Abstract—This article presents a novel control strategy to herd groups of noncooperative evaders by means of a team of robotic herders. In herding problems, the motion of the evaders is typically determined by strongly nonlinear and heterogeneous reactive dynamics, which makes the development of flexible control solutions a challenging problem. In this context, we propose Implicit Control, an approach that leverages numerical analysis theory to find suitable herding inputs even when the nonlinearities in the evaders’ dynamics yield implicit equations. The intuition behind this methodology consists in driving the input, rather than computing it, toward the unknown value that achieves the desired dynamic behavior of the herd. The same idea is exploited to develop an adaptation law, with stability guarantees, that copes with uncertainties in the herd’s models. Moreover, our solution is completed with a novel caging technique based on uncertainty models and control barrier functions, together with a distributed estimator to overcome the need of complete perfect measurements. Different simulations and experiments validate the generality and flexibility of the proposal.

Index Terms—Adaptive control, control theory, herding, multirobot systems (MRS).

I. INTRODUCTION

RECENT advances in multirobot systems (MRS) have favored the development of successful control strategies in real-life problems, such as entrapment [1], hunting [2], or escorting [3]. Despite the different nature of scenarios, these problems can all be seen as different instances of herding [4], where the objective is to drive a group of targets or evaders to specific locations using a team of robots or herders. Common to all of them is the noncooperative nature of the evaders with respect to the control objective, typically entangled in complex nonlinear and heterogeneous behaviors. In fact, the difficulties hidden in the herding problem have motivated broad interdisciplinary research assembling physiologists, mathematicians, and neurologists with engineers [5]–[7].

The authors are with RoPeRT group, DIIS - I3A, Universidad de Zaragoza, 50018 Zaragoza, Spain (e-mail: esebastian@unizar.es; emonti@unizar.es; csag ues@unizar.es).

Manuscript received 28 March 2022; accepted 3 June 2022. See https://www.ieee.org/publications/rights/index.html for more information.
This article is an evolved version of [8]. Compared to it, the theoretical development of Implicit Control is more comprehensive, including a stability analysis for general systems that were not discussed in [8]. The ability to deal with uncertain models using an adaptation law is also new. In addition, we formulate an extended version of the distributed Kalman filter (E-DKF) [9] to overcome the need of complete perfect measurements. The filter embeds the Implicit Control strategy in the prediction stage and works with input-dependent measurements, two properties that are atypical and innovative in this context. Furthermore, the caging technique combining uncertainty models and CBFs is also novel. We have added new simulated experiments, where we validate the caging technique and the distributed estimator. Furthermore, it is shown that Implicit Control can be exploited to herd several evaders with a few herders by simply changing the control objective. Finally, we include new experimental results that validate the herding solution.

II. RELATED WORK

To discuss the related work, we distinguish between 1) single-agent or multiagent herding solutions, 2) herding of evaders with linear or nonlinear dynamics, and 3) herding with known or unknown evaders’ dynamics.

Most of the existing works deal with the herding of a single entity. An early example is [1], which employs a null space-based behavioral control to escort/entrap the target. Zhu et al. [2] apply a bio-inspired neural network to hunt in an underwater environment. To produce an escorting behavior, Du et al. [10] develop a distributed switching strategy, where the escorting task is assigned from one herder to another when a Voronoi boundary is crossed. Escorting shares properties with the encirclement in [11], where decentralized geometrical strategies are derived depending on the evaders’ configuration.

Few papers solve the herding of more than one evader. In [12], a group of robots navigates around a certain area to prevent evaders to cross its boundaries. The work in [4] drives groups of entities by an active encirclement but does not consider specific final positions for each evader. Following a similar approach, Chipade and Panagou [13] go a step further and propose an active encirclement that avoids obstacles. With the same spirit, Stilwell et al. [14] and Antonelli and Chiaverini [15] develop solutions to drive a whole herd toward a certain region, where it is required that some global features of the herd (e.g., mean position) converge to the desired equilibrium. Besides, both works deal with solutions that have self-controllable entities, i.e., the controller, state to control, and input are all embedded in the same entity. Recently, Auletta et al. [16] exploits simple local rules to push all herders toward the same containment area. In [17], a single robot is in charge of herding multiple evaders by controlling them one by one, whereas Licitra et al. [18] extend the solution with a neural network that learns the dynamics of the targets. The herding in [19] considers a single robot to control a herd as well, but in this case, the evaders cooperate with the robotic leader to follow it. Song et al. [20] present an approach based on algebraic topology called “herding by caging,” where repulsive forces are leveraged to steer the evaders, again, toward a certain general region. In contrast, our method performs the herding of all the evaders to precise individual locations simultaneously. Additionally, our approach can also be used to steer large herds toward a global region or to split and steer each subherd toward individually assigned regions simultaneously.

Another feature of herding problems is the behavior of the evaders, where two assumptions are often considered: linear and homogeneous dynamics. The first assumption is considered in, e.g., [21], where the containment of linear heterogeneous agents is performed by a time-varying formation. The work in [22] drives a group of linear unmanned ground vehicles to ensure that a region is completely surveyed, whereas in [23], both pursuers and evaders have the same linear dynamics, solving the problem with geometric tools. The second assumption is used in [4], where agents with nonlinear homogeneous repulsive dynamics are controlled using a team of robots. A different instance is [24], where a complete control structure is presented, from the hexrotor motion to the pursuit layer. Our proposal offers a general framework to design a control strategy for the herding of general nonlinear and heterogeneous evaders.

Most of the aforementioned solutions assumes perfect knowledge of the herd dynamics. To deal with uncertainty, a common approach is the design of model-free strategies. The work in [25] localizes the target with relative-position sensors and then applies probabilistic tools for the encirclement. Similarly, Zhu et al. [26] include information from a camera to learn the behavior of the target. These instances deal with the herding of a single target. An extension to $n$ robots detecting $n$ targets can be found in [27], using a mixture of learning, optimization and fuzzy techniques. The use of gene regulatory networks is studied in [28] to entrap groups of targets. In both cases, no accuracy in the final position of the evaders is required, and in [28], the targets do not react against the robots’ efforts. This lack of accuracy and flexibility in the number of evaders under uncertainty is solved in our proposal by an adaptive technique. Thus, model information is exploited for the precise herding of repulsive heterogeneous evaders while uncertainty is tackled by adapting online the unknown parameters.

It is not the first time that the term “implicit control” appears in the literature, but this is never done in terms of a novel control technique, unlike our coined Implicit Control. Estrada and Manjarres [29] deal with a time-varying SISO system by first designing a trivial time-invariant controller and then reformulating it to account for unknown parameters. Therefore, a classical linear control law is reshaped to implicitly operate irrespective of the time-varying parameters. Meanwhile, in [30] and [31], the term “Implicit Control” is inspired by the fast adaptation that emerges in living entities. In such systems, the frontiers between plant and controller are fuzzy. They refer to “Implicit Control Law” as that abstract intersection. In contrast, our “Implicit Control” is a novel control technique that considers a general implicit equation to describe the control objective and then develops an expression for the time-derivative of the input such that the implicit equation converges toward its solution. That convergence implicitly implies that the state converges to the desired equilibrium, with the desired properties encoded in the implicit equation.
Lastly, regarding the control technique, we propose in this article, recent works in the control theory community support the advances in developing solutions where the control expression is in terms of the dynamics of the input instead of the classical algebraic expression. In [32], the time derivative of the input is used to characterize and tune static plants. However, to the best of our knowledge, this work has never been applied in dynamical systems to build an explicit controller. Meanwhile, Shivan et al. [33] show the advantages of this novel control perspective in applications with highly nonlinear input-nonaffine dynamics, with an input determined by the “flow” that a Newton–Raphson solver will follow to arrive at the roots of a nonlinear equation. Another example is [34], where a similar approach is applied in a tracking problem. Our proposal, sustained in [8], goes a step further in generality. We give formal proofs of existence and stability that are then corroborated in both simulations and real experiments, solving a problem where, unlike the aforementioned papers, the control objective is evasive with respect to the inputs. In addition, the dynamics to be controlled are partially unknown, which is also a distinguishing factor.

III. PROBLEM STATEMENT

We consider the problem of herding a group of \( m \) evaders using a team of \( n \) robotic herders. We denote the evaders by \( j \in \{1, \ldots, m\} \), and the herders by \( i \in \{1, \ldots, n\} \). Since we aim at using the position of the herders to control the position of the evaders, we define the state \( x \in X \subseteq \mathbb{R}^{2n} \) and the input \( u \in U \subseteq \mathbb{R}^{2n} \) as \( x = \left[ x_1^T \ldots \ x_m^T \right]^T \) and \( u = \left[ u_1^T \ldots \ u_n^T \right]^T \), where \( x_j = [x_j^e, y_j^e] \in X_j \subseteq \mathbb{R}^2 \) is the position of evader \( j \) and \( u_i = [x_i^h, y_i^h] \in U_i \subseteq \mathbb{R}^2 \) is the position of herder \( i \), operating in a 2-D space. The movement of each evader is described by general dynamics

\[
\dot{x}_j = f_j(x, u) \tag{1}
\]

allowing for any nonlinear behavior encoded in \( f_j(x, u) \). Notice that we account for the possibility of nonlinear dependencies on the input. The only assumption regarding \( f_j \) is that it is of class \( C^1 \) for all \( j \). We exemplify (1) using two dynamic models from the literature. Their choice is motivated by the strongly nonlinear behavior in the position of evaders and herders, described by input-nonaffine dynamics.

