

Adaptive Multi-robot Implicit Control of Heterogeneous Herds

Eduardo Sebastián, Eduardo Montijano and Carlos Sagüés

Abstract—This paper presents a novel control strategy to herd groups of non-cooperative 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 to *implicit equations*. The intuition behind this methodology consists in driving the input, rather than computing it, towards 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 (CBFs), 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, multi-robot systems.

I. INTRODUCTION

RECENT advances in Multi-Robot 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 non-cooperative 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].

To cope with these difficulties, we present a novel control solution, called *Implicit Control*, that is capable of steering a group of evaders towards *individually assigned goals simultaneously*, irrespective of their heterogeneous and non-cooperative nonlinear dynamics. The proposal leverages numerical analysis theory to derive proofs of existence and stability of the control. Furthermore, these fundamentals are used to

E. Sebastián, E. Montijano and C. Sagüés are with the RoPeRT group, at DIIS - I3A, Universidad de Zaragoza, Spain. {esebastian, emonti, csagues}@unizar.es

This work has been supported by the ONR Global grant N62909-19-1-2027, the Spanish projects PGC2018-098817-A-I00 and PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE), DGA T45-20R, and Spanish grant FPU19-05700.

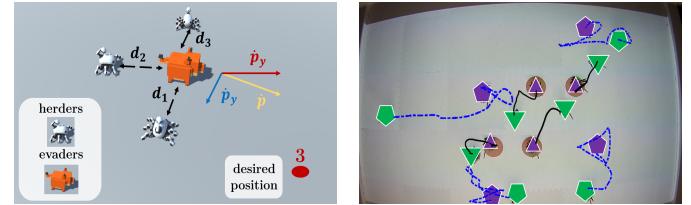


Fig. 1. In herding, the robotic herders drive the evaders towards specific locations simultaneously. The control strategy exploits repulsive forces (left) to place the evaders in their desired positions. The experiment in the right depicts four robots herding four evaders. The symbols are explained in Table I. The supplementary video includes the complete execution of the experiment.

develop a general adaptation law that tackles the herding in the presence of uncertainties in the evaders' models. With the inclusion of a distributed estimator and a caging strategy we present a complete, flexible, general herding solution. The flexibility and generality of Implicit Control also allows to steer large herds to a common region or to control different herds individually.

In more detail, the *contributions* of this work are:

- A novel control technique, called Implicit Control (Section IV), that solves a precision herding problem in the context of MRS. Implicit Control is able to impose desired dynamics for general input-nonaffine systems. This is done even if some of the parameters are unknown thanks to an adaptation law (Section V) that is derived from the principles of Implicit Control.
- A distributed estimator and a caging technique based on uncertainty models and CBFs (Section VI) that soften some requirements of the controller. The distributed estimator is a derived design from the Implicit Control: it considers, explicitly, the control proposal to enhance the estimation; besides, it accounts for partial measurements, overcoming the need of complete perfect measurements. With the caging technique the herders encircle the evaders, approaching and surrounding the evaders before they can escape. Simulations (Section VII) and experiments (Section VIII) with real robots validate the success of the proposal.

This paper 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 was 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 (i) single-agent or multi-agent herding solutions, (ii) herding of evaders with linear or nonlinear dynamics, and (iii) 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. The authors of [2] apply a Bio-inspired Neural Network to hunt in an underwater environment. To produce an escorting behavior, [10] develops 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, the authors in [13] go a step further and propose an active encirclement which avoids obstacles. With the same spirit, [14] and [15] develop solutions to drive a whole herd towards a certain region, where it is required that some global features of the herd (e.g., mean position) converge to a desired equilibrium. Besides, both works deal with self-controllable entities, i.e., the controller, state to control, and input are all embedded in the same entity. Recently, [16] exploits simple local rules to push all herders towards the same containment area. In [17], a single robot is in charge of herding multiple evaders by controlling them one by one, whereas [18] extends 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. The authors of [20] present an approach based on algebraic topology called “herding by caging”, where repulsive forces are leveraged to steer the evaders, again, towards 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 towards a global region or to split and steer each sub-herd towards 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 UGVs to ensure that a region is completely surveyed whilst in [23] both pursuers and evader 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 hex-rotor 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 assume 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, [26] includes 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. In [29], the authors 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 towards 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 paper, 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, [33] shows the advantages of this novel control perspective in applications with highly nonlinear input-affine 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 $j \in \{1, \dots, m\}$, and the herders by $i \in \{1, \dots, n\}$. Since we aim at using the position of the herders to control the position of the evaders, we define the state $\mathbf{x} \in X \subseteq \mathbb{R}^{2m}$ and the input $\mathbf{u} \in U \subseteq \mathbb{R}^{2n}$ as $\mathbf{x} = [\mathbf{x}_1^T \dots \mathbf{x}_m^T]^T$ and $\mathbf{u} = [\mathbf{u}_1^T \dots \mathbf{u}_n^T]^T$, where $\mathbf{x}_j = [x_j^e, y_j^e] \in X_j \subseteq \mathbb{R}^2$ is the position of evader j and $\mathbf{u}_i = [x_i^h, y_i^h] \in U_i \subseteq \mathbb{R}^2$ is the position of herder i , operating in a 2D space¹. The movement of each evader is described by general dynamics

$$\dot{\mathbf{x}}_j = f_j(\mathbf{x}, \mathbf{u}), \quad (1)$$

allowing for any nonlinear behavior encoded in $f_j(\mathbf{x}, \mathbf{u})$. 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 Eq. (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

$$\dot{\mathbf{x}}_j = f_j^{inv}(\mathbf{x}, \mathbf{u}) = \theta_j \sum_{i=1}^n \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|^3} \quad (2)$$

where $\mathbf{d}_{ij} = \mathbf{x}_j - \mathbf{u}_i$ is the relative position between evader j and herder i , and θ_j is a positive constant which expresses the aggressiveness in the repulsion provoked by the herders. Note that despite the model has a singularity in $\mathbf{d}_{ij} = \mathbf{0}$, in practice the speed remains bounded. The repulsion grows with $\frac{1}{\|\mathbf{d}_{ij}\|^3}$ so the closer the herders, the larger the repulsion. The only way of achieving $\mathbf{d}_{ij} = \mathbf{0}$ is that a perfectly evenly distributed number of herders approaches the evader. This does not happen in practice because herders need to impose non-zero repulsive forces to steer the evaders.

