

A Multi-Agent System Architecture for Smart Pastoralism 4.0: Coordinating Robotic Shepherds, Drones, and IoT-Enabled Livestock

Lorenzo De Lauretis^{1,*}, Tiziano Lombardi¹

¹Department of Information Engineering, Computer Science and Mathematics (DISIM), University of L'Aquila, Via Vetoio, 67100 L'Aquila, Italy

Abstract

The growing demand for sustainable and efficient livestock management, combined with advances in robotics, Internet of Things (IoT), and artificial intelligence, opens new opportunities for the digitalisation of pastoral activities. In this paper, we propose a *Multi-Agent System* (MAS) architecture for *Smart Pastoralism 4.0*, in which heterogeneous autonomous agents—a robotic shepherd dog and an unmanned aerial vehicle (UAV)—cooperate with IoT-equipped livestock, acting as reactive sensor nodes, to achieve herding goals defined by a human farmer. Each animal wears a GPS/GNSS collar with integrated LoRaWAN communication and biometric sensors for health monitoring. The robotic shepherd dog navigates behind the herd and emits directional audio cues through an onboard speaker to guide animal movement, while the surveillance drone is dispatched when an individual strays from the flock to provide real-time video streaming to the farmer's smartphone. We formalise three high-level herding goals: (i) keeping the herd stationary at a designated location, (ii) relocating the herd along a computed path from point *A* to point *B*, and (iii) returning the herd to the barn. A graph-based path-planning module, grounded in Dijkstra's algorithm and enriched with terrain-aware cost functions, computes optimal routes over a discretised pasture map. We detail the MAS coordination protocol, the communication infrastructure leveraging LoRaWAN and 5G, the agent behavioural models based on the Belief-Desire-Intention (BDI) paradigm, and a formal validation of the proposed algorithms. Preliminary simulation results demonstrate the feasibility and effectiveness of the approach in realistic pastoral scenarios.

Keywords

multi-agent systems, smart pastoralism, precision livestock farming, robotic shepherd, UAV, LoRaWAN, path planning, BDI agents, Internet of Things

1. Introduction

Pastoral farming has been practised for millennia and remains a cornerstone of global food production. Traditional herding relies on the experience of human shepherds and trained sheepdogs to manage livestock across vast, often rugged, landscapes. However, the sector faces mounting pressures: labour shortages in rural areas, the need for continuous animal welfare monitoring, predator threats, and the economic imperative to manage larger herds with fewer resources [1, 2].

At the same time, the convergence of *Internet of Things* (IoT) technologies, low-power wide-area networks (LPWANs), unmanned aerial vehicles (UAVs), and quadruped robotics is reshaping what is technologically feasible in open-field agriculture [3, 4]. The concept of *Precision Livestock Farming* (PLF) leverages sensors, actuators, and data analytics to monitor and manage individual animals in real time [1, 5]. Yet, most PLF solutions focus on indoor or semi-intensive settings; the integration of autonomous robotic agents for *extensive* pastoral management remains largely unexplored.

Multi-Agent Systems (MAS) provide a natural paradigm for modelling and engineering such scenarios, where multiple heterogeneous entities—each with its own sensors, actuators, and decision-making capabilities—must coordinate to achieve shared objectives [6, 7]. In a MAS perspective, each device (collar, robot, drone) can be modelled as an autonomous agent with beliefs about the environment,

WOA 2026: 27th Workshop "From Objects to Agents", June 15–17, 2026, Salerno, Italy

*Corresponding author.

✉ lorenzo.delauritis@univaq.it (L. D. Lauretis); tiziano.lombardi@univaq.it (T. Lombardi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

desires reflecting herding goals, and intentions embodying concrete plans of action—the well-known *Belief-Desire-Intention* (BDI) architecture [8, 9].

In this paper, we propose **Smart Pastoralism 4.0**, a MAS architecture for the intelligent management of livestock on open pastures. The system comprises two categories of deliberative agents and one category of reactive sensor nodes:

1. **Livestock nodes**: each animal is equipped with a GPS/GNSS collar integrating LoRaWAN connectivity and biometric sensors (heart rate, body temperature, rumination activity) for continuous health and position monitoring.
2. **Robotic Shepherd Dog (RSD) agent**: a commercial quadruped robot fitted with a directional speaker for emitting herding sounds, a GPS receiver, a LoRaWAN transceiver, and onboard cameras for local perception.
3. **Surveillance Drone (SD) agent**: a commercial UAV equipped with a high-resolution camera, GPS, an integrated SIM card for 5G/4G connectivity, and obstacle-avoidance sensors, capable of semi-autonomous flight.

A human *Farmer* interacts with the system through a mobile application, defining high-level *herding goals*:

- **Goal G1 – Hold**: keep the herd stationary at a designated geographic point.
- **Goal G2 – Move**: relocate the herd from a start point *A* to a destination point *B* along an optimal path.
- **Goal G3 – Return**: bring the herd back to the barn.

The main contributions of this work are: (i) a formal MAS architecture for smart pastoralism integrating heterogeneous robotic and IoT agents; (ii) a BDI-based coordination protocol for herding, including a stray-detection and drone-dispatch mechanism; (iii) a terrain-aware path-planning algorithm based on Dijkstra’s shortest-path algorithm [10] adapted with geographic cost functions; and (iv) a theoretical validation of the proposed algorithms with correctness and convergence arguments.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 presents the overall system architecture. Section 4 details the agent models. Section 5 describes the path-planning module. Section 6 formalises the multi-agent coordination protocol. Section 7 discusses the communication infrastructure. Section 8 provides theoretical validation and simulation results. Section 9 discusses limitations and future directions. Section 10 concludes the paper.

2. Related Work

2.1. Precision Livestock Farming and IoT

Precision Livestock Farming (PLF) applies engineering principles to monitor and manage livestock with the aim of improving animal welfare, productivity, and environmental sustainability [1]. Wearable sensors for livestock—including accelerometers, GPS receivers, and physiological monitors—have been extensively studied [11, 12]. GPS-based tracking of grazing animals has been used to study spatial behaviour [13], detect lameness, and implement virtual fencing [14, 15]. However, these systems are predominantly passive: they collect data for offline analysis rather than actuating real-time herding behaviours.

