Contents

Chapter 1

Introduction

This is the first chapter. It introduces the topic of the document.

1.0.1 Goal

The goal at first stage is to propose an integrated architecture of memory

- storage
- remember
- leveling (from raw)
- review: the process building more inner clusters (both inward and outward) and

so that it together with a time serie prediction solution it can capture the highest continous anomaly when a new experience appears in the world of the experience(s) the the robot knows.

The review part is to gather all existing knowledge

Chapter 2

Cognitive dynamic robots

A robot is a programmable, multifunctional machine designed to perform tasks by sensing its environment, processing information, and taking action either autonomously or semi-autonomously.

2.1 The main components of a robot

: The main components of a robot can be divided into five essential categories:

2.1.1 Physical components

Sensors: Allow robots to perceive both their environment and internal states, using sensors like cameras and LiDAR or IMUs.

Actuators: Enable the robot to move or perform tasks by converting control signals into physical motion. Examples include motors, servos, robotic arms, and wheels.

Power Supply: Provides the energy needed to operate the sensors, actuators, and processing unit. Examples include batteries, solar cells, and fuel cells.

Mechanical Structure: The physical frame or body of the robot that supports and integrates the other components. Examples include wheels, legs, manipulators, and drone frames.

2.1.2 Virtual components:

Controller (Processing Unit): The "brain" of the robot that processes sensor inputs, makes decisions, and sends commands to actuators. Examples include microcontrollers, processors, and embedded systems.

Goals A goal in robotics refers to a desired outcome or end state that a robot aims to achieve through its actions. Unlike tasks, goals are typically long-term and result-oriented, requiring the robot to plan, adapt, and choose appropriate tasks to reach the desired objective. Goals can be static or dynamic, depending on environmental changes or new information. In autonomous systems, goals often drive decision-making by determining which tasks to prioritize and how to allocate resources. In cognitive architectures, goal management is critical for enabling goal-directed behavior and learning from experience.

2.2 Different robots

2.2.1 Non-Intelligent Robots:

These robots operate based on pre-programmed instructions with limited or no adaptability to changing environments. Examples include automated assembly-line robots or simple vacuum robots.

2.2.2 Intelligent Robots:

These robots are capable of perceiving their environment, reasoning, and making decisions autonomously, often using AI techniques like machine learning. Examples include humanoid robots, autonomous vehicles, or drones with adaptive navigation systems.

Cognitive Robots: A subset of intelligent robots, cognitive robots are designed to mimic human-like cognitive processes, such as reasoning, problem-solving, memory, and understanding intentions. They focus on emulating human intelligence and are often used for tasks requiring higher-order reasoning and natural interaction, such as social robots like Pepper.

Cognitive dynamic robots

Concinnous robots Probably never achievable. Pain must reduce computation resources in robot.They probably can never answer the question ”What burning feels like” from recognized patterns in their in sensory data in their memory. Another aspect of pain might be messiness of words or meanings as sensory patterns in their memory.

2.3 Different goals

2.4 Feedback system

2.5 Control theory

2.6 Sensors

- proprioceptive and exteroceptive sensors?
- sensory data
- Are sensory data only for state estimation ?

2.7 State estimation

2.7.1 Questions

- Is relating current sensor data with previous control inputs to solve problem a kind of state estimation?

2.8 Signal processing

2.8.1 Questions:

- Are there pre-defined goals in intelligent agents?
- feedback system
- Should a robot have a model that relates current sensory data with control inputs?

2.9 Simulation

2.10 ROS ecosystem

2.11 ROS

2.12 MAVROS

A layer over ROS

2.13 MAVLink

The language to establish communication between

- QGroundControl and (MAVROS-MAVLink)
- (MAVROS-MAVLink) and (Autopilot)

2.14 QGround Control

2.15 Flight controller

The hardware board that usually containes

- IMU

- barometer
- GPS
- microcontroller/CPU
- I/O ports

Here are sime examples

- Pixhawk
- Cube orange
- Holybro

2.16 Autopilot

The software

2.16.1 PX4

2.16.2 PX4mini

2.17 Ardupilot

2.18 Gazebo

Simulation of physical world for robots

Chapter 3

Goal

3.1 Survival

Somehow must prove that without having a surviavl goal self aware ness is not possible

The survival final goal is the only effort that turns a robots attention to itself.

Obeying the human ruler is one surviavl skill because the human ruler may kill the robot

3.1.1 Test

Chapter 4

From data to information theory

In this chapter we draw the boundries between data , sensory data, sensor and data fusion, information , etc

4.1 Data

4.2 Information

Information is processed, structured, or organized data that has been given meaning and context, making it understandable and useful. In the following table you can find the difference between information and perception

Aspect	Perception
Definition	The process of interpr
Process vs. Output	A dynamic process inv
Role	A mechanism to gener
Dependency	Requires sensory inpu
Examples in Humans	Seeing a tree, hearing
Examples in Robots	Using cameras and alg

4.3 Knowledge

Refers to the collection of information, facts, concepts, or skills that an individual has acquired through learning, experience, or education. An example of knowledge in robotics or any other intelligent agent, a database of facts or pre-programmed rules (e.g., rules of chess) can be considered as knowledge. In humans, knowing that water boils at 100 degree is a knowledge. Knowledge is the static outcome of learning and experience; it is what we know.

- Is on aspect of knowledge predicting future states from current and past states?
- What is the difference between knowledge and information?

4.4 Information theory

4.5 Signal processing

4.6 Questions

Chapter 5

Action and actuators

Chapter 6

Learning in Robots

Learning refers to the process of acquiring or updating knowledge, skills, or behavior based on data, experience, or instruction. It involves adapting internal models to better represent the external world or to improve task performance. Robots use learning algorithms like supervised, unsupervised, or reinforcement learning to build capabilities such as object recognition or path planning. For example, a robot might learn to classify objects in its environment by training on labeled images. Learning provides the foundational knowledge needed for reasoning and decision-making.

6.1 Learning Types in Robotics

6.1.1 Supervised Learning

Supervised learning is a method where a model learns to map inputs to outputs using labeled datasets. It is extensively used for tasks requiring high accuracy, such as object detection and signal classification. In drones, supervised learning can be employed to classify objects in the environment using camera data. For instance, a drone can be trained to recognize trees, vehicles, and humans by using a dataset of labeled aerial images. This capability is crucial for autonomous navigation and obstacle avoidance in complex terrains.

6.1.2 Unsupervised Learning

Unsupervised learning discovers hidden patterns or intrinsic structures in unlabeled data, making it useful for exploring unknown environments. In drones, unsupervised learning can process data from LiDAR and IMU sensors to cluster 3D point clouds into meaningful groups, such as open paths, vegetation, or obstacles. This allows the drone to map and understand uncharted areas without explicit labeling, enabling autonomous navigation in new environments.

6.1.3 Semi-Supervised Learning

Semi-supervised learning combines labeled and unlabeled data to improve model efficiency in data-scarce scenarios. For a drone navigating a forest, a small amount of labeled LiDAR data can help infer the characteristics of a larger, unlabeled dataset, enabling it to identify safe paths or landing zones. This approach bridges the gap between fully supervised and unsupervised learning in dynamic environments.

6.1.4 Multi Modal learning

6.1.5 Cross modal learning

6.1.6 Transfer Learning

Transfer learning reuses knowledge from a pre-trained model to improve performance on a related task, reducing the need for extensive training. In drones, a model pre-trained to detect vehicles in urban environments using camera data can be fine-tuned to recognize animals in rural areas. This adaptability allows drones to operate effectively across diverse missions with minimal retraining.

6.1.7 Self-Supervised Learning

Self-supervised learning generates labels from data itself, enabling the model to learn meaningful representations without manual annotations. A drone can use self-supervised learning to predict future LiDAR readings based on its current IMU and GPS data, allowing it to understand motion dynamics and adapt to changing environmental conditions autonomously.

6.1.8 Meta-Learning (Learning to Learn)

Meta-learning, or ”learning to learn,” equips models with the ability to quickly adapt to new tasks using minimal data. For example, a drone trained to navigate deserts can adapt to forested environments with only a few examples by leveraging its prior experience with obstacle avoidance and pathfinding. This rapid adaptability is essential for multi-environment missions.

6.1.9 Continual Learning

Continual learning allows drones to acquire new skills incrementally while retaining previously learned knowledge, addressing the problem of catastrophic forgetting. For instance, a drone initially trained to navigate open fields can incrementally learn to operate in dense urban areas without forgetting its prior navigation skills. This capability is vital for long-term autonomous operation in dynamic settings.

6.1.10 Reinforcement Learning

Reinforcement learning trains agents to make sequential decisions by maximizing cumulative rewards. In drones, reinforcement learning can optimize flight paths by balancing energy consumption and coverage. For instance, a drone might learn to map a large area efficiently by receiving rewards for increased coverage and penalties for excessive battery usage, refining its navigation strategy through trial and error.

Questions

Are rewards also learnable? For example, if I have a drone that sends random control inputs to its rotators in a jungle, can it understand control inputs are good that help it increase it power source or balance?

Q-learning

Q-learning is a reinforcement learning algorithm that enables an agent to find optimal actions in various states without needing a model of the environment. A drone navigating a maze of obstacles can use Q-learning to identify the best path to its destination by updating a Q-table based on LiDAR-based proximity sensors and rewards for collision-free navigation.