The *Exponential Model* (adapted from [17]) is

$$\begin{aligned} \dot{\mathbf{x}}_j = f_j^{exp}(\mathbf{x}, \mathbf{u}) = \\ \theta_j \sum_{i=1}^n \mathbf{d}_{ij} e^{-\chi_{ij}} (1 - \beta_j \text{sigm}(-\|\mathbf{d}_{ij}\| + d_{\min})) \end{aligned} \quad (3)$$

where $\chi_{ij} = \frac{1}{\sigma_j^2} \mathbf{d}_{ij}^T \mathbf{d}_{ij}$ and $\sigma_j > 1$. In this model there is a switching condition if $\|\mathbf{d}_{ij}\| \leq d_{\min}$, where the evader j becomes “scared” when the distance with some herder i is

¹The choice of a 2D space is to fit the real experiments, but the solution can be generalized to higher dimensions without changes in the formulation.

smaller than d_{\min} and the intensity of the repulsive interaction increases, due to $0 < \beta_j < 1$. In practice, the sigmoid function $\text{sigm}(\cdot)$ 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 saturations in the speed of the evaders.

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

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad (4)$$

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

Our goal is to herd the evaders towards the desired positions $\mathbf{x}^* \in X$ simultaneously. To do this, we define the position error of the evaders as $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*$, and we set the control objective to be to drive $\tilde{\mathbf{x}}$ to zero. This is what we call *precise herding*. A particular instance of precise herding is the herding of a herd towards the same region, where some of the elements of \mathbf{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 \mathbf{x}_j has an individually assigned \mathbf{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 which 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_{\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 of developing a caging stage to complete the herding, described in Subsection VI-B.

IV. IMPLICIT CONTROL

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

$$\dot{\mathbf{x}} = f^*(\mathbf{x}). \quad (5)$$

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 structure² 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

²The choice of f^* as a state-dependent function is to follow the typical desired dynamics in regulation problems. However, the results can be directly extended to desired dynamics that depend on the input as well.

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

Firstly, 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

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

as the working equation, shifting the analysis to that of looking for the existence of \mathbf{u} such that $h(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, 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 smoothness³ of \mathbf{u} . The Implicit Function Theorem (Theorem 9.28 of [35]), 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*

$$h : I = X \times U \subset \mathbb{R}^M \times \mathbb{R}^N \rightarrow \mathbb{R}^N \quad (7)$$

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

$$\mathbf{J} = (\mathbf{J}_{\mathbf{x}} | \mathbf{J}_{\mathbf{u}}) = \begin{pmatrix} \frac{\partial h_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial h_1}{\partial \mathbf{x}_M} & \frac{\partial h_1}{\partial \mathbf{u}_1} & \cdots & \frac{\partial h_1}{\partial \mathbf{u}_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_M}{\partial \mathbf{x}_1} & \cdots & \frac{\partial h_M}{\partial \mathbf{x}_M} & \frac{\partial h_M}{\partial \mathbf{u}_1} & \cdots & \frac{\partial h_M}{\partial \mathbf{u}_N} \end{pmatrix}. \quad (8)$$

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

³Smoothness is not mandatory, but it is a desirable property.

The Theorem imposes three conditions to be fulfilled. Firstly, 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 $\mathbf{x} \in X$, 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.

Secondly, 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} , $\mathbf{J}_{\mathbf{u}}$, to be non-singular in the desired location. Since for $m \neq n$ the matrix is not square, the right Moore-Penrose inverse matrix $\mathbf{J}_{\mathbf{u}}^+ = \mathbf{J}_{\mathbf{u}}^T (\mathbf{J}_{\mathbf{u}} \mathbf{J}_{\mathbf{u}}^T)^{-1}$ is generally considered as the one to be non-singular. An alternative is to resort to the Constant Rank Theorem (Theorem 5.22 in [36]) and find a projection of lower dimension which 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 $\mathbf{J}_{\mathbf{u}}$, 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 on 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 \mathbf{u}_0 such that $h(\mathbf{x}, \mathbf{u}_0) = \mathbf{0}$, the functions f and f^* are of class C^1 and $\mathbf{J}_{\mathbf{u}}$ has rank m for all $(\mathbf{x}, \mathbf{u}) \in I$.*

The problems of 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 \mathbf{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 a desired dynamics h^* ,

$$\frac{dh(\mathbf{x}, \mathbf{u})}{dt} = h^*(\mathbf{x}, \mathbf{u}), \quad (9)$$

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,

$$\dot{\mathbf{u}} = \mathbf{J}_{\mathbf{u}}^{-1}(h^*(\mathbf{x}, \mathbf{u}) - \mathbf{J}_{\mathbf{x}}f(\mathbf{x}, \mathbf{u})), \quad (10)$$

where $\mathbf{J}_{\mathbf{x}}$ is the Jacobian of h with respect to \mathbf{x} and $\mathbf{J}_{\mathbf{u}}$ is the Jacobian of h with respect to \mathbf{u} . This allows to build an explicit expanded system of the form

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ \dot{\mathbf{u}} = \mathbf{J}_{\mathbf{u}}^{-1}(h^*(\mathbf{x}, \mathbf{u}) - \mathbf{J}_{\mathbf{x}}f(\mathbf{x}, \mathbf{u})) \end{cases}. \quad (11)$$

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 towards zero and achieves the desired closed-loop dynamics in (5).*

Proof. Applying the chain rule over dh/dt yields to

$$\frac{dh(\mathbf{x}, \mathbf{u})}{dt} = \mathbf{J}_{\mathbf{x}}(\mathbf{x}, \mathbf{u}) \frac{d\mathbf{x}}{dt} + \mathbf{J}_{\mathbf{u}}(\mathbf{x}, \mathbf{u}) \frac{d\mathbf{u}}{dt}, \quad (12)$$

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

This ensures that, at some point, $h = 0$. However, in the transient, $h \neq 0$ so the input \mathbf{u} is not the one that achieves $f = f^*$. This non-zero difference can be seen as a perturbation in the input \mathbf{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). \square

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

$$f^*(\mathbf{x}) = -\mathbf{K}_f(\mathbf{x})\mathbf{x}, \quad (13)$$

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

$$h^*(\mathbf{x}, \mathbf{u}) = -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))h(\mathbf{x}, \mathbf{u}), \quad (14)$$

with $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ a function which 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 $\mathbf{K}_f(\mathbf{x})$ and $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ are chosen in such a way that*

$$\mathbf{K} = \begin{pmatrix} -\mathbf{K}_f(\mathbf{x}) & 0.5\mathbf{I} \\ 0.5\mathbf{I} & -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u})) \end{pmatrix} \quad (15)$$

is negative definite, then the system in Eq. (11) is Globally Asymptotically Stable (GAS).

