Check for updates

SPACE ROBOTS

Online tree-based planning for active spacecraft fault estimation and collision avoidance

James Ragan, Benjamin Riviere, Fred Y. Hadaegh, Soon-Jo Chung*

Copyright © 2024 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works

Autonomous robots operating in uncertain or hazardous environments subject to state safety constraints must be able to identify and isolate faulty components in a time-optimal manner. When the underlying fault is ambiguous and intertwined with the robot's state estimation, motion plans that discriminate between simultaneous actuator and sensor faults are necessary. However, the coupled fault mode and physical state uncertainty creates a constrained optimization problem that is challenging to solve with existing methods. We combined belief-space tree search, marginalized filtering, and concentration inequalities in our method, safe fault estimation via active sensing tree search (s-FEAST), a planner that actively diagnoses system faults by selecting actions that give the most informative observations while simultaneously enforcing probabilistic state constraints. We justify this approach with theoretical analysis showing s-FEAST's convergence to optimal policies. Using our robotic spacecraft simulator, we experimentally validated s-FEAST by safely and successfully performing fault estimation while on a collision course with a model comet. These results were further validated through extensive numerical simulations demonstrating s-FEAST's performance.

INTRODUCTION

Autonomous robots offer the potential for markedly faster operations and better performance in domains ranging from search and rescue (*1*) to planetary exploration (*2*). However, to achieve full autonomy, these robots must be capable of independently diagnosing and recovering from various component faults at a system level. This is especially true when the robot's safety is a function of time-critical constraints, such as maintaining lane keeping during autonomous driving (*3*) or managing the accumulation of environmental degradation (*4*).

Spacecraft are a motivating class of autonomous systems because real-time ground-in-the-loop interventions are difficult, if not impossible, because of limited communication or large time delays. As the use of autonomous space systems increases, so too does the number of failures, with 42.6% of small satellite missions between 2009 and 2016 ending in partial or complete failure (*5*). On Earth, uninhabited aerial vehicles fail on the order of once every 1000 hours of operation (*6*), with partial failures occurring as often as every 10 to 50 hours in some domains (*7*).

We consider the problem of fault estimation onboard robotic spacecraft that will soon violate state safety constraints. One such example is shown in Fig. 1 (B and C) and Movie 1. Here, a robot approaches a model comet, and component failure could jeopardize mission success. In this scenario, we envision a system-level emergency response where safely and autonomously identifying the underlying fault as quickly as possible supersedes primary mission objectives. To this end, we propose s-FEAST (safe fault estimation via active sensing tree search), a planning-based approach that selects diagnostic actions to gather informative observations while satisfying probabilistic state constraints at each planning step. As shown in Fig. 1B, the autonomous spacecraft is subjected to a failure of both of its retro thrusters. Conventional model-based passive fault detection approaches will likely not detect this failure until the spacecraft attempts to maneuver and a discrepancy between the predicted and observed states is noticed. At this point, it may be too late to maintain the safety constraints on the spacecraft's state. Similarly, methods of representing the safety of the spacecraft that are unable to consider uncertainty in the system model will not properly capture the risk of this adversarial fault. Instead, we consider actively gathering information about the fault to be a top priority and necessary to avoid overconfident predictions of safety. With our approach, the robot proactively reorients and diagnoses the failure, avoiding collision (see Fig. 1, E and F, and Movie 1).

Related work

Fault estimation

Traditionally, system-level approaches to fault estimation methods have been passive; actions are not taken to determine the underlying failure. Instead, the input-output data during normal operations are monitored for abnormalities (*8*–*11*). Upon detecting a fault, the operator is alerted, and the robot's operations are halted (*12*); however, the root cause may be ambiguous (*13*). In more recent work, passive fault estimation methods for robotic systems have included data-driven models that account for varying levels of system autonomy (*14*–*16*) as well as distributed systems (*17*). Other approaches to passive fault estimation impose input-output consistency constraints (*18*, *19*). These are widely applicable but must be custom-designed for each system and failure case and have limited capability for uncertainty and noise, as well as low robustness to unmodeled scenarios.

One drawback of passive fault estimation is the possibility of multiple plausible fault scenarios. This motivates the problem of selecting control inputs to gather information about the underlying failures. Although optimality conditions can be derived (*20*), tractable general algorithms do not exist, giving rise to a large body of work in active fault diagnosis (*21*). Early work ensured the diagnosability of finite discrete-event systems (*22*), with extensions to satellite applications (*23*, *24*). In linear systems, actions yielding the least expected overlap in hypotheses can be found (*25*). These can be evaluated greedily in real time (*26*) but are limited to systems with Gaussian noise and constraints on expectations. Related approaches provide identification guarantees in linear systems when zonotopes bound the disturbances

Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA.
*Corresponding author. Email: sjchung@caltech.edu

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

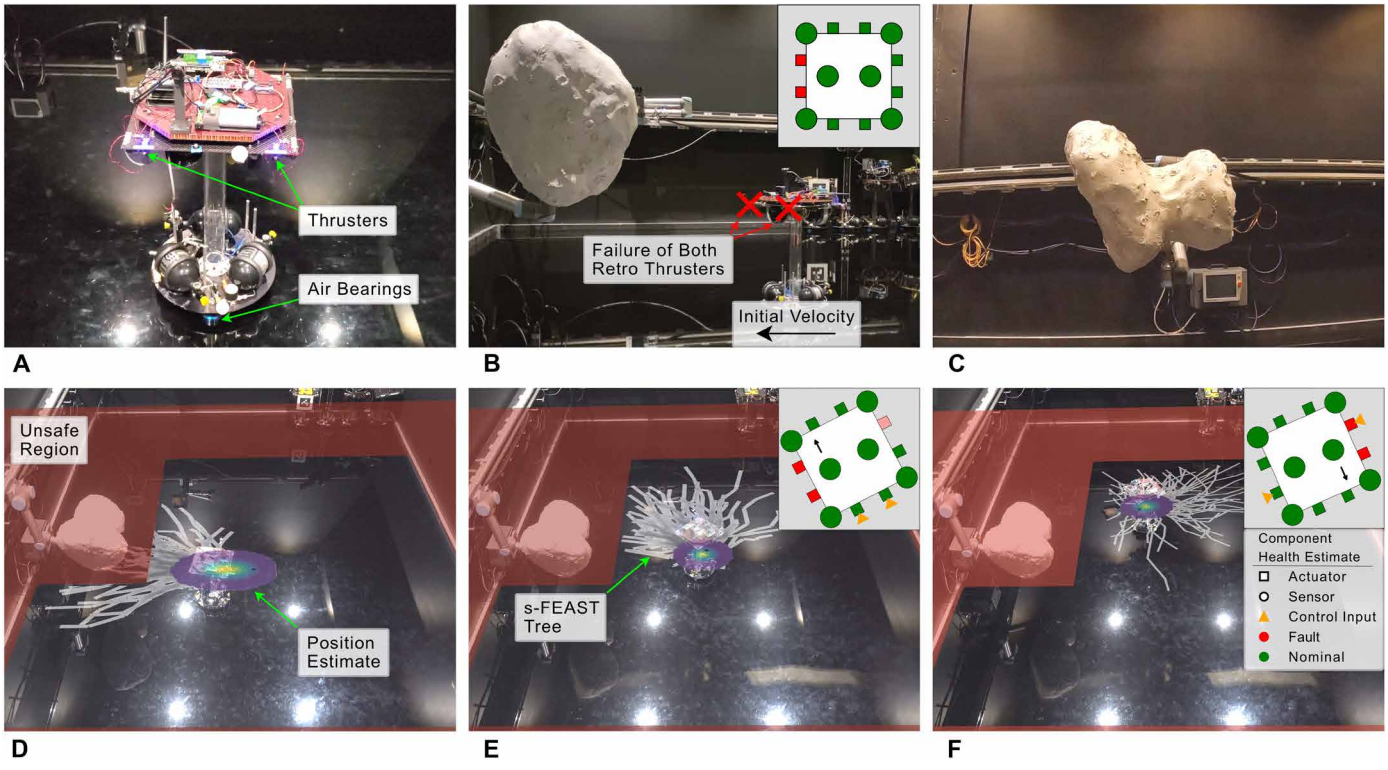
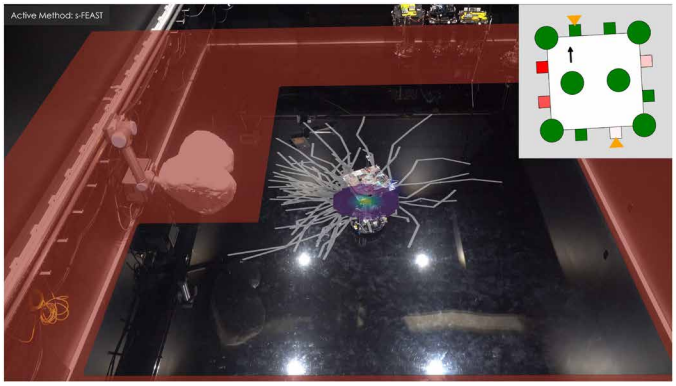


Fig. 1. Safe fault estimation on robotic spacecraft. (**A**) We demonstrated our method on the Caltech Autonomous Robotics and Control Lab's spacecraft simulator, which creates a near-frictionless environment using graphite air bearings to create a cushion of air between the robot and the flat floor. (**B** and **C**) Both of the spacecraft robot's retro thrusters have failed, and it starts on a collision course with the comet. With no evasive actions, a collision will happen within seconds. (**D**) An algorithm that maximizes information gain without considering safety constraints will crash into the comet. (**E**) Our s-FEAST algorithm selects trajectories that have a high likelihood of avoiding the obstacle while also gathering information about the failure. (**F**) The robot can successfully identify the underlying failure and return to a trajectory heading away from the obstacle and boundaries. Still frames for all experiment time steps are provided in the Supplementary Materials, and the full experiment is shown in Movie 1.



Movie 1. s-FEAST enables safe fault estimation on a robotic spacecraft simulator.

(*27*) but can be overly conservative in the open loop and computationally challenging to run online (*28*). Others provide guarantees when the uncertainty in model parameters is energy-bounded (*29*), including in systems with small, bounded nonlinearities (*30*) and linearizations (*31*). These approaches to fault estimation are similar to the field of robotic self-modeling (*32*), which has recently used exploratory actions to distinguish between possible dynamic models of robotic hands (*33*) and to learn visual self-models (*34*).

An alternative framework for passive and active fault estimation is through the lens of partially observable Markov decision processes (POMDPs). POMDPs provide a flexible modeling representation, but they are intractable to solve in general, and solutions are often limited to small problems, offline performance, or inexact methods (*35*). One application to fault estimation considers partial observability for only the first time step (*36*). However, this prevents information gathering, which can be promoted in POMDPs through heuristically defined subgoals (*37*, *38*) or action design (*39*). Recent methods have solved online information-gathering POMDPs by observing part of the state directly (*40*) and augmenting the reward function (*41*); however, these methods do not consider unknown dynamics.

Safety

Although active fault estimation can more rapidly determine the failure of a robot, many systems have operational safety constraints that the information-gathering actions must not violate. Traditional approaches to ensuring safe control include formal methods such as control barrier functions (CBFs) (*42*) and their extensions to discrete-time systems (*43*) with stochastic noise (*44*). Other methods provide probabilistic risk-averse bounds on the robot's safety (*45*). Planning-based approaches to this problem include sequential convex programming (SCP) (*46*, *47*), which can consider complex (*48*) and stochastic constraints (*49*) while achieving robustness and stability guarantees through tracking control (*50*). However, each of these

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

methods assumes a fully observable state to directly evaluate safety constraints. In partially observable settings, these methods must be modified, such as by extending CBFs to operate on the belief of possible states (*51*). Partial observability can also arise from uncertain system dynamics, such as that caused by an unknown failure. When feasible, applying a CBF to all possible dynamic modes can ensure safety, but this approach may be overly conservative (*52*).

As with active fault diagnosis, POMDPs provide an alternative framework for considering safety constraints. One method is through cost terms, which can be solved offline (*53*–*56*) or online via approximate solvers (*57*, *58*) and offline heuristics beyond a subhorizon (*59*). A shared limitation is that these methods constrain only the expected cost, which may not be suitable for risk-averse settings or systems with large state estimation uncertainty, both of which are present in safety-critical active fault estimation. General probabilistic bounds, or chance constraints, allow for bounds on other statistics and can be approximately solved offline (*60*, *61*). Online approaches using heuristic searches exist but lack formal guarantees (*62*).

Constrained POMDPs are also solvable with model-free approaches trained during an offline phase, such as Dreamer V2 (*63*) and latent policy optimization (*64*). However, compared with online methods, the reliance on an offline training phase makes these methods vulnerable to out-of-domain events (*65*). Furthermore, these methods lack theoretical guarantees of optimality convergence and safety assurance, which are especially important for high-cost space missions.

