International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 38, No. 1, February, 2024, pp. 351-363
Journal homepage: http://iieta.org/journals/ria

Elevating Mobile Robotics: Pioneering Applications of Artificial Intelligence and Machine Learning

Haider Sahib Nasrallah[1,2*], Ivan V. Stepanyan[1,3], Karrar Sahib Nassrullah[1,4], Neder Jair Mendez Florez[1], Israa M. Abdalameer AL-Khafaji[5], Abdelrrahmane Mohamed Zidoun[1], Ravi Sekhar[6], Pritesh Shah[6], Sushma Parihar[6]

[1] Department of Mechanics and Control Processes, Academy of Engineering, Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya Street, Moscow 117198, Russian Federation
[2] Computer Department, University of Kerbala, Karbala 56001, Iraq
[3] Mechanical Engineering Research Institute of the Russian Academy of Sciences (IMASH RAN), M. Kharitonyevskiy Pereulok, Moscow 101990, Russian Federation
[4] Mechanical Engineering Department, College of Engineering, University of Kerbala, Karbala 56001, Iraq
[5] ERP Systems Department, Institute of Information Technologies, MIREA-Russian Technological University 78, Vernadskogo pr., Moscow 119454, Russia Federation
[6] Symbiosis Institute of Technology (SIT) Pune Campus, Symbiosis International (Deemed University) (SIU), Pune 412115, Maharashtra, India

Corresponding Author Email: haider.sahib@uokerbala.edu.iq

https://doi.org/10.18280/ria.380137

Received: 17 September 2023
Revised: 20 November 2023
Accepted: 28 November 2023
Available online: 29 February 2024

Keywords:
mobile robot, artificial intelligence, machine learning, robot localization, artificial neural networks, convolutional neural networks

ABSTRACT

The present study delves into the utilization of subsumption architecture for the modeling of mobile robot behaviors, particularly those that respond adaptively to environmental dynamics and inaccuracies in sensor measurements. Central to this investigation is the deployment of reactive controller networks, wherein each node—representing a distinct state—is governed by sensor-triggered conditions that dictate state transitions. The methodology adopted comprises a thorough literature review, encompassing sources from IEEE Xplore, ScienceDirect, and the ACM Digital Library, which discuss the integration of subsumption architecture in the realm of mobile robot control. Through this review, the effectiveness of subsumption architecture in crafting reactive robotic behaviors is underscored. It has been established that augmented finite state machines (AFSMs), which are integral to the subsumption architecture and possess internal timing mechanisms, are pivotal in managing the temporal aspects of state transitions. Additionally, the technique of layering—merging multiple simple networks to form intricate behavior patterns—emerges as a significant finding, accentuating the architecture's capability to facilitate complex behavioral constructs. The prime contribution of this body of work lies in identifying and elucidating the strategic role of subsumption architecture in enhancing the adaptability and robustness of mobile robots. The insights gleaned from this study not only advance our understanding of robotic control systems but also hold implications for the amplification of industrial efficiency and effectiveness through the application of sophisticated AI and machine learning techniques in mobile robotics.

1. INTRODUCTION

Mobile robots have become increasingly prevalent in various industries, including manufacturing, healthcare, and transportation. The ability of mobile robots to perform complex tasks and make autonomous decisions has been made possible through advancements in artificial intelligence (AI) and machine learning algorithms. One approach to modeling robot behaviors that are reactive to environmental changes and sensor measurement errors is subsumption architecture. Subsumption architecture consists of networks made of reactive controllers, where a node or state contains sensor conditions that determine whether to stay in or leave that state [1]. While the use of sensors such as LIDAR, cameras, infrared technology, and ultrasonic systems is mentioned in the literature, there is a gap in the existing research regarding how each sensor contributes to perception and control operations in mobile robots. Therefore, the objective of this study is to explore the use of subsumption architecture in mobile robot control and to provide a comprehensive understanding of how sensors contribute to perception and control operations in mobile robots. The study includes a literature review of articles from various databases, such as IEEE Xplore, ScienceDirect, and ACM Digital Library, that discuss the use of subsumption architecture in mobile robot control. The main contribution of this research is to provide valuable insights into the use of subsumption architecture in mobile robot control, which has the potential to improve the efficiency and effectiveness of various industries [2].

Among these pivotal components, several key aspects bear significance when it comes to controlling mobile robots. First and foremost, sensing and perception constitute vital domains. Mobile robots depend on an array of sensors such as LIDAR, cameras, infrared technology [3, 4], and ultrasonic systems [5] to apprehend their surroundings. These sensors furnish the robot with essential data, enabling the detection of obstacles and object recognition, thus serving as the bedrock for perception and control operations [6]. In the realm of localization and mapping, mobile robots are tasked with estimating their position within a given environment-a process aptly termed localization [7, 8]. Simultaneously, mapping involves the construction of a model of the robot's surroundings, leveraging data harvested from its sensors [9]. The ingenious technique of Simultaneous Localization and Mapping (SLAM) deftly addresses both these tasks in tandem, allowing the robot to dynamically build a map while continually ascertaining its real-time position [10-12]. The third critical facet is path planning and navigation. Once a mobile robot has gleaned information about its surroundings, it must chart a course from its current location to a predefined destination while adeptly circumventing obstacles [13]. Navigation, in this context, encompasses the execution of the planned path, which necessitates continuous updates to the robot's position and precise trajectory adjustments. This section also delves into an exploration of the frameworks employed within the mobile robot navigation field [14]. Finally, the overarching element of control takes center stage. A proficient control system steers the robot along its designated path by manipulating its actuators. Various control techniques, such as the proportional-integral-derivative (PID) controller [15-18], model predictive control (MPC) [19-22], and adaptive control, find application in this realm. This research inquiry zeroes in on the initial block of localization-a pivotal prerequisite for successful navigation in the mobile robot domain. In the context of robotics, localization entails the identification of objects and entities within the robot's environment, including its own spatial positioning [23]. Mobile robot localization, sometimes referred to as positioning estimation or positioning tracking [24, 25], emerges as a focal point in the quest to unleash the full potential of these dynamic robotic systems.

2. PAGE SETUP SUBSUMPTION ARCHITECTURE

One of the most widely used approaches to mobile robot control is subsumption architecture, which is often used in conjunction with AI techniques such as machine learning and neural networks. Subsumption architecture is a behavior-based approach that involves the use of reactive controllers organized into a hierarchy of layers, where each layer is responsible for a specific behavior. The layers are arranged in a way that allows higher-level behaviors to subsume lower-level behaviors, hence the name "subsumption architecture."

The use of subsumption architecture in mobile robot control has several advantages. First, it allows for the creation of complex behaviors by combining simple behaviors. Second, it enables the robot to react quickly to changes in the environment, as the reactive controllers are designed to respond to sensor feedback in real-time. Finally, it allows for the creation of robust and fault-tolerant systems, as the layers can be designed to operate independently of each other.

The integration of subsumption architecture with AI techniques such as machine learning and neural networks has led to significant advancements in mobile robot control. For example, machine learning algorithms can be used to train the higher-level behaviors of a mobile robot, while subsumption architecture can be used to implement the low-level behaviors. This combination of techniques allows for the creation of highly adaptive and intelligent mobile robots that can perform complex tasks and make autonomous decisions. In this study, we will explore the current landscape of AI applications in the control of mobile robots, with a particular emphasis on machine learning and neural networks. We will also examine the role of subsumption architecture in mobile robot control and its integration with AI techniques. By the end of this study, you will have a comprehensive understanding of the cutting-edge applications of AI and machine learning in mobile robotics and the role of subsumption architecture in creating highly adaptive and intelligent mobile robots. For moving a mobile robot to a goal position, it is told to execute a particular type of force into the wheels' motor for a period of time rather than going directly to the goal position coordinates. The robot's movement is programmed using a controller, a piece of software that tells the robot what to do after sensing the environment until a control objective is achieved. Reactive controls are oriented to deal with simple reflex agents. Most of the actions a robot can execute are separable in subcomponents [26]. These subcomponents can be modelled as finite-state machines that react to sensor feedback [27]. Subsumption architecture emerged in the 1990s due to the need for more intelligent robot controllers that model robot behaviours that are reactive to environment changes and sensor measurement errors. The subsumption architecture was developed in the study [28]. The implementation consists of networks made of reactive controllers. A node or state often contains sensor conditions that determine whether to stay in or leave that state. It is possible to go from one state (the transmitter) to another (the receiver) through arcs containing messages that are the output of the recently left state. Sometimes, these arcs represent physical wiring, and the messages are the signals sent to the robot's motor. These augmented finite state machines (AFSM) have internal clocks that control the time it takes to traverse an arc; the word augmented refers to such internal clocks. Figure 1. Below shows a model of a robot that uses (AFSM) and explains its working mechanism. Two or more simple networks representing simple behaviours can be connected one to another to form more complex networks and thus represent more complex behaviours, as mentioned in the study [29]. This process is called layering.