The Inverse Model (adapted from [4]) is

\[
\dot{x}_j = f_j^{inv}(x, u) = \theta_j \sum_{i=1}^{n} \frac{d_{ij}}{|d_{ij}|^\beta} \tag{2}
\]

where \( d_{ij} = x_j - u_i \) is the relative position between evader \( j \) and herder \( i \), and \( \theta_j \) is a positive constant that expresses the aggressiveness in the repulsion provoked by the herders. Note that despite the model has a singularity in \( d_{ij} = 0 \), in practice, the speed remains bounded. The repulsion grows with \( \frac{1}{|d_{ij}|^\beta} \), so the closer the herders, the larger the repulsion. The only way of achieving \( d_{ij} = 0 \) is that a perfectly evenly distributed number of herders approaches the evader. This does not happen in practice because herders need to impose nonzero repulsive forces to steer the evaders.

The Exponential Model (adapted from [17]) is

\[
\dot{x}_j = f_j^{exp}(x, u) = \theta_j \sum_{i=1}^{n} \frac{d_{ij} e^{-\chi_{ij}} (1 - \beta_j \text{sign}(-|d_{ij}| + d_{min}))} \tag{3}
\]

where \( \chi_{ij} = \frac{1}{\sigma_j} d_{ij}^2 d_{ij} \) and \( \sigma_j > 1 \). In this model, there is a switching condition if \(|d_{ij}| \leq d_{min}\), where the evader \( j \) becomes “scared” when the distance with some herder \( i \) is smaller than \( d_{min} \) and the intensity of the repulsive interaction increases, due to \( 0 < \beta_j < 1 \). In practice, the sigmoid function \( \text{sign}(\cdot) \) is included in (3) to avoid the violation of the \( C^1 \) assumption. In a similar fashion, a sigmoid function or an hyperbolic tangent can be used to model saturation in the speed of the evaders.

Given the individual dynamics of the evaders in (1), the joint system dynamics can be defined as

\[
\dot{x} = f(x, u) \tag{4}
\]

where \( f(x, u) \) simply comes from stacking all \( f_j(x, u) \). This formulation allows us to consider heterogeneous herds, with different number of evaders and motion models.

Our goal is to herd the evaders toward the desired positions \( x^* \in X \) simultaneously. To do this, we define the position error \( \hat{x} = x - x^* \), and we set the control objective to be to drive \( \hat{x} \) to zero. This is what we call precise herding. A particular instance of precise herding is the herding of a herd toward the same region, where some of the elements of \( x^* \) are specified by the practitioner and the rest of them are free or directly dependent on the others. Nevertheless, in general, we consider the case where each evader \( x_j \) has an individually assigned \( x^*_j \).

It is noteworthy that the reactive behavior of the evaders is with respect to the position of the herders. Therefore, a control strategy that determines inputs in terms of herders’ positions is adequate to generalize the solution to different robotic platforms. This high-level approach can be combined with any robot-specific low-level controller.

To keep the generality of the solution, in this work, we assume that the maximum velocity of both herders and evaders is \( v_{\text{max}} \). This requires an initial caging phase, surrounding the evaders to avoid their escape before the precision herding begins. It is not mandatory to achieve a compact and closed encirclement to succeed but just the distribution of the herders near the evaders. This is the reason for developing a caging stage to complete the herding, described in Section VI-B.

IV. IMPLICIT CONTROL

Herding seeks an expression for the input \( u \) such that the evaders go to their desired positions \( x^* \). Besides, the herding may need to accomplish other requirements, such as the desired transient response. This can be translated into designing \( u \) such that the evaders follow the desired dynamics \( f^* \):

\[
\dot{x} = f^*(x) \tag{5}
\]
To ease the analysis, we consider, without loss of generality, \( \mathbf{x}^* = \mathbf{0} \) as the desired equilibrium point of the system. The function \( f^* \) can adopt any desired structure\(^2\) with the constraint of being stable in the equilibrium point.

In order to control the evaders, the typical practice consists in finding a closed expression for \( \mathbf{u} \) that depends only on the state and maps (4) into (5). Unfortunately, due to the complexity of the system, sometimes it is extremely challenging and time consuming to find such explicit mapping. Indeed, trying to find it generally yields to an implicit expression for the input. The analytical solution of the latter is not guaranteed, and the mechanisms to calculate such input do not certify that the desired properties prevail anymore. In contrast, we propose Implicit Control as a general control solution to achieve desired closed-loop dynamics despite the complexity of (4), overcoming the necessity of looking for a controller to perform the mapping from (4) to (5).

We study the conditions that allow to find \( \mathbf{u} \) such that the evaders evolve according to \( f^* \). Then, we propose a design procedure to solve the control input. The description is kept in general control terms since we believe that this procedure can be of interest in other control problems.

### A. Control Existence

First, it is necessary to know if there exists a smooth input that makes the actual dynamics equal to the desired ones. For convenience, let define

\[
    h(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x})
\]  
(6)

as the working equation, shifting the analysis to that of looking for the existence of \( \mathbf{u} \) such that \( h(\mathbf{x}, \mathbf{u}) = 0 \), i.e., looking for the existence of the roots of \( h(\mathbf{x}, \mathbf{u}) \).

At this point, a straightforward solution can be to compute the input using a numerical method to find the roots of \( h \). This is a simple and powerful approach, but lacks of formal guarantees, depends on the numerical method and is computationally expensive since we must ensure that the numerical method finds the roots of \( h \) at all instants. Therefore, we will use this approach as a baseline to compare the Implicit Control solution.

In any case, the first step is to determine the existence of the roots of \( h \), which is, in general, not trivial. We review some fundamentals of numerical analysis theory to show sufficient conditions to ensure existence and smoothness\(^3\) of \( \mathbf{u} \). The Implicit Function Theorem (see [35, Th. 9.28]), applied to \( h \), offers a set of sufficient conditions to guarantee the existence and smoothness of \( \mathbf{u} \) in a local region.

**Theorem 1 (Adapted from Theorem 9.28 of [35]):** Let

\[
    h : I = \mathbf{X} \times \mathcal{U} \subset \mathbb{R}^M \times \mathbb{R}^N \rightarrow \mathbb{R}^N
\]  
(7)

a \( C^1 \)-mapping, such that \( h(\mathbf{x}^*, \mathbf{u}_0^*) = 0 \) for some point \( \mathbf{u}_0^* \in \mathcal{U} \). Additionally, consider the Jacobian

\[
    \mathbf{J} = (J_x | J_u) = \begin{pmatrix}
        \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_M} & \frac{\partial h_1}{\partial u_1} & \cdots & \frac{\partial h_1}{\partial u_N} \\
        \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
        \frac{\partial h_M}{\partial x_1} & \cdots & \frac{\partial h_M}{\partial x_M} & \frac{\partial h_M}{\partial u_1} & \cdots & \frac{\partial h_M}{\partial u_N}
    \end{pmatrix}
\]  
(8)

such that \( \mathbf{J}_u \) is nonsingular in the point \( (\mathbf{x}^*, \mathbf{u}_0^*) \). Then, there exist open subsets \( I^* \subset \mathbb{R}^M \times \mathbb{R}^N \) and \( \mathbf{x}^* \subset \mathbb{R}^M \) with \( (\mathbf{x}, \mathbf{u}^*) \in I^* \) and \( \mathbf{x} \in \mathbf{x}^* \), having the following property: to every possible \( \mathbf{x} \in \mathbf{x}^* \) corresponds a unique \( \mathbf{u}^* \) such that \( (\mathbf{x}, \mathbf{u}^*) \in I^* \) and \( h(\mathbf{x}, \mathbf{u}^*) = 0 \).

The theorem imposes three conditions to be fulfilled. First, there must exist an input, \( \mathbf{u}_0^* \), which solves the control in \( \mathbf{x}^* \). In the herding context, there must exist a stable configuration of the herders when the evaders are in their desired positions. A general condition to ensure this, for all \( \mathbf{x} \in \mathbf{X} \), is to have at least the same number of inputs than states to control. Otherwise, it is not always possible to control the system because there are less degrees of freedom than states to control. In the case of the precision herding problem, this means \( n \geq m \), since both evaders and herders are first-order entities in the space and we want to control all the evaders simultaneously; i.e., at all instants, all the herders contribute to the motion of all the evaders. Besides, a team of herders is needed so that, at equilibrium, the total repulsive force in each evader is zero. Nevertheless, we demonstrate in Section VII that Implicit Control can also be used to control large-scale herds with only a few robotic herders if the control objective is the position of the centroid of the herd.