The Internet of Things has catalysed the integration of heterogeneous sensors and actuators in agricultural settings [3, 4]. Wireless sensor networks based on LoRaWAN offer long-range, low-power communication ideally suited to rural deployments where cellular infrastructure may be sparse [16, 17]. More recently, the roll-out of 5G networks promises ultra-reliable low-latency communication that can support real-time video streaming and remote robot control [18, 19].

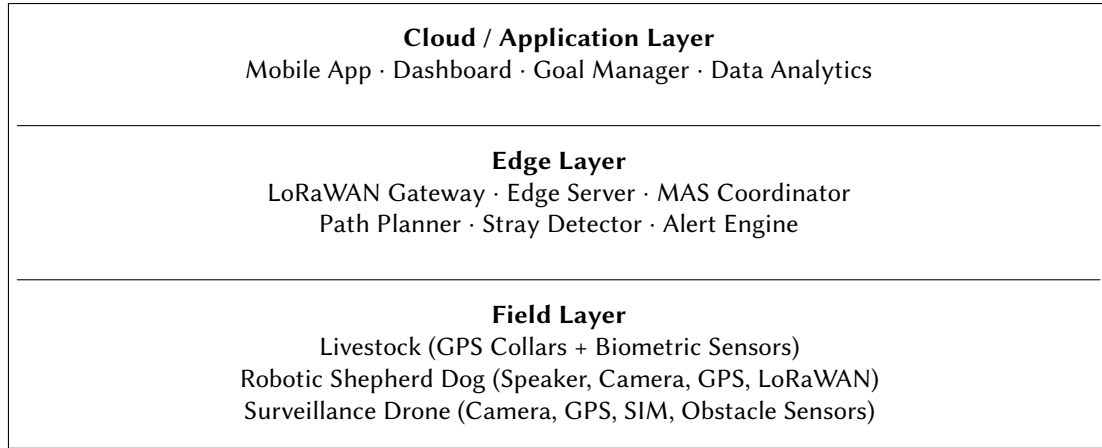


Figure 1: High-level architecture of Smart Pastoralism 4.0.

2.2. Shepherd Algorithms and Robotic Herding

The *shepherding problem*—how one or more controlling agents can guide a group of autonomous agents to a target—has attracted significant attention in computational biology and robotics. Strömbom et al. [20] proposed a heuristic model in which a single shepherd alternates between two behaviours: *collecting* (moving scattered agents into a cohesive group) and *driving* (pushing the group towards the goal). This work builds upon earlier studies by Reynolds [21], who introduced the celebrated *boids* model of flocking, and by Vaughan et al. [22], who demonstrated robotic flock control with a simple mobile robot.

Lien et al. [23] formalised shepherding behaviours within a motion-planning framework, while Fujioka and Hayashi [24] extended the approach to multi-agent shepherds. Collective animal motion itself is well captured by models such as Vicsek’s self-propelled particles [25] and Olfati-Saber’s flocking theory [26].

2.3. UAVs in Livestock Management

Unmanned aerial vehicles are increasingly deployed for agricultural monitoring, including crop surveillance, wildlife tracking, and livestock counting [27]. Xu et al. [28] demonstrated the use of deep learning on drone imagery for livestock classification and counting. While UAVs have been proposed as herding aids—leveraging their aerial perspective to detect stragglers—the systematic integration of drones within a multi-agent herding system has not been thoroughly addressed.

2.4. Multi-Agent Systems in Agriculture

Multi-Agent Systems offer a well-established theoretical and practical framework for coordinating autonomous entities [6, 29]. The BDI architecture [8] is particularly suited to agents that must reason about goals and plans in dynamic environments. Task allocation in multi-robot systems [30] and swarm robotics [31] provide complementary perspectives on coordination. Platforms such as JADE [32] facilitate the development and deployment of MAS. Despite these advances, the application of MAS specifically to robotic pastoralism remains an open research area.

3. System Architecture

Figure 1 provides a high-level overview of the Smart Pastoralism 4.0 architecture. The system is structured in three layers: the *Field Layer*, the *Edge Layer*, and the *Cloud/Application Layer*.

3.1. Field Layer

The Field Layer encompasses all physical devices deployed in the pasture:

Livestock Collars. Each animal wears a ruggedised collar housing: (i) a multi-constellation GNSS receiver (GPS + Galileo) providing 2–5 m positioning accuracy under open-sky conditions [33]; (ii) a LoRaWAN Class A radio module for periodic uplink of position and health telemetry to the nearest gateway [34, 16]; and (iii) biometric sensors measuring heart rate, skin temperature, and tri-axial acceleration (for activity classification). The collar transmits a telemetry packet every T_{collar} seconds (configurable, default 30 s).

Robotic Shepherd Dog (RSD). The RSD is a commercial quadruped robot (e.g., Boston Dynamics Spot [35] or similar) augmented with: (i) a directional loudspeaker capable of emitting pre-recorded herding sounds (whistles, barks, commands) at controlled volume and direction; (ii) a GNSS receiver; (iii) a LoRaWAN transceiver for communication with the edge server and collar network; and (iv) stereo cameras and LiDAR for local obstacle avoidance and terrain perception. The RSD operates behind the herd and uses audio stimuli to steer the animals, emulating the behaviour of a trained sheepdog [36, 37].

Surveillance Drone (SD). The SD is a commercial multi-rotor UAV equipped with: (i) a high-resolution gimbal camera for aerial video; (ii) a GNSS receiver; (iii) a 4G/5G SIM module for real-time video streaming and command reception; and (iv) ultrasonic and infrared sensors for obstacle avoidance [27]. The SD remains on a charging station during normal operations and is dispatched on demand when a stray animal is detected.

3.2. Edge Layer

A ruggedised edge server, co-located with the LoRaWAN gateway at the farm, hosts the core intelligence of the system:

- **MAS Coordinator:** orchestrates the agents according to the active herding goal, implements the BDI reasoning cycle, and dispatches commands to the RSD and SD.
- **Path Planner:** computes optimal herding routes over a graph representation of the pasture (see Section 5).
- **Stray Detector:** a software module *internal* to the Coordinator $\mathcal{C}$ that continuously evaluates the position of each animal relative to the herd centroid and raises a *stray alert* to $\mathcal{C}$ when an individual exceeds a configurable distance threshold d_{stray} . $\mathcal{C}$ then acts on this alert by dispatching the drone (see Section 6).
- **Alert Engine:** forwards alerts and video streams to the farmer’s mobile application.

