Multiple Mobile Robotic Formation Control Based on Differential Flatness

Lintle Tsiu(✉) and Elisha Didam Markus

Central University of Technology, Freestate, South Africa
lintletsiu@gmail.com, emarkus@cut.ac.za

Abstract. There are numerous applications whereby multi-robot cooperative systems are more useful than using a single robot. However, for a cooperative system to implement tasks accurately, an effective formation controller is essential. This paper presents a formation control method that is based on Differential Flatness theory to improve coordination control of a model-based cooperative multiple mobile robotic system. The Differential Flatness characterisation of the team robots allows for the linearization of the system to a stable linear equivalent. Also, the Flatness theory has the advantage of simplifying the trajectory planning task because nonlinear differential equations can be converted to algebraic equation, hence there is no need to integrate robot model differential equations. Each robot is represented by a reduced number of variables which greatly reduces the computational cost especially when dealing with multiple robots that can otherwise entail solving large robotic model differential equations. Simulations using a formation of three differentially driven mobile robots in a leader-follower formation, is used to validate the cooperative formation controller proposed in this paper.

Keywords: Formation control · cooperative control · Differential flatness · consensus control · synchronization control · differentially driven mobile robot

1 Introduction

Research on multiple robot formations has recently garnered a great deal of attention among robotics researchers [1–17]. This is because this type of system is more flexible, adaptable to unfamiliar environments, and robust than a single robot. Formation control applications include various cooperative tasks such as exploration [3], mining support [4], agriculture [5], to name a few. Several approaches to mobile robot formation control have been proposed in literature. These control approaches can be conceptually categorized into four categories, namely: cooperation, consensus synchronization, and coordination control.

In cooperative control, robots share information so that a common objective can be achieved. Potential application areas of this formation control include search and rescue, surveillance, and explorations. In a cooperative group, individual robots might have limited sensing, processing, and communication capabilities. It is for this reason that the study in [6] used a probabilistic localization and control method that considers the

M. Masinde and A. Bagula (Eds.): AFRICATEK 2022, LNICST 503, pp. 113–128, 2023.
https://doi.org/10.1007/978-3-031-35883-8_8

motion and sensing capabilities of individual robots. The method allowed for robots to alter their sensing topologies when needed. Additionally, to avoid communication breakdown between robots, the author in [7, 8] minimizes the frequency of the cooperative robots' communication and use minimal vision-based data. This enables the multi-robot formations to navigate safely in cluttered environments.

Consensus control, on the other hand, involves robots constantly altering their data so that they all come to the same decision. The possibility of consensus never being reached is high, however, when there are time delays, miscommunications, and noise. The study by [9] applied frequency domain analysis to convert the characteristic system equations into quadratic polynomials with pure imaginary eigenvalues. This helped the system to gain the maximum stability state during time delays and noise disturbances. Similarly, [10] Converged the individual information of a robot with the information of its neighbours despite communication delays. Also, the authors in [11, 12] used Lyapunov-based controller while [13] used a knowledge-oriented task and motion planning method. These enabled the robotic system to create a feasible obstacle free path.

Alternatively, synchronization control is used in several studies. In this approach each robot follows a desired trajectory while keeping a synchronized formation two of its neighbouring. Synchronous formation control is generally decentralized. The author in [14] developed a nonlinear synchronization controller for nonholonomic unicycle robots that allowed for directed and undirected information flow between the robots. Thus, robustness is achieved despite disturbances. Alternatively, in [15], geometric path following was developed with Lyapunov theory, backstepping, and helmsman behaviours applied to each controlled path. Unlike other approaches synchronous formation control has a simpler control structure with elevated motion coordination and robustness.

Finally, coordination control entails maintenance of certain kinematic relationship between team robots. It is usually centralized. The authors in [16, 17] increased the coordination of the multi-AUV. This was achieved by designed an integrated algorithm which merges the homodromous degree, district-difference degree, and the dispersion degree into the potential field function of the surface-water environment. In this approach, no pre-learning procedure was required and the multi-AUV coordination was increased, thereby decreasing the computational cost.

This paper aims to improve coordination control of cooperative multiple mobile robotic system using differential flatness theory. The robotic formation is made up of three differentially driven mobile robots with similar dynamics. One robot is designated as a leader and the other two are followers. Using the kinematic models of the robots, the robot differential flatness characterization is executed to exploit flatness properties in order to design a flatness-based control algorithm for trajectories generation and tracking.

Moreover, literature indicates that there are several problems associated with controlling multi-robotic systems, including high energy consumption [16, 17], uncertainty in observing robot workspaces [9], and resolution of communication protocols [6–8]. Also, the larger the system the more the dynamics of the coupled robotic network, thus its control entail solving large robotic model differential equations which are almost intractable to solve. Therefore, the main contribution of this paper is to significantly

reduce the computational cost involved in dealing with multiple robotic systems by representing the cooperative system with flat outputs and their derivatives up to a certain order instead of its system state variables. Also, to simplify the trajectory generation problem by using simple polynomials instead of integrating robot model differential equations. As a result, the proposed method therefore greatly simplifies the cooperative formation control.

This paper is sectioned as follows: Sect. 2 represent the mathematical of the cooperative multi-robotic system and the robots that make up the cooperative system. The concept of differential flatness is introduced, and a flatness analysis of a differential drive robot is presented in Sect. 3. Section 4 presents differential flatness-based trajectory generation and design of a flatness controller for trajectory tracking and for formation control. Also, the simulation tests results on effectiveness of the formation controllers designed in the previous section is presented in Sect. 5. Finally, Sect. 6 give some concluding remarks.