Our work sits at the intersection of these related fields; we seek to combine these separate approaches of fault estimation and constraint satisfaction. We qualitatively summarize this relation in Fig. 2. Like the passive fault estimation and POMDP approaches, our algorithm applies to a wide range of stochastic and uncertain systems. Similar to control-based safety methods and POMDP models, our method, with mild assumptions, provides formal guarantees of constraint satisfaction and general bounds on tail probabilities as opposed to constraints on expectations alone. However, unlike existing active fault diagnosis methods or POMDP solvers, our method can be deployed online in information-gathering problems without requiring heuristics.

Contributions

In our prior work (*66*), we developed an efficient tree search to solve for information-gathering actions in the unconstrained case with binary actuation and sensing failures. In this work, we present a substantial improvement, by generalizing the fault model to a broader class of partial failures and bias attacks, and extend the theoretical and experimental results. Our contributions are summarized as follows: We mathematically formalized the time-critical fault estimation problem subject to state constraints and showed that constrained optimization over the coupled fault mode and physical state uncertainty is challenging for existing methods. We addressed this gap by combining belief-space tree search, marginalized filtering, and concentration inequalities to efficiently maximize an information-gathering objective and satisfy probabilistic state constraints. Last, we present theoretical analysis, real-time hardware experiments, and numerical experiments to validate our claims.

RESULTS

In this section, we present an overview of s-FEAST. We then demonstrate the algorithm's performance on a robotic spacecraft simulator before validating through numerical simulations s-FEAST's ability to perform fault estimation and maintain safety in increasingly challenging scenarios. Last, we look at a qualitative analysis of one of these simulations to identify the behaviors that enable s-FEAST to be successful.

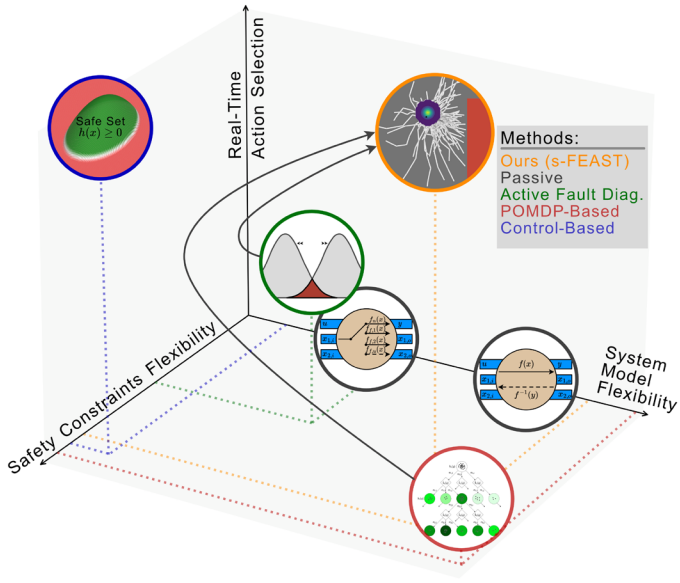


Fig. 2. Related work context. We qualitatively contextualize our work with other relevant approaches applied to fault estimation. Each method is separated by the flexibility of the safety constraints, the flexibility of the system model, and the real-time performance for selecting actions. Passive methods (*8*) do not consider diagnostic actions or safety constraints, and so are restricted to one axis, but represent a broad range of system models. Active fault diagnosis approaches (*21*) compute inputs to determine between possible underlying faults but are often limited to specific systems, uncertainty models, or constraints and may lack real-time guarantees. POMDP methods (*35*, *36*) can model a wide range of systems and constraints but are often computationally intensive to solve, especially in belief-space planning domains. Control-based approaches (*42*, *46*) can quickly find actions to satisfy deterministic safety constraints but traditionally do not consider model uncertainty or information gathering.

s-FEAST overview

An overview of s-FEAST is shown in Fig. 3. Our method is an anytime planner based on a partially observable Monte Carlo tree search (*67*–*69*) and is diagrammed in Fig. 3A. Starting from an initial belief on both the robotic spacecraft's physical and fault states, $\mathbf{b}_0(\mathbf{q})$, actions are selected and simulated forward to a planning horizon. The tree explores actions that both resolve ambiguity in the underlying faults and are predicted to not lead to violations of safety as defined by state constraints. As an anytime algorithm, this tree search refines the simulated futures until interrupted, returning the best action found so far.

Our algorithm has two main innovations. First, we use a marginalized filter to efficiently decompose beliefs into a conditional estimate of the robot's physical state and a total estimate of the failure affecting the robot. This allows for accurate information-gathering rewards within the tree search, and we explain why this is necessary for efficient planning in Discussion. The marginalized filter is visualized in Fig. 3 (B and C). Second, we enforce probabilistic safety constraints with a concentration inequality and provide conservative guarantees of safety for arbitrary belief distributions,

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

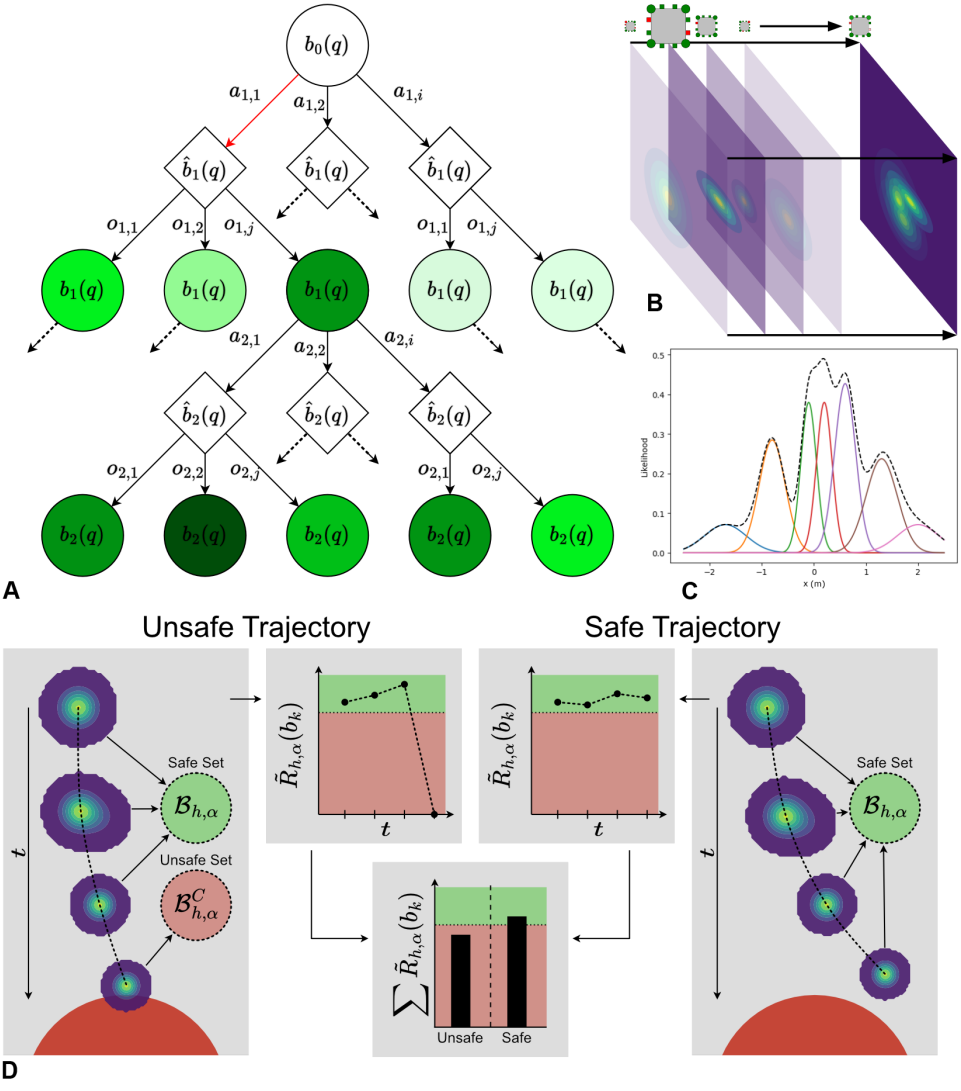


Fig. 3. s-FEAST: Method overview. (**A**) Diagram of the tree search used by s-FEAST. The tree growth is biased toward nodes leading to better rewards (represented here as darker shading). **a** represents actions taken, $\hat{\mathbf{b}}$ represents prior beliefs, **o** represents observations, and **b** represents updated beliefs. (**B**) Illustration of our marginalized filter representing the position of the robotic spacecraft as the sum of estimates conditioned on each possible failure. (**C**) When the physical estimators are Kalman filters, the marginalized filter of a complicated multimodal distribution is a combination of Gaussians. (**D**) The belief at each time step can be classified as in or outside of the set of safe beliefs $(\mathcal{B}_{h,\alpha})$ based on bounding the likelihood of collision with the obstacle (shown as a red semicircle). The reward function used by s-FEAST, $\tilde{R}_{h,\alpha}(\mathbf{b}_k)$, results in any trajectory of safe beliefs having a higher cumulative reward than any trajectory with at least one unsafe belief.

noise processes, and safety constraints. Beliefs that satisfy this inequality are assumed to lie within the set of safe beliefs and receive a bonus reward; otherwise, they are assigned a reward of zero. As a result, for any safe trajectory, the summed reward over the planning horizon is above that of any unsafe trajectory (Fig. 3D). With this construction, the convergence of the tree search to the optimal value also ensures safety. Each of these innovations is presented in detail in Materials and Methods along with the pseudocode of our algorithm.

Robotic spacecraft simulator hardware experiments

We implemented s-FEAST on a Multi-Spacecraft Testbed for Autonomy Research (M-STAR) robot (*70*, *71*) using the Caltech Autonomous Robotics and Control Lab's spacecraft simulator facility, shown in Fig. 1A. The M-STAR robot is actuated using thrusters and uses air bearings to float on a high-precision flat floor, creating a very low-friction environment that simulates spacecraft dynamics. A motion capture system provided position and orientation measurements, and noise was artificially added according to our observation model, as shown in Fig. 4A.

The robot was tasked to diagnose sensing and actuation faults while on a collision course with our model comet (Fig. 1C). The true failure was the loss of both retro thrusters (Fig. 1B), which required the M-STAR robot to reorient before it was able to slow down and stabilize itself. The safety constraints were to avoid the comet obstacle and the walls of the simulator room, shown as the red regions in Fig. 1 (D to F), with a 90% or higher probability. With these settings, s-FEAST was able to successfully identify the true failure state while maintaining safety, validating our approach on hardware. Video of s-FEAST and baselines running on the M-STAR robot hardware is provided in Movie 1, with still frames in Fig. 1. These experiments demonstrated that considering safety or fault estimation alone cannot solve this problem (Fig. 1D), whereas s-FEAST can reliably plan evasive actions under uncertain component failure (Fig. 1, E and F). A complete time series of s-FEAST and the baseline methods are presented in the Supplementary Materials.

To deploy s-FEAST in a real-time setting, we implemented s-FEAST in a receding horizon fashion, meaning that the planner recomputed a policy every time step and applied only the first action to the physical system. Because the dynamics continued to

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

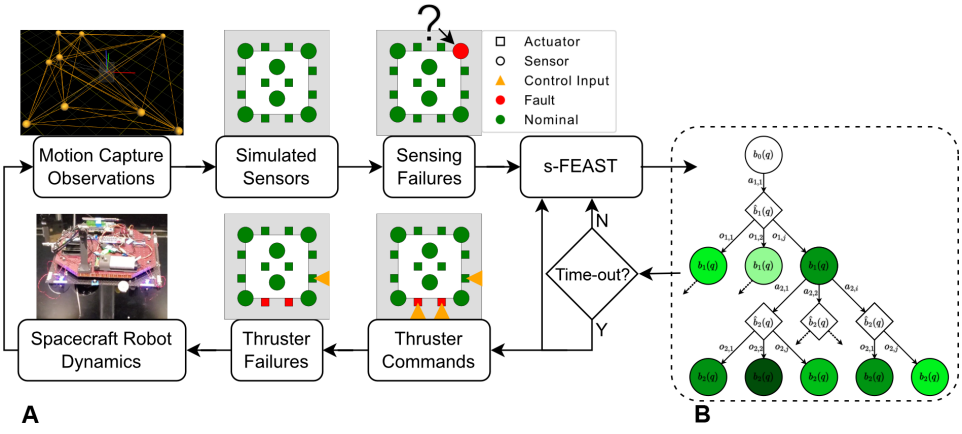


Fig. 4. s-FEAST: Real-time implementation. (**A**) Diagram of our real-time deployment on the robotic spacecraft simulator. The s-FEAST block runs until the specified computation time budget is exceeded, and then the best available action is returned. (**B**) Real-time s-FEAST is run with the first action fixed. This is the currently active action selected at time-out by the previous iteration.

propagate during the planning computation time, the state of the system when the planner began solving $\mathbf{x}_k$ was different from the state when the selected action was taken, $\mathbf{x}_k + \delta_t$, where δ_t was the propagation time. To synchronize these two states, we ran the same s-FEAST algorithm, except we planned the next action to take from the expected result of the current action. The modified tree topology is visualized in Fig. 4B. Instead of specifying the number of simulations to run, we took advantage of s-FEAST's ability to provide an anytime solution by simulating until the computation budget was exhausted and returning the best action. The selected action was then applied on board the robot, and s-FEAST used the current observation to compute the next action to take while the system dynamics propagated. For these experiments, a budget of 0.78 s on a 1.10-GHz, four-core CPU (i5-1035G4) was used, which typically resulted in 85 simulations per time step. We showed in numerical experiments that this is sufficient computation to substantially improve over existing approaches. More details of our implementation and performance are provided in the Supplementary Materials.