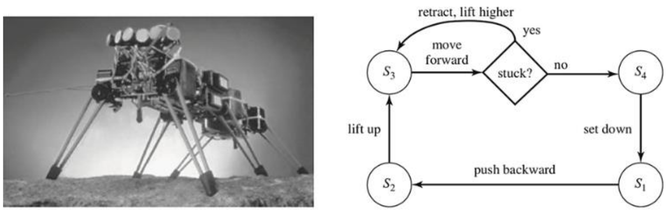

Figure 1. A hexapod robot called Genghis (left) was created by Brooks [30] using 57 augmented finite states machines (AFSM). An AFSM for controlling only one leg is illustrated on the right side. The AFSM receives feedback from a proximity sensor in order to see if the leg is stuck or not and to lift it higher next time

3. ARTIFICIAL INTELLIGENCE FOR CONTROLLING MOBILE ROBOTS

Artificial intelligence has significantly improved the capabilities of mobile robots, enabling them to learn and adapt to their environments, perform complex tasks, and make decisions autonomously. One of the most prominent AI techniques used in mobile robot control is machine learning, which involves training a model on a dataset to make predictions or decisions. Machine learning has been successfully applied in various tasks such as obstacle avoidance, navigation, and mapping. Fuzzy logic is another AI technique that has been widely used in mobile robot control. Fuzzy logic is a form of AI that excels in handling imprecise or uncertain data. In mobile robot control, fuzzy logic is often used for tasks such as obstacle avoidance, navigation, and mapping. For example, fuzzy logic can be used to control the speed and direction of a mobile robot based on the distance to an obstacle. Genetic algorithms are another AI technique that has been used in mobile robot control. Genetic algorithms are a type of optimization algorithm that mimics the process of natural selection. In mobile robot control, genetic algorithms can be used to optimize the parameters of a control system, such as the weights of a neural network. AI techniques such as machine learning, fuzzy logic, and genetic algorithms have been successfully applied in mobile robot control for tasks such as obstacle avoidance, navigation, and mapping. These techniques have enabled mobile robots to perform complex tasks and make autonomous decisions, making them valuable assets in various industries.

Mobile robots have been revolutionized by the incorporation of artificial intelligence (AI) techniques with other disciplines, such as control and electronics, that co-create them. Artificial intelligence pertains to the capacity of robots or machines controlled by computers to execute tasks that closely resemble or mimic human capabilities [31]. Artificial intelligence has significantly improved the capabilities of mobile robots. Mobile robots are equipped with sensors that enable them to perceive their environment, and AI algorithms are used to process this sensory data and make decisions based on it. AI has allowed mobile robots to learn and adapt to their environments, perform complex tasks, and make decisions autonomously, such as navigation, object detection, and localization. Within the domain of mobile robot control, a diverse array of AI techniques are leveraged to augment their capabilities. One such technique is Fuzzy Logic, a form of artificial intelligence that excels in handling imprecise or uncertain data. Its application in mobile robot control spans a spectrum of tasks, encompassing obstacle avoidance, navigation, and mapping, where its adaptability to uncertain environments proves invaluable. Another powerful tool in the arsenal of mobile robot control is the Genetic Algorithm (GA), an AI approach inspired by the mechanisms of natural selection and genetics. By mimicking nature's optimization processes, GA plays a pivotal role in refining the paths and trajectories of mobile robots. Through its iterative and evolutionary approach, GA enables robots to navigate and operate in complex and dynamically changing environments with enhanced efficiency and precision.

4. MACHINE LEARNING IN ROBOTICS

Machine learning is a powerful AI technique that has revolutionized the field of mobile robotics. Machine learning involves training a model on a dataset to make predictions or decisions. In mobile robot control, machine learning has been successfully applied in various tasks such as object detection, image recognition, path planning, and more. There are two main types of machine learning: supervised learning and unsupervised learning. Supervised learning involves training a model on labeled data, where the input data is paired with the correct output. The model learns to predict outputs based on input data by identifying patterns within the labeled dataset. Supervised learning algorithms find applications in a multitude of mobile robot control tasks, spanning from object detection and image recognition to path planning. For example, a mobile robot can be trained to recognize different objects in its environment using supervised learning. On the other hand, unsupervised learning involves training a model on unlabeled data, where the input data is not paired with any output. The model learns to discern patterns and structures within the data without any explicit guidance. Unsupervised learning algorithms contribute significantly to mobile robot control, serving purposes such as clustering, anomaly detection, and feature extraction. For example, unsupervised learning can be used to identify anomalies in sensor data, which can be indicative of a malfunctioning component in the robot. machine learning is a powerful AI technique that has been successfully applied in various mobile robot control tasks. Supervised learning involves training a model on labeled data, while unsupervised learning involves training a model on unlabeled data. Both approaches have their unique advantages and applications in mobile robot control, making them valuable tools for creating intelligent and adaptive mobile robots.

Machine learning (ML) is an expanded group of algorithms that can utilize datasets to determine patterns, deepen understanding and make determinations [32]. It gives the tools needed to analyze structured or unstructured data, draw conclusions, and make predictions [33]. The goal of machine learning is to investigate the concepts, algorithms, and theories used in statistics, computer science, and optimization [34]. According to the study [35], a robot can be seen as a system that interacts with the world and uses learning as a method of improving its performance. Machine learning plays a pivotal role in shaping the landscape of robotics, but it is not without its set of challenges. To fully harness its potential, machine learning in robotics must confront several key obstacles: Firstly, it must acquire the capability to learn representations of the world that align with the specific objectives programmed into the robot. These representations serve as the foundation for the robot to comprehend and achieve its intended purpose. Secondly, machine learning should encompass the calibration of sensors and actuators [36-42]. This calibration process enhances the precision and accuracy of these critical components, enabling the robot to interact more effectively with its environment. Lastly, it's crucial for machine learning to address the nuances of a robot's behavior in relation to its perceived environment and internal state. Understanding and managing the intricate interplay between these aspects are paramount for effective learning and decision-making. Within the realm of mobile robot control, several types of machine learning algorithms come into play:

Supervised Learning stands out as a prominent approach, involving the training of machine learning models on labeled data. Through this method, models acquire the ability to predict outputs based on input data by identifying patterns within the labeled dataset. Supervised learning algorithms find applications in a multitude of mobile robot control tasks, spanning from object detection and image recognition to path planning. On the other hand, Unsupervised Learning takes a different route, relying on unlabelled data for training. In this paradigm, machine learning models autonomously discern patterns and structures within the data without any explicit guidance. Unsupervised learning algorithms contribute significantly to mobile robot control, serving purposes such as clustering, anomaly detection, and feature extraction. These techniques further enrich the repertoire of machine learning tools at the disposal of robotics researchers and practitioners.

5. LITERATURE REVIEW

Machine learning, predictive analytics, or statistical learning represents a dynamic and ever-evolving research field situated at the fascinating crossroads of statistics, artificial intelligence, and computer science [43]. This interdisciplinary domain serves as a veritable powerhouse, equipping us with the essential tools required to extract knowledge and make predictions when faced with a deluge of structured or unstructured data [44]. To truly appreciate the profound impact of machine learning, it's insightful to take a step back to the early days of artificial intelligence. Back then, systems were deemed "intelligent" if they operated on meticulously hand-crafted if-else statements and rule-based logic [45]. However, as the volume and complexity of data continued to surge exponentially, the limitations of such rule-based systems became glaringly evident. This paradigm shift ushered in the era of machine learning, where algorithms and techniques have the capacity to autonomously learn from data, adapt, and improve their performance over time. In this section, we embark on a journey through the vast landscape of machine learning, a realm teeming with a rich assortment of algorithms and techniques. These tools empower us to traverse the intricate terrain of data, extracting meaningful insights, and enabling us to predict future trends, outcomes, and patterns. Whether it's the formidable power of deep neural networks, the interpretability of decision trees, the versatility of support vector machines, or the elegance of Bayesian methods, we'll delve into the most popular and widely embraced machine learning approaches. Each of these techniques serves as a potent arrow in the quiver of data scientists and researchers, equipping them to tackle diverse challenges and unlock the latent potential of data-driven decision-making.