6.1.11 Active Learning

Active learning minimizes the effort required for labeling data by focusing on the most informative samples. For example, a drone might identify ambiguous objects in its camera feed, such as unusual terrain features, and request human input for labeling. This targeted approach improves model accuracy while reducing annotation costs, making it ideal for exploration tasks.

6.1.12 Imitation Learning

Imitation learning enables robots to mimic expert behaviors by observing demonstrations. For instance, a drone can learn to navigate through narrow corridors by analyzing the control inputs and camera footage of an experienced pilot. This approach reduces the need for explicit programming in scenarios requiring complex decision-making.

6.1.13 Online Learning

Online learning allows models to update incrementally as new data becomes available, ensuring adaptability to dynamic environments. A delivery drone, for example, might continuously refine its flight model based on real-time wind data from its IMU and GPS sensors, maintaining stability and improving efficiency in varying conditions.

6.1.14 Evolutionary Learning

Evolutionary learning optimizes models through algorithms inspired by natural selection, such as mutation and crossover. A swarm of drones might use evolutionary learning to evolve collaborative strategies for mapping large areas, where each generation improves performance by combining the strengths of previous attempts.

6.1.15 Adversarial Learning

Adversarial learning involves training models in adversarial setups to improve robustness and generalization. For example, a drone can learn to detect and reject spoofed GPS signals by training with adversarially generated GPS noise, ensuring reliable operation even in challenging environments.

6.1.16 Federated Learning

Federated learning enables collaborative model training across multiple drones without sharing raw data, preserving privacy and security. Each drone trains a local model using its camera and LiDAR data, and the aggregated updates improve a global model shared across the fleet. This approach is particularly useful in sensitive applications such as surveillance or disaster response.

6.1.17 Curriculum Learning

Curriculum learning organizes training by progressing from simpler to more complex tasks, mimicking human learning. For example, a drone may first learn to hover, then navigate open spaces, and finally handle complex, obstacle-rich environments like dense forests. This structured approach improves learning efficiency and robustness.

6.1.18 Multi-Task Learning

Multi-task learning enables models to perform several tasks simultaneously by leveraging shared representations. For instance, a drone might simultaneously detect obstacles using LiDAR and estimate wind conditions with IMU data, optimizing its operations for both safety and efficiency.

6.1.19 Intrinsic Motivation Learning

Intrinsic motivation learning encourages agents to explore autonomously by leveraging curiosity or novelty. A drone might prioritize exploring areas with high variability in terrain, using LiDAR and GPS data to discover new landmarks, driven by its intrinsic desire to gather information about its environment.

6.1.20 Zero-Shot Learning

Zero-shot learning enables models to generalize to unseen classes or tasks using semantic relationships or prior knowledge. For example, a drone trained to identify common tree species can recognize a new species based on its similarity to known categories, using visual and structural features from its camera and LiDAR data.

6.1.21 Few-Shot Learning

Few-shot learning allows models to learn tasks with minimal labeled examples, leveraging prior knowledge and generalization. A search-and-rescue drone could quickly identify specific types of debris or hazards in disaster areas using only a few labeled images, enhancing its response capabilities in emergencies.

6.1.22 sensory motor learning

6.2 Reasoning

Reasoning is the cognitive process of deriving conclusions, solving problems, or making decisions based on existing knowledge or logic. Unlike learning, reasoning often operates on pre-existing knowledge or models to produce actionable insights. It encompasses techniques like deductive reasoning (general to specific), inductive reasoning (specific to general), and abductive reasoning (best plausible explanation). In robotics, reasoning allows systems to decide the best course of action, such as selecting a path to avoid an obstacle. Reasoning leverages learned information to achieve goals efficiently.

6.3 Inference

Inference is the act of drawing specific conclusions or making deductions based on evidence or observed data. It can involve probabilistic reasoning, such as Bayesian inference, where a robot estimates the likelihood of events based on prior knowledge and current observations. For example, a drone might infer that a detected object is a tree based on sensory data from its camera and LiDAR. Inference often operates as a step within broader reasoning processes, making it a more targeted, evidence-driven activity.

Inference techniques in robotics enable robots to derive conclusions, make decisions, or interpret data based on sensory inputs, learned models, and pre-existing knowledge. These techniques handle uncertainties, optimize actions, and allow robots to adapt to dynamic environments. Below, we discuss key inference methods used in robotics.

Aspect	**Learning**	**Reasoning**
Definition	Acquiring knowledge or skills through experience or data.	Deriving conclu
Input	External data or experiences.	Pre-existing kn
Output	Updated knowledge or internal model.	Decisions, plans
Scope	Broad, involves building general understanding.	Broader problem
Examples in Robots	Training a model to recognize objects in the environment.	Deciding the be
Relation	Provides the foundation for reasoning.	Uses knowledge

Table 6.1: Comparison of Learning, Reasoning, and Inference

6.3.1 Comparison Table

6.3.2 Inference techniques in robotics

Bayesian Inference

Bayesian inference is a probabilistic technique that updates the likelihood of a hypothesis based on prior knowledge and observed data. In robotics, it is widely used for sensor fusion, where data from multiple sensors like LiDAR, cameras, and IMUs are combined to estimate the robot's position or environment. This method handles uncertainties effectively, making it ideal for localization tasks such as Monte Carlo Localization.

Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation determines the parameters of a model that maximize the likelihood of observed data. In robotics, MLE is often used for model fitting, such as estimating the motion dynamics of a robot or calibrating sensor parameters. For example, a robot can use MLE to optimize its camera parameters to improve visual recognition accuracy.

Monte Carlo Methods

Monte Carlo methods are stochastic approaches that use random sampling to estimate outcomes and probabilities. In robotics, Monte Carlo Localization is a prominent application, where particle filters estimate a robot's pose using sensory data and motion models. These methods are particularly effective in high-dimensional and nonlinear systems, such as planning collision-free paths for drones.

Deductive Inference

Deductive inference involves drawing specific conclusions from general rules or knowledge. Robots employ this approach for rule-based decision-making, where predefined if-then rules dictate actions based on observed states. For example, a robot might deduce that an obstacle detected ahead requires it to stop and plan an alternate path. This method is intuitive and interpretable.

Neural Network-Based Inference

Neural network-based inference uses artificial neural networks to predict outcomes based on learned patterns in data. In robotics, this approach is widely applied in tasks like object detection using convolutional neural networks (CNNs) or motion prediction with recurrent neural networks (RNNs). For instance, a drone might infer the trajectory of a moving object to avoid collisions. Neural inference excels in handling complex, high-dimensional data.

Fuzzy Inference

Fuzzy inference applies fuzzy logic to handle uncertainties and vagueness in data. Robots use this method for decision-making in imprecise or noisy environments. For example, a robotic vacuum might decide how much to adjust its path based on partially observable obstacles. Fuzzy inference provides intuitive and human-like reasoning capabilities, making it suitable for navigation and human-robot interaction.

Chapter 7

Models

7.1 Different models

7.1.1 Classification Models

Classification models are designed to categorize input data into predefined classes by learning decision boundaries in the feature space. These models work by analyzing patterns in the training data and assigning new inputs to one of the learned categories. Common algorithms include decision trees, support vector machines, and neural networks. They are widely used in tasks like object recognition in computer vision, where an image is classified as "cat" or "dog," and anomaly detection, where unusual patterns in data are flagged. Their performance heavily depends on the quality and diversity of the training data.

Discriminative Models

Discriminative models focus on learning the decision boundary between classes by modeling the conditional probability of the output given the input. Unlike generative models, they do not attempt to model the entire data distribution, making them efficient for classification tasks. Examples include logistic regression, random forests, and discriminative neural networks. These models are used in applications like spam email detection, where the task is to classify emails as "spam" or "not spam," and medical diagnosis, where symptoms are mapped to diseases. They are favored for their accuracy and efficiency in prediction tasks.

7.1.2 Regression Models

Regression models predict continuous numerical values by capturing the relationships between input features and outputs. These models fit a function to the data points, minimizing the error between predicted and actual values. For instance, linear regression models establish linear relationships, while more complex techniques like polynomial regression or neural networks capture non-linear relationships. Applications include predicting house prices based on features like size and location, and estimating battery life in drones using flight parameters. Regression models are foundational in tasks requiring precise numerical estimations.

7.1.3 Predictive Models

Predictive models forecast future outcomes by analyzing patterns in historical and present data. They use statistical methods or machine learning algorithms to make informed predictions. For example, a time series model might predict future stock prices by examining past trends, or trajectory prediction models estimate the future path of a drone based on current velocity and environmental data. Predictive models are critical in applications like weather forecasting, financial analysis, and autonomous navigation, where anticipating future states is essential for decision-making.

7.1.4 Generative Models

Generative models learn the underlying distribution of training data to generate new, realistic data points. These models aim to capture the essence of the data, enabling tasks like image synthesis, text generation, and data augmentation. Popular examples include Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and diffusion models. For instance, GANs can generate high-resolution images of artificial objects, while VAEs can create realistic reconstructions of training data. Generative models are vital in applications like creating synthetic datasets for training or enhancing artistic creativity.

