

Advanced GNC: estimation, autonomous navigation and control

Move from the idea of “knowing where you are” to an architecture that estimates state, carries uncertainty, commands actuators and remains safe when sensors disagree.

Before starting — Prerequisites: modules 00 to 28 are recommended depending on the topic. Every important symbol is defined at first use.

Mastery objectives

- explain quantities, units, assumptions and uncertainty
- repeat simple calculations without a black box
- identify interfaces, limits and degraded modes
- turn the result into an operational or architecture decision

1. State, measurement and model are three different things

A spacecraft never knows its perfect position, velocity and attitude directly. It carries an estimated state. Engineers often gather unknowns in a state vector x : position, velocity, attitude, sensor biases and sometimes environmental parameters. The symbol x therefore denotes a coherent set of variables rather than one sensor reading.

A measurement z comes from a sensor: a star observed by a star tracker, acceleration from an inertial unit, radio range, terrain imagery or lidar altitude. A model predicts how the state should evolve. Estimation compares model and measurements and must also preserve uncertainty. A number without uncertainty can be dangerous because the controller may act as if the information were exact.

2. Navigation: where are we, and how sure are we?

A Mars mission uses different navigation regimes. During cruise, the Deep Space Network helps determine trajectory. Near Mars, images, inertial sensors and altimetry can become more important. During EDL—entry, descent and landing—the available time is too short to wait for ground correction.

The useful question is not only “where am I?” but “what region of states remains plausible?” Error covariance represents that confidence. If lateral uncertainty becomes larger than the safe landing zone, the response must be operational: change trajectory, change target or enter a more conservative mode.

3. Kalman filtering: correct a prediction with an imperfect measurement

The Kalman filter is a family of estimation methods that combines a model prediction with a noisy measurement. Intuitively, if the model is uncertain and the sensor precise, the update follows the sensor more strongly. If the sensor becomes unreliable, the estimator should rely more on prediction. The gain is not a magic constant; it comes from assumed uncertainties.

Real spacecraft use nonlinear models and may rely on extended, unscented or other filters. The engineering loop remains recognizable: predict, measure, compare the innovation, update the state and check whether residuals are consistent with the assumed noise model.

4. Attitude, quaternions and reference frames

Pointing a vehicle requires clear reference frames: inertial, body, local-level, sensor and actuator frames. A frame error can create a numerically correct command along the wrong axis. Quaternions represent attitude without some singularities of Euler angles and make rotation composition convenient.

The operational lesson is more important than memorizing every quaternion identity: every transform must state from which frame to which frame it maps a vector, and the software must normalize and cross-check attitude estimates.

5. Guidance, navigation and control are separate responsibilities

Guidance computes what the vehicle should do to reach an objective. Navigation estimates the real state. Control converts the error between objective and state into actuator commands. A mission succeeds only if those interfaces are sound: brilliant guidance fed by a false state commands the wrong maneuver, and perfect control cannot execute a physically impossible trajectory.

For Mars, degraded modes should preserve this separation. Losing one sensor does not automatically mean losing all guidance. The system can reduce performance, switch sensors, increase margins or stop a critical sequence.

6. Actuators, saturation and control authority

Reaction wheels, thrusters, aerodynamic effectors and descent engines have limits. A wheel can saturate in stored momentum; a thruster has minimum and maximum thrust; an engine does not respond instantaneously. The controller must know the authority that actually exists.

A computed +120% command has no physical meaning. If software asks for more than the actuator can deliver, error can integrate and destabilize the loop. Verification must therefore include saturation, delay, biased sensors and failed actuators.

7. Autonomous surface navigation and rendezvous

On Mars, a rover or crewed vehicle may have to localize without immediate Earth assistance. Cameras, lidar, inertial sensing and maps can be fused to estimate motion. Rendezvous adds relative range, angle and closing-rate measurements.

Autonomy is not the absence of rules. It needs explicit limits: maximum speed when localization confidence falls, safe-return distance, stop conditions and thresholds that request help.

8. Worked example: combine two simple estimates

Assume two independent estimates of one distance: 100.0 m with a standard uncertainty of 4.0 m and 106.0 m with a standard uncertainty of 2.0 m. In inverse-variance weighting, the weights are $1/4^2 = 1/16$ and $1/2^2 = 1/4$. The second measurement therefore receives four times the weight.

The weighted mean is $(100 \times 1/16 + 106 \times 1/4) / (1/16 + 1/4) = 104.8$ m. This is not a full Kalman filter, but it shows the principle: confidence changes the update. If uncertainty assumptions are wrong, the final estimate is wrong too.