5.1 Artificial neural networks

The concept of *Artificial Neural Networks (*ANN*)*, also called *neural networks (*NN*)*, is extensively used in controlling mobile robots in areas such as object detection, image recognition, and navigation. NN is inspired by networks of brain cells called *neurons, and* NN is modelled after the structure and function of the human brain [46, 47]. ANN is made of many layers; each layer contains a bunch of *units* that are connected with *units* in other layers by *links*. A link going from unit *i* to unit *j* has a numerical value associated with it that represents the strength of the link called *weight*, denoted as w_{ij}. There are three kinds of layers in the neural network [48] as follows:

- Input layer: It does not have any preceding layer; it receives as input direct values.
- Hidden layer: it is the layer that is between the input and output layers. It can connect other hidden layers.
- Output layer: It provides the output of the neural network.

Convolutional neural networks (CNN) are the most popular and frequently used type of neural network that is specifically designed for image processing and pattern recognition. CNN's primary advantage is that it determines pertinent features without human intervention [49]. The CNN improves upon the architecture of conventional artificial neural networks. Every layer in a CNN considers the optimal parameters necessary to provide a meaningful output while also reducing the complexity of the model [50]. CNN's structure, similar to a traditional neural network, has been inspired by the neurons in animal and human brains. In particular, the optic cortex of a cat's brain is made up of a complicated pattern of cells, and CNN simulates this sequence [51]. CNN has been widely used in various applications for mobile robots, particularly in tasks related to computer vision [52], such as object recognition, segmentation, and scene understanding. CNN (Convolutional Neural Networks) applications in mobile robotics encompass a wide array of functionalities that enhance the capabilities of these robotic systems. One prominent application is object recognition and classification, where CNNs excel at identifying and categorizing objects within the mobile robot's environment [53]. Another crucial role that CNNs play in mobile robotics is obstacle detection and avoidance, contributing significantly to navigation [54-58]. By leveraging CNNs, robots can discern obstacles in their path and make informed decisions to navigate around them safely. Semantic segmentation is another valuable application of CNNs in mobile robotics [59]. This involves the classification of each pixel in an image according to the object or category it represents. Such segmentation aids mobile robots in comprehending their surroundings, thereby facilitating tasks like obstacle avoidance and path planning, as mentioned in the study [60]. In the domain of visual SLAM (Simultaneous Localization and Mapping), CNNs prove instrumental. They enhance feature extraction, matching, and loop closure detection, as discussed in the study [61, 62], thereby contributing to the accuracy and efficiency of SLAM systems employed by mobile robots. CNNs also find utility in depth estimation for mobile robots [63]. By employing CNNs for monocular depth estimation, robots can derive depth information from single RGB images. This depth information is invaluable for navigation and obstacle avoidance. Furthermore, CNNs extend their reach to path planning and decision-making tasks for mobile robots [64]. These networks assist in learning to navigate in intricate environments and predicting the behavior of other agents, allowing robots to make informed decisions during their operations. In sum, CNNs play a pivotal role in advancing the capabilities of mobile robots across a spectrum of applications, encompassing object recognition, obstacle detection, semantic segmentation, visual SLAM, depth estimation, and decision-making in complex environments.

5.2 Reinforcement learning

Reinforcement Learning (RL): RL is a machine learning technique that enables a robot to learn a control policy through trial and error [65]. RL involves training a machine learning model to take actions based on rewards or punishments. Through the utilization of information collected through the examination of the surrounding environment [66], the model learns to optimize its actions to maximize rewards and minimize punishments [67]. RL is a highly effective technique for training machine learning models, which in turn can contribute to the advancement of automation and the optimization of operational efficiency within the field of robotics [68]. Reinforcement learning algorithms have been used for various tasks in mobile robot control, such as obstacle avoidance [69], navigation [70, 71], path planning, and objects manipulation [72].

5.3 Symbolic regression techniques

Symbolic regression techniques are employed as machine learning methodologies to facilitate the implementation of a novel form of control that involves the manipulation of the stable equilibrium point's position. The successful execution of such control necessitates the establishment of a dual feedback loop. The presence of an inner contour guarantees the establishment of stability for the control system with respect to a specific point within the state space. The external contour facilitates the attainment of optimal control over the position of the stable equilibrium point. Symbolic regression techniques utilize a specialized coding system to represent the mathematical expression being searched and employ a special genetic algorithm to optimize the solution within the codes space. A specialized crossover procedure has been created for this purpose. The utilization of a specialized crossover operation on two parental codes facilitates the generation of two novel child chromosomal. Various crossover techniques are employed for distinct code structures. Symbolic regression techniques are commonly employed in the domain of supervised machine learning to approximate data that requires modelling. The issue at hand in machine learning for control pertains to the absence of a training set, necessitating the pursuit of a control function through the minimization of a quality criterion. This particular methodology, within the context of established nomenclature, is commonly referred to as unsupervised learning. Currently, a variety of symbolic regression techniques exist, including genetic programming (GP) [73], Cartesian GP [74], parse matrix evolution [75], network operator method [76], complete binary GP [77] and others.

6. ROBOT LOCALIZATION

One of the critical challenges in mobile robotics is robot localization, which involves determining the position and orientation of a robot in its environment. Robot localization is essential for enabling mobile robots to navigate autonomously and perform tasks accurately. Several techniques have been developed to address the problem of robot localization, including Bayes Filter, Kalman Filter (KF), Extended Kalman Filter (EKF), and Particle Filter. Bayes Filter is a probabilistic approach to robot localization that involves estimating the probability distribution of the robot's position and orientation based on sensor measurements. Bayes Filter has been successfully applied in various mobile robot control tasks, such as simultaneous localization and mapping (SLAM). For example, a mobile robot can use Bayes Filter to estimate its position and orientation based on sensor measurements such as odometry and laser range finders. Kalman Filter (KF) is another popular technique for robot localization that involves estimating the state of a system based on noisy sensor measurements. KF is widely used in mobile robot control tasks such as tracking and navigation. For example, a mobile robot can use KF to estimate its position and velocity based on sensor measurements such as GPS and inertial sensors. Extended Kalman Filter (EKF) is an extension of KF that can handle nonlinear systems. EKF is widely used in mobile robot control tasks such as localization and mapping. For example, a mobile robot can use EKF to estimate its position and orientation based on sensor measurements such as visual odometry and laser range finders. Particle Filter is a probabilistic approach to robot localization that involves estimating the probability distribution of the robot's position and orientation based on a set of particles. Particle Filter has been successfully applied in various mobile robot control tasks, such as SLAM and tracking. For example, a mobile robot can use Particle Filter to estimate its position and orientation based on sensor measurements such as visual odometry and laser range finders. Robot localization is a critical challenge in mobile robotics, and several techniques have been developed to address this problem. Bayes Filter, Kalman Filter (KF), Extended Kalman Filter (EKF), and Particle Filter are some of the popular techniques used in mobile robot control tasks such as localization, mapping, and tracking. These techniques have enabled mobile robots to navigate autonomously and perform tasks accurately, making them valuable assets in various industries.

