

Robotics, autonomy and off-Earth maintenance

Design robots that perceive, plan, manipulate and fail safely when terrain, dust or communications invalidate their assumptions.

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. A robot is a perception–decision–action chain

A planetary robot combines sensing, estimation, planning, control, actuators and supervision. Failure can enter anywhere: a blinded camera, drifting localization, over-optimistic planner, seized joint or ambiguous human command.

Analysis must follow the entire chain. Saying “the arm works” is meaningless if perception cannot localize the part to grasp or software lacks a safe mode when the gripper cannot confirm contact.

2. Perception and localization in sparse terrain

Mars brings dust, hard shadows, relief, lighting changes and no terrestrial GNSS. Robots can combine stereo vision, lidar, inertial sensing, odometry and maps. SLAM—simultaneous localization and mapping—builds or corrects a map while estimating the robot pose.

A map is never the complete world. Autonomy needs confidence measures and must know when to slow down, stop or request supervision because the scene no longer matches its assumptions.

3. Planning: choose a path that remains recoverable

The shortest route is not always best. Mars mobility should include slope, slip, energy, visibility, return margin, thermal exposure and rescue options. A cost function can combine these constraints, but its weights must be explicit.

A logistics robot may rationally choose a longer route if it reduces wheel wear or keeps a recovery path open.

4. Manipulation: grasping is a tolerance problem

A robotic arm needs geometry, force limits, compliance and contact logic. A rigid grasp can damage hardware; a weak grasp can drop it. Dust and temperature also change interfaces used by both robots and suited humans.

Future equipment should be robot-friendly: standardized handles, visual fiducials, tolerant mechanical interfaces, protected connectors and accessible grasp zones.

5. Adjustable autonomy and human–robot teaming

Autonomy can be adjustable. Repetitive well-known work may be delegated; novel conditions can return to human supervision. The system should expose what it believes, what it intends and why it stopped.

The goal is not to eliminate people but to reserve human attention for decisions where judgment adds most value.

6. Multi-robot cooperation and common-cause failure

Several small robots can share mapping, transport and inspection. Cooperation increases capability but adds communications, synchronization and coordination. Three robots running identical software can also share the same defect.

Fleet resilience must therefore distinguish numerical redundancy from real diversity. One bad software update should not immobilize the entire fleet simultaneously.

7. Robotic maintenance and diagnostics

A useful Mars robot should expose wear before failure through motor current, vibration, temperature, mechanical backlash, motion time and position error. Trends support condition-based maintenance.

The machine must also be designed to be repaired: replaceable modules, accessible fasteners, compatible tools, locally manufacturable parts where feasible and a verifiable return-to-service procedure.

8. Worked example: energy autonomy of a robotic sortie

A rover has 8.0 kWh usable. Locomotion averages 600 W while moving and instruments consume 250 W continuously. For a 6 h sortie with 4 h driving, energy is $0.600 \times 4 + 0.250 \times 6 = 2.4 + 1.5 = 3.9$ kWh.

If a 30% reserve of capacity must remain untouched, mission energy is limited to $8.0 \times 0.70 = 5.6$ kWh. Remaining margin is $5.6 - 3.9 = 1.7$ kWh for cold, detours or wheel slip.

9. From rover speed to decision autonomy

An autonomous robot is not simply an uncrewed vehicle. It must turn a goal into actions, observe whether the real world still matches its model, detect divergence, and decide whether to continue, slow down, re-route or stop. A useful first distinction is between **movement speed** and **decision speed**.

Teaching assumption. A rover must cover 120 m across previously assessed terrain at a safe average speed of 0.12 m/s. Let L be path length in metres, v average speed in metres per second and t duration in seconds. Because $t = L / v$, $120 \text{ m} / 0.12 \text{ m/s} = 1,000 \text{ s}$, or 16 min 40 s. This is movement time, not mission time: stops, perception, replanning and safety checks must still be added.

If the communications architecture imposes roughly 25 min for a radio round trip in the module 30 example, asking Earth to approve each obstacle would take longer than the traverse itself. Autonomy therefore becomes a safety and productivity requirement. The rover needs at least an operating envelope, a reason-coded safe stop and a recoverable mission state that a local crew can understand.