Second, \( h \) must be of class \( C^1 \) in \( (\mathbf{x}^*, \mathbf{u}_0^*) \). If \( f^* \) is chosen of class \( C^1 \) in \( \mathbf{x}^* \), then the condition is accomplished because \( f_j \) in (1) is of class \( C^1 \) for all \( j \), so \( f \) in (4) is of class \( C^1 \).

The last condition requires the Jacobian of \( h \) with respect to \( \mathbf{u} \), \( \mathbf{J}_u \), to be nonsingular in the desired location. Since for \( m \neq n \), the matrix is not square, the right Moore–Penrose inverse matrix \( \mathbf{J}_u^+ = \mathbf{J}_u^T (\mathbf{J}_u \mathbf{J}_u^T)^{-1} \) is generally considered as the one to be nonsingular. An alternative is to resort to the Constant Rank Theorem (see [36, Th. 5.22]) and find a projection of lower dimension that locally preserves the properties of the Jacobians but achieving a square matrix. The use of the pseudoinverse can be seen as this projection. Given the aforementioned features of \( h \) and \( \mathbf{J}_u \), the last condition is accomplished in \( \mathbf{x}^* \). Moreover, by restricting \( I \) to the subspace without collisions, it is ensured that the two last conditions of Theorem 1 hold for all \( \mathbf{x} \) and, therefore, the theorem holds for all \( \mathbf{x} \) in this subspace. Considering that each herder provokes a repulsive reaction in every evader, collision among herders and evaders will not happen. Similarly, since herders are the controlled entities, it is easy to prevent collisions among them.

Theorem 1 is an analytical tool to guarantee the necessary conditions to afford the Implicit Control proposal. Further analysis of the existence of input will depend on the particularities of the controlled system. Therefore, we consider the following assumption, so that a suitable input can be computed.
Assumption 1: There exists \( u_0 \) such that \( h(x, u_0) = 0 \), the functions \( f \) and \( f^* \) are of class \( C^1 \) and \( J_u \) has rank \( m \) for all \( (x, u) \in I \).

The problems of the uniqueness of solution or existence of local minima are not a concern since no cost function is being optimized. We underline this to emphasize that Implicit Control does not search for any local or global optimum, but for the roots of \( h \).

**B. Control Design**

The proposed design method consists in expanding the initial system in (4) with an input dynamics that converges to the roots of \( h \). This transforms the problem to that of computing the input \( u \) as part of an expanded explicit system, described in continuous time and with analytical solution. To ease the presentation, we first assume \( n = m \), generalizing to \( n \neq m \) later.

The function \( h \) is determined by the desired closed-loop behavior \( f^* \). In an ideal scenario, \( h = 0 \) always holds. However, achieving \( h = 0 \) for all time is generally not an easy task. Our solution consists in considering \( h \) as a dynamic system. The evolution of \( h \) over time is not determined a priori; therefore, the input dynamics can be designed such that the evolution of \( h \) follows the desired dynamics \( h^* \):

\[
\frac{dh(x, u)}{dt} = h^*(x, u) \tag{9}
\]

so that \( h \) converges to zero. From now on, we assume that \( h^* \) is chosen to make \( h \) stable. With this in mind, we propose the following expression for the input dynamics:

\[
\dot{u} = J_u^{-1}(h^*(x, u) - J_x f(x, u)) \tag{10}
\]

where \( J_x \) is the Jacobian of \( h \) with respect to \( x \) and \( J_u \) is the Jacobian of \( h \) with respect to \( u \). This allows to build an explicit expanded system of the form

\[
\begin{aligned}
\dot{x} &= f(x, u) \\
\dot{u} &= J_u^{-1}(h^*(x, u) - J_x f(x, u))
\end{aligned} \tag{11}
\]

The control problem is then reduced to analyzing the stability of the system in (11). With the proposed expansion, the complete system becomes autonomous, i.e., the input becomes part of the state of the expanded system. As the structure in (11) is very general, we provide stability results for typical dynamics. The first one considers that the original system is input-to-state stable (ISS).

**Theorem 2:** If the system defined in (4) is ISS and Assumption 1 holds, then the input dynamics in (10) ensures the convergence of \( h \) toward zero and achieves the desired closed-loop dynamics in (5).

**Proof:** Applying the chain rule over \( dh/dt \) yields to

\[
\frac{dh(x, u)}{dt} = J_x(x, u) \frac{dx}{dt} + J_u(x, u) \frac{du}{dt} \tag{12}
\]

where \( J_x \) and \( J_u \) are the Jacobians of \( h \) with respect to \( x \) and \( u \). By Assumption 1, \( J_u \) is full rank. Then, the substitution of the input dynamics in (10) gives (9), meaning that \( \dot{u} \) imposes the desired dynamics over \( h \).

This ensures that, at some point, \( h = 0 \). However, in the transient, \( h \neq 0 \) so the input \( u \) is not the one that achieves \( f = f^* \). This nonzero difference can be seen as a perturbation in the input \( u \) of system (4). Nevertheless, since the system is ISS, the stability remains despite the perturbation [37]. Due to the stability of \( h \), the perturbation vanishes with time, and the expanded system in (11) makes (4) to converge to (5).

The ISS property is common but particular to some systems. The next case of study aims at using Implicit Control for more general dynamics. First, we restrict the possible desired closed-loop behaviors to

\[
f^*(x) = -K_f(x)x \tag{13}
\]

where \( K_f(x) \) is a function that returns a positive definite matrix. In particular, \( K_f(x) \) can be a polynomial whose terms are all of even order. For instance, if \( K_f(x) = K_f \), then \( f^*(x) = -K_f x \) has a linear behavior with its corresponding exponential-like transient, desired settling time and stability properties. Similarly, let

\[
h^*(x, u) = -K_h(h(x, u))h(x, u) \tag{14}
\]

with \( K_h(h(x, u)) \) a function that returns a positive definite matrix as well.

**Theorem 3:** Given Assumption 1, the definition of \( f^* \) in (13) and the definition of \( h^* \) in (14), if matrices \( K_f(x) \) and \( K_h(h(x, u)) \) are chosen in such a way that

\[
K = \begin{pmatrix} -K_f(x) & 0.5I \\ 0.5I & -K_h(h(x, u)) \end{pmatrix} \tag{15}
\]

is negative definite, then the system in (11) is globally asymptotically stable (GAS).

**Proof:** Consider \( V = \frac{1}{2}x^T x + \frac{1}{2}h^T h \) as a Lyapunov candidate function, whose derivative is

\[
\dot{V} = x^T \dot{x} + h^T \dot{h} = x^T f + h^T h^* \tag{16}
\]

and where \( \dot{h} \) is substituted by (9) following the same steps of the proof of Theorem 2. Using the definition of \( h \) in (6) yields to

\[
\dot{V} = x^T h + x^T f^* + h^T h^* \tag{17}
\]

Considering \( f^* \) and \( h^* \) from (13) and (14), \( \dot{V} \) is

\[
\dot{V} = \left( \begin{pmatrix} x^T & h^T \end{pmatrix} K \begin{pmatrix} x^T & h^T \end{pmatrix}^T \right)^{12}
\]

Then, designing \( K_f(x) \) and \( K_h(h(x, u)) \) such that \( K \) is negative definite guarantees that (11) is GAS, and both \( h \) and \( x \) go to zero.

In practice, \( ||K_h|| \gg ||K_f|| \) imposes the convergence of \( h \) to be much faster than the desired closed-loop dynamics, so the evaders will behave following \( f^* \).

**C. Control Extensions**

Assuming \( n = m \) is a restrictive condition. We deal with the generalization to \( n \neq m \) by replacing \( J_u^{-1} \) with the pseudoinverse \( J_u^+ \). In the case that there are some states whose dynamics are not explicitly dependent on \( u \) (for instance, the velocity of the evaders might not depend on herders’ position, but their
acceleration), then a linear transformation $T$ over $h$ can be applied

$$\tilde{h}(\tilde{x}, u) = Th(x, u).$$ \tag{19}$$

The purpose is to create a new group of states $\tilde{x}$ combining all the states that do not depend on the input with some of the states that do depend on the input. Then, Implicit Control can be applied, over the new function $\tilde{h}$, to control states that do not explicitly depend on the input.

Furthermore, all the previous results can be extended to dynamic references. Let consider the desired state reference $x^*$ with desired dynamics $\dot{x}^*$.