Localization, a critical process in robotics, revolves around determining a robot's precise position within its surroundings. When exploring the diverse landscape of localization techniques, several key comparisons come to light: Firstly, there's the distinction between Local and Global Localization. In scenarios where a robot lacks any prior information about its initial position, it must rely on sensor observations to infer its pose. This approach extends to challenging situations like the kidnapped robot problem, wherein a robot aware of its environment suddenly finds itself relocated to an unfamiliar location [78]. Another crucial consideration is whether localization occurs in Static or Dynamic Environments. The localization process must adapt depending on whether the robot operates in a stable, unchanging setting or contends with a dynamic environment characterized by moving entities [79]. Dynamic environments introduce added complexity as the constantly changing landscape necessitates the incorporation of dynamic elements into the environment state. Furthermore, the dimension of Single-robot versus Multi-robot Localization warrants attention. In the former, the focus revolves around a solitary robot navigating the environment, whereas the latter scenario entails deploying a group of robots within the same surroundings. Interestingly, the insights gleaned from single-robot localization can often be applied individually to each robot within a multi-robot setting, facilitating collaborative localization efforts [80, 81]. These distinctions underscore the intricate web of choices and considerations inherent in the realm of robotics localization. Sensors and actuators take part in determining the robot’s localization, but both are subjected to noise, and thus the problem of localization becomes difficult [82]. Another problem with sensors is that they usually do not provide enough information content to determine the position of the robot directly, and therefore, the robot might be in an ambiguous location. This problem is known as *sensor aliasing Siegwart: intro-autonomous-robots* and, along with sensors and actuators noise turns the localization problem into a difficult task. Here are some known techniques to deal with robot localization are mentioned:

(1) Bayes Filter: As claimed by Fox et al. [83], *Bayes Filter* is a statistical estimator of a dynamic system's state based on noisy observations. In other words, it is an algorithm that calculates the belief's distribution based on measurements and control data [84]. This is generally done in two steps: the *prediction* and the *correction* step, and thus each time a robot receives the sensor's measurement data, the robot controller software needs to compute the posterior densities but notice that such a task is computationally heavy. Consequently, its time complexity grows exponentially because the amount of sensor's measurements increases over time. A solution to this problem is to assume that such a dynamic system has the Markovian Property. That is, the future state x_{t+1} depends on the present state x_t because of the assumption that x_t carries all the needed information. Bayes filter algorithm is recursive. It requires the previous posterior distribution to calculate the current posterior distribution.

(2) Kalman Filter (KF): is a Gaussian technique invented in 1950 by Kalman [85] and was first implemented by NASA in the Apollo program to estimate the trajectory of the space capsule in real time [86]. It takes noisy data and takes the noise out to get information with less uncertainty [87]. It works with continuous states and represents the beliefs by the first and second moments [88] from multivariate normal distributions. Laskar et al. [89] proposed another version of KF called Maximum Correntropy Kalman Filter (MCKF), an effective and robust algorithm for non-Gaussian signals and heavy-tailed impulsive noise. Instead of using the Minimum Mean Square Error (MMSE) as KF, its optimality criterion is the Maximum Correntropy Criterion (MCC).

(3) Extended Kalman Filter (EKF): does not assume any linearity as KF does, and therefore, the next state and the measurement probabilities are nonlinear functions, as mentioned in the study [89]. Some variants of this method include the *Unscented Kalman Filter*, where the state distribution is approximated by a Gaussian random variable as in EKF, but now a minimal set of sample points is cautiously chosen for representing the state distribution [90]. Another improvement of EKF is the *Invariant Extended Kalman Filter* (IEKF) used for continuous-time systems with discrete observations [91].

(4) Particle Filter: Early in the nineties, the particle filter gained popularity and has been utilized to resolve estimate problems [92]. It acts as an approximate numerical solution to nonlinear Bayesian filtering problems [93], where the posterior is approximated by a set of M samples called *particles* where M is usually a large number (e.g., 2000). Under the context of localization, it is also known as *Monte Carlo localization* (MCL) [94]. Each particle has associated a *weight*, also called the *importance factor*, that represents the contribution to the overall estimate of the posterior, as mentioned in the study [95]. Thus, it shows the belief at time t approximated by a set of particles and weights of M particles [96-98].

7. PARTICLE FILTER APPLIED TO ROBOT LOCALIZATION

Many variants of particle filters have been popularized lately for addressing robot localization. Zhou et al. [99] proposed an improved PF algorithm called the *Pearson Particles Filter* (PPF) because it is based on the Pearson Correlation Coefficient (PCC). PCC is a statistical technique to determine the linear dependence between two random variables, and thus, it is used to decide how close the hypothetical particle state is to the true state value. Even though this technique solves the degeneracy and sample impoverishment problem present in PF, it is computationally complex, as mentioned in the study [95]. Below are some of the Particle Filter Applied:

(a) Differentiable Particle Filter: Jonschkowski and Brock [100] presented the *Differentiable Particle Filters* (DPFs), a differentiable version of the classic particle filter algorithm with end-to-end learnable models. The recursive state prediction and correction steps are encoded in the DPFs structure, which is made of a recurrent network representing the filtering loop. Even though their experiments reduce the error rate by 80% compared to algorithmic priors, they present the limitation of resampling as a non-differentiable operation that stops the gradient computation after a single loop iteration, limiting the scope of the implementation to supervised learning.

(b) Discriminative Particle Filter Reinforcement Learning: Ma et al. [101] introduced a reinforcement learning framework for complex partial observations called *Discriminative Particle Filter Reinforcement Learning* (DPFRL). It uses a DPF in the neural network policy for reasoning with partial observations over time. It is composed of two main components: a discriminatively trained particle filter that tracks a latent belief and an actor-network that learns a policy given the belief. DPFRL is benchmarked using different problems in different domains. First, the Mountain Hike Task [102]. Consisting of an agent that goes from a start to a goal position in a map that contains an irregular terrain with partial visibility, showing that DPFRL learns faster compared to other models such as *Deep Variational Reinforcement Learning* (DVRL) [102] and *Gated Recurrent Unit* (GRU) [103].

(c) Particles Filter Recurrent Neural Networks: Ma et al. [101] extended Recurrent Neural Networks (RNN) to use particle filters. This is different from RNNs that approximate the belief as a long latent vector, updating it using a deterministic nonlinear function, *Particles Filter Recurrent Neural Networks* (PF-RNNs) approximate the belief using a set of weighted particles and the stochastic particle filter algorithm to update them. PF-RNNs are applied to LSTM and GRU RNNs architectures, called PF-LSTM and PF-GRU, respectively. They are evaluated for the robot localization task in three custom 2D synthetic symmetric maps.

8. PROBABILISTIC ROBOTICS

Probabilistic robotics is a promising field that has the potential to revolutionize the field of mobile robotics. Probabilistic robotics employs probabilistic techniques and algorithms to allow robots to make better decisions in the face of uncertainty. The key idea behind probabilistic robotics is to use probability theory to model uncertainty and to develop algorithms that can reason about this uncertainty. This is particularly important in real-world environments, where sensors can be noisy, and the robot's state and environment can be uncertain. Probabilistic robotics is based on three main components: perception, control, and state estimation. Perception refers to the process of sensing the environment and extracting information from the sensors. Control refers to the process of selecting actions that achieve some objective. State estimation refers to the process of estimating the state of the robot and its environment. Probabilistic techniques and algorithms are applied in perception, control, and state estimation in the context of mobile robotics. For example, in perception, probabilistic techniques such as Bayesian inference and Markov models are used to estimate the probability distribution of the robot's environment based on sensor measurements. In control, probabilistic techniques such as optimal control and reinforcement learning are used to select actions that maximize some objective function. In state estimation, probabilistic techniques such as Kalman filters and particle filters are used to estimate the probability distribution of the robot's state based on sensor measurements.

Probabilistic robotics has been utilized in various mobile robot control tasks, such as robot localization and obstacle avoidance. For example, in robot localization, probabilistic techniques such as particle filters and Kalman filters have been used to estimate the robot's position and orientation based on sensor measurements. In obstacle avoidance, probabilistic techniques such as fuzzy logic and Bayesian networks have been used to detect and avoid obstacles in the robot's environment. probabilistic robotics is a promising field that has the potential to revolutionize the field of mobile robotics. Probabilistic techniques and algorithms are applied in perception, control, and state estimation in the context of mobile robotics, enabling robots to make better decisions in the face of uncertainty. Probabilistic robotics has been successfully applied in various mobile robot control tasks, such as robot localization and obstacle avoidance, making it a valuable tool for creating intelligent and adaptive mobile robots.

This section explains the critical and relevant concepts from *probabilistic robotics* applied to the current research. Probabilistic robotics is a promising field that has the potential to revolutionize the field of robotics. Probabilistic robotics has several applications, including autonomous driving and mobile robotics. In mobile robotics, probabilistic robotics is used to localize the robot in the environment and to plan paths that avoid obstacles. It employs probabilistic techniques and algorithms to allow robots to make better decisions in the face of uncertainty, as mentioned in the study [104]. Probabilistic robotics is based on three main components:

(1) Perception: Perception refers to the process of sensing the environment and extracting information from the sensors [105].

(2) Control: Control refers to the process of selecting actions that achieve some objective.

(3) State estimation: State estimation refers to the process of estimating the state of the robot and its environment.

