



الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة الوطنية العليا للتكنولوجيات المتقدمة
National Higher School of Advanced Technologies



Electrical Engineering and Industrial Informatics Department.

Final Year Project to Obtain the Diploma of Master

Automation
Automation and Industrial Informatics

Review of the state of the art in control laws for marine robots

Realized by
ADDA Mohamed Said
Presented publicly, 11 / 09 / 2025

First LAST NAME	University	Quality
REBAI Karima	ENSTA	President
KHELOUAT Samir	ENSTA	Examinator
OUSAAADI Zahira	ENSTA	Examinator
DJEKOUNE Oualid	CDTA	Supervisor
ZAOUCHE Mohamed	ENSTA	Supervisor

2024/2025

Abstract

This report presents a comprehensive review of the state of the art in **control and navigation strategies for Unmanned Surface Vehicles (USVs)**, with a focus on underactuated platforms operating in uncertain marine environments. It first explores the evolution, classification, and architecture of USVs, emphasizing their growing role in environmental monitoring, inspection, and surveillance missions. The study also examines the **ROS 2 software framework**, highlighting its modularity, scalability, and suitability for distributed robotic systems.

A detailed analysis of the main **sensor technologies** including IMU, DVL, GNSS, magnetometers, and acoustic positioning systems is provided, along with modern **data fusion techniques** such as the Extended and Unscented Kalman Filters (EKF, UKF) and Complementary Filters. The second part of the report reviews **control approaches** used for underactuated marine robots, ranging from classical PID to advanced methods like Model Predictive Control (MPC), Backstepping, Sliding Mode Control (SMC), and Adaptive Control.

Comparative evaluations underline the trade-offs between **simplicity, robustness, and computational complexity**, showing that hybrid architectures combining geometric guidance (LOS, Pure Pursuit, or Vector Fields) with robust or adaptive control provide the best performance for trajectory tracking under disturbances. This literature review establishes a theoretical foundation for future implementation and optimization of control systems for low-cost autonomous surface vehicles.

Keywords: USV, ROS2, Sensor Fusion, Control Strategies

Résumé

Ce rapport présente une revue complète de l'état de l'art des **stratégies de commande et de navigation des véhicules de surface autonomes (USV)**, en mettant l'accent sur les plateformes sous-actionnées évoluant dans des environnements marins incertains. Il explore tout d'abord l'évolution, la classification et l'architecture des USV, en soulignant leur rôle croissant dans les missions de surveillance environnementale, d'inspection et de sécurité maritime. L'étude examine également le **cadre logiciel ROS 2**, en mettant en avant sa modularité, sa scalabilité et son adéquation aux systèmes robotiques distribués.

Une analyse détaillée des principales **technologies de capteurs** notamment l'IMU, le DVL, le GNSS, les magnétomètres et les systèmes de positionnement acoustique est présentée, ainsi que des techniques modernes de **fusion de données** telles que les filtres de Kalman étendu et non linéaire (EKF, UKF) et les filtres complémentaires. La seconde partie du rapport passe en revue les **approches de commande** utilisées pour les robots marins sous-actionnés, allant des régulateurs PID classiques aux méthodes avancées telles que la commande prédictive par modèle (MPC), le backstepping, la commande par mode glissant (SMC) et la commande adaptative.

Les comparaisons effectuées mettent en évidence les compromis entre **simplicité, robustesse et complexité de calcul**, montrant que les architectures hybrides combinant une loi de guidage géométrique (LOS, Pure Pursuit ou champs de vecteurs) avec une commande robuste ou adaptative offrent les meilleures performances de suivi de trajectoire en présence de perturbations. Cette étude bibliographique constitue ainsi une base théorique pour la mise en œuvre et l'optimisation future des systèmes de commande destinés aux véhicules de surface autonomes à faible coût.

Mots-clés : USV, ROS2, Fusion de capteurs, Stratégies de commande

المخلص

يقدم هذا التقرير مراجعة شاملة لأحدث التقنيات في استراتيجيات التحكم والملاحة للركبات السطحية غير المأهولة (USVs)، مع التركيز على المنصات ناقصة التفعيل (underactuated) التي تعمل في بيئات بحرية غير مؤكدة. يستكشف التقرير أولاً تطور وتصنيف وهندسة هذه المركبات، مؤكداً على دورها المتنامي في مهام الرصد البيئي، والتفتيش، والمراقبة. كما تبحث الدراسة في إطار العمل البرمجي ROS 2، مبرزاً طابعه المعياري، وقابليته للتوسع، وملاءمته للأنظمة الروبوتية الموزعة.

يتم تقديم تحليل مفصل لتقنيات الاستشعار الرئيسية بما في ذلك وحدات القياس بالقصور الذاتي (IMU) وسجل سرعة دوبلر (DVL) وأنظمة الملاحة عبر الأقمار الصناعية (GNSS) ومقاييس المغناطيسية، وأنظمة التوضع الصوتي، جنباً إلى جنب مع تقنيات دمج البيانات الحديثة مثل مرشحات كالمان الممتدة وغير الخطية (EKF) والمرشحات التكميلية. يستعرض الجزء الثاني من التقرير مناهج التحكم المستخدمة للروبوتات البحرية ناقصة التفعيل، بدءاً من التحكم التقليدي (PID) وصولاً إلى الطرق المتقدمة مثل التحكم التنبؤي بالنموذج (MPC) وال Backstepping والتحكم بالوضع الانزلاقي (SMC) والتحكم التكيفي.

تؤكد التقييمات المقارنة على الموازنة الضرورية بين البساطة، والمتانة، والتعقيد الحسابي، موضحة أن الهيكليات الهجينة التي تجمع بين التوجيه الهندسي (مثل خط النظر، LOS أو التبع النقي Pure Pursuit أو المجالات المتجهة) مع تحكم متين أو تكيفي توفر أفضل أداء لتتبع المسار في ظل وجود اضطرابات. تؤسس هذه المراجعة المرجعية قاعدة نظرية للتنفيذ المستقبلي وتحسين أنظمة التحكم للركبات السطحية ذاتية القيادة منخفضة التكلفة.

الكلمات المفتاحية: USV، ROS2، دمج الحساسات، استراتيجيات التحكم

Contents

Introduction	1
1 Unmanned Surface Vehicles (USVs)	1
1.1 Background of Unmanned Surface Vehicles (USVs)	1
1.2 Classification of Unmanned Surface Vehicles (USVs)	2
2 ROS 2 Architecture, Modularity Benefits, and PC Deployment	6
2.1 ROS 2 Architecture	6
2.2 ROS 2 Architecture	7
2.3 Deployment on Desktop and Embedded PCs	8
2.4 Implications for USV Development	9
3 Sensors for marine robots	9
3.1 Inertial Measurement Unit (IMU)	9
3.2 Doppler Velocity Log (DVL)	9
3.3 Pressure Sensor	10
3.4 Magnetometer / Electronic Compass	10
3.5 GNSS (RTK-enabled)	11
3.6 Acoustic Positioning (LBL / USBL)	12
3.7 Sensor Fusion Perspectives	12
4 Data Fusion Techniques	12
4.1 Introduction	12
4.2 Standard Kalman Filter (KF)	12
4.3 Extended Kalman Filter (EKF)	14
4.4 Unscented Kalman Filter (UKF)	15
4.5 Complementary Filter	17
4.6 Comparative Summary	17
5 Control Strategies for Underactuated USVs	18
6 Trajectory Tracking Methods	19
7 Control and Navigation in Marine Robots under Uncertainty	20
7.1 Key Control Strategies	21
7.2 Comparative Summary	24
Conclusion	25

List of Figures

Figure 1.	The ATRA framework, applied to aquatic surface robots with the different levels of external system independence (ESI), of environmental complexity (EC) and of mission complexity (MC).	4
Figure 2.	Catamaran USV	5
Figure 3.	Monohull USV	6
Figure 4.	The basic functioning of a DVL	10
Figure 5.	The basic functioning of a digital compass	11
Figure 6.	the basic functioning of a GPS	11
Figure 7.	Diagram of a typical Kalman filter application [1]	14
Figure 8.	Flowchart of Kalman filtering algorithm [1]	14

List of Tables

Table 1.	USV Classification Summary	5
Table 2.	Comparison of Data Fusion Techniques	17
Table 3.	Comparison of Control Strategies for Marine Robots	24

Introduction

The rapid progress of marine robotics in recent years has brought Unmanned Surface Vehicles (USVs) to the forefront of autonomous systems research. Their ability to collect data over long periods, operate in environments that may be difficult or unsafe for humans, and support a wide range of scientific and industrial missions has made them increasingly relevant across multiple disciplines. Despite this growing interest, designing reliable autonomy for USVs remains challenging. Their dynamics are inherently nonlinear and often underactuated, and their operation is strongly affected by waves, wind, and currents, all of which introduce significant uncertainty.