**Corollary 1:** Assuming that $\dot{x}^*$ is smooth and bounded, the results in Theorems 2–3 hold if the function $h$ is redefined as

$$h(x, u) = f(x, u) - f^*(x) - \dot{x}^*. \tag{20}$$

**Proof:** By assuming that $\dot{x}^*$ is smooth and bounded, we maintain the existence and smoothness of the input. The original system is still described by (4) but the control objective changes to

$$f^*(x, u) = f(x, u) - \dot{x}^*. \tag{21}$$

The expression comes from imposing that the error dynamics follow $\dot{x}^*$. This information can be incorporated in the design of the function $h$ by adding the known time-varying term in (21) to compensate the dynamic reference. The time derivative of $\dot{x}^*$ is implicitly included in the Jacobians of the new $h$; therefore, the structure in system (11) remains. Moreover, since $\dot{x}^*$ does not depend on $u$, the Jacobians in Theorems 2–3 do not change and the statements in Theorems 2–3 hold.

The importance of Corollary 1 is that our herding solution can be seen as a generalization of previous herding works [4], [20] where the purpose is to drive the evaders to a certain region. By imposing a time-varying reference, herders not only drive evaders toward particular positions but also steer them toward a particular region while maintaining a specific formation. Besides, note that no particular restrictions regarding the initial configuration of the entities is specified. Nevertheless, it is convenient that the herders surround the evaders to avoid escapes before the beginning of the herding, so in Section VI-B, we develop a secure caging stage.

From an algorithmic point of view, the calculation of the control input is very simple. At each instant, the controller receives $x$ and $u$ from an observer and/or from measurements. Then, $f^*(x)$ is computed, which, together with the dynamic model of the evaders $f(x, u)$ [e.g., (2) or (3)], allows us to compute $\dot{h}^*(x, u)$ with (14). Besides, the Jacobians $J_x$ and $J_u$ can be calculated either analytically or numerically, depending on their complexity. Finally, $\dot{u}$ is computed from (11). Regarding the estimation of $x$ and $u$, another advantage of Implicit Control is that $u$ now belongs to the state of an expanded system. Therefore, it is possible to apply the same estimation techniques as with $x$, robustifying the herding solution.

V. ADDING ADAPTATION

In the previous section, perfect knowledge of the evaders’ dynamics is assumed. However, in real problems, there exist sources of uncertainty. We derive a general adaptation law that overcomes this issue and preserves the control properties.

Consider the parameters $\theta$ in (2) and (3). If we define $\tilde{\theta} = \text{diag}(\theta_1, \ldots, \theta_m) \in \mathbb{R}^{m \times m}$, then the original system (4) is rewritten to the following, with a little abuse of notation for the sake of readability

$$\dot{x} = f_{\theta}(x, u, \theta) = \theta f(x, u). \tag{22}$$

We assume a linear dependence with respect to the uncertain parameters and that the parameters are constant. This assumption is typical in adaptive control theory and, in the case of Implicit Control, implies that the gradients with respect to the parameters do not depend on the parameters, which is the key that yields to the proposed adaptation law. Nevertheless, despite simple, linear parameters can still model the aggressiveness in the repulsion provoked by the herders. Besides, notice that $\theta_j > 0$ because otherwise the evaders do not move or are not repelled by the herders.

We represent the uncertainty in $\theta$ by the vector of estimates $\hat{\theta}$, defining the error between the real and estimated parameters as $\tilde{\theta} = \theta - \hat{\theta}$. From now on, the symbols $\hat{\cdot}$ and $\cdot$ denote the quantities that depend on $\hat{\theta}$ and $\theta$. In general, $\theta \neq 0$ so Implicit Control no longer guarantees stability. The difference between actual and estimated dynamics can be defined by a new function

$$\tilde{h}(x, u, \hat{\theta}) = h(x, u, \theta) - \tilde{h}(x, u, \theta) = \theta f(x, u). \tag{23}$$

This is a standard adaptive control problem, for which there exist different solutions for both linear and nonlinear systems [37]. Nevertheless, we propose a different adaptation law, following the principles of Implicit Control, that simplifies the stability analysis.

The objective of the adaptation law is to design $\dot{\hat{\theta}}$ such that $\tilde{h}$ converges to zero while the closed-loop behavior of the system is preserved. The structure of the problem, and, in particular, of $\tilde{h}$, is similar to that of Implicit Control. We can follow the same input design procedure of Implicit Control, in this case over $\tilde{h}$, to design an adaptation law of the form

$$\dot{\hat{\theta}} = -\tilde{J}_\theta^{-1}(\tilde{h}^* + \tilde{J}_x \hat{x} + \tilde{J}_u \hat{u}). \tag{24}$$

Here

$$\tilde{J}_x = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_m} \end{pmatrix}, \quad \tilde{J}_u = \begin{pmatrix} \frac{\partial h_1}{\partial u_1} & \cdots & \frac{\partial h_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial u_1} & \cdots & \frac{\partial h_m}{\partial u_m} \end{pmatrix} \tag{25}$$

$\tilde{J}_\theta = \text{diag}(f)$, and $\tilde{h}^*$ is a free-design function.

As in the design of the input dynamics, $\tilde{h}^*$ determines the desired dynamic behavior for the derivative of (23). Concatenating
the new set of equations yields a new expanded system

$$\begin{align*}
\dot{x} &= \theta f \\
\dot{u} &= J_u^\dagger \left( \hat{h}^* - \dot{J}_x \dot{x} \right) \\
\dot{\theta} &= -J_{\dot{\theta}}^\dagger \left( \hat{h}^* + \dot{J}_x \dot{x} + \ddot{J}_u \right)
\end{align*}$$

(26)

Acknowledging the similarities between (26) and (11), we extend the results in Theorems 2 and 3.

**Proposition 1:** If the system defined in (4) is ISS and Assumption 1 holds, then the adaptive law in (24) preserves the properties of the Implicit Control if

$$\hat{h}^* = h_0^* - h + \dot{J}_x (\ddot{x} - \dot{x})$$

(27)

where $h_0^*$ is a free-design function chosen to make $\hat{h}$ stable.

**Proof:** The proof follows from Theorem 3 by considering the chain rule expansion of $\dot{h}$ in (26), we get

$$\dot{h} = \frac{\partial \hat{h}}{\partial \dot{x}} \ddot{x} + \frac{\partial \hat{h}}{\partial \dot{u}} \ddot{u} + \frac{\partial \hat{h}}{\partial \theta} \ddot{\theta}.$$ 

(28)

Since $\theta$ is constant, $\ddot{\theta} = 0$, and the substitution of (24) gives

$$\dot{h} = \dot{J}_x \ddot{x} - \dot{J}_x \dot{x} + \dot{J}_u \ddot{u} + J_x \dddot{x} + J_u \dddot{u}.$$ 

(29)

Note that $J_x \dddot{x} + J_u \dddot{u}$ is equal to $\ddot{h}$, and the $J_u \dddot{u}$ terms cancel out. Therefore, (29) can be written as

$$\frac{d\hat{h}}{dt} = \ddot{h} - \dddot{h} - J_x \dddot{x} = h_0^*$$

(30)

where the later follows from the definition of $\dddot{h}$ in (27). Now, let consider the chain rule expansion of $d\hat{h}/dt$

$$\frac{d\hat{h}}{dt} = \frac{\partial \hat{h}}{\partial \dot{x}} \ddot{x} + \frac{\partial \hat{h}}{\partial \dot{u}} \ddot{u} + \frac{\partial \hat{h}}{\partial \theta} \ddot{\theta}.$$ 

(31)

If we use the definitions of $u$ and $\dot{\theta}$ in (26), (31) reads

$$\frac{d\hat{h}}{dt} = \dot{J}_x \ddot{x} + \ddot{h} - \dot{J}_x \dot{x} - \dddot{h} - J_x \dddot{x} = J_x \dddot{x} + J_u \dddot{u}.$$ 

(32)

By exploiting $\dddot{h}$ in (27), we get

$$\frac{d\hat{h}}{dt} = \ddot{h} - h_0^*.$$ 

(33)

Finally, let consider the chain rule expansion of $d\hat{h}/dt$

$$\frac{d\hat{h}}{dt} = \frac{\partial \hat{h}}{\partial \dot{x}} \ddot{x} + \frac{\partial \hat{h}}{\partial \dot{u}} \ddot{u} + \frac{\partial \hat{h}}{\partial \theta} \ddot{\theta} = \dot{J}_x \ddot{x} + J_u \dddot{u}.$$ 

(34)

The substitution of $\dddot{u}$ in (26) gives

$$\frac{d\hat{h}}{dt} = \dot{J}_x \ddot{x} + J_u \dot{J}_u^\dagger \left( \hat{h}^* - \dot{J}_x \ddot{x} \right) = J_x (\dddot{x} - \dot{x}) + \dddot{h} - J_x \dddot{x} = \dddot{h}.$$ 

(35)

With (30), (33), and (35), we use the same arguments in the proof of Theorem 2. Let $h \neq 0$ be considered as a perturbation. Due to (30), $h$ converges to zero at some point. In the meantime, since the system is ISS, the stability remains. Besides, $\dddot{h}$ is stable, so the perturbation vanishes with time. Then, according to (33), $\dddot{h} = \dddot{h}$ and, according to (35), $\dddot{h} = h^*$. Therefore, we recover the expanded system in (11) and the results of Theorem 2 hold.

