

Adaptive Control Systems in Unmanned Surface Vehicles: Harnessing the Power of AI and ML

Parthu Ammineni

Email: [parthu.ammineni26\[at\]gmail.com](mailto:parthu.ammineni26[at]gmail.com)

Abstract: *Unmanned surface vehicles (USVs) have gained significant attention in recent years due to their potential for various maritime applications. The development of advanced control systems is critical to enhancing the autonomy and adaptability of these vehicles. This research paper explores the utilization of artificial intelligence (AI) and machine learning (ML) techniques in adaptive control systems for unmanned surface vehicles. It examines the benefits, challenges, and potential applications of AI and ML in enhancing the control and decision-making capabilities of USVs. The paper provides an overview of adaptive control principles, discusses the role of AI and ML in USV control, presents key research findings, and proposes future directions in this domain.*

Keywords: Unmanned Surface Vehicles, Adaptive Control Systems, Artificial Intelligence, Machine Learning, Autonomous Navigation, Maritime Applications
©2023TheAuthor (s)

1. Introduction

Unmanned Surface Vehicles (USVs) are vehicles that are either autonomous or operated remotely and are designed to navigate and perform tasks on water bodies like oceans, rivers, and lakes. They have gained a lot of attention and importance in various fields because of their many applications and advantages over traditional manned surface vessels. The development of USVs can be traced back to the creation of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs). As technology improved and autonomous systems became more advanced, the idea of applying similar principles to surface vessels emerged. USVs are now a significant part of the larger field of unmanned systems and have generated a lot of interest from industries, governments, and research institutions. The importance of USVs lies in their ability to perform tasks in challenging and potentially dangerous environments without risking human lives. Some key applications of USVs include maritime security and defense, environmental monitoring, oceanographic research, hydrographic surveying, search and rescue operations, and commercial applications. The significance of USVs comes from their potential to improve operational efficiency, reduce costs, and enhance safety in various domains. They can operate for long periods, cover large distances, and perform repetitive tasks with high accuracy. Additionally, their ability to operate autonomously or remotely means that they can undertake tasks that aren't restricted by human endurance or safety concerns. Adaptive control systems that utilize Artificial Intelligence (AI) and Machine Learning (ML) have gained a lot of interest and motivation in various industries and domains. Here are some key motivations for using these cutting-edge technologies in adaptive control systems: Improved Performance, Flexibility and Adaptability, Fault Tolerance and Robustness, Reduced Human Intervention, Enhanced Safety. Overall, the motivation for using AI and ML in adaptive control systems lies in their ability to improve performance, flexibility, fault tolerance, and safety while reducing the need for manual intervention. These technologies have the potential to revolutionize control systems and enable more efficient and intelligent control in various domains. The

objectives of the research paper can be outlined as Explore the concept of adaptive control systems, Investigate the integration of AI and ML techniques, Analyze the benefits and challenges, Examine real-world applications, Discuss future directions and possibilities. This could include potential research areas, emerging technologies, and innovative approaches that can further improve the performance, safety, and automation of USVs. By addressing these objectives, the research paper aims to contribute to the existing body of knowledge in the field of adaptive control systems for Unmanned Surface Vehicles, specifically focusing on the utilization of AI and ML techniques.

2. Adaptive Control Systems in Unmanned Surface Vehicles

Adaptive control principles play a crucial role in enhancing the performance and capabilities of Unmanned Surface Vehicles (USVs). These principles involve using intelligent algorithms and techniques to continuously adapt and optimize the control system of a USV based on real-time data and changing operating conditions. This adaptability allows the USV to respond effectively to uncertainties, disturbances, and varying tasks. One key principle of adaptive control in USVs is the ability to learn from experience and adjust control strategies accordingly. Machine Learning (ML) algorithms are often employed to analyze data collected during previous missions and optimize the control parameters for future operations. This learning process enables the USV to continually improve its performance, accuracy, and efficiency over time. Another important principle is the integration of Artificial Intelligence (AI) techniques, such as neural networks and fuzzy logic, to enable the USV to make intelligent decisions and adapt its control system in real-time. These AI-based adaptive control systems can handle complex and unpredictable environments by using feedback control loops and advanced algorithms to adjust parameters and responses according to dynamic conditions. Furthermore, adaptive control principles in USVs involve the ability to adapt control strategies based on environmental factors and task