To situate the work carried out in this project, this chapter reviews the main concepts and tools that underpin modern USV development. It begins with an overview of USV evolution and classification, outlining how different mission profiles and hull architectures influence the technical requirements of these platforms. The discussion then turns to the role of *ROS 2*, whose modular communication structure has become a standard for building distributed robotic systems on both desktop and embedded hardware.

A survey of commonly used sensors in marine robotics follows, highlighting the strengths and limitations of IMUs, GNSS receivers, DVLs, magnetometers, and acoustic positioning systems. Because no single sensor is sufficient on its own, the need for reliable state estimation leads naturally to a review of data fusion techniques, including both classical Kalman filters and more advanced nonlinear or optimization-based approaches.

The chapter concludes with an examination of control and trajectory-tracking strategies for underactuated USVs, emphasizing the trade-offs between simplicity, robustness, and computational effort. Together, these elements provide the theoretical background necessary to motivate and support the design and implementation choices presented throughout this thesis.

1 Unmanned Surface Vehicles (USVs)

1.1 Background of Unmanned Surface Vehicles (USVs)

The concept of autonomous watercraft has evolved significantly over the past century, driven by the need to reduce human exposure to hazardous maritime environments and enhance operational efficiency in marine exploration. Early iterations of USVs emerged during World War II, primarily for military applications such as mine clearance and battle damage assessment [2]. These rudimentary radio-controlled vessels laid the groundwork for modern systems, with post-war advancements focusing on refining remote operation capabilities for naval missions like radioactive sample collection and minesweeping [3]. By the 1960s, remotely controlled target drones became integral to missile testing, marking a shift toward more sophisticated control mechanisms [4].

The 1990s saw a diversification of USV applications beyond military use, spurred by academic research and commercial interest. Institutions like MIT pioneered educational and scientific applications, developing autonomous surface crafts (ASCs) for tasks such as bathymetric surveys and harbor monitoring [5]. Concurrently, private companies began exploring USVs for environmental and industrial purposes, leading to platforms like MIMIR and Roboski, which demonstrated potential for hydrographic mapping and marine data acquisition [6, 7].

Technological progress in the 21st century, particularly advancements in GPS, sensor fusion, and autonomous navigation, has expanded USV capabilities across domains. Modern systems now integrate inertial measurement units (IMUs), Doppler velocity logs (DVLs), and acoustic positioning systems to achieve precise localization and environmental perception [8]. This has enabled USVs to perform complex missions, including anti-submarine warfare, port security, and ecological monitoring. For instance, the U.S. Navy’s Blackfish and Protector USVs employ high-resolution cameras and sonar for force protection [9, 10], while commercial platforms like the WAM-V and ASV Global’s C-Worker series focus on hydrographic surveys and environmental sampling [11, 12].

Despite these advancements, challenges persist in achieving full autonomy. Navigation in dynamic marine environments remains hindered by sensor limitations, environmental disturbances (e.g., currents, waves), and the need for robust path-planning algorithms [13]. Additionally, balancing cost-effectiveness with reliability, particularly for low-cost, long-endurance missions, requires innovative data fusion techniques to mitigate sensor inaccuracies [14]. As USVs transition from niche applications to broader commercial and civilian use, addressing these technical barriers will be critical to unlocking their potential in areas such as autonomous shipping and climate research [15].

This evolution reflects a trajectory from specialized military tools to versatile platforms capable of transforming maritime operations, with ongoing research focused on enhancing autonomy, scalability, and adaptability across diverse missions.

1.2 Classification of Unmanned Surface Vehicles (USVs)

USVs are categorized based on size, autonomy level, mission type, and design architecture. These classifications guide platform selection for specific applications, such as dam inspection or offshore monitoring.

By Size and Operational Scope

- **Micro USVs** (<1 m):
 - Designed for shallow or confined waters (e.g., ponds, rivers).
 - Example: *AquaBOT* [16], used for algal bloom detection in lakes.
 - Limitations: Limited payload capacity and endurance.
- **Small-Scale USVs** (1–3 m):
 - Ideal for inland water tasks (dam inspection, reservoir monitoring).
 - Example: *Platypus* USV [17], deployed for water quality sampling.
 - Advantages: Portability, low cost, and maneuverability in cluttered environments.
- **Medium USVs** (3–7 m):
 - Used for coastal surveillance, harbor security, and oceanic research.
 - Example: *SeaHunter* [18], designed for anti-submarine warfare.
 - Features: Extended endurance (72+ hours), modular payload bays.
- **Large USVs** (>7 m):

- Deployed for long-duration offshore missions (oil rig inspections, transoceanic surveys).
- Example: *Saildrone* [19], wind-powered for oceanic climate studies.
- Capabilities: High autonomy, storm resilience, and satellite communication.

By Autonomy Level

- **Remote-Controlled:**
 - Human-operated via radio/joystick; minimal onboard intelligence.
 - Applications: Short-range tasks (e.g., *VideoRay Pro 5* for underwater inspections [20]).
- **Semi-Autonomous:**
 - Hybrid control (human supervises waypoints; USV handles local path planning).
 - Example: *EMILY* USV [21] for search-and-rescue in surf zones.
- **Fully Autonomous:**
 - Self-sufficient decision-making via AI/ROS-based stacks (e.g., SLAM, dynamic replanning).
 - Example: *C-Worker* [22], used for offshore pipeline inspections.

By Mission-Specific Design

- **Environmental Monitoring:**
 - Equipped with multi-parameter sondes (pH, turbidity, dissolved oxygen).
 - Example: *Riverine* USV [23] for pollutant tracking.
- **Security and Surveillance:**
 - Integrated with radar, thermal cameras, and anomaly detection algorithms.
 - Example: *Protector* USV [24] for naval base patrols.
- **Research-Oriented:**
 - Modular designs for experimental control algorithms or sensor suites.
 - Example: MIT's *ROBOAT* [25] for adaptive swarm navigation.

