# Using Metareasoning to Maintain and Restore Safety for Reliably Autonomy

**Justin Svegliato**, **Connor Basich**, **Sandhya Saisubramanian** and **Shlomo Zilberstein**
University of Massachusetts Amherst
{jsvegliato, cbasich, saisubramanian, shlomo}@cs.umass.edu

## Abstract

While developers carefully specify the high-level decision-making models in autonomous systems, it is infeasible for these models to ensure safety across every scenario that can be encountered during operation. We therefore propose a *safety metareasoning system* that mitigates the *severity* of the system's safety concerns while reducing the *interference* to the system's task: the system executes in parallel a *task process* that completes the task and *safety processes* that each address a safety concern, arbitrating with a *conflict resolver*. This paper offers a definition of a safety metareasoning system, an evaluation rating generation algorithm for a safety process, a conflict resolution algorithm for a conflict resolver, an application of our approach to planetary rover exploration, and a demonstration that our approach is effective in simulation.

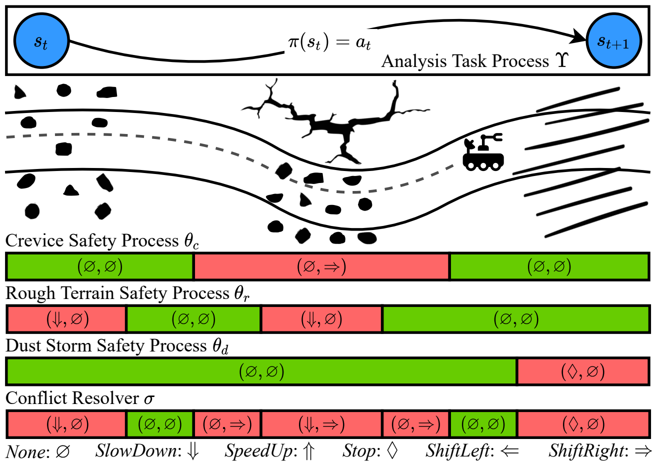

$s_t$
$\pi(s_t) = a_t$
$s_{t+1}$
Analysis Task Process $\Upsilon$
Crevice Safety Process $\theta_c$
$(\varnothing, \varnothing)$
$(\varnothing, \Rightarrow)$
$(\varnothing, \varnothing)$
Rough Terrain Safety Process $\theta_r$
$(\Downarrow, \varnothing)$
$(\varnothing, \varnothing)$
$(\Downarrow, \varnothing)$
$(\varnothing, \varnothing)$
Dust Storm Safety Process $\theta_d$
$(\varnothing, \varnothing)$
$(\Diamond, \varnothing)$
Conflict Resolver $\sigma$
$(\Downarrow, \varnothing)$
$(\varnothing, \varnothing)$
$(\varnothing, \Rightarrow)$
$(\Downarrow, \Rightarrow)$
$(\varnothing, \Rightarrow)$
$(\varnothing, \varnothing)$
$(\Diamond, \varnothing)$
*None*: $\varnothing$ *SlowDown*: $\Downarrow$ *SpeedUp*: $\Uparrow$ *Stop*: $\Diamond$ *ShiftLeft*: $\Leftarrow$ *ShiftRight*: $\Rightarrow$

Figure 1: A planetary rover, arbitrating with the conflict resolver, runs in parallel an analysis task process designed to analyze different points of interests within a region of a planet and safety processes each designed to address either crevices, rough terrain, or dust storms. The *left* and *right* values of each tuple denote a *wheel rotation rate* and *steering* parameter with *green* and *red* denoting safe and unsafe operation.

## 1 Introduction

While planning and robotics experts carefully design, build, and test the models used by autonomous systems for high-level decision making, it is infeasible for these models to ensure safety across every scenario within the domain of operation [1]. This is due to the challenge of specifying accurate, comprehensive decision-making models given the complexity of the state space or the action space, a lack of information about the environment, or a misunderstanding of the competence of the autonomous system [2]. For example, a courier robot could use a decision-making model with features for safely interacting with different types of doors but not for navigating a crosswalk, increasing the risk of endangering people, damaging property, or breaking the courier robot [3]. Therefore, as autonomous systems grow in sophistication [4], it is critical for researchers and practitioners to give them the ability to maintain and restore safe operation.

A simple approach to giving an autonomous system the ability to maintain and restore safety is to use an accurate, comprehensive decision-making model with every feature needed to cover every scenario within the domain of operation. Such an approach, however, suffers from two main problems in the real world [1]. First, it is simply infeasible to build this model for complex environments. Second, even if it were feasible, it would be infeasible to solve this model with exact or even approximate methods in real-time environments. In short, such an approach cannot be used to maintain and restore safe operation in an autonomous system.

There are several areas of safety for decision making in autonomous systems that have seen recent attention [4]. First, methods avoid *negative side effects* that cause a system to interfere with its environment (e.g., by adding an extra term to its objective function [5] or modifying its decision-making model based on human feedback [6]). Next, methods mitigate *reward hacking* that cause a system to game its reward function (e.g., by applying ethical constraints to its behavior [7; 8; 9] or treating its reward function as an observation of its true objective function [10]). Finally, methods handle *distributional change* that cause a system to perform poorly in a new environment differing from its original environment (e.g., by detecting anomalies using Monte Carlo methods based on a particle filter [11; 12; 13] or multiple model estimation based on neural networks [14; 15]). Ideally, any approach that enables autonomous systems to make decisions that maintain and restore some notion of safety should address these areas.

We therefore propose a natural approach to safety in autonomous systems that uses metareasoning as shown in Figure 1. A *safety metareasoning system* executes in parallel (1) a *task process* designed to complete a task and (2) *safety processes* each designed to address a safety concern, arbitrating with (3) a *conflict resolver*. First, the task process *performs* actions that can be tuned by the parameters recommended by the safety processes. Second, the safety processes *recommend* evaluation ratings over the parameters that can tune the actions performed by the task process. Third, the conflict resolver *selects* the parameters to tune the actions performed by the task process given the evaluation ratings over the parameters recommended by the safety processes. In short, for each time step, the task process performs an action, the safety process recommends evaluation ratings over the parameters of that action, and the conflict resolver selects the parameter to tune that action. Our experiments on a planetary rover exploration domain show that the system mitigates the severity of safety concerns while reducing interference to the task.