Proof. Consider $V = \frac{1}{2}\mathbf{x}^T\mathbf{x} + \frac{1}{2}h^Th$ as a Lyapunov candidate function, whose derivative is

$$\dot{V} = \mathbf{x}^T\dot{\mathbf{x}} + h^T\dot{h} = \mathbf{x}^Tf + h^Th^* \quad (16)$$

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

$$\dot{V} = \mathbf{x}^Th + \mathbf{x}^Tf^* + h^Th^*. \quad (17)$$

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

$$\dot{V} = (\mathbf{x}^T \quad h^T) \mathbf{K} (\mathbf{x}^T \quad h^T)^T. \quad (18)$$

Then, designing $\mathbf{K}_f(\mathbf{x})$ and $\mathbf{K}_h(h(\mathbf{x}, \mathbf{u}))$ such that \mathbf{K} is negative definite guarantees that (11) is GAS, and both h and \mathbf{x} go to zero. \square

In practice, $\|\mathbf{K}_h\| \gg \|\mathbf{K}_f\|$ 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 $\mathbf{J}_{\mathbf{u}}^{-1}$ with the pseudoinverse $\mathbf{J}_{\mathbf{u}}^+$. In the case that there are some states whose dynamics are not explicitly dependent on \mathbf{u} (for instance, the velocity of the evaders might not depend on herders' position, but their acceleration), then a linear transformation \mathbf{T} over h can be applied

$$\bar{h}(\bar{\mathbf{x}}, \mathbf{u}) = \mathbf{T}h(\mathbf{x}, \mathbf{u}). \quad (19)$$

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

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

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

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

Proof. By assuming that $\dot{\mathbf{x}}^*$ 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

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

The expression comes from imposing that the error dynamics follow f^* . 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 $\dot{\mathbf{x}}^*$ is implicitly included in the Jacobians of the new h , therefore the structure in system (11) remains. Moreover, since $\dot{\mathbf{x}}^*$ does not depend on \mathbf{u} , the Jacobians in Theorems 2-3 do not change and the statements in Theorems 2-3 hold. \square

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 towards particular positions but also steer them towards a particular region whereas 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 Subsection 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 \mathbf{x} and \mathbf{u} from an observer and/or from measurements. Then, $f^*(\mathbf{x})$ is computed, which, together with the dynamic model of the evaders $f(\mathbf{x}, \mathbf{u})$ (e.g., (2) or (3)), allows to compute $h^*(\mathbf{x}, \mathbf{u})$ with Eq. (14). Besides, the Jacobians \mathbf{J}_x and \mathbf{J}_u can be calculated either analytically or numerically, depending on their complexity. Finally, $\dot{\mathbf{u}}$ is computed from Eq. (11). Regarding the estimation of \mathbf{x} and \mathbf{u} , another advantage of Implicit Control is that \mathbf{u} now belongs to the state of an expanded system. Therefore, it is possible to apply the same estimation techniques as with \mathbf{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 which overcomes this issue and preserves the control properties.

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

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

We assume a linear dependency 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 θ 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 $\tilde{\cdot}$ denote the quantities that depend on $\hat{\theta}$ and $\tilde{\theta}$. In general, $\tilde{\theta} \neq 0$ so Implicit Control no longer guarantees stability. The difference between actual and estimated dynamics can be defined by a new function

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

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 $\dot{\tilde{\theta}}$ 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

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

Here,

$$\hat{\mathbf{J}}_x = \begin{pmatrix} \frac{\partial \hat{h}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \hat{h}_1}{\partial \mathbf{x}_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{h}_m}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \hat{h}_m}{\partial \mathbf{x}_m} \end{pmatrix}, \quad \hat{\mathbf{J}}_u = \begin{pmatrix} \frac{\partial \hat{h}_1}{\partial \mathbf{u}_1} & \cdots & \frac{\partial \hat{h}_1}{\partial \mathbf{u}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{h}_m}{\partial \mathbf{u}_1} & \cdots & \frac{\partial \hat{h}_m}{\partial \mathbf{u}_n} \end{pmatrix}, \quad (25)$$

$\tilde{\mathbf{J}}_\theta = \text{diag}(f)$, 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

$$\begin{cases} \dot{\mathbf{x}} = \theta f \\ \dot{\mathbf{u}} = \hat{\mathbf{J}}_u^+(\tilde{h}^* - \hat{\mathbf{J}}_x \dot{\mathbf{x}}) \\ \dot{\tilde{\theta}} = -\tilde{\mathbf{J}}_\theta^{-1}(\tilde{h}^* + \hat{\mathbf{J}}_x \dot{\mathbf{x}} + \hat{\mathbf{J}}_u \dot{\mathbf{u}}) \end{cases}. \quad (26)$$

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

$$\tilde{h}^* = h_\theta^* - \dot{h} + \hat{\mathbf{J}}_x(\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}), \quad (27)$$

where h_θ^* is a free-design function chosen to make \tilde{h} stable.

Proof. The proof follows some of the steps in Theorem 2. Let consider the chain rule expansion of $d\tilde{h}/dt$,

$$\frac{d\tilde{h}}{dt} = \frac{\partial \tilde{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \tilde{h}}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial \tilde{h}}{\partial \tilde{\theta}} \dot{\tilde{\theta}}. \quad (28)$$

Since θ is constant, $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$ and the substitution of Eq. (24) gives

$$\frac{d\tilde{h}}{dt} = \mathbf{J}_x \dot{\mathbf{x}} - \hat{\mathbf{J}}_x \dot{\mathbf{x}} + \mathbf{J}_u \dot{\mathbf{u}} - \hat{\mathbf{J}}_u \dot{\mathbf{u}} + \tilde{h}^* + \hat{\mathbf{J}}_x \dot{\hat{\mathbf{x}}} + \hat{\mathbf{J}}_u \dot{\mathbf{u}}. \quad (29)$$

Note that $\mathbf{J}_x \dot{\mathbf{x}} + \mathbf{J}_u \dot{\mathbf{u}}$ is equal to \dot{h} , and the $\hat{\mathbf{J}}_u \dot{\mathbf{u}}$ terms cancel out. Therefore, Eq. (29) can be written as

$$\frac{d\tilde{h}}{dt} = \dot{h} + \tilde{h}^* - \hat{\mathbf{J}}_x (\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}) = h_\theta^*, \quad (30)$$

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

$$\frac{d\hat{h}}{dt} = \frac{\partial \hat{h}}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \hat{h}}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial \hat{h}}{\partial \theta} \dot{\theta}. \quad (31)$$

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

$$\frac{d\hat{h}}{dt} = \hat{\mathbf{J}}_x \dot{\mathbf{x}} + \hat{h}^* - \hat{\mathbf{J}}_x \dot{\hat{\mathbf{x}}} - \tilde{h}^* - \hat{\mathbf{J}}_x \dot{\hat{\mathbf{x}}} - \hat{h}^* + \hat{\mathbf{J}}_x \dot{\hat{\mathbf{x}}}. \quad (32)$$

By exploiting \tilde{h}^* in Eq. (27), we get

$$\frac{d\hat{h}}{dt} = \dot{h} - h_\theta^*. \quad (33)$$

Finally, let consider the chain rule expansion of dh/dt

$$\frac{dh}{dt} = \frac{\partial h}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial h}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial h}{\partial \theta} \dot{\theta} = \mathbf{J}_x \dot{\mathbf{x}} + \mathbf{J}_u \dot{\mathbf{u}}. \quad (34)$$

