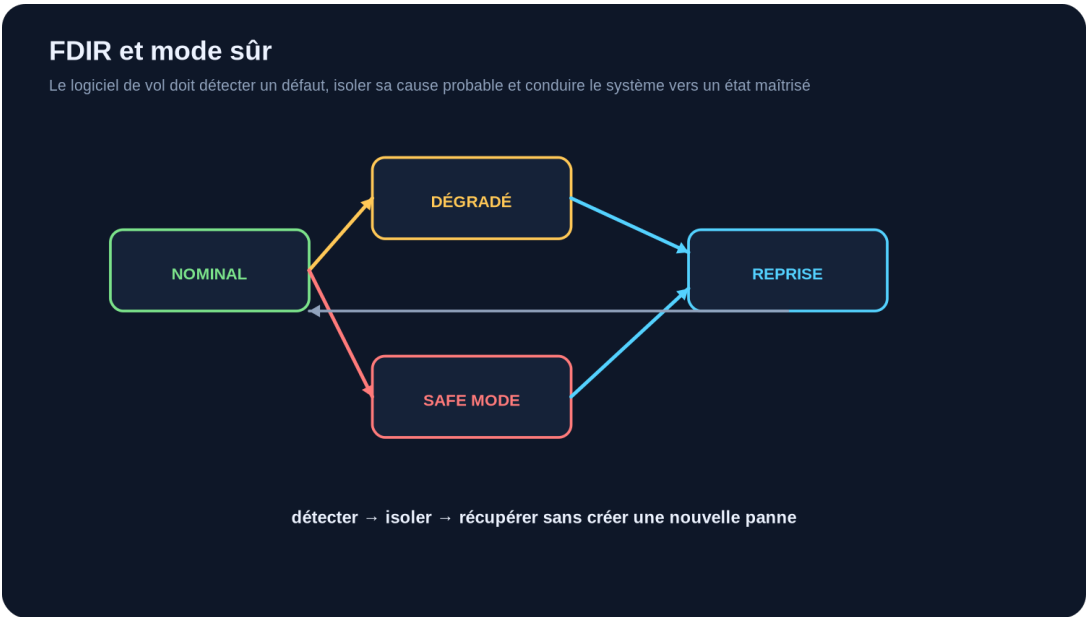


Logiciel de vol, watchdog, FDIR, redondance et mode sûr

Réponse essentielle

Logiciel de vol, watchdog, FDIR, redondance et mode sûr. La question à résoudre est la suivante : Comment le vaisseau se protège-t-il quand un capteur devient incohérent ou un calculateur se bloque ? Ici, la compréhension du système compte plus qu'une équation unique. On raisonne avec des états, des interfaces, des marges et des critères de réussite. La suite du cours montre comment passer de cette idée à un raisonnement contrôlable : unités explicites, hypothèses annoncées, calculs reproductibles, ordre de grandeur et limites d'interprétation. Le résultat n'est utile que si l'on peut expliquer ce qu'il mesure, d'où viennent les nombres et quelle décision d'ingénierie il éclaire.



Configuration logique autorisant certaines actions.

Surveillance attendant un signe de vie.

Avant de calculer

Grandeur:	Logiciel de vol, watchdog, FDIR, redondance et mode sûr
Contrôle:	unités + ordre de grandeur + hypothèses
Décision:	expliquer ce que le résultat autorise réellement à conclure

Logiciel de vol, watchdog, FDIR, redondance et mode sûr

Configuration logique autorisant certaines actions.

Surveillance attendant un signe de vie.

Détection, isolation et récupération après défaut.

Moyen supplémentaire pouvant reprendre une fonction.

Dépendance pouvant faire tomber plusieurs voies.

État simplifié privilégiant la survie et le diagnostic.

3 — Voir l'architecture avant de calculer

Seuils, cohérence, votes et temporisations.

Identifier la zone probable, pas seulement signaler une alarme.