Inverse problem. If 300 m must be covered in 40 min of actual movement, $40 \text{ min} = 2,400 \text{ s}$. Required average speed is $v = L / t = 300 / 2,400 = 0.125 \text{ m/s}$. This does not prove the terrain permits that speed; it creates a requirement that perception, traction, slope, energy and risk must then validate.

Progressive exercise

A four-robot fleet must move 2,400 kg of regolith. Each robot carries 80 kg per trip and one trip takes 25 minutes, but only 75% of scheduled time is productive. Compute the ideal completion time and identify at least three causes that break the independence assumption.

Reasoned correction

The job requires $2,400 / 80 = 30$ trips. Perfectly divided among four robots, that is 7.5 trips per robot, or 187.5 productive minutes at 25 minutes per trip. If only 75% of scheduled time is productive, the planned duration becomes $187.5 / 0.75 = 250$ minutes, about 4 h 10 min. That result assumes independent robots. It becomes optimistic when they share a charger, loading point, narrow route, operator, radio link or common failure mode. Queueing, rough terrain or one disabled vehicle can therefore reduce the throughput of the entire fleet.

Mini-project

Design an exterior maintenance robot for solar arrays, radiators and antennas: sensors, mobility, arm, autonomy, safing, replaceable modules, health log and task sharing with an astronaut on EVA.

Primary sources and pathways

- [NASA — Autonomous Systems & Robotics](#)
- [NASA JSC — Robotics](#)
- [NASA — Lunar Surface Technology: autonomy and robotics](#)

FR ↔ EN parity matrix

Robotic energy endurance	Autonomie énergétique robotique
Travelable range	Portée parcourable
Robotic localization uncertainty	Incertitude de localisation robotique
Traction margin	Marge de traction
Robot utilization	Taux d'utilisation robotique
Robotic fleet availability	Disponibilité de flotte robotique

Robotics-autonomy foundations and mission reasoning

This English robotics module now matches the certified French teaching intent by connecting sensors, confidence, planning, actuation and maintenance to explicit safe-state and fleet decisions.

Four concepts to master first

autonomy

Definition. Autonomy is the ability of a system to pursue defined goals without immediate human intervention, within explicit limits and rules.

Mission example. A rover can detect an obstacle, choose a local path and stop when confidence becomes too low.

Pitfall. Autonomy is not unlimited freedom and does not remove the need for fault containment.

Mental check. State what decisions the robot may make, which evidence it uses and what condition forces a safe stop or human escalation.

Evidence

Use mission-replay logs showing the robot's perception confidence, selected action, safety constraints and escalation behavior in nominal and off-nominal cases.

Decision use

If autonomy leaves the validated envelope or confidence falls too low, the robot must stop, enter a safe state or request higher-level assistance.

perception

Definition. Perception converts sensor data into useful information about the environment or robot state.

Mission example. Cameras, lidar, wheel encoders and inertial sensors can jointly estimate terrain, obstacles and slip.

Pitfall. A sensor reading is not automatically a correct world model; lighting, dust and geometry can create ambiguity.

Mental check. Compare independent sensing modes and track confidence instead of treating perception as a binary success/failure flag.

Evidence

Use labelled sensor data and test scenes covering lighting, dust, terrain and obstacle conditions, with measured detection errors and confidence.

Decision use

Poor or uncertain perception should reduce speed, change sensing strategy or stop motion rather than let the planner assume the world model is correct.

actuator

Definition. An actuator turns a command into physical action, such as wheel torque, arm motion, valve movement or latch operation.

Mission example. A commanded wheel speed matters only if electronics, motor and drivetrain can deliver the required torque.

Pitfall. A larger command does not guarantee a larger real action when saturation, friction or a fault intervenes.

Mental check. Measure the physical response and compare commanded versus achieved motion or force.

Evidence

Use commanded and measured torque, speed, current and temperature from the actuator under representative loads and environmental conditions.

Decision use

Saturation, overheating or unexplained command-tracking error requires load reduction, fault isolation or a safe stop before continued operation.

condition-based maintenance

Definition. Condition-based maintenance uses indicators of actual equipment health to trigger inspection or intervention rather than relying only on a fixed calendar.