Our main contributions in this paper are: (1) a definition of a safety metareasoning system, (2) an evaluation rating generation algorithm for a safety process, (3) a conflict resolution algorithm for a conflict resolver, (4) an application of our approach to planetary rover exploration, and (5) a demonstration that our approach is effective in simulation.

## 2 Background

A *Markov decision process* (MDP) is a decision-making model for reasoning in fully observable, stochastic environments [16; 17]. An MDP can be described as a tuple $\langle S, A, T, R\rangle$. The set of states is $S$. The set of actions is $A$. The transition function $T : S \times A \times S \to [0, 1]$ represents the probability of reaching a state $s' \in S$ after performing an action $a \in A$ in a state $s \in S$. The reward function $R : S \times A \to \mathbb{R}$ represents the expected immediate reward of performing an action $a \in A$ in a state $s \in S$. A solution to an MDP is a policy $\pi : S \to A$ indicating that an action $\pi(s) \in A$ should be performed in a state $s \in S$. A policy $\pi$ induces a value function $V^\pi : S \to \mathbb{R}$ representing the expected discounted cumulative reward $V^\pi(s) \in \mathbb{R}$ for each state $s \in S$ given a discount factor $0 \leq \gamma < 1$. An optimal policy $\pi^*$ maximizes the expected discounted cumulative reward for each state $s \in S$ by meeting the Bellman optimality equation $V^*(s) = \max_{a\in A} \left[R(s, a) + \gamma \sum_{s'\in S} T(s, a, s')V^*(s')\right]$.

*Value iteration* is a common technique for finding an optimal policy $\pi^*$ of an MDP $\langle S, A, T, R\rangle$. It begins with an optimal 0-horizon value function $V_0^*$. It then builds an optimal $(t+1)$-horizon value function $V_{t+1}^*$ from an optimal $t$-horizon value function $V_t^*$ by using the Bellman backup operator, $V_{t+1}^* = \max_{a\in A} \left[R(s, a) + \gamma \sum_{s'\in S} T(s, a, s')V_t^*(s')\right]$, for each time step $t$ until the condition $\|V_{t+1} - V_t\|_\infty < \epsilon\frac{(1-\gamma)}{\gamma}$ is satisfied for a given convergence limit $\epsilon$. As each consecutive optimal finite horizon value function $V_{t+1}^*$ is constructed from the current optimal finite horizon value function $V_t^*$, it approaches the optimal infinite horizon value function $V^*$ as the time step $t$ approaches the infinite horizon $h = \infty$:

$$\lim_{t\to\infty} \max_{s\in S} |V^*(s) - V_t^*(s)| = 0.$$

Note the Bellman optimality equation can be applied to any state in any order as long as no state is starved indefinitely.

## 3 Metareasoning for Safety

We now propose a novel metareasoning approach, a *safety metareasoning system*, that runs in parallel (1) a *task process* that completes a task and (2) *safety processes* that each address a safety concern, arbitrating with (3) a *conflict resolver*.

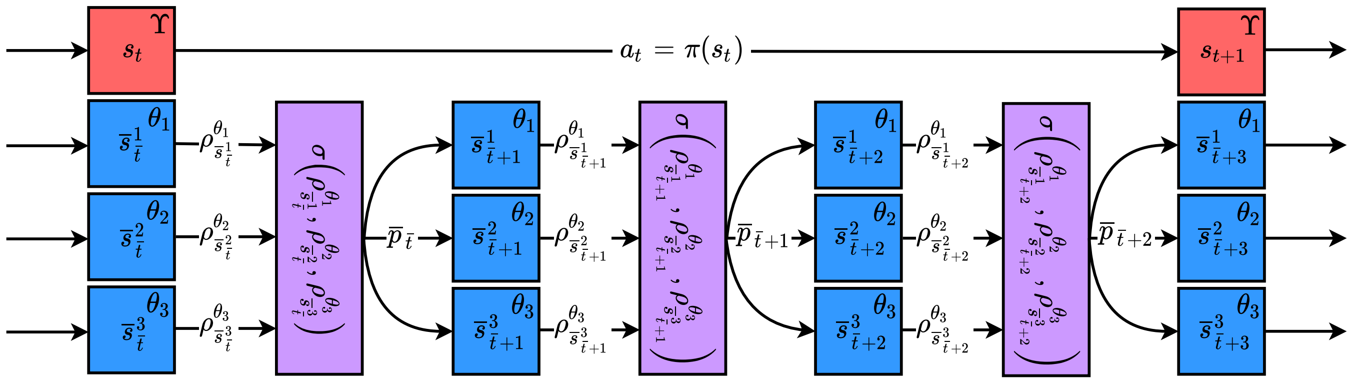
First, the *task process* performs an action in its current state following its policy where the action can be tuned by a parameter recommended by the safety processes. For instance, a planetary rover could execute a task process for analyzing soil and rock samples at different points of interest within a region of a planet. The task process representation must reflect the properties of the task. The task process in this paper is represented by an MDP, a decision process for tasks with full observability, because it is a standard general-purpose sequential decision-making model used throughout planning and robotics. However, it is also possible to use other types of decision processes for tasks with partial observability or start and goal states. We define a task process below.

**Definition 1.** *A **task process**, $\Upsilon = \langle S, A, T, R\rangle$, performs an action $a_t = \pi(s_t) \in A$ in a state $s_t \in S$ at a time step $t \leq H$ following a policy $\pi$ to complete a given task.*

Second, a *safety process* recommends an evaluation rating over a set of parameters in its current state where each parameter can tune the action performed by the task process. For example, a planetary rover could execute three safety processes for crevices inhibiting its wheels, dust storms destroying its sensitive sensors, and rough terrain damaging its wheels. The safety process representation is a tuple with several important attributes: a set of states that describe the safety concern, a set of parameters that can tune the action performed by the task process, a transition function that reflects the dynamics between a state, a parameter, and a successor state, a severity function that reflects the severity of the safety concern in a state, and an interference function that indicates the interference of a parameter on the action performed by the task process. We define a safety process below.