**Proposition 2:** Given Assumption 1

$$h_0^*(x, u) = -K_\theta(x, u) \hat{h}$$

(36)

with $K_\theta(x, u)$ a function that returns a positive definite matrix, and $K(h(x, u))$ defined as in (15), if

$$\bar{K} = \begin{pmatrix}
-K_f(x) & 0.5I \\
0.5I & -K_h(h(x, u)) - 0.5J_x K_\theta(x, u)
\end{pmatrix}$$

(37)

is negative definite, then the adaptive law preserves the stability properties of Theorem 3.

**Proof:** The proof follows from Theorem 3 by considering

$$V = \frac{1}{2} x^T x + \frac{1}{2} h^T \dot{h} + \frac{1}{2} \dot{h}^T \dddot{h}$$

(38)

as a Lyapunov candidate function. Developing its derivative and using (30) and (35) from Proposition 1, we obtain

$$\dot{V} = x^T \dot{h} + x^T f^* + h^T \ddot{h} + \ddot{h}^T \hat{h}$$

$$= x^T h - x^T K_f x - h^T J_x K_\theta \dot{h} - h^T J_h \hat{h} - h^T \dot{h} \hat{h} K_\theta \dot{h}.$$ 

(39)

As in Theorem 3, the latter reads

$$\dot{V} = \left( x^T \dot{h} T \dddot{h} T \right) \bar{K} \left( x^T \dot{h} T \dddot{h} T \right) T.$$ 

(40)

Then, if $K, K_\theta$ are designed such that $\dot{V}$ is negative definite, then the adaptation law preserves the stability properties of Theorem 3.

The adaptation law, then, evolves the estimates toward the values that ensure the correct behavior of the control. Remarkably, the convergence of $h$ to zero implies that $\dot{h}$ converges to $h$. Therefore, the convergence of $h$ to zero implies the convergence of $\dot{h}$ to zero, and the evaders behave as desired. In practice, to implement the controller, a slight change in (26) is required to compute $\dot{h}^*$. Since $\dddot{h} = 0$ is not available, an approximation (e.g., by finite differences) must be used to obtain $\dddot{x}$. Note that this only affects the implementation, as long as the stability analysis involves the real dynamics of the evaders.

**Example:** We provide a brief description of how to design the control and adaptation gains.

The first step is to set the desired behavior of the herd. Theorem 3 and Proposition 2 allow for several possible choices, but to keep it simple, consider the desired settling time of $1/\tau = 12s$ and an exponential transient according to $m$ first-order independent systems. Thus

$$f^*(x) = -K_f(x) x = -K_f x = -3(\tau 1)x = -0.25 I x.$$ 

(41)

Then, according to Theorem 3, $K_h(h(x, u))$ in (15) must be such that $K_\theta$ is negative definite. To preserve simplicity, let $K_h(h(x, u)) = K_h$ be a constant matrix. Under these conditions, for example, $K_h = 50 I$ works.

In the same way, if an adaptation law is desired, then $K_\theta(x, u)$ in (37) must be such that $K_\theta$ is negative definite. In the herding case, the speed of evaders and herders is bounded, so $J_x$ is bounded and there always exist a matrix $K_\theta(x, u)$ that makes $K_\theta$ negative definite. To preserve simplicity again, let $K_\theta(x, u) =$
be a constant matrix. For instance, \( K_\theta = 200I \) works in the examples of Section VII. In general, starting from \( K_f \), choosing the other gains such that \( ||K_f|| < ||K_h|| < ||K_\theta|| \) will make the system stable. Nevertheless, there are no particular restrictions on the gains, providing a lot of design possibilities. If needed/desired, \( K_f \) might not be a constant, nor \( f^* \) a linear function of the state. Similarly, \( K_h \) can change with time to preserve the negative definiteness of \( K \), and the same applies to \( K_\theta \).

VI. HERDING SOLUTION

Implicit Control, along with the adaptation law, constitutes the core of our proposal. Nonetheless, to have a complete herding solution, additional issues are considered.

A. Distributed Estimator

Up to now, the solution assumes that herders know the state and input of the system perfectly, i.e., the position of all the herders and evaders. This section describes a distributed estimator that fulfills this assumption. Particularly, we develop an E-DKF [9], tailored to work with the Implicit Control expanded dynamics. The objective is, for all the herders, to have an accurate estimation of evaders’ and herders’ position, aggregated in the vector \( \xi = [x^T_i \ldots x^T_m \; u^T_1 \ldots u^T_n]^T \). Notice that, in contrast with typical DKFs, the control inputs must be included as quantities to be estimated. In typical control solutions, this is not possible because the input has no dynamics because it is just an algebraic expression.

Implicit Control addresses this issue by providing the input dynamics. In particular, for the prediction step of the filter, we consider

\[
\dot{\xi} = \frac{\partial \Sigma(\xi)}{\partial \xi} \bigg|_{\xi = \xi(t)} + w
\]

where \( \Sigma(\xi) \) denotes the expanded system in (11) and \( w \sim N(0, Q) \) is a zero-mean Gaussian noise with covariance \( Q \).

Note that the linearization in (42) is necessary because the overall expanded dynamics are strongly nonlinear, so we use \( \frac{\partial \Sigma(\xi)}{\partial \xi} \) evaluated at \( \xi = \xi(t) \). Since the evaders’ dynamics are deterministic, \( Q \) will have small values, useful to mitigate the effects of the uncertainty during the adaptation law’s transient.

For the filter update, it is important to note that the herders’ measurements will depend on its position, i.e., a herder will only be able to measure the entities that are close to it. Therefore, the measurement equation is of the form

\[
z_i = H_i(\xi)\xi + v_i
\]

where \( z_i \) is the measurement taken by herder \( i \), \( v_i \sim N(0, R_i) \) is the measurement noise with covariance \( R_i \), and \( H_i \) depends on \( \xi \). This dependence makes the dimension of \( z_i \) to change at each filter iteration, complicating the implementation.

The solution lies in using an information form of the E-DKF so, at each instant, each herder has a vector of measurements and a vector of booleans. The latter says whether an entity has been sensed or not, whereas the former contains the actual measurements and zeroes for the rest of entities, i.e., \( z_i \in \mathbb{R}^{2m+2n} \).

Given the two vectors, the information form of the E-DKF permits to add up all the measurements in common state coordinates, only adding those measurements where the boolean vector is equal to one. Besides, communication among herders happens only if the distance between two herders is less than \( d_c \). Thus, the estimator is input-dependent in the topology as well.

It is noteworthy that the stochastic nature of the estimator can absorb discrepancies from the adaptation and the unmodeled perturbations. This, together with the information form of the filter, permits to run the control and the estimation at different frequencies. The control loop can be the fastest while the estimation runs slower to account for communication bandwidth issues. Regarding the influence of the estimator in the overall stability of the system, if the estimator is designed to converge sufficiently fast, from a practical point of view, its influence in the controller can be overlooked, and the controller works with accurate states and inputs. Regarding global awareness, the estimator still assumes a common reference frame for all the agents (herders and evaders) as well as knowledge of their number. Nevertheless, there are works available that deal with the distributed computation of these quantities [38], [39].

B. Caging the Evaders

To avoid escapes, herders must be close to the evaders before beginning the precision herding using Implicit Control. This is not a complete solution because robots might not be deployed in the proximity of the evaders. Therefore, we propose a caging stage to prevent these escapes.

We consider a caging strategy based on the mean position and covariance of the herd. We denote them by \( \bar{x}_\text{avg} := \frac{1}{m} \sum_j x_j \) and \( P\text{\_avg} := \frac{1}{m} \sum_j (x_j - \bar{x}_\text{avg})(x_j - \bar{x}_\text{avg})^T \). The combination of both gives a description of the herd in terms of an ellipsoid centered in \( \bar{x}_\text{avg} \). The geometry of the setup for a 2-D herding is depicted in Fig. 2. It can be observed that the ellipse (in red) delimits a region around the evaders (in green), which suggests a uniform distribution of the herders along the perimeter. However, these locations might be too close to the evaders. Therefore, it is convenient to use an augmented version of the
ellipse, \( \mu_1 \mathbf{P}_{\text{avg}} \) with \( \mu_1 > 1 \), to place the herders sufficiently far from the evaders. We do not make any assumptions about the initial distribution of the evaders. For instance, a very spread distribution of evaders with a large \( \mathbf{P}_{\text{avg}} \) will only imply that it will take more time for the herders to surround the evaders. Moreover, \( \mu_1 \) can be tuned so that herders provoke weak repulsive forces in the evaders. During the caging, evaders will not escape, and afterward, the herders will approach the evaders to provoke large enough repulsive forces to steer the herd. Anyway, this is transparent for both the caging and the precise herding stage.

