around the target. It is obvious that when the four robots loop around the target symmetrically, the resultant force converges to zero. But after hunting the target, the velocity converges to zero; the target tends to change its direction, which leads the group to become more asymmetrical. This phenomenon was illustrated in Figure 10. As seen from the estimation error diagram of Figure 13, the assumption that disturbance force F_{error} is a white noise is acceptable.

Figure 14 shows the relative distance between a robot and the target and its estimation error. The above estimation is done by a robot.

Figure 15 shows velocity of the robots and moving target. The decrease in speed by passing time shows that the target is hunted and the robots are circling around it.

Figure 16 shows exerted torques on the robots by considering the effect of the filtering and saturation of the actuators. The torque diagram expresses suitable and reasonable results.

Figure 17 shows the following and surrounding of the target. This maneuver is done whenever the goal is not hunting the target, but the goal is to make an adequate coverage around it. This coverage can be used to protect a leader. Under the initial condition, one of the robots (robot 1) does not have a suitable orientation, but it is seen that under a specific maneuver, it returns and follows the target with its front-side. This phenomenon was explained more in Section 9. The next diagram shows the body angle of the robot.

Figure 18 shows the steering angle in the maneuver (Figure 18(a)) and the body angle of the robot in the maneuver (Figure 18(b)). As seen, the body angle of robot 1 changes from 120° to the desired acceleration angle (the group direction). Meanwhile, it follows the group with its front-side.

11. Conclusions

In this article, the following and hunting of a moving target by a group of car-like robots were investigated. The tracking

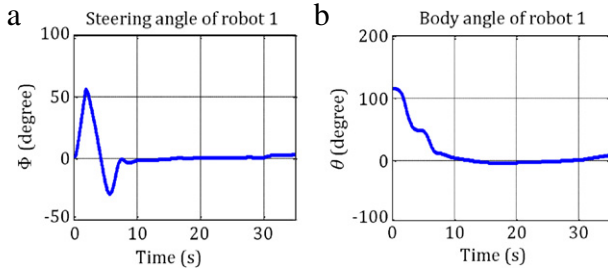


Figure 18: Steering angle (a) and body angle of robot 1 (b).

robots have a nonholonomic constraint and the target did not have any constraint. It is assumed that the measurements of state variables and control parameters by the robots are not complete and also the target moves away from the robots by a repulsion force. In spite of these presumptions, the results of the simulations prove that the represented control algorithm is properly applicable to nonholonomic robots. The results show that in order to do team work, complicated algorithms are not needed. On the other hand by knowing the dynamics behavior of the robots, simpler and stronger algorithms can be represented.

Two main reasons behind undertaking a correct maneuver are the dynamics analysis of the inertial agents and the varying controller coefficients. The results also show that the estimation algorithms were successful and combining the estimation algorithms with car-like robot group maneuvers achieved the results of this article. By using the estimation algorithm and despite not knowing the value of the target escape force, proper estimation of the target movement is achieved. The output of the controllers is torque and it can be used for checking the possibility of the maneuver. The control algorithm characteristics behave in such a way that the robots always move with their front-side; this phenomenon is physically important and reasonable.

By varying the controller coefficients and equilibrium distances, both the following and covering operations were done successfully. As seen from the results of the covering operation, this maneuver can be used in order to make security coverage around a leader. By slight changes in the proposed algorithm, other maneuvers can be done too. This is a step towards finding a solution for real problems and real applications.

Future work in this field is mostly about improving indirect control of body angles, estimating environmental disturbances and using them in the control procedure, controlling car-like robots equipped with special accessories, preventing robots from rolling while performing group maneuvers, estimating surrounding obstacles, performing different kinds of maneuver while obstacles are present and performing practical operations; for example, in traffic engineering.