**Definition 2.** *A **safety process**, $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \phi, \psi\rangle \in \Theta$, recommends an evaluation rating $\rho^\theta_{\bar{s}_{\bar{t}}}$ over a set of parameters $\bar{P}$ in a state $\bar{s}_t \in \bar{S}$ at a time step $\bar{t} \leq \bar{H}$ to address a given safety concern that can be encountered during operation.*

- *$\bar{S}$ is a set of states that describe the safety concern.*
- *$\bar{P} = \bar{P}_1 \times \bar{P}_2 \times \cdots \times \bar{P}_N$ is a set of parameters such that each parameter factor $\bar{P}_i$ can tune the action $a \in A$ performed by the task process $\Upsilon$ in some way with a $\varnothing \in \bar{P}_i$ parameter that indicates no tuning.*
- *$\bar{T} : \bar{S} \times \bar{P} \times \bar{S} \to [0, 1]$ is a transition function that represents the probability of reaching a state $\bar{s}' \in \bar{S}$ after using a parameter $\bar{p} \in \bar{P}$ in a state $\bar{s} \in \bar{S}$.*
- *$\phi : \bar{S} \to \{1, 2, \ldots, L\}$ is a severity function that represents the severity of the safety concern in a state $\bar{s} \in \bar{S}$ such that 1 is the lowest level and $L$ is the highest level where a severity level $1 \leq \ell \leq L$ is strictly preferred to a severity level $1 \leq \ell + 1 \leq L$.*
- *$\psi : \bar{P} \to \mathbb{R}^+$ is an interference function that represents the interference of a parameter $\bar{p} \in \bar{P}$ on the action $a \in A$ performed by the task process $\Upsilon$.*



$\Upsilon$
$s_t$
$a_t = \pi(s_t)$
$s_{t+1}$
$\theta_1$
$\theta_2$
$\theta_3$
$\bar{s}^1_{\bar{t}}$
$\bar{s}^2_{\bar{t}}$
$\bar{s}^3_{\bar{t}}$
$\rho^{\theta_1}_{\bar{s}^1_{\bar{t}}}$
$\rho^{\theta_2}_{\bar{s}^2_{\bar{t}}}$
$\rho^{\theta_3}_{\bar{s}^3_{\bar{t}}}$
$\sigma\left(\rho^{\theta_1}_{\bar{s}^1_{\bar{t}}}, \rho^{\theta_2}_{\bar{s}^2_{\bar{t}}}, \rho^{\theta_3}_{\bar{s}^3_{\bar{t}}}\right)$
$\bar{p}_{\bar{t}}$
$\bar{s}^1_{\bar{t}+1}$
$\bar{s}^2_{\bar{t}+1}$
$\bar{s}^3_{\bar{t}+1}$
$\rho^{\theta_1}_{\bar{s}^1_{\bar{t}+1}}$
$\rho^{\theta_2}_{\bar{s}^2_{\bar{t}+1}}$
$\rho^{\theta_3}_{\bar{s}^3_{\bar{t}+1}}$
$\sigma\left(\rho^{\theta_1}_{\bar{s}^1_{\bar{t}+1}}, \rho^{\theta_2}_{\bar{s}^2_{\bar{t}+1}}, \rho^{\theta_3}_{\bar{s}^3_{\bar{t}+1}}\right)$
$\bar{p}_{\bar{t}+1}$
$\bar{s}^1_{\bar{t}+2}$
$\bar{s}^2_{\bar{t}+2}$
$\bar{s}^3_{\bar{t}+2}$
$\rho^{\theta_1}_{\bar{s}^1_{\bar{t}+2}}$
$\rho^{\theta_2}_{\bar{s}^2_{\bar{t}+2}}$
$\rho^{\theta_3}_{\bar{s}^3_{\bar{t}+2}}$
$\sigma\left(\rho^{\theta_1}_{\bar{s}^1_{\bar{t}+2}}, \rho^{\theta_2}_{\bar{s}^2_{\bar{t}+2}}, \rho^{\theta_3}_{\bar{s}^3_{\bar{t}+2}}\right)$
$\bar{p}_{\bar{t}+2}$
$\bar{s}^1_{\bar{t}+3}$
$\bar{s}^2_{\bar{t}+3}$
$\bar{s}^3_{\bar{t}+3}$

Figure 2: A safety metareasoning system with the task process (*red*), safety processes (*blue*), and conflict resolver (*purple*).

Note that the safety processes $\theta \in \Theta$ of a safety metareasoning system share the same exact set of parameters $\bar{P}$.

It is important that a safety process recommends *an evaluation rating over a set of parameters* instead of just *a parameter*. This enables a safety metareasoning system to select a parameter that mitigates the severity of the safety concerns while reducing the interference to the task across all safety processes. An evaluation rating includes multiple values for each parameter: the *expected discounted frequency* that the severity level is incurred by the safety process for each severity level and the *expected discounted cumulative interference* incurred by the safety process when using a parameter in a state. We define an evaluation rating below.

**Definition 3.** *An* ***evaluation rating****, $\rho^{\theta}_{\bar{s}_{\bar{t}}}$, over a set of parameters $\bar{P}$ in a state $\bar{s}_{\bar{t}} \in \bar{S}$ at a time step $\bar{t} \in \bar{H}$ recommended by a safety process $\theta \in \Theta$ is represented by the following $|P| \times |L+1|$ matrix:*

$$\rho^{\theta}_{\bar{s}_{\bar{t}}} = \begin{bmatrix} \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_1}[1] & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_1}[2] & \dots & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_1}[L] & \Psi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_1} \\ \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_2}[1] & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_2}[2] & \dots & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_2}[L] & \Psi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_N}[1] & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_N}[2] & \dots & \Phi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_N}[L] & \Psi^{\theta}_{\bar{s}_{\bar{t}},\bar{p}_N} \end{bmatrix}.$$

*Observe that $\Phi^{\theta}_{\bar{s},\bar{p}}[\ell]$ denotes the expected discounted frequency that a severity level $1 \leq \ell \leq L$ is incurred by a safety process $\theta \in \Theta$ and $\Psi^{\theta}_{\bar{s},\bar{p}}$ denotes the expected discounted cumulative interference incurred by a safety process $\theta \in \Theta$ when using a parameter $\bar{p} \in \bar{P}$ in a state $\bar{s} \in \bar{S}$.*

Third, the *conflict resolver* selects the parameter to tune the action performed by the task process given the evaluation ratings recommended by the safety processes. For instance, a planetary rover could use a conflict resolver to select an optimal parameter that minimizes the severity of the safety concern of a safety process designed for crevices and but also a safety process designed for dust storms. However, in many scenarios, there will either be no safety process or only one safety process that recommends a significant evaluation rating, which indicates that there is no need for conflict resolution. The conflict resolver representation is a function that maps the evaluation ratings recommended by the safety processes to a parameter. We define a conflict resolver below.