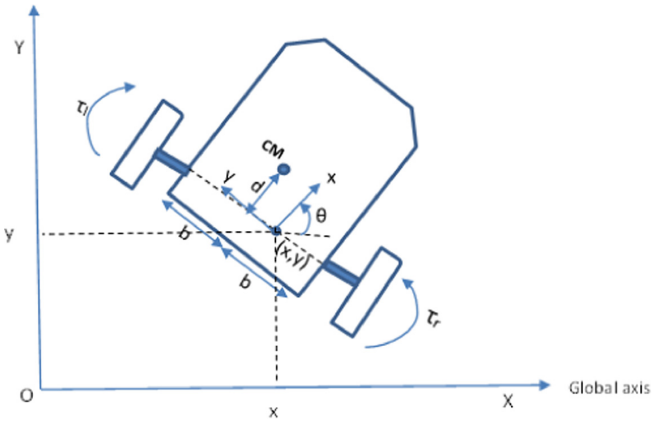

Fig. 1. Differentially driven wheeled mobile robot (DDWMR).

2 System Description

The cooperative multi-robotic system is homogeneous. That is, it is made up of a group of differentially driven wheeled mobile robots of similar physical characteristics. The differentially driven wheeled mobile robot shown in Fig. 1 is made up of two drive wheels attached on one axis, and each wheel is able to be independently driven forward or backward. Also, the robot is non-holonomic, which basically means it has constraints that cause a reduction in the local mobility of the robot.

The robot configuration is represented by the generalised coordinates $q\ [x\ y\ \theta]^T$ where (x, y) is the position and θ is the heading orientation expressed in cartesian coordinate system of inertial frame. The wheels with radii r are each positioned at a distance b from the centre of the robot chassis. The Cartesian coordinates $(x\ y)$ are located at a distance d from and the centre of mass (CM). In the kinematic model the system inputs

are the wheel velocities $\dot{\theta}_r$ and $\dot{\theta}_l$, while in the dynamic model they are the driving motor torques τ_r and τ_l. The subscript r and l represent quantities of right and left wheels respectively.

A. Kinematic Model

The kinematics of the WMR including the nonholonomic constraints is described in [18] as:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S(q)\bar{v}(t) \tag{1}$$

Where v is the forward (linear) velocity and ω is the angular velocity. Furthermore, the equivalent wheel inputs are then given by:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

Because robot is nonholonomic, its wheels do not slide to the sides, that is velocity component for the contact point perpendicular to the plane is zero. This is expressed as:

$$\dot{x}\sin\theta - \dot{y}cos\theta = 0$$

Or in matrix form:

$$A(q) = [sin\theta, -\cos\theta, 0] \tag{3}$$

Note that *S(q)* is the null space of the non-holonomic constraint A(q).

B. Dynamic Model

Euler-Lagrange dynamic equations of motion are used to obtain the dynamic equations of the mobile robot However, friction and viscous forces will be ignored. The dynamic model is therefore given as:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} = E(q)\tau - A^T(q)\lambda \tag{4}$$

Or can also be written as:

$$\begin{bmatrix} m_0 & 0 & -dm_0\sin\theta \\ 0 & m_0 & dm_0\cos\theta \\ -dm_0\sin\theta & dm_0\cos\theta & d^2m_0 + I_0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix}$$
$$+ \begin{bmatrix} 0\,0 & -dm_0\dot{\theta}\cos\theta \\ 0\,0 & -dm_0\dot{\theta}\sin\theta \\ 0\,0 & 0 \end{bmatrix} + \begin{bmatrix} 0\,0 & -dm_0\dot{\theta}\cos\theta \\ 0\,0 & -dm_0\dot{\theta}\sin\theta \\ 0\,0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$
$$= \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} - \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$

For a mobile robot with n degree-of-freedom (DOF) that is subjected to m constraints and has p inputs the generalized coordinates are $q(t) \in \mathbb{R}^{n\times 1}$ and $M(q) \in \mathbb{R}^{n\times n}$ is the symmetric positive-definite inertia matrix. $V(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the centripetal and Coriolis force matrix, $E(q) \in \mathbb{R}^{n\times p}$ is the actuation matrix, $\tau \in \mathbb{R}^{p\times 1}$ is the input vector, $A^T(q) \in \mathbb{R}^{m\times n}$ is the kinematic constraint matrix associated with the nonholonomic constraint equation, and $\lambda \in \mathbb{R}^{m\times 1}$ is the vector of constraint forces. m_o is the robot mass while I_o is the total equivalent moment of inertia. The input driving motor torques τ_r and τ_l provide the robot motion hence the position of the robot (output) changes.

The unconstrained form Eq. (4) is attained by eliminating λ by multiplying $S(q)^T$ on both sides of the equation and taking $\overline{v}(t) = [v, \omega]^T$ as the minimal projected coordinate. Recall from the kinematic model, S(q) is the null space of the nonholonomic constraint $A(q)$. That is, $A(q).S(q) = 0$. Thus, the dynamic model is then written in the state space form as:

$$\dot{x} = f(x) + g(x)u \tag{5}$$

By choosing the state variables as: $x_2 = y$, $x_3 = \theta$,
$x_4 = v$ and $x_5 = \omega$, the first order differentiation of equations is given as:

$$\begin{aligned}
\dot{x}_1 &= \dot{x} = v\cos\theta = x_4\, cos\, x_3 \\
\dot{x}_2 &= \dot{y} = v\, sin\, \theta = x_4\, sin\, x_3 \\
\dot{x}_3 &= \dot{\theta} = \omega = x_5 \\
\dot{x}_4 &= \dot{v} = d\dot{\theta}^2 - \frac{1}{Rm_0}(\tau_{r+}\tau_l) = d{x_5}^2 - \frac{1}{Rm_0}(u_{1_+}u_2) \\
\dot{x}_5 &= \dot{\omega} = \frac{1}{d^2m_o + I_o}\left[-dm_o v\dot{\theta} - \frac{b}{R}(\tau_r - \tau_l)\right] \\
&= \frac{1}{d^2m_o + I_o}\left[-dm_o x_4 x_5 - \frac{b}{R}(u_1 - u_2)\right]
\end{aligned} \tag{6}$$

