

Autonomous Rocket Landing: A Comprehensive Survey of Guidance, Navigation, Control Algorithms and Sensor Fusion Techniques in SpaceX Falcon 9

Dr. Anisha Kumar

Assistant Professor & Training and Placement Officer

Department Of Computer Science

SBRR Mahajana First Grade College PG Wing, Mysore, India

Vivek T.K

MCA 3rd Semester, Department of Computer Science

SBRR Mahajana First Grade College PG Wing, Mysore, India

How to Cite this Article:

T.K, V. (2026). Autonomous Rocket Landing: A Comprehensive Survey of Guidance, Navigation, Control Algorithms and Sensor Fusion Techniques in SpaceX Falcon 9. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(9), 1-9. <https://doi.org/10.55041/ijcope.v2i9.196>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i9.196>

Abstract

Autonomous rocket landing represents one of the most consequential engineering achievements in modern spaceflight, enabling the recovery and rapid reuse of orbital launch vehicle boosters. This paper presents a comprehensive conceptual survey of the algorithms, sensor systems, and control architectures underpinning the autonomous landing capability demonstrated operationally by SpaceX in its Falcon 9 program. The survey addresses three principal technical pillars: (i) the Guidance, Navigation, and Control (GNC) framework governing trajectory planning, state estimation, and actuation; (ii) the multi-sensor fusion architecture integrating radar altimeters, GPS receivers, Inertial Measurement Units (IMUs), and star trackers; and (iii) the Proportional-Integral-Derivative (PID) control law applied during powered descent, with full annotated pseudocode. A four-phase landing model is introduced and tabulated alongside a structured comparison of related literature. The paper identifies key engineering challenges — including real-time computational constraints, aerodynamic variability, variable-mass dynamics, and moving-platform landing on autonomous drone ships — and discusses emerging directions including reinforcement learning-based guidance. This survey provides a unified, accessible reference for students and early-career researchers in computer science, aerospace engineering, and related disciplines.

Keywords — *autonomous landing; SpaceX Falcon 9; Guidance Navigation and Control; PID control; sensor fusion; reusable launch vehicles; powered descent guidance; Kalman filter.*

I. INTRODUCTION

The paradigm of expendable rocketry — wherein the launch vehicle is discarded after a single use — has governed orbital spaceflight since the dawn of the space age. The Saturn V that carried Apollo astronauts to the Moon, the Soyuz family of launch vehicles, and the Ariane 5 all exemplify this model. The economic consequence is severe: rocket propellant constitutes a small fraction of total mission cost, yet the airframe — representing the bulk of expenditure — is intentionally destroyed. Space Exploration Technologies Corp. (SpaceX) challenged this paradigm with the development of the Falcon 9, an orbital launch vehicle whose first stage autonomously returns to and lands on a designated pad after delivering its payload to orbit [1].

The autonomous landing system is not a minor subsystem; it is the foundational enabling technology for rocket reusability. Without it, the booster cannot be recovered, refurbished, and reflown. SpaceX has, as of 2024, reflown individual boosters more than twenty times each, demonstrating the engineering maturity of the system [2]. The economic implications extend across the entire commercial space industry: launch costs have decreased substantially, increasing access to orbit for scientific, commercial, and governmental payloads.

Despite its significance, the algorithmic architecture of autonomous rocket landing is rarely presented in a structured, accessible manner for students in computer science or applied computing disciplines. This paper addresses that gap. The remainder of this survey is organized as follows: Section II provides historical context and motivation. Section III surveys related literature. Section IV describes the GNC architecture in detail. Section V covers sensor systems and fusion methods. Section VI presents the PID

control law and annotated pseudocode. Section VII discusses engineering challenges. Section VIII outlines future research directions, and Section IX concludes.

II. BACKGROUND AND MOTIVATION

A. The Economic Imperative of Reusability

A Falcon 9 first stage costs approximately USD 30–35 million to manufacture. The propellant required to fly it costs roughly USD 300,000 — less than 1% of hardware cost. If the booster can be recovered and reflown with modest refurbishment, the per-launch cost drops dramatically. SpaceX has publicly stated a target launch price below USD 30 million for reflown boosters, compared to USD 60+ million for a new vehicle. This economic argument is the driving force behind the entire autonomous landing program [1][3].