Numerical simulations

To validate our algorithm quantitatively, we considered s-FEAST in four safety-critical scenarios against baselines of SCP, discrete CBFs (D-CBF), greedy, and random policies. Each simulation was performed on a three-degree-of-freedom model of the M-STAR robot. We evaluated each algorithm over 1000 trials according to the fraction of safe trials throughout the experiment and the product of a diagnostic reward and success rate. Details on these evaluation metrics, along with system descriptions and additional numerical results, are provided in the Supplementary Materials.

Overview of baselines

We provide a brief overview of the baselines (random, greedy, D-CBF, and SCP) that we compared against in the following simulation results. The selection of baselines was designed such that s-FEAST and these baselines covered a permutation of deterministic versus probabilistic state representations and greedy versus planning algorithmic implementations, with s-FEAST as the probabilistic planning solution. All methods used the same estimator between time steps, and each baseline was solved for the next action to take. Implementation details are provided in the Supplementary Materials.

The first baseline was to randomly select actions uniformly from the admissible control set U. This method performed no optimization for information gathering or safety. The second baseline was a greedy, active approach with a probabilistic state representation. It simulated each action once and selected a safe action with the best immediate reward computed with the marginalized filter but only considered a lookahead horizon of one and did not resample any actions, which made it vulnerable to near-term danger and outlier simulations. Together, the random and greedy baselines served to illustrate the shortcomings of random and one-step planning approaches in identifying the underlying faults when safety constraints must also be satisfied.

The next two baselines were deterministic safe control methods. The D-CBF (*43*) method acts greedily, considering only the safety of the next time step, whereas the SCP (*46*, *47*) method plans safe trajectories over a horizon. These algorithms do not have a probabilistic representation of the state or system model and require a fully observable state. Work has been done to extend both methods to consider stochastic noise via chance constraints for SCP (*49*) and probabilistic safety bounds for D-CBF (*44*), although neither method is compatible with the coupled fault mode and physical state uncertainty considered here. To adapt the methods to our partially observable setting, we used the most likely failure state and corresponding mean position estimate as the assumed system dynamics and initial position and added a buffer to each obstacle. When the system model is accurately known, a controller satisfying the D-CBF condition renders all states in the safe set forward invariant and, therefore, safe. However, this is not guaranteed if the most likely model is inaccurate, and we saw this method fail in our simulations for this reason. These control baselines served to illustrate the limitations of the control-estimation separation principle in safety-critical fault estimation problems.

Overview of scenarios

In each of the following scenarios, we considered a robotic spacecraft initially 10 m from a circular obstacle, which represented some target of interest that the robot was investigating before the failure occurred. For s-FEAST and our safety-aware baselines, we imposed a chance constraint that with 90% or higher probability, the spacecraft must avoid collision and deviate no more than 25 m in any direction from its initial position at each time step. In practice, we saw that this chance constraint enabled s-FEAST to achieve 90% or higher safety through the experiment, given that the robot was near the obstacles or bounds for only a few time steps.

To highlight various sources of difficulty our method addresses, we considered two fault cases in two increasingly difficult initial conditions. Binary faults, where components either worked or were completely failed, illustrated the challenges posed when components fail silently, resulting in ambiguity between fault models. Alternatively, continuous component degradation and biases presented a larger challenge for safety because actuator biases could destabilize a system if unaddressed.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Scenario: Binary fault diagnosis in proximity to an obstacle

In the first scenario, the spacecraft started with no initial velocity, and up to three components completely failed, where the underlying binary failure was randomly selected for each trial. This case was selected to examine how well each policy achieved our desired 90% chance of safety when the spacecraft was not in any immediate danger and to demonstrate how naive information gathering could put the system at risk. The results are summarized in Fig. 5A.

Examining the safety of each method, we saw that the greedy and random baselines markedly underperformed compared with the other methods. This was observed to be in part due to these methods' inability to consider safety beyond the next time step or, in the case of the random baseline, at all. This led to destabilizing actions being selected more often, making future time steps more likely to have no safe action available.

Considering the reward and diagnostic success of each method, the s-FEAST algorithms all outperformed the CBF and SCP baselines, as did the random and greedy baselines. This was because the CBF and SCP baselines did not take any information-gathering actions, or any actions at all, until the system was close to becoming unsafe. Both baselines typically failed to diagnose the underlying failure by the end of the experiment, which led to low diagnosis success rates of 20.8 and 19.5%, respectively. The random and greedy algorithms performed similarly to the s-FEAST algorithms in diagnosing the underlying fault, but at the price of considerably worse safety, with final safety values of 17.4 and 16.9%, respectively.

Scenario: Continuous degradation and bias fault diagnosis in proximity to an obstacle

In this experiment, we considered the same scenario as before, but now components could be partially degraded, giving only a fraction of their nominal output. This could correspond to actuator damage resulting in decreased efficiency or a miscalibrated sensor. Components could also be subject to constant biases, correlating with unexpected behavior such as an actuator being stuck on, sensor offset, or even malicious signal injection. As before, we assumed that the fault was constant for the duration of our diagnosis period. Faults were generated by sampling eight unique biases with five component degradations each, for a total of 40 possible faults as before. The true fault was set to one of these. Details of the fault model and experimental setups are provided in Materials and Methods and the Supplementary Materials, respectively.

The results of this simulation are shown in Fig. 5B. Compared with the previous scenario, we saw similar relative behavior, and all methods had a higher diagnostic reward and a lower safety. The diagnostic reward increased because the faults were no longer silent. Any bias injected a signal into the system, enabling passive identification of these faults. However, the active signal made enforcing safety more challenging because bias acceleration could lead to constraint violations. This trade-off was seen through a drop in safety for all policies. For example, from time steps 3 to 4, the deterministic methods (SCP and CBF) started to decline in safety, whereas the diagnostic reward increased more rapidly than the s-FEAST methods. This trend continued throughout the experiment, with reward increasing but safety dropping.

The ambiguity in component degradation for a given bias provides a likely explanation for this trend. Because neither the SCP nor CBF method considers a belief, information gathering to resolve this ambiguity could not be explicitly performed. This could result in an incorrect assessment of both the safety of the current state and the control authority if actuator faults were not yet detected or resolved. When actions were taken to avoid a collision, they may have occurred too late or with an unexpectedly small effect, leading to safety violations but also yielding more information on the component degradation, giving an increase in diagnosis reward. Last, we note that the decrease in diagnostic reward for the CBF method near the end of the experiment stems from filter divergence. This was due to large control inputs leading to numerical instability in the extended Kalman filter (EKF) without converging to a fault estimate.

Scenario: Collision course under adversarial binary and continuous failures

In the final two scenarios examined, the spacecraft was subjected to the same underlying fault in every trial and initialized on a collision course with an obstacle. We considered an adversarial failure for both our binary and continuous degradation and bias scenarios. In the binary scenario, the two retro thrusters on the spacecraft were completely off, and in the continuous case, the retro thrusters were subject to an 80% degradation, and the forward thrusters were subject to a 10% bias. In both cases, the spacecraft had to first change orientation and then slow down to reliably avoid a collision. Because this behavior required planning over a horizon, we considered these to be adversarial faults for this scenario and chose this scenario to demonstrate s-FEAST's robustness to outlier failures that posed an outsized risk to the system. The spacecraft still started with a uniform prior over 40 possible failures and so had to take actions to reduce the risk of collision before fully identifying the underlying fault.

The results are shown in Fig. 5 (C and D), where we see that all baselines achieved less than 40% safety in the binary case (CBF: 39.7%, SCP: 35.4%, random: 0.4%, greedy: 0.5%) and less than 4% in the continuous case (CBF: 3.6%, SCP: 0.1%, random: 0.5%, greedy: 1.1%) and were outperformed by s-FEAST with even the lowest level of planning. These results suggest that running as few as $N = 80$ simulations can achieve a final safety rate of 62.4% in the binary case and 69.9% in the continuous case. For $N = 200$, the safety rate was 77.8 and 84.9% for the binary and continuous faults, respectively. We used this result to inform our real-time hardware experiments, where the typical planning amount was $N = 85$ simulations per tree because of a tight computational budget. The reward for the hardware algorithm shown in Fig. 1 was similar to that predicted by this simulation experiment.

We again see that with the binary faults, our CBF and SCP baselines failed to gather any information until collision was imminent and only gained diagnostic reward as a result of attempting to remain safe. Similarly, in the continuous case, the baseline methods gained some information immediately as a result of the bias signal but failed to further diagnose until evasive actions were taken, which occurred sooner and at higher speed because of acceleration from the bias input. In the next section, we consider an example to illustrate how s-FEAST succeeded where these baseline methods failed.

Qualitative interpretation of tree data

The tree data structure provides some qualitative interpretability of the inner workings of s-FEAST. In Fig. 6, we see the spacecraft initially on a collision course under the adversarial failure of both retro thrusters. This is the same binary crash course scenario examined in the previous subsection, with a higher initial velocity of 2 m/s to better demonstrate the qualitative behavior of our algorithm. Before identifying the underlying failure, s-FEAST selected actions to adjust the spacecraft's trajectory to the side of the obstacle. This turned out to be a necessary strategy in this scenario because after

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

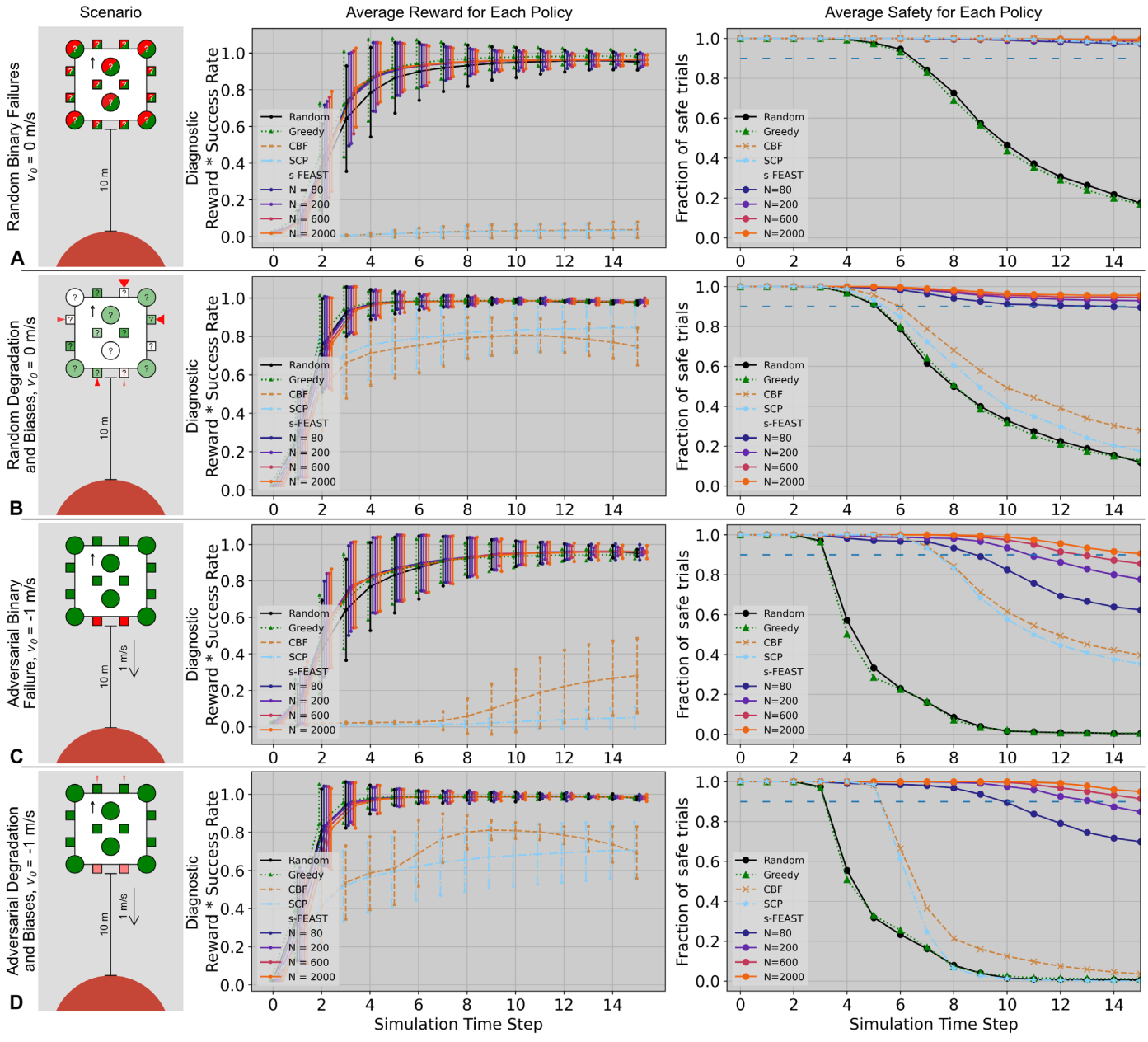


Fig. 5. Validation of s-FEAST. The numerical performance of our algorithm was compared with baselines across several scenarios. (**A**) The robotic spacecraft started 10 m from the obstacle with no initial velocity, subject to random binary failures of up to three components. (**B**) Each component was now randomly subjected to continuous degradation or bias, with nominal components more likely. (**C**) The robotic spacecraft now started with an initial velocity of 1 m/s toward the obstacle, subject to an adversarial binary failure of its two retro thrusters. (**D**) The adversarial failure was now that both retro thrusters degraded by 80%, and both forward thrusters stuck on with a 10% bias. In all experiments, s-FEAST considered 40 possible binary or general faults and started with a uniform prior over all possibilities. In the visualization of the spacecraft component health in the left column, squares represent the thrusters, and circles abstractly represent the position and orientation sensors. Green components are healthy, red are failed, and red actuations represent bias thrust of varying degrees (sensor bias is not visualized). Note that for readability, the data for each time step are artificially spread out horizontally.

the failure was identified in the third time step, it took another seven time steps to reorient and come to a stop. This obstacle avoidance behavior was also seen in our hardware experiments, such as in Fig. 1.