Finally, the dynamics of the differential drive robot in State space is given as:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0\,0\,0 & \cos\, x_3 & 0 \\ 0\,0\,0 & \sin\, x_3 & 0 \\ 0\,0\,0 & 0 & 1 \\ 0\,0\,0 & 0 & dx_5 \\ 0\,0\,0 & \frac{-dm_o x_5}{d^2m_o+I_o} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \frac{1}{R}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m_0} & -\frac{1}{m_0} \\ \frac{-b}{d^2m_o+I_o} & \frac{b}{d^2m_o+I_o} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{7}$$

C. Modelling the Cooperative System

The cooperative robotic system is homogeneous, that is, all the robots have similar physical parameters and dynamics. In this study, leader-follower formation approach is applied. This means, the system is designed such that a leader robot moves along a desired trajectory and the two followers maintain a desired distance and orientation in relation to the leader robot. Note that leader robot is the only one with information about the desired trajectories to be tracked and the followers use information from the leader to coordinate their motion.

Let the leader and a follower robot be denoted as R_j and R_i, respectively. In this study, the subscript "*j*" and "*i*" denotes leader and follower, respectively. Also, q_j and

q_i are the generalized coordinates of the leader robot and follower robot respectively. Thus, the states and the inputs of the leader and a follower can now be denoted as:

$$\dot{q}_j = \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix} = \begin{bmatrix} \cos\theta_j & 0 \\ \sin\theta_j & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_j \\ \omega_j \end{bmatrix}$$

$$\dot{q}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \tag{8}$$

Moreover, the constant relative distance between the leader and the follower robot is denoted as l^{ref} and the separation bearing angle is ϕ^{ref}. These are defined as:

$$\begin{aligned} d^{ref} &= \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} \\ \phi^{ref} &= \pi - arctan2\left(y_i - y_j,\, x_i - x_j\right) - \theta_j \end{aligned} \tag{9}$$

In order to maintain the desired l^{ref} and ϕ^{ref} a reference robot for the follower should exist and it is given by:

$$q_{ref} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ \theta_{ref} \end{bmatrix} = \begin{bmatrix} x_j - l^{ref} cos(\phi^{ref} + \theta_i) \\ y_j - l^{ref} sin(\phi^{ref} + \theta_i) \\ \theta_i \end{bmatrix} \tag{10}$$

The velocity equation of the reference robot is given by:

$$\dot{q}_{ref} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} \dot{x}_j + l^{ref} sin(\phi^{ref} + \theta_i)\dot{\theta}_i \\ \dot{y}_j - l^{ref} cos(\phi^{ref} + \theta_i)\dot{\theta}_i \\ \dot{\theta}_i \end{bmatrix} \tag{11}$$

The acceleration equation of the reference robot is given by:

$$\ddot{q}_{ref} = \begin{bmatrix} \ddot{x}_{ref} \\ \ddot{y}_{ref} \\ \ddot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} \ddot{x}_j + l^{ref}[\ddot{\theta}_i sin(\phi^{ref} + \theta_i) + \dot{\theta}_i^2 cos(\phi^{ref} + \theta_i)] \\ \ddot{y}_j - l^{ref}[\ddot{\theta}_i cos(\phi^{ref} + \theta_i) - \dot{\theta}_i^2 sin(\phi^{ref} + \theta_i)] \\ \ddot{\theta}_i \end{bmatrix} \tag{12}$$

For a non-linear system

$$\dot{x} = f(x, u) \tag{13}$$

Where x is the state vector, ($x \epsilon R^n$), and u is the control input vector ($u \epsilon R^m$) where $m \leq n$.

The system is differently flat if there exists an output y of m dimensions, such that y is locally a function of x, u and a finite number of successive derivatives of the component of u. That is:

$$(y_1 \ldots\ldots\ldots y_m) = h(x, u, \dot{u}, \ddot{u} \ldots\ldots u^{(l)}) \tag{14}$$

Conversely, x and u should be able to be expressed as functions of y up to a finite number of its successive derivative, that is:

$$x = \alpha(y, \dot{y}, \ddot{y} \ldots\ldots y^{(q)}), \;\; u = \beta(y, \dot{y}, \ddot{y} \ldots\ldots y^{(q+1)}) \tag{15}$$

If x and u are then substituted in the non-linear system equation, it gets identically satisfied:

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{f}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{16}$$

Thus, y is a complete parametrization of the trajectories of the system, and hence y is called a *flat output.*

A. Robot's Differential Flatness Characterisation

The robots forming the cooperative system are differentially driven mobile robots and their flatness analysis is similar. The kinematic model has two control inputs. Therefore, the two flat outputs that have been selected are x and y. This is because the number of flat outputs is always equal to the number of control inputs [19]. Thus:

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{17}$$

When (17) is differentiated with respect to time the following equation is obtained:

$$\dot{F} = \begin{bmatrix} \dot{F}_1 \\ \dot{F}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{18}$$

$$\begin{aligned} \dot{x} &= v \cos\ \theta \\ \dot{y} &= v \sin\ \theta \\ \dot{v} &= \bar{u} \\ \dot{\theta} &= \omega \end{aligned} \tag{19}$$

Where the new input for the new extended system is $\bar{u}$. To linearize the system a second derivative of the system is given as:

$$\ddot{F} = \begin{bmatrix} \ddot{F}_1 \\ \ddot{F}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -v\,sin\,\theta \\ \sin\theta & v\,cos\,\theta \end{bmatrix} \begin{bmatrix} \bar{u} \\ \omega \end{bmatrix} \tag{20}$$

To create diffeomorphism between the original states and flat outputs and their derivatives, input prolongation is used.

Therefore, $\ddot{F}$ is linear with respect to the new input $\bar{u}$ and ω only if v not zero. For a flat system, all the state variables can be expressed in terms of the flat outputs and their

derivatives as follows:

$$\begin{aligned} x &= F_1 \\ y &= F_2 \\ \theta &= \tan^{-1}\frac{\dot{F}_1}{\dot{F}_2} \\ v &= \sqrt{\dot{F}_1{}^2 + \dot{F}_2{}^2} \\ \bar{u} = \dot{v} &= \frac{\dot{F}_1\ddot{F}_1 + \dot{F}_2\ddot{F}_2}{\sqrt{\dot{F}_1{}^2 + \dot{F}_2{}^2}} \\ \omega = \dot{\theta} &= \frac{\dot{F}_1\ddot{F}_2 - \dot{F}_2\ddot{F}_1}{\dot{F}_1{}^2 + \dot{F}_2{}^2} \end{aligned} \tag{21}$$

Furthermore, the flat outputs and their derivatives can completely be expressed in terms of all the state variable and their derivatives:

$$\begin{aligned} F_1 &= x \\ F_2 &= y \\ \dot{F}_1 &= \dot{x} = v\cos\ \theta \\ \dot{F}_2 &= \dot{y} = v\sin\ \theta \\ \ddot{F}_1 &= \bar{u}\cos\ \theta - v\omega\sin\ \theta \\ \ddot{F}_2 &= \bar{u}\sin\ \theta - v\omega\cos\ \theta \end{aligned} \tag{22}$$

Moreover, this diffeomorphic relationship proves that there exists a one-to-one relationship between the state space and the flat output space. This makes it possible to obtain full state.

3 Motion Planning and Control

A. Trajectory Generation

Recall that leader robot is the only one with information about the desired trajectories to be tracked. Therefore, this section presents a polynomial-based trajectory planning that satisfies a specific set of terminal conditions for the leader robot.

Since there are six terminal conditions for the six states x, y, θ, v, $\bar{u}and\omega$, a fifth-order polynomial is used:

$$\begin{aligned} F_1(t) &= a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\ F_2(t) &= b_5t^5 + b_4t^4 + b_3t^3 + b_2t^2 + b_1t + b_0 \end{aligned} \tag{23}$$

With consideration of the limits of the robot, the terminal conditions are set and used to determine the coefficients a_k and b_k where k = [1, …,5].

With the terminal conditions shown in Table 1, a_k and b_k were obtained and the desired trajectories were derived. The destination should be reached in fifteen seconds, thus $t = 15$ s. The robot is driven from rest-to-rest positions.

As a result, the corresponding leader's desired trajectories are:

$$F_1^d(t) = 0.00001185t^5 - 0.0004444t^4 + 0.004444t^3$$
$$F_2^d(t) = 0.000003951t^5 - 0.0001481t^4 + 0.001481t^3 \quad (24)$$

The leader will then track these trajectories and the followers will follow the leader while maintaining a desired distance l^{ref} and orientation ϕ^{ref} in relation to the leader robot.

B. Flatness-Based Controller Design

Following the derivation of the desired trajectory, a tracking controller based on flatness is developed. Given that, $F_1(t)$ and $F_2(t)$ are the actual flat output trajectories, and $F_1{}^d(t)$ and $F_2{}^d(t)$ are the desired flat output trajectories, then error is defined as:

$$e_1 = F_1^d - F_1$$
$$e_2 = F_2^d - F_2 \quad (25)$$

Let

$$\begin{bmatrix} \ddot{F}_1 \\ \ddot{F}_2 \end{bmatrix} = \begin{vmatrix} \delta_1 \\ \delta_2 \end{vmatrix} \quad (26)$$

then the feedback *control laws* to the new inputs can then be defined as:

$$\delta_1 = \ddot{F}_1^d + \rho_1 \dot{e}_1 + \rho_0 e_1$$
$$\delta_2 = \ddot{F}_2^d + \tilde{\rho}_1 \dot{e}_2 + \tilde{\rho}_0 e_2 \quad (27)$$

Where $\overset{\sim}{\rho}_1$, $\overset{\sim}{\rho}_0$, ρ_1, ρ_0 are control the gains. Next the error dynamics of the system in flat output space is determined.

$$\ddot{F}_1^d - \delta_1 = \ddot{F}_1^d - \ddot{F}_1 = \ddot{e}_1$$
$$\ddot{F}_2^d - \delta_2 = \ddot{F}_2^d - \ddot{F}_2 = \ddot{e}_2 \quad (28)$$

Therefore, the closed-loop error dynamics is defined as:

$$0 = \ddot{e}_1 + \rho_1 \dot{e}_1 + \rho_0 e_1$$
$$0 = \ddot{e}_2 + \tilde{\rho}_1 \dot{e}_2 + \tilde{\rho}_0 e_2 \quad (29)$$

To guarantee exponential stability, the control gains were chosen such that all the roots of the equations in (29) lie in the left half-plane of the complex plane.