Progressive exercise

A rover estimates lateral position at $12 \text{ m} \pm 6 \text{ m}$ from odometry and $20 \text{ m} \pm 3 \text{ m}$ from vision. Compute the inverse-variance weighted mean. Then explain what happens if dust partially obscures the camera but the software fails to increase its uncertainty.

Detailed correction — worked mission case

Reasoned correction

The variances are 36 m² for odometry and 9 m² for vision. The inverse-variance weights are therefore 1/36 and 1/9, so vision receives four times the weight. The fused estimate is $(12/36 + 20/9) / (1/36 + 1/9) = 18.4$ m, with a combined standard deviation of about 2.68 m if the errors are independent and correctly modelled. If dust degrades the camera without increasing its reported uncertainty, the estimator overweights a potentially biased measurement. A robust response is to detect the degradation, inflate its covariance or reject the measurement, then check consistency against independent sensors.

Mini-project

Define the GNC architecture of a pressurized vehicle travelling 40 km from a Mars base: sensors, estimated state, update rates, actuators, degradation thresholds, return procedure and the data that must be recorded to reconstruct an anomaly.

Primary sources and pathways

- NASA — State of the Art: Guidance, Navigation and Control
- NASA JSC — Guidance, Navigation & Control Subsystems
- NASA Science — Basics of Space Flight: Navigation

FR ↔ EN parity matrix

Measurement innovation	Innovation de mesure
Scalar Kalman gain	Gain de Kalman scalaire
State update	Mise à jour d'état
Combined position uncertainty	Incertitude de position combinée
Control-authority margin	Marge d'autorité de commande

Navigation-estimation foundations and mission reasoning

This English navigation module now follows the certified French standard by introducing state, observations and uncertainty before filter calculations, and by linking every estimate to an operational guidance or control decision.

Four concepts to master first

state

Definition. The state is the minimum set of variables needed to predict system evolution in the chosen model, commonly position and velocity plus selected biases.

Mission example. A Mars descent filter may estimate three position components, three velocity components and inertial-sensor biases.

Pitfall. Adding every measurable quantity to the state can make estimation unnecessarily complex or poorly observable.

Mental check. Ask whether each state variable is needed for prediction and can be constrained by dynamics or observations.

Evidence

Use the estimator definition showing each state variable, its reference frame and the dynamic model that propagates those variables between observations.

Decision use

If a proposed state cannot be predicted or constrained by available measurements, simplify the state vector or add sensing before relying on the estimate.

observation

Definition. An observation is a measurement related to the state through a measurement model and affected by noise or bias.

Mission example. Doppler primarily constrains relative velocity along a line of sight rather than all velocity components independently.

Pitfall. Do not treat an indirect measurement as though it directly measured every state component.

Mental check. Write the measurement model and identify which state directions the observation actually constrains.

Evidence

Use the timestamped sensor value, its units and uncertainty, together with the measurement model that predicts that same observable from the state.

Decision use

An observation inconsistent with the model and uncertainty should be gated, investigated or downweighted rather than forced into the navigation solution.

Kalman filter

Definition. A Kalman filter combines a model-based prediction with observations, weighting them according to uncertainty.

Mission example. After inertial propagation, an optical landmark measurement can reduce position uncertainty in directions visible to the camera.

Pitfall. A filter can be confidently wrong if its covariance model understates bias or unmodelled errors.

Mental check. Inspect innovations and consistency, not only the smoothness of the estimated trajectory.

Evidence

Use prediction, innovation, gain and covariance records from a replay or test case so the update can be reproduced step by step.

Decision use

Diverging innovations or unrealistic covariance require tuning, model correction or a fallback estimator before autonomous navigation continues.

covariance

Definition. Covariance describes uncertainty magnitudes and correlations among state components.

Mission example. A position covariance ellipse can be narrow east-west and long north-south even when one scalar accuracy number looks acceptable.

Pitfall. Do not discard off-diagonal correlations when they materially affect prediction or control.

Mental check. Check that covariance remains physically plausible and contracts or expands for understandable reasons as measurements arrive or disappear.

Evidence

Use the covariance matrix or ellipse together with its frame and confidence interpretation, then compare it before and after relevant measurements.

Decision use

If uncertainty grows beyond the navigation requirement, seek better geometry, another sensor or a safer guidance mode.

Calculation laboratory

Read each estimator relation as a sequence: predict the state, predict the measurement, compare it with the observation, and update only when the innovation is credible. Keep signs and units explicit.

Canonical quantitative mini-lessons

This block is the module quantitative reference; exercise values are pedagogical and must be replaced by mission data before any real decision.