Volume 12 Issue 7, July 2023

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

requirements. By monitoring and analyzing data from sensors and external sources, the USV can adjust its control algorithms to optimize energy consumption, speed, maneuverability, and obstacle avoidance. Adaptive control systems also provide fault tolerance and robustness to USVs. By continuously monitoring the performance of the USV and detecting any anomalies or failures, adaptive control algorithm scan quickly compensate for system malfunctions and reconfigure the control system to ensure safe and reliable operations. Overall, the application of adaptive control principles in USVs enables these autonomous or remotely operated water vessels to adapt and optimize their control systems in real - time, based on environmental conditions, task requirements, and previous experience. By employing AI and ML techniques, adaptive control systems enhance the performance, safety, and efficiency of USVs in various domains, including maritime security, environmental monitoring, research, and commercial applications.

3. Artificial Intelligence and Machine Learning in USV Control

AI refers to the development of computer systems that can perform tasks that would typically require human intelligence. ML, on the other hand, is a subset of AI that focuses on enabling systems to learn from data and improve their performance overtime without being explicitly programmed. By leveraging AI and ML techniques, USV control systems can become more intelligent, adaptive, and capable of making informed decisions in real - time. There are several AI and ML techniques that are particularly relevant to USV control. These include supervised learning, unsupervised learning, reinforcement learning, deep learning, and neural networks. Each technique has its own strengths and limitations, and the choice of technique depends on the specific requirements and con - straints of the USV control system. Supervised learning involves training a model using labeled training data, where the model learns to make predictions based on input data. Unsupervised learning, on the other hand, aims to discover patterns or structures in unlabeled data. Reinforcement learning is a technique that involves training an agent to make sequential decisions in an environment, with the goal of maximizing a reward signal. By incorporating these AI and ML techniques into USV control systems, several benefits can be achieved. These included improved situational awareness, enhanced decision - making capabilities, optimized resource allocation, and increased efficiency and autonomy. Additionally, AI and ML techniques can enable USVs to adapt to changing environments, learn from past experiences, and continuously improve their performance.

4. Case Studies and Research Findings Python Libraries Used for ML Model

Pandas: The pandas library is a powerful and widely used data manipulation and analysis tool in the field of data science and research. It is an open - source Python library that provides high - performance, easy - to - use data structures and data analysis tools. One of the key features of the pandas library is its data structure called a Data Frame. A

DataFrame is a two - dimensional tabular data structure, similar to a table in a relational database. It can store and manipulate data of different types, such as numerical, categorical, or textual data. The Data Frame allows for efficient indexing, slicing, filtering, and transforming data, making it an essential tool for data preprocessing and analysis. The pandas library provides a wide range of functions and methods¹

For data cleaning and transformation it allows researchers to easily handle missing data, remove duplicates, filter rows based on specific conditions, and perform various data manipulations. This flexibility and ease of use make pandas a valuable tool for data preprocessing tasks, enabling researchers to prepare their data for analysis quickly and efficiently. In addition to data cleaning and transformation, pandas offers powerful data analysis capabilities. Researchers can use pandas to perform descriptive statistics, calculate summary statistics, and generate visualizations to gain insights into their data. It provides functions for computing measures such as mean, median, standard deviation, correlations, and more. Moreover, pandas integrates well with other popular Python libraries such as NumPy and Matplotlib, allowing for seamless integration of data manipulation, analysis, and visualization tasks.