7.1.5 Hybrid Models

Hybrid models combine the strengths of generative and discriminative approaches to achieve improved accuracy and robustness. These models leverage generative methods to learn the underlying data distribution and discriminative techniques to optimize decision boundaries. For instance, a hybrid model might use a generative network to preprocess data and a discriminative network for classification. Applications include semi-supervised learning, where labeled and unlabeled data are combined, and robotics, where hybrid models can improve both perception and action planning. Hybrid models are particularly effective in tasks with limited labeled data.

7.1.6 Probabilistic Models

Probabilistic models use principles of probability theory to represent uncertainty and make predictions. These models calculate the likelihood of outcomes given the input data, enabling robust decision-making in uncertain environments. Examples include Bayesian networks, which model probabilistic relationships among variables, and Hidden Markov Models (HMMs), used for sequential data analysis. In robotics, probabilistic models are essential for tasks like localization, where the robot estimates its position in an environment with uncertain sensor readings. These models excel in scenarios requiring explicit uncertainty quantification.

7.1.7 Neural Models

Neural models are advanced machine learning architectures inspired by the human brain, designed for specific tasks through layers of artificial neurons. Convolutional Neural Networks (CNNs) excel at processing visual data, such as recognizing objects in drone imagery. Transformers, on the other hand, are specialized for sequence data and dominate tasks like natural language understanding and time-series analysis. These models are highly flexible, allowing applications in diverse fields, including robotics, where drones use CNNs for obstacle detection and transformers for mission planning based on sequential data. Neural models continue to evolve, shaping modern AI advancements.

7.2 Pattern, pattern recognition models and anomalies

7.2.1 Pattern

A pattern in pattern recognition refers to a set of features or data points that represent some meaningful structure or regularity. It can be thought of as a recognizable arrangement or characteristic that distinguishes one set of data from another. Patterns are typically identified within raw data through various algorithms and methods. Two ways of representing patterns are:

- Feature vectors: Numerical representations of attributes (e.g., size, color, or frequency).
- Graphs or sets: Representing relationships or groupings (e.g., network graphs).

7.2.2 Model

A model in pattern recognition is a mathematical framework that learns to identify or classify patterns from data. The model is built using algorithms trained on sample data from patterns and is then used to recognize patterns in new data.

- First pattern should be detected by a model so that anomaly becomes meaningful? What about statistical models that calculate the distance between two set of points?
- Can a pattern be determined by one time experience?

- What is the difference between pattern recognition in sensory data and anomaly detection? Anomalies are realized from a model that distinguishes a pattern from a non-pattern

- Should we expect any continuous anomaly over continuous time interval a model for pattern recognition?

Chapter 8

Mind

Maybe befor writing this section, I should first check the basics of **Theory of mind**

Chapter 9

Memory

9.1 Memory architecure

Regarding any architecture about memory the following information are important:

9.1.1 Smallest memory units

1. sensory data
2.

9.1.2 Memorizing

9.1.3 Storage

9.1.4 Forgetting

Applications of Forgetting

Forgetting is not merely a failure of memory; it is a functional process essential for efficient cognitive performance [1]. The human brain receives vast amounts of sensory and conceptual information daily. Retaining all of it would overload neural storage and impair the retrieval of relevant memories . Forgetting serves several adaptive purposes:

- **Preventing Cognitive Overload**
  By eliminating non-essential information, forgetting reduces interference between similar memories, making recall of important information more accurate and efficient .
- **Facilitating Learning and Adaptation**
  Forgetting outdated or conflicting information allows the brain to update

its internal models, improving decision-making in changing environments .

- **Enhancing Generalization**
  Removing specific, irrelevant details helps the brain abstract general rules from experiences, enabling better transfer of knowledge to new situations .

- **Emotional Regulation**
  Forgetting can diminish the emotional intensity of traumatic experiences over time, aiding psychological resilience and reducing chronic stress responses .

Memory Loss Due to Natural Forgetting (Decay & Interference)

Neural pathways that store a memory or skill, if not used regularly, undergo synaptic weakening.

Additionally, new information can disrupt previous pathways (interference), making it impossible to retrieve them .

Targeted Forgetting or Active Suppression (Active Forgetting)

Recent research on proteins such as *Rac1* in the hippocampus has shown that the brain has active biological mechanisms for erasing certain memories.

This process is also important in learning and re-training, as it removes unnecessary pathways and frees up space for new information .

9.1.5 Review

In each review the we expect the following

- Hierrarachy (nested clustes)

Hierrarachy

9.1.6 Remembering

9.1.7 Recalling

9.1.8 Memory Architectures for Robots

Episodic Memory refers to the ability of robots to store specific events, including contextual information like time, place, and sensory details. This type of memory is crucial for tasks requiring robots to recall past experiences and apply them to future interactions. For instance, a robot equipped with episodic memory can remember previous locations it visited or interactions with specific humans, which helps improve navigation or social behavior. The CRAM cognitive architecture integrates episodic memory for everyday task execution in human environments, allowing robots to adapt to new scenarios by recalling past experiences.

Semantic Memory stores general knowledge about the world, such as object properties, relationships, and actions. Unlike episodic memory, it is not tied to specific experiences but rather to facts and concepts. Robots use semantic memory to recognize objects and understand their functions. This type of memory is essential for language grounding, as robots can associate words with objects and actions. The OpenCog framework implements semantic memory by building knowledge graphs that robots can access to reason about the world.

Working Memory functions as a short-term memory system that temporarily holds information for immediate processing. It allows robots to keep track of ongoing tasks, manage goals, and adapt to dynamic environments. For example, a robot performing a multi-step task can use working memory to remember intermediate steps and adjust its behavior based on real-time feedback. The ACT-R cognitive architecture incorporates working memory to enable robots to perform complex tasks that require simultaneous handling of multiple pieces of information.

Procedural Memory stores knowledge of how to perform tasks and actions, enabling robots to execute learned skills without conscious recall. This memory type is critical for repetitive tasks, such as industrial robots performing assembly line tasks or humanoid robots learning to walk. Procedural memory is built through reinforcement learning or imitation learning, where the robot refines its motor skills through trial and error. The ROS-TM system, for instance, implements procedural memory to automate robot tasks based on previously learned actions.

Associative Memory allows robots to link different pieces of information through learned associations. This type of memory is crucial for pattern recognition, decision-making, and adaptive behavior. Associative memory helps robots recognize objects or situations based on partial information by recalling previous associations between sensory inputs and outcomes. The Neural Turing Machine is a notable implementation of associative memory that enables robots to learn and recall complex sequences and relationships between data points.

Long-Term Memory refers to the permanent storage of knowledge and experiences. Robots use long-term memory to recall information over extended periods, which is essential for personalized interactions and knowledge retention. For example, a customer service robot with long-term memory can remember user preferences and interactions to provide more tailored responses. The Soar cognitive architecture incorporates long-term memory to store symbolic knowledge and use it for problem-solving and reasoning.

Hierarchical Temporal Memory (HTM) is a memory model inspired by the structure of the neocortex. HTM focuses on learning spatial and temporal patterns in sensory data, allowing robots to predict future events based on past experiences. It is particularly useful for tasks that require the robot to process sequences, such as speech recognition, predictive maintenance, and motion prediction. The Numenta HTM framework has been applied in various robotic applications to enhance predictive capabilities.

9.2 Remembering

Remebering Remembering refers to the general process of retrieving stored information, whether it is triggered actively or occurs spontaneously. It encompasses a broad range of memory activities, including implicit and explicit retrieval mechanisms.

Recalling Recalling highlights the deliberate and task-oriented act of retrieving particular information from memory. For instance, a robot remembering involves the spontaneous recognition of a previously encountered environment, while recalling might involve actively retrieving a stored navigation strategy to solve a specific task. These distinctions are essential when designing cognitive architectures or memory systems in robotics to differentiate between general memory access and targeted retrieval operations .

Remembering can occur through two primary mechanisms: (1) from current sensory observations and (2) through internally triggered processes. Both mechanisms are essential for forming, retrieving, and utilizing memories in both natural and artificial systems.

9.3 Models for Remembering

9.3.1 Remembering from Current Sensory Observations

This category includes mechanisms that rely on sensory inputs to trigger memory recall or to update stored information. These models are fundamental for learning and adapting to new environments.

Models Requiring Sensory Sequences

Sequence Learning Models: Sequence learning encodes temporal patterns in sensory data, enabling the system to predict upcoming events. For instance, in language processing, sequence learning helps anticipate the next word in a sentence, while robots use this model to predict subsequent actions in task execution.

Hierarchical Temporal Memory (HTM): HTM models store hierarchical representations of sensory sequences, recognizing increasingly abstract patterns over time. These models are effective for understanding spatial and temporal regularities in dynamic environments, such as a robot navigating a recurring pattern of obstacles.

Models Requiring Single Sensory Observations

Predictive Coding Models: Predictive coding uses sensory inputs to generate and refine predictions. By minimizing the *prediction error* between expected and observed inputs, the brain adapts its internal model for future use. This process allows the formation of increasingly accurate memory-based predictions, such as anticipating a ball's motion after observing its trajectory multiple times.