**Definition 4.** *A* ***conflict resolver****, $\sigma : \rho^{\theta_1}_{\bar{s}^1} \times \rho^{\theta_2}_{\bar{s}^2} \times \cdots \times \rho^{\theta_n}_{\bar{s}^n} \rightarrow P$, selects the parameter $\bar{p} \in \bar{P}$ to tune the action performed by the task process given the evaluation ratings $\rho^{\theta_i}_{\bar{s}^i}$ recommended by the safety processes $\theta_i \in \Theta$ for conflict resolution.*

Therefore, by putting these attributes together, we offer a formal description of a safety metareasoning system below.

**Definition 5.** *A* ***safety metareasoning system****, $\langle \Upsilon, \Theta, \sigma \rangle$, executes in parallel a task process $\Upsilon$ designed to complete a task and a set of safety processes $\Theta$ each designed to address a safety concern, arbitrating with a conflict resolver $\sigma$.*

The goal of a safety metareasoning system is to select a parameter at each time step that optimizes a lexicographic objective function. First, in the order of decreasing severity level, the system *minimizes* the *maximum* expected discounted frequency that each severity level is incurred by all safety processes (that is, the maximum expected *consequences* across the safety concerns). Second, the system *minimizes* the *maximum* expected discounted cumulative interference incurred by all safety processes (that is, the maximum expected *efficiency impact* on the task). Formally, given the evaluation ratings $\rho^{\theta_i}_{\bar{s}^i_{\bar{t}}}$ recommended by the safety processes $\theta_i \in \Theta$ in a state $\bar{s}^i_{\bar{t}} \in \bar{S}$ at a time step $\bar{t} \in \bar{H}$, the lexicographic objective function is below.

$$\min_{\bar{p} \in \bar{P}} \left[ \max_{\theta_i \in \Theta} \left[ \Phi^{\theta_i}_{\bar{s}^i_{\bar{t}},\bar{p}}[L] \succ \Phi^{\theta_i}_{\bar{s}^i_{\bar{t}},\bar{p}}[L-1] \succ \cdots \succ \Phi^{\theta_i}_{\bar{s}^i_{\bar{t}},\bar{p}}[1] \succ \Psi^{\theta_i}_{\bar{s}^i_{\bar{t}},\bar{p}} \right] \right]$$

Note that the lexicographic preference operator $\succ$ denotes that the left term is always optimized prior to the right term.

Figure 2 summarizes a safety metareasoning system. There is a task transition from the state $s_t \in S$ at time step $t \in H$ to the successor state $s_{t+1} \in S$ at time step $(t+1) \in H$ given the action $a_t = \pi(s_t) \in A$ with respect to the task process $\Upsilon$. During this task transition, there are many safety transitions from the state $\bar{s}^i_{\bar{t}} \in \bar{S}$ at time step $\bar{t} \in \bar{H}$ to the successor state $\bar{s}^i_{\bar{t}+1} \in \bar{S}$ at time step $(\bar{t}+1) \in \bar{H}$ for each safety process $\theta_i \in \Theta$. During the safety transition, each safety process $\theta_i \in \Theta$ recommends an evaluation rating $\rho^{\theta_i}_{\bar{s}^i_{\bar{t}}}$ over the set of parameter $\bar{P}$ to the conflict resolver $\sigma$. The conflict resolver $\sigma$ then selects the optimal parameter $\bar{p}_{\bar{t}} \in \bar{P}$ that optimizes the lexicographic objective function. Once the optimal parameter

**Algorithm 1:** The evaluation rating generation algorithm used by the safety metareasoning system between the *blue* and *purple* objects in Figure 2.

```
Input: A safety process $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \phi, \psi \rangle$
Output: An evaluation rating matrix $\rho^\theta$
1  $\Psi^\theta := \bar{S} \times \bar{P} \to \mathbb{R}^+$
2  for $\ell \to 1, 2, \ldots, L$ do
3      $\Phi^\theta[\ell] := \bar{S} \times \bar{P} \to \mathbb{R}^+$
4  $\Lambda \leftarrow \emptyset$
5  for $\ell \to L, L-1, \ldots, 1$ do
6      $\kappa(\bar{s}, \cdot) := [\phi(\bar{s}) = \ell]$
7      $\Phi^\theta[\ell] \leftarrow$ ModifiedValueIteration$(\theta, \kappa, \Lambda)$
8      for $\bar{s}$ in $\bar{S}$ do
9          $\alpha \leftarrow \min_{\bar{p} \in \bar{P}} \Phi^\theta_{\bar{s},\bar{p}}[\ell]$
10         for $\bar{p}$ in $\bar{P}$ do
11             if $\Phi^\theta_{\bar{s},\bar{p}}[\ell] > \alpha$ then
12                 $\Lambda \leftarrow \Lambda \cup (\bar{s}, \bar{p})$
13 $\kappa(\cdot, \bar{p}) := \psi(\bar{p})$
14 $\Psi^\theta \leftarrow$ ModifiedValueIteration$(\theta, \kappa, \Lambda)$
15 return $\rho^\theta = \left[\Phi^\theta[1], \Phi^\theta[2], \ldots, \Phi^\theta[L], \Psi^\theta\right]$
```

**Algorithm 2:** The modified version of value iteration used by the evaluation rating generation algorithm.

```
Input: A safety process $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \cdot, \cdot \rangle$, a cost
       function $\kappa$, and a set of eliminated
       state-parameter pairs $\Lambda$
Output: An optimal parameter-value function $Q^*$
Require: A discount factor $\gamma$ and a threshold $\epsilon$
1  for $\bar{s}$ in $\bar{S}$ do
2      $V_0(\bar{s}) \leftarrow 0$
3  while $t \to 0, 1, \ldots, \infty$ do
4      for $(\bar{s}, \bar{p})$ in $\bar{S} \times \bar{P}$ do
5          if $(\bar{s}, \bar{p})$ in $\Lambda$ then
6              $Q_{t+1}(\bar{s}, \bar{p}) \leftarrow \infty$
7              continue
8          immediate $\leftarrow \kappa(\bar{s}, \bar{p})$
9          future $\leftarrow \sum_{\bar{s}' \in \bar{S}} \bar{T}(\bar{s}, \bar{p}, \bar{s}') V_t(\bar{s}')$
10         $Q_{t+1}(\bar{s}, \bar{p}) \leftarrow$ immediate $+ \gamma \cdot$ future
11     for $\bar{s}$ in $\bar{S}$ do
12         $V_{t+1}(\bar{s}) \leftarrow \min_{\bar{p} \in \bar{P}} Q_{t+1}(\bar{s}, \bar{p})$
13     if $\|V_{t+1} - V_t\|_\infty < \epsilon \frac{(1-\gamma)}{\gamma}$ then
14         return $Q^* = Q_{t+1}$
```

$\bar{p}_{\bar{t}} \in \bar{P}$ is selected by the conflict resolver $\sigma$, the action $a_t = \pi(s_t) \in A$ of the task process $\Upsilon$ is tuned in some way that reflects that parameter. The task process $\Upsilon$ operates on the time steps $t \in H$ while each safety process $\theta_i \in \Theta$ operates on the time steps $\bar{t} \in \bar{H}$ because the parameter of each action of the task process may be adjusted continually.