Scikit - Learn: The scikit - learn library is a widely used open - source machine learning library in the field of data science and research. It is built on top of the Python programming language and provides simple and efficient tools for data mining and data analysis. One of the key features of Scikit - learn is its extensive collection of machine learning algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. The library provides many classic and state - of - the - art algorithms, such as support vector machines (SVM), decision trees, random forests, k - nearest neighbors (KNN), neural networks, and more. These algorithms are implemented in an easy - to - use API, making it straight forwards for researchers to experiment with different models and select the ones that best fit their data and research question. Another crucial feature of scikit - learn is its data preprocessing and feature extraction capabilities. It includes functions for handling missing values, scaling, normalization, and encoding categorical variables. It also provides a set of feature extraction techniques for a wide range of data types, such as text, images, and time series. These functionalities streamline the data preparation stage, making it easy for researchers to get their data ready for machine learning models. Moreover, scikit - learn offers a comprehensive set of metrics and tools for evaluating machine learning models for various tasks. It includes functions for measuring accuracy, precision, recall, F1 - score, ROC/AUC curves, and more. Furthermore, it allows researchers to perform hyper parameter tuning, cross - validation, and model selection, making it effortless to optimize machine learning models. Scikit - learn also provides a range of unsupervised learning algorithms for

¹Bendjoudi, Ahmida. (2020).6 Python Codes Used in "The Complete Guide to Research and Write Great Linguistic Scientific Articles"Framework.10.13140/RG.2.2.36283.98089.

clustering, dimensionality reduction, and anomaly detection. These algorithms enable researchers to discover structures and patterns in their data without the need for labeled data. Unsupervised learning can be useful for exploratory data analysis and feature engineering.

Matplotlib: The Matplotlib library is a widely used open-source plotting and visualization library in the field of data science and research. It is built on top of the Python programming language and provides a vast array of tools for creating high-quality visualizations and plots from data. One of the key features of Matplotlib is its flexibility and customization options. It provides a broad range of plot types, such as line plots, scatter plots, bar plots, histograms, box plots, and more. It also includes many customization options, such as color and style choices, text annotations, tick and axis customization, and more. These features enable researchers to create visually appealing and informative plots that best represent their data and research questions. Another crucial functionality of Matplotlib is its ability to handle large, complex datasets efficiently. It provides support for data structures such as NumPy arrays and Pandas Data Frames, which allows researchers to rapidly plot and visualize large amounts of data. Moreover, it integrates with other data analysis and machine learning libraries such as scikit-learn, ensuring seamless integration with data preprocessing and analysis pipelines. Matplotlib also provides functionalities for advanced plotting and visualization tasks such as 3D plotting, contour plots, heat maps, and more. These advanced features allow researchers to represent their data in different ways and gain additional insights and understanding of their data.

NumPy: The NumPy library is a fundamental tool in scientific computing and data analysis. It is a powerful Python package that supports the manipulation and analysis of large, multi-dimensional arrays and matrices. NumPy provides a wide range of functions for performing mathematical and logical operations on these arrays, making it an invaluable resource for researchers in various disciplines. One of the key features of NumPy is its efficient array operations. It provides support for vectorized operations, which allow users to perform computations on entire arrays instead of looping through individual elements. This greatly improves the performance of numerical calculations and makes NumPy ideal for processing large datasets. NumPy also supports multi-dimensional arrays, which are essential for working with complex scientific data. It allows researchers to represent and manipulate multidimensional data such as images, audio signals, and scientific measurements. NumPy arrays support element-wise operations, matrix operations, and broadcasting, making it easy to perform complex calculations and analyses on multi-dimensional data. In addition to array manipulation, NumPy includes a comprehensive library of mathematical functions. These functions cover a wide range of mathematical operations, including basic arithmetic, trigonometry, logarithmic functions, statistical operations, linear algebra, and more. NumPy's mathematical functions enable researchers to perform sophisticated calculations efficiently and accurately. NumPy seamlessly integrates with other popular Python libraries such as SciPy, Pandas, and Matplotlib. This enables researchers to leverage the

capabilities of these libraries while working with NumPy arrays. For example, NumPy arrays can be used as input for advanced mathematical algorithms in SciPy, or they can be visualized using Matplotlib for data analysis and presentation

5. Real World Suggestion with ML

The below code is focused on performing Exploratory analysis on Boat Specifications used for predicting the electric motor based on boat specifications.