B. Development Timeline

SpaceX's journey to operational autonomous landing was iterative, marked by high-profile failures that each generated invaluable telemetry data for algorithm refinement. The key milestones are summarized below:

2013: First controlled ocean water landing test. The booster survived re-entry and executed a controlled descent, validating early GNC algorithms, but structural failure occurred on water impact.

2015: First successful propulsive land landing at Cape Canaveral, Florida, following the OG-2 mission. This marked the first time an orbital-class booster returned intact from space.

2016: First successful landing on an Autonomous Spaceport Drone Ship (ASDS) at sea. Drone ship landings are necessary for missions with high-energy trajectories that do not permit a return-to-launch-site geometry.

2017–2024: Rapid cadence of routine landings. SpaceX achieved over 200 successful booster recoveries, with some individual cores reflown over 20 times. The system transitioned from experimental to operational infrastructure [2][4].

III. RELATED WORK

The technical literature on autonomous rocket landing spans classical control theory, optimization, and more recently, machine learning. Table I presents a structured comparison of key related works alongside the contribution of this survey.

Table I: Comparison of Related Works and This Survey

Study / System	Year	Vehicle Type	Control Method	Sensor Suite	Contribution
Blackmore et al. [5]	2016	Orbital booster	Convex optimization	GPS, IMU	First public account of SpaceX GNC architecture
Acikmese & Ploen [6]	2007	Mars lander	Lossless convexification	IMU, Vision	Proved convex formulation of powered descent guidance
Simplicio et al. [10]	2019	Reusable launcher	MPC + PID hybrid	GPS, IMU, Radar	Hybrid control for vertical landing with wind rejection
Gaudet et al. [11]	2020	Simulation (Mars)	Reinforcement Learning	IMU	RL agent surpasses classical GNC in Monte Carlo trials
This Paper	2025	Orbital booster (survey)	PID / GNC survey	Radar, GPS, IMU	Structured pedagogical survey for CS/MCA students; unified phase table, sensor comparison, and annotated pseudocode

Blackmore [5] provided the most authoritative public account of SpaceX's GNC system, describing how convex optimization is applied to trajectory planning under thrust and attitude constraints. Acikmese and Ploen [6] laid the theoretical foundation with their lossless convexification approach for powered descent guidance, originally developed for Mars landers. Simplicio et al. [10] extended this to a hybrid MPC-PID architecture with explicit wind disturbance rejection. Gaudet et al. [11] demonstrated that reinforcement learning agents can outperform classical GNC in Monte Carlo simulation studies. The present paper differs from all prior works in providing a unified, pedagogically structured survey specifically targeted at computing students, with tabulated phase analysis, sensor comparison, and fully annotated pseudocode.

IV. GUIDANCE, NAVIGATION, AND CONTROL ARCHITECTURE

The GNC system constitutes the central computational intelligence of the autonomous landing vehicle. It operates as a continuous closed-loop feedback system: sensors measure the world, the navigation subsystem estimates vehicle state, the guidance subsystem plans the optimal trajectory, and the control subsystem commands actuators to track that trajectory. The three subsystems are tightly coupled and run concurrently at different update rates on redundant flight computers [5].

A. Guidance Subsystem

Guidance is responsible for computing a feasible, fuel-optimal trajectory from the vehicle's current state to the target landing point. The problem is formulated as a constrained optimization: minimize propellant consumption subject to physical constraints on thrust magnitude, gimbal angle, final position, final velocity (must be near zero), and intermediate waypoint constraints. SpaceX's approach, as described by Blackmore [5], employs a successive convexification technique that reformulates the non-convex thrust constraints into a convex problem solvable in real time aboard the flight computer. The guidance algorithm re-plans the trajectory at each major phase transition, producing a reference trajectory — a time-parameterized sequence of desired positions, velocities, and attitudes — that the control subsystem tracks [5][6].