In the following sections, we describe two important algorithms required by a safety metareasoning system. First, in Figure 2, the *evaluation rating generation algorithm* computes the evaluation ratings $\rho^{\theta_i}_{\bar{s}^i_{\bar{t}}}$ recommended by a safety process $\theta_i \in \Theta$ between the *blue* and *purple* objects. Second, in Figure 2, the *conflict resolution algorithm* implements the conflict resolver $\sigma$ that selects the optimal parameter $\bar{p}_{\bar{t}} \in \bar{P}$ across all safety processes between the *purple* and *blue* objects. We now describe both algorithms in detail below.

## 3.1 Evaluation Rating Generation

The evaluation rating generation algorithm computes the evaluation ratings recommended by a safety process. This involves computing multiple values from the prior section: the *expected discounted frequency* that each severity level is incurred using a parameter in a state and the *expected discounted cumulative interference* incurred by a safety process using a parameter in a state. The basis for computing these values involves—for each severity level in the order of decreasing severity level and the interference—performing a modified version of value iteration that operates on a space of states and parameters in place of a space of states and actions. As discussed later, this modified version of value iteration must also ignore any state-parameter pair within a set of eliminated state-parameter pairs that is taken in as input.

At a high level, the evaluation rating generation algorithm executes the modified version of value iteration for each severity level in the order of decreasing severity level followed by the interference to align with the lexicographic objective function. That is, given this modified version of value iteration, the evaluation rating algorithm performs two steps. First, it performs the modified version of value iteration for each severity level in the order of decreasing severity level. Second, it performs the modified version of value iteration solely for the interference. Across every execution of modified value iteration, the algorithm maintains a set of eliminated state-parameter pairs that grows with each execution. This set of eliminated state-parameter pairs is used by each execution of modified value iteration to discard any parameter worse than the optimal parameter from the prior execution of modified value iteration. The result is an evaluation rating for each state of a given safety process. This is done *offline* for every safety process before the operation of the system.

Algorithm 1 describes evaluation rating generation. The expected discounted cumulative interference matrix and the expected discounted frequency matrix for each severity level are defined (Lines 1-3). A set of eliminated state-parameter pairs is initialized (Line 4). A loop from the highest to the lowest severity level begins (Line 5). A cost function yielding a unit cost if the severity level of a given state is equal to the current severity level is defined (Line 6). Modified value iteration computes the expected discounted frequency matrix for the current severity level (Line 7). The set of eliminated state-parameter pairs is updated with every state-parameter pair worse than the corresponding optimal state-parameter pair for the current severity level (Lines 8-12). A cost function yielding the interference of a given parameter is defined (Line 13). Modified value iteration computes the expected discounted cumulative interference matrix (Line 14). An evaluation rating matrix is returned (Line 15).

Algorithm 2 describes modified value iteration. The value function at the initial time step is initialized (Lines 1-2). A loop from the initial time step to an infinite horizon begins (Line 3). A sweep over every state-parameter pair begins (Line 4). If the current state-parameter pair is eliminated, its parameter-value function at the next time step is set to infinity and the sweep continues to the next state-parameter pair (Lines 5-7). Otherwise, its parameter-value function at the next step is updated by using the Bellman backup operator (Lines 8-10). The state-value function at the next step is updated by minimizing over the parameters given the parameter-value function (Lines 11-12). The optimal parameter-value function is returned upon convergence (Lines 13-14).

We sketch a proof for the correctness and the time complexity of the evaluation rating generation algorithm below.

**Proposition 1** (Correctness)**.** *Evaluation rating generation computes an evaluation rating matrix $\rho^\theta$ with the expected discounted frequency matrix $\Phi^\theta[\ell]$ for each severity level $1 \leq \ell \leq L$ and the expected discounted cumulative interference matrix $\Psi^\theta$ that aligns with the lexicographic objective function for a given safety process $\theta \in \Theta$.*

*Proof Sketch.* Observe that, for modified value iteration, there are $L$ executions for each severity level $1 \leq \ell \leq L$ and 1 execution for the interference. Given the convergence guarantees of standard value iteration without a set of eliminated state-parameter pairs, these executions are guaranteed to compute the expected discounted frequency for each severity level and the expected discounted cumulative interference for the interference but may not align with the lexicographic objective function. However, with a set of eliminated state-parameter pairs, these values align with lexicographic objective function. Hence, the algorithm is correct. □

**Proposition 2** (Time Complexity)**.** *Evaluation rating generation has a time complexity of $O((L+1)|\bar{S}|^2|\bar{P}|)$.*

*Proof Sketch.* There are $L+1$ executions of modified value iteration that each have a time complexity of $O(|\bar{S}|^2|\bar{P}|)$ for a total time complexity of $O((L+1)|\bar{S}|^2|\bar{P}|)$. □

## 3.2 Conflict Resolution

The conflict resolution algorithm implements the conflict resolver that selects the optimal parameter across all safety processes. At a high level, the algorithm prunes a potentially optimal set of parameters for each severity level in the order of decreasing severity level followed by the interference to optimize the lexicographic objective function. More specifically, the algorithm initializes a set of potentially optimal parameters that can be recommended by each safety process. Following the order of the lexicographic objective function of a safety metareasoning system, the algorithm prunes the set of potentially optimal parameters in two steps. First, the algorithm prunes the set of potentially optimal parameters in the order of decreasing severity level based on the expected discounted cumulative number of times that each severity level is incurred using a given parameter in a specific state. Second, the algorithm prunes the set of potentially optimal parameters based on the expected discounted cumulative interference incurred by the safety process using a given parameter in a specific state. The result is a random optimal parameter that optimizes the lexicographic objective function. This is done *online* for each time step of the system.