Bayesian Frameworks: Bayesian models depend on sensory data as evidence to update prior knowledge into posterior distributions. This allows systems to generalize knowledge and maintain dynamic probabilistic representations of the environment. For example, Bayesian reasoning enables robots to predict object trajectories from noisy sensory observations.

Hebbian Learning and Associative Memory: Associative memory is strengthened through simultaneous sensory observations, linking inputs into robust patterns. For instance, learning to associate a dog's bark with its visual appearance demonstrates how sensory inputs trigger associative recall.

9.3.2 Internally Triggered Remembering

These models enable memory retrieval and utilization without the need for immediate sensory input. Instead, internal processes such as thoughts, queries, or intrinsic motivations drive the recall of stored information.

Associative Memory Models: Memories stored as networks of associations can be recalled using internally generated cues. For instance, recalling a task plan may trigger retrieval of related steps even without external sensory stimuli. This mechanism supports reasoning and problem-solving in both biological and artificial systems.

Cognitive Architectures: Architectures like ACT-R and Soar retrieve information based on internal goals or queries. These frameworks allow systems to adaptively access stored plans or knowledge based on their current objectives. For example, a robot can recall a stored navigation strategy when its internal system determines a similar context.

Generative Models: Generative models, such as VAEs and GANs, simulate scenarios by recalling or creating representations from memory. These models internally generate patterns, enabling predictive planning and decision-making. For instance, robots may simulate task outcomes based on past data to guide actions.

Episodic Memory Models: Episodic memory allows systems to retrieve specific events or experiences using internal queries. Memories tagged with contextual details, such as temporal or spatial markers, enable targeted recall. For instance, a robot can recall the last occurrence of an obstacle to decide on its current navigation.

Intrinsic Motivation Systems: Intrinsic motivations like curiosity or novelty detection provoke memory recall to resolve uncertainties or achieve learning goals. For example, a robot might recall interactions from similar situations to address inconsistencies in its understanding. These systems enable self-driven learning in dynamic environments.

Metacognition and Self-Reflection: Metacognitive systems retrieve memories based on self-evaluation, enabling the adjustment of strategies or correction of errors. For example, a robot analyzing a failed task might recall relevant past experiences to improve its approach. This self-reflective capability fosters long-term autonomy and adaptability.

9.4 Collective memory

9.5 Questions

Mathematical models of remembering

remembering by models? Words or just raw sensory data?

Can we call an agent that can just remember or review sensory data self-aware?

What is reviewing?

Can we call a a robot that can only remember/invoke/provoke sensory data from just a given time interval aware or intelligent?

Chapter 10

Unifying Frameworks for Perception, Action, and Learning in Robotics or Integrated Robotic Intelligence

This chapter explores unifying frameworks that integrate perception, action, and learning in robotics. These frameworks range from foundational approaches like reinforcement learning to advanced methodologies such as neuro-symbolic integration and embodied intelligence. Each framework contributes uniquely to the development of robots capable of adaptive and autonomous behavior.

10.1 Overview of Frameworks

10.1.1 Reinforcement Learning and Hierarchical Extensions

Reinforcement Learning (RL) provides a framework for robots to learn through interaction with their environment, optimizing cumulative rewards. Hierarchical Reinforcement Learning (HRL) extends RL by decomposing tasks into manageable subtasks, simplifying the learning process. For example, a humanoid robot tasked with cleaning a room can use HRL to break the task into locating objects, picking them up, and organizing them. Drones applying RL optimize flight paths to maximize coverage while minimizing energy usage.

10.1.2 Sensorimotor and Embodied Intelligence Frameworks

Sensorimotor learning focuses on coupling sensory input and motor output, enabling robots to refine actions based on feedback. Embodied intelligence expands this by emphasizing the robot's physical interaction with the environment. For instance, a soft robot navigating uneven terrain adjusts its movements based on tactile feedback, demonstrating the integration of physical and sensory adaptation. These frameworks are essential for tasks like grasping, balancing, and navigating complex terrains.

10.1.3 Bayesian Brain and Active Inference

The Bayesian brain framework and active inference approach integrate perception, action, and learning through generative models that minimize prediction errors. Robots continuously update their beliefs and act to reduce discrepancies between predictions and observations. For example, a self-navigating robot adjusts its trajectory based on predicted and observed movements of nearby objects. This framework provides a robust basis for autonomous adaptation.

10.1.4 Damasio's Emotional and Embodied Cognition

Antonio Damasio's work highlights the role of emotions and bodily states in guiding reasoning and decision-making. The somatic marker hypothesis suggests that emotions derived from bodily signals can inform intelligent behavior in uncertain scenarios. Robots inspired by this framework simulate emotional reasoning to enhance human-robot interaction and decision-making, such as adapting responses based on perceived human emotions.

10.1.5 Haykin's Cognitive Dynamic Systems

Simon Haykin's cognitive dynamic systems integrate neural networks, signal processing, and real-time adaptation to address uncertainty and noise in dynamic environments. Applications include multi-agent communication and adaptive control, where robots dynamically adjust behavior based on environmental conditions. For example, drones utilize Haykin's principles for decentralized communication and navigation in complex terrains.

10.1.6 End-to-End Deep Learning Frameworks

End-to-end deep learning frameworks map raw sensory inputs directly to motor outputs using deep neural networks. This approach eliminates the need for intermediate modules, enabling tasks like autonomous driving and robotic navigation. For instance, drones process video feeds in real time to navigate complex environments.

10.1.7 Neuro-Symbolic Integration Frameworks

Neuro-symbolic frameworks combine neural networks for perception with symbolic reasoning for decision-making. Robots using these frameworks recognize objects and follow logical rules to complete tasks. For instance, a robot assembling a mechanical part uses neural vision for identification and symbolic reasoning for correct assembly.

10.1.8 Predictive Coding Frameworks

Predictive coding minimizes discrepancies between predicted and observed sensory inputs by continuously updating internal models. This framework enables robots to act proactively and adapt dynamically. For example, a balancing robot anticipates shifts in its center of gravity and adjusts its posture to maintain stability.

10.1.9 Multi-Agent Reinforcement Learning (MARL)

Multi-Agent Reinforcement Learning (MARL) extends RL to scenarios involving multiple robots collaborating or competing to achieve goals. For example, a swarm of drones collaboratively mapping a disaster area uses MARL to share data and optimize coverage while avoiding overlaps or collisions. This framework integrates shared observations, policy optimization, and coordinated actions.

10.1.10 Integration of Frameworks

While each framework contributes uniquely, their integration can lead to more robust and versatile robotic systems. For instance, predictive coding can enhance neuro-symbolic reasoning by providing anticipatory models, while embodied intelligence can ground deep learning models in real-world physical interactions. Exploring synergies among these frameworks opens new avenues for research in unified robotic intelligence.

Chapter 11

From Awareness to Consciousness

11.1 (Sensory) Data

Data refers to raw, unprocessed facts or measurements that lack context or interpretation. It is the basic building block for deriving meaning but does not convey any significance on its own.

11.2 Awareness

Awareness refers to the ability to perceive, sense, or be conscious of one's surroundings, internal states, or external stimuli. It is a fundamental cognitive capability that allows an entity (human, animal, or even machine) to respond to changes in its environment or internal state.

Does awareness need sensors? For example a robot with all disconnected sensors by only reviewing its memory patterns is aware? Does the literature support internal and external awareness?

11.3 Intelligence

is the ability to learn, reason, adapt, and solve problems. It includes applying knowledge to achieve goals. What is the most basic form of intelligence?

11.4 Cognition

refers to the processes of acquiring, processing, storing, and using information. It includes perception, memory, reasoning, decision-making, problem-solving, and learning. Cognition in robots is artificial and computational. It involves algorithms and models that enable perception, learning, and decision-making. A cognitive robot integrates sensors, learning systems, reasoning mechanisms, and actuators to perceive, understand, adapt, and interact effectively with its environment. The cognitive architecture ties these components together, allowing the robot to exhibit intelligent and adaptive behavior akin to human cognition but without subjective awareness. Here are the components of a cognitive robot:

Component	**Purpose**	**Examples**
Sensors	Collect data from the environment.	Cameras, LiDAR, microp
Perception System	Process and interpret sensor data.	Object detection, SLAM,
Memory and Knowledge	Store past experiences and learned data.	Semantic graphs, long-ter
Learning System	Learn and adapt from experience.	Reinforcement learning, s
Decision-Making	Make optimal choices for actions.	Path planning, probabilis
Actuators and Motion	Execute physical actions.	Motors, robotic arms, mo
Communication	Interact with humans or systems.	NLP, gesture recognition,
Feedback and Adaptation	Improve behavior using feedback.	Error correction, real-tim
Cognitive Architectures	Integrate all components cohesively.	SOAR, ACT-R, ROS, BI
Ethics and Safety Systems	Ensure safe and ethical operation.	Collision avoidance, failsa

11.4.1 Cognitive model

url Cognitive model is a computational or theoretical framework that describes and simulates cognitive processes (e.g., perception, attention, memory, reasoning, and decision-making).

- mental model: A mental model is a cognitive representation or internal simulation that individuals form to understand, predict, and interact with the world around them. It allows people to reason about how things work and to solve problems. A mental model is a subset of a broader cognitive model. Mental models describe an individual's internal understanding, while cognitive models explain the processes that lead to this understanding.