Edge computing [38] is preferred over pure cloud processing to minimise latency for time-critical decisions (e.g., drone dispatch, sound emission timing).

3.3. Cloud / Application Layer

The cloud layer provides long-term data storage, analytics dashboards, and the mobile application through which the farmer defines herding goals, reviews animal health data, and receives live video feeds from the drone.

4. Agent Models

We adopt the BDI paradigm [8, 9] to model the cognitive architecture of the deliberative agents. Table 1 summarises all system participants and their roles.

Table 1

System participants in Smart Pastoralism 4.0.

Participant	Type	Primary Role
Livestock node ($\mathcal{L}_i$)	Reactive sensor node	Position & health telemetry
RSD ($\mathcal{S}$)	BDI	Herd driving & collecting
SD ($\mathcal{D}$)	BDI	Stray surveillance & alert
Farmer ($\mathcal{F}$)	Human-in-the-loop	Goal definition & oversight
Coordinator ($\mathcal{C}$)	BDI	MAS orchestration

4.1. Livestock Node ($\mathcal{L}_i$)

We include each instrumented animal $\mathcal{L}_i$ ($i = 1, \dots, N$) in the MAS as a *reactive sensor node* rather than a fully deliberative agent. Unlike the RSD and SD, $\mathcal{L}_i$ does not pursue goals autonomously, execute plans, or reason about its environment; it is not an agent in the classical sense of [6]. Nevertheless, treating it as a first-class participant in the MAS framework is a deliberate architectural choice: it provides a uniform message-passing interface, allows the Coordinator to apply the same belief-update logic to every entity in the field, and leaves room for future extensions in which collar firmware may incorporate lightweight reactive rules (e.g., a virtual-fence shock trigger [14]). Throughout the paper, the term *livestock node* is therefore preferred for $\mathcal{L}_i$, reserving *agent* for the deliberative entities ($\mathcal{S}$, $\mathcal{D}$, $\mathcal{C}$).

The belief set of $\mathcal{L}_i$ consists of its own GPS coordinates $\mathbf{p}_i(t) = (x_i(t), y_i(t))$ and biometric readings $\mathbf{h}_i(t)$. It has no explicit desires or intentions; it periodically broadcasts a telemetry message:

$$m_i(t) = \langle \text{id}_i, \mathbf{p}_i(t), \mathbf{h}_i(t), t \rangle \quad (1)$$

over the LoRaWAN uplink channel every T_{collar} seconds.

4.2. Robotic Shepherd Dog Agent ($\mathcal{S}$)

The RSD agent $\mathcal{S}$ is a full BDI agent:

- **Beliefs:** current position $\mathbf{p}_S(t)$; positions of all livestock $\{\mathbf{p}_i(t)\}$; herd centroid $\bar{\mathbf{p}}(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i(t)$; current herding goal; planned path waypoints; terrain map.
- **Desires:** fulfil the active herding goal (G1, G2, or G3).
- **Intentions:** execute the current plan, which alternates between *collecting* scattered animals and *driving* the cohesive herd along the planned path, following the Strömbom model [20].

Belief update and temporal consistency. The collar telemetry period $T_{\text{collar}} = 30$ s is deliberately longer than the RSD control cycle $T_{\text{ctrl}} = 5$ s in order to conserve battery life. This means that for most control iterations the RSD must act on *stale* position beliefs. The system handles this asynchrony in two complementary ways. First, when a fresh telemetry packet $m_i(t_k)$ arrives at the edge server, the Coordinator $\mathcal{C}$ immediately forwards updated waypoints and a refreshed herd-centroid estimate to $\mathcal{S}$, together with the current behaviour mode (COLLECT or DRIVE); it is $\mathcal{C}$ that retains decision authority over mode switching throughout the system’s operational lifetime (see Section 6). In the intermediate cycles the RSD retains the last mode command received and applies a lightweight *dead-reckoning* step solely to update its own *positioning target*: given the last observed centroid $\bar{\mathbf{p}}(t_k)$, the estimated current centroid is

$$\hat{\bar{\mathbf{p}}}(t) = \bar{\mathbf{p}}(t_k) + \bar{\mathbf{v}}(t_k) (t - t_k), \quad (2)$$

where $\bar{\mathbf{v}}(t_k)$ is the herd’s mean velocity estimated from the two most recent collar updates. The RSD uses $\hat{\bar{\mathbf{p}}}(t)$ *only* to recompute where to physically position itself (Equations 3–4) between Coordinator updates; it does not autonomously change its behaviour mode. This strategy is conservative: typical sheep

walking speeds (≤ 1.5 m/s [39]) imply a worst-case dead-reckoning error of at most $1.5 \times 30 = 45$ m after one full collar cycle—slightly exceeding the cohesion radius $d_{\text{coh}} = 40$ m. In the worst case, this may cause the mode-switching decision (COLLECT vs. DRIVE) to lag by one telemetry period (30 s). This is an accepted limitation of the current design: the RSD’s physical response time and audio influence radius are of the same order, so a single missed mode transition does not jeopardise overall herding progress. In practice, dead-reckoning errors are much smaller because herds move at sub-walking speeds when undisturbed, and any fresh telemetry within the cycle immediately resets the estimate.

The RSD implements two core behaviours:

Collecting. When the farthest animal from the herd centroid exceeds a cohesion threshold d_{coh} , the RSD positions itself on the far side of the outlier (relative to the centroid) and emits a high-intensity bark sound to push the animal back towards the group [36]. Formally, the RSD moves to position:

$$\mathbf{p}_S^{\text{collect}} = \bar{\mathbf{p}}(t) + (1 + \delta) (\mathbf{p}_{\text{far}}(t) - \bar{\mathbf{p}}(t)) \quad (3)$$

where $\mathbf{p}_{\text{far}}(t)$ is the position of the farthest animal and $\delta > 0$ is a small offset parameter ensuring the RSD is behind the outlier.