To solve the caging, we propose a controller, adapted from [40], that steers the herders toward an even distribution along the perimeter of \( \mu_1 \mathbf{P}_{\text{avg}} \). The position of herder \( i \), in polar coordinates with respect to the ellipse, is driven by

\[
\ddot{\psi}_i = (\psi_{i+1} - \psi_i) + (\psi_{i-1} - \psi_i) \tag{44}
\]

\[
\dot{\rho}_i = \rho_i - \rho_i^* \tag{45}
\]

where \( \psi_i \) and \( \rho_i \) are the angular and radial positions, respectively, and \( i + 1, \ i - 1 \) denote the left and right closest herders, respectively. The policy is inspired in the well known \( N \)-bugs problem, which evenly distributes \( N \) agents over a closed curve. An even distribution of herders along the perimeter of an ellipse requires the herders to be distributed with a uniform angle. The agreement in the angle is given by (44). Meanwhile, to each angle corresponds a certain radial position to place herders over the desired ellipse, given by \( \rho_i^* \). Therefore, the controller of the radial position is just the linear feedback controller in (45). For formal guarantees of convergence, we refer to the work in [40].

Herder \( i \) only needs to measure the closest left and right neighbor. These commands are then translated to Cartesian coordinates. Under this control policy, the herders arrive to equally-distributed positions along the perimeter of the ellipse, denoted by \( \mathbf{u}^*_i \) and depicted in blue in Fig. 2.

Due to perturbations or unmodeled phenomena, the caging might not prevent herders to move too close to the evaders. To solve this issue, we propose the use of CBFs [41]. The controller in (44)–(45) computes the velocity \( \dot{\mathbf{u}}_{\text{nom}}^i \), which is then translated into Cartesian coordinates. This is called the nominal action. The nominal action might move the herder toward a region where the interaction with the evaders is not convenient. We define this region by \( \mu_2 \mathbf{P}_{\text{avg}} \), where \( \mu_1 > \mu_2 > 1 \), and it is depicted in purple in Fig. 2.

To correct the nominal action, we use a quadratic program based on CBFs. In particular, at each instant, the closest point in the perimeter of the ellipse \( \mu_2 \mathbf{P}_{\text{avg}} \) to herder \( i \) is computed, denoted by \( \mathbf{e}_i \). Then, an optimization process [41] is carried out

\[
\hat{\mathbf{u}}_{i}^{\text{cbf}} = \arg \min_{\dot{\mathbf{u}}_{i}^{\text{cbf}}} \| \dot{\mathbf{u}}_{i}^{\text{cbf}} - \dot{\mathbf{u}}_{i}^{\text{nom}} \|^2 \tag{46a}
\]

s.t. \( L_f h(\Delta_i) + L_q h(\Delta_i) \hat{\mathbf{u}}_{i}^{\text{cbf}} \geq -\kappa(h(\Delta_i)). \tag{46b} \)

Here, \( \Delta_i = \mathbf{u}_i - \mathbf{e}_i \), depicted in orange in Fig. 2. Moreover, \( \hat{\mathbf{u}}_{i}^{\text{cbf}} \) is the action to be optimized, \( h(\Delta_i) \) is the control barrier function (CBF), \( \kappa(h(\Delta_i)) \) is a class \( \mathcal{K} \) function and \( L_f, L_q \) denote the Lie derivatives. It is important to remark that the abuse of notation with function \( h \) is to fit the typical notation of CBFs [41].

Finally, a high-level algorithm of the complete herding sequence is described in Algorithm 1.

### VII. SIMULATION RESULTS

The purpose of these simulations is to validate and demonstrate the success of the proposal against challenging situations using the models in Section III. The herding solution is formed by different parts, so the simulations will be conducted progressively to analyze each individual component.

#### A. Implicit Control and Adaptation Law

The first case of study consists in the herding of five evaders by five herders. The herders receive perfect measurements and they are already deployed near the evaders. The details are in Table II. We set \( \mathbf{K}_f = 0.25 \mathbf{I}_{3 \times 3} \), yielding a settling time of 12 s with an exponential transient according to 2 \( m \) first-order independent systems. To ensure that the conditions of Theorem 3 and Proposition 2 hold, we set \( \mathbf{K}_h = 50 \mathbf{I}_{2 \times 2} \) and \( \mathbf{K}_{\theta} = 200 \mathbf{I}_{2 \times 2} \). For the sake of comparison, we use the Levenberg–Marquardt numerical method [43] as a baseline to compute the control input.

The first row of Fig. 3 shows the trajectories followed by herders and evaders for the different test cases. Implicit Control is able to herd the evaders successfully, achieving almost the
same trajectories obtained by the numerical baseline in all the experiments. Conversely, the trajectories of the herders present some differences depending on the motion model, which highlights the complexity of the control problem at hand, greater for the exponential model due to the switching dynamics. The second row of Fig. 3 represents the evolution over time of the evaders’ position error. Implicit Control achieves the desired settling time of 12 s and the state evolves as an exponential function according to the imposed closed-loop dynamics. Interestingly, it is corroborated that the Implicit Control solution does not assign any particular evader to any robotic herder, so herders just move to steer all the evaders toward their particular assigned positions simultaneously. All herders are contributing at all instants to the motion of all evaders. But, to achieve a stable equilibrium, a team of herders is needed because otherwise the herding cannot be simultaneous, as it is seen in the second row of Fig. 3. At equilibrium, each herder is compensating the repulsive forces provoked by the other herders, reason why the evaders end up in the desired positions with zero velocity.

The only discrepancy between Implicit Control and the numerical baseline is in the first instants of the transient. The numerical baseline applies the “correct input” from the beginning, because at each instant, it iterates until it finds the input that solves $f(x, u) - f^*(x) = 0$. Meanwhile, Implicit Control requires some time to converge to $f(x, u) - f^*(x) = 0$, which is when the “correct input” is applied. Therefore, what only differs between methods is the shape of the transient in the first instants. Then, the total settling time with the Implicit Control is at most the settling time of $f^*(x)$ plus the time $h(x, u)$ takes to converge to zero. If $K_h$ is large enough, then the latter is negligible. This is why the differences in the transient are minimal in the second row of Fig. 3. Meanwhile, the differences in the input are much greater, as it is seen in Fig. 4. The controlled system is strongly input-nonaffine, so there are many roots of $h$ that solve $h(x, u) = 0$. After Implicit Control converges to $h(x, u) = 0$, herders are in a different configuration with respect to the herders driven by the numerical baseline, and this difference implies different roots for $h(x, u)$ until the evaders are close to the desired configuration.

Fig. 5 shows how the adaptation law adjusts the parameters to ensure stability. Initially, all the parameters are equal; however, they evolve differently because each evader is in a specific situation with respect to its desired position and the herders. At the steady state, the estimated parameters are equal to the real ones.

B. Effects of the E-DKF in the Herding

The first series of simulations has validated the success of Implicit Control and the adaptation law. Next, we show how the herding works when perfect state and input knowledge is replaced by the distributed estimator. The communication and sensing thresholds are in Table II. We set $Q = 0.02I_{2m+2n}$, and $R_i = 0.07I_{2m+2n}$ $\forall i$ to simulate a scenario with bad sensing
because the movement of herders and evaders is deterministic, so $Q$ just models the noise associated with the difference between real and estimated parameters during the adaptation.

C. Complete Precise Herding

The flexibility and generality of the solution can be extended to heterogeneous groups and time-varying references, resulting in a more realistic herding. In this example, we evaluate the inclusion of the caging stage as considered in Algorithm 1. Fig. 7 shows how three herders herd a group of three evaders using the complete herding solution in Algorithm 1. The red evader is exponential, whereas the purple evaders are inverse. The CBFs have been tuned as follows, choosing

$$h(\Delta_t) = ||\Delta_t|| + T \Delta_t \delta_{t-1} \delta_v^{\text{nom}} - \varphi$$

and

$$\kappa(h(\Delta_t)) = k_{\text{cdef}} h(\Delta_t)^3$$

where $T$ is the sampling time, $\varphi = 3$ is the desired distance with the perimeter of $\mu_2 P_{\text{avg}}$ ($\mu_1 = 7$ and $\mu_2 = 4$) and $k_{\text{cdef}} = 50$ is a gain that modulates how strong is the repulsion provoked by the proximity with $e_i$. We recall that in the caging stage, the dynamics of the herders are

$$u_i = u_i + T \delta u_i^{\text{cbf}}.$$  

Meanwhile, the desired herding configuration evolves with

$$\dot{x}_j^* = v_j^*, \quad \dot{y}_j^* = 0.5 w_j^* \cos(w_j^* t + 2\pi/j)$$

where $w^* = [0.05, 0.1, 0.02] \text{rad/s}$ and $v^* = [0.05, 0.05, 0.05] \text{m/s}$. Due to the distances involved in this herding instance, we set $d_m = d_c = 15 \text{m}$.