Level	Description	Guidance	Navigation	Control	ESI	EC	MC
9	Fully autonomous	Human-level decision-making, accomplishment of most missions without any intervention from ES (100% ESI).	Human-like navigation capabilities. Situational wareness in extremely complex environments and situations.	Same or better control performance as human controlled vessels in the same situation and conditions.	100%	Extreme Level	Highest Level
8	Full mission planning	High-level decision making. Evaluation and optimisation of mission performance.	Higher level entities and properties are derived from the environment perception according to the desired task to be performed.	Same as previous levels.	High Level	Difficult Level	High Level
7	Dynamic global planning	Same as Level 6 but planning in a dynamic environment.	Same as Level 6 but mapping in a dynamic environment.	Same as previous levels.			
6	Global planning	Goal waypoint provided by ES. Global path planning determines optimal path to goal.	Global map includes properties of the environment. Localisation of aquatic surface robot relative to map.	Same as previous levels.			
5	Local planning with environment awareness	Same as Level 4 but local motion planner also takes into account properties of the environment (flow speed, resistance, etc.).	Same as Level 4 but local map also includes properties of the environment (flow speed, resistance, etc.).	Same as previous levels.	Mid Level	Moderate Level	Mid Level
4	Navigation from A to B	Drive from start point A and stop at point B using sensors to collect and react to the feedback signals.	Local mapping with geometrical representation of immediate surroundings. Localisation of aquatic surface robot relative to map.	Local adaptive control to compensate for possible deviations between map and actual environment.			
3	Reactive control	Motion planner reacts to sensor input feedback and detects if aquatic surface robot is jammed in environment.	Aquatic surface robot can detect contacts between its own body and the environment. No mapping.	Local adaptation to resolve jammed situations and/or local surface adaptation.			
2	Pre-planned motion	Pre-programmed motion patterns.	Same as Level 1.	Automatic control to follow specified motion pattern. No adaptation.	Low Level	Simple Level	Low Level
1	Remote control	All guidance functions are performed by external systems (mainly human operator).	Sensors may be adopted, but all data is processed and analysed by a remote ES. No mapping. No localisation.	All control commands are given by a remote ES (mainly human operator).			
					0%	Lowest Level	Lowest Level

Figure 1: The ATRA framework, applied to aquatic surface robots with the different levels of external system independence (ESI), of environmental complexity (EC) and of mission complexity (MC).

Table 1: USV Classification Summary

Category	Size/Scope	Example Platforms	Primary Missions
Micro USV	<1 m	AquaBOT, MicroCat	Pond/river monitoring
Small-Scale USV	1–3 m	Platypus, Jetyak	Dam inspection, water sampling
Medium USV	3–7 m	SeaHunter, EMILY	Coastal surveillance, search-and-rescue
Large USV	>7 m	Saildrone, C-Worker	Offshore exploration, climate research

Design Architecture

- **Monohull:**
 - Single-hull design (e.g., *ASV Global C-Enduro* [26]), optimized for speed and deep-water stability.
- **Catamaran:**
 - Twin-hull design (e.g., *L3 Harris USV* [27]), provides stability for sensor payloads in rough waters.
- **Hybrid (Surface/Submersible):**
 - Capable of limited underwater operation (e.g., *ECA Group's A18* [28]).



Figure 2: Catamaran USV



Figure 3: Monohull USV

Challenges in Dam Environments

- **Underactuation:** Most USVs control only surge and yaw.
- **Environmental Disturbances:** Turbulent currents and wind gusts.
- **Sensor Limitations:** GPS-denied areas near canyon walls.

2 ROS 2 Architecture, Modularity Benefits, and PC Deployment

Robot Operating System 2 (ROS 2) is a next-generation, open-source middleware framework designed to support complex, distributed robotics applications. In the context of Unmanned Surface Vehicles (USVs), ROS 2 provides the scaffolding for cleanly separating perception, state estimation, planning, and control into reusable components that can be deployed on standard PC hardware. This section presents:

1. An in-depth look at the ROS 2 architecture and communication paradigms.
2. The advantages of its modular design for robotics software development.
3. Practical considerations for deploying ROS 2 on a desktop or embedded PC.

2.1 ROS 2 Architecture

2.1.1 Node-Based Component Model

ROS 2 applications are structured as *nodes*—independent processes encapsulating a single responsibility (e.g., sensor driver, state estimator, or controller). Nodes may be composed into the same process or run as separate executables to balance isolation and performance.

2.1.2 Communication Primitives

ROS 2 uses the Data Distribution Service (DDS) standard as its communication backbone, exposing the following primitives:

- **Topics** (Publish/Subscribe):
 - Asynchronous, many-to-many streams for high-throughput data (e.g., IMU, DVL, GNSS).
 - Quality of Service (QoS) policies (reliability, durability, latency budgets) tunable on a per-topic basis.
- **Services** (Request/Response):
 - Synchronous API suitable for configuration or parameter queries.
 - Supports custom-defined request and response messages.
- **Actions** (Goal/Feedback/Result):
 - Handles long-running tasks with preemption (e.g., “follow this waypoint path”).
 - Provides continuous feedback and allows goal cancellation.

These abstractions map directly onto DDS entities (publishers, subscribers, clients, servers, and uninterested participants), ensuring real-time performance and robust communication across heterogeneous networks [29].

2.1.3 Lifecycle and Composition

Managed Lifecycles: ROS 2 nodes can implement the `rclcpp::LifecycleNode` interface, exposing well-defined state transitions (unconfigured, inactive, active, finalized). This enables:

- Deterministic startup/shutdown sequences (e.g., sensor calibration before data publication).
- Runtime activation/deactivation for power or CPU management.

Component Composition: Multiple nodes can be *composed* into a single process, reducing inter-process communication overhead. Composition is particularly advantageous for high-frequency data paths (e.g., IMU → EKF → controller) on resource-constrained hardware [30].

2.2 Benefits for Modular Robotics Software

2.2.1 Decoupling and Reusability

Each node implements a well-defined interface (message or service type). As long as the interface remains stable, implementations can be swapped or upgraded without affecting the rest of the system. This decoupling fosters:

- *Rapid prototyping:* Replace a simulated sensor node with a hardware driver transparently.
- *Library sharing:* Leverage community packages for state estimation (e.g., `robot_localization`), planning (`nav2`), and control (`ros2_control`).

2.2.2 Scalability and Distribution

Because DDS handles discovery and data routing automatically, ROS 2 systems can scale from single-PC setups to multi-computer or multi-robot fleets. Key features include:

- **Automatic Discovery:** New ROS 2 nodes join the network and begin communication without manual reconfiguration.
- **Multi-robot Coordination:** DDS domains segregate robot networks, enabling multiple USVs to operate concurrently without topic collisions.

2.2.3 Real-Time and Security

ROS 2 introduces:

- *QoS Profiles:* Control over reliability (reliable vs. best-effort), history depth, and deadlines to meet real-time constraints in control loops [31].
- *Security Plugins:* DDS-Security integration provides authentication, encryption, and access control, safeguarding remote USV operations in adversarial or lossy networks [32].

2.3 Deployment on Desktop and Embedded PCs

2.3.1 System Requirements and Installation

- **Supported OS:** Ubuntu LTS (for Jazzy 24.04); Debian-based distributions for binary packages.
- **Installation Methods:**
 - *Binary Packages:* "apt install ros-distro-desktop" for setup.
 - *Colcon Workspace:* Source builds enable development of custom packages.

2.3.2 Hardware Considerations

A modern quad-core PC (Intel i5 or ARM-based like Raspberry Pi 4/5) suffices for typical USV stacks:

- *CPU:* Multi-threaded executors exploit parallelism across nodes.
- *Memory:* 4–8 GB RAM accommodates map and point-cloud data.
- *Network:* Gigabit Ethernet or Wi-Fi for sensor streams and remote monitoring.

2.3.3 Performance Tuning

- **Executor Choice:**
 - *Single-threaded:* Deterministic but may bottleneck.
 - *Multi-threaded:* Parallel callback execution for high throughput.
- **QoS Optimization:**

- Use *best-effort* for lossy, high-rate sensors (e.g., IMU) to reduce latency.
- Use *reliable* for critical data (e.g., thruster commands).
- **Node Composition:** Group high-frequency data paths into components to eliminate serialization and inter-process stalls.

2.4 Implications for USV Development

By leveraging ROS 2 on a PC:

- Developers can integrate diverse sensors (IMU, DVL, GPS, cameras) via existing drivers.
- Navigation stacks (e.g., `nav2`) provide waypoint following and obstacle avoidance out of the box.
- Control frameworks (`ros2_control`) enable real-time thruster management with hardware abstraction.
- Lifecycle and composition features ensure safe startup/shutdown of mission-critical subsystems (e.g., power-down of actuators upon node failure).

Together, these capabilities make ROS 2 on a PC a robust, scalable, and secure foundation for USV autonomy research and deployment.