Mission example. Rising motor current, vibration or temperature can reveal wheel degradation before complete failure.

Pitfall. A single noisy excursion should not automatically trigger replacement.

Mental check. Use trends, thresholds, corroborating indicators and mission consequence to decide whether the evidence supports intervention.

Evidence

Use health trends such as vibration, temperature, motor current and diagnostic events tied to known component condition over repeated sorties.

Decision use

A persistent degrading trend should trigger inspection, maintenance or spare replacement before functional failure occurs.

Calculation laboratory

Treat every robotics calculation as part of a control loop. Identify what is sensed, what is estimated, what command is issued, which physical limit applies, and what independent check prevents unsafe continuation.

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.

Robotic energy endurance

$$t_{\text{end}} = E_{\text{useful}} / P_{\text{avg}}$$

1 — Concrete question

What does “ $t_{\text{end}} = E_{\text{useful}} / P_{\text{avg}}$ ” compute in “Robotic energy endurance”?

2 — Intuition without symbols

A battery gives mission duration only after usable energy is compared with actual average power.

3 — Quantities

t_{end} : energy endurance [h]; E_{useful} : usable energy [kWh]; P_{avg} : average power [kW]

4 — Formula

$$t_{\text{end}} = E_{\text{useful}} / P_{\text{avg}}$$

5 — Read aloud

Read “ $t_{\text{end}} = E_{\text{useful}} / P_{\text{avg}}$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

t_{end} : energy endurance [h]; E_{useful} : usable energy [kWh]; P_{avg} : average power [kW]

7 — Pronunciation

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

8 — Units

t_{end} [h]; E_{useful} [kWh]; P_{avg} [kW]

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 $E_{\text{useful}} = 24$ kWh, $P_{\text{avg}} = 3$ kW: $t_{\text{end}} = 24 / 3 = 8$ h.

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

Keep a return reserve and never plan to theoretical depletion.

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 $E_{\text{useful}} = 15 \text{ kWh}$, $P_{\text{avg}} = 2.5 \text{ kW}$: $t_{\text{end}} = 15 / 2.5$?

Detailed guided correction — open after trying

With $E_{\text{useful}} = 15 \text{ kWh}$, $P_{\text{avg}} = 2.5 \text{ kW}$: $t_{\text{end}} = 15 / 2.5 = 6 \text{ h}$. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $E_{\text{useful}} = 40 \text{ kWh}$, $P_{\text{avg}} = 5 \text{ kW}$: $t_{\text{end}} = 40 / 5$?

Autonomous correction — open after trying

With $E_{\text{useful}} = 40 \text{ kWh}$, $P_{\text{avg}} = 5 \text{ kW}$: $t_{\text{end}} = 40 / 5 = 8 \text{ h}$. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Keep a return reserve and never plan to theoretical depletion.

Travelable range

$$d_{\text{range}} = v_{\text{rover}} \times t_{\text{end}}$$

1 — Concrete question

What does " $d_{\text{range}} = v_{\text{rover}} \times t_{\text{end}}$ " compute in "Travelable range"?

2 — Intuition without symbols

Range follows from sustainable speed over the time actually available.

3 — Quantities

d_{range} : travelable distance [km]; v_{rover} : average rover speed [km/h]; t_{end} : available endurance [h]

4 — Formula

$$d_{\text{range}} = v_{\text{rover}} \times t_{\text{end}}$$

5 — Read aloud

Read " $d_{\text{range}} = v_{\text{rover}} \times t_{\text{end}}$ " by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

d_{range} : travelable distance [km]; v_{rover} : average rover speed [km/h]; t_{end} : available endurance [h]

7 — Pronunciation

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

8 — Units

d_{range} [km]; v_{rover} [km/h]; t_{end} [h]

9 — Convention

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

10 — Why this operation

Multiplication converts a rate, fraction or specific quantity into a total.

11 — Assumptions

Factors must belong to the same scenario and their units must combine correctly.

12 — Independent check

Dividing the result by a non-zero factor should recover the product of the others.

