Research Assistant at ZHAW School of Engineering

Technical Skills Assessment

Ding-wei Hu

Taoyuan, Taiwan  
+886 925391057

dingweihu.contact@gmail.com

**ABSTRACT**

For wildlife management and agriculture, animal behavior analysis techniques are crucial for early detection of disease, roaming predators or nutritional deficiencies in remote herds. This paper proposes the initial research steps for a model simulating animal movement on the example of cows. It involves the development of software tools for generating and simulating the positioning movement data of cows converted into World Geodetic System (WGS) as well as calculating the average cow trajectory based on a data transformation system. The method used consist in applying mathematical modeling and algorithmic techniques to the movement data streams to recognize patterns and detecting anomalies over various timeframes.

**CCS Concepts**

• **Information systems➝Database management system engines;** • **Computing methodologies➝Massively parallel and high-performance simulations.**

**Keywords**

Animal behaviour analysis; Cow’s movement algorithm; Data transmission; GPS system; Kubernetes cluster.

# INTRODUCTION

The evolution of the internet from a research exchange tool to a technique connecting humans with their environment reflect the fundamental transformation, which integrating advanced technologies such as AIoT[1][2], and data analytics into our life will bring. A glimpse of that development is already visible in the field of (smart) agriculture. IoT-based agriculture enhances efficiency and sustainability of agricultural practices by connecting physical and virtual elements of the agricultural system with the internet, allowing for comprehensive monitoring, management and information exchange. One crucial application of agricultural IoT solutions consists in analyzing and monitoring animal behaviors, in particular livestock (herding), providing deeper insights into animal behaviors. This requires research on tracking animals, machine learning from regular movement, training custom models to capture the contextualized movement patterns, and detection of anomalies (irregular movement) both short-term and over longer periods such as a full day.

The aim of this paper is simulating cows’ trajectories in view of developing cloud-based solutions that will provide farmers with timely alerts and information allowing them to monitor and manage the health and safety of their livestock. This is significant as it contributes to improving animals’ life as well as food quality and security overall.

Based on the assignment, the first research question is how to design a simplified algorithm both generating and simulating how and how fast cows move by using generated positioning movement data in x/y coordinates that can be converted into WGS. To answer it, an algorithm will be developed that first transforms the coordinates before generating the movement of cows. Based on the designed algorithm, we search to know in a second step how to develop the prototypes of two tools: the first allows producing movement of cows, the second enables calculating the average trajectory of their movements based on a data transformation system. A server tool for simulating concurrent animal movement, a client tool for calculating metrics from the movement data stream together with the mathematical formulation of metrics will be developed. The research question of subtask 3 seeks to develop a comparison of behaviors of Horizontal Pod Autoscaler (HPA) and Vertical Pod Autoscaler (VPA)

# IMPLEMENTATION

## Subtask 1: Movement Generation

This two-week technical skills assessment has been carried out as part of the application procedure for the position of research assistant in industry projects related to distributed systems for animal tracking and similar IoT/edge-cloud continuum applications, optionally linked with a cooperative doctorate (3-years PhD program), at the Zurich University of Applied Sciences (ZHAW) School of Engineering, Switzerland.

Start date: 02.07.2024.

Submission:16.07.2024.

Github: https://github.com/HuDingWei/Technical-Skills-Assessment-Task---ZHAW-Research-Assistantship.git

To generate a realistic and simplified animal movement trajectory on a plain field, we design an algorithm inspired by Laube and Purves' study, "How fast is a cow?" (2011)[3]. The algorithm produces a series of x/y coordinates, which are then translated into WGS84 longitude/latitude pairs to create realistic movement traces. This section describes the complete idea and process including coordinate transformation, movement generation along with the mathematical foundations and constraints applied.

### Coordinate Transformation

The algorithm begins by defining the number of movement points (num\_points) and setting the origin coordinates in WGS84 format (latitude, longitude). For example, the x/y coordinates set as (0, 0) with the origin be set to the geographical coordinates of Zurich (latitude: 47.3769, longitude: 8.5417). These coordinates are crucial as they provide the starting point for the movement simulation match to the real-world.

Based on the Pyproj library, the geographical coordinates can be converted from WGS84 to UTM (Universal Transverse Mercator). The WGS84 to UTM transformation function is used and shown as:

|  |  |
| --- | --- |
|  | (1) |

where transformer\_to\_utm is the transformation function from WGS84 to UTM zone 32N.

The geographical coordinates can be converted from UTM to WGS84. The UTM to WGS84 transformation function is used and shown as:

where transformer\_to\_latlon is the transformation function from UTM zone 32N to WGS84.

This transformation is necessary because UTM coordinates provide a more straightforward Cartesian coordinate system for calculating movements.

### Movement Generation

The core of the algorithm lies in generating movements based on random steps while constraining the turning angles to simulate more realistic animal behavior. For each movement step, a random distance is generated within a specified range (e.g., 0 to 0.3 meters). The direction () is adjusted within a limited angle to prevent abrupt changes in direction, which is consistent with natural movement patterns.

The direction adjustment is controlled by allowing a turning angle of up to ±15 degrees per step, and the cumulative turning angle is restricted to ±90 degrees over any sequence of six steps. This constraint is necessary as the animal does not make unrealistic sharp turns frequently. The equations for calculating the new position are shown as:

|  |  |
| --- | --- |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |

where 𝜃 represents the adjusted direction angle. and represent the changes in the x and y coordinates for each step, respectively.

These calculations are performed in UTM coordinates to ensure accuracy in meters, and then transformed back to WGS84 format for real-world applicability. By integrating cosine and sine functions, the algorithm effectively simulates both random and realistic animal movements, maintaining constraints within natural turning angles.