The substitution of $\dot{\mathbf{u}}$ in Eq. (26) gives

$$\frac{dh}{dt} = \mathbf{J}_x \dot{\mathbf{x}} + \mathbf{J}_u \hat{\mathbf{J}}_u^+ (\hat{h}^* - \hat{\mathbf{J}}_x \dot{\hat{\mathbf{x}}}) = \mathbf{J}_x (\dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}) + h^* = \mathbf{J}_x \dot{\hat{h}} + h^*. \quad (35)$$

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

Proposition 2. *Given Assumption 1,*

$$h_\theta^*(\mathbf{x}, \mathbf{u}) = -\mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) \tilde{h} \quad (36)$$

with $\mathbf{K}_\theta(\mathbf{x}, \mathbf{u})$ a function which returns a positive definite matrix, and $\mathbf{K}(h(\mathbf{x}, \mathbf{u}))$ defined as in (15), if

$$\bar{\mathbf{K}} = \begin{pmatrix} -\mathbf{K}_f(\mathbf{x}) & 0.5\mathbf{I} & \mathbf{0} \\ 0.5\mathbf{I} & -\mathbf{K}_h(h(\mathbf{x}, \mathbf{u})) & -0.5\mathbf{J}_x \mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) \\ \mathbf{0} & -0.5\mathbf{J}_x \mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) & -\mathbf{K}_\theta(\mathbf{x}, \mathbf{u}) \end{pmatrix} \quad (37)$$

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

Proof. The proof follows from Theorem 3 by considering

$$V = \frac{1}{2} \mathbf{x}^T \mathbf{x} + \frac{1}{2} h^T h + \frac{1}{2} \tilde{h}^T \tilde{h} \quad (38)$$

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

$$\begin{aligned} \dot{V} = & \mathbf{x}^T h + \mathbf{x}^T f^* + h^T h + \tilde{h}^T \dot{\tilde{h}} = \\ & \mathbf{x}^T h - \mathbf{x}^T \mathbf{K}_f \mathbf{x} - h^T \mathbf{J}_x \mathbf{K}_\theta \tilde{h} - h^T \mathbf{K}_h h - \tilde{h}^T \mathbf{K}_\theta \tilde{h}. \end{aligned} \quad (39)$$

As in Theorem 3, the latter reads

$$\dot{V} = (\mathbf{x}^T \ h^T \ \tilde{h}^T) \bar{\mathbf{K}} (\mathbf{x}^T \ h^T \ \tilde{h}^T)^T. \quad (40)$$

Then, if $\mathbf{K}, \mathbf{K}_\theta$ are designed such that $\bar{\mathbf{K}}$ is negative definite, then the adaptation law preserves the stability properties of Theorem 3. \square

The adaptation law, then, evolves the estimates towards the values which ensure the correct behavior of the control. Remarkably, the convergence of \tilde{h} to zero implies that \hat{h} converges to h . Therefore, the convergence of \hat{h} to zero implies the convergence of h to zero, and the evaders behave as desired. In practice, to implement the controller, a slight change in Eq. (26) is required to compute \tilde{h}^* . Since $\dot{\mathbf{x}} = \theta f$ is not available, an approximation (e.g., by finite differences) must be used to obtain $\dot{\hat{\mathbf{x}}}$. 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 a desired settling time of $1/\tau = 12$ s and an exponential transient according to m first order independent systems. Thus,

$$f^*(\mathbf{x}) = -\mathbf{K}_f(\mathbf{x}) \mathbf{x} = -\mathbf{K}_f \mathbf{x} = -(3\tau \mathbf{I}) \mathbf{x} = -0.25 \mathbf{I} \mathbf{x}. \quad (41)$$

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

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

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 Subsection describes a distributed estimator that fulfills this assumption. Particularly, we develop an extended version of the Distributed Kalman Filter [9] (E-DKF), 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 $\xi = [\mathbf{x}_1^T \dots \mathbf{x}_m^T \mathbf{u}_1^T \dots \mathbf{u}_n^T]^T$. 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

$$\dot{\xi} = \frac{\partial \Sigma(\xi)}{\partial \xi} \Big|_{\xi=\xi(t)} + \mathbf{w}, \quad (42)$$

where $\Sigma(\xi)$ denotes the expanded system in (11) and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is a zero mean Gaussian noise with covariance \mathbf{Q} . Note that the linearization in Eq. (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, \mathbf{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

$$\mathbf{z}_i = \mathbf{H}_i(\xi)\xi + \mathbf{v}_i, \quad (43)$$

where \mathbf{z}_i is the measurement taken by herder i , $\mathbf{v}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i)$ is the measurement noise with covariance \mathbf{R}_i , and \mathbf{H}_i depends on ξ . This dependency makes the dimension of \mathbf{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., $\mathbf{z}_i \in \mathbb{R}^{2m+2n}$. 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 $\mathbf{x}_{avg} := \frac{1}{m} \sum_j \mathbf{x}_j$ and $\mathbf{P}_{avg} := \frac{1}{m} \sum_j (\mathbf{x}_j - \mathbf{x}_{avg})(\mathbf{x}_j - \mathbf{x}_{avg})^T$. The combination of both gives a description of the herd in terms of an ellipsoid centered in \mathbf{x}_{avg} . The geometry of the setup for a 2D 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, $\mu_1 \mathbf{P}_{avg}$ 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}_{avg} will only imply that it will take more time for the herders to surround the evaders. Moreover, μ_1 can be tuned so that herders provoke weak repulsive forces in the evaders. During the caging, evaders will not escape, and afterwards, 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 towards an even distribution along the perimeter of $\mu_1 \mathbf{P}_{avg}$. The position of herder i , in polar coordinates with respect to the ellipse, is driven by

$$\dot{\psi}_i = (\psi_{i+1} - \psi_i) + (\psi_{i-1} - \psi_i) \quad (44)$$

$$\dot{\rho}_i = \rho_i - \rho_i^* \quad (45)$$

where ψ_i and ρ_i are the angular and radial positions, and $i+1$, $i-1$ denote the left and right closest herders. The policy is inspired in the well known N -bugs problem, that 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 Eq. (44). Meanwhile, to each angle corresponds a certain radial position to place herders over the desired ellipse, given by ρ_i^* . Therefore, the controller of the radial position is just the linear feedback controller in Eq. (45). For formal guarantees of convergence, we refer to [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.

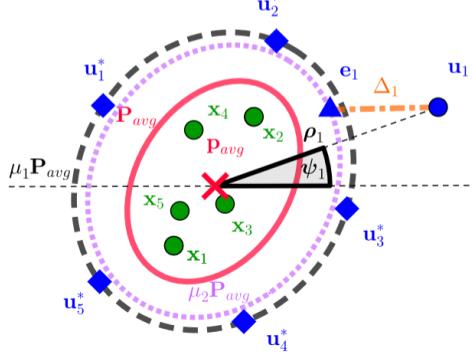


Fig. 2. Geometrical view of the caging stage. The evaders (green dots) are in the region determined by their mean position and covariance (in red). To avoid escapes and high repulsive forces, the herder (blue dot) must be placed in the perimeter (blue diamonds) of an augmented ellipse (black-dashed). Besides, since there is uncertainty, CBFs prevent the herders to enter in the magenta ellipse, computing the closest point to its perimeter (blue triangle).