13 — Numerical case

With $v_{\text{rover}} = 6 \text{ km/h}$, $t_{\text{end}} = 8 \text{ h}$: $d_{\text{range}} = 6 \times 8 = 48 \text{ km}$.

14 — Why the calculation works

Multiplication converts a rate, fraction or specific quantity into a total.

15 — Verification

Dividing the result by a non-zero factor should recover the product of the others.

16 — Mental estimate

Estimating each factor separately quickly predicts product scale.

17 — Interpretation

Apply return, detour and degradation margin before authorizing the sortie.

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

A relative change in one factor transfers directly to the product when the others stay fixed.

20 — Guided and autonomous exercises

Guided exercise. Recalculate this scenario: With $v_{\text{rover}} = 4 \text{ km/h}$, $t_{\text{end}} = 6 \text{ h}$: $d_{\text{range}} = 4 \times 6$?

Detailed guided correction — open after trying

With $v_{\text{rover}} = 4 \text{ km/h}$, $t_{\text{end}} = 6 \text{ h}$: $d_{\text{range}} = 4 \times 6 = 24 \text{ km}$. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $v_{\text{rover}} = 10 \text{ km/h}$, $t_{\text{end}} = 3.5 \text{ h}$: $d_{\text{range}} = 10 \times 3.5$?

Autonomous correction — open after trying

With $v_{\text{rover}} = 10 \text{ km/h}$, $t_{\text{end}} = 3.5 \text{ h}$: $d_{\text{range}} = 10 \times 3.5 = 35 \text{ km}$. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Apply return, detour and degradation margin before authorizing the sortie.

Robotic localization 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 “Robotic localization uncertainty”?

2 — Intuition without symbols

Localization combines independent axis uncertainties into a position envelope.

3 — Quantities

σ_{pos} : combined uncertainty [m]; σ_x : x-axis uncertainty [m]; σ_y : y-axis 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 uncertainty [m]; σ_x : x-axis uncertainty [m]; σ_y : y-axis 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 $\sigma_x = 2$ m, $\sigma_y = 3$ m: $\sigma_{\text{pos}} = \sqrt{2^2 + 3^2} = 3.6056$ 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

Reduce speed or mission scope when the envelope becomes too large for obstacle avoidance or return navigation.

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 $\sigma_x = 1$ m, $\sigma_y = 1$ m: $\sigma_{\text{pos}} = \sqrt{1^2 + 1^2}$?

Detailed guided correction — open after trying

With $\sigma_x = 1$ m, $\sigma_y = 1$ m: $\sigma_{\text{pos}} = \sqrt{1^2 + 1^2} = 1.4142$ m. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $\sigma_x = 0.3$ m, $\sigma_y = 0.4$ m: $\sigma_{\text{pos}} = \sqrt{0.3^2 + 0.4^2}$?

Autonomous correction — open after trying

With $\sigma_x = 0.3$ m, $\sigma_y = 0.4$ m: $\sigma_{\text{pos}} = \sqrt{0.3^2 + 0.4^2} = 0.5$ m. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Reduce speed or mission scope when the envelope becomes too large for obstacle avoidance or return navigation.

Traction margin

$$F_{\text{margin}} = \mu \times N_{\text{load}} - F_{\text{resist}}$$

1 — Concrete question

What does “ $F_{\text{margin}} = \mu \times N_{\text{load}} - F_{\text{resist}}$ ” compute in “Traction margin”?

2 — Intuition without symbols

Available traction must exceed resisting forces to preserve mobility.

3 — Quantities

F_{margin} : traction margin [N]; μ : traction coefficient [sans dimension]; N_{load} : normal load [N]; F_{resist} : resisting force [N]

4 — Formula

$$F_{\text{margin}} = \mu \times N_{\text{load}} - F_{\text{resist}}$$

5 — Read aloud

Read “ $F_{\text{margin}} = \mu \times N_{\text{load}} - F_{\text{resist}}$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

F_{margin} : traction margin [N]; μ : traction coefficient [sans dimension]; N_{load} : normal load [N]; F_{resist} : resisting force [N]

7 — Pronunciation

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

8 — Units

F_{margin} [N]; μ [sans dimension]; N_{load} [N]; F_{resist} [N]

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 $\mu = 0.6$ sans dimension, $N_{\text{load}} = 1200$ N, $F_{\text{resist}} = 500$ N: $F_{\text{margin}} = 0.6 \times 1200 - 500 = 220$ N.