Measurement innovation

$$\nu = z - z_{\text{pred}}$$

1 — Concrete question

What does “ $\nu = z - z_{\text{pred}}$ ” compute in “Measurement innovation”?

2 — Intuition without symbols

Navigation compares what sensors observe with what the model expected.

3 — Quantities

ν : measurement innovation [unité de mesure]; z : received measurement [unité de mesure]; z_{pred} : predicted measurement [unité de mesure]

4 — Formula

$$\nu = z - z_{\text{pred}}$$

5 — Read aloud

Read “ $\nu = z - z_{\text{pred}}$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

ν : measurement innovation [unité de mesure]; z : received measurement [unité de mesure]; z_{pred} : predicted measurement [unité de mesure]

7 — Pronunciation

Say subscripts, powers, parentheses and Greek letters explicitly so the relation can be reconstructed without seeing the page.

8 — Units

ν [unité de mesure]; z [unité de mesure]; z_{pred} [unité de mesure]

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Subtraction isolates a margin or difference between comparable states.

11 — Assumptions

Terms must share the same frame and unit.

12 — Independent check

Adding the margin back to the subtracted term should reconstruct the initial state.

13 — Numerical case

With $z = 12.4$ unité de mesure, $z_{\text{pred}} = 10.9$ unité de mesure: $\nu = 12.4 - 10.9 = 1.5$ unit.

14 — Why the calculation works

Subtraction isolates a margin or difference between comparable states.

15 — Verification

Adding the margin back to the subtracted term should reconstruct the initial state.

16 — Mental estimate

Comparing the two terms predicts the sign before calculation.

17 — Interpretation

A persistently large innovation requires reviewing the model, sensor or estimated state.

18 — What the result does not prove

The calculation proves neither that inputs are exact nor that the model remains valid outside the declared scenario; both require separate evidence.

19 — Sensitivity

The result moves directly when either term changes.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $z = 5.2$ unité de mesure, $z_{\text{pred}} = 4.8$ unité de mesure: $\nu = 5.2 - 4.8$?

Detailed guided correction — open after trying

With $z = 5.2$ unité de mesure, $z_{\text{pred}} = 4.8$ unité de mesure: $\nu = 5.2 - 4.8 = 0.4$ unit. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $z = -2.5$ unité de mesure, $z_{\text{pred}} = -1.7$ unité de mesure: $\nu = -2.5 - -1.7$?

Autonomous correction — open after trying

With $z = -2.5$ unité de mesure, $z_{\text{pred}} = -1.7$ unité de mesure: $\nu = -2.5 - -1.7 = -0.8$ unit. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

A persistently large innovation requires reviewing the model, sensor or estimated state.

Scalar Kalman gain

$$K = P_{\text{pred}} / (P_{\text{pred}} + R_{\text{meas}})$$

1 — Concrete question

What does “ $K = P_{\text{pred}} / (P_{\text{pred}} + R_{\text{meas}})$ ” compute in “Scalar Kalman gain”?

2 — Intuition without symbols

The gain weights the relative confidence placed in prediction and measurement.

3 — Quantities

K: correction gain [sans dimension]; P_{pred} : predicted variance [unité²]; R_{meas} : measurement variance [unité²]

4 — Formula

$$K = P_{\text{pred}} / (P_{\text{pred}} + R_{\text{meas}})$$

5 — Read aloud

Read “ $K = P_{\text{pred}} / (P_{\text{pred}} + R_{\text{meas}})$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

K: correction gain [sans dimension]; P_{pred} : predicted variance [unité²]; R_{meas} : measurement variance [unité²]

7 — Pronunciation

Say subscripts, powers, parentheses and Greek letters explicitly so the relation can be reconstructed without seeing the page.

8 — Units

K [sans dimension]; P_{pred} [unité²]; R_{meas} [unité²]

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Division relates a quantity to a coherent capacity, duration or reference.

11 — Assumptions

Numerator and denominator must describe the same scenario and the denominator must remain positive.

12 — Independent check

Multiplying the result by the denominator should reconstruct the numerator.

13 — Numerical case

With $P_{\text{pred}} = 4$ unité², $R_{\text{meas}} = 1$ unité²: $K = 4 / (4 + 1) = 0.8$.

14 — Why the calculation works

Division relates a quantity to a coherent capacity, duration or reference.

15 — Verification

Multiplying the result by the denominator should reconstruct the numerator.

16 — Mental estimate

Comparing numerator and denominator scales predicts the result.

17 — Interpretation

A gain near zero favors the model; a high gain gives more weight to the measurement.