Driving. When all animals are within the cohesion radius, the RSD positions itself behind the herd (opposite to the desired direction of travel $\hat{\mathbf{d}}$) and emits a moderate-intensity whistle to propel the group forward:

$$\mathbf{p}_S^{\text{drive}} = \bar{\mathbf{p}}(t) - r_d \hat{\mathbf{d}} \quad (4)$$

where r_d is the driving distance behind the centroid and $\hat{\mathbf{d}}$ is the unit vector from the current centroid towards the next waypoint.

4.3. Surveillance Drone Agent ($\mathcal{D}$)

The SD agent $\mathcal{D}$ is event-driven:

- **Beliefs:** own position; stray animal position; battery level; obstacle map.
- **Desires:** provide aerial surveillance when a stray event occurs; stream video to the farmer.
- **Intentions:** upon receiving a STRAY_ALERT from the Coordinator, take off, navigate to the stray animal’s last known position, acquire visual lock, and begin streaming. If the stray animal returns to the herd, the SD returns to its charging station.

4.4. Farmer ($\mathcal{F}$)

The Farmer $\mathcal{F}$ is a *human-in-the-loop* participant rather than an automated agent. $\mathcal{F}$ interacts with the system exclusively through the mobile application provided by the Cloud/Application Layer:

- **Inputs to the MAS:** selection of a herding goal (G1, G2, or G3) and the associated parameters (target point P , origin A , destination B , or barn coordinates P_{barn}).
- **Outputs from the MAS:** push notifications on stray events, live video feed from the SD, real-time herd position on a map, and health telemetry summaries.
- **Intervention:** $\mathcal{F}$ may at any time override the active goal, recall the drone, or issue an emergency stop to the RSD.

The explicit modelling of $\mathcal{F}$ as a named participant in the MAS reflects the *supervisory control* pattern [6]: automation handles routine herding, while the human retains authority for exceptional situations and policy decisions.

4.5. Coordinator Agent ($\mathcal{C}$)

The Coordinator agent $\mathcal{C}$ runs on the edge server and acts as a mediator:

- It receives the herding goal from the Farmer $\mathcal{F}$.
- It invokes the Path Planner to compute the route (for G2 and G3).
- It monitors livestock positions and computes the herd centroid and dispersion.
- It issues navigation commands to $\mathcal{S}$ (waypoints, behaviour mode) and dispatch commands to $\mathcal{D}$.
- It evaluates goal completion: for G1, the herd remains within radius r_{hold} of the target; for G2 and G3, the centroid is within r_{goal} of the destination.

5. Path Planning

When the farmer issues a *Move* (G2) or *Return* (G3) goal, the system must compute an optimal path for the herd from origin A to destination B across the pasture. We model the pasture as a weighted graph and apply a variant of Dijkstra's algorithm [10] enriched with domain-specific cost functions.

5.1. Pasture Graph Construction

The pasture is discretised into a regular grid of cells, each of side length ℓ (e.g., 10 m). Each cell corresponds to a node $v \in V$ of the graph $G = (V, E)$. Edges connect each cell to its 8-connected neighbours (horizontal, vertical, diagonal). The weight of an edge $(u, v) \in E$ captures the traversal cost:

$$w(u, v) = \alpha \cdot d_{\text{geo}}(u, v) + \beta \cdot s(v) + \gamma \cdot o(v) + \eta \cdot f(v) \quad (5)$$

where:

- $d_{\text{geo}}(u, v)$ is the Euclidean distance between cell centres (accounting for diagonal moves: ℓ for cardinal, $\ell\sqrt{2}$ for diagonal).
- $s(v) \in [0, 1]$ is the normalised slope of cell v , penalising steep terrain that is difficult for livestock to traverse [39].
- $o(v) \in \{0, M\}$ is an obstacle penalty ($M \gg 1$ for impassable cells such as cliffs, water bodies, or fences).
- $f(v) \in [0, 1]$ is a forage-avoidance factor, reducing the attractiveness of cells with crops or sensitive vegetation.
- $\alpha, \beta, \gamma, \eta > 0$ are configurable weighting coefficients.

5.2. Dijkstra's Algorithm for Herding Paths

Given the weighted graph G , the path from node v_A (cell containing point A) to node v_B (cell containing point B) is computed via Dijkstra's algorithm [10, 40]. The algorithm maintains a priority queue of tentative distances and iteratively relaxes edges:

Complexity. Using a binary heap, the time complexity is $O(|E| \log |V|)$. For a grid of $n \times n$ cells, $|V| = n^2$ and $|E| = O(n^2)$, yielding $O(n^2 \log n)$. For typical pasture sizes (e.g., 500×500 m with $\ell = 10$ m, giving $n = 50$), this is computed in under one second on the edge server.

Alternative: A* Search. For very large pastures, the A* algorithm [41] can be used as a heuristic enhancement of Dijkstra's algorithm, employing the Euclidean distance to the goal as an admissible heuristic. This reduces the number of explored nodes while preserving optimality.

Algorithm 1 Terrain-Aware Herding Path (Dijkstra variant)

Require: Graph $G = (V, E)$, source v_A , destination v_B , weight function w

Ensure: Shortest path $\pi^* = (v_A, v_1, \dots, v_B)$

- 1: Initialise $dist[v] \leftarrow \infty$ for all $v \in V$; $dist[v_A] \leftarrow 0$
 - 2: Initialise priority queue Q with all $v \in V$
 - 3: $prev[v] \leftarrow \text{NIL}$ for all $v \in V$
 - 4: **while** $Q \neq \emptyset$ **do**
 - 5: $u \leftarrow \text{extract-min}(Q)$
 - 6: **if** $u = v_B$ **then**
 - 7: **break**
 - 8: **end if**
 - 9: **for all** neighbour v of u in G **do**
 - 10: $alt \leftarrow dist[u] + w(u, v)$
 - 11: **if** $alt < dist[v]$ **then**
 - 12: $dist[v] \leftarrow alt$; $prev[v] \leftarrow u$
 - 13: Decrease-key v in Q
 - 14: **end if**
 - 15: **end for**
 - 16: **end while**
 - 17: Reconstruct π^* by following $prev$ pointers from v_B to v_A
 - 18: **return** π^*
-

5.3. Path Smoothing and Waypoint Extraction

The raw grid path π^* is post-processed to extract a sequence of waypoints $W = (w_1, w_2, \dots, w_K)$ by applying a Douglas-Peucker simplification [42]. These waypoints are transmitted to the RSD agent, which uses them to set the driving direction $\hat{\mathbf{d}}$ at each step.