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 small margin rules out nominal-speed assumptions and requires a more conservative route or load.

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 $\mu = 0.4$ sans dimension, $N_{\text{load}} = 900$ N, $F_{\text{resist}} = 300$ N: $F_{\text{margin}} = 0.4 \times 900 - 300$?

Detailed guided correction — open after trying

With $\mu = 0.4$ sans dimension, $N_{\text{load}} = 900$ N, $F_{\text{resist}} = 300$ N: $F_{\text{margin}} = 0.4 \times 900 - 300 = 60$ N. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $\mu = 0.7$ sans dimension, $N_{\text{load}} = 1500$ N, $F_{\text{resist}} = 800$ N: $F_{\text{margin}} = 0.7 \times 1500 - 800$?

Autonomous correction — open after trying

With $\mu = 0.7$ sans dimension, $N_{\text{load}} = 1500$ N, $F_{\text{resist}} = 800$ N: $F_{\text{margin}} = 0.7 \times 1500 - 800 = 250$ N. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

A small margin rules out nominal-speed assumptions and requires a more conservative route or load.

Robot utilization

$U_{\text{robot}} = t_{\text{active}} / t_{\text{window}}$

1 — Concrete question

What does “ $U_{\text{robot}} = t_{\text{active}} / t_{\text{window}}$ ” compute in “Robot utilization”?

2 — Intuition without symbols

Utilization compares productive time with total mission time.

3 — Quantities

U_robot: utilization [sans dimension]; t_active: active time [h]; t_window: total window [h]

4 — Formula

$$U_{\text{robot}} = t_{\text{active}} / t_{\text{window}}$$

5 — Read aloud

Read “ $U_{\text{robot}} = t_{\text{active}} / t_{\text{window}}$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

U_robot: utilization [sans dimension]; t_active: active time [h]; t_window: total window [h]

7 — Pronunciation

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

8 — Units

U_robot [sans dimension]; t_active [h]; t_window [h]

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 $t_{\text{active}} = 6 \text{ h}$, $t_{\text{window}} = 8 \text{ h}$: $U_{\text{robot}} = 6 / 8 = 0.75$.

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

Very high utilization leaves little margin for diagnostics, charging or recovery.

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 $t_{\text{active}} = 10 \text{ h}$, $t_{\text{window}} = 12 \text{ h}$: $U_{\text{robot}} = 10 / 12$?

Detailed guided correction — open after trying

With $t_{\text{active}} = 10 \text{ h}$, $t_{\text{window}} = 12 \text{ h}$: $U_{\text{robot}} = 10 / 12 = 0.8333$. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $t_{\text{active}} = 3 \text{ h}$, $t_{\text{window}} = 10 \text{ h}$: $U_{\text{robot}} = 3 / 10$?

Autonomous correction — open after trying

With $t_{\text{active}} = 3 \text{ h}$, $t_{\text{window}} = 10 \text{ h}$: $U_{\text{robot}} = 3 / 10 = 0.3$. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Very high utilization leaves little margin for diagnostics, charging or recovery.

Robotic fleet availability

$$A_{\text{fleet}} = n_{\text{ready}} / n_{\text{total}}$$

1 — Concrete question

What does “ $A_{\text{fleet}} = n_{\text{ready}} / n_{\text{total}}$ ” compute in “Robotic fleet availability”?

2 — Intuition without symbols

Real fleet capability depends on units actually ready, not units purchased.

3 — Quantities

A_{fleet} : fraction of robots ready [sans dimension]; n_{ready} : ready robots [robot]; n_{total} : total robots [robot]

4 — Formula

$$A_{\text{fleet}} = n_{\text{ready}} / n_{\text{total}}$$

5 — Read aloud

Read “ $A_{\text{fleet}} = n_{\text{ready}} / n_{\text{total}}$ ” by naming every operation, subscript and grouping explicitly.