The key idea behind probabilistic robotics is to use probability theory to model uncertainty and to develop algorithms that can reason about this uncertainty. In traditional robotics, uncertainty is often treated as a nuisance that needs to be eliminated. In probabilistic robotics, uncertainty is explicitly modelled and used to improve decision-making, as mentioned in the study [100]. This is particularly important in real-world environments, where sensors can be noisy, and environments can be dynamic and unpredictable. Probabilistic robotics provides a robust framework for dealing with the challenges of uncertainty in real-world robotic applications. Robots can make more informed decisions by modelling and accounting for uncertainty, improving performance and reliability.

9. STATE

According to the study [75], the notion of State refers to the set of all aspects of the robot and its environment that can influence the future. Some instances of state are:

- Static state does not change its position over time. Such as the walls or other moveless object locations.
- Dynamic state changes its position over time. For instance, people or other robot locations.
- The robot pose is usually defined by its position (Localization) and orientation (rotation) relative to a global reference frame. A robot moves on a fixed frame attached to the ground and does not move. This frame is called the global reference frame (GRF). Additionally, a robot is linked within a frame that moves along with the robot. This frame is referred to as a local reference frame (LRF) [106]. Communication between the coordinate frames is known as the transformation of frames, and it is a crucial concept when modelling and programming a robot [107]. The difference between GRF and LRF is that the XR axis points to the robot's right side, the YR axis is aligned to its longitudinal axis, and the ZR axis indicates upward. The notation used to represent a state at time t will be denoted as xt. The control data is information about the change of state in the environment [108]. It can be, for instance, the velocity or the acceleration of the robot at a given time t, and thus it will be represented as ut. The sensor data provided at time t is denoted as zt, and therefore, a collection of percepts, called here a sequence of sensor observations, will be denoted as z1:t=z1, z2, ..., zt. In Figure 2, the comparison between the local and global reference frames illustrates the distinctive characteristics and relationships between these two frames of reference within the context of the study.

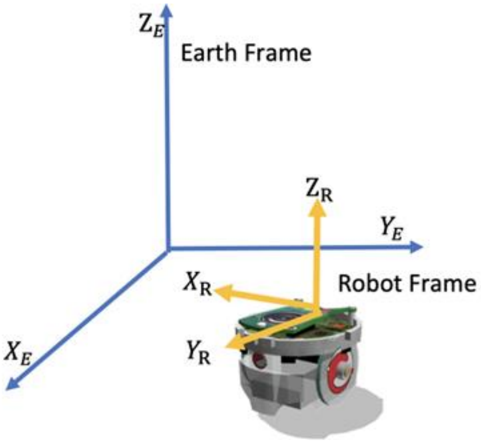

Figure 2. Local vs. Global reference frame

10. DIFFERENTIABLE PROGRAMMING

Differentiable programming (DP) is a programming paradigm that enables the automatic differentiation of algorithms. DP is a combination of the ideas of automatic differentiation and functional programming. DP has demonstrated that it is more than just a compilation of machine learning algorithms. DP can be applied to various types of robots, including mobile robots, manipulators, and humanoid robots. DP can be used for tasks such as trajectory planning, motion control, and reinforcement learning algorithms. DP works by allowing the user to define a function that maps inputs to outputs. The function can be a complex algorithm that involves multiple operations, such as matrix multiplication, convolution, and activation functions. DP then automatically computes the gradient of the function with respect to its inputs using the chain rule of calculus. The gradient can be used to optimize the function using gradient-based methods such as stochastic gradient descent (SGD). DP has several advantages over traditional programming approaches for controlling robots. DP enables the efficient optimization of control policies using gradient-based methods, which can lead to faster convergence and better performance. DP also allows the integration of sensor feedback into the control policy, which can improve the robot's ability to adapt to changing environments. DP is a promising technology that has the potential to revolutionize the field of robotics.

DP has been successfully applied in various mobile robot control tasks, such as trajectory planning, motion control, and reinforcement learning algorithms. For example, in trajectory planning, DP can be used to optimize trajectories by differentiating through the trajectory generation algorithm. The objective function can be defined as a function of the robot's state and the control inputs. DP can be used to compute the gradient of the objective function with respect to the control inputs, which can be used to update the control policy. In motion control, DP can be used to optimize the control policy by differentiating through the dynamics of the robot. DP can be used to compute the gradient of the control policy with respect to the robot's state and the control inputs, which can be used to update the control policy. In reinforcement learning, DP can be used to optimize the Reinforcement Learning algorithm by differentiating through the policy evaluation and improvement steps. DP can be used to compute the gradient of the policy with respect to the parameters of the policy, which can be used to update the policy. DP is a promising technology that enables the automatic differentiation of algorithms.

In order to understand what *differentiable programming* (DP). The study [35] is restated here. DP is a programming paradigm that enables the automatic differentiation of algorithms. DP is a combination of the ideas of automatic differentiation and functional programming. It has demonstrated that it is more than just a compilation of machine learning algorithms [109]. DP can be applied to various types of robots [110], including mobile robots, manipulators, and humanoid robots. DP can be used for tasks. In this section, we will discuss some of the applications of DP in controlling robots.

(1) Trajectory Planning: Trajectory planning is the process of generating a sequence of robot poses that satisfy some objective. It is an essential problem facing mobile robotics [111] and is computationally extremely costly [112]. DP can be used to optimize trajectories by differentiating through the trajectory generation algorithm. The objective function can be defined as a function of the robot's state and the control inputs. DP can be used to compute the gradient of the objective function with respect to the control inputs, which can be used to update the control policy.

(2) Motion Control: Motion control is the process of controlling the robot's motion to achieve some objective [113]. DP can be used to optimize the control policy by differentiating through the dynamics of the robot.

(3) Reinforcement Learning: DP can be used to optimize the Reinforcement Learning algorithm by differentiating through the policy evaluation and improvement steps.

DP has several advantages over traditional programming approaches for controlling robots. DP enables the efficient optimization of control policies using gradient-based methods [114], which can lead to faster convergence and better performance. DP also allows the integration of sensor feedback into the control policy, which can improve the robot's ability to adapt to changing environments. DP is a promising technology that has the potential to revolutionize the field of robotics. However, there are some limitations of DP that need to be addressed. DP requires the algorithm to be differentiable, which can limit the complexity of the algorithm that can be used. DP also requires the computation of the Jacobian matrix, which can be computationally expensive for large systems.

11. CONCLUSION

The reviewed research articles related to AI techniques in mobile robot control have made significant contributions to the field. The main findings of these studies include the successful application of AI techniques such as machine learning, neural networks, and probabilistic robotics in various mobile robot control tasks such as localization, mapping, path planning, and obstacle avoidance. The research has shown that AI techniques can significantly improve the performance of mobile robots in real-world environments by enabling them to make better decisions in the face of uncertainty. The use of probabilistic techniques and algorithms has been particularly effective in addressing the challenges of uncertainty and noise in mobile robot control. The reviewed research articles have also highlighted the importance of integrating sensor feedback into the control policy of mobile robots. This can improve the robot's ability to adapt to changing environments and perform tasks accurately. Overall, the reviewed research articles have demonstrated the potential of AI techniques in mobile robot control and have paved the way for the development of intelligent and adaptive mobile robots. The findings of these studies have significant implications for various industries, including manufacturing, healthcare, and transportation, where mobile robots can be used to perform tasks that are dangerous, tedious, or impossible for humans.

REFERENCES

[1] Fabregas, E., Farias, G., Peralta, E., Vargas, H., Dormido, S. (2016). Teaching control in mobile robotics with V-REP and a Khepera IV library. In 2016 IEEE Conference on Control Applications (CCA), Buenos Aires, Argentina, pp. 821-826. https://doi.org/10.1109/CCA.2016.7587920

[2] BASHEER, M.M., ÇAVAŞ, M., QASIM, M. (2022). Mobile robot indoor localization using color-coded beacons and a depth camera. Turkish Journal of Science and Technology, 17(1): 37-46. https://doi.org/10.55525/tjst.1054213

[3] Guo, L., Cheng, K., Jia, L. (2023). Design and research of robot image perception system based on artificial intelligence. In Journal of Physics: Conference Series. IOP Publishing, 2493(1): 012014. https://doi.org/10.1088/1742-6596/2493/1/012014

[4] Rosenberger, P., Cosgun, A., Newbury, R., Kwan, J., Ortenzi, V., Corke, P., Grafinger, M. (2020). Object-independent human-to-robot handovers using real time robotic vision. IEEE Robotics and Automation Letters,

6(1): 17-23. https://doi.org/10.1109/LRA.2020.3026970

[5] Yi, S.Y., Choi, B.W. (2007). Autonomous navigation of indoor mobile robot using global ultrasonic system. INTECH Open Access Publisher.