3 Sensors for marine robots

To support robust navigation and trajectory control in marine robots (USVs and AUVs), the following sensor modalities are essential: IMU, Doppler Velocity Loggers (DVL), pressure sensors, magnetometer/compass, GNSS (for USVs), and acoustic positioning (LBL/USBL). Each offers complementary strengths, recent literature highlights their integration via advanced fusion methods.

3.1 Inertial Measurement Unit (IMU)

MEMS-based IMUs provide high-rate (up to 1 kHz) inertial data—acceleration and angular velocity—for motion estimation. Despite low cost and compact size, they suffer from drift over time. Recent AUV navigation systems couple IMU with DVL and LBL/GNSS in a tightly-coupled Factor Graph Optimization (FGO) scheme, achieving significantly enhanced positioning accuracy and robustness across GNSS-available and denied conditions [33].

3.2 Doppler Velocity Log (DVL)

DVLs measure velocity relative to the seabed or water column with 0.2–1% accuracy at 0.5–10 Hz update rates, enabling precise dead-reckoning. They are indispensable in GNSS-denied environments and, when fused with IMU, can maintain drift under control—Neira *et al.* report position drift below 0.5% of total distance [8]. USVs like MARVEL utilize IMU, DVL, and electromagnetic logs for waypoint tracking under currents [34].

In practical applications, it is essential to estimate and compensate for biases in the Inertial Measurement Unit (IMU) to ensure accurate measurements. While Doppler Velocity Logs (DVLs) are not always available due to their high cost, they significantly enhance position and velocity estimation, particularly during GNSS dropouts. In the absence of a DVL, a water-speed sensor can serve as a more affordable alternative, providing useful data for improving navigation accuracy. As shown in Figure 4, the integration of such sensors, when available, helps mitigate the effects of signal loss and increases the reliability of the system.

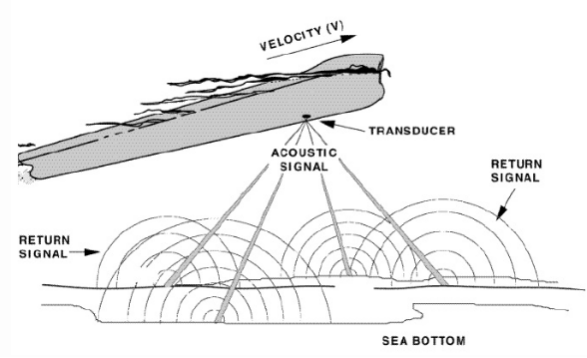


Figure 4: The basic functioning of a DVL

3.3 Pressure Sensor

Pressure transducers provide accurate depth measurements ($\pm 0.1\text{--}1\text{ m}$) and are critical for maintaining vertical position control. They are routinely integrated into INS/DVL/IMU fusion systems, both for trajectory control and as altitude references in AUVs [8].

3.4 Magnetometer / Electronic Compass

Fluxgate or MEMS magnetometers deliver heading estimates ($\pm 0.5\text{--}5^\circ$ accuracy). Fused with IMU within an Attitude and Heading Reference System (AHRS), they stabilize yaw estimate and address gyro drift. These sensors form an integral part of navigation stacks both for surface vessels and underwater platforms [8].

A magnetic compass operates based on the Earth's geomagnetic field. The Earth itself behaves like a giant magnet, generating a magnetic field that extends from its magnetic south pole to its magnetic north pole. This field interacts with magnetized materials, such as the iron or steel needle within a compass. As a result, the needle aligns itself with the Earth's magnetic field lines, typically settling in a north-south orientation. This fundamental principle enables the compass to serve as a reliable directional reference in navigation applications.

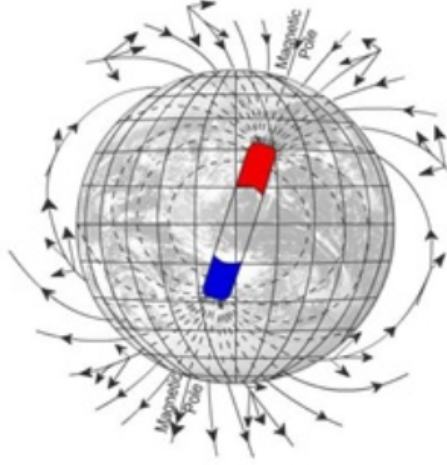


Figure 5: The basic functioning of a digital compass

3.5 GNSS (RTK-enabled)

GNSS provides absolute 2D/3D position on the surface: standard accuracy of 1–3 m, and sub-decimeter with RTK. Essential for USVs, GNSS enhances trajectory following and waypoint performance. Cohen-Salmon Klein recorded sub-decimeter positioning using GNSS, IMU, and DVL fusion [34]. GNSS remains unavailable underwater, prompting reliance on acoustic systems or surfaced updates.

GNSS receivers determine their position by receiving signals from a constellation of satellites orbiting the Earth. The process is as follows:

- Each satellite broadcasts its location and the exact time the signal was transmitted.
- The receiver calculates the distance to each satellite by measuring the time it takes for the signal to reach it.
- By triangulating signals from at least four satellites, the receiver can compute its 3D position (latitude, longitude, altitude) and synchronize its internal clock.

This process allows for accurate positioning, as shown in Figure 6.

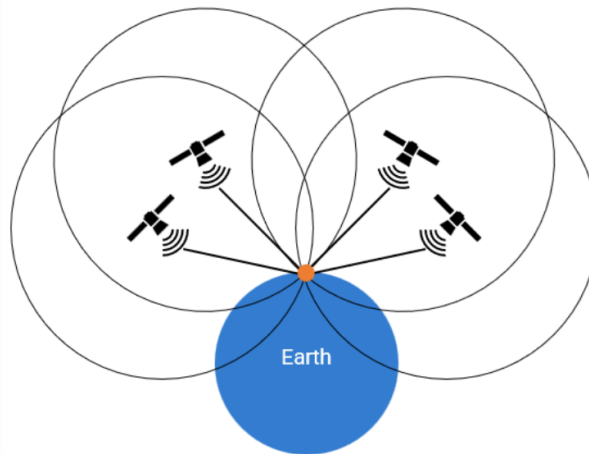


Figure 6: the basic functioning of a GPS

3.6 Acoustic Positioning (LBL / USBL)

LBL and USBL systems supply absolute underwater positioning (0.1–5 m accuracy). LBL uses seabed beacons, while USBL employs ship- or USV-mounted arrays; USBL error grows with range and angle [35]. These systems complement IMU and DVL in areas lacking GNSS, enabling end-to-end trajectory accuracy and were recently integrated into FGO schemes for seamless above/below surface navigation [33].

3.7 Sensor Fusion Perspectives

Achieving accurate navigation and trajectory control demands the integration of these heterogeneous sensors using Bayesian filtering or optimization-based methods:

- *Extended/Unscented Kalman Filter (EKF/UKF)*: Commonly used to fuse IMU, DVL, depth, magnetometer, GNSS, and acoustic range measurements [8].
- *Factor Graph Optimization (FGO)*: Emergent trend, enabling absorbable inclusion of asynchronous sensors like GNSS/DVL/LBL into a unified factor-graph pose estimation (Song *et al.*) [33].
- *Deep fusion*: Neural architectures learning fusion maps across modalities—especially visible in multi-modal underwater leader localization (optical-acoustic-pressure) [36].

4 Data Fusion Techniques

4.1 Introduction

This document details the step-by-step implementation of various sensor fusion filtering techniques for a low-cost Unmanned Surface Vehicle (USV). The USV is equipped with a GY-87 IMU and a NEO-M8N GPS, and requires fusion of high-frequency inertial data with low-frequency GPS measurements. The filtering methods discussed include:

- Standard Kalman Filter (KF)
- Extended Kalman Filter (EKF)
- Unscented Kalman Filter (UKF)
- Complementary Filter

4.2 Standard Kalman Filter (KF)

4.2.1 System Modeling

1. **State Definition:** Define the state vector, e.g.,

$$\mathbf{x} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}. \quad (1)$$

2. **State Transition Model:**

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k, \quad (2)$$

where $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the control matrix, $\mathbf{u}_k$ is the control input, and $\mathbf{w}_k$ is the process noise.