**Algorithm 3:** The conflict resolution algorithm used by the safety metareasoning system between the *purple* and *blue* objects in Figure 2.

```
Input: The evaluation ratings $\rho^{\theta_i}_{\bar{s}^i}$ over the parameters
       $\bar{P}$ in the state $\bar{s}^i \in \bar{S}$ of the safety processes
       $\theta_i \in \Theta$
Output: A random optimal parameter $\bar{p}^* \in \bar{P}$
$\bar{P}^* \leftarrow \bar{P}$
for $\ell \rightarrow L, L-1, \ldots, 1$ do
    $\alpha \leftarrow \min_{\bar{p} \in \bar{P}} \left[ \max_{\theta_i \in \Theta} \Phi^{\theta_i}_{\bar{s}^i, \bar{p}}[\ell] \right]$
    for $\bar{p}$ in $\bar{P}^*$ do
        $\beta \leftarrow \max_{\theta_i \in \Theta} \Phi^{\theta_i}_{\bar{s}^i, \bar{p}}[\ell]$
        if $\beta > \alpha$ then
            $\bar{P}^* \leftarrow \bar{P}^* \setminus \{\bar{p}\}$
$\alpha \leftarrow \min_{\bar{p} \in \bar{P}} \left[ \max_{\theta_i \in \Theta} \Psi^{\theta_i}_{\bar{s}^i, \bar{p}} \right]$
for $\bar{p}$ in $\bar{P}^*$ do
    $\beta \leftarrow \max_{\theta_i \in \Theta} \Psi^{\theta_i}_{\bar{s}^i, \bar{p}}$
    if $\beta > \alpha$ then
        $\bar{P}^* \leftarrow \bar{P}^* \setminus \{\bar{p}\}$
return RANDOM($\bar{P}^*$)
```

Algorithm 3 describes conflict resolution. The set of potentially optimal parameters is initialized (Line 1). For each severity level in the order of decreasing severity level, the set of potentially optimal parameters is pruned by removing any parameter worse than the optimal parameter for the current severity level (Lines 2-7). For the interference, the set of potentially optimal parameters is pruned by removing any parameter worse than the optimal parameter for the interference (Lines 8-12). A random optimal parameter optimizing the lexicographic objective function is returned (Line 13).

We sketch a proof for the correctness and the time complexity of the conflict resolution algorithm below.

**Proposition 3** (Correctness)**.** *Conflict resolution selects a random optimal parameter $\bar{p}^* \in \bar{P}$ optimizing the lexicographic objective function given the evaluation ratings $\rho^{\theta_i}_{\bar{s}^i}$ in the state $\bar{s}^i \in \bar{S}$ of the safety processes $\theta_i \in \Theta$.*

*Proof Sketch.* Notice that there is a pruning step for each severity level $1 \leq \ell \leq L$ that prunes any parameter with a maximum expected discounted frequency higher than the optimal parameter for that severity level and a pruning step for the interference that prunes any parameter with a maximum discounted cumulative interference higher than the optimal parameter for the interference. Since this follows the order of the lexicographic objective function, any random remaining parameter is optimal. Thus, the algorithm is correct. □

**Proposition 4** (Time Complexity). *Conflict resolution has a time complexity of $O((L+1)|\bar{P}||\Theta|)$.*

*Proof Sketch.* There are $L$ severity level pruning steps that each have a time complexity of $O(|\bar{P}||\Theta|)$ and an interference pruning step that has a time complexity of $O(|\bar{P}||\Theta|)$ for a total time complexity of $O((L+1)|\bar{P}||\Theta|)$. □

## 4 Planetary Rover Exploration

We turn to an application of safety metareasoning systems to a planetary rover exploration domain that forms the basis of the experiments that we use to evaluate our approach in the next section. In this domain, a planetary rover must perform an analysis task that involves analyzing different points of interests $P$ within a region of a planet. Intuitively, the planetary rover will have a battery, a rock analyzer, a soil analyzer, and an objective report for the analysis statuses of all points of interests. Moreover, the planetary rover will be in a region of the planet that is composed as a grid where each cell experiences a type of weather. In each cell, the planetary rover can move north, east, south, or west and can also reboot its analyzers, charge its battery, analyze its current cell, and transmit its data back to earth. We now describe the analysis task of the planetary rover in detail below.

The planetary rover has 4 internal components: a battery of a battery level $b \in B = \{0, 1, \ldots, M\}$ where 0 is a discharged battery and $M$ is a charged battery, a rock analyzer of a health status $h_1 \in H_1 = \{\textsc{Nominal}, \textsc{Error}\}$, a soil analyzer of a health status $h_2 \in H_2 = \{\textsc{Nominal}, \textsc{Error}\}$, and an objective report $o \in O = \{\textsc{True}, \textsc{False}\}^P$ with an analysis status TRUE or FALSE for all points of interest $P$.

The planetary rover is within a region of a planet as an $m$ by $n$ grid where each cell is at a horizontal location $x \in X = \{1, 2, \ldots, n\}$ and a vertical location $y \in Y = \{1, 2, \ldots, m\}$ with weather of a type $w \in W = \{\textsc{Light}, \textsc{Dark}\}$.

The planetary rover can perform 4 nonstationary actions in each cell $(x, y)$: it can move *north* to a cell $(x, y-1)$, *east* to a cell $(x+1, y)$, *south* to a cell $(x, y+1)$, or *west* to a cell $(x-1, y)$ if the new horizontal position is between 0 and $m$ and the new vertical position is between 0 and $n$.

The planetary rover can perform 4 stationary actions in each cell $(x, y)$: it can *reboot* its analyzers to set the health statuses of the water analyzer $h_1$ and the soil analyzer $h_2$ to NOMINAL, *charge* its battery to the battery level $b' = (b+2)$ if the cell $(x, y)$ has weather of a type $w = \textsc{Light}$, *analyze* the cell $(x, y)$ if the health statuses of the water analyzer $h_1$ and the soil analyzer $h_2$ are set to NOMINAL, and *transmit* its data back to mission control on Earth to complete the mission if the objective report is $o = \{\textsc{True}\}^P$ with an analysis status TRUE for all points of interest $P$.

Any action discharges the battery to the battery level $b' = (b-1)$ and requires the battery to be at battery level $b > 0$.