6. Multi-Agent Coordination Protocol

The coordination among agents follows a clearly defined protocol driven by the active herding goal. We describe the protocol for the most complex goal, G2 (*Move*), from which G1 and G3 are derived as special cases.

6.1. Goal G2: Move Herd from A to B

1. **Goal Reception.** The Farmer $\mathcal{F}$ submits goal G2 with parameters (A, B) via the mobile app. The Coordinator $\mathcal{C}$ receives the goal.
2. **Path Computation.** $\mathcal{C}$ invokes the Path Planner, which runs Algorithm 1 and returns waypoint sequence W .
3. **RSD Dispatch.** $\mathcal{C}$ sends the waypoint sequence and the initial behaviour mode (COLLECT or DRIVE) to the RSD agent $\mathcal{S}$.
4. **Herding Loop.** At each control cycle (period T_{ctrl}):
 - a) $\mathcal{C}$ updates the herd centroid $\bar{\mathbf{p}}(t)$ and the maximum deviation $d_{\text{max}}(t) = \max_i \|\mathbf{p}_i(t) - \bar{\mathbf{p}}(t)\|$.
 - b) If $d_{\text{max}}(t) > d_{\text{coh}}$: $\mathcal{C}$ instructs $\mathcal{S}$ to switch to COLLECT mode (Equation 3).
 - c) Else: $\mathcal{C}$ instructs $\mathcal{S}$ to switch to DRIVE mode towards the next waypoint (Equation 4).
 - d) Waypoint advancement: if $\|\bar{\mathbf{p}}(t) - w_k\| < r_{\text{wp}}$, advance to waypoint w_{k+1} .
5. **Stray Detection and Drone Dispatch.** If any animal i satisfies $\|\mathbf{p}_i(t) - \bar{\mathbf{p}}(t)\| > d_{\text{stray}}$ (with $d_{\text{stray}} > d_{\text{coh}}$):
 - a) $\mathcal{C}$ generates a STRAY_ALERT($\text{id}_i, \mathbf{p}_i$).

- b) $\mathcal{C}$ dispatches the drone $\mathcal{D}$ to position $\mathbf{p}_i$.
 - c) $\mathcal{D}$ takes off, navigates to $\mathbf{p}_i$, acquires the animal on camera, and initiates a live video stream to $\mathcal{F}$'s smartphone via 5G.
 - d) $\mathcal{F}$ receives a push notification with the video feed and can decide on manual intervention if needed.
 - e) When $\|\mathbf{p}_i(t) - \bar{\mathbf{p}}(t)\| \leq d_{\text{coh}}$ (animal reintegrated into the herd core), $\mathcal{D}$ is recalled to the charging station. Note that the recall threshold $d_{\text{coh}} = 40$ m is intentionally stricter than the dispatch threshold $d_{\text{stray}} = 80$ m; this *hysteresis gap* prevents chattering, whereby the drone would be recalled and re-dispatched repeatedly if the animal hovers near the stray boundary.
6. **Goal Completion.** When $\|\bar{\mathbf{p}}(t) - B\| < r_{\text{goal}}$, the Coordinator declares goal G2 achieved and notifies $\mathcal{F}$.

6.2. Goal G1: Hold Herd at Point P

Goal G1 is a degenerate case of G2 where $A = B = P$. The RSD maintains position around the herd and continuously operates in COLLECT mode whenever $d_{\text{max}}(t) > d_{\text{coh}}$. The path-planning step is skipped since no relocation is required.

6.3. Goal G3: Return Herd to Barn

Goal G3 is equivalent to G2 with $A = \bar{\mathbf{p}}(t_0)$ (current centroid at goal activation time t_0) and $B = P_{\text{barn}}$ (barn coordinates). The barn entrance may define a *funnel zone* where the RSD must slow down and emit specific calming sounds to guide animals through the gate without stress [37].

7. Communication Infrastructure

Reliable communication is critical for real-time coordination. The system employs a hybrid communication architecture combining LoRaWAN and cellular (4G/5G) networks.

7.1. LoRaWAN for Sensor Telemetry

LoRaWAN [34, 16] operates in the sub-GHz ISM bands (868 MHz in Europe) and provides ranges exceeding 10 km in rural line-of-sight conditions with very low power consumption [17]. This makes it ideal for periodic, low-bandwidth telemetry from livestock collars. Each collar transmits a compact packet (approximately 20 bytes: 4 bytes ID, 8 bytes GPS, 6 bytes biometrics, 2 bytes CRC) at a configurable spreading factor (SF7–SF12) to balance range and data rate. The LoRaWAN gateway at the farm forwards all uplink packets to the edge server via Ethernet.

The RSD also uses LoRaWAN for command reception and status reporting. Its data requirements are modest (waypoint updates, mode switches), well within the LoRaWAN duty-cycle constraints.

7.2. 5G/4G for Video Streaming and Low-Latency Control

The surveillance drone requires high-bandwidth, low-latency connectivity for real-time video streaming (typically 2–5 Mbps for 720p/1080p H.264). This exceeds LoRaWAN's capacity and is served by a 4G/5G SIM module. Where 5G coverage is available, network slicing can guarantee a dedicated quality-of-service (QoS) profile for the drone video stream [18, 19]. In areas lacking cellular coverage, the drone can store video locally and transmit upon returning within range, although real-time alerting is then degraded.

7.3. GNSS Considerations

All agents rely on Global Navigation Satellite System (GNSS) positioning. Multi-constellation receivers (GPS + Galileo + GLONASS) improve accuracy and availability, particularly in mountainous or forested terrain [33]. For the RSD and SD, Real-Time Kinematic (RTK) corrections, delivered via the cellular network, can achieve centimetre-level accuracy when required for precise manoeuvres near fences or barn entrances. The livestock collars use standard single-frequency GNSS, providing 2–5 m accuracy sufficient for herd-level tracking [13, 43].

8. Theoretical Validation

In this section we provide theoretical arguments for the correctness and convergence of the proposed algorithms, supplemented by preliminary simulation results.