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 Eqs. (44)-(45) computes the velocity $\dot{\mathbf{u}}_i^{nom}$, which is then translated into Cartesian coordinates. This is called the *nominal action*. The nominal action might move the herder towards a region where the interaction with the evaders is not convenient. We define this region by $\mu_2 \mathbf{P}_{avg}$, 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 $\mu_2 \mathbf{P}_{avg}$ to herder i is computed, denoted by \mathbf{e}_i [42]. Then, an optimization process [41] is carried out,

$$\dot{\mathbf{u}}_i^{cbf} = \arg \min_{\dot{\mathbf{u}}_i^{cbf}} \|\dot{\mathbf{u}}_i^{cbf} - \dot{\mathbf{u}}_i^{nom}\|^2 \quad (46a)$$

$$s.t. \quad L_f h(\Delta_i) + L_g h(\Delta_i) \dot{\mathbf{u}}_i^{cbf} \geq -\kappa(h(\Delta_i)) \quad (46b)$$

Here, $\Delta_i = \mathbf{u}_i - \mathbf{e}_i$, depicted in orange in Fig. 2. Moreover, $\dot{\mathbf{u}}_i^{cbf}$ is the action to be optimized, $h(\Delta_i)$ is the control barrier function, $\kappa(h(\Delta_i))$ is a class \mathcal{K} function and L_f , L_g 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.

Algorithm 1 Complete herding for herder i

- 1: Receive \mathbf{x}_{avg} and \mathbf{P}_{avg}
 - 2: **while** True **do**
 - 3: Compute estimate $\bar{\xi}_i$ with E-DKF
 - 4: **if** herders are far from evaders **then**
 - 5: **Do caging:** compute $\dot{\psi}_i$, $\dot{\rho}_i$, calculate $\dot{\mathbf{u}}_i^{cbf}$
 - 6: **else**
 - 7: **Do precise herding:** compute $\dot{\mathbf{u}}_i$, $\dot{\hat{\theta}}_i$ with Adaptive Implicit Control, and update $\hat{\theta}_i$
 - 8: **end if**
 - 9: Update \mathbf{u}_i
 - 10: **end while**
-

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 5 evaders by 5 herders. The herders receive perfect measurements and they are already deployed near the evaders. The details are in Table II. We set $\mathbf{K}_f = 0.25\mathbf{I}_{2m}$, yielding a settling time of 12s with an exponential transient according to $2m$ first order independent systems. To ensure that the conditions of Theorem 3 and Proposition 2 hold, we set $\mathbf{K}_h = 50\mathbf{I}_{2m}$ and $\mathbf{K}_\theta = 200\mathbf{I}_{2m}$. For the sake of comparison, we use the Levenberg-Marquardt numerical method [43] as a *baseline* to compute the control input.

TABLE I
LIST OF SYMBOLS IN THE FIGURES.

Symbol		Meaning	
		Desired positions	
Symbol	Meaning	Symbol	Meaning
●	Initial position herders	▼	Initial position evaders
●	Final position herders	▲	Final position evaders
—	Trajectories herders	—	Trajectories evaders

TABLE II
PARAMETERS OF THE SIMULATIONS.

θ_j (Inv.)	θ_j (Exp.)	β_j	σ_j	d_{min}	T	v_{max}	d_m	d_c
1.0	0.5	0.5	2.0	1.0m	10ms	0.4m/s	6.5m	6.5m

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 12s 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 towards 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 can not 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.

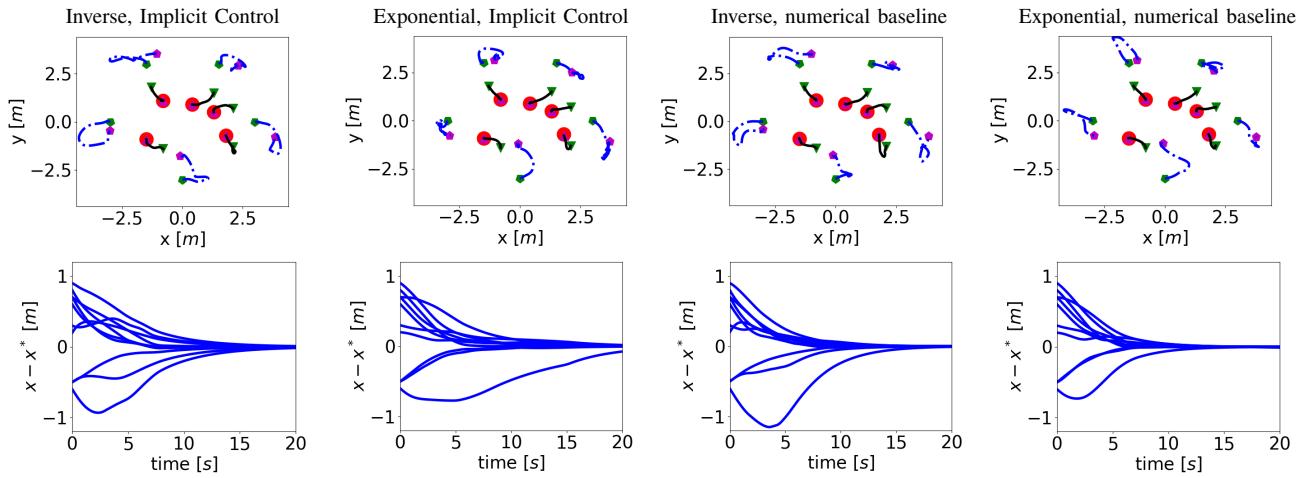


Fig. 3. Simulation results of the herding of 5 evaders by 5 herders. The first row shows the trajectories followed by the herders and the evaders. The symbols are explained in Table I. The second row presents the evolution of the error between desired and current position of the evaders.

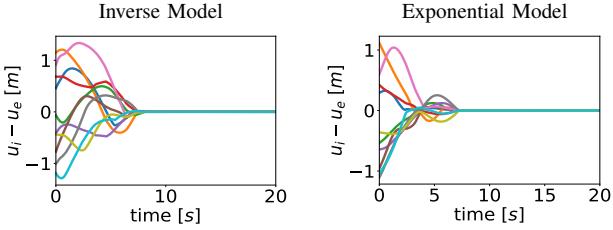


Fig. 4. Difference in the control input calculated by Implicit Control and the numerical baseline (u_i and u_e respectively). An arbitrary color has been assigned to each input for the sake of visibility.