6 — Symbols and meaning

A_{fleet} : fraction of robots ready [sans dimension]; n_{ready} : ready robots [robot]; n_{total} : total robots [robot]

7 — Pronunciation

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

8 — Units

A_{fleet} [sans dimension]; n_{ready} [robot]; n_{total} [robot]

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 $n_{\text{ready}} = 8$ robot, $n_{\text{total}} = 10$ robot: $A_{\text{fleet}} = 8 / 10 = 0.8$.

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

Delay a mission when ready robots no longer cover critical functions and planned redundancy.

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 $n_{\text{ready}} = 5$ robot, $n_{\text{total}} = 6$ robot: $A_{\text{fleet}} = 5 / 6$?

Detailed guided correction — open after trying

With $n_{\text{ready}} = 5$ robot, $n_{\text{total}} = 6$ robot: $A_{\text{fleet}} = 5 / 6 = 0.8333$. The decision must then be checked against the module margins and assumptions.

Autonomous exercise. Recalculate this scenario: With $n_{\text{ready}} = 12$ robot, $n_{\text{total}} = 15$ robot: $A_{\text{fleet}} = 12 / 15$?

Autonomous correction — open after trying

With $n_{\text{ready}} = 12$ robot, $n_{\text{total}} = 15$ robot: $A_{\text{fleet}} = 12 / 15 = 0.8$. The decision must then be checked against the module margins and assumptions.

21 — Mission decision

Delay a mission when ready robots no longer cover critical functions and planned redundancy.

Mission reasoning

Autonomy under communication delay

Mars robots cannot wait many minutes for Earth to approve every obstacle avoidance decision. Local autonomy therefore handles perception, short-horizon planning and protective stops. Mission rules define the authority envelope: which terrain can be traversed, how much uncertainty is tolerable and when the vehicle must pause. The safest autonomous system is not the one that always continues, but the one that knows when confidence has left the validated domain.

Perception needs diversity

Dust, shadows and low texture can defeat vision exactly when a rover needs reliable localisation. Sensor diversity reduces dependence on one failure mode. Wheel odometry, inertial sensors, lidar, radar or terrain-relative navigation can provide complementary information. Fusion must still model correlation and common environmental effects; multiple sensors do not automatically mean independent evidence.

Maintenance is an operational capability

On Mars, maintenance determines availability because replacement hardware may be months away. Robots should expose diagnostic data, allow inspection access and support modular replacement where mass permits. Condition-based maintenance combines trends with mission consequence: a slowly rising bearing temperature may justify planned intervention before a critical traverse, while the same trend on a nonessential unit could be monitored.

Fleet resilience and shared bottlenecks

Several robots can increase throughput and provide redundancy, but only if common resources do not become single points of failure. A four-rover fleet that shares one charger, one narrow route and one operator can lose most of its theoretical parallelism. Fleet planning therefore models queueing, spares, charging, communications and recovery towing rather than multiplying one-robot productivity by the vehicle count.

Robotics practice — decide the safe action before consulting the correction

Exercise A — Autonomy boundary

A rover sees an obstacle class that was absent from training and confidence drops below the validated threshold. What should a well-designed autonomy policy do?

Detailed correction — Exercise A

Stop or enter the predefined safe state, preserve diagnostic data and request higher-level assistance if communication is available. Continuing with low confidence outside the validated domain converts uncertainty into uncontrolled risk.

When perception confidence leaves the validated range, stop hazardous motion or enter the declared safe state; continuing as if the classifier were certain converts uncertainty into unmanaged risk.

Exercise B — Perception diversity

Explain why camera + wheel odometry is stronger than either source alone on nominal terrain, but still may fail during a dust event and severe wheel slip.

Detailed correction — Exercise B

The sources constrain different errors, so fusion can improve nominal estimation. Dust can degrade the camera while slip corrupts wheel odometry at the same time; the pair therefore still needs confidence monitoring or another sensing mode.

Explain the complementarity and the shared vulnerability: camera constrains visual geometry, odometry constrains short-term motion, but dust and severe slip can degrade both at the same time.

Exercise C — Energy autonomy

A robot has 7.5 kWh usable energy and averages 1.5 kW. Estimate operating duration before applying mission reserve.