8.1. Path Optimality

Proposition 1. *Algorithm 1 computes the minimum-cost path from v_A to v_B in the pasture graph $G = (V, E)$ with non-negative edge weights $w(u, v) \geq 0$ for all $(u, v) \in E$.*

Proof. This follows directly from the correctness of Dijkstra’s algorithm [10, 40]. Since all weight components in Equation (5) are non-negative ($d_{\text{geo}} \geq 0$, $s(v) \geq 0$, $o(v) \geq 0$, $f(v) \geq 0$, and all coefficients $\alpha, \beta, \gamma, \eta > 0$), the total edge weight is non-negative. Under this condition, Dijkstra’s algorithm is guaranteed to find the shortest path in $O(|E| \log |V|)$ time. $\square$

8.2. Herd Convergence under Shepherding

We analyse the convergence of the herding protocol under simplifying assumptions, following the theoretical framework of Strömbom et al. [20].

Assumption 1. *Each livestock node $\mathcal{L}_i$ moves according to:*

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{f}_{\text{flock}}(\mathbf{p}_i, \{\mathbf{p}_j\}) + \mathbf{f}_{\text{repel}}(\mathbf{p}_i, \mathbf{p}_S) + \boldsymbol{\epsilon}_i(t) \quad (6)$$

where $\mathbf{f}_{\text{flock}}$ is an attraction-alignment term towards nearby herd members [21, 25], $\mathbf{f}_{\text{repel}}$ is a repulsion from the shepherd (the animal moves away from the RSD when it is within the flight zone [36, 37]), and $\boldsymbol{\epsilon}_i(t)$ is a bounded noise term representing individual variability.

Proposition 2. *Under Assumption 1, if the RSD applies the collecting strategy (Equation 3) whenever $d_{\text{max}}(t) > d_{\text{coh}}$ and the driving strategy (Equation 4) otherwise, then for sufficiently large repulsion magnitude and bounded noise, the herd centroid $\bar{\mathbf{p}}(t)$ converges to the destination B along the waypoint sequence W .*

Proof sketch. The proof follows the structure of [20]. The collecting phase reduces d_{max} at each step because the RSD’s repulsive influence on the farthest animal pushes it towards the centroid. Once $d_{\text{max}} \leq d_{\text{coh}}$, the driving phase applies a net force on the entire herd in the direction $\hat{\mathbf{d}}$, advancing the centroid towards the next waypoint. Since noise is bounded and the repulsion magnitude exceeds the noise bound, progress towards the goal is monotonic in expectation. The waypoint sequence W obtained from Algorithm 1 is finite, and each waypoint is reached in finite expected time, so the destination B is reached in finite expected time. $\square$

8.3. Stray Detection Correctness

Proposition 3. *With telemetry period T_{collar} and stray threshold d_{stray} , a stray event is detected within at most T_{collar} seconds of the animal exceeding the threshold, assuming reliable LoRaWAN delivery.*

Proof. The Stray Detector evaluates $\|\mathbf{p}_i(t) - \bar{\mathbf{p}}(t)\|$ upon each telemetry reception. Since telemetry arrives every T_{collar} seconds, the worst-case detection latency equals T_{collar} . Reducing T_{collar} improves responsiveness at the cost of higher energy consumption and channel utilisation. $\square$

Table 2

Key simulation parameters.

Parameter	Symbol	Value
Grid cell size	ℓ	10 m
Number of animals	N	30
Telemetry period	T_{collar}	30 s
Control period	T_{ctrl}	5 s
Cohesion threshold	d_{coh}	40 m
Stray threshold	d_{stray}	80 m
Waypoint radius	r_{wp}	15 m
Goal radius	r_{goal}	20 m
Hold radius	r_{hold}	30 m
Driving offset	r_d	25 m
Noise bound	$\ \epsilon_i\ $	≤ 2 m/step

Table 3

Simulation comparison: proposed MAS vs. baselines (100 runs, Goal G2).

Configuration	Mean herding time (min)	Stray events / run	Success rate
Proposed MAS (full)	38.4 ± 6.2	2.1 ± 1.0	100%
Baseline 1 (Strömbom only)	54.7 ± 18.3	4.8 ± 2.3	89%
Baseline 2 (MAS, no drone)	43.1 ± 8.7	3.6 ± 1.6	100%

8.4. Preliminary Simulation

We implemented a discrete-time simulation of the MAS on a 500×500 m pasture grid ($\ell = 10$ m, $n = 50$) with $N = 30$ sheep, one RSD, and one SD. Animals follow the dynamics of Equation (6) with parameters calibrated from the literature [20, 36]. Table 2 lists the key simulation parameters.

Results. Over 100 Monte Carlo runs of Goal G2 (relocation from corner (50, 50) to (450, 450)), the mean herding time was 38.4 ± 6.2 minutes, with a 100% success rate. On average, 2.1 stray events per run triggered drone dispatch. The path planner consistently found routes avoiding steep slopes and obstacles, with a mean path length 12% longer than the Euclidean distance—a reasonable detour cost for terrain safety.

Comparison with a baseline. To evaluate the benefit of the proposed MAS coordination, we compare against two baselines run on the same 100 scenarios. **Baseline 1 (Strömbom only)** implements the pure heuristic shepherding model of [20] without BDI reasoning, a Coordinator, or a Surveillance Drone: the RSD alternates collect/drive based solely on the cohesion check, with no terrain-aware path planning (straight-line driving towards the goal). **Baseline 2 (MAS without drone)** is identical to the proposed system but the SD is disabled; stray events are still logged but no aerial response is issued.

Table 3 summarises the results. The full MAS achieves the shortest mean herding time and the highest success rate. Baseline 1 struggles in the presence of terrain obstacles (simulated as impassable grid cells): in 11 of 100 runs the herd reached a dead-end and the run was aborted (hence the lower success rate). Baseline 2 succeeds in all runs but accumulates more stray time because unrecovered stragglers slow the overall herd advance until the RSD repositions. These results confirm that terrain-aware path planning and drone-assisted stray recovery each provide measurable gains over simpler alternatives.

These preliminary results demonstrate the feasibility and comparative advantage of the proposed MAS approach.

9. Discussion

The proposed architecture leverages the complementary strengths of ground and aerial robots within a unified MAS framework. The RSD provides continuous, close-range herding influence, while the SD offers rapid, wide-area situational awareness. This combination mirrors traditional pastoral practice, where the shepherd dog works the flock while the human overseer maintains a broader view [44, 37].