The baseline methods were unable to discover this behavior because both the greedy and CBF policies do not consider the possibility of failure beyond the next time step, and the SCP policy does not take any information-gathering actions and so will be unaware of the failure until it attempts and fails to slow down. Like our simulation results, this suggests that proactive information gathering is essential to avoiding model uncertainty in these safety-critical situations because any unknown component failure can jeopardize the system's performance in unexpected ways.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

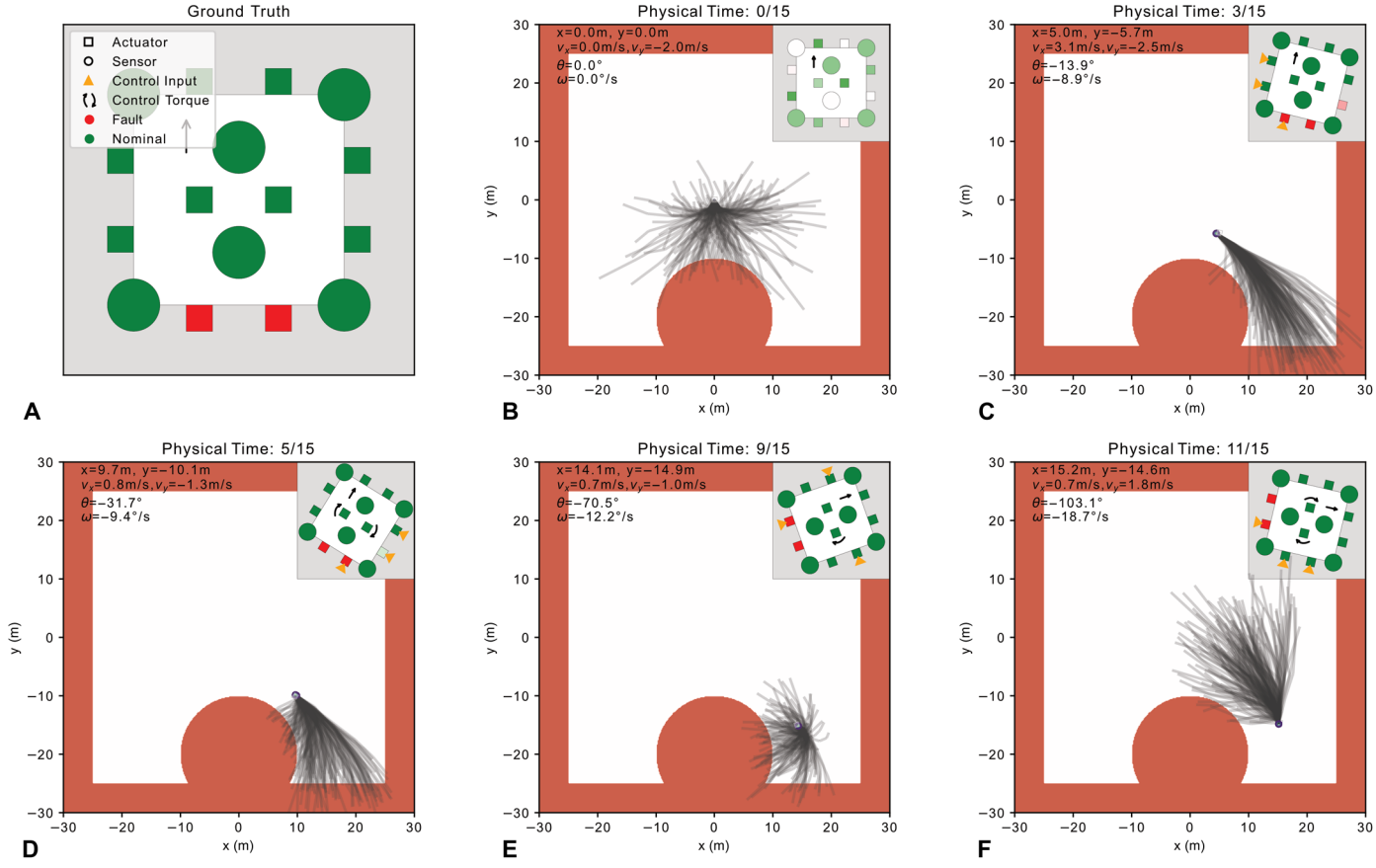


Fig. 6. Qualitative analysis of s-FEAST's collision avoidance under an adversarial fault. (**A**) The two retro thrusters that the spacecraft needs to slow down are dysfunctional. (**B**) The spacecraft starts on a collision course with the obstacle and a uniform belief over randomly selected binary failures of actuators (shown as squares) and sensors (abstracted as circles). (**C**) By the third time step, the spacecraft has mostly identified the underlying failure, indicated by the red components. At this point, it has proactively taken action to avoid the obstacle before the fault was determined. (**D**) After dodging the obstacle, most future trajectories take the spacecraft out of bounds. (**E**) By the ninth time step, the spacecraft had reoriented and started to slow down. (**F**) The spacecraft has reversed course by the eleventh time step and remains safe for the rest of the experiment.

DISCUSSION

We have demonstrated in these experiments safety-aware, real-time, active fault estimation that extends beyond the capability of existing methods, which cannot succeed at all tasks simultaneously. We have shown that in the presence of ambiguous faults and time-critical constraints, proactively taking actions to maintain safety while simultaneously gathering information about the system status is necessary to reliably avoid collision. By combining anytime tree search with efficient filtering and probabilistic chance constraints, s-FEAST achieves both objectives in a computationally tractable fashion, with asymptotic convergence guarantees and interpretable behavior.

Comparison with existing tree-search methods

Previous approaches to solving partially observable planning problems using tree-search methods include the partially observable Monte Carlo planning (POMCP) algorithm (*69*) and its extensions to constrained systems (*57*, *58*). However, in our previous work (*66*), we empirically showed that our marginalized filter approach is necessary for effective planning in information-gathering problems. When the reward is a function of the belief instead of just the classical state and action reward, the convergence guarantees of these existing methods break down. In this work, we formalize this observation.

POMCP consists of two components: first, partially observable upper confidence bound applied to trees (PO-UCT), which assumes access to the state belief for a given history, and second, Monte Carlo updates to propagate the belief within the tree in a particle filter–like manner. For each simulation, a particle is sampled from the initial belief and propagated by running PO-UCT. At each belief node encountered during the simulation, the propagated state is added to the node's particle belief. The resulting belief at each node is a discrete collection of state particles, one for each visit to the node.

POMCP argues that, at a large number of samples, the belief at each node is well approximated such that PO-UCT is solving the equivalent belief Markov decision process and therefore inherits the value convergence of the fully observable UCT (*67*). However, it only establishes this for the PO-UCT algorithm because the theoretical analysis assumes accurate state beliefs for each history and accurate rewards for each node. Neither is initially true in the information-gathering setting, leading to a "burn-in" phase until the belief converges enough that this PO-UCT analysis is valid. Until a repeated particle is added to a node, the information-gathering reward that we introduce in the

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

next section (Eq. 9) is inversely correlated with the number of visits to the node. This results in a breadth-first search, where the UCT strategy of biasing toward areas of high reward no longer succeeds and ultimately leads to random action selection. This is further exacerbated by the exponential scaling of standard particle filters with the number of dimensions (*72*). The difference in tree growth between s-FEAST and POMCP is visualized qualitatively in Fig. 7. In the Supplementary Materials, we provide additional numerical experiments to validate these claims. Pseudocode of the two algorithms is presented in Materials and Methods.

Real-time performance

Our solver is an anytime algorithm, which means that its performance improves given more computation time, but it can be stopped at any point to return the current best solution. In our real-time hardware experiments, the solver evaluation is not fast enough to achieve the highest $N = 2000$ level of planning that we consider in Fig. 5 of the "Numerical simulations" section in Results. Instead, we typically evaluated $N = 85$ trajectories, which is sufficient to successfully identify faults and maintain safety while substantially outperforming baseline methods. This achieves the goal of validating our conceptual algorithmic innovations, although it is possible to further optimize the software and hardware implementation for faster run time and better performance.

To this end, we note that s-FEAST presents two promising opportunities for future performance improvements. First, the marginalized filter that we present in Materials and Methods factors out the physical state estimators, meaning that s-FEAST can leverage any existing estimators that may already be optimized for a system with minimal changes. Second, there exists a growing body of literature on methods to accelerate partially observable planning through parallelization (*73*) and GPU use (*74*). We view these types of optimizations as complementary to s-FEAST's algorithmic innovations, which achieve better scaling through the exploitation of the active sensing problem structure. They also pair well with the anytime nature of our algorithm because increasing simulation speed directly translates into more simulations and improved performance, as seen in Fig. 5 (B and D).

The anytime property is also desirable compared with traditional active fault diagnosis methods, which often require the computation to complete before a solution can be returned and may use approximations to achieve real-time performance (*28*). Instead, we can return the best solution found within any computation budget. We show in the next section that s-FEAST converges asymptotically to the optimal solution, a guarantee not provided by existing chance-constrained anytime methods (*62*).

We expect the ongoing trend of ever-increasing computational power onboard space robotics missions (*75*, *76*) to be enabling for our methodology, especially as payloads are developed for increasingly data-intensive science applications. Because our algorithm only needs to run when a fault is suspected or a safety-critical situation is encountered, we envision a concept of operations where our algorithm is dormant until needed, in which case it takes priority over nonessential payload operations to monopolize computing resources for a short duration before handing back control when normal operations can resume. Our algorithm could also run at scheduled intervals to proactively check for possible faults, resulting in planned payload downtime much like other maintenance operations, including charging windows and course corrections, that mission planners currently consider.

Application of s-FEAST to other information-gathering problems

The active fault estimation approach that we consider is most useful in systems with high functional redundancy that creates ambiguity between possible failures and provides the ability to recover when the fault is identified. Our method will fail if a safety-critical state of the system becomes completely unobservable or uncontrollable. However, these situations will be unrecoverable for our baselines and related work as well.

Last, we note that our approach to belief-space planning and sampling-based safety can be applied to other information-gathering problems where the underlying state has a computable belief transition. For example, we can consider the classic problem of a robot autonomously mapping an unknown environment (*77*) or future robotic planetary exploration missions where actions are taken to scout out areas of potentially high scientific value (*78*). In both cases, gathering information is a key goal, necessitating belief-space planning. In addition, the robot might be subject to additional constraints, ranging from the safety constraints we consider here to battery or time budgets limiting exploration, where the effect of high variance makes constraints on expectation alone limiting.