C. Flatness-based Formation Controller

Desired trajectories of the flat outputs of the followers can be constructed using the position output of the leader and predefined distance l^{ref} and orientation ϕ^{ref} with respect to the leader robot. For the first follower (follower1):

$$\begin{bmatrix} F_{1\ ref}^{f1} \\ F_{2\ ref}^{f1} \end{bmatrix} = \begin{bmatrix} F_1 - l^{ref1} cos(\phi^{ref1} + \theta_{i1}) \\ F_2 - l^{ref1} sin(\phi^{ref1} + \theta_{i1}) \end{bmatrix} \quad (30)$$

And for the second follower (follower2):

$$\begin{bmatrix} {F_1}^{f2}{}_{ref} \\ {F_2}^{f2}{}_{ref} \end{bmatrix} = \begin{bmatrix} F_1 - l^{ref2} cos(\phi^{ref2} + \theta_{i1}) \\ F_2 - l^{ref2} sin(\phi^{ref2} + \theta_{i1}) \end{bmatrix} \tag{31}$$

Having constructed the desire trajectories, the next step is to define the error.

$$\begin{aligned} {e_1}^{f1} &= {F_1}^{f1}{}_{ref} - {F_1}^{f1} \\ {e_2}^{f1} &= {F_2}^{f1}{}_{ref} - {F_2}^{f1} \\ {e_1}^{f2} &= {F_1}^{f2}{}_{ref} - {F_1}^{f2} \\ {e_1}^{f2} &= {F_2}^{f2}{}_{ref} - {F_2}^{f2} \end{aligned} \tag{32}$$

Let $\begin{bmatrix} \ddot{F}_1^{fs} \\ \ddot{F}_2^{fs} \end{bmatrix} = \begin{vmatrix} {\delta_1}^{fs} \\ {\delta_2}^{fs} \end{vmatrix}$ then the feedback control laws to the new inputs can then be defined as

$$\begin{aligned} {\delta_1}^{fs} &= {\ddot{F}_1^{fs}}_{ref} + {\rho_1}^{fs} {\dot{e}_1}^{fs} + {\rho_0}^{fs} {e_1}^{fs} \\ {\delta_2}^{fs} &= {\ddot{F}_2^{fs}}_{ref} + {\tilde{\rho}_1}^{fs} {\dot{e}_2}^{fs} + {\tilde{\rho}_0}^{fs} {e_2}^{fs} \end{aligned} \tag{33}$$

Where ${\overset{\sim}{\rho}_1}^{fs}$, ${\overset{\sim}{\rho}_0}^{fs}$, ${\rho_1}^{fs}$, ${\rho_0}^{fs}$ are control the gains and subscript s = [1, 2] specifies the robot. Next the error dynamics of the system in flat output space is determined.

$$\begin{aligned} {\ddot{F}_1^{fs}}_{ref} - {\delta_1}^{fs} &= {\ddot{F}_1^{fs}}_{ref} - \ddot{F}_1 = {\ddot{e}_1}^{fs} \\ {\ddot{F}_2^{fs}}_{ref} - {\delta_2}^{fs} &= {\ddot{F}_2^{fs}}_{ref} - \ddot{F}_2 = {\ddot{e}_2}^{fs} \end{aligned} \tag{34}$$

Therefore, the error dynamics is defined as:

$$\begin{aligned} 0 &= \ddot{e}_1^{fs} + \rho_1^{fs} \dot{e}_1^{fs} + \rho_0^{fs} e_1^{fs} \\ 0 &= \ddot{e}_2^{fs} + \tilde{\rho}_1^{fs} \dot{e}_2^{fs} + \tilde{\rho}_0^{fs} e_2^{fs} \end{aligned} \tag{35}$$

To track the reference trajectory with added disturbances or uncertainty, [1] defines a fd which represents the set of variables (x and u) of the actual follower robot that is set to converge to their reference $\boldsymbol{F_L^d(t)}$. The control law and error e_s where $\boldsymbol{e_s = F_L^d - F_f}$, are such that

$$\dot{\boldsymbol{e}}_{\boldsymbol{s}}(\boldsymbol{t}) = \boldsymbol{f_d}\Big(\boldsymbol{e_s(t)} + \boldsymbol{F_L^d(t)}, \boldsymbol{u}\Big(\boldsymbol{e_s(t)} + \boldsymbol{F_L^d(t)}\Big)\Big) - \boldsymbol{f}\Big(\boldsymbol{F_L^d(t)}, \boldsymbol{u(t)}\Big)\Big)$$

is asymptotically stable for all disturbances.

4 Simulation and Results

MATLAB/Simulink simulation software is used to demonstrate the effectiveness of the flatness controller. Two main cases are presented in this section. Firstly, the ability of leader to follow its predefined trajectories and secondly, the ability of the followers to maintain a desired distance and orientation in relation to the leader robot. As previously mentioned, the system is made up of three similar robots, one of the robots is a leader and the other two are followers.

A. Effect of a Flatness Controller on the Leader Robot

Using fifth-degree polynomial together with terminal conditions shown in Table 1 below, the desired trajectories have been derived (Eq. 24). The robot should reach its desired destination in fifteen seconds, thus $t = 15$ s. The robot is driven from rest-to-rest positions.