Several challenges remain for real-world deployment. First, the robustness of LoRaWAN communication in densely vegetated or mountainous terrain requires field validation [17]. Second, the RSD’s audio-based herding presumes that livestock respond predictably to sound stimuli; behavioural variability across species and individuals must be accounted for through adaptive sound profiles. Third, regulatory constraints on drone flights in agricultural areas, particularly concerning altitude, line-of-sight requirements, and airspace coordination, vary by jurisdiction and must be addressed.

Towards greater decentralisation. The current architecture delegates most decision-making to the centralised Coordinator running on the edge server. While this design simplifies reasoning and keeps the field agents lightweight, it raises a legitimate concern: a single point of orchestration may become a bottleneck or a single point of failure in scenarios with intermittent connectivity or very large herds. A more decentralised variant is worth considering as a future evolution. In such a variant, each RSD could maintain its own local BDI state and autonomously trigger collect/drive transitions based on the collar data it receives directly (via LoRaWAN), without waiting for Coordinator commands—reserving the Coordinator only for global goal management and path planning. Similarly, once dispatched, the SD could autonomously track a stray animal using onboard computer vision, reporting back to the Coordinator only when the animal has been recovered or the battery is low. A fleet of multiple RSDs could coordinate peer-to-peer through a lightweight negotiation protocol (e.g., a Contract Net [30]) to divide responsibility over different flock sub-groups. This progression from centralised orchestration to a genuinely decentralised MAS represents a natural and compelling direction for future work.

The use of the BDI architecture ensures that the system can be extended with additional agent types (e.g., a second RSD for very large herds, or a perimeter patrol drone) without redesigning the coordination protocol. The potential field approach [45] could complement the current graph-based path planner to handle dynamic obstacles encountered during herding.

Future work will focus on: (i) full-scale field trials with real animals and commercial robots; (ii) integration of computer vision on the RSD and SD for individual animal identification and anomaly detection; (iii) machine learning for adaptive parameterisation of herding behaviours based on herd response; and (iv) formal verification of the MAS coordination protocol using model-checking tools.

10. Conclusion

We have presented Smart Pastoralism 4.0, a Multi-Agent System architecture for intelligent livestock management on open pastures. The system integrates IoT-enabled animal collars, a robotic shepherd dog, and a surveillance drone within a BDI-based MAS framework coordinated by an edge server. Three herding goals—hold, move, and return—are supported through a combination of terrain-aware path planning (Dijkstra’s algorithm with domain-specific cost functions) and a herding coordination protocol inspired by biological herding heuristics. Theoretical analysis confirms path optimality and herd convergence, while preliminary simulations demonstrate the practical viability and comparative advantage of the approach.

The proposed architecture directly addresses several limitations identified in the related work. First, most Precision Livestock Farming solutions reviewed in Section 2 are *passive*: they collect sensor data for offline analysis but do not actuate real-time herding behaviour. Smart Pastoralism 4.0 closes this loop by coupling IoT telemetry with active robotic control. Second, existing PLF deployments are predominantly designed for indoor or semi-intensive settings; our system explicitly targets the *extensive* open-pasture scenario, with terrain-aware path planning and long-range LoRaWAN communication

suited to rural environments. Third, while UAVs have been applied individually to livestock counting and monitoring [28], their systematic integration as a reactive agent within a MAS herding framework had not previously been demonstrated; the SD agent fills this gap by providing aerial situational awareness triggered by the stray-detection mechanism. Finally, the shepherding literature [20, 23] has largely treated the shepherd as an isolated controller; our BDI Coordinator enables goal-directed, multi-robot coordination that extends naturally to larger fleets, as discussed in Section 9.

Smart Pastoralism 4.0 thus represents a step towards fully autonomous, welfare-conscious livestock management, bridging the gap between precision livestock farming and multi-agent robotics.

Declaration on Generative AI

During the preparation of this work, the author(s) used AI-based tools for grammar and spelling verification. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] D. Berckmans, Precision livestock farming technologies for welfare management in intensive livestock systems, *Revue Scientifique et Technique de l'OIE* 33 (2014) 189–196. doi:10.20506/rst.33.1.2273.
- [2] A. Herlin, E. Brunberg, J. Hultgren, N. Högborg, A. Rydberg, A. Skarin, Animal welfare implications of digital tools for monitoring and management of cattle and sheep on pasture, *Animals* 11 (2021) 829. doi:10.3390/ani11030829.
- [3] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, M. N. Hindia, An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges, *IEEE Internet of Things Journal* 5 (2018) 3758–3773. doi:10.1109/JIOT.2018.2844296.
- [4] A. Tzounis, N. Katsoulas, T. Bartzanas, C. Kittas, Internet of things in agriculture, recent advances and future challenges, *Biosystems Engineering* 164 (2017) 31–48. doi:10.1016/j.biosystemseng.2017.09.007.
- [5] C. Aquilani, A. Confessore, R. Bozzi, F. Sirtori, C. Pugliese, Review: Precision livestock farming technologies in pasture-based livestock systems, *Animal* 55 (2015) 100429. doi:10.1016/j.animal.2021.100429.
- [6] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed., John Wiley & Sons, 2009.
- [7] A. Dorri, S. S. Kanhere, R. Jurdak, Multi-agent systems: A survey, *IEEE Access* 6 (2018) 28573–28593. doi:10.1109/ACCESS.2018.2831228.
- [8] A. S. Rao, M. P. Georgeff, BDI agents: From theory to practice, in: *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, 1995, pp. 312–319. doi:10.1007/978-3-662-03678-5_8.
- [9] M. E. Bratman, *Intention, Plans, and Practical Reason*, volume 100, JSTOR, 1987. doi:10.2307/2185304.
- [10] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik* 1 (1959) 269–271. doi:10.1007/BF01386390.
- [11] S. Neethirajan, Recent advances in wearable sensors for animal health management, *Sensing and Bio-Sensing Research* 12 (2017) 15–29. doi:10.1016/j.sbsr.2016.11.004.
- [12] R. N. Handcock, D. L. Swain, G. J. Bishop-Hurley, K. P. Patison, T. Wark, P. Valencia, P. Corke, C. J. O'Neill, Monitoring animal behaviour and environmental interactions using wireless sensor networks, GPS collars and satellite remote sensing, *Sensors* 9 (2009) 3586–3603. doi:10.3390/s90503586.
- [13] D. L. Swain, M. A. Friend, G. J. Bishop-Hurley, R. N. Handcock, T. Wark, Tracking livestock using global positioning systems—are we still lost?, *Animal Production Science* 51 (2011) 167–175. doi:10.1071/AN10255.