[6] Bogoiavlenskaia, O., Korzun, D. (2021). Intelligent data selection in autonomous robot movement. In 2021 29th Conference of Open Innovations Association (FRUCT), Tampere, Finland, pp. 49-54. https://doi.org/10.23919/FRUCT52173.2021.9435576

[7] Cotera, P., Velazquez, M., Cruz, D., Medina, L., Bandala, M. (2016). Indoor robot positioning using an enhanced trilateration algorithm. International Journal of Advanced Robotic Systems, 13(3): 110. https://doi.org/10.5772/63246

[8] Dinnouti, L.S., Victorino, A.C., Silveira, G.F. (2004). Simultaneous localization and map building by a mobile robot using sonar sensors. In ABCM Symposium Series in Mechatronics, 1: 115-123.

[9] Basavanna, M., Shivakumar, M., Prakash, K.R. (2022). Navigation of mobile robot through mapping using Orbbec Astra camera and ROS in an indoor environment. In Recent Advances in Manufacturing, Automation, Design and Energy Technologies: Proceedings from ICoFT 2020, Springer Singapore, pp. 465-474. https://doi.org/10.1007/978-981-16-4222-7_53

[10] Cebollada, S., Payá, L., Flores, M., Peidró, A., Reinoso, O. (2021). A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems with Applications, 167: 114195. https://doi.org/10.1016/j.eswa.2020.114195

[11] Khan, M.S.A., Hussian, D., Ali, Y., Rehman, F.U., Aqeel, A.B., Khan, U.S. (2021). Multi-Sensor SLAM for efficient navigation of a mobile robot. In 2021 4th International Conference on Computing & Information Sciences (ICCIS), Karachi, Pakistan, pp. 1-5. https://doi.org/10.1109/ICCIS54243.2021.9676374

[12] Mando, R., Özer, E., İnner, B. (2022). Autonomous mobile robot mapping and exploration of an indoor environment. In International Conference on Computing, Intelligence and Data Analytics. Cham: Springer International Publishing, pp. 17-26. https://doi.org/10.1007/978-3-031-27099-4_2

[13] Liu, Y. (2022). Localization and navigation system for indoor mobile robot. arXiv Preprint arXiv: 2212.06391. https://doi.org/10.48550/arXiv.2212.06391

[14] Reinoso, O., Paya, L. (2020). Special issue on mobile robots navigation. Applied Sciences, 10(4): 1317. https://doi.org/10.3390/app10041317

[15] Khan, H., Khatoon, S., Gaur, P., Abbas, M., Saleel, C.A., Khan, S.A. (2023). Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller. Processes, 11(4): 1202. https://doi.org/10.3390/pr11041202

[16] Chia, K.S. (2018). Ziegler-nichols based proportional-integral-derivative controller for a line tracking robot. Indonesian Journal of Electrical Engineering and Computer Science, 9(1): 221-226. https://doi.org/10.11591/ijeecs.v9.i1

[17] Albab, A.N., Rahmawati, E., Yantidewi, M., Sucahyo, I., Firmansyah, R.R. (2020). Control position of mobile robot based on odometry method and PID controller. In Journal of Physics: Conference Series. IOP Publishing, 1491(1): 012039. https://doi.org/10.1088/1742-6596/1491/1/012039

[18] YİGİT, S., SEZGİN, A. (2023). Trajectory tracking via backstepping controller with PID or SMC for mobile robots. Sakarya University Journal of Science, 27(1): 120-134. https://doi.org/10.16984/saufenbilder.1148158

[19] Meng, H., Zhang, J., Li, S. (2023). Dual-mode robust model predictive control for the tracking control of nonholonomic mobile robot. International Journal of Robust and Nonlinear Control, 33(6): 3627-3639. https://doi.org/10.1002/rnc.6582

[20] Yeom, K. (2021). Path planning for autonomous driving of mobile robots using deep neural network based model predictive control. International Journal of Emerging Technology and Advanced Engineering, 11(11): 31-38. https://doi.org/10.46338/IJETAE1121_04

[21] Meng, J., Xiao, H., Jiang, L., Hu, Z., Jiang, L., Jiang, N. (2023). Adaptive model predictive control for mobile robots with localization fluctuation estimation. Sensors, 23(5): 2501. https://doi.org/10.3390/s23052501

[22] Shin, J., Hakobyan, A., Park, M., Kim, Y., Kim, G., Yang, I. (2022). Infusing model predictive control into meta-reinforcement learning for mobile robots in dynamic environments. IEEE Robotics and Automation Letters, 7(4): 10065-10072. https://doi.org/10.1109/LRA.2022.3191234

[23] Bruyninckx, H. (2008). Simulation, modeling and programming for autonomous robots: The open source perspective. In Simulation, Modeling, and Programming for Autonomous Robots: First International Conference, SIMPAR 2008 Venice, Italy. Springer Berlin Heidelberg. Proceedings, 1: 1-1. https://doi.org/10.1007/978-3-540-89076-8_1

[24] Hughes, J., Shimizu, M., Visser, A. (2019). A review of robot rescue simulation platforms for robotics education. Robot World Cup, 86-98. https://doi.org/10.1007/978-3-030-35699-6_7

[25] Wu, Y., Wu, Y., Gkioxari, G., Tian, Y. (2018). Building generalizable agents with a realistic and rich 3d environment. arXiv Preprint arXiv: 1801.02209. https://doi.org/10.48550/arXiv.1801.02209

[26] Bertsekas, D.P., Tsitsiklis, J.N., Com, A. (2002). Introduction to Probability. http://www.athenasc.com, accessed on May 30, 2023.

[27] Mathai, A.M., Haubold, H.J. (2017). Probability and statistics: A course for physicists and engineers. De Gruyter, p. 582. https://doi.org/10.1515/9783110562545

[28] Bontempi, G. (2009). Statistical foundations of machine learning. Machine Learning Group, Computer Science Department, Université Libre de Bruxelles, Belgium. https://www.academia.edu/2615389/Statistical_foundations_of_machine_learning, accessed on May 30, 2023.

[29] Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 186(1007): 453-461. https://doi.org/10.1098/rspa.1946.0056

[30] Brooks, R.A. (1991). New approaches to robotics. Science, 253(5025): 1227-1232. https://doi.org/10.1126/science.253.5025.1227

[31] Devi, M.S., Lakshmi, B.S., MadhuLatha, M. (2021). Role of artificial intelligence in future. Science, Technology and Development, 2: 1074-1082. https://doi.org/10.5281/ZENODO.7856202

[32] Möller, D.P.F. (2023). Machine learning and deep learning. In Guide to Cybersecurity in Digital

Transformation, pp. 347-384. https://doi.org/10.1007/978-3-031-26845-8_8

[33] Trivedi, K.S. (2023). Machine learning fundamental concepts. In Microsoft Azure AI Fundamentals Certification Companion, pp. 33-88. https://doi.org/10.1007/978-1-4842-9221-1_3

[34] Arabzadeh Jamali, M., Pham, H. (2023). Statistical machine learning. In Springer Handbook of Engineering Statistics. Springer Handbooks. Springer, London. https://doi.org/10.1007/978-1-4471-7503-2_42

[35] Bishop, C.M., Nasrabadi, N.M. (2006). Pattern recognition and machine learning. Information Science and Statistics. New York: Springer, 4(4): 738.