4.2.2 Measurement Model

1. **Measurement Definition:** Define the measurement vector, e.g.,

$$\mathbf{z} = \begin{bmatrix} \text{GPS position} \\ \text{IMU acceleration} \end{bmatrix}. \quad (3)$$

2. **Measurement Equation:**

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k, \quad (4)$$

where H is the measurement matrix and $\mathbf{v}_k$ is the measurement noise.

4.2.3 Initialization

1. Set the initial state estimate $\hat{\mathbf{x}}_0$.
2. Set the initial error covariance P_0 .

4.2.4 Prediction Step

1. Predict the state:

$$\hat{\mathbf{x}}_{k|k-1} = A\hat{\mathbf{x}}_{k-1|k-1} + B\mathbf{u}_{k-1}. \quad (5)$$

2. Predict the error covariance:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q, \quad (6)$$

where Q is the process noise covariance.

4.2.5 Update Step

1. Compute the Kalman Gain:

$$K_k = P_{k|k-1}H^T (HP_{k|k-1}H^T + R)^{-1}, \quad (7)$$

where R is the measurement noise covariance.

2. Update the state estimate:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - H\hat{\mathbf{x}}_{k|k-1}). \quad (8)$$

3. Update the error covariance:

$$P_{k|k} = (I - K_k H)P_{k|k-1}. \quad (9)$$

4.2.6 Iteration

Repeat the prediction and update steps for each new measurement.

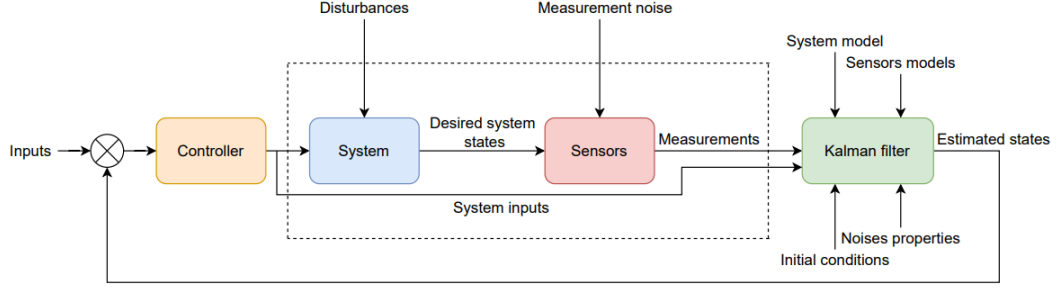


Figure 7: Diagram of a typical Kalman filter application [1]

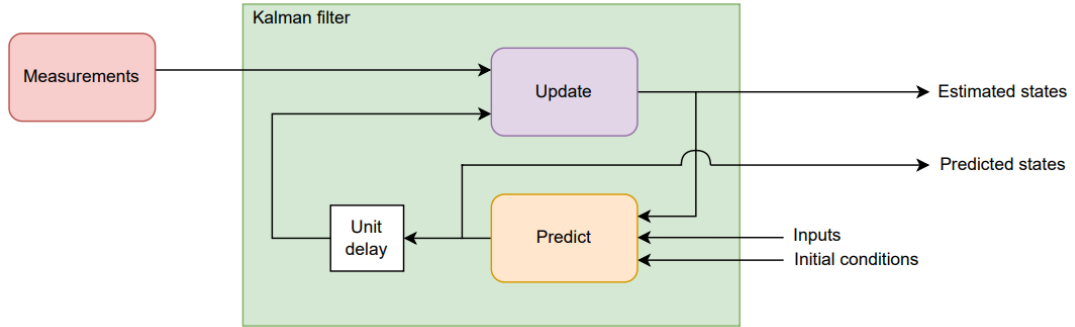


Figure 8: Flowchart of Kalman filtering algorithm [1]

4.3 Extended Kalman Filter (EKF)

4.3.1 Definition

The Extended Kalman Filter (EKF) extends the classical Kalman filter to nonlinear systems by linearizing the state transition and observation models around the current estimate using Jacobian matrices [37, 38]. The prediction step propagates the state mean and covariance through the nonlinear dynamics, while the update step corrects these estimates using incoming measurements and the linearized measurement model [39]. Despite its widespread use, the EKF can suffer from divergence when the process or measurement models are highly nonlinear or when noise characteristics are poorly modeled [40].

4.3.2 Nonlinear System Modeling

1. Define the nonlinear state transition function:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k. \quad (10)$$

2. Define the nonlinear measurement function:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k. \quad (11)$$

4.3.3 Jacobian Computation

1. Compute the Jacobian of the state transition:

$$F_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k}}. \quad (12)$$

2. Compute the Jacobian of the measurement function:

$$H_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}. \quad (13)$$

4.3.4 Initialization

1. Set the initial state estimate $\hat{\mathbf{x}}_0$.
2. Set the initial error covariance P_0 .

4.3.5 Prediction Step

1. Propagate the state:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}). \quad (14)$$

2. Update the error covariance:

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q. \quad (15)$$

4.3.6 Update Step

1. Compute the Kalman Gain:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1}. \quad (16)$$

2. Update the state estimate:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})). \quad (17)$$

3. Update the covariance:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (18)$$

4.3.7 Iteration

Repeat the prediction and update steps for each new measurement.

4.4 Unscented Kalman Filter (UKF)

4.4.1 Definition

The Unscented Kalman Filter (UKF) employs the unscented transform to propagate a set of deterministically chosen sigma points through the true nonlinear dynamics, capturing up to the third moments of the distribution without linearization [41, 42]. After propagation, the posterior mean and covariance are reconstructed from the transformed sigma points, yielding more accurate estimates than the EKF in many nonlinear scenarios [43]. However, the UKF requires generating and processing $2n + 1$ sigma points for an n -dimensional state, increasing computational cost compared to the EKF [38].

4.4.2 Nonlinear System Modeling

1. Define the nonlinear state transition $f(\cdot)$ and measurement function $h(\cdot)$ as in the EKF.

4.4.3 Sigma Points Generation

1. Generate sigma points χ based on the current state estimate $\hat{\mathbf{x}}$ and covariance P .
2. A typical method is:

$$\chi_0 = \hat{\mathbf{x}}, \quad \chi_i = \hat{\mathbf{x}} + \left[\sqrt{(L+\lambda)P} \right]_i, \quad \chi_{i+L} = \hat{\mathbf{x}} - \left[\sqrt{(L+\lambda)P} \right]_i, \quad (19)$$

where L is the state dimension and λ is a scaling parameter.

4.4.4 Prediction Step

1. Propagate each sigma point:

$$\chi_{k|k-1}^{(i)} = f\left(\chi_{k-1}^{(i)}, \mathbf{u}_{k-1}\right). \quad (20)$$

2. Compute the predicted state:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_i W_i^{(m)} \chi_{k|k-1}^{(i)}. \quad (21)$$

3. Compute the predicted covariance:

$$P_{k|k-1} = \sum_i W_i^{(c)} \left(\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right)^T + Q. \quad (22)$$

4.4.5 Update Step

1. Transform sigma points into measurement space:

$$\mathbf{z}_k^{(i)} = h\left(\chi_{k|k-1}^{(i)}\right). \quad (23)$$

2. Compute the predicted measurement:

$$\hat{\mathbf{z}}_k = \sum_i W_i^{(m)} \mathbf{z}_k^{(i)}. \quad (24)$$

3. Compute the measurement covariance:

$$S_k = \sum_i W_i^{(c)} \left(\mathbf{z}_k^{(i)} - \hat{\mathbf{z}}_k \right) \left(\mathbf{z}_k^{(i)} - \hat{\mathbf{z}}_k \right)^T + R. \quad (25)$$

4. Compute the cross-covariance:

$$C_{xz} = \sum_i W_i^{(c)} \left(\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1} \right) \left(\mathbf{z}_k^{(i)} - \hat{\mathbf{z}}_k \right)^T. \quad (26)$$

5. Compute the Kalman Gain:

$$K_k = C_{xz} S_k^{-1}. \quad (27)$$

6. Update the state estimate:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - \hat{\mathbf{z}}_k). \quad (28)$$

7. Update the covariance:

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T. \quad (29)$$

4.4.6 Iteration

Repeat the prediction and update steps as new measurements arrive.