Initially, the herders move to uniformly surround the evaders, positioning themselves on the perimeter of the desired ellipse. Once the herders are there, the precise herding begins, driving the evaders to their sinusoidal references. This yields to herders’ trajectories surrounding and modulating the interaction forces with the evaders. With the evaders in their desired trajectories, the system reaches a steady-state behavior where the periodic movement of the evaders is shared by the herders. The clip of this simulation is included in the supplementary material.

D. Controlling Large Herds With a Few Herders

To demonstrate the generality and flexibility of Implicit Control, we conduct some experiments where Implicit Control is used to control a large number of evaders with a few herders. To do this, the control goal is to steer the centroid of the herd toward a particular location. At each instant, the control algorithm first calculates the centroid of the herd and, then, it computes the input with Implicit Control, using the centroid as if it was a virtual evader with its corresponding evader’s dynamics. The computation of the centroid induces a little noise in the dynamics of the virtual evader, but it is negligible because, pointwise, the centroid’s dynamics is still that of the virtual evader. Besides, this discrepancy can be absorbed by the adaptation.

TABLE III

| R [m$^2$] | Steady-state error $||x - \bar{x}^*||$ [m] |
|-----------------|-----------------|
| Inverse Model, 5s×5 | Exponential Model, 5s×5 |
| $0.074_{20}$ | $0.02$ | $0.01$ |
| $0.154_{20}$ | $0.06$ | $0.06$ |
| $0.303_{20}$ | $1.34$ | $0.12$ |

In this vein, it is interesting to analyze how increasing the noise covariance affects the performance. To do so, we have run the same simulations in Fig. 6 but changed the measurement covariance matrix, checking to what extent the noise can increase without affecting the performance. The results are in Table III. Implicit Control is robust against measurement noise, achieving good performance with noises up to 0.3 m$^{-2}$. We underline that this implies measurement errors in the order of 0.5 m, which is significant considering that the herding takes place in a 5 × 5 m square (i.e., a 10% of the side size of the arena). It is also observed that the tolerance to noise depends on the evaders’ dynamics. On the other hand, we do not consider changes in $Q$ to illustrate how incorporating Implicit Control in the prediction stage is useful. Finally, to be realistic in the use of the communication bandwidth, messages are exchanged every 100 ms, i.e., the communication process runs 10 times slower than the control. Fig. 6 depicts the performance of the herders against the same herding scenario of Fig. 3. The results are almost identical, with herders and evaders following similar trajectories as if there was perfect feedback. The reason for that is answered in the second row of Fig. 6. The root-mean-square error (RMSE) in the estimation of evaders’ (blue) and herders’ (red) position, averaged over the five herders.

Fig. 6. Simulation results of the herding of five evaders by five herders using Implicit Control with adaptation law and the distributed estimator. The first row shows the trajectories followed by the herders (same symbols of Table I). The second row depicts the RMSE in the estimation of evaders’ (blue) and herders’ (red) position, averaged over the five herders.

TABLE III

Impact of the Measurement Noise in the Herding

| R [m$^2$] | Steady-state error $||x - \bar{x}^*||$ [m] |
|-----------------|-----------------|
| Inverse Model, 5s×5 | Exponential Model, 5s×5 |
| $0.074_{20}$ | $0.02$ | $0.01$ |
| $0.154_{20}$ | $0.06$ | $0.06$ |
| $0.303_{20}$ | $1.34$ | $0.12$ |
In Fig. 8, it is shown how 5 herders can control 50 evaders. There are 10 inputs and 2 states to control (the x and y positions of the centroid). To maintain the cohesion of the herd, but also to avoid collisions among the evaders, we have added very weak repulsive and coalition forces among evaders

\[ \dot{x}_j = f^{\text{inv/exp}}_j(x, u) + \vartheta \sum_{j' = 1}^{m} d_{jj'} \left( \frac{1}{||d_{jj'}||^3} - ||d_{jj'}||^2 \right) \]  

(50)

with \( d_{jj'} = x_j - x_{j'} \) and \( \vartheta = 2 \times 10^{-4} \), i.e., 4 orders of magnitude smaller than the dynamics in (2) and (3).

The first row of Fig. 8 shows how the five herders can steer the whole herd toward a certain region. As a curiosity, since the number of available control inputs is much greater than the number of states to control, there is a “leftover” herder in steady-state (right panel): There are two herders that share a similar position. More interestingly, the second row of Fig. 8 shows how the herders can split the herd into two subherds and steer each of them, simultaneously, toward individually assigned locations. This confirms that, with our herding, it is possible to consider each evader to virtually represent a herd (see, e.g., [4] for a similar consideration).

The supplementary material includes a video with different configurations of evaders, demonstrating that Implicit Control can herd large heterogeneous groups of evaders while mixing inverse and exponential evaders.

VIII. EXPERIMENTS

In this section, we extend the experiments to the real framework provided by the Robotarium [44]. To do so, some robots play the role of herders, whereas the others act as evaders, following the dynamics in Section III. The robots are GRITSBot X moving on a 3.2 m × 2 m area, coordinated by a central server that receives odometry and sends velocity commands at an approximate delay of 0.033 s. Thus, a low-level controller is used to translate the control output into velocity commands, with \( v_{\text{max}} = 0.2 \text{ m/s} \). We use the experiments to validate Implicit Control, the adaptation law, and the caging stage. Finally, we adjust some parameters to fit the conditions of the experiment: \( T = 0.033 \text{s}, \theta = 0.02, \hat{\theta}(0) = 0.015 \) (inverse), \( \theta = 0.05, \hat{\theta}(0) = 0.04 \) (exponential), \( \sigma = 1.2 \).

The experiment in Fig. 1 exhibits a similar behavior to the simulations in Section VII. The evaders try to evade the herders, going in the direction of lower density of herders. To tackle this, the closest herders surround the evaders to align with the other herders, which move away to modulate the interaction forces. These conclusions are reaffirmed by the experiments in...
III. EXPERIMENTAL RESULTS

Fig. 9, where different combinations of evaders, in their number and dynamics, are tested. Additionally, the herders successfully herd heterogeneous groups of evaders, as it is demonstrated in Fig. 9(c) and (d) and in the time-varying experiment in Fig. 10.

The latter consists of the herding of an inverse evader and an exponential evader by three herders, where the caging stage has been included. Initially, the herders approach the evaders, surrounding them to avoid escapes. Then, they steer the evaders toward the desired trajectories (in red). Despite the space limitations and the complex nonlinear repulsive dynamics, the evaders successfully follow the references.

As a detail, in the video, it is observed that, in the case of four evaders versus four herders, the two herders on the left side of the frame exchange their positions at a certain instant. This is because the control input is defined in terms of its dynamics, so it “follows the flow of a numerical solver.” Therefore, herders may exchange their position because this is how the input dynamics is evolving toward the desired dynamics. To avoid this behavior in a particular application, a low-level controller with an obstacle-avoidance mechanism can be incorporated. Anyway, this is not a concern of Implicit Control but of the particular implementation in the application.

All the experiments are successful despite uncertainty due to the adaptation law. The supplementary video includes the complete clips of all the experiments.

IV. CONCLUSION

This article has addressed a novel control strategy to solve the herding problem in MRS. This strategy, based on numerical analysis theory and coined as Implicit Control, finds suitable herding inputs even when, due to the complex nonlinearities of the herd, the control law is given by a set of implicit equations. Implicit Control develops a continuous-time expansion of the system and comes with formal proofs of convergence. In addition, it is flexible in the number of evaders and general with respect to their motion model. The theoretical concepts of Implicit Control allow us to derive an adaptation law. It preserves the stability properties of the control while naturally integrates the control architecture. To complete the proposal, we have developed a novel solution for the caging problem. The herders can depart from any arbitrarily far region because they are able to approach and surround the evaders. This is done in conjunction with a novel version of an extended distributed Kalman filter, which has solved the problem of complete perfect measurements. The flexibility, robustness, and generality of the solution were demonstrated in numerous simulations and experiments with real robots.

Implicit Control paves the way for novel research lines of potential interest, including the full distribution of the control computation, e.g., using distributed optimization techniques. Another interesting direction is to add learning tools to infer online not only linear-dependent parameters but also the nonlinear structure of the herd dynamics.