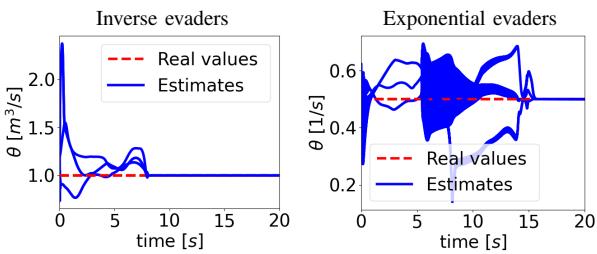


Fig. 5. Adaptation results of the herding of 5 evaders by 5 herders. The pannels depict the evolution of the estimated parameters.

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 $f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) = 0$. Meanwhile, Implicit Control requires some time to converge to $f(\mathbf{x}, \mathbf{u}) - f^*(\mathbf{x}) = 0$, 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 $f^*(\mathbf{x})$ plus the time $h(\mathbf{x}, \mathbf{u})$ takes to converge to zero. If \mathbf{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 $h(\mathbf{x}, \mathbf{u}) = 0$. After Implicit Control converges to $h(\mathbf{x}, \mathbf{u}) = 0$, herders are in a different configuration with respect to the herders driven by

the numerical baseline, and this difference implies different roots for $h(\mathbf{x}, \mathbf{u})$ 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 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 $\mathbf{Q} = 0.02\mathbf{I}_{2m+2n}$ and $\mathbf{R}_i = 0.07\mathbf{I}_{2m+2n} \forall i$ to simulate a scenario with bad sensing capabilities, i.e., $\mathbf{R}_i \succ \mathbf{Q}$ 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 100ms, 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) between the estimates and real positions of the entities rapidly decreases to the noise level. We remark that both axis of the panels are in logarithmic scale. Thus, before the first 400ms (4 communication rounds) the estimator has converged, which justifies the assumption of perfect knowledge in the control and adaptation. Despite the different frequencies between communication and control, the herders are successful because they can predict the movement of the evaders and the other herders.

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 changing the measurement covariance matrix, checking to what extent the

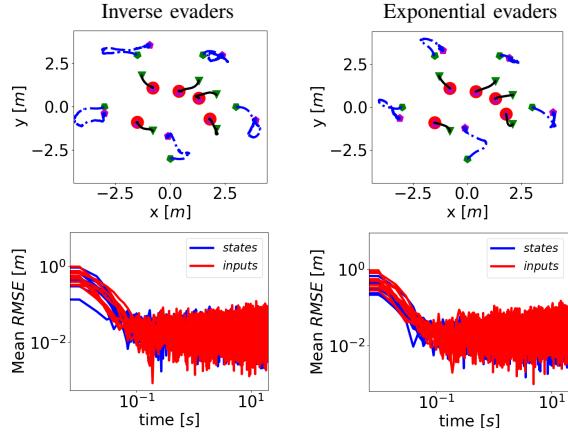


Fig. 6. Simulation results of the herding of 5 evaders by 5 herders using Implicit Control with adaptation law and the distributed estimator. The first row shows the trajectories followed by the herders (same symbols of Table I). The second row depicts the RMSE in the estimation of evaders' (blue) and herders' (red) position, averaged over the five herders.

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 0.3m^2 . We underline that this implies measurement errors in the order of 0.5m, which is significant considering that the herding takes place in a $5 \times 5\text{m}$ 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 \mathbf{Q} because the movement of herders and evaders is deterministic, so \mathbf{Q} just models the noise associated with the difference between real and estimated parameters during the adaptation.

TABLE III
IMPACT OF THE MEASUREMENT NOISE IN THE HERDING.

$\mathbf{R} [\text{m}^2]$	Steady-state error $\ \mathbf{x} - \mathbf{x}^*\ [\text{m}]$	
	Inverse Model, 5vs5	Exponential Model, 5vs5
$0.07\mathbf{I}_{20}$	0.02	0.01
$0.15\mathbf{I}_{20}$	0.06	0.06
$0.30\mathbf{I}_{20}$	1.34	0.12

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 while the purple evaders are Inverse. The CBFs have been tuned as follows, choosing

$$h(\Delta_i) = \|\Delta_i\| + T \frac{\Delta_i \delta \mathbf{u}_i^{nom}}{\|\Delta_i\|} - \varphi \quad (47)$$

and

$$\kappa(h(\Delta_i)) = k_{cbf} h(\Delta_i)^3, \quad (48)$$

where T is the sampling time, $\varphi = 3$ is a desired distance with the perimeter of $\mu_2 \mathbf{P}_{avg}$ ($\mu_1 = 7$ and $\mu_2 = 4$) and $k_{cbf} = 50$ is a gain which modulates how strong is the repulsion provoked

by the proximity with \mathbf{e}_i . We recall that in the caging stage the dynamics of the herders are

$$\mathbf{u}_i = \mathbf{u}_i + T \delta \mathbf{u}_i^{cbf}. \quad (49)$$

Meanwhile, the desired herding configuration evolves with

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

where $\mathbf{w}^* = [0.05, 0.1, 0.02]\text{rad/s}$ and $\mathbf{v}^* = [0.05, 0.05, 0.05]\text{m/s}$. 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 towards 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, point-wise, the centroid's dynamics is still that of the virtual evader. Besides, this discrepancy can be absorbed by the adaptation law.

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 position of the centroid). To maintain cohesion of the herd, but also to avoid collisions among the evaders, we have added very weak repulsive and coalition forces among evaders

$$\dot{\mathbf{x}}_j = f_j^{inv/exp}(\mathbf{x}, \mathbf{u}) + \vartheta \sum_{j'=1}^m \mathbf{d}_{jj'} \left(\frac{1}{\|\mathbf{d}_{jj'}\|^3} - \|\mathbf{d}_{jj'}\|^2 \right) \quad (50)$$

with $\mathbf{d}_{jj'} = \mathbf{x}_j - \mathbf{x}_{j'}$ and $\vartheta = 2 \times 10^{-4}$, i.e., 4 orders of magnitude smaller than the dynamics in Eqs. (2) and (3).

The first row of Fig. 8 shows how the five herders can steer the whole herd towards 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 pannel): there are two herders which share a similar position. More interestingly, the second row of Fig. 8 shows how the herders can split the herd in two sub-herds and steer each of them, simultaneously, towards 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).

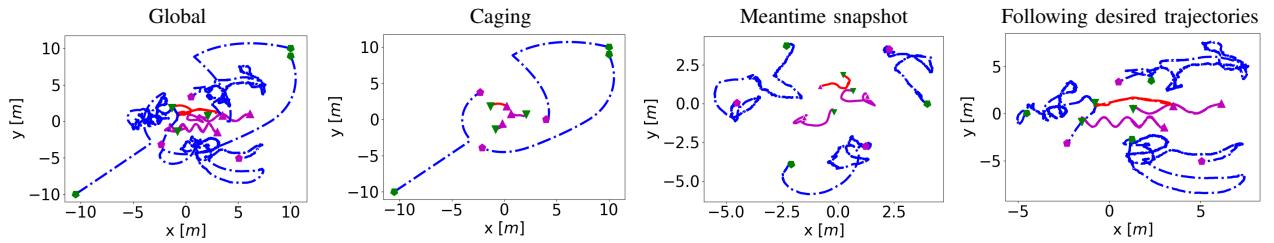


Fig. 7. Three robotic herders herding three heterogeneous evaders. The magenta trajectories correspond to Inverse evaders while the red trajectory is an Exponential evader. The other symbols follow the convention in Table I.