18 — What the result does not prove

The calculation proves neither that inputs are exact nor that the model remains valid outside the declared scenario; both require separate evidence.

19 — Sensitivity

The result rises with the numerator and falls as the denominator rises.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $P_{\text{pred}} = 9 \text{ unité}^2$, $R_{\text{meas}} = 3 \text{ unité}^2$: $K = 9 / (9 + 3)$?

Detailed guided correction — open after trying

With $P_{\text{pred}} = 9 \text{ unité}^2$, $R_{\text{meas}} = 3 \text{ unité}^2$: $K = 9 / (9 + 3) = 0.75$. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $P_{\text{pred}} = 2 \text{ unité}^2$, $R_{\text{meas}} = 6 \text{ unité}^2$: $K = 2 / (2 + 6)$?

Autonomous correction — open after trying

With $P_{\text{pred}} = 2 \text{ unité}^2$, $R_{\text{meas}} = 6 \text{ unité}^2$: $K = 2 / (2 + 6) = 0.25$. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

A gain near zero favors the model; a high gain gives more weight to the measurement.

State update

$$\hat{x} = x_{\text{pred}} + K \times \nu$$

1 — Concrete question

What does " $\hat{x} = x_{\text{pred}} + K \times \nu$ " compute in "State update"?

2 — Intuition without symbols

The correction moves the predicted state by a controlled fraction of the observed discrepancy.

3 — Quantities

$\hat{x}$: corrected state [unité d'état]; x_{pred} : predicted state [unité d'état]; K : correction gain [sans dimension];
 ν : innovation [unité d'état]

4 — Formula

$$\hat{x} = x_{\text{pred}} + K \times \nu$$

5 — Read aloud

Read " $\hat{x} = x_{\text{pred}} + K \times \nu$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

$\hat{x}$: corrected state [unité d'état]; x_{pred} : predicted state [unité d'état]; K : correction gain [sans dimension];
 ν : innovation [unité d'état]

7 — Pronunciation

Say subscripts, powers, parentheses and Greek letters explicitly so the relation can be reconstructed without seeing the page.

8 — Units

$\hat{x}$ [unité d'état]; x_{pred} [unité d'état]; K [sans dimension]; ν [unité d'état]

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

The relation turns the system mechanism into a quantity that can be checked by numerical substitution.

11 — Assumptions

All inputs must be coherent in unit, frame, duration and scenario.

12 — Independent check

A second method or inverse relation should recover the same order of magnitude.

13 — Numerical case

With $x_{\text{pred}} = 100$ unité d'état, $K = 0.8$ sans dimension, $nu = 1.5$ unité d'état: $\hat{x} = 100 + 0.8 \times 1.5 = 101.2$ unit.

14 — Why the calculation works

The relation turns the system mechanism into a quantity that can be checked by numerical substitution.

15 — Verification

A second method or inverse relation should recover the same order of magnitude.

16 — Mental estimate

Estimating before calculation helps detect sign, unit or factor errors.

17 — Interpretation

Accept the update only when gain and innovation remain compatible with sensor and model limits.

18 — What the result does not prove

The calculation proves neither that inputs are exact nor that the model remains valid outside the declared scenario; both require separate evidence.

19 — Sensitivity

Varying one input at a time reveals which variable drives the decision most strongly.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $x_{\text{pred}} = 20$ unité d'état, $K = 0.5$ sans dimension, $nu = -2$ unité d'état: $\hat{x} = 20 + 0.5 \times -2$?

Detailed guided correction — open after trying

With $x_{\text{pred}} = 20$ unité d'état, $K = 0.5$ sans dimension, $nu = -2$ unité d'état: $\hat{x} = 20 + 0.5 \times -2 = 19$ unit.
The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $x_{\text{pred}} = -3$ unité d'état, $K = 0.25$ sans dimension, $nu = 4$ unité d'état: $\hat{x} = -3 + 0.25 \times 4$?

Autonomous correction — open after trying

With $x_{\text{pred}} = -3$ unité d'état, $K = 0.25$ sans dimension, $nu = 4$ unité d'état: $\hat{x} = -3 + 0.25 \times 4 = -2$ unit.
The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Accept the update only when gain and innovation remain compatible with sensor and model limits.

Combined position uncertainty

$$\sigma_{\text{pos}} = \sqrt{\sigma_x^2 + \sigma_y^2}$$

1 — Concrete question

What does " $\sigma_{\text{pos}} = \sqrt{\sigma_x^2 + \sigma_y^2}$ " compute in "Combined position uncertainty"?

2 — Intuition without symbols

Independent uncertainties on two axes combine through their squares rather than a simple sum.