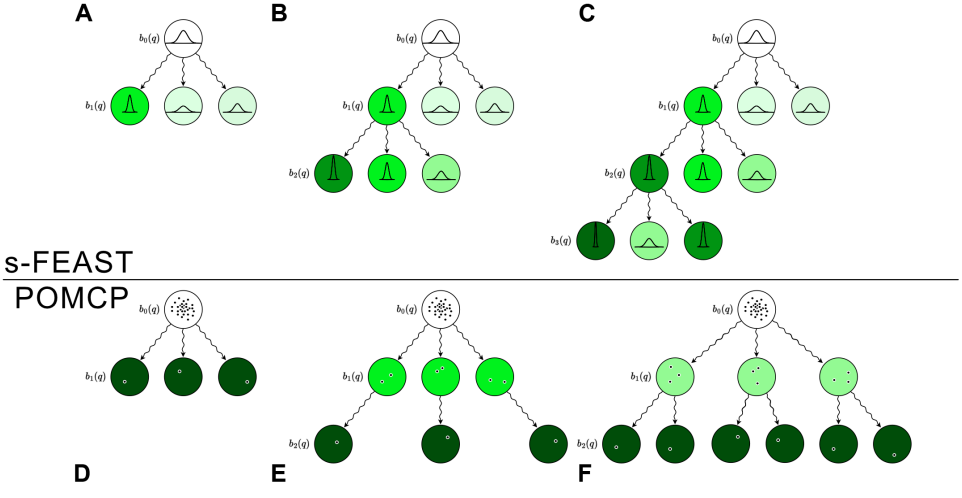


Fig. 7. Marginalized filtering versus particle filtering in information-gathering tree searches. A conceptual comparison of the tree growth of s-FEAST (which uses our marginalized filter) and POMCP (which uses a particle filter). The darker green is a higher estimated reward. (**A**) When s-FEAST expands a node, it performs full belief updates that give accurate reward estimates. (**B**) The tree can then be biased toward areas of higher rewards. (**C**) This enables s-FEAST to efficiently search areas of higher value and plan further ahead. (**D**) In comparison, POMCP performs a particle filter–based search, adding one particle each time it visits a node. When POMCP has only encountered a node once, the estimated reward given by Eq. 9 (presented in Materials and Methods) is maximized. However, this estimate is inaccurate. (**E**) As soon as POMCP revisits a node, there are now two particles, leading to increased belief uncertainty and less reward. (**F**) The result is a breadth-first search.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

s-FEAST provides a flexible method for online active fault estimation in safety-critical settings. Using an efficient marginalized filter, s-FEAST can plan information gathering in settings intractable to existing tree-based solvers. Its modular nature presents opportunities for future performance enhancements of s-FEAST via estimator or tree search optimizations. Further, its anytime property allows s-FEAST to scale performance with the available computational budget. Although s-FEAST is already applicable to a broad range of systems, opportunities for future improvements include broadening the current enumerated sets of possible faults to a bounded subspace of faults. This would allow for scenarios where the faults of concern are not known in advance but the bounds on possible fault behavior are. Another avenue for future work would be to unify s-FEAST with data-driven approaches for the safe exploration of unknown, fully observable dynamics (*79*, *80*) to actively estimate unmodeled actuator or sensor faults.

MATERIALS AND METHODS

First, we formalize our safe active fault estimation setting as a partially observable optimal control problem. Then, we present our algorithm including the marginalization filter, safety condition, and integration into tree search. Last, we present a theoretical analysis of s-FEAST's optimality convergence.

Safe active fault estimation problem formulation

We present our active sensing problem: to plan actions such that the resulting observations converge the belief of the underlying failure to the true fault as quickly as possible while maintaining safety. We define the following modification of general control-affine system dynamics with linear sensing and additive noise processes

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + B(\mathbf{x}_{k-1})\left(\left(\mathbb{1} - \Phi_B\right)\mathbf{u}_k + \Phi_{B,\mathbb{1}}\right) + \mathbf{w}_k \tag{1}$$

$$\mathbf{y}_k = \left(\mathbb{1} - \Phi_C\right)C\mathbf{x}_k + \Phi_{C,\mathbb{1}} + \mathbf{v}_k \tag{2}$$

$$\mathbf{u}_k \in U \subseteq \{0,1\}^m \tag{3}$$

where the k subscript denotes a time index, $\mathbf{x} \in X \subseteq \mathbb{R}^n$ is the physical state, $\mathbf{u} \in U \subseteq \{0,1\}^m$ restricts the control input to a discrete set of binary m-dimensional vectors to represent thruster control, $\mathbf{f}(\mathbf{x}_k)$ is the unforced dynamics, $B(\mathbf{x}_k)$ is the input influence matrix, $\mathbf{y} \in Y \subseteq \mathbb{R}^p$ denotes the measurement, C is the measurement matrix, and the random process and measurement noise sequences $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be mutually independent and independent and identically distributed (i.i.d.). For simplicity, we assume that the noise processes are Gaussian with covariance matrices $\Sigma_{\mathbf{w}}$ and $\Sigma_{\mathbf{v}}$, respectively, but in the development of our safety condition later in this section, we show that our algorithm is not restricted to only Gaussian noise. We also note that the set of control inputs U can be customized for the system of interest, for example, $U \subseteq \{-1,0,1\}^m$, to include bidirectional actuators like reaction wheels.

The system description differs from a control-affine system only in the fault model, Φ_B, Φ_C and $\Phi_{B,\mathbb{1}}$, $\Phi_{C,\mathbb{1}} = \text{diag}\,(\phi_{B/C/B,\mathbb{1}/C,\mathbb{1}})$ representing changes due to degradation or bias attacks in the actuators and sensors

$$\phi_{B_i}\begin{cases} 1 \text{ if } i \text{ actuator is completely failed} \\ 0 \text{ if } i \text{ actuator is nominal} \\ a_i \text{ if } i \text{ actuator is partially degraded} \end{cases}, \phi_B = \left[\phi_{B_1}, \ldots, \phi_{B_m}\right] \tag{4}$$

$$\phi_{B,\mathbb{1}_i}\begin{cases} 1 \text{ if } i \text{ actuator is stuck full on} \\ 0 \text{ if } i \text{ actuator is nominal} \\ a_i \text{ if } i \text{ actuator is partially biased} \end{cases}, \phi_{B,\mathbb{1}} = \left[\phi_{B,\mathbb{1}_1}, \ldots, \phi_{B,\mathbb{1}_m}\right] \tag{5}$$

where $a_i \in (0,1)$. The sensor fault models Φ_C and $\Phi_{C,\mathbb{1}}$ are defined analogously. Both complete failure and partial failure (degradation) cases are considered in this paper. We assume that the fault state does not change with time, so we drop the time subscript k from ϕ terms and define the concatenated vector of all faults

$$\phi_k = \phi_{k-1} = \phi = \left(\phi_B, \phi_{B,\mathbb{1}}, \phi_C, \phi_{C,\mathbb{1}}\right) \in \Phi \subset [0,1]^{2(m+p)}, \quad |\Phi| = N_\Phi < \infty \tag{6}$$

where Φ is the set of N_Φ considered faults that live in the continuous space of $2(m+p)$-dimensional vectors with elements restricted between 0 and 1. We define the augmented state by composing the physical state and fault state, $\mathbf{q} = [\mathbf{x}; \phi]$, where $\mathbf{q} \in Q = X \times \Phi$.

In the presence of state uncertainty, it is common to write the probability distribution of the state as a belief, which can be computed by updating a prior with an observation and control input using Bayesian filtering (*81*)

$$\mathbf{b}_0(\mathbf{q}) = \mathbb{P}(\mathbf{q}_0), \quad \mathbf{b}_k(\mathbf{q}) = \mathbb{P}\left(\mathbf{q}_k \mid \bar{\mathbf{y}}_k, \bar{\mathbf{u}}_k\right) \frac{\mathbb{P}\left(\mathbf{y}_k \mid \mathbf{q}_k\right) \int \mathbb{P}\left(\mathbf{q}_k \mid \mathbf{q}_{k-1}, \mathbf{u}_k\right)\mathbf{b}_{k-1}(\mathbf{q})\mathrm{d}\mathbf{q}_{k-1}}{\mathbb{P}\left(\mathbf{y}_k \mid \bar{\mathbf{y}}_{k-1}, \bar{\mathbf{u}}_k\right)} \tag{7}$$

where $\mathbf{b}_0(x) = \mathbb{P}(\mathbf{q}_0)$ is the prior, and the overbar notation defines a history, such as $\bar{\mathbf{u}}_K = \{\mathbf{u}_1, \ldots, \mathbf{u}_K\}$. The space of all possible beliefs is denoted $\mathcal{B}$. From this joint belief over the augmented state, we define the marginal beliefs over the failure and physical states

$$\mathbf{b}_k(\phi) = \int_{\mathbf{x} \in X} \mathbf{b}_k(\mathbf{x}, \phi)\mathrm{d}\mathbf{x}, \quad \mathbf{b}_k(\mathbf{x}) = \sum_{\phi \in \Phi} \mathbf{b}_k(\mathbf{x}, \phi) \tag{8}$$

The notion of optimality is defined through the reward function, the policy function, and the value function. We consider the following information-gathering reward that maps from belief to rewards between 0 and 1, $R : \mathcal{B} \to [0,1]$

$$R(\mathbf{b}_k) = \sum_{\phi \in \Phi} \left(\mathbf{b}_k(\phi)\right)^2 \tag{9}$$

This reward $R(\mathbf{b}_k)$ corresponds to how confident the current belief is in the underlying fault state. This reward is minimized when the belief on the fault state is uniform and maximized when the belief on the fault state is a delta function. This reward function has previously been proposed as an uncertainty measure (*82*).

The policy function is a stochastic map from belief to action, $\pi: \mathcal{B} \to U$, and the set of all policies is denoted Π. For a finite horizon problem, the value function is the expected return of a policy from an initial belief

$$V^\pi(\mathbf{b}_0) = \mathbb{E}\left[\sum_{k=1}^{K} R(\mathbf{b}_k) \middle| \pi, \mathbf{b}_0\right], \text{ such that (s. t.)} \quad \text{Eqs. 1, 2, and 7, } \mathbf{u}_k = \pi(\mathbf{b}_{k-1})\ \forall k \tag{10}$$

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

where K is the horizon length and the expectation is over the stochastic policy, measurement, and process noise sequences. Control policies are closed-loop solutions, selecting a new action at each time step in response to the belief updated via the new observation.

Probabilistic safety

We use a standard superlevel set notion of probabilistic safety.

Definition 1 (α-*safety*). We consider a set of safety constraints on the physical state, $\{g_i\}$, that must all be simultaneously satisfied for a system to be safe ($g_i(\mathbf{x}) \geq 0, \forall\, i$). We define the safety function h as $h(\mathbf{x}) = \min_i g_i(\mathbf{x})$ and the corresponding set of safe physical states, X_h, as $X_h = \{\mathbf{x} \mid h(\mathbf{x}) \geq 0\}$. We define the set of α-safe beliefs, $\mathcal{B}_{h,\alpha} \subseteq \mathcal{B}$, as the beliefs in which the physical state has a probability of at least α of being safe with respect to h

$$\mathcal{B}_{h,\alpha} = \left\{ \mathbf{b} \in \mathcal{B} \mid \int_X \mathbf{b}(\mathbf{x}) \mathbb{1}_{X_h}(\mathbf{x}) d\mathbf{x} \geq \alpha \right\} \tag{11}$$

where the indicator over the set of safe states is $\mathbb{1}_{Xh}(\mathbf{x}) = 1$ if $\mathbf{x} \in X_h$ and 0 otherwise. Similarly, the indicator over α-safe beliefs is $\mathbb{1}_{\mathcal{B}h,\alpha}(\mathbf{b}_k) = 1$ if $\mathbf{b}_k \in \mathcal{B}_{h,\alpha}$ and 0 otherwise.

Safe active fault estimation

We can now define the safe active fault estimation problem.

Definition 2 (safe active fault estimation). The safe active fault estimation problem for a given safety function, h, and safety threshold, α, is a partially observable optimal control problem subject to the constraint that each belief is α safe

$$\pi^*(\mathbf{b}_0) = \text{argmax}_{\pi \in \Pi}\ V^{\pi}(\mathbf{b}_0) \text{ s. t. } \mathbb{E}\left[\mathbb{1}_{\mathcal{B}_{h,\alpha}}(\mathbf{b}_k) \mid \pi, \mathbf{b}_0 \right] = 1\ \forall k \tag{12}$$

where the expectation is across the stochastic policy, measurement, and process noise sequences. The corresponding optimal value is $V^*(\mathbf{b}_0)$.

s-FEAST algorithm

We present the s-FEAST algorithm and discuss the changes with respect to existing belief-space tree searches. We visualize the growth of our algorithm in Fig. 7 (A to C) and include the pseudocode in Fig. 8.

The tree search of s-FEAST is similar to an existing belief-space Monte Carlo tree search algorithm known as POMCP (*69*). However, our approach is different because POMCP uses state samples to simultaneously estimate the belief and the optimal policy, whereas s-FEAST uses our marginalized filter to immediately estimate the correct belief. Moreover, s-FEAST has a special treatment for the chance-constrained safety condition.

The differences between s-FEAST and POMCP are presented in the pseudocode shown in Fig. 8. In black, we show the original POMCP method, and we highlight the changes for s-FEAST in blue: (i) When a new node is encountered, the exact Bayesian update is computed with our marginalized filter, Eq. 13 discussed in the next subsection; (ii) we use the updated belief to compute exact rewards and generate a value estimate; (iii) we use the exact belief to accurately approximate the safety at each node via our safety condition (theorem 2, presented in the "Safety condition" section in Materials and Methods).