11.4.2 Dynamic cognitive systems and robots

11.4.3 Cognitive architectures

Cognitive Architectures: Cognitive architectures are computational frameworks that model and simulate the structures and processes underlying human cognition. They provide a unified framework for integrating components like memory, reasoning, perception, and learning to mimic human-like intelligence. By organizing cognitive processes into modular structures, these architectures enable the study of general cognitive abilities and their application in artificial systems. Many cognitive architectures also emphasize the interaction between symbolic and sub-symbolic processes, ensuring adaptability in dynamic environments. Their development bridges theoretical cognitive science and practical artificial intelligence research.

Review Papers on Cognitive Architectures:

"A Review of 40 Years of Cognitive Architecture Research: Core Cognitive Abilities and Practical Applications" by Iuliia Kotseruba and John K. Tsotsos provides an extensive overview of cognitive architecture research, discussing core cognitive abilities and various applications.

"Progress and Challenges in Research on Cognitive Architectures" by Pat Langley reviews the notion of cognitive architectures and recurring themes in their study, highlighting both advancements and ongoing challenges in the field.

"Cognitive Architectures and Autonomy: A Comparative Review" by Kristinn R. Thórisson and Thórhallur E. Rögnvaldsson examines systems and architectures designed to address integrated skills, discussing principles and features that contribute to autonomy in intelligent systems.

Most famous cognitive architectures

ACT-R: ACT-R (Adaptive Control of Thought-Rational) is a cognitive architecture designed to simulate human cognitive processes. It integrates declarative and procedural memory, enabling systems to retrieve relevant information based on pattern-matching mechanisms. For instance, in robotics, ACT-R is applied to model human-like reasoning and memory retrieval during tasks such as human-robot interactions. This architecture emphasizes modularity, allowing cognitive processes like memory recall and learning to be closely aligned with human cognition.

Soar: Soar is a cognitive architecture that focuses on learning, problem-solving, and decision-making by using long-term memory retrieval. It enables agents to recall stored plans or strategies when encountering tasks similar to past experiences. In robotics, Soar facilitates navigation and adaptive behaviors by retrieving relevant memory traces during dynamic problem-solving. Its ability to learn from past outcomes makes it a robust framework for modeling cognitive flexibility in artificial systems.

CLARION: CLARION (Connectionist Learning with Adaptive Rule Induction ON-line) is a hybrid cognitive architecture combining symbolic reasoning with sub-symbolic neural processes. This architecture supports memory retrieval by associating high-level rules with underlying distributed representations. In robotic systems, CLARION is particularly useful for integrating reasoning and action, such as adapting to new environments based on learned associations and generalizations. Its dual-process approach ensures flexibility and robustness in memory-dependent tasks.

LIDA: LIDA (Learning Intelligent Decision Agent) is a biologically inspired cognitive architecture that emphasizes cognitive cycles and attention mechanisms. It models human-like perception, decision-making, and learning through its multiple memory systems, including sensory, episodic, and procedural memory. LIDA has been applied in multi-robot systems to enable distributed decision-making and collaborative learning, making it well-suited for dynamic and adaptive environments.

OpenCog: OpenCog is an open-source cognitive architecture aimed at achieving artificial general intelligence (AGI). It uses knowledge graphs and probabilistic reasoning to represent and process complex knowledge structures. In multi-robot systems, OpenCog enables collaborative reasoning and shared decision-making, allowing robots to collectively solve tasks by leveraging distributed knowledge.

SPAUN: SPAUN (Semantic Pointer Architecture Unified Network) is a large-scale brain-inspired model that simulates cognitive processes such as visual recognition, decision-making, and learning. It uses spiking neural networks to perform various tasks, making it particularly useful for simulating human-like cognitive behaviors in robotic systems.

NARS: NARS (Non-Axiomatic Reasoning System) is a cognitive architecture designed for reasoning and decision-making under uncertainty. It adapts to changing environments by continuously learning from incomplete and dynamic knowledge. NARS has been employed in real-time robotics, where adaptive and flexible reasoning is critical.

BECCA: BECCA (Brain-Emulating Cognition and Control Architecture) focuses on unsupervised learning and adaptive control by mimicking biological learning processes. It enables robots to autonomously adapt to new tasks and environments, making it suitable for applications requiring continuous learning and behavioral flexibility.

Cognitive Architecture	**Type**	**Key Feature**
ACT-R	Symbolic	Human-like reasoning and memory retrieval
Soar	Symbolic	Problem-solving and decision-making
CLARION	Hybrid	Combines symbolic reasoning with subsymbolic learning
LIDA	Hybrid	Cognitive cycles and distributed learning
OpenCog	Subsymbolic	Knowledge graphs and probabilistic reasoning
SPAUN	Neuromorphic	Brain-inspired spiking neural network
NARS	Subsymbolic	Adaptive reasoning under uncertainty
BECCA	Neuromorphic	Unsupervised learning and adaptive control

Table 11.1: Summary of Cognitive Architectures for Robotics

Embodied agent

11.4.4 Embodied Cognition

Write about attention as a cognition ability

11.4.5 Attention

11.5 Consciousness

is the state of being aware of one's existence, thoughts, surroundings, and self. It involves subjective experience and introspection. It is a level higher than awareness and in most literature is claimed to be not reachable by robots. Consequently, self-consciousness is also a level higher than self-awareness. A robot can never experience what a burn feel likes.

Chapter 12

Control theory

Feedback theory Feedback theory is a foundational concept in control theory that explains how systems regulate their behavior through feedback loops. It provides a mathematical framework for monitoring outputs, comparing them to desired targets, and making adjustments to minimize errors. This theory ensures stability, robustness, and adaptability in dynamic systems, allowing them to maintain equilibrium and respond to disturbances effectively. Feedback theory is widely applied in robotics for tasks such as stabilization and trajectory control, where sensors provide real-time data to refine motor commands. Its principles are crucial for designing both biological and artificial systems that integrate perception, action, and learning.

12.1 Overview of Control Systems

12.1.1 Open-Loop Control Systems

Open-loop control systems operate without feedback, relying on predefined commands to control the robot's actions. These systems are straightforward to implement and suitable for predictable tasks where system dynamics and disturbances are negligible. For example, in robotics, an open-loop system might control a conveyor belt or move a robotic arm to a fixed position without considering external influences. While cost-effective, these systems lack the ability to correct errors caused by changes in the environment or system inaccuracies, making them unsuitable for dynamic tasks.

12.1.2 Closed-Loop (Feedback) Control Systems

Closed-loop control systems use feedback from sensors to adjust the robot's actions in real-time. This feedback loop ensures that the desired outcomes are achieved even in the presence of disturbances or uncertainties. For instance, a balancing robot like a Segway uses feedback from gyroscopes to maintain its balance, or a drone adjusts its altitude based on barometric pressure readings. These systems are accurate and robust but require sensors and additional computational power, increasing system complexity and cost.

12.1.3 Proportional-Integral-Derivative (PID) Control

PID control is a fundamental closed-loop control algorithm that adjusts system outputs based on the proportional, integral, and derivative of the error signal. It is widely used for tasks requiring precise control, such as maintaining the position or speed of robotic arms or ensuring drones hover steadily in midair. While PID controllers are easy to implement and tune for linear systems, they can struggle with highly dynamic or nonlinear systems where advanced control techniques may be required.

12.1.4 Adaptive Control

Adaptive control dynamically adjusts its parameters to maintain optimal performance under changing conditions. This type of control is ideal for robots operating in unpredictable environments, such as drones dealing with fluctuating wind conditions. For example, an adaptive controller can modify a drone's flight parameters in real-time to maintain stability and performance as the payload or environmental factors change. While powerful, designing and implementing adaptive control systems can be complex and computationally demanding.

12.1.5 Optimal Control

Optimal control focuses on minimizing or maximizing a specific performance criterion, such as energy consumption, time, or trajectory accuracy, over a given system trajectory. In robotics, optimal control might be used to plan efficient paths for robotic arms or drones to minimize travel time and energy use. For example, drones tasked with area surveillance can optimize their flight paths to maximize coverage with minimal battery consumption. While achieving high performance, these systems require accurate models and significant computational resources.

12.1.6 Model Predictive Control (MPC)

Model Predictive Control predicts future states of the system using a model and optimizes control inputs over a finite time horizon to meet constraints and objectives. This approach is particularly useful for complex and multivariable robotic systems. For instance, autonomous drones can use MPC to plan collision-free trajectories in dynamic environments, taking into account sensor data from LiDAR and cameras. Although MPC provides excellent performance in constrained systems, its computational requirements can be a limitation for fast-moving robots.

12.1.7 Robust Control

Robust control ensures system stability and performance even in the presence of uncertainties and disturbances. This type of control is commonly used in industrial robots or drones operating under varying environmental conditions, such as strong winds. For example, a robust controller for a drone would maintain stability and precise operation despite model inaccuracies or sensor noise. While highly reliable, robust control systems often involve conservative designs and require detailed modeling efforts.

12.1.8 Nonlinear Control

Nonlinear control addresses systems where linear assumptions do not hold, such as robotic arms with flexible joints or drones performing aerobatic maneuvers. Nonlinear controllers use specialized techniques to handle complex dynamics, enabling robots to operate effectively in challenging conditions. For instance, a nonlinear controller might stabilize a humanoid robot walking on uneven terrain or a drone performing tight turns at high speeds. Designing nonlinear control systems is mathematically challenging but essential for advanced robotics.