Réessayer, redémarrer, basculer ou désactiver.

Réduire la complexité pour conserver énergie, thermique et communication.

4 — Les formules, seulement lorsqu'elles répondent à une question

Ici, la compréhension du système compte plus qu'une équation unique. On raisonne avec des états, des interfaces, des marges et des critères de réussite.

5 — Ce que les unités et les marges veulent dire

Seuils gardent l'unité de la grandeur ; temporisations en s ou ms. La logique doit écrire à la fois valeur, durée et état applicable.

6 — Trois démonstrations concrètes, calculées pas à pas

Dépassement 0,2 s, seuil logique 2 s.

$0,2 < 2$: pas de déclenchement

21,0 °C ; 21,3 °C ; 85 °C.

La voie à 85 °C est suspecte, cause non prouvée.

Fonction maintenue, tolérance supplémentaire perdue.

7 — Approfondissement : ce que le schéma simplifié cache

Les transitions explicites réduisent les comportements implicites.

Trop sensible ou trop permissif sont deux risques.

Deux copies identiques peuvent partager la même erreur de conception.

Il faut restaurer un état cohérent après reboot.

Simulations et hardware-in-the-loop testent les cas rares.

8 — Application à un vaisseau Terre-Mars

Lecture active

Sections:	3 — Voir l'architecture avant de calculer ; 4 — Les formules, seulement lorsqu'elles répondent à une question
Vérification:	Une valeur n'est exploitable qu'avec son unité et sa frontière de calcul.
Exercice:	modifiez une hypothèse puis prédissez le sens de variation avant de recalculer

Logiciel de vol, watchdog, FDIR, redondance et mode sûr

9 — Dossier de référence : ce qu'un vrai projet doit encore prendre en compte

Un logiciel spatial robuste décrit explicitement des modes et transitions : initialisation, nominal, manœuvre, communication, mode sûr, récupération, etc. Chaque transition possède des conditions d'entrée et des actions. Cette approche réduit les comportements implicites qui deviennent difficiles à tester. Une commande qui est valide en croisière peut être interdite pendant une manœuvre. Les machines à états rendent ces règles visibles et facilitent la vérification des chemins rares.

Un watchdog est un mécanisme qui attend un signe périodique prouvant que le logiciel ou un processeur fonctionne. Si ce signe disparaît, il peut déclencher un reset ou une reconfiguration. Mais un watchdog mal conçu peut provoquer des redémarrages inutiles ou ne rien détecter lorsqu'un logiciel tourne mais produit de mauvaises décisions. Il faut donc définir ce qui est réellement surveillé : simple activité du processeur, respect d'un délai, cohérence de données ou combinaison de plusieurs critères.

FDIR signifie Fault Detection, Isolation and Recovery. Détecter répond à « quelque chose est-il anormal ? ». Isoler répond à « où est la cause probable ? ». Récupérer répond à « quelle action permet de continuer ou de se mettre en sécurité ? ». Ces trois étapes sont différentes. Couper immédiatement un équipement à la première mesure suspecte peut transformer une fausse alarme en vraie perte de mission. À l'inverse, tolérer trop longtemps une anomalie peut propager le défaut.

Le mode sûr réduit souvent les objectifs pour préserver énergie, température, attitude et communication minimale. Il doit être suffisamment simple pour rester disponible lorsque le reste du système est douteux. Mais il n'est pas magique : un mode sûr exige toujours des capteurs, de l'énergie, du logiciel et des actionneurs. Il faut donc analyser quelles pannes peuvent empêcher même son activation et prévoir des chemins de récupération appropriés.

Deux ordinateurs exécutant exactement le même code peuvent prendre exactement la même mauvaise décision si l'erreur est dans l'algorithme ou les données. On distingue redondance matérielle et diversité de conception. Dans certains cas, une fonction de sauvegarde volontairement plus simple peut être plus robuste qu'une copie complète. Mais la diversité coûte en développement, vérification et maintenance. Le bon choix dépend de la criticité et des scénarios crédibles.