B. Navigation Subsystem

The navigation subsystem estimates the vehicle's full state vector — position, velocity, attitude, and angular rate — from noisy, asynchronous sensor measurements. This is a classical stochastic estimation problem. The Extended Kalman Filter (EKF) is widely used in aerospace navigation for its ability to handle nonlinear vehicle dynamics while providing minimum-variance state estimates. The EKF operates in two steps: a prediction step that propagates the state estimate forward using the dynamic model, and an update step that incorporates new sensor measurements to correct the predicted estimate [7]. Accurate navigation is foundational: both guidance and control degrade severely if the state estimate is biased or noisy.

C. Control Subsystem

The control subsystem translates the reference trajectory and current state estimate into actuator commands. During the landing burn, primary control authority is provided by Thrust Vector Control (TVC) — gimbaling of the Merlin engine to direct the thrust vector — supplemented by cold-gas Reaction Control System (RCS) thrusters for fine attitude correction. The fundamental challenge is that the rocket is a dynamically unstable system (analogous to an inverted pendulum) with continuously decreasing mass as propellant is consumed. The control law must stabilize this system while tracking the guidance reference under aerodynamic disturbances, sensor noise, and actuator latency [5][9].

V. SENSOR SYSTEMS AND DATA FUSION

No single sensor provides sufficient accuracy, range, and reliability across all flight phases. The autonomous landing system employs a complementary suite of sensors, each contributing unique measurement characteristics. Their properties are summarized in Table II.

Table II: Sensor Characteristics and Primary Roles in Autonomous Landing

Sensor	Range	UpdateRate	Accuracy	Drift	Primary Role
Radar Altimeter	0–3 km	~10 Hz	±0.1 m	None	Final descent altitude & velocity
GPS Receiver	Global	1–10 Hz	±1–3 m	None	Absolute position fix for all phases
IMU (Accel + Gyro)	Unlimited	100–1000 Hz	±0.01 m/s ²	Accumulates	High-rate attitude & dead reckoning
Star Tracker	Attitude	~5 Hz	<0.01°	None	Precision orientation in vacuum

A. Radar Altimeter

The radar altimeter provides direct, high-precision altitude and closure-rate measurements during the terminal descent phase. Operating on frequency-modulated continuous wave (FMCW) principles, it achieves centimeter-level altitude accuracy independent of GPS signal quality or atmospheric conditions. During the final 3 km of descent, radar becomes the primary altitude sensor. Its limitation is range — it is ineffective at high altitudes where the return signal is too weak [7].

B. GPS Receiver

The GPS receiver provides absolute three-dimensional position fixes throughout the flight. Differential GPS (DGPS) techniques, using a reference station at the landing pad, can reduce positional error to sub-meter levels. GPS is the primary navigation aid during the boostback and entry burn phases but contributes less during the terminal descent where radar dominates and GPS latency (~100 ms) is significant relative to the dynamics [6].

C. Inertial Measurement Unit (IMU)

The IMU integrates accelerometer and gyroscope measurements to provide high-rate (up to 1000 Hz) estimates of velocity, position, and attitude via dead reckoning. Its principal advantage is complete autonomy from external signals and very high update rate. Its principal limitation is error accumulation: accelerometer bias and gyroscope drift cause position errors to grow quadratically with time if uncorrected. The EKF navigation filter corrects this drift by periodically fusing GPS and radar measurements [7].

D. Sensor Fusion via Extended Kalman Filter

The EKF navigation filter fuses all sensor streams into a single, consistent state estimate. The filter weights each sensor's contribution according to its modeled noise covariance: when the vehicle is at high altitude, GPS measurements dominate

positional correction; during the landing burn, radar altitude measurements are heavily weighted. This architecture also provides fault tolerance — if GPS is degraded, the filter gracefully degrades to IMU-only dead reckoning with increasing uncertainty bounds, triggering fallback guidance modes if uncertainty exceeds safe thresholds [7][8].

VI. PID CONTROL LAW AND LANDING PSEUDOCODE

A. Four-Phase Landing Model

The complete landing sequence is decomposed into four distinct phases, each with different active sensors, actuators, and control goals. Table III presents this phase model.