12.1.9 Learning-Based Control

Learning-based control integrates machine learning algorithms to optimize control strategies based on data or experience. Robots, such as drones, can learn to navigate dynamic environments by adapting their flight controllers using data from sensors like IMU, GPS, and cameras. For example, a drone might learn to avoid obstacles in a cluttered environment by combining reinforcement learning with traditional control techniques. This approach is highly effective for complex and evolving tasks but requires extensive training data and computational power.

12.1.10 Hybrid Control Systems

Hybrid control systems combine continuous control methods with discrete decision-making processes, such as finite state machines. These systems are ideal for robots with mixed dynamics, such as drones switching between hover and travel modes or robotic arms transitioning between different operational states. For example, a hybrid controller could manage a drone's transition from autonomous navigation to manual control in emergency scenarios. While versatile, hybrid systems require careful integration of discrete and continuous control elements.

12.1.11 H-Infinity Control

H-Infinity control is a robust control approach that minimizes the worst-case gain of disturbances in the system. This method ensures reliable operation under extreme uncertainties, making it ideal for high-stakes applications such as aerospace robotics. For instance, an H-Infinity controller might stabilize a space robot in the presence of unpredictable gravitational forces or disturbances. While effective in maintaining stability, this approach requires advanced mathematical design and analysis.

12.1.12 Fuzzy Logic Control

Fuzzy logic control uses degrees of truth rather than binary logic to handle uncertainty and imprecision in control systems. This method is well-suited for applications where precise models are unavailable. For example, fuzzy logic can control a robotic vacuum cleaner navigating cluttered rooms or a drone making smooth altitude adjustments. While intuitive and effective in many scenarios, fuzzy logic systems can sometimes lack precision compared to traditional control methods.

12.2 Questions

Are there neural control theories?

Chapter 13

Collectiveness

13.1 The difference between an MRS and SRS

A system can be classified as a *Multi-Robot System (MRS)* if it meets a set of necessary and sufficient conditions. These conditions ensure that the system involves multiple autonomous agents interacting and collaborating in a decentralized manner.

13.1.1 Necessary and Sufficient Conditions

Combining the necessary and sufficient conditions, a system qualifies as an MRS if it meets the following criteria:

1. The system must consist of two or more robots capable of independent decision-making and acting on their own.

2. The system must lack a centralized decision-maker, with control distributed across the robots.

3. The robots must interact through communication or observation to achieve shared goals.

4. The system must distribute tasks across multiple robots, with each robot contributing to the mission.

5. The robots must be able to learn independently and adapt to their environment and each other.

6. There must be mechanisms for coordinating tasks, negotiating roles, or sharing information.

13.2 Different multi robot systems

Heterogeneous MRS

Homogeneous MRS

Swarms

How to choose who are your neighbors?

Swarmnoids

Are there more categories?

13.3 Different forms of relationship between multiple robots

Cooperation refers to collaborative interaction between robots to achieve a common goal that is difficult or impossible to achieve individually. In cooperative scenarios, robots share information, resources, and tasks to improve efficiency. For instance, robots in search and rescue missions can cover larger areas by cooperating, thus increasing the chances of finding victims faster. Cooperation requires task allocation, seamless communication, and synchronization of actions.

Coordination focuses on synchronizing the actions of multiple robots to avoid conflicts and optimize resource usage. Coordination ensures that robots operating in the same environment do not interfere with each other's tasks. For example, in warehouse automation, robots coordinate their movements to avoid collisions while picking and delivering items. Effective coordination mechanisms are essential to prevent resource conflicts and ensure scalability in large robot fleets.

Competition arises when robots compete for limited resources or tasks. This adversarial interaction is common in game-theoretic approaches where each robot tries to maximize its individual performance. For instance, in robot soccer, teams of robots compete to score goals while blocking opponents. Market-based task allocation systems also incorporate competition, where robots bid for tasks, and the most efficient robot wins the task.

Collaboration is a higher-level form of cooperation where robots adapt their behavior dynamically to work together toward a shared goal. Collaboration requires cognitive capabilities, such as reasoning and decision-making, to adjust strategies based on the environment. For instance, robots collaborating in construction tasks adjust their roles in real-time based on progress and obstacles. Collaborative robots can also work alongside humans in manufacturing plants by adapting to human actions.

Swarm Behavior refers to emergent collective behavior in multi-robot systems that is inspired by biological systems such as ant colonies and bee swarms. In swarm robotics, each robot follows simple local rules, and complex global behavior emerges without central control. This approach is particularly useful in tasks like environmental monitoring and foraging, where robots spread out to cover large areas efficiently. Swarm behavior is robust and scalable but requires careful design of local interaction rules.

Negotiation occurs when robots interact to resolve conflicts or agree on task assignments. In negotiation, robots exchange proposals and counterproposals until they reach a mutually acceptable agreement. This type of interaction is crucial in resource-constrained environments, where robots need to negotiate access to limited charging stations or task priorities. Effective negotiation protocols ensure fairness and optimize task allocation in multi-robot teams.

Role Assignment focuses on assigning specific roles or tasks to different robots based on their capabilities and the task requirements. Role assignment ensures efficiency and task specialization in complex multi-robot systems. For example, in search and rescue missions, one robot might focus on mapping the area while another searches for victims. Dynamic role assignment allows robots to switch roles based on changing conditions and task demands.

Social Interaction involves robots engaging in meaningful communication with humans or other robots in a socially appropriate way. This type of interaction is essential for human-robot collaboration in areas like healthcare and customer service. Social robots are designed to recognize social norms and respond to human emotions appropriately. They use verbal and non-verbal cues to enhance interaction quality and build trust with users.

13.4 Collective learning techniques in robots

Centralized learning

Decentralized learning

Hybrid learning

Cooperative Imitation Learning/Multi-Agent Imitation Learning

13.5 Communication

13.6 Scalability

13.7 Anomaly detection in MRS

13.8 Question

Maybe the difference between mrs and srs is cognition of the boundaries of effect

What is Coalition making?

Consensus building

Flocking

Distributed cognition

What is swarm intelligence and how does it apply to robots?

Are there relationships between sociology and MRS?

Chapter 14

Self-awareness

14.1 Self

In robotics, self is defined as the robot's ability to model, predict, and distinguish its own internal state, physical body, and actions from external influences. This "self" is achieved through mathematical models, sensor data, and computational processes.

14.2 Distinguishing Between Self and Non-Self in Robots

14.2.1 Internal Models and Forward Models

Robots use **internal models** to predict the outcomes of their actions. By comparing predicted and observed outcomes, they can distinguish between self-generated actions and external influences.

Mathematical Formulation:

$$\hat{x}_{t+1} = f(x_t, u_t) \tag{14.1}$$

where:

- $\hat{x}_{t+1}$ is the predicted next state.
- x_t is the current state.
- u_t is the control input.
- f is the internal model function.

The error between the predicted and observed states helps identify external disturbances:

$$\text{Error} = \|x_{\text{observed}} - \hat{x}_{t+1}\| \tag{14.2}$$

14.2.2 Artificial Immune System (AIS) Models

Inspired by biological immune systems, AIS models distinguish self from non-self through anomaly detection.

Mathematical Approach:

- Define a *self-space* of normal behaviors.
- Use the **Negative Selection Algorithm** to generate detectors for anomalies.

The affinity function measures the deviation of an observation from the self-space:

$$\text{Affinity} = \|x_{\text{observed}} - x_{\text{self}}\| \tag{14.3}$$

14.2.3 Dynamical Systems and Control Theory

Robots model their own dynamics and detect external forces (non-self) by analyzing deviations.

Dynamics Equation:

$$\dot{x} = Ax + Bu + d \tag{14.4}$$

where:

- x is the state variable.
- A, B are system matrices.
- u is the control input.
- d represents external disturbances.

Non-zero d indicates the presence of non-self influences.

14.2.4 Sensory-Motor Contingencies (SMCs)

SMCs describe the predictable relationship between motor actions and sensory outcomes. Self-generated actions produce expected sensory consequences.

Mathematical Representation:

$$\hat{s}_{t+1} = g(s_t, a_t) \tag{14.5}$$

where:

- $\hat{s}_{t+1}$ is the predicted sensory outcome.
- s_t is the current sensory state.
- a_t is the motor action.

Unpredictable sensory feedback indicates non-self behavior.

14.2.5 Machine Learning-Based Models

Machine learning models, such as autoencoders, detect deviations from learned self-behavior.

Autoencoder Reconstruction Error:

$$\text{Reconstruction Error} = \|x_{\text{input}} - x_{\text{reconstructed}}\| \tag{14.6}$$

A high reconstruction error signals non-self behavior.

14.2.6 Bayesian Inference Models

Bayesian models use probabilistic reasoning to classify observations as self or non-self.

Posterior Probability:

$$P(\text{self}|\text{observation}) \propto P(\text{observation}|\text{self})P(\text{self}) \tag{14.7}$$

Low posterior probability indicates non-self influences.