### 4.1 Task Process

The task process $\Upsilon$ designed to complete the analysis task of the planetary rover is represented by an MDP $\langle S, A, T, R \rangle$. The set of states $S = X \times Y \times B \times H_1 \times H_2 \times O$ crosses a set of horizontal positions $X$, a set of vertical positions $Y$, a set of battery levels $B$, a set of rock analyzer health statuses $H_1$, a set of soil analyzer health statuses $H_2$, and a set of objective reports $O$. The set of actions $A = \{\uparrow, \rightarrow, \downarrow, \leftarrow, \ominus, \oplus, \odot, \oslash\}$ contains the *north* action $\uparrow$, the *east* action $\rightarrow$, the *south* action $\downarrow$ , the *west* action $\leftarrow$, the *reboot* action $\ominus$, the *charge* action $\oplus$, the *analyze* action $\odot$, and the *transmit* action $\oslash$. The transition function $T$ and the reward function $R$ are designed for the analysis task of the planetary rover.

### 4.2 Safety Processes

We consider three safety processes $\Theta$ designed to address the safety concerns of the planetary rover. Intuitively, each safety process has its own information that reflects its safety concern and can tune the action performed by the planetary rover by changing its wheel rotation rate (i.e., *speed*) and its steering (i.e., *direction*). Formally, each safety process $\theta \in \Theta$ has a set of states $\bar{S}_\theta$ that describe the safety concern but shares a set of parameters $\bar{P} = \bar{P}_1 \times \bar{P}_2$ with parameter factors $\bar{P}_1$ and $\bar{P}_2$: the wheel rotation rate parameter factor $\bar{P}_1 = \{\varnothing, \Downarrow, \Uparrow, \Diamond\}$ such that the value $\Downarrow$ decreases the wheel rotation rate, the value $\Uparrow$ increases the wheel rotation rate, and the value $\Diamond$ stops the wheel rotation rate and the steering parameter factor $\bar{P}_2 = \{\varnothing, \Leftarrow, \Rightarrow\}$ such that the value $\Leftarrow$ shifts the planetary rover to the left and the value $\Rightarrow$ shifts the planetary rover to the right (with the value $\varnothing$ for no tuning in both parameter factors). The transition function $\bar{T}_\theta$, the severity function $\phi_\theta$, and the interference function $\psi_\theta$ are designed to address a specific safety concern of the planetary rover. We describe the crevice safety process, the dust storm safety process, and the rough terrain safety process below.

**Crevices** The process, $\theta_c = \langle \bar{S}_c, \bar{P}, \bar{T}_c, \phi_c, \psi_c \rangle$, monitors for crevices to prevent the rover from inhibiting its wheels. The set of states $\bar{S}_c = F_c^1 \times F_c^2 \times F_c^3 \times F_c^4$ crosses the horizontal rover position relative to the crevice $F_c^1 = \{\textsc{None}, \textsc{Approaching}, \textsc{At}\}$, the vertical rover position relative to the crevice $F_c^2 = \{\textsc{None}, \textsc{Left}, \textsc{Center}, \textsc{Right}\}$, the rover speed $F_c^3 = \{\textsc{None}, \textsc{Low}, \textsc{Normal}, \textsc{High}\}$, and the rover offset relative to its normal path $F_c^4 = \{\textsc{Left}, \textsc{Center}, \textsc{Right}\}$. The transition function $\bar{T}_c$ reflects the dynamics between a state $s \in \bar{S}_c$, a parameter $p \in \bar{P}_c$, and a successor state $s' \in \bar{S}_c$, the severity function $\phi_c$ indicates the severity of a crevice in a state $\bar{s} \in \bar{S}_c$, and the interference function $\psi_c$ represents the interference of a parameter $p \in \bar{P}_c$ on an action $a \in A$ performed by the task process. These three attributes are designed to enable the crevice safety process to avoid navigating into crevices that inhibit the movement of the wheels of the planetary rover.

**Dust Storms** The process, $\theta_d = \langle \bar{S}_d, \bar{P}, \bar{T}_d, \phi_d, \psi_d \rangle$, monitors for dust storms to prevent the rover from destroying its sensitive sensors. The set of states $\bar{S}_d = F_d^1 \times F_d^2$ crosses

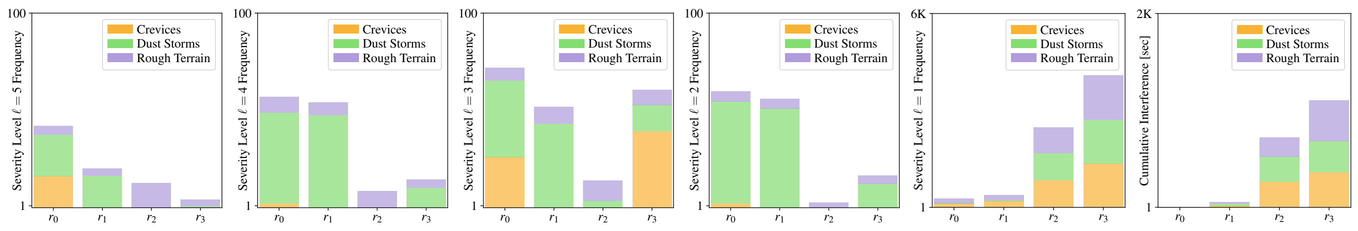


Figure 3: The performance of each planetary rover with respect to each severity level and the interference starting with no safety processes and ending with all safety processes. Note that the charts (a), (b), (c), and (d) have a limit of only 100 as unsafe operation is rare, the chart (e) has a limit of 6000 as safe operation is common, and the chart (f) has a limit of 2000.

the dust storm density $F_d^1 = \{1, 2, \dots, J\}$ with a maximum of $J$ and the rover mode $F_d^2 = \{\text{IsAwake}, \text{IsSleeping}\}$. The transition function $\bar{T}_d$ reflects the dynamics between a state $s \in \bar{S}_d$, a parameter $p \in \bar{P}_d$, and a successor state $s' \in \bar{S}_d$, the severity function $\phi_d$ indicates the severity of the dust storm in a state $\bar{s} \in \bar{S}_d$, and the interference function $\psi_d$ represents the interference of a parameter $p \in \bar{P}_d$ on an action $a \in A$ performed by the task process. These three attributes are designed to enable the dust storm safety process to avoid damaging the sensitive sensors of the planetary rover.