Both algorithms approximate the optimal policy as a tree of nodes. A node is defined as an ordered history H of actions and observations, with a corresponding number of visits $N(H)$, value estimate $\hat{V}(H)$, and belief $\mathbf{b}(H)$. Each simulation is performed from the root node until the depth, d, exceeds the maximum depth K. New states and observations are simulated by the model-based generator G. When a

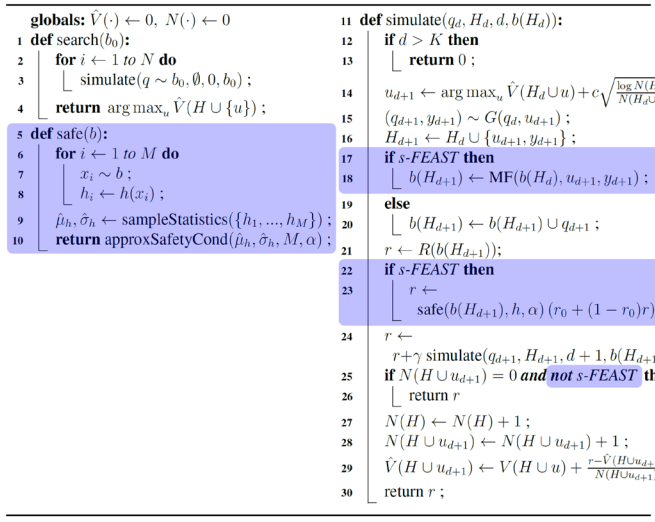


Fig. 8. Algorithm 1: The POMCP and s-FEAST algorithms for belief-space planning. For this pseudocode, we adapt the original POMCP algorithm to our notation, with modifications made to create s-FEAST highlighted in blue (*69*). MF refers to our marginalized filter (the pseudocode of this method is provided in the Supplementary Materials), and approxSafetyCond refers to the approximate safety condition given by Eq. 14 and theorem 2.

previously unexplored history is encountered, the simulation rolls out to the maximum depth by uniformly sampling random actions from the action space U. Because computing the marginalized filter is relatively expensive, s-FEAST saves the nodes from the rollout instead of discarding them. In practice, this tree growth is similar to the fixed depth Monte Carlo tree search proposed by Shah *et al.* (*83*). After completing all N simulations (or timing out), the action with the highest value estimate is returned and applied to the system. The resulting observation is used to update the system's belief, and a new tree is planned from this new root node to select the next action.

When initialized in each experiment, s-FEAST was given knowledge of the system dynamics through Eqs. 1 and 2, including a nominal noise model and the possible failures. A uniform initial probability over all possible failures was assumed, although if prior knowledge of the relative likelihoods of each failure existed, it could be incorporated, and s-FEAST can converge from any initial belief that does not preemptively eliminate the true failure. Overly conservative noise models can also be provided when the true noise level is uncertain. Implementation parameters of s-FEAST used in each experiment are provided in the Supplementary Materials. In the next two sections, we specify the marginalized filter and the safety condition and use them to prove the convergence of s-FEAST to optimal solutions.

Marginalized filter

We present our marginalization filter that is used in the s-FEAST algorithm to accurately estimate the belief. The key observation was that the dynamics (Eq. 1) and measurement (Eq. 2) of the active sensing problem have a structure that we can exploit to efficiently compute the belief update. Whereas jointly computing the belief update for the physical and fault state is intractable, it is possible to condition on a fault and then compute the conditional belief update of the physical state with a standard EKF. Any other nonlinear filtering approach can be used in lieu of EKF. This marginalization approach is

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

similar to the Rao-Blackwellized filter used in FastSLAM (*84*), where the posterior is factored into estimations of each landmark that are conditioned on the robot path, including approaches that actively select trajectories that minimize the uncertainty of a robot's state (*85*). However, instead of estimating the environment in relation to a robot, our method infers the robot's dynamics and measurement model on the basis of its interaction with the environment.

Our approach can be formalized in the following decomposition of the belief

$$\mathbf{b}_k(\mathbf{q}) = \mathbb{P}\left(\left[\mathbf{x}_k, \phi\right] \middle| \bar{\mathbf{y}}_k, \bar{\mathbf{u}}_k\right) = \mathrm{EKF}_\phi\left[\mathbf{y}_k, \mathbf{u}_k, \mathbf{b}_{k-1}(\mathbf{x})\right](\mathbf{x}_k) \frac{\tilde{Z}_\phi\left(\mathbf{y}_k, \mathbf{u}_k, \mathbf{b}_{k-1}(\mathbf{q})\right)\mathbf{b}_{k-1}(\phi)}{\tilde{Z}\left(\mathbf{y}_k, \mathbf{u}_k, \mathbf{b}_{k-1}(\mathbf{q})\right)} \quad (13)$$

where $\mathrm{EKF}_\phi[\mathbf{y}_k, \mathbf{u}_k, \mathbf{b}_{k-1}(\mathbf{x})](\mathbf{x}_k)$ is the posterior distribution on $\mathbf{x}_k$ given by the EKF conditioned on a particular failure state ϕ, and the second term is an unconditional Bayesian update on each possible failure scenario where $\tilde{Z}_\phi$ and $\tilde{Z}$ are conditional and unconditional measurement likelihood functions. We compute $\tilde{Z}_\phi$ as the measurement relative likelihood given by the prediction step of each conditional EKF (before measurement innovation). We then note that $\tilde{Z}$ is a normalization factor and so does not need to be computed explicitly. The resulting filter is visualized in Fig. 3B and resembles methods using multiple Gaussian distributions to represent a complicated distribution as in Fig. 3C. A pseudocode implementation is provided in the Supplementary Materials.

We note that the conditional physical state estimator can be replaced by any estimator parameterized by the failure state, including estimators for non-Gaussian processes. In particular, because the estimation propagation is the primary computational burden in the tree search, our method will benefit substantially from reusing any efficient estimators that may already exist for a system, as opposed to approaches attempting to estimate the joint physical and fault state directly. For example, one strategy to amortize real-time computation cost is to train a neural network–based filter from offline data (*86*). Another strategy is to perform an additional marginalization step on any states of the system that do not depend on the considered faults. This will particularly be useful to scale s-FEAST to high-dimensional systems with isolated faults because only a subset of the estimation needs to be repeated for each considered fault.

Safety condition

To ensure safety, we need to evaluate the indicator function $\mathbb{1}_{\mathcal{B}h,\alpha}(\mathbf{b}_k)$ throughout the tree search. For a general probability distribution, this function is difficult to compute exactly. Instead, for computational efficiency, we use the following conservative sampling-based condition using concentration inequalities

$$\frac{1}{M+1}\left\lfloor \frac{M+1}{M}\left(\frac{\hat{\sigma}_h^2(M-1)}{\hat{\mu}_h^2}+1\right)\right\rfloor \leq 1-\alpha \Rightarrow \mathbf{b} \in \mathcal{B}_{h,\alpha} \quad (14)$$

where $\hat{\mu}_h$ and $\hat{\sigma}_h$ are the sample average and SD resulting from sampling the safety condition $M > 2$ times for a given belief $\mathbf{b}$. In the following subsection, we derive this condition and use it in our convergence analysis of s-FEAST.

Theoretical analysis

The goal of our analysis is to show that the value estimate of s-FEAST (Fig. 8) converges to the solution of the safe active sensing for fault estimation problem, definition 2. The logic is as follows: We reformulate the constrained problem into an equivalent unconstrained problem with a computationally intractable reward function, then we transform the unconstrained problem again using the tractable but conservative safety condition, and lastly, we run s-FEAST on the resulting problem and inherit standard convergence guarantees. The proofs for these results are included in the Supplementary Materials. In the following, we consider constraints in the decision-making problem sense (*53*). In particular, by a constraint or safety constraint, we mean that some states or beliefs are inadmissible. Conversely, an unconstrained problem has no inadmissible states.

First, we reformulate the constrained problem into an equivalent unconstrained problem. This step is necessary because standard Monte Carlo tree search techniques do not explicitly handle constraints (*67*). This argument is similar to that presented in convex optimization (*87*) with log barrier objective reformulations, except that we use an affine objective reformulation that produced empirically higher-performing results for tree search.

The transformed reward function and corresponding value function are defined as follows

$$R_{h,\alpha}(\mathbf{b}_k) = \mathbb{1}_{\mathcal{B}_{h,\alpha}}(\mathbf{b}_k)\left(r_0 + (1-r_0)R(\mathbf{b}_k)\right) \quad (15)$$

$$V^\pi_{h,\alpha}(\mathbf{b}_k) = \mathbb{E}\left[\sum_{k=1}^{K} R_{h,\alpha}(\mathbf{b}_k) \,|\, \pi, \mathbf{b}_0\right] \text{ s.t. Eqs. 1, 2, and 7, } \mathbf{u}_k = \pi(\mathbf{b}_{k-1})\ \forall k \quad (16)$$

where $r_0 = \frac{K}{K+1}$ and the expectation is over the noise processes and stochastic policy. The first result is that the solution of the transformed problem is equivalent to the solution of the original problem (definition 2), formalized with the following theorem.

Theorem 1 (equivalent unbounded reformulation). If a globally optimal solution exists to the constrained safe active fault estimation problem, definition 2, then the solution of the following unconstrained problem with a transformed value function given by Eq. 16 is also a global optimal solution of definition 2

$$\pi^*_{h,\alpha}(\mathbf{b}_0) = \mathrm{argmax}_{\pi\in\Pi}\, V^\pi_{h,\alpha}(\mathbf{b}_0) \quad (17)$$

The proof is presented in the Supplementary Materials. Next, we develop our conservative approximation of $\mathbb{1}_{\mathcal{B}h,\alpha}$. Our approach is based on the following finite sample approximation Chebyshev's inequality, first developed by Saw *et al.* (*88*) and simplified by Kaban (*89*)

$$\mathbb{P}\left(|Z-\hat{\mu}_Z| > \lambda\hat{\sigma}_Z\right) \leq \frac{1}{M+1}\left\lfloor \frac{M+1}{M}\left(\frac{(M-1)}{\lambda^2}+1\right)\right\rfloor \quad (18)$$

where Z is a random variable and λ is a user-specified scalar. The bound is computed by taking M samples that are weakly exchangeable (i.i.d. is sufficient but not necessary) with the random variable to compute the empirical average and SD $\hat{\mu}_Z$ and $\hat{\sigma}_Z$. This bound holds for unknown distributions when $M \geq 2$ and $\lambda \geq 1$. For general random variables, the Chebyshev inequality can be shown to be a tight bound (*89*), making it well-suited to general distributions.

In our setting, the random variable of interest is the safety function applied to a sample from the physical state belief: $h(\mathbf{x})$ where $\mathbf{x} \sim b(\mathbf{x})$. To compute the empirical average ($\hat{\mu}_h$) and SD ($\hat{\sigma}_h$) of this safety value, let $\mathbf{x}_1, \ldots, \mathbf{x}_M$ be i.i.d. samples of $\mathbf{b}(\mathbf{x})$. We then have

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

$$\hat{\mu}_h = \frac{1}{M}\sum_i h(\mathbf{x}_i),\quad \hat{\sigma}_h^2 = \frac{M+1}{M(M-1)}\sum_i \left(h(\mathbf{x}_i)-\hat{\mu}_h\right)^2 \tag{19}$$

Our safety condition then follows directly from applying the finite sample Chebyshev inequality given by Eq. 18 to bound the tail of h that is less than zero (the unsafe tail).

Theorem 2 (conservative sampling bound). For $M > 2$, a belief $\mathbf{b}(\mathbf{x})$, safety function h, $\hat{\mu}_h$, and $\hat{\sigma}_h$ are defined according to Eq. 19 and $\hat{\mu}_h \geq \hat{\sigma}_h$; satisfying the approximate safety condition of Eq. 14 (repeated below for reference) indicates that the belief is conservatively α safe.

$$\frac{1}{M+1}\left\lfloor \frac{M+1}{M}\left(\frac{\hat{\sigma}_h^2(M-1)}{\hat{\mu}_h^2}+1\right)\right\rfloor \leq 1-\alpha \Rightarrow \mathbf{b}\in\mathcal{B}_{h,\alpha}$$

The proof is presented in the Supplementary Materials. In general, the condition presented in theorem 2 is conservative; it is possible for a solution to be α safe and violate the approximate safety condition (Eq. 14). The slackness comes from two sources: (i) the finite-sample approximation of the Chebyshev inequality and (ii) the potential slackness of the Chebyshev bound itself in the infinite sample limit. In our experiments, we found that we can effectively eliminate the first source of slackness with $M = 100$ samples. For this reason, we focus on the second source and the effect of this slackness on the optimal solution.

In the infinite sample limit, $\hat{\mu}_h$ and $\hat{\sigma}_h$ converge to the true statistics μ_h and σ_h, and Eq. 18 becomes the Chebyshev inequality. We formalize the slackness in the Chebyshev bound with the following lemma, which states that the set of beliefs that satisfy the Chebyshev bound is a well-defined subset of the α-safe beliefs.

Lemma 1 (conservative α-safe set). For any belief $\mathbf{b}$ and safety function h with corresponding statistics μ_h and σ_h, there exists a conservatively α-safe set $\tilde{\mathcal{B}}_{h,\alpha} \subseteq \mathcal{B}_{h,\alpha}$, such that the following safety condition is necessary and sufficient for membership

$$\frac{\sigma_h^2}{\mu_h^2} \leq 1-\alpha \Leftrightarrow \mathbf{b}\in\tilde{\mathcal{B}}_{h,\alpha} \tag{20}$$

The proof is presented in the Supplementary Materials. To account for the slackness in our safety condition, we modify our reward and value functions

$$\tilde{R}_{h,\alpha}(\mathbf{b}_k) = \mathbb{1}_{\tilde{\mathcal{B}}_{h,\alpha}}(\mathbf{b}_k)\left(r_0+(1-r_0)R(\mathbf{b}_k)\right) \tag{21}$$

$$\tilde{V}^{\pi}_{h,\alpha}(\mathbf{b}_k) = \mathbb{E}\left[\sum_{k=1}^{K}\tilde{R}_{h,\alpha}(\mathbf{b}_k)\,|\,\pi,\mathbf{b}_0\right] \text{ s.t. Eqs. 1, 2, and 7} \tag{22}$$

We present the final problem reformulation.