- [14] D. M. Anderson, R. E. Estell, J. L. Holechek, S. Ivey, G. B. Smith, Virtual herding for flexible livestock management – a review, *The Rangeland Journal* 36 (2014) 205–221. doi:10.1071/RJ13092.
- [15] C. Umstätter, The evolution of virtual fences: A review, *Computers and Electronics in Agriculture* 75 (2011) 10–22. doi:10.1016/j.compag.2010.10.005.
- [16] A. Augustin, J. Yi, T. Clausen, W. M. Townsley, A study of LoRa: Long range & low power networks for the Internet of Things, *Sensors* 16 (2016) 1466. doi:10.3390/s16091466.
- [17] M. Cattani, C. Boano, K. Römer, An experimental evaluation of the reliability of LoRa long-range low-power wireless communication, *Journal of Sensor and Actuator Networks* 6 (2017) 7. doi:10.3390/jsan6020007.
- [18] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, G. Wunder, 5G: A tutorial overview of standards, trials, challenges, deployment, and practice, *IEEE Journal on Selected Areas in Communications* 35 (2017) 1201–1221. doi:10.1109/JSAC.2017.2692307.
- [19] M. Marchese, A. Moheddine, F. Patrone, Iot and uav integration in 5g hybrid terrestrial-satellite networks, *Sensors* 21 (2019) 3704. doi:10.3390/s19173704.
- [20] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. T. Sumpter, A. J. King, Solving the shepherding problem: heuristics for herding autonomous, interacting agents, *Journal of the Royal Society Interface* 11 (2014) 20140719. doi:10.1098/rsif.2014.0719.
- [21] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH Computer Graphics* 21 (1987) 25–34. doi:10.1145/37402.37406.
- [22] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, S. Cameron, Experiments in automatic flock control, *Robotics and Autonomous Systems* 25 (1998) 195–211. doi:10.1016/S0921-8890(98)00050-6.
- [23] J.-M. Lien, O. Bayazit, R. Sowell, S. Rodriguez, N. Amato, Shepherding behaviors, in: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 4159–4164 Vol.4. doi:10.1109/ROBOT.2004.1308924.
- [24] K. Fujioka, S. Hayashi, Effective shepherding behaviours using multi-agent systems, in: *Proceedings of the IEEE Region 10 Conference (TENCON)*, 2016, pp. 3179–3182. doi:10.1109/tencon.2016.7848636.
- [25] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, *Physical Review Letters* 75 (1995) 1226–1229. doi:10.1103/PhysRevLett.75.1226.
- [26] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, *IEEE Transactions on Automatic Control* 51 (2006) 401–420. doi:10.1109/TAC.2005.864190.
- [27] L. Gonzalez, G. Montes, E. Puig, S. Johnson, K. Mengersen, K. Gaston, Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation, *Sensors* 16 (2016) 97. doi:10.3390/s16010097.
- [28] B. Xu, W. Wang, G. Falzon, P. Kwan, L. Guo, Z. Sun, C. Li, Livestock classification and counting in quadcopter aerial images using Mask R-CNN, *International Journal of Remote Sensing* 41 (2020) 8121–8142. doi:10.1080/01431161.2020.1734245.
- [29] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.
- [30] B. P. Gerkey, M. J. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, *The International Journal of Robotics Research* 23 (2004) 939–954. doi:10.1177/0278364904045564.
- [31] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, *Swarm Intelligence* 7 (2013) 1–41. doi:10.1007/s11721-012-0075-2.
- [32] F. L. Bellifemine, G. Caire, D. Greenwood, *Developing Multi-Agent Systems with JADE*, John Wiley, 2007.
- [33] E. D. Kaplan, C. J. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd ed., Artech House, 2017.
- [34] A. Yegin, T. Kramp, P. Dufour, R. Gupta, R. Soss, O. Hersent, D. Hunt, N. Sornin, Lorawan protocol: specifications, security, and capabilities, *LPWAN Technologies for IoT and M2M Applications*

- (2015) 37–63. doi:10.1016/b978-0-12-818880-4.00003-x, version 1.0.
- [35] Boston Dynamics, Spot: The agile mobile robot, Boston Dynamics Technical Report (2023). <https://www.bostondynamics.com/spot>.
 - [36] G. D. Hutson, 'flight distance' in Merino sheep, *Animal Production* 35 (1982) 231–235. doi:10.1017/S0003356100027343.
 - [37] T. Grandin, *Improving Animal Welfare: A Practical Approach*, 2nd ed., CABI, 2015.
 - [38] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, *IEEE Internet of Things Journal* 3 (2016) 637–646. doi:10.1109/JIOT.2016.2579198.
 - [39] R. J. Kilgour, K. Uetake, T. Ishiwata, G. J. Melville, The behaviour of beef cattle at pasture, *Applied Animal Behaviour Science* 138 (2012) 12–17. doi:10.1016/j.applanim.2011.12.001.
 - [40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd ed., MIT Press, 2009.
 - [41] P. Hart, N. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics* 4 (1968) 100–107. doi:10.1109/TSSC.1968.300136.
 - [42] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (1973) 112–122. doi:10.3138/FM57-6770-U75U-7727.
 - [43] B. C. Fargas, M. N. Petersen, GPS-free geolocation using LoRa in low-power WANs, in: *Proceedings of the Global Internet of Things Summit (GloTS)*, 2017, pp. 1–6. doi:10.1109/GIOTS.2017.8016251.
 - [44] C. M. Dwyer, Welfare of sheep: providing for welfare in an extensive environment, *Small Ruminant Research* 86 (2009) 14–21. doi:10.1016/j.smallrumres.2009.09.010.
 - [45] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research* 5 (1986) 90–98. doi:10.1177/027836498600500106.