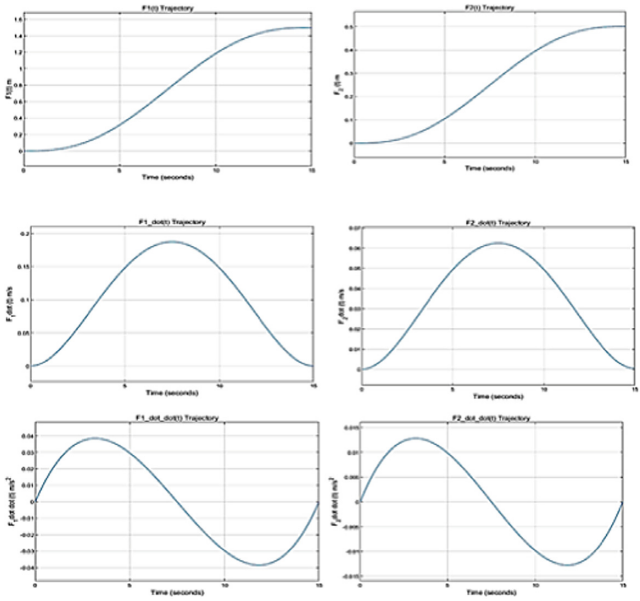

Fig. 2. ader robot Flat Outputs Trajectories.

Figure 2 therefore shows the desired trajectories of flat outputs F_1 and F_2, together with their desired velocities and accelerations (Table 1).

Table 1. Leader's Desired Terminal Conditions.

State	Initial condition	Final condition
x(m)	0.0	1.5
y(m)	0.0	0.5
θ(rad)	0.0	0.0
v(m/s)	0.0	0.0
ω(rad/s)	0.0	0.0
$\overline{u}$**(m/s^2)**	0.0	0.0

Furthermore, the effectiveness of this controller was tested. A comparison was made between WMR's open-loop response, which is the response of WMR without a controller, and its response with a flatness-based controller. The flatness controller gains were chosen to be:

$$\rho_{0x} = 2, \rho_{1x} = 1, \widetilde{\rho}_{1y} = 6, \widetilde{\rho}_{0y} = 9$$

The open-loop response is shown in Fig. 3. It can be seen that although the desired final terminal condition of the x-coordinate (F1) is 1.5m, the actual value reached is 0.78m. Also, instead of the desired 0.5m, the final y-coordinate (F2) value reached by the WMR was 1.25m. Evidently, in the absence of a controller, the robots fails to follow the desired trajectory. Contrarily, in Fig. 4, the flatness-based control performs well and reaches the desired trajectory in about 0 s with the error converging to zero. Consequently, the robot can follow the desired trajectory accurately.

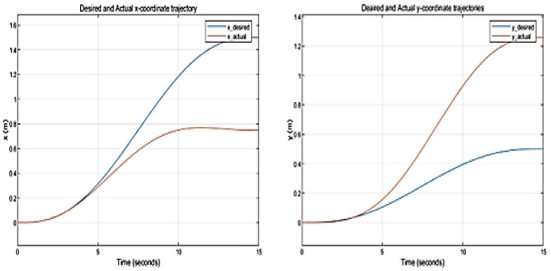

Fig. 3. Leader's Flat output Trajectory tracking open-loop response.

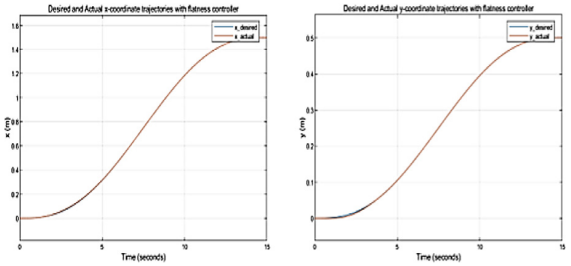

Fig. 4. Leader's Flat output Trajectory tracking response with flatness-based controller.

B. Effect of a Flatness based Controller on the Robotic Formation

To maintain formation, the followers track the leader, while maintaining a constant bearing ϕ^{ref} and distance d^{ref} to the leader. Using the trajectory of the leader and the desired bearing and distance from the leader, terminal conditions and desired trajectories of the followers can be constructed. Table 2 shows these terminal conditions, together with the desired distance and orientation of follower to the leader. Figure 5 then presents the reference trajectories of the followers such that formation is maintained and the

desired terminal conditions in Table 2 are met. The flatness property of each robot is established, and flatness controllers are used on each robot to track these trajectories. Here again a comparison between WMR's open-loop response and a flatness-based controller response is made. Figure 6 shows the F_1 and F_2 open-loop response of follower1 and follower2. From this figure compares the F_1 and F_2 desired and actual trajectories. The two robots fail to track the desired trajectories, therefore there is a need for a controller.

Recall that the desired trajectories are calculated such that they ensure maintenance of a constant distance and orientation relative to the leader. Therefore, failure for the follower robots in tracking these trajectories means failure in the maintenance of the predefined formation. This is evident of the errors in the separation distance and bearing from the leader as seen in Fig. 7 and Fig. 8. It can be seen from the two figures that the distance and orientation errors do not converge to zero. This is a clear indication that the three robots are out of formation.

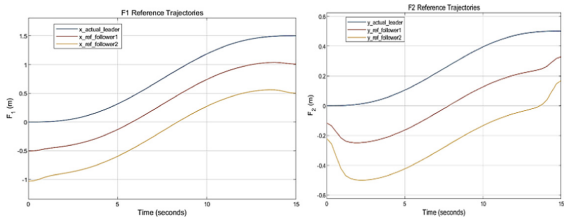

Fig. 5. F_1 and F_2 Formation Reference Trajectories for Formation Maintenance.