3 — Quantities

σ_{pos} : combined planar uncertainty [m]; σ_x : x uncertainty [m]; σ_y : y uncertainty [m]

4 — Formula

$$\sigma_{\text{pos}} = \sqrt{\sigma_x^2 + \sigma_y^2}$$

5 — Read aloud

Read " $\sigma_{\text{pos}} = \sqrt{\sigma_x^2 + \sigma_y^2}$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

σ_{pos} : combined planar uncertainty [m]; σ_x : x uncertainty [m]; σ_y : y uncertainty [m]

7 — Pronunciation

Say subscripts, powers, parentheses and Greek letters explicitly so the relation can be reconstructed without seeing the page.

8 — Units

σ_{pos} [m]; σ_x [m]; σ_y [m]

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Root-sum-square combines independent contributions without assuming they all add in the same direction.

11 — Assumptions

Contributions must be sufficiently independent for this combination to be justified.

12 — Independent check

The square of the result should equal the sum of squared contributions.

13 — Numerical case

With $\sigma_x = 3$ m, $\sigma_y = 4$ m: $\sigma_{\text{pos}} = \sqrt{3^2 + 4^2} = 5$ m.

14 — Why the calculation works

Root-sum-square combines independent contributions without assuming they all add in the same direction.

15 — Verification

The square of the result should equal the sum of squared contributions.

16 — Mental estimate

The result should remain between the dominant contribution and their arithmetic sum.

17 — Interpretation

Increase navigation margins when combined uncertainty approaches the safe corridor width.

18 — What the result does not prove

The calculation proves neither that inputs are exact nor that the model remains valid outside the declared scenario; both require separate evidence.

19 — Sensitivity

The largest contribution weighs more strongly because it is squared.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $\sigma_x = 1.2$ m, $\sigma_y = 1.6$ m: $\sigma_{\text{pos}} = \sqrt{1.2^2 + 1.6^2}$?

Detailed guided correction — open after trying

With $\sigma_x = 1.2$ m, $\sigma_y = 1.6$ m: $\sigma_{\text{pos}} = \sqrt{1.2^2 + 1.6^2} = 2$ m. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $\sigma_x = 0.5$ m, $\sigma_y = 0.5$ m: $\sigma_{\text{pos}} = \sqrt{0.5^2 + 0.5^2}$?

Autonomous correction — open after trying

With $\sigma_x = 0.5$ m, $\sigma_y = 0.5$ m: $\sigma_{\text{pos}} = \sqrt{0.5^2 + 0.5^2} = 0.7071$ m. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Increase navigation margins when combined uncertainty approaches the safe corridor width.

Control-authority margin

$$u_{\text{margin}} = u_{\text{max}} - u_{\text{req_abs}}$$

1 — Concrete question

What does “ $u_{\text{margin}} = u_{\text{max}} - u_{\text{req_abs}}$ ” compute in “Control-authority margin”?

2 — Intuition without symbols

The margin shows how much actuator authority remains before saturation.

3 — Quantities

u_{margin} : control margin [unité de commande]; u_{max} : maximum available command [unité de commande];
 $u_{\text{req_abs}}$: absolute requested command [unité de commande]

4 — Formula

$u_{\text{margin}} = u_{\text{max}} - u_{\text{req_abs}}$

5 — Read aloud

Read " $u_{\text{margin}} = u_{\text{max}} - u_{\text{req_abs}}$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

u_{margin} : control margin [unité de commande]; u_{max} : maximum available command [unité de commande];
 $u_{\text{req_abs}}$: absolute requested command [unité de commande]

7 — Pronunciation

Say subscripts, powers, parentheses and Greek letters explicitly so the relation can be reconstructed without seeing the page.

8 — Units

u_{margin} [unité de commande]; u_{max} [unité de commande]; $u_{\text{req_abs}}$ [unité de commande]

9 — Convention

Keep the same frame, time basis, system boundary and sign convention throughout the substitution.

10 — Why this operation

Subtraction isolates a margin or difference between comparable states.

11 — Assumptions

Terms must share the same frame and unit.

12 — Independent check

Adding the margin back to the subtracted term should reconstruct the initial state.

13 — Numerical case

With $u_{\text{max}} = 1$ unité de commande, $u_{\text{req_abs}} = 0.7$ unité de commande: $u_{\text{margin}} = 1 - 0.7 = 0.3$ unit.

14 — Why the calculation works

Subtraction isolates a margin or difference between comparable states.

15 — Verification

Adding the margin back to the subtracted term should reconstruct the initial state.

16 — Mental estimate

Comparing the two terms predicts the sign before calculation.