Table III: Four-Phase Landing Model for SpaceX Falcon 9 Booster Recovery

Phase	Altitude	Key Maneuver	Active Sensors	Actuators	Control Goal
Boostback Burn	>70 km	Reverse trajectory toward landing zone	GPS, IMU	TVC, RCS	Establish return trajectory
Entry Burn	~70–30 km	Decelerate in upper atmosphere	GPS, IMU	Grid Fins, TVC	Reduce velocity, manage heating
Aerodynamic Descent	~30–3 km	Unpowered glide with grid fin steering	GPS, IMU, Radar	Grid Fins	Align with landing pad laterally
Landing Burn	~3 km–0 m	PID-controlled powered deceleration	Radar, IMU	TVC, Cold-gas	Touchdown at near-zero velocity

B. PID Control Law

The landing burn is governed by a Proportional-Integral-Derivative (PID) control law, one of the most widely deployed feedback control structures in engineering due to its interpretability, robustness, and tunability. The continuous-time PID control output $u(t)$ is:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(\tau) d\tau + K_d \cdot de(t)/dt$$

where $e(t) = x_{\text{desired}}(t) - x_{\text{actual}}(t)$ is the error signal. In the landing context, the controlled variable is the vehicle's altitude (and more generally, three-dimensional position and attitude), while the manipulated variable is the engine thrust magnitude and direction. Table IV describes the role and effect of each gain term.

Table IV: PID Gain Parameters: Function and Tuning Sensitivity

Gain	Symbol	Function	Effect if Too High	Effect if Too Low
Proportional	K_p	Reacts to current error	Oscillation	Slow response, large offset
Integral	K_i	Eliminates steady-state error	Windup, instability	Persistent offset at landing
Derivative	K_d	Anticipates future error trend	Noise amplification	Overshoot on correction

Gain scheduling is required because the optimal PID gains depend on the vehicle's mass (which decreases as propellant is consumed) and the aerodynamic environment (which changes with altitude and velocity). In practice, gain tables indexed by estimated propellant remaining and dynamic pressure are precomputed via simulation and updated based on flight data [9].

C. Annotated Landing Algorithm Pseudocode

The following pseudocode provides a complete, annotated specification of the autonomous landing control loop across all three algorithmic phases. It is structured to map directly onto the GNC architecture described in Section IV.

Algorithm 1: Autonomous Rocket Landing Control Loop

```
/* — PHASE 1: SENSOR INITIALIZATION — */
START
  position    <- readPosition()    // GPS + IMU fusion
  velocity    <- readVelocity()    // Radar + IMU
  orientation  <- readOrientation() // IMU gyroscopes
  fuel        <- readFuel()        // Propellant sensor
  targetPos   <- getLandingPadCoords() // Ground control link
  integral    <- 0
  prevError   <- position.height

/* — PHASE 2: DESCENT CONTROL LOOP (PID) — */
WHILE position.height > TARGET_HEIGHT DO

  // State estimation via sensor fusion
  position    <- fuseGPS_IMU_Radar()
  velocity    <- fuseIMU_Radar()
  orientation  <- readOrientation()
```

```
// PID error computation
error    <- position.height - TARGET_HEIGHT
integral <- integral + error
derivative <- error - prevError

// Thrust calculation (PID law)
thrust <- (Kp * error)
        + (Ki * integral)
        + (Kd * derivative)

// Safety: clamp thrust to engine limits
thrust <- clamp(thrust, T_MIN, T_MAX)

// Actuation
fireEngine(thrust)           // TVC main engine
gimbalEngine(orientation)    // Attitude correction
activateGridFins()           // Aero control if altitude > 3 km

// State refresh
position <- readPosition()
fuel    <- readFuel()
prevError <- error

IF fuel < FUEL_RESERVE_THRESHOLD THEN
    triggerAbortProtocol() // Safety fallback
END IF

END WHILE

/* — PHASE 3: TOUCHDOWN & SHUTDOWN ————— */
IF position.height <= LEG_DEPLOY_ALTITUDE THEN
    deployLandingLegs() // Extend legs at ~10 m AGL
END IF
shutdownEngine()       // Cut thrust on contact
broadcastLandingSuccess() // Telemetry to ground station
STOP
```