4.5 Complementary Filter

4.5.1 Definition

Complementary filters perform sensor fusion by combining high-pass filtered measurements from fast but drifting sensors (e.g., gyroscopes) with low-pass filtered measurements from slow but stable sensors (e.g., accelerometers, magnetometers) [44]. A typical implementation uses a single blending parameter α to balance the trade-off between noise attenuation and drift correction [45]. While lacking the optimality properties of Kalman filters, complementary filters are computationally trivial and easily implemented on resource-constrained platforms, making them well-suited for real-time attitude estimation on Arduino and Raspberry Pi systems [46].

4.5.2 Sensor Data Acquisition

1. Collect high-frequency gyroscope data.
2. Collect low-frequency accelerometer and magnetometer data.

4.5.3 Filter Design and Implementation

1. Design a high-pass filter for the gyroscope to remove low-frequency drift.
2. Design a low-pass filter for the accelerometer/magnetometer to remove high-frequency noise.

4.5.4 Fusion Algorithm

1. Fuse the sensor data using a weighted sum:

$$\theta = \alpha \theta_{\text{gyro}} + (1 - \alpha) \theta_{\text{acc/mag}}, \quad (30)$$

where α is the weighting factor.

4.5.5 Tuning and Iteration

1. Tune the parameter α to balance responsiveness and noise rejection.
2. Continuously update the filter as new sensor data is available.

4.6 Comparative Summary

Table 2: Comparison of Data Fusion Techniques

Technique	Advantages	Limitations	Applications
Kalman Filter (KF)	Optimal for linear systems	Assumes linear dynamics	Basic state estimation
Extended Kalman Filter (EKF)	Handles nonlinear systems	Linearization errors	USV navigation
Unscented Kalman Filter (UKF)	Better for highly nonlinear systems	Computationally intensive	Advanced state estimation
Complementary Filter	Simple and computationally efficient	Less accurate for complex systems	Attitude estimation

5 Control Strategies for Underactuated USVs

Underactuated USVs (three DOF with only two independent actuators) require control laws that can handle nonlinear dynamics and external disturbances (e.g., wind, waves). Key approaches include:

- **PID Control:**

- Simple proportional–integral–derivative loops on speed and heading.
- Easy to implement on microcontrollers (e.g., Arduino); low computational load.
- Poor disturbance rejection and limited by linear assumptions; requires manual tuning for each operating point [47].

- **Model Predictive Control (MPC):**

- Solves an optimization over a prediction horizon, handling multivariable coupling and actuator constraints.
- Demonstrated to reduce cross-track error under currents when combined with disturbance observers [48].
- High computational demand; typically implemented on SBCs (e.g., Raspberry Pi, onboard PC running ROS 2).

- **Nonlinear Backstepping / Feedback Linearization:**

- Utilizes the full kinematic and dynamic model to systematically cancel nonlinearities and design stabilizing inputs.
- Guarantees asymptotic or exponential convergence of position and heading errors [49].
- Requires accurate model parameters and moderate computation (floating-point operations).

- **Adaptive Control:**

- Online estimation of uncertain parameters (e.g., drag coefficients).
- Enhances performance under model variations; update laws add computational complexity [50].

- **Fuzzy Logic Control:**

- Rule-based, model-free approach handling nonlinearities and uncertainties without explicit models.
- Computationally heavier than PID but feasible on SBCs; requires careful rule-base design and tuning [51].

- **Robust Control (e.g., Sliding Mode, H_∞):**

- Explicitly accounts for matched disturbances and model uncertainties.
- Sliding Mode Control (SMC) offers finite-time convergence at the cost of chattering (addressed by boundary layers).

- H^∞ designs minimize worst-case error norms; both require more math but remain implementable on modern embedded Linux setups.

Implementation Considerations: Simple PID or basic SMC can run in real time on low-cost microcontrollers (e.g., Arduino DUE), while MPC, adaptive, and learning-based controllers typically require the increased computing power of a MiniPC or Raspberry Pi under ROS 2 [52].

6 Trajectory Tracking Methods

Trajectory tracking ensures a USV follows a desired time-parameterized path $(x_d(t), y_d(t), \psi_d(t))$. The underactuated kinematics are

$$\begin{aligned}\dot{x} &= u \cos \psi - v \sin \psi, \\ \dot{y} &= u \sin \psi + v \cos \psi, \\ \dot{\psi} &= r,\end{aligned}\tag{31}$$

where (u, v, r) denote the surge, sway and yaw rates expressed in the body frame. Note that many small USVs are *underactuated*: they have direct control over surge (thrust) and yaw (rudder or differential thrust) but not over sway v .

Define error coordinates in the body frame:

$$\begin{aligned}e_x &= (x - x_d) \cos \psi + (y - y_d) \sin \psi, \\ e_y &= -(x - x_d) \sin \psi + (y - y_d) \cos \psi, \\ e_\psi &= \psi - \psi_d.\end{aligned}\tag{32}$$

The guidance objective is to produce kinematic references (desired heading and/or a desired speed) so that, when the motion controller acts on these references, the closed-loop errors satisfy $e_x, e_y, e_\psi \rightarrow 0$ despite disturbances.

- **Line-of-Sight (LOS) Guidance:** Line-of-Sight guidance steers the vehicle toward a look-ahead point on the path (or waypoint): the desired heading is set to the bearing from the vehicle to that target point. The single major tuning parameter is the look-ahead distance Δ , which trades tracking agility against smoothness.

$$\psi_{\text{LOS}} = \arctan 2(y_L - y, x_L - x),\tag{33}$$

where (x_L, y_L) is a look-ahead point on the trajectory. Variants such as ILOS, ELOS, and ALOS improve steady-state error and robustness [49].

Where used: Common baseline in USV literature and autopilots, ILOS/ELOS variants appear in research to reduce steady-state error under currents.

- **Pure Pursuit (PP)**
 - aim the vehicle’s velocity vector at a (possibly moving) target point; the vehicle “chases” that point. Often used in waypoint-following.
 - **Pros:** Extremely simple; typically produces short, direct trajectories (efficient).

- **Cons:** Ignores cross-track offset causing overshoot/oscillation in presence of drift; can cut corners on curves; less robust than LOS in many marine settings.
 - **Where used:** Mobile robotics and simple USV setups; sometimes combined with LOS for hybrid benefit (energy vs tracking).
- **Guidance Vector Fields:**
 - Define a continuous heading vector at each point guiding the USV to the path’s zero-level set. A common local form for a straight or gently curved path is
$$\chi_d = \chi_p - \frac{\pi}{2} \frac{2}{\pi} \arctan(K y_e), \quad (34)$$
where χ_p is the path tangent and y_e the lateral error; K sets the convergence rate.
 - Offer smooth convergence and provable stability; used for complex path shapes and obstacle avoidance [53].
 - **Where used:** Popular in research (UAV and marine literature); shown in laboratory and some field experiments for USVs and formation control.
 - **Nonlinear Feedback (Backstepping, SMC):**
 - Directly stabilize the error dynamics via Lyapunov-based designs.
 - Sliding-mode controllers switch control to maintain the system on a sliding manifold $s(e, \dot{e}) = 0$ for robust tracking [50].
 - **Neural-Network-Enhanced Guidance:**
 - Deep or adaptive NNs approximate the vessel model or tune guidance parameters online.
 - Demonstrated improved performance post-2020, at the cost of high computational and training demands [54].
 - **Other Methods:**
 - *Artificial Potential Fields* generate attractive/repulsive forces for path following and collision avoidance [55].
 - *Virtual-Target Approaches* have a notional “leader” vehicle that the USV tracks, combining with LOS or vector fields.