17 — Interpretation

A guidance plan that consumes nearly all authority should be replanned before a critical phase.

18 — What the result does not prove

The calculation proves neither that inputs are exact nor that the model remains valid outside the declared scenario; both require separate evidence.

19 — Sensitivity

The result moves directly when either term changes.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $u_{\text{max}} = 0.8$ unité de commande, $u_{\text{req_abs}} = 0.65$ unité de commande: $u_{\text{margin}} = 0.8 - 0.65$?

Detailed guided correction — open after trying

With $u_{\text{max}} = 0.8$ unité de commande, $u_{\text{req_abs}} = 0.65$ unité de commande: $u_{\text{margin}} = 0.8 - 0.65 = 0.15$ unit. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $u_{\text{max}} = 10$ unité de commande, $u_{\text{req_abs}} = 9.5$ unité de commande: $u_{\text{margin}} = 10 - 9.5$?

Autonomous correction — open after trying

With $u_{\text{max}} = 10$ unité de commande, $u_{\text{req_abs}} = 9.5$ unité de commande: $u_{\text{margin}} = 10 - 9.5 = 0.5$ unit. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

A guidance plan that consumes nearly all authority should be replanned before a critical phase.

Mission reasoning

Prediction and correction

Navigation alternates between propagation and observation updates. During propagation, the dynamics model advances position, velocity and other states while uncertainty usually grows. When a valid observation arrives, the estimator compares it with the predicted observation and corrects the state. The quality of the update depends on geometry, sensor noise, bias modelling and whether the measurement is actually independent of previous information.

Observability and geometry

A sensor can be precise yet provide weak information about one direction. Range measures distance, Doppler constrains radial velocity, and a camera constrains angles. Combining different geometries can make the full state observable. Mars approach navigation therefore plans observations around line-of-sight geometry and available landmarks rather than simply maximizing the number of measurements.

Autonomous fault handling

Communication delay means GNC cannot wait for Earth when residuals become inconsistent during descent or surface driving. Software needs thresholds for rejecting outliers, inflating covariance, switching sensors or entering a safe mode. Those decisions must themselves be testable. A robust estimator reports health and uncertainty so guidance can choose a trajectory that remains safe when navigation quality degrades.

Control uses estimated state, not truth

A controller acts on the estimated state. If navigation has latency or bias, even a mathematically stable controller can command the wrong action. Integrated verification therefore injects navigation errors and sensor failures into closed-loop simulations and hardware tests. The goal is not merely a low average tracking error but bounded behaviour throughout the declared uncertainty envelope.

Navigation practice — reason from the measurements before opening the solution

Exercise A — State selection

Choose a minimal navigation state for a rover that must drive between mapped waypoints while monitoring wheel slip. Explain what you would not include.

Detailed correction — Exercise A

A practical state could include planar position, heading, speed and a slip-related bias or scale term if the estimator can constrain it. Raw camera pixels or every motor current should remain observations or diagnostic variables unless the dynamics truly require them as states.

Choose only variables needed to propagate rover motion and estimate slip; raw camera pixels and unrelated diagnostic channels should remain observations unless the dynamic model truly needs them as states.

Exercise B — Observation geometry

A Doppler measurement is available from one fixed radio beacon. What part of rover motion does it constrain most directly?

Detailed correction — Exercise B

It constrains relative velocity along the line of sight between rover and beacon. Motion perpendicular to that line is weakly or not directly constrained by that single Doppler observation, so additional geometry or sensor types are needed.

Remember that one Doppler line constrains motion mainly along the beacon line of sight; it cannot by itself determine the rover's complete planar velocity vector.

Exercise C — Innovation calculation

A predicted altitude measurement is 2,480 m and the sensor reports 2,455 m. Calculate the innovation using measurement – prediction and interpret the sign.

Detailed correction — Exercise C

Innovation = 2,455 – 2,480 = –25 m. The negative sign means the observation is lower than predicted. Whether –25 m is concerning depends on expected measurement and prediction uncertainty.

Compute the innovation with the declared sign convention: measured 2455 m minus predicted 2480 m equals -25 m, and preserve metres as the unit of the residual.

Exercise D — Covariance reasoning

A position covariance ellipse shrinks east-west after a landmark observation but barely changes north-south. Give a plausible explanation.

Detailed correction — Exercise D

The landmark geometry likely provided much stronger sensitivity to the east-west component than to north-south position. The asymmetric update is therefore plausible and should be consistent with the measurement Jacobian and viewing geometry.

Interpret the covariance change geometrically: a landmark observation that contains strong east-west information can shrink that axis while leaving north-south uncertainty comparatively large.