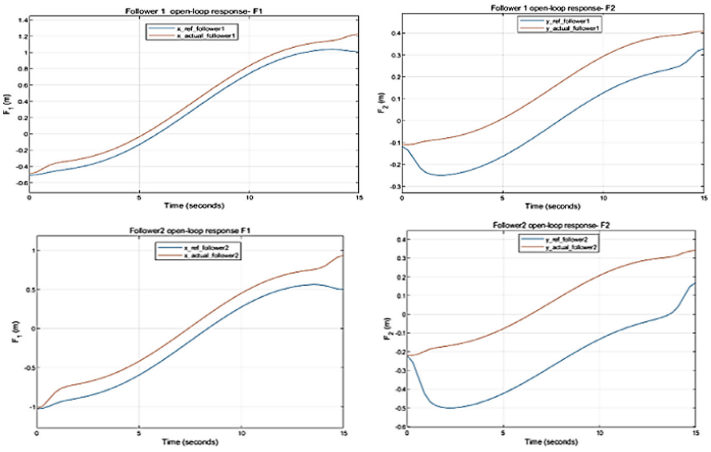

Fig. 6. followers' open loop response: F_1 and F_2 trajectory tracking.

The distance and bearing errors are significant, as a result formation is not achieved. It is therefore evident that there is a need for a controller in order to compensate for the open-loop insufficiency. As a result, a flatness controller was designed for each follower.

Table 2. Desired Terminal Conditions for followers.

Robot	Initial (F1)m	Final (F1) m	Initial (F2) m	Final (F2) m	Distance (d^{ref})m	Bearing (ϕ^{ref}) rad
Leader	0	1.5	0	0.5	–	–
Follower 1	−0.49	1	−0.11	0.33	0.52	0.2276
Follower2	−1.03	0.17	−0.22	0.5	1.054	0.2109

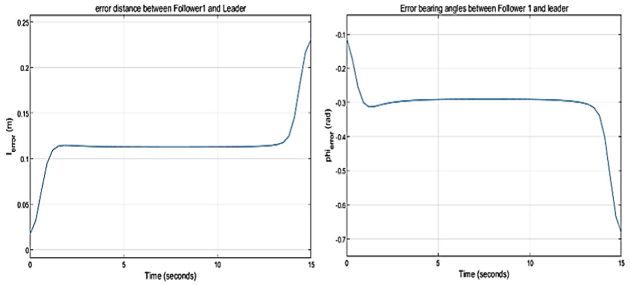

Fig. 7. l^{ref} and φ^{ref} Open-Loop Error Response of Follower1.

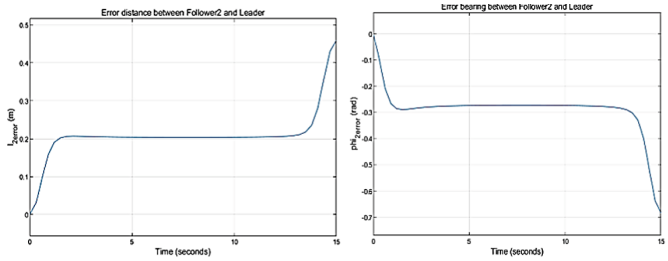

Fig. 8. l^{ref} and φ^{ref} Open-Loop Error Response of Follower2.

The gains for follower1 controller were chosen to be:

$$\rho_0 = 3, \rho_1 = 2, \tilde{\rho}_1 = 5, \tilde{\rho}_0 = 8$$

Also, the gains for follower2 controller were chosen to be:

$$\rho'_0 = 2, \rho'_1 = 1, \tilde{\rho'}_1 = 6, \tilde{\rho'}_0 = 9$$

Figure 9 shows the tracking response of the follower robots when flatness-based controllers are used. Furthermore, in this case, it is evident the tracking ability of the robot has improved; the robots successfully track the desired trajectories. Thus, the distance and bearing errors converge to zero and thus a constant distance and bearing is maintained (Fig. 10). Both robots can maintain the desired distance and bearing relative to the leader, thus formation was successfully maintained.

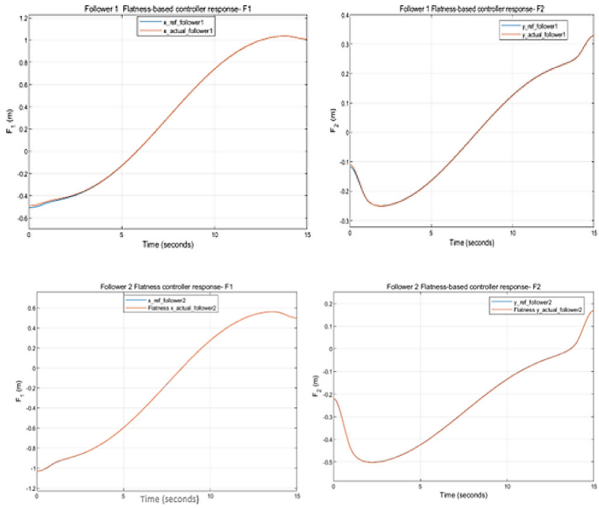

Fig. 9. Flatness-Based Trajectory Tracking Response of the Follower Robots.