Modern hybrid schemes often combine guidance laws (e.g., LOS or vector fields) with advanced controllers (MPC, backstepping, neural) for enhanced disturbance rejection and accuracy [48, 54].

7 Control and Navigation in Marine Robots under Uncertainty

Marine surface and underwater vehicles operate in highly uncertain environments—exposed to currents, winds, waves and turbulence—while also facing propulsion and actuation limits (e.g. thruster saturation or underactuation). These disturbances and constraints make

precise navigation and control challenging. Modern marine control systems must be robust and adaptive: able to reject or compensate disturbances and respect actuator limits, while maintaining stability and performance. A variety of control strategies has been proposed, ranging from classic PID loops to advanced model-based, robust and learning-based controllers. The key approaches are summarized below, followed by a comparative analysis.

7.1 Key Control Strategies

7.1.1 PID Control

Classical Proportional–Integral–Derivative (PID) controllers apply gains on the error, its integral, and its derivative.

- **Advantages:** Simple, well-understood, easy to implement and tune. Widely used for basic functions (e.g. depth or heading hold) in ROVs/AUVs [56].
- **Limitations:** Assumes linear dynamics, performs poorly under strong nonlinearities or severe disturbances; requires frequent retuning; cannot explicitly handle actuator constraints.
- **Applications:** Inner-loop depth/heading control of simple ROVs/AUVs.
- **Suitability:** Low robustness under environmental uncertainty without augmentation.

7.1.2 Model Predictive Control (MPC)

MPC solves a finite-horizon optimization at each timestep using an internal predictive model, minimizing a cost (tracking error, control effort) subject to constraints (actuator saturations, safety). Only the first control action is applied, the optimization repeats in a receding-horizon fashion.

- **Advantages:** Explicitly handles constraints (e.g. thrust limits), optimizes future trajectories; effective for station-keeping and path following [48].
- **Limitations:** High computational load; performance degrades with model inaccuracies.
- **Applications:** Trajectory tracking of USVs/AUVs under actuation and safety constraints.
- **Suitability:** Good disturbance anticipation if disturbances are modeled; robust variants improve performance under uncertainty.

7.1.3 Backstepping Control

Backstepping is a recursive nonlinear control design technique that stabilizes cascaded subsystems by designing virtual control inputs and Lyapunov functions at each step. For USVs it can handle the nonlinear kinematics and partial actuation structure, producing smooth feedback laws with provable stability (given an accurate model).

- **Advantages:** Systematic handling of nonlinear and underactuated dynamics; provable stability under nominal conditions [49].
- **Limitations:** Relies on accurate model parameters; sensitive to unmodeled disturbances unless augmented.
- **Applications:** Underactuated AUV trajectory and depth control.
- **Suitability:** Moderate; often combined with disturbance observers or adaptive elements.

7.1.4 Adaptive Control

Online adjustment of controller parameters to cope with uncertainties.

- **Advantages:** Can compensate for unknown or slowly varying parameters (e.g. drag coefficients) [50].
- **Limitations:** Requires persistent excitation; potential oscillations; careful tuning needed.
- **Applications:** Course, depth and velocity regulation under variable payloads.
- **Suitability:** Good for parametric uncertainty; additional design needed for high-frequency disturbances.

7.1.5 Sliding Mode Control (SMC)

SMC defines a sliding surface (a function of tracking errors) and applies a switching/high-gain control action that drives the system state to this surface and keeps it there. On the surface the closed-loop dynamics follow a reduced-order stable manifold, yielding robustness to matched disturbances.

- **Advantages:** Intrinsically robust to matched disturbances; finite-time convergence [48].
- **Limitations:** Chattering; needs smoothing techniques; unmatched disturbances not directly handled.
- **Applications:** Robust depth and heading control under current disturbances.
- **Suitability:** High robustness if chattering is mitigated.

7.1.6 H_∞ Control

Worst-case disturbance attenuation via H_∞ norm minimization.

- **Advantages:** Formal guarantees on disturbance rejection.
- **Limitations:** Linear model assumptions; conservative designs; complex tuning.
- **Applications:** Theoretical robust control benchmarks.
- **Suitability:** Limited for fully nonlinear marine systems; useful on linearized sub-systems.

7.1.7 Fuzzy Logic Control

Rule-based inference handling nonlinearities without an explicit model.

- **Advantages:** Tolerates imprecision; encodes expert knowledge [51].
- **Limitations:** Rule design is subjective; limited formal stability guarantees.
- **Applications:** Supervisory control, fuzzy-PID hybrids for ROVs/AUVs.
- **Suitability:** Works well when models are poor; often hybridized for constraint handling.

7.1.8 AI-Based Control (NN/RL)

Learning-based controllers using neural networks or reinforcement learning.

- **Advantages:** Can learn complex, unknown dynamics; adaptive disturbance compensation [54].
- **Limitations:** Data-hungry; high compute; black-box nature; limited guarantees.
- **Applications:** Vision-based navigation, data-driven disturbance observers.
- **Suitability:** Promising for highly uncertain scenarios; still mostly experimental.

7.2 Comparative Summary

Table 3: Comparison of Control Strategies for Marine Robots

Strategy	Advantages	Limitations	Applications	Suitability to Uncertainty / Constraints
PID	Simple, easy to tune	Poor disturbance rejection	Depth/heading loops	Low without augmentation
MPC	Handles constraints, predictive	High computational cost, model-sensitive	Path following, station-keeping	Good if accurately modelled
Backstepping	Systematic non-linear design	Model-dependent, sensitive to uncertainties	Underactuated AUV control	Moderate, especially with observers
Adaptive	Compensates unknown parameters	May induce oscillations, complex tuning	Variable-load control	Good for parametric changes
SMC	Robust to matched disturbances	Chattering effect	Robust depth/heading control	High robustness with smoothing
H_∞	Guarantees worst-case performance	Conservative, assumes linear models	Benchmark control designs	Limited for strongly nonlinear systems
Fuzzy Logic	Model-free, handles imprecision	Ad hoc rule design, lacks generality	Fuzzy-PID hybrid systems	Good for systems with poor models
NN/RL	Learns complex, nonlinear dynamics	Data- and computation-intensive	Advanced navigation and decision-making	Promising, but experimental stage

Conclusion

In summary, the literature shows a clear trade-off between simplicity and robustness for underactuated USV path-following: geometric guidance laws such as LOS and Pure Pursuit offer intuitive, easy-to-tune baselines that work well in mild conditions, while non-linear and robust approaches (backstepping, SMC and its higher-order variants) provide stronger disturbance rejection at the cost of greater modeling/tuning effort and the need to mitigate chattering. MPC is attractive when constraints and multi-objective optimization matter, but its real-time compute burden limits off-the-shelf deployment on smaller platforms. Across studies, the most practical and widely validated architecture is a cascaded design—guidance (LOS/VFG/virtual target) feeding an inner motion controller (PD/PI, yaw-rate PID, or a robust SMC)—supported by a tightly tuned EKF/UKF for reliable state estimates. For an engineering project targeting low-cost, underactuated inspection USVs, the recommended path is therefore to adopt LOS+PD/PI as a baseline, add a smoothed or super-twisting SMC for disturbance trials, and evaluate both under the same estimator configuration to quantify tracking, control effort, and robustness trade-offs.