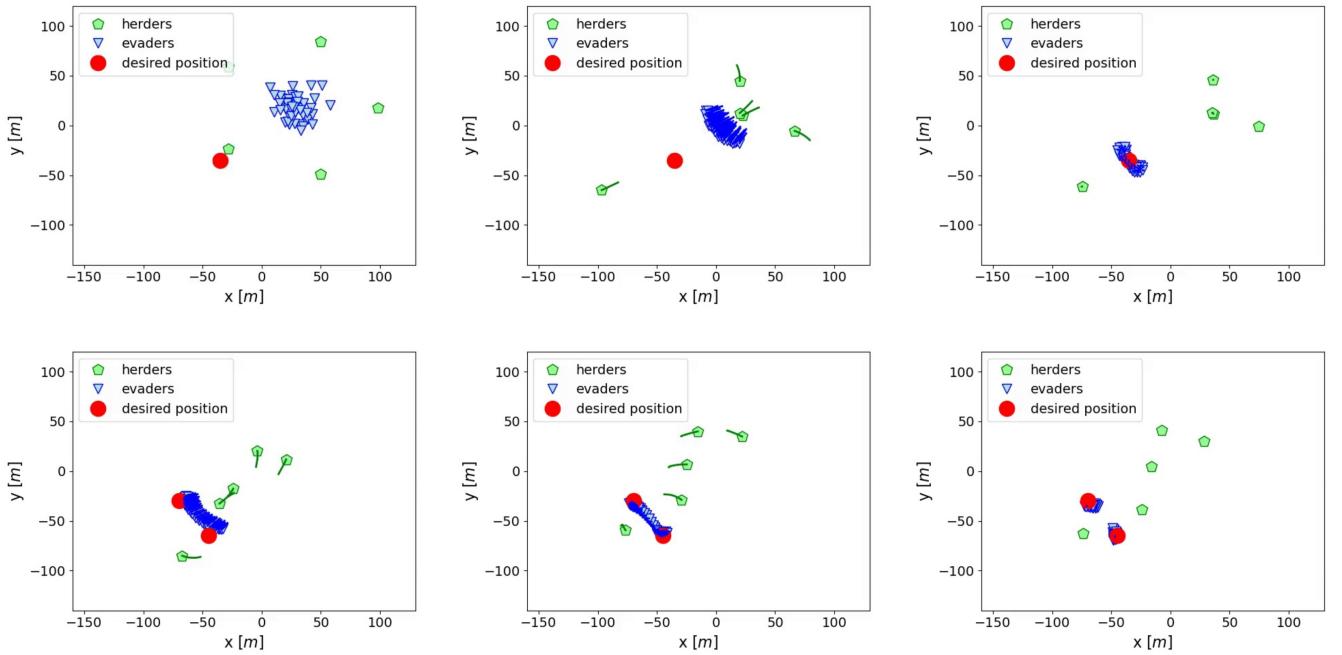


Fig. 8. 5 herders herding 50 Inverse evaders. In the first 50 seconds, the herd is steered towards the desired location (first row). Afterwards, the herd is split in two sub-herds and driven towards two different desired locations simultaneously (second row).

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 while the others act as evaders, following the dynamics in Section III. The robots are GRITSBot X moving on a 3.2m x 2m area, coordinated by a central server which receives odometry and sends velocity commands at an approximately delay of 0.033s. Thus, a low level controller is used to translate the control output into velocity commands, with $v_{max} = 0.2\text{m/s}$. 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 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. 9c, Fig. 9d, and in the time-varying experiment in Fig. 10.

The latter consists in 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 towards 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, for the case of 4 evaders vs 4 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 towards the desired dynamics. To avoid this behavior in a particular application, a low-level controller with an obstacle-avoidance mechanism can be incorporated.

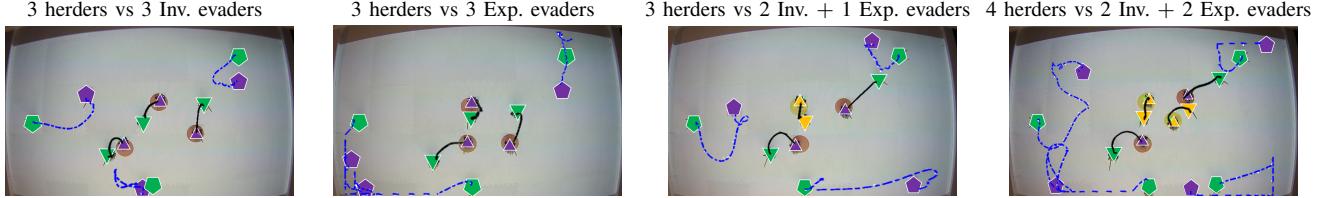


Fig. 9. Results of the herding using Robotarium [44]. The symbols are explained in Table I. When evaders' dynamics are mixed, initial, final and desired position of Exponential evaders are yellow. A video with the complete experiments is included as supplementary material.

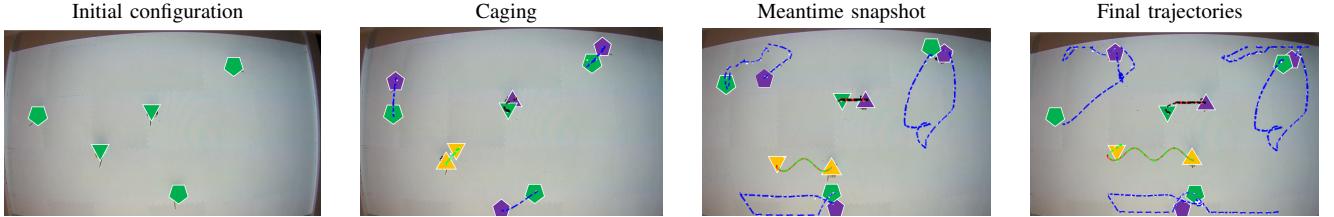


Fig. 10. Three herders herding two heterogeneous evaders in a time-varying path. More details in the supplementary video.

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.