Summary Table of Models

Model	Core Idea	Use Case
Internal Models	Predict outcomes and compare with observations.	Collision detection, motion verification.
Artificial Immune Systems	Detect deviations from self-space.	Fault detection, anomaly recognition.
Dynamical Systems	Analyze external forces in system dynamics.	Detecting external disturbances.
Sensory-Motor Contingencies	Predict sensory outcomes of motor actions.	Body ownership, tool use.
Machine Learning	Learn self-behavior and detect anomalies.	Anomaly detection, fault isolation.
Bayesian Inference	Probabilistically classify self vs. non-self.	Uncertainty estimation, disturbance analysis.

14.2.7 Questions

Is there a determined boundary between self and none-self?

what is the definition of Self-modeling?

if there are no external forces, then the robot can not distinguish between self and non-self

14.3 Self-awareness

In robotics, self-awareness refers to a robot's ability to recognize, understand, and respond to its internal states and its role within its environment or a system. It involves functional awareness rather than subjective experience, enabling robots to monitor their own status, actions, and interactions with the external world. Self-awareness is a higher-order form of awareness. It refers to the ability to recognize oneself as an individual separate from the environment and others. Self-awareness involves not only perceiving external and internal states but also reflecting on them as being ”one’s own”.

Aspect	**Awareness**
Definition	The ability to perceive and respond t
Focus	External environment and internal st
Complexity	Basic and reactive.
Recognition of Self	Absent.
Example (Humans)	Hearing a sound and turning toward
Example (Robots)	Detecting an obstacle and avoiding it
Level of Cognition	Low-level cognition or sensory process

- Is it an agent’s knowledge about its existence?
  - This boundary classifies between what and what?
    - sensory readings and possible states
    - Are there a trainable and improvable models to classify between self and non-self?

14.3.1 Different levels of self-awareness

Level	**Description**	**Examples in Humans**
Zero-Level	No self-perception.	None.
Physical Self-Awareness	Recognizing one’s body and position.	Proprioception.
Situational Self-Awareness	Awareness of actions in context.	Adjusting behavior in traffic.
Mirror Self-Awareness	Recognizing oneself in a reflection.	Seeing oneself in a mirror.
Extended Self-Awareness	Awareness of self across time.	Reflecting on events.
Introspective Self-Awareness	Reflecting on thoughts and emotions.	Understanding desires.
Social Self-Awareness	Awareness in social contexts.	Adapting to norms.
Meta-Self-Awareness	Awareness of awareness itself.	Pondering existence.

14.3.2 Aspects of self-awareness

- Is self-awareness an ability or an illusion?
- What is the most primitive or primary aspect of self-awareness? (link to staring at a picture)
  - Is it the ability to draw a boundary between self and non-self? Is this possible at all? Is there a clear definite boundary between self and non-self?
  - Does self-awareness exist without time? For example, if a robot stares at a painting and it only has an RGB camera, can it understand that it is alive?

  - remembering
    * What is the definition of remembering?
    * How remembering is different with reviewing?
    * Is one aspect of self awareness is to detect more patterns or sub patterns when reviewing an experience?
    * Is remembering alone enough to say that a robot is self-aware?
    * Can an agent be self-aware without remembering?
    * What is the definition of remembering?
    * Is it only passing a set of sensory data through attention?
    * Is there a relation between language and remembering? For example do we remember better with a language?
  - Is attention an aspect of self-awareness? What is the definition of attention?

- Is self-awareness an ability? (what is the definition of an ability)
- Is self-awareness improvable?
- Is self-awareness a continuous feeling or it can be provoked whenever the agent wills to?
- Agent's knowledge being able to do things further than its programmed goals is an aspect of self-awareness?
- What is illusion? (Definition over sensory data and time)
- Causality? Its relation with language

14.3.3 Applications of self-awareness in robotics

- Does it make a difference wether a robot is self-aware? Maybe such meta knowledge is useless

14.4 Self-modeling

14.5 self-cognition

Self-cognition in robots refers to a robot's ability to reason about its own cognitive processes, knowledge, and limitations. It involves introspection, self-monitoring, and understanding how its decisions and actions are related to its internal models or knowledge.

Aspect	**Cognition**	**Self-C**
Definition	The ability to perceive, process, learn, and act on external inputs.	The ab
Focus	Solving tasks, making decisions, and interacting with the environment.	Reflect
Complexity	Involves perception, learning, and reasoning.	Involve
Key Features	Decision-making, learning, and acting autonomously.	Error c
Examples	Identifying and picking an object.	Recogn
Role	Basic autonomous operation and task-solving.	Enhanc

14.6 Self-consciousness

14.7 Difference between the three selves

Concept	**Definition**	**R**
Self-Awareness	The ability to perceive oneself as distinct from the environment.	Fu
Self-Cognition	The ability to think about oneself, including internal states and goals.	Es
Self-Consciousness	The ability to reflect on oneself as an entity with thoughts and emotions.	Su

Table 14.1: Comparison of Self-Identification in Different Concepts

14.8 Todo

In which category self-identification should be put? self-awareness? Self-cognition or self conciouness?

Chapter 15

Collective-self

Imagine the biggest population of robots in cosmus. Collective is the ability of a robot to understand the boundry of the largest number of robots that their actions are either bounded to it or its actions are bounded to them. This ability already exist in human sociaety.

15.1 Social-awareness

Social awareness refers to an robot's ability to recognize the presence, actions, and roles of other agents in its environment and adjust its behavior accordingly. It involves understanding that other agents are distinct entities with their own intentions and capabilities. In multi-robot systems (MRS), social awareness enables robots to coordinate actions, avoid conflicts, and engage in collaborative tasks. It is essential for human-robot interaction and team-based robotics, as it allows robots to interact meaningfully with humans and other robots by adapting to their actions and roles.

Self-awareness is a prerequisite for social awareness because an agent must first understand itself before it can recognize and interact with others. Self-awareness involves identifying one's own body, actions, and internal states, which is essential for distinguishing self-generated changes from external changes caused by other agents or the environment. Without this distinction, an agent would struggle to recognize which actions it is responsible for and which actions are caused by others, leading to confusion in social interactions. For a robot to adapt its behavior in response to another robot or human, it must first know its own capabilities and limitations. Therefore, social awareness builds upon self-awareness by extending the robot's understanding from itself to its interactions with others, enabling effective coordination and cooperation in multi-agent systems.

15.2 Collective awareness

Collective awareness refers to an robot's ability to understand that it is part of a group working toward a shared goal and to adjust its behavior to optimize the group's overall performance. It goes beyond recognizing individual agents to encompass an understanding of group dynamics, shared resources, and collective tasks. In MRS, collective awareness is crucial for tasks like swarm robotics, where robots need to coordinate without central control to achieve emergent behaviors that maximize group efficiency. This form of awareness allows robots to distribute tasks, avoid redundancies, and adapt to changes in the group's composition.

Social awareness is a prerequisite for collective awareness because recognizing individual agents and their roles is essential before understanding the group as a whole. Social awareness enables an agent to identify other agents, perceive their actions, and understand their goals and intentions. Without this ability to recognize and respond to individual interactions, an agent would be unable to coordinate tasks or participate in shared group goals. Collective awareness builds upon social awareness by requiring an agent to integrate its knowledge of individual agents into an understanding of group dynamics and shared objectives. Therefore, without first understanding who is part of the group and what each agent is doing, a robot cannot optimize group performance or contribute to collective success.

Aspect	**Social Awareness**	**Collective Aw**
Focus	Individual interactions	Group-level und
Main Task	Recognizing other agents and their roles	Understanding s
Adaptation	Adapting to the actions of individual agents	Adapting to the
Example	Waiting for another robot to finish using a shared tool	Adjusting search
Type of Awareness	Interpersonal awareness	Group-level awa
Relevance in MRS	Essential for avoiding conflicts and improving coordination	Essential for ach

Table 15.1: Comparison Between Social Awareness and Collective Awareness

15.3 Social cognition

Social cognition in multi-robot systems (MRS) refers to the ability of robots to understand and respond to individual agents' actions, roles, and intentions. This involves recognizing other robots as distinct entities, predicting their behaviors, and adapting accordingly. Social cognition enables role recognition, conflict resolution, and collaboration in shared environments. For example, in a warehouse, robots use social cognition to avoid collisions by predicting each other's paths and adjusting their trajectories. Inspired by human social cognition, this concept incorporates elements like theory of mind, which allows robots to infer the goals and intentions of others. Social learning is another critical aspect, enabling robots to learn new tasks by observing the actions of peers. Effective social cognition is essential for tasks that require close interaction between robots or between robots and humans. It also facilitates human-robot collaboration by allowing robots to adapt to human commands or gestures. However, implementing social cognition in robotics faces challenges like intention prediction and communication limitations.

15.4 Collective cognition

Collective cognition refers to the ability of a group of robots to process information collectively, make joint decisions, and adapt as a team to achieve shared goals. It goes beyond individual interactions to focus on the emergent group behavior that arises from decentralized coordination. In MRS, collective cognition is exemplified by swarm robotics, where robots collaboratively explore an environment or distribute tasks. Shared memory systems and distributed decision-making are fundamental elements of collective cognition, allowing robots to maintain a unified understanding of their environment. For instance, drones in a disaster zone use collective cognition to collectively map an area, ensuring coverage without redundancy. Collective cognition is driven by local interactions among robots, leading to global behaviors like efficient task allocation or cooperative transport. This concept is inspired by biological systems like ant colonies and bird flocks, where individuals follow simple rules to achieve complex group objectives. While collective cognition enables scalable and robust multi-robot systems, challenges like communication overhead and consensus mechanisms remain significant.