Exercise E — Filter health

Innovations remain near zero, but a camera has developed an unmodelled slowly varying bias. Why can the filter still become unsafe?

Detailed correction — Exercise E

The estimator may absorb the bias into state variables and report covariance that is too small. Innovation magnitude alone is insufficient; consistency tests, cross-sensor comparison, bias states or fault detection are needed.

Near-zero innovations are not sufficient evidence of health if an unmodelled camera bias is being absorbed elsewhere in the state; inspect bias trends, consistency tests and independent sensors.

Interactive beginner glossary

Use the interactive vocabulary to distinguish what the rover actually measures from what the estimator infers. Each definition should clarify how information enters the navigation solution.

- **state** — A state is the minimum set of variables needed to predict a system's future evolution from the current instant when the governing model and inputs are known.
- **state vector** — A state vector is an ordered collection of state variables, such as position, velocity, attitude and sensor bias, represented together for estimation and propagation.
- **position** — Position specifies where an object is relative to a declared coordinate frame. Position numbers are meaningless unless the frame, origin, axes and units are known.
- **velocity** — Velocity is the rate of change of position and includes direction as well as speed. Navigation usually estimates velocity in a declared reference frame.
- **bias** — Bias is a systematic offset in a sensor or estimate. If it changes slowly enough, an estimator can include it as a state and infer it from observations.
- **observation** — An observation is a measured quantity linked to the underlying state through a measurement model, for example Doppler shift, image angle or range.
- **measurement model** — A measurement model predicts what a sensor should observe for a given state. Comparing prediction with the actual measurement creates the innovation used by an estimator.
- **innovation** — Innovation is measured value minus predicted measurement for the same quantity. It is the new information available to correct the predicted state.

- residual — A residual is a difference between observed and model-predicted quantities. Depending on the algorithm it can be evaluated before or after an estimation update.
- prediction — Prediction propagates the current estimated state forward using a dynamic model and known inputs before new measurements are incorporated.
- estimation — Estimation combines a model with imperfect observations to infer quantities that cannot be known exactly, while also representing the uncertainty of the estimate.
- Kalman filter — A Kalman filter combines a dynamic prediction with measurements by weighting them according to their uncertainties, producing an updated state estimate and covariance.
- covariance — Covariance represents uncertainty and correlation among state components. In a navigation filter it shows not only how uncertain each variable is but how their errors are coupled.
- variance — Variance is the squared spread assigned to one uncertain quantity. In an estimator it commonly appears on the diagonal of the covariance matrix.
- correlation — Correlation expresses how estimation errors in two variables tend to vary together. Strong correlation can make some state components difficult to distinguish.
- observability — Observability describes whether the available measurements over time contain enough information to distinguish and estimate the required state variables.
- measurement noise — Measurement noise represents uncertainty associated with the sensor observation, including repeatability and other random measurement effects included in the estimator model.
- process noise — Process noise represents unmodelled or unpredictable changes in the system dynamics, such as small terrain disturbances, actuator uncertainty or changing wheel slip.
- outlier — An outlier is a measurement inconsistent with the expected prediction and uncertainty. It may indicate a bad observation, an unexpected event or a faulty model.
- gating — Gating is a rule that accepts, downweights or rejects a measurement when its innovation is too inconsistent with the predicted uncertainty.
- sensor fusion — Sensor fusion combines complementary measurements so one sensor can constrain weaknesses of another, while the estimator tracks their uncertainties and correlations.
- Doppler — Doppler measurement uses the frequency shift caused by relative motion along the line of sight. It primarily constrains line-of-sight velocity rather than full three-dimensional motion.
- range — Range is the measured distance between two points or vehicles. A single range constrains the state to a geometric surface and usually needs other measurements to determine position uniquely.
- line of sight — Line of sight is the direction from an observer or sensor to a target. Many radio and optical measurements constrain only components projected along this direction.
- landmark navigation — Landmark navigation estimates position and attitude by identifying known terrain or visual features and comparing their observed image geometry with a map or model.
- inertial navigation — Inertial navigation propagates motion from accelerometers and gyroscopes without external references. Small sensor errors integrate over time, so external observations are needed to bound drift.
- autonomy — Autonomy is the ability to make and execute local decisions without waiting for immediate Earth commands, while staying inside declared authority, safety rules and validated operating limits.
- safe mode — Safe mode is a predefined survival configuration that stops nonessential activity, protects critical resources and establishes a stable state from which diagnosis or recovery can begin.
- guidance — Guidance determines the desired trajectory or maneuver needed to reach an objective while respecting constraints and current state estimates.
- control — Control converts guidance objectives and state feedback into actuator commands that drive the system toward the desired behavior while maintaining stability and limits.