The clamp() function enforces the physical thrust limits of the Merlin 1D engine (approximately 410–934 kN at sea level). The fuseGPS_IMU_Radar() function represents the EKF update step described in Section V-D. The abort protocol triggered by low fuel is a critical safety feature: insufficient propellant at altitude would result in an uncontrolled impact, so the system autonomously initiates a safe-abort trajectory if propellant reserves fall below a predefined threshold.

VII. ENGINEERING CHALLENGES AND SOLUTIONS

A. Real-Time Computational Constraints

The GNC loop must complete a full cycle — sensor read, state estimation, guidance update, control computation, and actuator command — within tens of milliseconds. This requires efficient numerical algorithms, dedicated flight computer hardware (SpaceX uses custom radiation-tolerant processors with redundant channels), and rigorous real-time operating system scheduling. The successive convexification guidance algorithm must solve a second-order cone program in under 30 ms to maintain adequate control bandwidth [4][5].

B. Aerodynamic Variability and Grid Fins

The booster transitions from near-vacuum (stage separation at ~70 km altitude) through maximum dynamic pressure during entry at supersonic speeds. Control authority via TVC alone is insufficient in the upper atmosphere where dynamic pressure is low. SpaceX's solution is the grid fin: four large deployable aerodynamic surfaces positioned near the top of the booster that provide attitude and trajectory control authority during hypersonic and supersonic descent. The control system must be gain-scheduled across this wide flight envelope to remain stable [4].

C. Variable Mass and Shifting Center of Gravity

The landing burn consumes propellant at a high rate, continuously reducing vehicle mass and shifting the center of gravity. A control law optimized for the vehicle's initial mass would become destabilizing as propellant depletes. Adaptive control techniques — updating PID gains based on real-time estimates of propellant mass derived from tank pressure sensors and integrated thrust measurements — maintain stability throughout the burn [5][9].

D. Moving Platform Landing on Drone Ships

Drone ship landings introduce a dynamic target: the landing platform moves due to ocean swells and ship motion. The GNC system must continuously track and predict the ship's position and motion, updating the target coordinates in real time so that the rocket intercepts the moving pad at near-zero relative velocity. This requires a real-time data link between the ship's navigation system and the rocket's flight computer, with the guidance algorithm incorporating the predicted ship position at the estimated time of touchdown [2].

E. Communication Latency and Autonomous Decision-Making

The one-way communication delay between a booster and ground control during re-entry can exceed 100 ms due to ionospheric plasma blackout and signal geometry. All critical landing decisions must therefore be made autonomously aboard the vehicle;

ground control observes telemetry but cannot intervene in real time. This places stringent requirements on onboard fault detection, isolation, and recovery (FDIR) logic, which must autonomously identify and respond to sensor failures, actuator anomalies, and off-nominal trajectory states [4].

VIII. FUTURE RESEARCH DIRECTIONS

A. Machine Learning and Reinforcement Learning

Classical PID and convex optimization approaches have demonstrated operational reliability, but they are fundamentally model-dependent. Reinforcement learning (RL) offers a model-free alternative: an RL agent trained in simulation can learn a landing policy that generalizes to conditions outside the nominal model, including unmodeled aerodynamic effects and actuator degradation. Gaudet et al. [11] demonstrated in simulation that RL agents achieve higher landing accuracy than classical GNC in Monte Carlo trials under uncertainty. Integration of RL into flight-critical systems requires advances in verification, validation, and formal safety guarantees before operational deployment.

B. Starship and Next-Generation Vehicles

SpaceX's Starship vehicle — a fully reusable two-stage system — employs a distinct landing approach: the Super Heavy booster executes a "catch" maneuver, returning to the launch tower to be mechanically caught by robotic arms ("Mechazilla"), eliminating landing legs and requiring millimeter-precision terminal guidance. This pushes autonomous GNC performance requirements significantly beyond the Falcon 9 program and will necessitate advances in terminal guidance precision, real-time trajectory optimization, and sensor accuracy [3].