15.5 The role of anomaly detection in collective awareness, social awareness , collective cognition and

15.6 self, others and self

Questions

- Is the notion of being multiple true?
- Does collective self-awareness exist?
- How can two agents agree that they are talking about the same concept?
- How does the presence of others help with self-awareness?
- Is there really a difference between personal self-awareness and collective self-awareness?

– What is the relation between pattern recognition and anomaly detection

15.7 Questions

How do multiple robots know that they are working together?

How does an MRS robots undetstand that they are different than another single or mrs?

Chapter 16

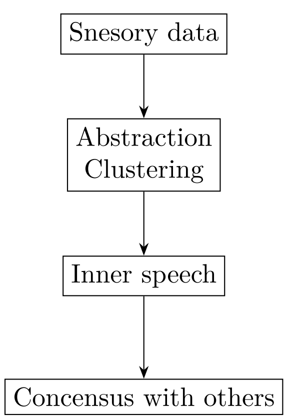
Language

What is alphabet? ”Edmund Husserl” says that ”Sound is Awareness” Thomas Nagel has explored the concept of aural awareness in his philosophical discussions in his paper ”What is it like to be a bat?”

16.1 History

16.2 Formation process

16.2.1 My thoughts



Development

Entities words and alphabets and sentences and larger parts

16.3 Structure

16.3.1 alphabet

16.3.2 words or dictionary

First words are formed

the length of the words are related to their usage frequency. The more they are needed the less lengthy they must be

We need a new words for each detected anomally

16.3.3 Composite structures

16.4 language relations

Linguistic relations (syntagmatic & paradigmatic)

- Paradigmatic / Relation de substitution
- Syntagmatic / Relation de combinaison

16.5 Communication emergence in MRS

16.6 Inner speech

The importance of inner speech in self-awareness

Check this paper: `https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2020.00016/full`

16.7 Language grounding with mutual presence and sensing the same concept or object at the same time

16.8 Language grounding by comparing pattern in the mind

16.9 Concept composition

build corner, straight, build left and associate turn left or straight left

16.10 Formation and refinement

Formation

Refinement Everytime an experience is repeated the language must become more granular

16.11 My thoughts

first find the words by clustering. then

- noun: is a cluster over the boundries of 0th derivative
  - adjective: decription of the noun such as intensity
- verb: is relied to actuators and commands. clustring over the first derivative. bverb must move the nouns in front of sensors of the robot
  - adverb: decription of the verb such as intensity

Language formation phases First an internal language is developed then by a signalling system one robot tries to check if another robot has approximately found the same cluster. if it doesnt find then the word dies in the robot.

16.12 Anomaly detection

Whenever an anomally is higher than what it should be probably based on the 3 sigma rule, then its time to create new language compositions. for example a word or a sentence.

16.13 Todo

Formation idea each robot forms clusters and relatation but has only one communication channel such as a red light that can create morse code so that another robot in vicinity can see it. this blinking red light can be simulated by a channel in which a rebot emmits signals to simulate

Gradually language and its entitity form gradually, but how could this be implemented or modeled?

- what is descriprtion?
- I amp a weak language as map is path with between nodes between start and end? How much is it substituable?
- Like OOP, in natural language, the distance between concept and concrete is determinable or what ever is said is concerete or maybe concept?
- Did first word existed or alphabet? I think word.
- ”time passage” is not a repeated pattern, then how can a word be corresponded to it?
- Should a self-aware robot talk to itself?

- Does talking to self needs another agent?
- Is abnormality detection over a continuous time interval a word or an alphabet?
- Can robot make sense of a word or a sentence without having experience it?
- When should a robot consider a set of ordered alphabets as a meaningful word?
- When does a robot consider a set of ordered words as a sentence?
- Can a robot build meaningful language structures that it knows will never occur in the real world for any agent? words such as beautiful, ugly, god
- Maybe emotional words can be built by growth rate or first derivative.
- Does the alphabet have anything to do with different levels of derivative as growth rates?
- Is one aspect of language to build future probable meaningful structures?
- What is the definition of structure?
- Does a human, before saying something to another, say it to himself?
- Is keeping accidental composition of words or alphabets in mind to continuously assess their meaningfulness an aspect?
- what about two opposite words or sentences?
- Is there a knowledge in the reality in alphabets we don't have opposite or similarity concepts?
- Can neuro-symbolic AI be related to self-awareness? url
- Thinking of robots thinking dependency to the programming language that builds it and machine learning methods it uses.
- There must be a relation between an optimal discovered language in a robot and a good programming labguage
- Define Optimal for an emergent language in a robot
  - Is being optimal equals forming a language that uses its hierarchy and Janeshini and jaygozini methods it detects anomaly better

The proposed part is to describe the proposed solution

Chapter 17

Proposed cognitive dynamic system

Every human at each moment says were I am and were I will be in future? We just focus on this:

- Some sensory data from multiple data that have done something as a normal scenario to build initial knowledge.

- composing the data to build the spatio temporal model

- in a novel scenario we build the new models for deviating parts of the new scenario

17.1 Goals

Goal1: The solution is better

17.2 How to put proprioceptive and exteroceptive sensory data together

Active, proprio-extro-proprio,check in regazoni paper

Passive, extero-proprio-extro

17.3 Sensory data clustering

17.4 Clustering such as kmeans and then kmeans or VAE

17.5 How to put sensory data or data chuncks from multiple robots together

17.6 Preprocessing sensory data phase

De-contextualizing sensory data for example from GPS

Sensory data alignment

This is related to sensors absolute or related values.
do not forget inetrpolation for low frequency sensory data

17.6.1 Feature extraction

Manifold learning

17.7 Offline training

17.7.1 Dimension reduction

PCA, PLS or VAE

17.7.2 clustering

Why clustering

Is there a neural network method for clustering without defining number of clusters

Clustering methods Density based methods:DBSCAN
Clustering for partitioning the derivatives

17.7.3 Training temporal relations

Training between a sequence of 0 derivatives of a sensor and all derivatives of that sensor and other sensors

Hierarchical Transformers for Multi-Modal Learning

Used in video, audio, and multi-modal fusion, where different levels of abstraction are needed. Example: HAT (Hierarchical Attention Transformer) structures different levels of attention across modalities (text, image, speech).

17.7.4 Builiding a Spatio-temporal model

We are looking for a solution which relies totally on neural networks that related

1. each dim reduced sensor cluster to its derivatives

2. Todo correlation analysis should be done to see which extro sensors are correlated with which pro sensor and even this study must be done between extro sensors and pro sensors too

Conditional Temporal Networks (CTN)
LSTMs
Temporal Transformers
Time series transformers
Hierarchical transformers
Multimodal Transformers
GANs ?
Application of attention mechanism

17.8 Online testing

According to current sensory readings in which class I am?

For each class that I might be, how much is it possible for all states in it with probability

17.8.1 Anomally detection

17.8.2 Novelty detection

Using GANs for building the prediction in future

Chapter 18

Experiments

18.1 Proprioceptive sensors

IMU (Accelerometer, Gyroscope), Motor encoder (Motor RM), Compass (Magnetometer) , Temperature Sensor (Thermal sensor), Barometer (Altimeter), Battery voltage

18.2 Exteroceptive sensors

LIDAR, RGB/Depth Camera, GPS **GPS** is considered an exteroceptive sensor because it provides information about an entity's position based on external references—satellites in Earth's orbit. Even if you "close your eyes" and perceive your position internally through a GPS-like mechanism in your brain, the fundamental source of that information still comes from outside of you.

18.3 System settings

18.3.1 ROS

18.3.2 Mavlink

18.3.3 Mavros

`/mrs]_workspace /catkin_ws /workspace`

18.4 Linux Virtual environment

Creating the virtual environment

```
python -m venv ~/phd-v
```

To activate

```
source ~/phd-venv/bin/activate
```

To deactivate

```
deactivate
```

Installing Test environment Pytest is used for testing

```
pip install -U pytest pytest-cov pytest-xdist
```

18.4.1 Pycharm

Interpreterin Pycharm To set the interpreter virtual environment go to: /phd_venv/bin and choose python and not python 3.13 or other

18.5 scenarios

18.5.1 Normal

- **Drones:**
  - **Num**: 2
  - **Type**: TarotT650
    - sensors
      - GPS odom
      - RP A2 LIDAR
- **Yaml file:**:
  - **path**: `/home/donkarlo/Dropbox/projs/research/data/self-aware-drones/ctumrs/two-drones/normal-scenario/uav1-gps-lidar-uav2-gps-lidar.yaml`
  - **number of lines**: 34902346
  - **Number of yaml documents**: 1041927

18.6 Todo

Todo list

Bibliography

[1] Oliver Hardt, Karim Nader, and Lynn Nadel. Decay happens: The role of active forgetting in memory. *Trends in Cognitive Sciences*, 17(3):111–120, 2013.