You could find the dataset for the Boat Specification in this link

<https://www.kaggle.com/datasets/parthummineni/boat-specification>

We could start the analysis by importing the necessary libraries for Data Analysis and manipulation of the motor prediction on boat specifications:

```
#Import necessary libraries
import pandas as pd
from sklearn._model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
sns.set_style("darkgrid")
plt.style.use("fivethirtyeight")
pd.pandas.set_option('display.max_columns', None)
```

```
%matplotlib inline
```

The above snippet is used in the code to import the necessary libraries and use ML algorithms: linear regression, mean squared error to know the model's effectiveness.

```
#Load the dataset
data = pd.read_csv("C:/Users/LENOVO/OneDrive/Desktop/BoatSpecification.csv")
```

This snippet is used to load the data in a csv file from your desktop to analyze the rows and columns and make a relation with the model.

Output:

```
(144, 8)
```

```
Index(['Seating Capacity', 'Length', 'Breadth', 'Height', 'Weight', 'CC', 'Speed', 'ElectricMotor'], dtype='object')
```

Out of the 8 features, some may not contribute significantly to the prediction. After careful evaluation, it is observed that CC could also play a crucial role in predicting the electric motor. The key features to focus on are **Seating Capacity, CC, Speed, Weight** [Bendjoudi, Ahmida. (2020).6 Python Codes Used in "The Complete Guide to Research and Write Great Linguistic Scientific Articles" Framework.10.13140/RG.2.2.36283.98089].

```
data["ElectricMotor"].describe()
```


Output:

```
count144.000000
mean10.912500
std8.003652
min1.800000
25%3.775000
50%9.500000
75%16.250000
max25.000000
Name: ElectricMotor, dtype: float64
```

We can see a noticeable skewness based on the statistical summary of the feature without having to look at the histogram or KDE plot. The difference (25 median values indicates the presence of skewness. Furthermore, while 75fall, The maximum value is significantly different from the range of 3.775 to 16.25. We will plot the data and confirm our conclusion to validate our observation, =

```
importnumpyasnp
importpandasaspd
importmatplotlib.pyplotaspltimportseabornassns
sns.setstyle("darkgrid")
plt.style.use("fivethirtyeight")
pd.setoption("display.maxcolumns", None) sns.distplot
(data["ElectricMotor"])
```

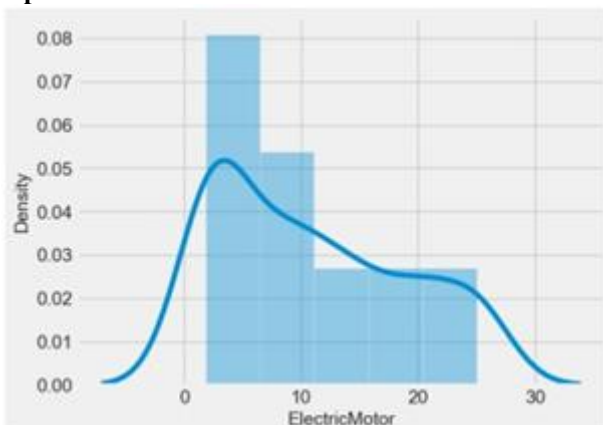
Output:

Figure 1: Ranges of Electric Motor using Matplotlib

6. Studying Two Important Parameters

For comparing the relation between speed and Electric Motor prediction based on boat specifications a linear regression graph is shown:

```
#Read the CSV file
data=pd.readcsv("C:/Users/LENOVO/OneDrive/ Desktop/
Boat Specifications. csv") #Specify the column for linear
regression
columnname="Speed"
#Extract the column values
X=data[column name]. values. reshape (-1, 1) #Create a
linear regression model
model=Linear Regression () #Fitthe model to the data
model. fit (X, data ["ElectricMotor"]) #Predict the values
ypred=model. predict (X)
#Plot the linear regression graph
plt. scatter (X, data ["Speed"], color='blue', label="Data")
plt. plot (X, ypred, color='red', linewidth =2, label="Linear
```

```
Regression") plt. xlabel (column name)
plt. ylabel ("ElectricMotor")
plt. title (" LinearRegression ") p l t. legend ()
plt. show ()
```