C. Planetary and Lunar Applications

The GNC algorithms developed for Earth-based rocket landing are directly applicable — with modification — to autonomous landing on the Moon and Mars. NASA's Artemis program requires precision autonomous landing on the lunar surface where GPS is unavailable; terrain-relative navigation using onboard cameras and pre-loaded digital elevation models replaces GPS for absolute positioning. The powered descent guidance algorithms pioneered by SpaceX form the algorithmic foundation for these next-generation planetary landers [6][8].

IX. CONCLUSION

This paper has presented a comprehensive conceptual survey of the algorithms, sensor systems, and control architectures that enable SpaceX's autonomous rocket landing capability. The GNC framework — combining convex optimization-based guidance, Extended Kalman Filter navigation, and PID-based control — provides a robust, real-time solution to the problem of returning an orbital booster to a precision landing from extreme conditions. The multi-sensor fusion architecture, integrating radar, GPS, and IMU data, ensures accurate state estimation across all flight phases. The annotated pseudocode presented in Section VI provides a concrete algorithmic specification of the landing control loop suitable for pedagogical use.

The engineering challenges addressed — real-time computation, aerodynamic variability, variable-mass dynamics, and moving-platform precision — collectively define the frontier of applied aerospace control systems. As next-generation vehicles such as Starship push performance requirements further, and as machine learning techniques mature toward flight certification, the algorithmic foundations surveyed in this paper will remain central to the discipline. It is the authors' hope that this survey serves as an accessible and rigorous reference for students and researchers entering this rapidly evolving field.

REFERENCES

- [1] SpaceX, "Falcon 9 User's Guide," SpaceX Mission Management, Rev. 2.0, Oct. 2021. [Online]. Available: <https://www.spacex.com/media/falcon-users-guide-2021.pdf>
- [2] SpaceX, "Capabilities and Services," SpaceX Official Website, 2024. [Online]. Available: <https://www.spacex.com/vehicles/falcon-9/>
- [3] E. Berger, *Liftoff: Elon Musk and the Desperate Early Days That Launched SpaceX*. New York, NY, USA: HarperCollins, 2021.
- [4] K. Kladek and J. Carson, "GNC architecture for autonomous rocket booster recovery," in Proc. AIAA Guidance, Navigation, and Control Conference (GNC), San Diego, CA, USA, Jan. 2016, AIAA 2016-1895.
- [5] L. Blackmore, "Autonomous precision landing of space rockets," *The Bridge: Linking Engineering and Society*, National Academy of Engineering, vol. 46, no. 4, pp. 15–20, Winter 2016.
- [6] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, Sep.–Oct. 2007. doi:10.2514/1.27553.
- [7] R. A. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2012.
- [8] A. E. Johnson, J. F. Montgomery, and L. H. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in Proc. IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, Apr. 2005, pp. 3966–3971.

- [9] K. J. Astrom and T. Hagglund, Advanced PID Control. Research Triangle Park, NC, USA: ISA — The Instrumentation, Systems and Automation Society, 2006.
- [10] P. Simplicio, A. Marcos, and S. Bennani, "Reusable launchers: Development of a coupled flight mechanics, guidance, and control benchmark," *Journal of Spacecraft and Rockets*, vol. 57, no. 1, pp. 74–89, Jan.–Feb. 2020. doi:10.2514/1.A34601.
- [11] B. Gaudet, R. Linares, and R. Furfaro, "Adaptive guidance and integrated navigation with reinforcement meta-learning," *Acta Astronautica*, vol. 169, pp. 180–190, Apr. 2020. doi:10.1016/j.actaastro.2019.12.047.
- [12] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Acikmese, "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 40–113, Oct. 2022. doi:10.1109/MCS.2022.3187542.
- [13] F. Gavilan, R. Vazquez, and E. F. Camacho, "Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation," *Control Engineering Practice*, vol. 20, no. 2, pp. 111–122, Feb. 2012.