[36] Goel, A., Goel, A.K., Kumar, A. (2023). The role of artificial neural network and machine learning in utilizing spatial information. Spatial Information Research, 31(3): 275-285. https://doi.org/10.1007/s41324-022-00494-x

[37] Ren, Z., Wang, S., Zhang, Y. (2023). Weakly supervised machine learning. CAAI Transactions on Intelligence Technology, 8(3): 549-580. https://doi.org/10.1049/cit2.12216

[38] Liu, Y., Zhang, Y., Wang, Z., Ma, R., Qiu, F., Coleman, S., Kerr, D. (2023). WUSL-SOD: Joint weakly supervised, unsupervised and supervised learning for salient object detection. Neural Computing and Applications, 1-20. https://doi.org/10.1007/s00521-023-08545-6

[39] Rani, V., Nabi, S.T., Kumar, M., Mittal, A., Kumar, K. (2023). Self-supervised learning: A succinct review. Archives of Computational Methods in Engineering, 30(4): 2761-2775. https://doi.org/10.1007/s11831-023-09884-2

[40] Hsu, K., Levine, S., Finn, C. (2018). Unsupervised learning via meta-learning. arXiv Preprint arXiv: 1810.02334. https://doi.org/10.48550/arXiv.1810.02334

[41] Fung, A., Benhabib, B., Nejat, G. (2023). Robots autonomously detecting people: A multimodal deep contrastive learning method robust to intraclass variations. IEEE Robotics and Automation Letters, 8(6): 3550-3557. https://doi.org/10.1109/LRA.2023.3269306

[42] Ji, Z., Zou, X., Huang, T., Wu, S. (2019). Unsupervised few-shot learning via self-supervised training. arXiv Preprint arXiv: 1912.12178. https://doi.org/10.48550/arXiv.1912.12178

[43] Chen, B., Liu, X., Zhao, H., Principe, J.C. (2017). Maximum correntropy Kalman filter. Automatica, 76: 70-77. https://doi.org/10.1016/j.automatica.2016.10.004

[44] Zarchan, P., Musoff, H. (2015). Fundamentals of kalman filtering: A practical approach, fourth edition. American Institute of Aeronautics and Astronautics, Inc. https://doi.org/10.2514/4.102776

[45] Julier, S.J., Uhlmann, J.K. (1997). New extension of the Kalman filter to nonlinear systems. In Signal Processing, Sensor Fusion, and Target Recognition VI. Spie, 3068: 182-193. https://doi.org/10.1117/12.280797

[46] Aminzadeh, F., Temizel, C., Hajizadeh, Y. (2022). Artificial neural networks. In Artificial Intelligence and Data Analytics for Energy Exploration and Production. Wiley, pp. 39-84. https://doi.org/10.1002/9781119879893.ch3

[47] Puri, M., Solanki, A., Padawer, T., Tipparaju, S.M., Moreno, W.A., Pathak, Y. (2016). Introduction to artificial neural network (ANN) as a predictive tool for drug design, discovery, delivery, and disposition: Basic concepts and modeling. In Artificial Neural Network for Drug Design, Delivery and Disposition. Academic Press, pp. 3-13. https://doi.org/10.1016/B978-0-12-801559-9.00001-6

[48] Ahmed, W., Chaudhary, A., Naqvi, G. (2023). Role of artificial neural networks in AI. https://doi.org/10.14704/nq.2022.20.13.NQ88416

[49] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8: 1-74. https://doi.org/10.1186/s40537-021-00444-8

[50] Sarker, I.H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2(6): 420. https://doi.org/10.1007/s42979-021-00815-1

[51] Hubel, D.H., Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology, 160(1): 106. https://doi.org/10.1113%2Fjphysiol.1962.sp006837

[52] Fang, W., Love, P.E., Luo, H., Ding, L. (2020). Computer vision for behaviour-based safety in construction: A review and future directions. Advanced Engineering Informatics, 43: 100980. https://doi.org/10.1016/j.aei.2019.100980

[53] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6): 84-90. https://doi.org/10.1145/3065386

[54] Mechal, C.E., El Idrissi, N.E.A., Mesbah, M. (2022). CNN-based obstacle avoidance using RGB-Depth image fusion. In WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems. Springer Singapore, pp. 867-876. https://doi.org/10.1007/978-981-33-6893-4_78

[55] Liu, C., Zheng, B., Wang, C., Zhao, Y., Fu, S., Li, H. (2017). CNN-based vision model for obstacle avoidance of mobile robot. In MATEC Web of Conferences. EDP Sciences, Vol. 139. https://doi.org/10.1051/matecconf/201713900007

[56] Garcia, A., Mittal, S.S., Kiewra, E., Ghose, K. (2019). A convolutional neural network vision system approach to indoor autonomous quadrotor navigation. In 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, pp. 1344-1352. https://doi.org/10.1109/ICUAS.2019.8798183

[57] Wei, W., He, S., Wang, D., Yeboah, Y. (2018). Multi-objective deep cnn for outdoor auto-navigation. In Proceedings of the 2018 2nd International Conference on Deep Learning Technologies, pp. 81-85. https://doi.org/10.1145/3234804.3234823

[58] Wang, W., Zhang, B., Wu, K., Chepinskiy, S.A., Zhilenkov, A.A., Chernyi, S., Krasnov, A.Y. (2022). A visual terrain classification method for mobile robots' navigation based on convolutional neural network and support vector machine. Transactions of the Institute of Measurement and Control, 44(4): 744-753. https://doi.org/10.1177/0142331220987917

[59] Jokić, A., Đokić, L., Petrović, M., Miljković, Z. (2022). Data augmentation methods for semantic segmentation-based mobile robot perception system. Serbian Journal of

Electrical Engineering, 19(3): 291-302. http://dx.doi.org/https://doi.org/10.2298/SJEE2203291J

[60] Shelhamer, E., Long, J., Darrell, T. (2016). Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4): 640-651. https://doi.org/10.1109/TPAMI.2016.2572683

[61] Kendall, A., Grimes, M., Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, pp. 2938-2946. https://doi.org/10.1109/ICCV.2015.336

[62] Cen, H. (2023). Target location detection of mobile robots based on R-FCN deep convolutional neural network. International Journal of System Assurance Engineering and Management, 14(2): 728-737. https://doi.org/10.1007/s13198-021-01514-z

[63] Liu, F., Shen, C., Lin, G. (2015). Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 5162-5170. https://doi.org/10.1109/CVPR.2015.7299152

[64] Petrović, M., Miljković, Z., Jokić, A. (2023). Efficient machine learning of mobile robotic systems based on convolutional neural networks. In Artificial Intelligence for Robotics and Autonomous Systems Applications. Cham: Springer International Publishing, pp. 1-26. https://doi.org/10.1007/978-3-031-28715-2_1

[65] Nagaraj, A., Sood, M., Patil, B.M. (2022). A concise introduction to reinforcement learning in robotics. arXiv Preprint arXiv: 2210.07397. https://doi.org/10.48550/arXiv.2210.07397

[66] Mohammed, M., Khan, M.B., Bashier, E.B.M. (2016). Machine learning: Algorithms and applications. Crc Press, pp. 1-204. https://doi.org/10.1201/9781315371658

[67] Elguea-Aguinaco, Í., Serrano-Muñoz, A., Chrysostomou, D., Inziarte-Hidalgo, I., Bøgh, S., Arana-Arexolaleiba, N. (2023). A review on reinforcement learning for contact-rich robotic manipulation tasks. Robotics and Computer-Integrated Manufacturing, 81: 102517. https://doi.org/10.1016/j.rcim.2022.102517

[68] Sarker, I.H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3): 160. https://doi.org/10.1007/s42979-021-00592-x

[69] Tariverdi, A., Côté-Allard, U., Mathiassen, K., Elle, O.J., Kalvoy, H., Martinsen, Ø.G., Torresen, J. (2023). Reinforcement learning-based switching controller for a milliscale robot in a constrained environment. IEEE Transactions on Automation Science and Engineering. https://doi.org/10.1109/TASE.2023.3259905

[70] Bruce, J., Sünderhauf, N., Mirowski, P., Hadsell, R., Milford, M. (2017). One-shot reinforcement learning for robot navigation with interactive replay. arXiv Preprint arXiv: 1711.10137. https://doi.org/10.48550/arXiv.1711.10137

[71] Lee, M.F.R., Yusuf, S.H. (2022). Mobile robot navigation using deep reinforcement learning. Processes, 10(12): 2748. https://doi.org/10.3390/pr10122748

[72] Boularias, A., Bagnell, J., Stentz, A. (2015). Learning to manipulate unknown objects in clutter by reinforcement. In Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). https://doi.org/10.1609/aaai.v29i1.9378

[73] Koza, J.R., Poli, R. (2005). Genetic programming. In Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, pp. 127-164. https://doi.org/10.1007/0-387-28356-0_5

[74] Miller, J.F., Thomson, P. (2000). Cartesian genetic programming. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol 1802. https://doi.org/10.1007/978-3-540-46239-2_9

[75] Luo, C., Zhang, S.L. (2012). Parse-matrix evolution for symbolic regression. Engineering Applications of Artificial Intelligence, 25(6): 1182-1193. https://doi.org/10.1016/j.engappai.2012.05.015

[76] Diveev, A.I., Sofronova, E.A. (2009). Numerical method of network operator for multiobjective synthesis of optimal control system. In 2009 IEEE International Conference on Control and Automation, Christchurch, New Zealand, pp. 701-708. https://doi.org/10.1109/ICCA.2009.5410619