Detailed correction — Exercise C

$7.5/1.5 = 5$ h. Operational duration must be shorter once reserve, temperature effects and return energy are included.

Divide usable energy by average power only after confirming compatible units: 7.5 kWh / 1.5 kW gives 5 h before the mission reserve and other derating factors are applied.

Exercise D — Condition trend

Motor current rises gradually over five sorties while commanded torque and terrain remain comparable. What evidence should be checked before replacing the motor?

Detailed correction — Exercise D

Check temperature, vibration, wheel slip, supply voltage, calibration and mechanical drag. A consistent multi-sensor trend strengthens the degradation hypothesis; a single current trend can also reflect changing load or instrumentation error.

Compare motor current against commanded torque, terrain and temperature across sorties; a persistent rise under comparable demand is evidence of degradation that merits targeted diagnosis.

Exercise E — Fleet bottleneck

Four robots each need 20 minutes of charging after every 40 minutes of work, but only one charger exists. Why is simple four-times throughput impossible?

Detailed correction — Exercise E

The charger serialises part of the operation and creates a queue. Fleet output is limited by the shared charging resource, schedule and battery capacity, so throughput must be calculated as a coupled system rather than by multiplying one-robot output by four.

Model the charger as a shared queueing resource: four robots cannot all achieve their individual 40/20 work-charge cycle when one charger serializes every recharge.

Interactive beginner glossary

Use the interactive vocabulary to follow information from sensor to decision to physical action and then into maintenance evidence. Each definition should explain where uncertainty or failure can enter that chain.

- **autonomy** — Autonomy is a robot's ability to perceive, decide and act locally without continuous human commands, while remaining within validated rules, authority and safety constraints.
- **perception** — Perception converts raw sensor data into useful estimates of the environment or robot state, such as terrain, obstacles, pose, hazards or equipment condition.
- **sensor** — A sensor converts a physical quantity into data. Its output always has limits such as range, resolution, noise, bias, latency and environmental sensitivity.
- **camera** — A camera records light intensity or color across an image. It provides rich terrain information but depends on illumination, visibility, calibration and interpretation.
- **lidar** — Lidar estimates distance by emitting laser light and measuring its return. It can build precise geometric maps but performance can degrade with dust, reflectivity and range.
- **radar** — Radar uses radio waves to detect distance, motion or structure. It can operate in darkness and some obscurants but provides different resolution and interpretation from optical sensors.

- wheel odometry — Wheel odometry estimates rover motion from wheel rotation. It is simple and high-rate but accumulates error when wheels slip, sink or deform the terrain.
- inertial sensor — An inertial sensor measures angular motion or specific force with gyroscopes and accelerometers. Its estimates are available continuously but drift when small errors are integrated over time.
- sensor fusion — Sensor fusion combines complementary measurements so weaknesses in one source can be constrained by others while uncertainty and disagreement remain explicit.
- confidence — Confidence is a quantified or classified indication of how trustworthy a perception or estimate is under current conditions. Low confidence should change robot behavior rather than be ignored.
- path planning — Path planning selects a feasible route from the current state to a goal while accounting for obstacles, terrain cost, vehicle capability, energy and safety constraints.
- obstacle avoidance — Obstacle avoidance detects and steers around hazards during motion, usually on a shorter time scale than global path planning.
- safe state — A safe state is a predefined configuration that minimizes immediate hazard, stops risky motion, preserves critical energy and keeps enough capability for diagnosis or recovery.
- actuator — An actuator converts a command into physical action, such as wheel torque, joint motion, valve movement or tool force.
- motor — A motor converts electrical energy into mechanical rotation or force. Its usable output is limited by torque, speed, current, temperature and gearing.
- torque — Torque is a turning effect of force about an axis, measured in newton-metres. Robotic joints and wheels must provide enough torque without exceeding structural or thermal limits.
- saturation — Saturation occurs when an actuator reaches a physical or commanded limit and cannot produce the additional force, torque, speed or rate requested by the controller.
- robot arm — A robot arm is a chain of powered joints and links used to position an end effector for manipulation, inspection, repair or sample handling.
- condition-based maintenance — Condition-based maintenance triggers inspection or intervention from measured equipment health indicators rather than relying only on fixed calendar intervals.
- diagnostics — Diagnostics is the process of detecting, isolating and identifying faults from symptoms, sensor data, tests and system models.
- vibration — Vibration is repeated mechanical motion around an equilibrium. Changes in amplitude or frequency content can reveal imbalance, looseness, bearing wear or structural problems.
- temperature trend — A temperature trend is the change of temperature over time rather than a single reading. A rising trend can reveal growing friction, cooling loss or electrical overload before a limit is crossed.
- predictive maintenance — Predictive maintenance uses condition trends and models to estimate future degradation or remaining useful life so maintenance can be scheduled before failure.
- spare — A spare is a replacement component or consumable reserved to restore function after wear, damage or failure. Its usefulness depends on compatibility, storage life and replacement capability.
- modularity — Modularity divides a system into replaceable units with controlled interfaces so repair, upgrades and reconfiguration can be performed without redesigning the whole robot.
- fleet — A fleet is a group of robots managed as a coordinated operational resource. Fleet performance depends on task allocation, availability, charging, maintenance and shared infrastructure.
- throughput — Robotic throughput is the useful work completed per unit time, such as tonnes moved, inspections completed or parts delivered, after delays and downtime are included.
- queueing — Queueing occurs when tasks or robots wait for a limited resource such as a charger, airlock, workstation or communications link. Waiting time can dominate fleet productivity.
- charger — A charger transfers electrical energy into a robot's battery within voltage, current, temperature and state-of-charge limits. Charger availability can become a fleet bottleneck.
- common-cause failure — A common-cause failure defeats multiple supposedly independent units through one shared cause, such as the same software defect, dust environment, power bus or maintenance error.