IX. CONCLUSIONS

This paper 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 to derive an adaptation law. It preserves the stability properties of the control while naturally integrates in 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 has been 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 Robotics Automation Magazine*, vol. 15, no. 1, pp. 22–29, 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," *International Journal of Advanced Robotic Systems*, vol. 12, no. 11, p. 166, 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. 41 883–41 891, 2018.
- [4] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, 2018.
- [5] S. Nolfi, "Power and the limits of reactive agents," *Neurocomputing*, vol. 42, no. 1-4, pp. 119–145, 2002.
- [6] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. Sumpter, and A. J. King, "Solving the shepherding problem: heuristics for herding autonomous, interacting agents," *Journal of the royal society interface*, vol. 11, no. 100, 2014.
- [7] N. K. Long, K. Sammut, D. Sgarioto, M. Garratt, and H. A. Abbass, "A comprehensive review of shepherding as a bio-inspired swarm robotics guidance approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 523–537, 2020.
- [8] E. Sebastián and E. Montijano, "Multi-robot implicit control of herds," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 1601–1607.
- [9] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *IEEE Conference on Decision and 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. 155–161, 2017.
- [11] A. Franchi, P. Stegagno, and G. Oriolo, "Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance," *Autonomous Robots*, vol. 40, no. 2, pp. 245–265, 2016.
- [12] A. Jahn, R. J. Alitappeh, 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 *IEEE International Conference on Robotics and Automation*, 2017, pp. 273–278.
- [13] V. S. Chipade and D. Panagou, "Herding an adversarial swarm in an obstacle environment," in *IEEE Conference on Decision and Control*, 2019, pp. 5859–5864.
- [14] D. J. Stilwell, B. E. Bishop, and C. A. Sylvester, "Redundant manipulator techniques for partially decentralized path planning and control of a platoon of autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 4, pp. 842–848, 2005.
- [15] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1285–1292, 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," *Autonomous Robots*, pp. 1–13, 2022.
- [17] R. A. Licitira, Z. I. Bell, E. A. Doucette, and W. E. Dixon, "Single agent indirect herding of multiple targets: A switched adaptive control

- approach,” *IEEE Control System Letters*, vol. 2, no. 1, pp. 127–132, 2018.
- [18] R. A. Licitra, Z. I. Bell, and W. E. Dixon, “Single-agent indirect herding of multiple targets with uncertain dynamics,” *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 847–860, 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 Transactions on Robotics*, vol. 37, no. 2, pp. 418–432, 2020.
- [20] H. Song, A. Varava, O. Kravchenko, D. Kragic, M. Y. Wang, F. T. Pokorny, and K. Hang, “Herding by caging: a formation-based motion planning framework for guiding mobile agents,” *Autonomous Robots*, pp. 1–19, 2021.
- [21] W. Jiang, G. Wen, Z. Peng, T. Huang, and A. Rahmani, “Fully distributed formation-containment control of heterogeneous linear multiagent systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3889–3896, 2018.
- [22] D. A. Anisi, P. Ögren, and X. Hu, “Cooperative minimum time surveillance with multiple ground vehicles,” *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2679–2691, 2010.
- [23] M. V. Ramana and M. Kothari, “A cooperative pursuit-evasion game of a high speed evader,” in *IEEE Conference on Decision and Control*, 2015, pp. 2969–2974.
- [24] A. Alexopoulos, B. Kirsch, and E. Badreddin, “Realization of pursuit-evasion games with unmanned aerial vehicles,” in *International Conference on Unmanned Aircraft Systems*, 2017, pp. 797–805.
- [25] A. Franchi, P. Stegagno, M. D. Rocco, and G. Oriolo, “Distributed target localization and encirclement with a multi-robot system,” in *IFAC Proceedings Volumes*, vol. 43, 2010, pp. 151–156.
- [26] J. Zhu, W. Zou, and Z. Zhu, “Learning evasion strategy in pursuit-evasion by deep Q-network,” in *International Conference on Pattern Recognition*, 2018, pp. 67–72.
- [27] S. F. Desouky and H. M. Schwartz, “Learning in n-pursuer n-evader differential games,” in *IEEE International Conference on Systems, Man and Cybernetics*, 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,” in *International Journal of Advanced Robotic Systems*, vol. 13, 2016, pp. 5859–5864.
- [29] M. B. Estrada and S. P. Manjarres, “Implicit control law for linear time varying SISO systems with bounded disturbances,” in *IEEE Conference on Decision and Control*, vol. 3, 1996, 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 *IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 2120–2125.
- [31] K. Osuka, A. Ishiguro, X.-Z. Zheng, Y. Sugimoto, and D. Owaki, “Dual structure of mobilgence-implicit control and explicit control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2407–2412.
- [32] F. Blanchini, G. Fenu, G. Giordano, and F. A. Pellegrino, “Model-free plant tuning,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2623–2634, 2017.
- [33] S. Shivam, I. Buckley, Y. Wardi, C. Seatzu, and M. Egerstedt, “Tracking control by the newton-raphson flow: Applications to autonomous vehicles,” in *IEEE European Control Conference*, 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 *IEEE Conference on Decision and Control*, 2017, pp. 6462–6468.
- [35] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. McGraw-Hill, Inc., 1976.
- [36] J. M. Lee, *Introduction to Smooth Manifolds*. Springer, 2003.
- [37] H. K. Khalil, *Nonlinear Control*. Pearson Education, 2014.
- [38] M. Franceschelli and A. Gasparri, “Gossip-based centroid and common reference frame estimation in multiagent systems,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 524–531, 2013.
- [39] E. Montijano, S. Martínez, and C. Sagüés, “Distributed robust consensus using ransac and dynamic opinions,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 150–163, 2014.
- [40] E. Montijano, A. Priolo, A. Gasparri, and C. Sagüés, “Distributed entrapment for multi-robot systems with uncertainties,” in *IEEE Conference on Decision and Control*, 2013, pp. 403–408.
- [41] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 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 Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [44] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, “The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.



Eduardo Sebastián (S’21) received the B.Eng. in Electronic and Automatic Engineering (Hons.) and the M.Eng. in Electronics (Hons.) from the Universidad de Zaragoza in 2019 and 2020 respectively. He is currently a Ph.D. Candidate in the Departamento de Informática e Ingeniería de Sistemas at the Universidad de Zaragoza, funded by a FPU national grant (1st rank). His research interests are nonlinear control, distributed systems and multi-robot systems.



2013 academic year.

Eduardo Montijano (M’12) is an Associate Professor in the Departamento de Informática e Ingeniería de Sistemas at Universidad de Zaragoza in Spain. He received the M.Sc. and Ph.D. degrees from the Universidad de Zaragoza, Spain, in 2008 and 2012 respectively. He was a faculty member at Centro Universitario de la Defensa, Zaragoza, between 2012 and 2016. His main research interests include distributed algorithms and automatic control in perception problems. His Ph.D. obtained the extraordinary award of the Universidad de Zaragoza in the 2012-



Carlos Sagüés (Senior Member, IEEE) received the M.Sc. degree in computer science and systems engineering and the Ph.D. degree in industrial engineering from the University of Zaragoza, Zaragoza, Spain, in 1989 and 1992, respectively. In 1994, he joined as an Associate Professor with the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, where he became a Full Professor in 2009 and also the Head Teacher. He was engaged in research on force and infrared sensors for robots. His current research interests include control systems and industry applications, computer vision, visual control, and multivehicle cooperative control.