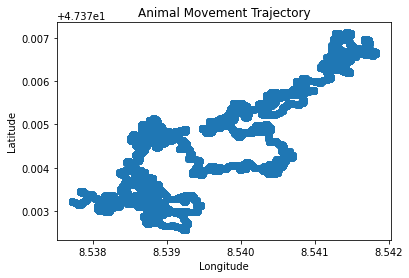


Figure 1. Example of Animal Movement Trajectory.

This step-by-step process not only generates accurate movement traces but also provides insights into the dynamics of animal behavior within controlled environments.

## Subtask 2: Concurrent Movement Simulation and Metrics Calculation

The second subtask involves developing two tools: one for simulating concurrent animal movements and another for calculating metrics from the movement data stream, which aims at simulating concurrent animal movements and calculating metrics from the movement data stream. This section elaborates on the design and implementation of these tools, ensuring they operate effectively in a concurrent environment.

### Server Tool (Movement Simulation)

|  |  |
| --- | --- |
|  | (2) |

The server-side tool is designed to simulate animal movements using a multi-threaded approach. It is responsible for simulating concurrent animal movements based on the previously described algorithm. It initializes a server to listen for incoming connections and spawns a thread for each animal to generate movement data. Each thread follows the same movement generation process, continuously sending the generated movement data (animal\_id, longitude, latitude) over the network. The server setup and threading mechanism ensure that multiple animals can be simulated simultaneously without performance degradation.

### Client Tool (Metrics Calculation)

The client tool connects to the server to receive the movement data stream. It is designed to handle concurrent data reception and metrics calculation efficiently. The client continuously receives movement data, parses it, and calculates metrics such as the average distance traveled by each animal. The use of threading ensures that data reception and metrics calculation occur simultaneously, providing real-time insights into animal movements.

### Mathematical Formulation for Metrics

|  |  |
| --- | --- |
|  | (7) |

A metric calculated by the client is the distance traveled by each animal. This metric was derived using the Euclidean distance formula between consecutive points in an animal's movement trajectory:

where and represent the changes in coordinates (x and y) between consecutive points of an animal's movement trajectory. The formula sums up these distances over the trajectory's entire length (denoted by 𝑛), providing a comprehensive measure of the spatial extent covered by each animal over time.

# RESULTS AND DISCUSSION

The movement generation algorithm successfully produced realistic animal trajectories using Zurich's coordinates (latitude: 47.3769, longitude: 8.5417). The algorithm employed constraints on the turning angle (±15 degrees per step, cumulative ±90 degrees over six steps) and random step sizes to mimic natural animal movement patterns as shown in Figure 1. The generated trajectories, visualized with matplotlib, demonstrated smooth and realistic paths, adhering to the constraints.

The server tool effectively simulated the concurrent movements of multiple animals, with each animal generating movement data asynchronously. Threading allowed the server to handle multiple animals simultaneously, and data was transmitted as JSON-encoded longitude and latitude pairs. The client tool receives this data stream, calculating real-time metrics such as average distance traveled by each animal.

While Subtask 1 and 2 of the technical skills assignment have been successfully completed, the implementation of Subtask 3 involving deploying the software to a local Kubernetes cluster has not yet been realized. The reasons are that we found it challenging to set up the appropriate environment for Kubernetes architecture. Further study about building a container orchestration system is required to solve Subtask 3. Based on the background research so far, multiple tools will have to be studied for setting up the appropriate environment for working on the Kubernetes cluster. In addition, I plan to deepen my expertise about Kubernetes and Docker to solve Subtask 3.

It is further worth mentioning that several publications provide an overview of state-of-the-art agricultural IoT technology include sources from China[1][2]. However, the reviews conducted contain a gap regarding related and relevant research carried out in Taiwan. The focus of Taiwanese researchers in the field of smart agriculture has been on husbandry over the past years.

# CONCLUSION AND FUTURE DIRECTIONS

Subtasks 1 has proven the feasibility of designing a simplified algorithm generating and simulating animal positioning movement data in x/y coordinates that can be translated into WGS on the example of cows. In completing Subtask 2, we demonstrate that the prototypes manage to produce and calculate the average distance of animal movements based on a data transformation system. These achievements establish a solid groundwork for continuing future development and deployment in Subtask 3.

Future work will focus on overcoming the challenges encountered during the implementation of Subtask 3. This will involve gaining expertise in Docker and Kubernetes to efficiently package and deploy the server and client applications. Setting up a robust local Kubernetes cluster will be prioritized, along with exploring cloud-based alternatives to enhance scalability and reliability. Additionally, implementing and testing load balancing and autoscaling mechanisms will be crucial to handle large-scale data generation, ensuring the system can manage increased loads effectively. These steps will address the limitations encountered and advance the project towards its full potential.

# REFERENCES

1. Xu, J., Gu, B. and Tian, G. 2022. Review of agricultural IoT Technology. In *Artificial Intelligence in Agriculture*. Vol. 6: 10-22. DOI= <https://doi.org/10.1016/j.aiia.2022.01.001>
2. Farooq, M.S., Riaz, S., Abid, A., Umer, T., and Zikria, Y.B. 2020. Role of IoT Technology in Agriculture: A Systematic Literature Review. *Electronics*. Vol. 9(2):319. DOI= <https://doi.org/10.3390/electronics9020319>.
3. Laube, P. and Purves, R.S. 2011. How fast is a cow? Cross-Scale Analysis of Movement Data. *Transactions in GIS*. Vol.15: 401-418.