Operational depth: from calculation to mission decision

Autonomy needs a validated envelope

Autonomous behaviour should be defined by what has been validated, not by what software can technically attempt. Terrain slope, lighting, dust density, communication state and mechanical health may all define the authorised operating envelope. When observations fall outside that envelope, a safe system reduces speed, stops, requests help or changes mode. The policy must be testable. Saying that a robot should “use judgement” is not enough unless confidence metrics and escalation rules make that judgement observable.

Perception confidence and disagreement

Sensor fusion works best when the system can notice disagreement. If wheel odometry says the rover moved ten metres but visual landmarks barely shift, the estimator should consider slip. If a camera sees an obstacle that lidar does not, the robot should reason about range, field of view and dust rather than simply averaging the two. Confidence values become operational inputs. They decide whether planning can continue normally, must slow down or should stop while a better observation is acquired.

Maintenance design starts in hardware

A robot that cannot be opened, diagnosed or safely manipulated is difficult to maintain no matter how good the maintenance procedure is. Mars hardware benefits from accessible fasteners, modular connectors, lifting points, diagnostic ports and components sized for suited or robotic handling. Every maintenance task consumes tools, crew time and contamination control. Design reviews should therefore estimate replacement time and failure isolation as seriously as nominal performance.

Fleet scheduling with shared resources

Fleet productivity is constrained by chargers, communications windows, loading stations, maintenance bays and operator attention. Queueing can dominate when several robots return at once. A dispatch algorithm should account for battery state, task urgency, travel time and recovery options instead of sending every available vehicle to the highest-priority task. Resilience improves when the fleet can redistribute work after one robot fails without creating a new bottleneck elsewhere.

Learning without uncontrolled behaviour

Robotic systems may adapt models or policies from experience, but mission-critical adaptation needs boundaries. A learned change should not silently invalidate verified safety constraints. Safer architectures separate adaptive performance improvements from hard protective layers, log model updates and allow rollback. Engineers evaluate whether new behaviour remains inside known force, speed, terrain and energy limits. Learning is useful when it improves prediction or efficiency without turning validation into a moving target.

Operational review checklist

Before accepting a robotic result, identify the sensing conditions, confidence level, actuator limits, energy or time units and mission reserve. State the safe behavior if any assumption or validated limit is violated.

Design the fleet for degraded operation: dust can hide terrain, slip can defeat odometry, a charger can become a bottleneck and one software defect can affect every robot. Recovery must be planned at both robot and fleet level.