References

- [1] Jun, O. "Multi-agent robot systems as distributed autonomous systems", *Advanced Engineering Informatics*, 20(1), pp. 59–70 (2006).
- [2] Sayyaadi, H. and Moarref, M. "A distributed algorithm for proportional task allocation in networks of mobile agents", *IEEE Journal of Automatic Control*, 56, pp. 405–410 (2011).
- [3] Sayyaadi, H. and Dostmohammadian, M.R. "Finite time consensus in directed switching network topologies and time-delayed communications", *Scientia Iranica: Transaction B, Mechanical Engineering*, 18(1), pp. 77–85 (2011).
- [4] Yamaguchi, H. "A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations", *Robotics and Autonomous Systems*, 43(4), pp. 257–282 (2003).
- [5] Lihua, D., Hao, F. and Haiqiang, Z. "Task allocation for multi-robot cooperative hunting behavior based on improved auction algorithm", *IEEE Conference, CCC 2008, 27th Chinese*, pp. 435–440 (2008).
- [6] Matveev, A.S., Teimoori, H. and Savkin, A.V. "Range-only measurements based target following for wheeled mobile robots", *Automatica*, 47(1), pp. 177–184 (2011).
- [7] Lan, Y., Yan, G. and Lin, Z. "Distributed control of cooperative target enclosing based on reachability and invariance analysis", *Systems & Control Letters*, 59(7), pp. 381–389 (2010).
- [8] Wu, M., Huang, F. and Wang, L. "A distributed multi-robot cooperative hunting algorithm based on limit-cycle", *IEEE International Conferences on Automation and Robotics*, Kobe, Japan (May 12–17 2009).
- [9] Franchi, A., Stegagno, P., Rocco, M.D. and Oriolo, G. "Distributed target localization and encirclement with a multi-robot system", *Conference Paper of 7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy (2010).
- [10] Jun, L., Qi-shu, P., Bing-Rong, H. and Mao-Hai, L. "Multi-robot cooperative pursuit based on association rule data mining", *IEEE Conferences on Fuzzy Systems and Knowledge Discovery*, pp. 303–308 (2009).
- [11] Zhang, J. "Practical issues in formation control of multi-robot system", A Dissertation Submitted to the Office of Graduate Studies of Texas A&M University in partial fulfillment of the requirements for the degree of Doctor of Philosophy, May 2010. Available at: <http://repository.tamu.edu/bitstream/handle/1969.1/ETD-TAMU-2010-05-7886/ZHANG-DISSERTATION.pdf?sequence=2>.
- [12] Pinto, G. "Towards a control architecture for cooperative nonholonomic mobile robots", Doctoral Thesis, 26 November 2007. Available at: <http://tel.archivesouvertes.fr/docs/00/19/38/35/PDF/thesis.pdf>.
- [13] Pinto, A.G., Fraisse, P. and Zapata, R. "A decentralized algorithm to adaptive trajectory planning for a group of nonholonomic mobile robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 404–417 (2006).
- [14] Pinto, G., Fraisse, P. and Zapata, R. "Decentralized strategy for car-like robot formations", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4176–4181 (2007).
- [15] Pinto, A.G., Fraisse, P. and Zapata, R. "Wireless communication for secure positioning in multi robot formations of nonholonomic ground vehicles", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 4198 (2008).
- [16] Pinto, A.G., Fraisse, P. and Zapata, R. "Wireless reception signal strength for relative positioning in a vehicle robot formation", *IEEE Conferences on Robotics Symposium, LARS '06. IEEE 3rd Latin American*, pp. 100–105 (2006).
- [17] Laumond, J.P. and Jacobs, P.E. "A motion planner for nonholonomic mobile robots", *IEEE Transaction on Robotics and Automation*, 10(5), pp. 577–593 (1994).
- [18] Xi, W. and Baras, J.S. "MPC based motion control of car-like vehicle swarms", *Proceeding of the 15th Mediterranean Conference on Control & Automation*, Athens-Greece, pp. 1–6, 27–29 (July 2007).
- [19] Bemporad, A., Luca, A.D. and Oriolo, G. "Local incremental planning for a car-like robot navigating among obstacles", *Proceeding of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, vol. 2, pp. 1205–1211 (April 1996).
- [20] Makloul, O.M., Halwagy, Y. and Beumi, M. "Cascade Kalman filter application in GPSINS integrated navigation for car like robot", *IEEE Conference on Radio Science*, pp. 1–15 (2009).
- [21] Feng, D. and Krogh, B.H. "Satisfying feedback strategies for local navigation of autonomous mobile robots", *IEEE Transactions on Systems, Man and Cybernetics*, 20(6), pp. 1383–1395 (1990).
- [22] Vanualailai, J., Sharma, B. and Nakagiri, S. "An asymptotically stable collision-avoidance system", *Non-Linear Mechanics*, 43(9), pp. 925–932 (2008).
- [23] Moret, E.N. "Dynamic modeling and control of a car-like robot", MS Thesis, Thesis Submitted to the Faculty of the Virginia Polytechnic Institute, 2003. Available at: http://scholar.lib.vt.edu/theses/available/etd-03242003.../thesis_ENMoret.pdf.
- [24] Luca, A.D. and Oriolo, G., *Modeling and Control of Nonholonomic Mechanical Systems*, In *Kinematics and Dynamics of Multi-Body Systems*, 360, pp. 277–342, Springer Verlag, Wien (1995).
- [25] Bowska, E.J. "Stabilizability and motion tracking conditions for mechanical nonholonomic control systems", *Mathematical Problems in Engineering*, doi:10.1155/2007/31267. Hindawi Publishing Corporation, Article ID 31267, 20 pages (2007).
- [26] Fraisse, P. and Zapata, R. "Remote secure decentralized control strategy for mobile robots", *Advanced Robotics*, 19(9), pp. 1027–1040 (2005).
- [27] Roberts, R. "IEEE P802.15 ranging subcommittee final report 2004", 2004. Available at: <http://grouper.ieee.org/groups/802/15/pub/04/15-04-0581-07-004a-ranging-subcommittee-final-report.doc>.
- [28] Li, X. "Performance study of RSS-based location estimation techniques for wireless sensor networks", *Preceding IEEE Military Communication Conference*, 1, pp. 1–5 (2005).
- [29] Frew, E.W. "Cooperative standoff tracking of uncertain moving targets using active robot networks", *IEEE International Conference on Robotics and Automation*, Roma, pp. 3277–3285 (April 10–14, 2007).