REFERENCES

[1] G. Antonelli, F. Arrichiello, and S. Chiaverini, “The entrapment/escorting mission,” IEEE Robot. Automat. Mag., vol. 15, no. 1, pp. 22–29, Mar. 2008.
[2] D. Zhu, R. Lv, X. Cao, and S. X. Yang, “Multi-AUV hunting algorithm based on bio-inspired neural network in unknown environments,” Int. J. Adv. Robot. Syst., vol. 12, no. 11, pp. 166–178, 2015.
[3] S. Gao, R. Song, and Y. Li, “Cooperative control of multiple nonholonomic robots for escorting and patrolling mission based on vector field,” IEEE Access, vol. 6, pp. 41883–41891, 2018.
[4] A. Pierson and M. Schwager, “Controlling noncooperative herds with robotic herders,” IEEE Trans. Robot., vol. 34, no. 2, pp. 517–525, Apr. 2018.
[5] S. Nolfi, “Power and the limits of reactive agents,” Neurocomputing, vol. 42, no. 1–4, pp. 119–145, 2002.
[6] D. Strömbo et al., “Solving the shepherd problem: Heuristics for herding autonomous, interacting agents,” J. Roy. Soc. Interface, vol. 11, no. 100, 2014, Art. no. 20140719.
[7] N. K. Long, K. Sammut, D. Sgarro, M. Garratt, and H. A. Abbass, “A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 4, no. 4, pp. 523–537, Aug. 2020.
[8] E. Sebastián and E. Montijano, “Multi-robot implicit control of herds,” in Proc. IEEE Int. Conf. Robot. Automat., 2021, pp. 1601–1607.
[9] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in Proc. IEEE Conf. Decis. Control, 2007, pp. 5492–5498.

[10] S. L. Du, X. M. Sun, M. Cao, and W. Wang, “Pursuing an evader through cooperative relaying in multi-agent surveillance networks,” Automatica, vol. 83, pp. 245–255, 2017.

[11] A. Franchi, P. Stiegano, and G. Oriolo, “Decentralized multi-robot en- circlement of a 3D target with guaranteed collision avoidance,” Auton. Robots, vol. 40, no. 2, pp. 245–265, 2016.

[12] A. Jahn, R. J. Althappeh, D. Saldaña, L. C. A. Pimenta, A. G. Santos, and M. F. M. Campos, “Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments,” in Proc. IEEE Int. Conf. Robot. Automat., 2017, pp. 273–278.

[13] V. S. Chipade and D. Panagou, “Herding an adversarial swarm in an obstacle environment,” in Proc. IEEE Conf. Decis. Control, 2019, pp. 5859–5864.

[14] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester, “Redundant manipulator techniques for partially known path planning and control of a platoon of autonomous vehicles,” IEEE Trans. Syst., Man., Cybern., Part B (Cybern.), vol. 35, no. 4, pp. 842–848, Aug. 2005.

[15] G. Antonelli and S. Chiaverini, “Kinematic control of platoons of autonomous vehicles,” IEEE Trans. Robot., vol. 22, no. 6, pp. 1285–1292, Dec. 2006.

[16] F. Auletta, D. Fiore, M. J. Richardson, and M. di Bernardo, “Herding stochastic autonomous agents via local control rules and online target selection strategies,” Auton. Robots, 46, no. 3, pp. 469–481, 2022.

[17] R. A. Licitra, Z. I. Bell, E. A. Doucette, and W. E. Dixon, “Single agent indirect herding of multiple targets: A switched adaptive control approach,” IEEE Control Syst. Lett., vol. 2, no. 1, pp. 127–132, Jan. 2018.

[18] R. A. Licitra, Z. I. Bell, and W. E. Dixon, “Single-agent indirect herding of multiple targets with uncertain dynamics,” IEEE Trans. Robot., vol. 35, no. 4, pp. 847–860, Aug. 2019.

[19] K. Elamvazhuthi, Z. Kakish, A. Shirsat, and S. Berman, “Controllability and stabilization for herding a robotic swarm using a leader: A mean-field approach,” IEEE Trans. Robot., vol. 37, no. 2, pp. 418–432, Apr. 2021.

[20] H. Song et al., “Herding by caging: A formation-based motion planning framework for guiding mobile agents,” Auton. Robots, vol. 45, no. 5, pp. 613–631, 2021.

[21] W. Jiang, G. Wen, Z. Peng, T. Huang, and A. Rahmani, “Fully distributed formation-containment control of heterogeneous linear multiagent systems,” IEEE Trans. Autom. Control, vol. 64, no. 9, pp. 3889–3896, Sep. 2019.

[22] D. A. Anisi, P. Ögren, and X. Hu, “Cooperative minimum time surveillance with multiple ground vehicles,” IEEE Trans. Autom. Control, vol. 55, no. 12, pp. 2679–2691, Dec. 2010.

[23] M. V. Ramana and M. Kothari, “A cooperative pursuit-evasion game of a high speed evader,” in Proc. IEEE Conf. Decis. Control, 2015, pp. 2969–2974.

[24] A. Alexopoulos, B. Kirsch, and E. Badreddin, “Realization of pursuit-evasion games with unmanned aerial vehicles,” in Proc. Int. Conf. Unmanned Aircr. Syst., 2017, pp. 797–805.

[25] A. Franchi, P. Stiegano, M. D. Rocco, and G. Oriolo, “Distributed target localization and encirclement with a multi-robot system,” IFAC Proc. Vol., vol. 43, no. 16, pp. 151–156, 2010.

[26] J. Zhu, W. Zou, and Z. Zhu, “Learning evasion strategy in pursuit-evasion by deep Q-network,” in Proc. Int. Conf. Pattern Recognit., 2018, pp. 67–72.

[27] S. F. Desouky and H. M. Schwartz, “Learning in n-pursuer n-evader differential games,” in Proc. IEEE Int. Conf. Man Cybern., 2010, pp. 4069–4074.

[28] X. Peng, S. Zhang, and X. Lei, “Multi-target trapping in constrained environments using gene regulatory network-based pattern formation,” Int. J. Adv. Robotic Syst., vol. 13, pp. 5859–5864, 2016.

[29] M. B. Estrada and S. P. Manjarres, “Implicit control law for linear time varying SISO systems with bounded disturbances,” in Proc. IEEE Conf. Decis. Control, 1996, vol. 3, pp. 3347–3352.

[30] Y. Sueoka, Y. Sugimoto, M. Ishikawa, K. Osuka, and A. Ishiguro, “Analysis of implicit control structure in object clustering phenomena,” in Proc. IEEE Int. Conf. Robot. Biomimetics, 2012, pp. 2120–2125.

[31] K. Osaka, A. Ishiguro, X.-Z. Zheng, Y. Sugimoto, and D. Owaki, “Dual structure of mobilities-implicit control and explicit control,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2010, pp. 2407–2412.

[32] F. Blanchini, G. Fenu, G. Giordano, and F. A. Pellegrino, “Model-free plant tuning,” IEEE Trans. Autom. Control, vol. 62, no. 6, pp. 2623–2634, Jun. 2017.

[33] S. Shivam, I. Buckley, Y. Ward, C. Seatzu, and M. Egerstedt, “Tracking control by the Newton-Raphson flow: Applications to autonomous vehicles,” in Proc. IEEE Euc Control Conf., 2019, pp. 1562–1567.

[34] Y. Wardi, C. Seatzu, M. Egerstedt, and I. Buckley, “Performance regulation and tracking via lookahead simulation: Preliminary results and validation,” in Proc. IEEE Conf. Decis. Control, 2017, pp. 6462–6468.

[35] W. Rudin, Principles of Mathematical Analysis, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.

[36] J. M. Lee, Introduction to Smooth Manifolds. Berlin, Germany: Springer, 2003.

[37] H. K. Khalil, Nonlinear Control. New York, NY, USA: Pearson Education, 2014.

[38] M. Franceschelli and A. Gasparri, “Gossip-based centroid and common reference point estimation in multiagent systems,” IEEE Trans. Robot., vol. 30, no. 2, pp. 524–531, Apr. 2014.

[39] E. Montijano, S. Martinez, and C. Sagues, “Distributed robust consensus using RANSAC and dynamic opinions,” IEEE Trans. Control Syst. Technol., vol. 23, no. 1, pp. 150–163, Jan. 2015.

[40] E. Montijano, A. Priolo, A. Gasparri, and C. Sagues, “Distributed entrapped multi-robot systems with uncertainties,” in Proc. IEEE Conf. Decis. Control, 2013, pp. 403–408.

[41] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” IEEE Trans. Robot., vol. 33, no. 3, pp. 661–674, Jun. 2017.

[42] D. Eberly, “Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid,” Geometric Tools, LLC, 2011.

[43] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” SIAM J. Appl. Math., vol. 11, no. 2, pp. 431–441, 1963.

[44] S. Wilson et al., “The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” IEEE Control Syst. Mag., vol. 40, no. 1, pp. 26–44, Feb. 2020.