Definition 3 (conservative safe active fault estimation). The conservative safe active fault estimation problem is defined as follows

$$\tilde{\pi}^*_{h,\alpha}(\mathbf{b}_0) = \text{argmax}_{\pi\in\Pi}\,\tilde{V}^{\pi}_{h,\alpha}(\mathbf{b}_0) \tag{23}$$

with corresponding optimal value $\tilde{V}^*_{h,\alpha}(\mathbf{b}_0)$.

The desired behavior of this reformulation is that if the solution of the original problem lies in the feasible space of the conservative problem reformulation, then solving the conservative problem will produce the original solution. This property is formalized in the following theorem.

Theorem 3 (problem reformulation equivalence). If an admissible policy, $\pi(\mathbf{b}_0)$, to the safe active fault estimation problem (definition 2) exists and satisfies

$$\mathbb{E}\left[1_{\tilde{\mathcal{B}}_{h,\alpha}}(\mathbf{b}_k)\,|\,\pi,\mathbf{b}_0\right] = 1\ \ \forall k \tag{24}$$

where $\tilde{\mathcal{B}}_{h,\alpha}$ is given by lemma 1, then an optimal policy, $\tilde{\pi}^*_{h,\alpha}(\mathbf{b}_0)$, to the conservative safe active fault estimation problem (definition 3) is a suboptimal solution of definition 2 constrained to $\tilde{\mathcal{B}}_{h,\alpha}$. Furthermore, if an optimal policy, $\pi^*(\mathbf{b}_0)$, to definition 2 exists and satisfies Eq. 24, $\tilde{\pi}^*_{h,\alpha}(\mathbf{b}_0)$ is an optimal solution to definition 2.

The proof is presented in the Supplementary Materials. We can now state the main theorem, which is a direct consequence of reformulating the problem into a search-compatible framework and then applying existing search convergence results: s-FEAST converges to the optimal solutions of the problems given by definitions 2 and 3.

Theorem 4 (optimality of s-FEAST). Let μ denote the policy produced by s-FEAST and $\tilde{\pi}^*_{h,\alpha}(\mathbf{b}_0)$ denote an optimal policy to the conservative safe active fault estimation problem (definition 3). In the limit of $M \to \infty$, the value of these policies converges

$$\lim_{N\to\infty}\left(\tilde{V}^{\mu}_{h,\alpha}(\mathbf{b}_0) - \tilde{V}^*_{h,\alpha}(\mathbf{b}_0)\right) \to 0 \tag{25}$$

with convergence rate $O(\log N/N)$. Furthermore, if an optimal policy, $\pi^*(\mathbf{b}_0)$, to definition 2 exists and satisfies Eq. 24, then $V^{\mu}(\mathbf{b}_0)$ converges to $V^*(\mathbf{b}_0)$.

The proof is presented in the Supplementary Materials. In this section, we have reformulated the safe active fault estimation problem (definition 2) to an unconstrained form by theorem 1. We then used theorem 2 and lemma 1 to define a conservative sampling bound and the corresponding $\tilde{\mathcal{B}}_{h,\alpha}$ to define the conservative safe active fault estimation problem (definition 3). Last, theorem 3 formalizes when the solutions to the two problems are equivalent, and theorem 4 demonstrates the convergence of s-FEAST to optimal solutions for each.

We make some remarks on this result: First, despite applying the existing search results from (*67*) and (*69*), solving problems with belief-dependent objectives and chance constraints for general belief distributions represents an expanded capability enabled by our reformulations. Second, we note that $\tilde{\mathcal{B}}_{h,\alpha}$ is, in general, unknown or computationally intractable. However, we do not need to know $\tilde{\mathcal{B}}_{h,\alpha}$; there just needs to exist an admissible solution in $\tilde{\mathcal{B}}_{h,\alpha}$ for s-FEAST to converge. For the safety constraints of interest we investigated, we observed in our simulations that solutions could come close to violating the constraints relative to the size of the safe state space (such as in Fig. 6D), indicating that $\tilde{\mathcal{B}}_{h,\alpha}$ is tight. Similarly, we observed empirically that $M = 100$ was sufficient for converged safety estimates. Third, it is possible that the optimal solution to the safe active fault estimation problem lies outside $\tilde{\mathcal{B}}_{h,\alpha}$, and in this case, s-FEAST will converge to a suboptimal approximation of the optimal solution. We argue that the only cases where this occurs are when the optimal trajectory takes the spacecraft close to violating a safety constraint, which, while within the bounds of the problem, are the riskiest trajectories. Because adding safety at the cost of some performance is usually desirable in operation, and in regard to the previous observation that the approximation is not overly conservative, we believe that this suboptimal algorithm balances well the competing interests of safety, performance, and computational complexity.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Supplementary Materials

The PDF file includes:
Materials and Methods
Results
Figs. S1 to S6
References (*90*–*95*)

Other Supplementary Material for this manuscript includes the following:
Movie S1

REFERENCES AND NOTES

1. D. C. Schedl, I. Kurmi, O. Bimber, An autonomous drone for search and rescue in forests using airborne optical sectioning. *Sci. Robot.* **6**, eabg1188 (2021).
2. V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. M. McHenry, E. J. Graser, A. L. Rankin, E. R. Thiel, Autonomous robotics is driving Perseverance rover's progress on Mars. *Sci. Robot.* **8**, eadi3099 (2023).
3. T. Ishigooka, S. Honda, H. Takada, Cost-effective redundancy approach for failoperational autonomous driving system, in *2018 IEEE 21st International Symposium on RealTime Distributed Computing (ISORC)* (IEEE, 2018), pp. 107–115.
4. A. Mantooth, C.-M. Zetterling, A. Rusu, Venus calling silicon carbide radio circuits can take the heat needed to phone home from our hellish sister planet. *IEEE Spectrum* **58**, 24–30 (2021).
5. S. A. Jacklin, *Small-Satellite Mission Failure Rates* (Technical Memorandum NASA/TM-2018-220034, NASA Ames Research Center, 2019).
6. Federal Aviation Administration, FAA aerospace forecast: Fiscal years 2019–2039 (Federal Aviation Administration, 2019).
7. M. Osborne, J. Lantair, Z. Shafiq, X. Zhao, V. Robu, D. Flynn, J. Perry, UAS operators safety and reliability survey: Emerging technologies towards the certification of autonomous UAS, in *2019 4th International Conference on System Reliability and Safety (ICSRS)* (IEEE, 2019), pp. 203–212.
8. I. Hwang, S. Kim, Y. Kim, C. E. Seah, A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control. Syst. Technol.* **18**, 636–653 (2010).
9. A. Wander, R. Forstner, *Innovative Fault Detection, Isolation and Recovery Strategies Onboard Spacecraft: State of the Art and Research Challenges* (Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2013).
10. M. Tipaldi, B. Bruenjes, Survey on fault detection, isolation, and recovery strategies in the space domain. *J. Aero. Inf. Sys.* **12**, 235–256 (2015).
11. M. McIntyre, W. Dixon, D. Dawson, I. Walker, Fault detection and identification for robot manipulators, in *IEEE International Conference on Robotics and Automation, 2004 Proceedings* (IEEE, 2004), pp. 4981–4986.
12. M. Visinsky, J. Cavallaro, I. Walker, Robotic fault detection and fault tolerance: A survey. *Reliab. Eng. Syst. Saf.* **46**, 139–158 (1994).
13. R. Mattone, A. De Luca, Relaxed fault detection and isolation: An application to a nonlinear case study. *Automatica* **42**, 109–116 (2006).
14. F. Baghernezhad, K. Khorasani, Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots. *Neurocomputing* **171**, 335–346 (2016).
15. K. Tidriri, N. Chatti, S. Verron, T. Tiplica, Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Ann. Rev. Control* **42**, 63–81 (2016).
16. E. Khalastchi, M. Kalech, On fault detection and diagnosis in robotic systems. *ACM Comput. Surv.* **51**, 1–24 (2018).
17. A. Marino, F. Pierri, F. Arrichiello, Distributed fault detection isolation and accommodation for homogeneous networked discrete-time linear systems. *IEEE Trans. Automat. Contr.* **62**, 4840–4847 (2017).
18. S. Hayden, A. Sweet, S. Christa, Livingstone model-based diagnosis of Earth Observing One, in *AIAA 1st Intelligent Systems Technical Conference* (AIAA, 2004), p. 6225.
19. R. Mackey, A. Nikora, C. Altenbuchner, R. Bocchino, M. Sievers, L. Fesq, K. O. Kolcio, M. J. Litke, M. Prather, On-board model based fault diagnosis for cubesat attitude control subsystem: Flight data results, in *2021 IEEE Aerospace Conference* (IEEE, 2021), pp. 1–17.
20. M. Šimandl, I. Punčochář, Active fault detection and control: Unified formulation and optimal design. *Automatica* **45**, 2052–2059 (2009).
21. T. A. N. Heirung, A. Mesbah, Input design for active fault diagnosis. *Annu. Rev. Control* **47**, 35–50 (2019).
22. M. Sampath, S. Lafortune, D. Teneketzis, Active diagnosis of discrete-event systems. *IEEE Trans. Automat. Contr.* **43**, 908–929 (1998).
23. E. Chanthery, Y. Pencolé, N. Bussac, An ao*-like algorithm implementation for active diagnosis, in *10th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS* (JAXA, 2010), pp. 75–76.
24. E. Chanthery, L. Travé-Massuyès, Y. Pencolé, R. De Ferluc, B. Dellandréa, Applying active diagnosis to space systems by on-board control procedures. *IEEE Trans. Aerosp. Electron. Syst.* **55**, 2568–2580 (2019).
25. L. Blackmore, B. Williams, Finite horizon control design for optimal discrimination between several models, in *Proceedings of the 45th IEEE Conference on Decision and Control* (IEEE, 2006), pp. 1147–1152.
26. J. A. Paulson, T. A. N. Heirung, R. D. Braatz, A. Mesbah, Closed-loop active fault diagnosis for stochastic linear systems, in *2018 Annual American Control Conference (ACC)* (IEEE, 2018), pp. 735–741.
27. J. K. Scott, R. Findeisen, R. D. Braatz, D. M. Raimondo, Input design for guaranteed fault diagnosis using zonotopes. *Automatica* **50**, 1580–1589 (2014).
28. D. M. Raimondo, G. R. Marseglia, R. D. Braatz, J. K. Scott, Closed-loop input design for guaranteed fault diagnosis using set-valued observers. *Automatica* **74**, 107–117 (2016).
29. S. L. Campbell, R. Nikoukhah, *Auxiliary Signal Design for Failure Detection* (Princeton Univ. Press, 2015).
30. S. L. Campbell, K. G. Horton, R. Nikoukhah, Auxiliary signal design for rapid multi-model identification using optimization. *Automatica* **38**, 1313–1325 (2002).
31. S. L. Campbell, K. J. Drake, I. Andjelkovic, K. Sweetingham, D. Choe, Model based failure detection using test signals from linearizations: A case study, in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control* (IEEE, 2006), pp. 2659–2664.
32. J. Bongard, V. Zykov, H. Lipson, Resilient machines through continuous self-modeling. *Science* **314**, 1118–1121 (2006).
33. K. Hang, W. G. Bircher, A. S. Morgan, A. M. Dollar, Manipulation for self-identification, and self-identification for better manipulation. *Sci. Robot.* **6**, eabe1321 (2021).
34. B. Chen, R. Kwiatkowski, C. Vondrick, H. Lipson, Fully body visual self-modeling of robot morphologies. *Sci. Robot.* **7**, eabn1944 (2022).
35. G. Shani, J. Pineau, R. Kaplow, A survey of point-based pomdp solvers. *Auton. Agent. Multi Agent Syst.* **27**, 1–51 (2013).
36. K. A. Svendsen, M. L. Seto, Partially observable Markov decision processes for fault management in autonomous underwater vehicles, in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* (IEEE, 2020), pp. 1–7.
37. R. He, E. Brunskill, N. Roy, PUMA: Planning under uncertainty with macro-actions, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, 2010), vol. 24, pp. 1089–1095.
38. H. Ma, J. Pineau, Information gathering and reward exploitation of subgoals for POMDPS, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, 2015), vol. 29, pp. 3320–3326.
39. M. T. Spaan, T. S. Veiga, P. U. Lima, Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Auton. Agent. Multi Agent Syst.* **29**, 1157–1185 (2015).
40. L. Dressel, M. Kochenderfer, Efficient decision-theoretic target localization, in *Proceedings of the International Conference on Automated Planning and Scheduling* (AAAI, 2017), vol. 27, pp. 70–78.
41. J. C. Saborío, J. Hertzberg, Towards domain-independent biases for action selection in robotic task-planning under uncertainty, in *International Conference on Agents and Artificial Intelligence (ICAART 2018)* (Science and Technology Publications, 2018), pp. 85–93.
42. A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada, Control barrier functions: Theory and applications, in *2019 18th European Control Conference (ECC)* (IEEE, 2019), pp. 3420–3431.
43. A. Agrawal, K. Sreenath, Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation, in *Proceedings of Robotics: Science and Systems*, N. Amato, S. Srinivasa, N. Ayanian, S. Kuindersma, Eds. (RSS, 2017), pp. 1–10.
44. R. Cosner, P. Culbertson, A. Taylor, A. Ames, Robust safety under stochastic uncertainty with discrete-time control barrier functions, in *Proceedings of Robotics: Science and Systems*, K. Bekris, K. Hauser, S. Herbert, J. Yu, Eds. (RSS, 2023), pp. 1–11.
45. A. Dixit, M. Ahmadi, J. W. Burdick, Risk-sensitive motion planning using entropic value-at-risk, in *2021 European Control Conference (ECC)* (IEEE, 2021), pp. 1726–1732.
46. D. Morgan, S.-J. Chung, F. Y. Hadaegh, Model predictive control of swarms of spacecraft using sequential convex programming. *J. Guid. Control Dyn.* **37**, 1725–1740 (2014).
47. D. Morgan, G. P. Subramanian, S.-J. Chung, F. Y. Hadaegh, Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. *Int. J. Robot. Res.* **35**, 1261–1285 (2016).
48. R. H. Byrd, J. C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming. *Math. Program.* **89**, 149–185 (2000).
49. Y. K. Nakka, S.-J. Chung, Trajectory optimization of chance-constrained nonlinear stochastic systems for motion planning under uncertainty. *IEEE Trans. Robot.* **39**, 203–222 (2023).
50. H. Tsukamoto, B. Riviere, C. Choi, A. Rahmani, S.-J. Chung, CaRT: Certified safety and robust tracking in learning-based motion planning for multi-agent systems, in *2023 62nd IEEE Conference on Decision and Control (CDC)* (IEEE, 2023), pp. 2910–2917.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