Bibliography

- [1] F. Novák, “State estimation of an unmanned surface vehicle by an unmanned multirotor helicopter,” Master’s thesis, Czech Technical University in Prague, 2022.
- [2] H. B. Shurliff, *Unmanned Surface Vehicles in World War II: Mine Clearance and Battle Damage Assessment*. U.S. Navy Historical Center, 1947.
- [3] V. Bertram, “Post-war developments in remotely operated naval vehicles,” *Naval Engineering Journal*, vol. 120, pp. 45–67, 2008.
- [4] P. Corfield, *The History of Target Drones in Missile Testing*. Aerospace Historical Society, 2006.
- [5] M. J. Manley, “Autonomous surface crafts for educational and scientific applications,” in *Proceedings of the International Conference on Marine Robotics*, 2008, pp. 112–125.
- [6] J. Robert and D. Sutton, “Development of the mimic usv for hydrographic mapping,” *Journal of Oceanic Engineering*, vol. 31, pp. 89–101, 2006.
- [7] T. e. a. Bremer, “Roboski: A platform for marine data acquisition,” *IEEE Journal of Robotics*, vol. 24, pp. 301–310, 2007.
- [8] P. Neira *et al.*, “State-of-the-Art Navigation Systems and Sensors for Unmanned Underwater Vehicles,” *MDPI Journal*, 2021.
- [9] E. Sonnenburg, “Blackfish usv: Maritime force protection and port surveillance,” *Naval Technology Review*, vol. 26, pp. 45–52, 2012.
- [10] R. Hanlon, “Protector usv: Mine countermeasure and reconnaissance,” *Defense Systems Magazine*, vol. 18, pp. 22–28, 2006.
- [11] S. e. a. Pearson, “Wam-v usv: Autonomous waypoint tracking and auv deployment,” *Marine Technology Society Journal*, vol. 48, pp. 78–90, 2014.
- [12] A. G. Ltd, “Asv global c-worker series: Survey and military applications,” <https://www.asvglobal.com>, 2018.
- [13] Y. e. a. Liu, “Challenges in low-cost, long-endurance usv navigation,” *Journal of Field Robotics*, vol. 33, pp. 1023–1041, 2016.
- [14] B. e. a. Carragher, “Wave glider: Hybrid solar-wave propelled usv for environmental monitoring,” *Ocean Engineering*, vol. 63, pp. 112–123, 2013.

- [15] T. Gain, “Ocean usvs: Commercial and military applications in china,” *Marine Technology Review*, vol. 53, pp. 67–75, 2019.
- [16] H. Liu *et al.*, “Micro-usvs for freshwater monitoring: Design and field trials,” *IEEE Transactions on Robotics*, 2021.
- [17] G. Ferri *et al.*, “The platypus usv: A low-cost platform for lake monitoring,” *Journal of Field Robotics*, 2016.
- [18] DARPA, “Sea hunter: An autonomous surface vessel for naval applications,” *Naval Engineering Journal*, 2018.
- [19] S. Inc., “Wind-powered usvs for oceanic climate research,” *Marine Technology Reporter*, 2023.
- [20] V. LLC, “Videoray pro 5: A compact rof/usv hybrid for underwater inspections,” 2022.
- [21] B. Hobson *et al.*, “Emily: A semi-autonomous usv for surf zone rescue,” *Marine Technology Society Journal*, 2017.
- [22] *C-Worker USV Product Specifications*, Teledyne Marine, 2023. [Online]. Available: <https://www.teledynemarine.com>
- [23] S. Khan *et al.*, “Riverine usvs for pollutant tracking in urban waterways,” *Environmental Monitoring and Assessment*, 2020.
- [24] R. A. D. Systems, “Protector usv: Autonomous security for naval bases,” *Defense Technology Monthly*, 2021.
- [25] M. S. C. Lab, “Roboat: A modular usv for urban waterway research,” *Science Robotics*, 2020.
- [26] A. Global, “C-enduro: A monohull usv for offshore applications,” *Marine Technology Society Journal*, 2022.
- [27] L. Technologies, “L3 harris asv: A catamaran platform for sensor payloads,” *Ocean Engineering*, 2021.
- [28] E. Group, “A18: A hybrid usv/rov for subsurface inspections,” *Underwater Technology*, 2020.
- [29] B. Miller and A. Scalzo, “Integration of dds into ros 2: Enabling real-time communication for robotics,” *Journal of Robotics Software*, vol. 37, no. 2, pp. 45–58, 2021.
- [30] J. Thomas and C. Wendt, “Component composition in ros 2 for high-frequency control loops,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020, pp. 1234–1240.
- [31] M. Otte, “Quality of service in ros 2 for real-time robotics applications,” *Robotics and Autonomous Systems*, vol. 145, 2022.

- [32] B. Unal and D. Gonzalez, “Securing robotic systems with dds-security in ros 2,” *International Journal of Secure Software Engineering*, vol. 9, no. 1, pp. 12–27, 2021.
- [33] J. Song, W. Li, R. Liu, and X. Zhu, “FGO-ILNS: Tightly Coupled Multi-Sensor Integrated Navigation System Based on Factor Graph Optimization for Autonomous Underwater Vehicle,” *arXiv*, Oct 2023.
- [34] S. Cohen-Salmon and I. Klein, “Design and Experimental Validation of an Autonomous USV for Sensor Fusion-Based Navigation in GNSS-Denied Environments,” *arXiv*, Mar 2025.
- [35] F. Frontiers *et al.*, “AUV planning and calibration method considering concealment in uncertain environments,” *Frontiers in Marine Science*, 2023.
- [36] M. Yang, Z. Sha, and F. Zhang, “Learning-Based Leader Localization for Underwater Vehicles with Optical-Acoustic-Pressure Sensor Fusion,” *arXiv*, Feb 2025.
- [37] G. Welch and G. Bishop, “An introduction to the kalman filter,” University of North Carolina at Chapel Hill, Tech. Rep., 1995. [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf
- [38] S. J. Julier and J. K. Uhlmann, “The unscented kalman filter: Theory and applications,” *Advances in Neural Information Processing Systems*, vol. 16, pp. 237–244, 2004.
- [39] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Blackwell, 2006.
- [40] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1979.
- [41] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” in *Proceedings of SPIE*, 1997.
- [42] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [43] F. Gustafsson and G. Hendeby, “Some relations between extended and unscented kalman filters,” *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545–555, 2012.
- [44] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [45] InvenSense, “Mpu-6000 and mpu-9150 product specification revision 4.4,” 2015. [Online]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- [46] A. M. Sabatini, “Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.
- [47] X. Yang and Y. Xiang, “Survey of path planning algorithms for unmanned surface vehicles,” *Journal of Marine Robotics*, vol. 10, no. 2, pp. 123–136, 2024.

- [48] J. Fu, L. Chen, and H. Zhao, “Adaptive line-of-sight guidance integrated with mpc for underactuated usv trajectory tracking,” *Ocean Engineering*, vol. 245, pp. 110–123, 2022.
- [49] L. Shang, M. Zhang, and Q. Li, “Improved los guidance methods for usvs under current disturbances,” *International Journal of Control*, vol. 95, no. 3, pp. 457–469, 2022.
- [50] Y. Liu and X. Wang, “Model-free adaptive sliding mode control for underactuated surface vessels,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 98–110, 2023.
- [51] R. Ma, P. Sun, and Z. Gao, “Fuzzy-logic-based robust control for small usvs in turbulent environments,” *Journal of Marine Science and Technology*, vol. 30, no. 1, pp. 45–59, 2025.
- [52] S. Ryu, H. Park, and J. Kim, “idroneboat: A low-cost autonomous usv platform using ros2,” in *Proceedings of MTS/IEEE OCEANS*, 2021, pp. 343–347.
- [53] C. Belta, V. Kumar, and M. Egerstedt, “Synthesis of vector fields for motion planning,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 1064–1079, 2017.
- [54] H. Wang, T. Zhou, and F. Li, “Deep learning-based adaptive guidance for maritime path following,” *Neurocomputing*, vol. 550, pp. 210–222, 2023.
- [55] X. Li and J. Zhang, “Artificial potential field method for usv path planning with obstacle avoidance,” *Ocean Engineering*, vol. 232, pp. 108–120, 2022.
- [56] Y. Xiang and X. Yang, “Analysis of pid control in marine robots,” *Ocean Engineering*, vol. 12, no. 4, pp. 490–502, 2024.