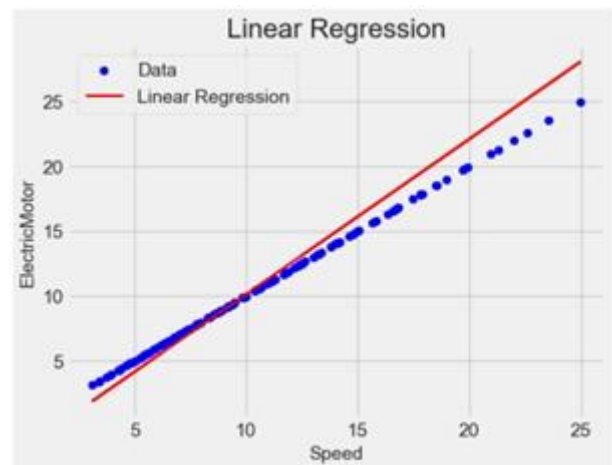
Output:

Figure 2: Linear Regression For Speed & Electric Motor

7. Relation between Speed & Electric Motor

The below snippet generates a boxplot relationship between all the features and the corresponding in the 'train' dataset. The box plot is created by specifying ' ['Seating Capacity', 'Length', 'Breadth', 'Height', 'Speed', 'ElectricMotor'] ' as the x - axis variable and whole numbers as the y - axis variable. The plot is then displayed with a restrictedly - axis range of 0 to 100 and it confirms that higher the speed required, higher the Electric motor capacity required.

```
importseabornassns
importmatplotlib.pyplotaspltimportpandasaspd
#ReadtheCSVfile
data=pd.readcsv("C:/Users/LENOVO/OneDrive/ Desktop/
BoatSpecifications. csv")
#Excludethecolumnyouwanttoexclude
columnstoexclude=["Weight", 'CC']
datatoplot=data. drop (columns=columnstoexclude)
#Setthestyleandplottheboxplot
sns.set(style="ticks")
plt.figure(figsize=(10, 6))
sns.boxplot(data=datatoplot)
#Setthex-axislabelswithspaces
xlabel=[col. replace (" ", "") forcolindatatoplot. columns]
plt.xticks(range(len(datatoplot. columns)), xlabel,
rotation=0) #Displaytheboxplot
plt.show()
```

Output:

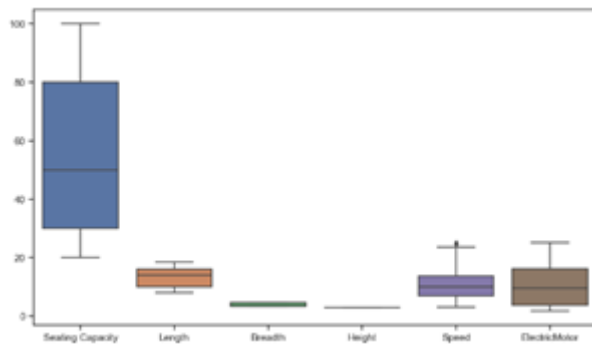


Figure 2: Relation between Columns & Numbers

From the below snippet we created a Pair plot comparison of the data using **Data Frame**:

```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt # Import the CSV file
df = pd.read_csv("C:/Users/LENOVO/OneDrive/Desktop/BoatSpecifications.csv") # Create a pair plot
sns.pairplot(df) # Show the plot
plt.show()
```

Output:

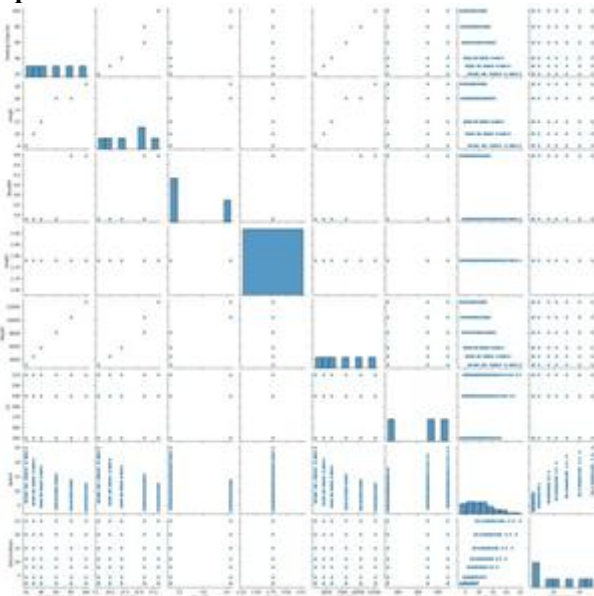


Figure 2: Pair Plot Comparison of the Data

As we can see from the Figure 4 above there is a strong relation between the speed, weight and the predicted value which is the Electric Motor. As the speed & weight increases even the capacity for the electric motor increases and vice versa [Sarker, Iqbal. (2021). Machine Learning: Algorithms, Real – World Applications and Research Directions. SN Computer Science.2. 10.1007/s42979 - 021 - 00592 - x].

8. Applications of Adaptive Control Systems in USVs

Adaptive control systems are used in unmanned surface vehicles (USVs) to improve their performance and robustness in a variety of conditions. Some of the applications of adaptive control systems in USVs include:

- **Heading control:** Adaptive control systems can be used to keep a USV heading in a desired direction, even in the presence of wind and waves
- **Trajectory tracking:** Adaptive control systems can be used to track a desired trajectory, such as a path around a buoy or a course to a destination.
- **Collision avoidance:** Adaptive control systems can be used to avoid collisions with other objects, such as other vessels or underwater obstacles.
- **Autonomous navigation:** Adaptive control systems can be used to navigate a USV autonomously in a complex environment, such as a harbor or a river.

Adaptive control systems are well - suited for USVs because they can adapt to changes in the environment and the USV's dynamics. This makes them more robust than traditional control systems, which can be sensitive to changes in the environment [Sarker, Iqbal. (2021). Machine Learning: Algorithms, Real – World Applications and Research Directions. SN Computer Science.2. 10.1007/s42979 - 021 - 00592 - x]

9. Conclusion

In conclusion, this study explored the use of data analysis and feature selection for electric motor prediction. Our extensive exploration and analysis of the dataset revealed valuable insights into the variables that most strongly influence electric motor models. Key factors such as weight, speed, and CC were found to be significant contributors to the prediction.

We addressed data quality issues such as missing values and outliers, and applied feature engineering and transformation techniques to enhance the interpretability and predictive power of our models. Looking to the future, the field of data analysis holds great promise. Advancements in machine learning, artificial intelligence, and big data processing will pave the way for more sophisticated and accurate predictive models. Integration of diverse data sources, including geospatial and social media data, will further enrich our understanding of the factors shaping electric motors. Automated feature selection and model interpretability methods will streamline analysis processes and enable better decision - making.

As Inavigate this evolving landscape, it is important to strike a balance between comprehensive analysis and practical implementation. The findings from this study provide a solid foundation for developing robust electric motor prediction models. By leveraging data - driven insights and advanced analytical techniques, we can make more informed decisions in the electric motor industry, empowering stakeholders with accurate pricing information and valuable insights. Ultimately, the potential for transformative advancements in predictive analytics within the electric motor domain is substantial.

References

- [1] Qun Deng, Yan Peng, Dong Qu, Tao Han, Xisheng Zhan (2022). Neuro - adaptive containment control of

unmanned surface vehicles with disturbance observer and collision – free

- [2] Yuanda Wang, Wenzhang Liu, Jian Liu, Changyin Sun, Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning - based control systems
- [3] <https://arxiv.org/ftp/arxiv/papers/1702/1702.05314.pdf>
- [4] ZhangG, DongX, ShanQ, ZhangW. Event – triggered robust adaptive control for unmanned surface vehicle in presence of deception attacks. Proceedings of the Institution of Mechanical Engineers, PartI: Journal of Systems and Control Engineering,2023; 0 (0).
- [5] MengJooEr, ChuangMa, TianheLiu, Huibin Gong, Intelligent motion control of unmanned surface vehicles: Acritical review, Ocean Engineering
- [6] Wang, Jian, Jing - yangLiu, HongYi, andNai - longWu.2018. "AdaptiveNon - StrictTrajectoryTrack - ing Control Scheme for a Fully Actuated Unmanned Surface Vehicle" Applied Sciences 8, no.4: 598. <https://doi.org/10.3390/app8040598>