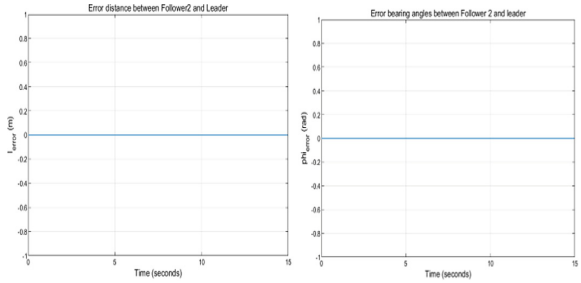

Fig. 10. Separation Distance l^{ref} and Separation Bearing φ^{ref} Error Response for the Followers with Flatness-Based controller.

5 Conclusion

This paper presented coordination control of a leader-follower cooperative multiple mobile robotic system using differential flatness theory. Simulations are presented to demonstrate the effectiveness of the approach. The formation is made up of three differentially driven mobile robots with similar dynamics. One robot is a leader, while the other two are followers. Using their kinematic models, the robots' flatness properties were exploited to design a flatness-based control algorithm for motion planning. Moreso, trajectories were generated and tracked such that the follower robots maintained a constant desired relative distance and orientation with reference to the leader. Based on the results obtained, differential flatness characterisation enabled the linearization of the system to a stable linear equivalent system and significantly reduced the computational cost especially when dealing with multiple robots that can otherwise entail solving large robotic model differential equations. For further study, the effects of flatness trajectory

planning would be tested for the mobile robots under friction, or with modelling errors, unreliable estimates, and additional perturbation.

References

1. Markus, E.D., Yskander, H., Agee, J.T., Jimoh, A.A.: Coordination control of robot manipulators using flat outputs. Robot. Auton. Syst. **83**, 169–176 (2016)
2. Tsiu, Lintle, Markus, Elisha Didam: a survey of formation control for multiple mobile robotic systems. Int. J. Mech. Eng. Robot. Res. 1515–1520 (2020). https://doi.org/10.18178/ijmerr.9.11.1515-1520
3. Petrovsky, A., Kalinov, I., Karpyshev, P., Tsetserukou, D., Ivanov, A., Golkar, A.: The two-wheeled robotic swarm concept for Mars exploration. Acta Astronaut. **194**, 1–8 (2022)
4. Li, Y., et al.: Development and applications of rescue robots for explosion accidents in coal mines. J. Field Robot. **37**(3), 466–489 (2020)
5. Lei, G., Zheng, Y.: Research on cooperative trajectory planning algorithm based on Tractor-Trailer Wheeled Robot. IEEE Access **10**, 64209–64221 (2022). https://doi.org/10.1109/ACCESS.2021.3062392
6. Hausman, K., Müller, J., Hariharan, A., Ayanian, N., Sukhatme, G.S.: Cooperative multi-robot control for target tracking with onboard sensing. Int. J. Robot. Res. **34**(13), 1660–1677 (2015)
7. Panagou, D., Kumar, V.: Cooperative visibility maintenance for leader–follower formations in obstacle environments. IEEE Trans. Rob. **30**(4), 831–844 (2014)
8. Jain, R.P., Aguiar, A.P., de Sousa, J.B.: Cooperative path following of robotic vehicles using an event-based control and communication strategy. IEEE Robot. Autom. Lett. **3**(3), 1941–1948 (2018)
9. Wei, H., Lv, Q., Duo, N., Wang, G., Liang, B.: Consensus algorithms based multi-robot formation control under noise and time delay conditions. Appl. Sci. **9**(5), 1004 (2019)
10. Wang, G., Wang, C., Du, Q., Li, L., Dong, W.: Distributed cooperative control of multiple nonholonomic mobile robots. J. Intell. Rob. Syst. **83**(3), 525–541 (2016)
11. Liu, L., Yu, J., Ji, J., Miao, Z., Zhou, J.: Cooperative adaptive consensus tracking for multiple nonholonomic mobile robots. Int. J. Syst. Sci. **50**(8), 1556–1567 (2019)
12. Du, H., Wen, G., Cheng, Y., He, Y., Jia, R.: Distributed finite-time cooperative control of multiple high-order nonholonomic mobile robots. IEEE Trans. Neural Networks Learn. Syst. **28**(12), 2998–3006 (2016)
13. Akbari, A., Muhayyuddin, Rosell, J.: Knowledge-oriented task and motion planning for multiple mobile robots. J. Experim. Theoret. Artific. Intell. **31**(1), 137–162 (2019)
14. Gutiérrez, H., Morales, A., Nijmeijer, H.: Synchronization control for a swarm of unicycle robots: analysis of different controller topologies. Asian J. Control **19**(5), 1822–1833 (2017)
15. Xiang, X., Liu, C., Lapierre, L., Jouvencel, B.: Synchronized path following control of multiple homogenous underactuated AUVs. J. Syst. Sci. Complexity **25**(1), 71–89 (2012)
16. Ge, H., Chen, G., Xu, G.: Multi-AUV cooperative target hunting based on improved potential field in a surface-water environment. Appl. Sci. **8**(6), 973 (2018)
17. Zhao, R., Xiang, X., Yu, C., Jiang, Z.: Coordinated formation control of autonomous underwater vehicles based on leader-follower strategy. In: OCEANS 2016 MTS/IEEE Monterey, pp. 1–5. IEEE (2016)
18. Tang, C.P.: Differential flatness-based kinematic and dynamic control of a differentially driven wheeled mobile robot. In: 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2267–2272. IEEE (2009)
19. Sira-Ramirez, H., Agrawal, S.K.: Differentially Flat Systems. Crc Press (2018)