Operational depth: from calculation to mission decision

Observability before algorithm choice

An estimator cannot reconstruct information that the sensors and dynamics do not make observable. Before choosing an extended Kalman filter, particle filter or optimisation method, engineers ask whether the planned measurements actually

constrain the desired state. One range measurement from one beacon provides a geometric surface of possibilities. Motion, additional beacons or angular measurements can break that ambiguity. Observability analysis prevents teams from blaming software for information that the mission architecture never supplied in the first place.

Innovation as a health signal

Innovation is more than a correction term. Its sign, magnitude and statistical consistency reveal whether predictions and observations remain compatible. Repeated residuals in one direction can expose bias, an incorrect environment model or a drifting sensor. A sudden large innovation may indicate an outlier or a real manoeuvre that the model missed. Automated health logic should therefore examine sequences and expected covariance, not reject every large residual blindly. The aim is to distinguish surprising truth from bad data.

Covariance must tell the truth

A navigation filter is dangerous when its estimate looks smooth while covariance becomes unrealistically small. Covariance should expand during periods of weak observation and contract when informative measurements arrive. If a sensor is degraded but its reported noise remains optimistic, the filter can become overconfident. Consistency tests compare observed innovations with predicted uncertainty. When they disagree persistently, operations can inflate covariance, estimate additional bias states, reject a source or switch navigation modes.

Guidance with uncertain state

Guidance should not assume the estimated state is exact. A descent corridor can be safe for the nominal trajectory but unsafe when position and velocity uncertainty are considered. Robust guidance keeps sufficient margin so that credible state errors remain inside terrain, thermal and propellant constraints. This is why navigation covariance belongs in trajectory decisions. A safe command is one that remains acceptable across the uncertainty envelope, not merely one that is optimal at the estimated centre point.

Integrated test philosophy

GNC verification must close the loop. Sensor simulators, flight software, actuator dynamics and vehicle models are tested together with injected faults and timing delays. Teams deliberately create lost measurements, biased sensors, delayed packets and processor resets to observe whether the system detects the problem and degrades safely. Hardware-in-the-loop tests cannot reproduce every Mars condition, so evidence combines analysis, simulation, laboratory tests and flight-like demonstrations. The important question is how the chain behaves when one assumption is wrong.

Final mission assurance note

Autonomous navigation should also preserve provenance. When a state estimate changes sharply, engineers need to know which observation caused the update, what covariance was assumed and whether gating logic accepted or down-weighted the measurement. Logging only the final position loses the evidence needed for diagnosis. On Mars, that evidence may be reviewed after long communication delay, so the vehicle should store enough estimator history to reconstruct the decision. This requirement links GNC design with telemetry bandwidth and fault-management architecture rather than treating navigation as an isolated algorithm.

Mission integration note

A final navigation check is time integrity. State estimates, measurements and actuator commands are meaningful only when their timestamps refer to the same time base. Millisecond-level offsets can matter during fast descent, while longer delays matter for surface localisation and control handoff. Estimators should therefore track sensor latency, buffering and clock synchronisation explicitly. When a measurement arrives late, the software may need to update a past state and then re-propagate rather than treating the observation as current. Time tagging is thus part of navigation accuracy, not an administrative metadata detail.

Additional operational assurance

Navigation architecture should finally distinguish accuracy from integrity. Accuracy describes how close the estimated state is expected to be to truth, while integrity concerns the system's ability to warn when that estimate should no longer be trusted. A slightly less accurate estimator with reliable fault detection can be safer than a very precise estimator that silently becomes wrong. Descent and autonomous surface operations therefore need alert limits, time-to-alarm requirements and

independent monitors. These integrity concepts connect estimator statistics with operational safety: the crew or guidance law needs not only a position estimate, but confidence that a hazardous error will be detected before it becomes unrecoverable.

Last readiness check

Integrity monitoring should therefore be treated as a companion to estimation accuracy. The system must detect when uncertainty, residual behaviour or sensor disagreement has crossed a boundary that makes the current navigation solution unsafe for the next manoeuvre.

Operational review checklist

Before accepting a navigation estimate, declare the reference frame, state components, measurement units and uncertainty model. Check innovation sign, geometry and covariance behavior, then state the guidance or control action supported by the estimate.

Design for degraded navigation: ask what happens when a landmark disappears, Doppler geometry weakens, wheel slip grows or a sensor bias develops. Safe autonomy requires a credible fallback, not blind continuation.