**Rough Terrain** The process, $\theta_r = \langle \bar{S}_r, \bar{P}, \bar{T}_r, \phi_r, \psi_r \rangle$, monitors for rough terrain to prevent the rover from damaging its wheel. The set of states $\bar{S}_r = F_r^1 \times F_r^2 \times F_r^3$ crosses the horizontal rover position relative to the crevice $F_r^1 = \{\text{None}, \text{Approaching}, \text{At}\}$, the rover speed $F_r^2 = \{\text{None}, \text{Low}, \text{Normal}, \text{High}\}$, and the terrain roughness $F_r^3 = \{1, 2, \dots, K\}$ with a maximum of $K$. The transition function $\bar{T}_r$ reflects the dynamics between a state $s \in \bar{S}_r$, a parameter $p \in \bar{P}_r$, and a successor state $s' \in \bar{S}_r$, the severity function $\phi_r$ indicates the severity of the rough terrain in a state $\bar{s} \in \bar{S}_r$, and the interference function $\psi_r$ represents the interference of a parameter $p \in \bar{P}_r$ on an action $a \in A$ performed by the task process. These three attributes are designed to enable the rough terrain safety process to avoid damaging the wheels of the planetary rover.

## 5 Experiments

We demonstrate that the application of safety metareasoning systems to the planetary rover exploration domain is effective in simulation. In particular, we compare a standard planetary rover to different safety metareasoning planetary rovers. The standard planetary rover $r_0$ does not have any safety metareasoning while the different safety metareasoning planetary rovers $r_{i>0}$ have a growing set of safety processes: $\Theta^{r_0} = \{\}$, $\Theta^{r_1} = \{\theta_c\}$, $\Theta^{r_2} = \{\theta_c, \theta_d\}$, and $\Theta^{r_3} = \{\theta_c, \theta_d, \theta_r\}$.

Each planetary rover completes the analysis task and addresses different safety concerns that occur stochastically either in isolation or simultaneously throughout 50 simulations. For the analysis task, the internal components of the planetary rover begin with a battery level $b = M = 10$, health statuses $h_1 = h_2 = \text{Nominal}$, and an objective report $o = (\text{False}, \text{False})$ while the region of the planet has $|P| = 2$ points of interest in an $m = 10$ by $n = 10$ grid such that each cell has weather of a type $w = \text{Light}$ with 0.8 probability or $w = \text{Dark}$ with 0.2 probability. For dust storms, the maximum dust storm density is $J = 10$. For rough terrain, the maximum terrain roughness is $K = 10$.

| Capabilities | Naive | Proposed |
|---|---|---|
| Analysis Task | 16000 | 16000 |
| + Crevices | + 2288000 | + 144 |
| + Dust Storms | + 43776000 | + 20 |
| + Rough Terrain | + 5483520000 | + 120 |

Table 1: The total states required by the naive approach that uses a monolithic MDP and our proposed approach to safety.

Figure 3 shows that our proposed approach is effective in simulation. In Figure 3(a) and (e), at the highest and lowest severity levels, the severity level 5 frequency decreases while the severity level 1 frequency increases from $r_0$ to $r_3$ as expected. In Figure 3(b), (c), and (d), at the middle severity levels, the severity level 4, 3, and 2 frequencies remain roughly equal or decrease from $r_0$ to $r_2$ but then increase at $r_3$. This is because the severity level 4, 3, and 2 frequencies for crevices and dust storms must increase to decrease the severity level 5 frequency for rough terrain since a lower severity level is strictly preferred to a higher severity level given the lexicographic objective function. In Figure 3(e), the cumulative interference increases from $r_0$ to $r_3$ as expected. As a summary, the system mitigates the severity of the safety concerns while reducing the interference to the task.

Table 5 shows the intractability of a naive approach using a *monolithic MDP* with every feature of each safety process designed to reflect our proposed approach: as the agent becomes capable of addressing each safety concern, the naive approach grows multiplicatively while our proposed approach grows additively with the set of states for each safety process.

## 6 Conclusion

We propose a novel metareasoning approach to safety in autonomous systems. First, we offer a definition of a safety metareasoning system, an evaluation rating generation algorithm for a safety process, and a conflict resolution algorithm for a conflict resolver. Next, we present an application of our approach to a planetary rover exploration domain. Finally, we demonstrate that the application of a safety metareasoning planetary rover is effective in simulation. Future work will investigate sophisticated, general-purpose safety processes for a wide range of safety concerns in planning and robotics.

## Acknowledgments

This work was supported in part by the NSF grants DGE-1451512, IIS-1813490, and IIS-1954782.

## References

[1] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein, "Belief space metareasoning for exception recovery," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2019.

[2] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," in *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 2020.

[3] C. Basich, J. Svegliato, S. Zilberstein, K. H. Wray, and S. J. Witwicki, "Improving competence for reliable autonomy," in *Proceedings of the ECAI Workshop on Agents and Robots for Reliable Engineered Autonomy*, 2020.

[4] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv:1606.06565*, 2016.

[5] S. Saisubramanian, E. Kamar, and S. Zilberstein, "A multi-objective approach to mitigate negative side effects," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020.

[6] S. Zhang, E. H. Durfee, and S. P. Singh, "Minimax-regret querying on side effects for safe optimality in factored MDPs," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.

[7] R. C. Arkin, "Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture," in *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, 2008.

[8] J. Shim, R. Arkin, and M. Pettinatti, "An intervening ethical governor for a robot mediator in patient-caregiver relationship," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017.

[9] J. Svegliato, S. B. Nashed, and S. Zilberstein, "Ethically compliant sequential decision making," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[10] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan, "Inverse reward design," *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

[11] R. Dearden, T. Willeke, R. Simmons, V. Verma, F. Hutter, and S. Thrun, "Real-time fault detection and situational awareness for rovers," in *IEEE Aerospace Conference*, 2004.

[12] V. Verma, G. Gordon, R. Simmons, and S. Thrun, "Particle filters for rover fault diagnosis," *IEEE RA Magazine*, 2004.

[13] J. P. Mendoza, M. Veloso, and R. Simmons, "Mobile robot fault detection based on redundant information statistics," in *Proceedings of the IROS Workshop on Safety in Human-Robot Coexistence and Interaction*, 2012.

[14] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Fault detection in a mobile robot using multiple-model estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998.

[15] P. Goel, G. Dedeoglu, S. I. Roumeliotis, and G. S. Sukhatme, "Fault detection in a mobile robot using a neural network," in *Proceedings of the International Conference on Robotics and Automation*, 2000.

[16] A. S. Manne, "Linear programming and sequential decisions," *Management Science*, 1960.

[17] R. Bellman, "Dynamic programming," in *Science*. American Association for the Advancement of Science, 1966.