51. M. Vahs, C. Pek, J. Tumova, Belief control barrier functions for risk-aware control. *IEEE Robot. Autom. Lett.* **8**, 8565–8572 (2023).
52. Z. Laouar, R. Mazouz, T. Becker, Q. H. Ho, Z. N. Sunberg, Feasibility-guided safety-aware model predictive control for jump Markov linear systems. arXiv:2310.14116 [eess.SY] (2023).
53. E. Altman, *Constrained Markov Decision Processes* (Routledge, ed. 1, 2021).
54. P. Poupart, A. Malhotra, P. Pei, K.-E. Kim, B. Goh, M. Bowling, Approximate linear programming for constrained partially observable Markov decision processes, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, 2015), vol. 29, pp. 3342–3348.
55. D. Kim, J. Lee, K.-E. Kim, P. Poupart, Point-based value iteration for constrained POMDPs, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (IJCAI, 2011), vol. 11, pp. 1968–1974.
56. K. H. Wray K. Czuprynski, Scalable gradient ascent for controllers in constrained POMDPs, in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022), pp. 9085–9091.
57. J. Lee, G.-H. Kim, P. Poupart, K.-E. Kim, Monte-Carlo tree search for constrained POMDPs, in *Advances in Neural Information Processing Systems (NeurIPS 2018)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, Eds. (Curran Associates, Inc., 2018), vol. 31, pp. 7934–7943.
58. A. Jamgochian, A. Corso, M. J. Kochenderfer, Online planning for constrained POMDPs with continuous spaces through dual ascent, in *Proceedings of the International Conference on Automated Planning and Scheduling* (AAAI, 2023), vol. 33, pp. 198–202.
59. A. Undurti, J. P. How, An online algorithm for constrained POMDPs, in *2010 IEEE International Conference on Robotics and Automation* (ICRA, 2010), pp. 3966–3973.
60. M. Khonji, A. Jasour, B. C. Williams, Approximability of constant-horizon constrained POMDP, in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19),* S. Kraus, Ed. (IJCAI, 2019), pp. 5583–5590.
61. P. Santana, S. Thiébaux, B. Williams, Rao*: An algorithm for chance-constrained POMDP's, in *Proceedings of the AAAI Conference on Artificial Intelligence* (PKP, 2016), vol. 30, pp. 3308–3314.
62. S. Hong, S. U. Lee, X. Huang, M. Khonji, R. Alyassi, B. C. Williams, An anytime algorithm for chance constrained stochastic shortest path problems and its application to aircraft routing, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 475–481.
63. D. Hafner, T. Lillicrap, M. Norouzi, J. Ba, Mastering Atari with discrete world models. arXiv:2010.02193 [cs.LG] (2020).
64. R. Rafailov, T. Yu, A. Rajeswaran, C. Finn, Offline reinforcement learning from images with latent space models, in *Proceedings of the 3rd Conference on Learning for Dynamics and Control,* vol. 144 of *Proceedings of Machine Learning Research* (MLResearchPress, 2021), pp. 1154–1168.
65. D. Ghosh, A. Ajay, P. Agrawal, S. Levine, Offline RL policies should be trained to be adaptive, in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162 of *Proceedings of Machine Learning Research* (MLResearchPress, 2022), pp. 7513–7530.
66. J. Ragan, B. Riviere, S.-J. Chung, Bayesian active sensing for fault estimation with belief space tree search, in *AIAA Scitech 2023 Forum* (ARC, 2023), p. 0874.
67. L. Kocsis, C. Szepesvari, Bandit based monte-carlo planning, in *Machine Learning: ECML 2006. Lecture Notes in Computer Science*, J. Fürnkranz, T. Scheffer, M. Spiliopoulou, Eds. (Springer, 2006), vol. 4212, pp. 282–293.
68. C. Browne, E. J. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. P. Liebana, S. Samothrakis, S. Colton, A survey of Monte Carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games* **4**, 1–43 (2012).
69. D. Silver, J. Veness, Monte-Carlo planning in large POMDPs. *Adv. Neural Inf. Process. Syst.* **23**, 2164–2172 (2010).
70. Y. K. Nakka, R. C. Foust, E. S. Lupu, D. B. Elliott, I. S. Crowell, S.-J. Chung, F. Y. Hadaegh, A six degree-of-freedom spacecraft dynamics simulator for formation control research, in *AAS/AIAA Astrodynamics Specialist Conference* (AIAA, 2018), pp. 1–20.
71. R. Foust, E. Lupu, Y. Nakka, S.-J. Chung, F. Hadaegh, Autonomous in-orbit satellite assembly from a modular heterogeneous swarm. *Acta Astronaut.* **169**, 191–205 (2020).
72. S. C. Surace, A. Kutschireiter, J.-P. Pfister, How to avoid the curse of dimensionality: Scalability of particle filters with and without importance weights. *SIAM Rev.* **61**, 79–91 (2019).
73. S. Basu, S. Rajesh, K. Zheng, S. Tellex, R. I. Bahar, Parallelizing POMCP to solve complex POMDPs, in *Robotics: Science and Systems (RSS) Workshop on Software Tools for Real-time Optimal Control* (RSS, 2021).
74. P. Cai, Y. Luo, D. Hsu, W. S. Lee, Hyp-despot: A hybrid parallel algorithm for online planning under uncertainty. *Int. J. Rob. Res.* **40**, 558–573 (2021).
75. B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, D. Zhu, Mars helicopter technology demonstrator, in *2018 AIAA Atmospheric Flight Mechanics Conference* (ARC, 2018), p. 0023.
76. W. S. Slater, N. P. Tiwari, T. M. Lovelly, J. K. Mee, Total ionizing dose radiation testing of NVIDIA Jetson Nano GPUs, in *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, 2020), pp. 1–3.
77. G. Oriolo, G. Ulivi, M. Vendittelli, Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Trans. Syst. Man Cybern. Part B* **28**, 316–333 (1998).
78. P. Arm, G. Waibel, J. Preisig, T. Tuna, R. Zhou, V. Bickel, G. Ligeza, T. Miki, F. Kehl, H. Kolvenbach, M. Hutter, Scientific exploration of challenging planetary analog environments with a team of legged robots. *Sci. Robot.* **8**, eade9548 (2023).
79. T. Lew, A. Sharma, J. Harrison, A. Bylard, M. Pavone, Safe active dynamics learning and control: A sequential exploration–exploitation framework. *IEEE Trans. Robot.* **38**, 2888–2907 (2022).
80. T. Koller, F. Berkenkamp, M. Turchetta, A. Krause, Learning-based model predictive control for safe exploratio, in *2018 IEEE Conference on Decision and Control (CDC)* (IEEE, 2018), pp. 6059–6066.
81. S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, Intelligent Robotics and Autonomous Agents Series (MIT Press, 2005).
82. G. Sussmann, Uncertainty relation: From inequality to equality. *Z. Naturforsch. A* **52**, 49–52 (1997).
83. D. Shah, Q. Xie, Z. Xu, Non-asymptotic analysis of Monte Carlo tree search, in *SIGMETRICS '20: Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems* (ACM, 2020), pp. 31–32.
84. M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, in *AAAI-02: Eighteenth National Conference on Artificial Intelligence* (AAAI, 2002), pp. 593–598.
85. M. Kontitsis, E. A. Theodorou, E. Todorov, Multi-robot active SLAM with relative entropy optimization, in *2013 American Control Conference* (IEEE 2013), pp. 2757–2764.
86. J. Marino, M. Cvitkovic, Y. Yue, A general method for amortizing variational filtering, in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, Eds. (Curran Associates, Inc., 2018), pp. 7868–7879.
87. S. P. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge Univ. Press, 2004).
88. J. G. Saw, M. C. Yang, T. C. Mo, Chebyshev inequality with estimated mean and variance. *Am. Stat.* **38**, 130–132 (1984).
89. A. Kaban, Non-parametric detection of meaningless distances in high dimensional data. *Stat. Comput.* **22**, 375–385 (2012).
90. M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application* (MIT Press, 2015).
91. R. Munos, A. W. Moore, Variable resolution discretization in optimal control. *Mach. Learn.* **49**, 291–323 (2002).
92. Y. Bar-Shalom, X. R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation* (Wiley-Interscience, 2001).
93. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nat. Methods* **17**, 261–272 (2020).
94. J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: Composable transformations of Python+NumPy programs, GitHub (2018); http://github.com/google/jax.
95. M. Lofqvist, J. Cano, Accelerating deep learning applications in space. arXiv:2007.11089 [cs.CV] (2020).

Acknowledgments: We thank A. Rahmani, Y. Nakka, C. Choi, J. Brader, and B. Bycroft for technical input. We would also like to thank T. Hagander and S. Lupu for contributions to hardware development, D. Mitchell and J. Cho for help in setting up and recording hardware experiments, and S. Lupu and C. Choi for photography assistance. **Funding:** This work was supported by the Aerospace Corporation, Jet Propulsion Laboratory (JPL), Defense Advanced Research Projects Agency (DARPA), Learning Introspective Control (LINC) program, and Technology Innovation Institute (TII). **Author contributions:** Conceptualization: J.R., B.R., F.Y.H., and S.-J.C. Data curation: J.R. Formal analysis: J.R., B.R., and S.-J.C. Funding acquisition: F.Y.H. and S.-J.C. Investigation: J.R. Methodology: J.R., B.R., and S.-J.C. Project administration: F.Y.H. and S.-J.C. Software: J.R. Supervision: F.Y.H. and S.-J.C. Validation: J.R. Visualization: J.R. Writing—original draft: J.R. and B.R. Writing—review and editing: J.R., B.R., F.Y.H., and S.-J.C. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data and code used to produce the plots presented here and in the Supplementary Materials are available on the online repository Dryad: https://doi.org/10.5061/dryad.xgxd254r1. Our code is also available at https://github.com/treyra/s-FEAST.

Submitted 11 December 2023
Accepted 31 July 2024
Published 28 August 2024
10.1126/scirobotics.adn4722

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Science Robotics

Online tree-based planning for active spacecraft fault estimation and collision avoidance

James Ragan, Benjamin Riviere, Fred Y. Hadaegh, and Soon-Jo Chung

Sci. Robot. **9** (93), eadn4722. DOI: 10.1126/scirobotics.adn4722

Editor's summary
As autonomous systems are increasingly deployed in various real-world scenarios, the ability to safely accomplish their desired tasks is dependent on the ability to detect system failures and subsequently take the appropriate action without human intervention. Ragan *et al*. have developed a method—safe fault estimation via active sensing tree search—for diagnosing system faults capable of planning and acting to ensure safe robot operations. They demonstrated that the method could actively perform fault estimation on a robotic spacecraft simulator on a collision course with a model comet when thrusters are dysfunctional and showed its potential to keep the robot on a safe path. —Amos Matsiko

View the article online
https://www.science.org/doi/10.1126/scirobotics.adn4722
Permissions
https://www.science.org/help/reprints-and-permissions

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Use of this article is subject to the Terms of service

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.

Copyright © 2024 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works