- [30] Das, A.K. and Fierro, R. "A vision-based formation control framework", *IEEE Transactions on Robotics and Automation*, 18(5), pp. 813–825 (2002).
- [31] Sarkar, T.K. and Zhong, J. "A survey of various propagation models for mobile communication", *IEEE Antennas and Propagation Magazine*, 45(3), pp. 51–82 (2003).
- [32] Urcola, P. and Riazuelo, L. Cooperative navigation using environment compliant robot formations, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems* Acropolis Convention Center Nice, France, pp. 2789–2794 (September 22–26, 2008).

Hassan Sayyaadi received a B.S. degree from Amirkabir University of Technology, Tehran, Iran, in 1987, and an M.S. degree from Sharif University of Technology, Tehran, Iran, in 1990. He received his Ph.D. degree from the University of Tokyo, Japan, in 2001, all in Mechanical Engineering. From 1990 to 1996, and from 2001 until now, he has been with the Mechanical Engineering Department of Sharif University of Technology. He is now Associate

Professor and his research interests include; Dynamics and Control, Robotics and Mechanisms, Artificial Intelligence and Neural Networks.

Hamed Kouhi Gilvan was born in Masal, Guilan, in 1986. He received a B.S. degree in Mechanical Engineering from the University of Guilan in 2008 and, in 2010, he received his M.S. degree in Mechanical Engineering from Sharif University of Technology, Tehran, Iran, on the topic of the "Control of Car-Like Multi Robot for Following and Hunting a Moving Target". His current research interests include: Control, Sensor Fusion, Multi Agent System, Dynamics, Robotics, Optimization and Neural Networks.

Hassan Salarieh was born in Tehran, in 1980. He received his B.S., M.S. and Ph.D. degrees, all in Mechanical Engineering, from Sharif University of Technology, Tehran, Iran, in 2002, 2004 and 2007, respectively, and is now Assistant Professor at the same university. He is working in research fields such as Dynamics and Control.