[77] Diveev, A., Sofronova, E. (2020). Automation of synthesized optimal control problem solution for mobile robot by genetic programming. In Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys). Springer International Publishing, 2: 1054-1072. https://doi.org/10.1007/978-3-030-29513-4_77

[78] Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D. (2019). Habitat: A platform for embodied AI research. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), pp. 9339-9347. https://doi.org/10.1109/ICCV.2019.00943

[79] Huang, S., Huang, H.Z., Zeng, Q. (2021). A robust localization method in indoor dynamic environment. In IOP Conference Series: Materials Science and Engineering. IOP Publishing, 1043(5): 052025. https://doi.org/10.1088/1757-899X/1043/5/052025

[80] Amigoni, F., Schiaffonati, V. (2010). Good experimental methodologies and simulation in autonomous mobile robotics. In Model-Based Reasoning in Science and Technology: Abduction, Logic, and Computational Discovery. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 315-332. https://doi.org/10.1007/978-3-642-15223-8_18

[81] Wang, S., Wang, Y., Li, D., Zhao, Q. (2023). Distributed relative localization algorithms for multi-robot networks: A survey. Sensors, 23(5): 2399. https://doi.org/10.3390/s23052399

[82] Chen, S., Yin, D., Niu, Y. (2022). A survey of robot swarms' relative localization method. Sensors, 22(12): 4424. https://doi.org/10.3390/s22124424

[83] Fox, V., Hightower, J., Liao, L., Schulz, D., Borriello, G. (2003). Bayesian filtering for location estimation. IEEE Pervasive Computing, 2(3): 24-33. https://doi.org/10.1109/MPRV.2003.1228524

[84] Choi, D.K. (2023). Motion tracking of four-wheeled mobile robots in outdoor environments using bayes' filters. International Journal of Precision Engineering and Manufacturing, 24(5): 767-786. https://doi.org/10.1007/s12541-023-00777-9

[85] Kalman, R.E. (1960). A new approach to linear filtering

and prediction problems. Journal of Basic Engineering, 82(1): 35-45. https://doi.org/10.1115/1.3662552

[86] Barrau, A., Bonnabel, S. (2018). Invariant kalman filtering. Annual Review of Control, Robotics, and Autonomous Systems, 1: 237-257. https://doi.org/10.1146/annurev-control-060117-105010

[87] Urrea, C., Agramonte, R. (2021). Kalman filter: Historical overview and review of its use in robotics 60 years after its creation. Journal of Sensors, 2021: 1-21. https://doi.org/10.1155/2021/9674015

[88] Padmakumar, S., Agarwal, V., Roy, K. (2009). A tutorial on dynamic simulation of dc motor and implementation of kalman filter on a floating point dsp. World Academy of Science, Engineering and Technology, 53(126): 781-786.

[89] Laskar, M.N.U., Tawhid, M.N.A., Chung, T. (2012). Extended kalman filter (EKF) and K-means clustering approach for state space decomposition of autonomous mobile robots. In 2012 7th International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, pp. 113-116. https://doi.org/10.1109/ICECE.2012.6471498

[90] Wan, E.A., Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), Lake Louise, AB, Canada, pp. 153-158. https://doi.org/10.1109/ASSPCC.2000.882463

[91] Barrau, A., Bonnabel, S. (2016). The invariant extended Kalman filter as a stable observer. IEEE Transactions on Automatic Control, 62(4): 1797-1812. https://doi.org/10.1109/TAC.2016.2594085

[92] Elfring, J., Torta, E., van de Molengraft, R. (2021). Particle filters: A hands-on tutorial. Sensors, 21(2): 438. https://doi.org/10.3390/s21020438

[93] Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. IEEE Aerospace and Electronic Systems Magazine, 25(7): 53-82. https://doi.org/10.1109/MAES.2010.5546308

[94] Thrun, S., Fox, D., Burgard, W., Dellaert, F. (2001). Robust monte carlo localization for mobile robots. Artificial Intelligence, 128(1-2): 99-141. https://doi.org/10.1016/S0004-3702(01)00069-8

[95] Ma, X., Karkus, P., Hsu, D., Lee, W.S. (2020). Particle filter recurrent neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(04): 5101-5108. https://doi.org/10.1609/aaai.v34i04.5952

[96] Stachniss, C., Burgard, W. (2014). Particle filters for robot navigation. Foundations and Trends® in Robotics, 3(4): 211-282. http://dx.doi.org/10.1561/2300000013

[97] Thrun, S. (2002). Particle filters in robotics. In Proceedings of Uncertainty in AI (UAI), 2: 511-518.

[98] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., Nordlund, P.J. (2002). Particle filters for positioning, navigation, and tracking. IEEE Transactions on Signal Processing, 50(2): 425-437. https://doi.org/10.1109/78.978396

[99] Zhou, H., Deng, Z., Xia, Y., Fu, M. (2016). A new sampling method in particle filter based on Pearson correlation coefficient. Neurocomputing, 216: 208-215. https://doi.org/10.1016/j.neucom.2016.07.036

[100] Jonschkowski, R., Brock, O. (2016). End-to-end learnable histogram filters.

[101] Ma, X., Karkus, P., Hsu, D., Lee, W.S., Ye, N. (2020). Discriminative particle filter reinforcement learning for complex partial observations. arXiv Preprint arXiv: 2002.09884. https://doi.org/10.48550/arXiv.2002.09884

[102] Igl, M., Zintgraf, L., Le, T.A., Wood, F., Whiteson, S. (2018). Deep variational reinforcement learning for POMDPs. In 35th International Conference on Machine Learning, PMLR, pp. 2117-2126. https://doi.org/10.48550/arXiv.1806.02426

[103] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv Preprint arXiv: 1406.1078. https://doi.org/10.48550/arXiv.1406.1078

[104] Thrun, S., Burgard, W., Fox, D. (2005). Probabilistic robotics. In Intelligent Robotics and Autonomous Agents. The MIT Press, pp. 52. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Probabilistic+Robotics+(Intelligent+Robotics+and+Autonomous+Agents)#0, accessed on Jun. 05, 2023.

[105] Shabbir, J., Anwer, T. (2018). A survey of deep learning techniques for mobile robot applications. arXiv Preprint arXiv: 1803.07608. https://doi.org/10.48550/arXiv.1803.07608

[106] Alencastre-Miranda, M., Munoz-Gomez, L., Murrieta-Cid, R., Monroy, R. (2006). Local reference frames vs. global reference frame for mobile robot localization and path planning. In 2006 Fifth Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, pp. 309-318. https://doi.org/10.1109/MICAI.2006.27

[107] Jazar, R.N. (2010). Theory of applied robotics: Kinematics, dynamics, and control. Springer. https://doi.org/10.1007/978-1-4419-1750-8

[108] Rosenbaum, D., Besse, F., Viola, F., Rezende, D.J., Eslami, S.M. (2018). Learning models for visual 3d localization with implicit mapping. arXiv Preprint arXiv: 1807.03149. https://doi.org/10.48550/arXiv.1807.03149

[109] Sajovic, Ž., Vuk, M. (2016). Operational calculus for differentiable programming. arXiv Preprint arXiv: 1610.07690. https://doi.org/10.48550/arXiv.1610.07690

[110] Dawson, C., Fan, C. (2022). Certifiable robot design optimization using differentiable programming. arXiv Preprint arXiv: 2204.10935. https://doi.org/10.48550/arXiv.2204.10935

[111] Contreras-Cruz, M.A., Ayala-Ramirez, V., Hernandez-Belmonte, U.H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. Applied Soft Computing, 30: 319-328. https://doi.org/10.1016/j.asoc.2015.01.067

[112] Nagib, G., Gharieb, W. (2004). Path planning for a mobile robot using genetic algorithms. International Conference on Electrical, Electronic and Computer Engineering, 2004, ICEEC '04. https://doi.org/10.1109/ICEEC.2004.1374415

[113] Yu, Y., Wen, K., Yi, X., Chuang, G., Wei, J. (2023). Research on navigation motion control method of intelligent delivery robot for substation maintenance. In Journal of Physics: Conference Series. IOP Publishing, 2433(1): 012005. https://doi.org/10.1088/1742-6596/2433/1/012005

[114] Verma, K., Maiti, A. (2023). WSAGrad: A novel adaptive gradient based method. Applied Intelligence,

53(11): 14383-14399. https://doi.org/10.1007/s10489-022-04205-9