Les pannes critiques apparaissent justement dans des combinaisons peu fréquentes : capteur en défaut pendant une manœuvre, communication perdue durant un redémarrage, batterie froide lors d'une transition, etc. La simulation, le software-in-the-loop puis le hardware-in-the-loop permettent d'injecter des anomalies et de tester les réactions. On vérifie aussi les temporisations, les courses entre tâches et les transitions après reset. Une campagne de test qui ne couvre que le nominal donne une confiance trompeuse.

Un logiciel peut nécessiter une correction après le lancement. Mettre à jour un véhicule lointain exige une image validée, un protocole de transfert robuste, une copie de secours et un moyen de revenir à l'ancienne version si la nouvelle ne démarre pas. Le délai radio impose parfois d'exécuter l'opération avec une grande autonomie locale. Pour une base martienne, conserver outils de compilation, signatures, procédures et compatibilités devient une question de souveraineté technique autant que de maintenance.

Lecture active

Sections: 9 — Dossier de référence : ce qu'un vrai projet doit encore prendre en compte

Vérification: Une valeur n'est exploitable qu'avec son unité et sa frontière de calcul.

Exercice: modifiez une hypothèse puis prédissez le sens de variation avant de recalculer

Logiciel de vol, watchdog, FDIR, redondance et mode sûr

10 — Pièges fréquents et mauvaises intuitions

- Croire que deux copies identiques éliminent les causes communes.
- Utiliser le redémarrage comme réponse universelle.
- Créer un mode sûr jamais testé en chaîne complète.

Le logiciel de vol doit savoir perdre des fonctions sans perdre le vaisseau

- Un watchdog surveille qu'un calculateur ou une tâche continue d'exécuter le comportement attendu. Un redémarrage automatique peut récupérer un blocage transitoire, mais il devient dangereux si la cause persiste et entraîne une boucle de redémarrages. Le logiciel de vol doit donc mémoriser le contexte, limiter certaines répétitions et pouvoir entrer dans un mode sûr.
- Le FDIR sépare détection, isolation et récupération. Une mesure aberrante ne prouve pas immédiatement quel capteur est fautif ; plusieurs capteurs ou modèles peuvent être comparés. Une action de récupération doit ensuite préserver les ressources : couper un équipement, basculer sur une chaîne redondante, modifier une consigne ou attendre une commande terrestre selon l'autonomie autorisée.
- La mise à jour distante ajoute un risque de configuration. Le véhicule doit conserver une image de repli, vérifier intégrité et compatibilité, puis pouvoir revenir à une version connue si l'activation échoue. En espace lointain, cette capacité vaut autant que la nouveauté du logiciel envoyé.

11 — Exercices guidés

- Question : Quelle question faut-il poser avant de choisir un équipement ?
- Question : Pourquoi un résultat nominal ne suffit-il pas ?

12 — À retenir

- Expliquer le sujet avec des mots simples avant les symboles.
- Relier au moins quatre interfaces avec les autres sous-systèmes.
- Refaire les trois exemples numériques sans saut de raisonnement.

13 — Sources NASA pour approfondir

- NASA - 2026 State-of-the-Art: Small Spacecraft Avionics
- NASA - 2026 State-of-the-Art: Ground Data Systems and Mission Operations

Lecture active

Sections:	10 — Pièges fréquents et mauvaises intuitions ; Le logiciel de vol doit savoir perdre des fonctions sans perdre le vaisseau
Vérification:	NASA - 2026 State-of-the-Art: Small Spacecraft Avionics NASA Systems Engineering Handbook NASA - 2026 State-of-the-Art: Ground Data Systems and Mission Operations
Exercice:	modifiez une hypothèse puis prédisez le sens de variation avant de recalculer