

Driver Override for Safety-Critical Vehicles and Networks

E. Atkins

University of Michigan

Copyright © 2009 SAE International

ABSTRACT

Operator error is a primary cause of vehicle accidents, yet human ingenuity is critical to effectively react in situations automation is not prepared to handle. Human operators have always been the ultimate authority, but their decisions may or may not be safe. This paper explores the constraints and requirements of vehicle systems that support automation override of a human operator. We adopt the view that a human operator remains the ultimate authority until grave risk is encountered, at which time the automation overrides strictly to re-establish a safe operating state. An override system must continually monitor vehicle state, predict near-term risk levels, compute a strategy to mitigate substantial risk, and warn the operator of the impending risk given sufficient time. Override action must occur just-in-time to re-establish a safe state before risk increases beyond the "grave" threshold. Controlled flight into terrain and aircraft damage/failure override case studies are presented, along with a discussion of how such aircraft override technologies could translate to ground vehicle systems.

INTRODUCTION

The proliferation of reliable, low-cost sensing, processing, and communication systems is revolutionizing the transportation industry. Drivers and pilots expect a myriad of real-time data from onboard sensors and networks (e.g., weather and traffic). Embedded monitoring and control systems have substantially improved safety, from anti-lock braking and active maneuvering systems in automobiles to automatic fuel management and zero-visibility approaches for aircraft. The driver or pilot, however, remains the

"ultimate authority" for all decisions. Humans are currently our benchmark for devising innovative solutions to unexpected and potentially dangerous situations. However, these same operators are a primary cause of aircraft and automobile accidents, most often due to mistakes or improper situational awareness, on occasion due to lack of training or malicious intent. We can work to improve operator situational awareness and training, but we can never guarantee a human operator will make perfect decisions or always have good intentions. How, then, can automation technology assist human operators to further minimize or eliminate accidents?

The so-called "refuse-to-crash" automation concept [1] has been proposed for commercial aircraft, a platform already possessing sophisticated flight management systems proven reliable over the last three decades. As stated in [1], "accident statistics show that so-called "loss-of-control" (LOC) events, infrequent situations where a pilot, for whatever reason, cannot fly straight and level, have been the primary cause of fatal U.S. commercial airline crashes for most of the last decade". The goal of refuse-to-crash automation is not to eliminate the pilot, but rather to intervene (i.e., override) in dire situations where aircraft safety is unquestionably jeopardized. The refuse-to-crash automation concept is applicable to surface vehicles as well as aircraft.

This paper explores the high-level requirements for a general refuse-to-crash system in the context of the necessary decisions and safeguards. It then explores system specifics for a piloted aircraft and presents two representative case studies, each of which is a major cause of aviation accidents each year. At the highest level, a refuse-to-crash system must be capable of robustly accomplishing the following sequence of actions:

- (i) *Monitoring* the vehicle and environment states to identify impending high-risk situations that might require override. Potential high-risk scenarios will also prompt computation of a mitigation strategy.
- (ii) Clearly *warning* the operator of impending safety violations and required remediation it will execute should the operator do nothing. This procedure minimizes “automation surprise” during override.
- (iii) *Overriding* the operator to re-establish safe operation, providing real-time data on the override condition, maneuvers, and termination criteria.
- (iv) Re-establishing a *stable, safe operating state* that enables smooth transition of control authority back to the operator.

As with current automation systems, clear safety violation warnings provided well in advance will typically prompt the operator to take corrective action without the need for override. With sufficient time, the operator could initiate the recommended remediation or an alternative recovery of their choice. Warnings may not always be provided well in advance, however, or may rely on data sources difficult to quickly convey to the operator and even more difficult for the operator to translate into an effective resolution (e.g., deconfliction with multiple nearby vehicles, or re-establishing stable operation after severe damage). The operator, due to incapacitation, disorientation, or malicious intent, may also not properly heed the provided warnings. In such situations, automation can and should override. In most cases, re-establishment of a safe, stable operating condition will enable the operator to re-establish control without further incident. With incapacitation or malice initially unknown to the automation, further override will likely be required, although explicit detection of malice or incapacitation is a difficult challenge beyond the scope of this paper.

Numerous questions emerge when considering override systems. How can we guarantee the automation will only intervene when absolutely necessary? How can we guarantee override responses will assure rather than compromise safety? Mode confusion and automation surprise are already major concerns for automation systems operating on manned vehicles. How can we design the override system to always convey its actions and intent clearly and unambiguously to a potentially disoriented pilot or driver? Pilots and drivers have a long history of being “in control”. How can we gain operator and passenger acceptance of override systems? Once accepted, how do we prevent a sense of complacency from the operator, who will still need to handle situations beyond those managed through automation? Finally, how can we guarantee the override system will itself recognize

thus not override situations it either doesn’t fully understand or for which it cannot devise a safe response strategy?

This paper focuses on design requirements that will ensure the override automation is itself operating effectively, recognizing but not directly addressing questions related to operator cognizance and interaction through a potential override scenario. Thus, the primary objective of this manuscript is to characterize override system properties that will guarantee it acts only in a safety-preserving manner and only when necessary. Below, a high-level override behavior model is presented, and an operational architecture for override is proposed to guarantee correct operation over a suite of override scenario classes. To ground this discussion, two aircraft override scenarios are presented. In the first, a controlled-flight-into-terrain (CFIT) situation must be resolved through a refuse-to-crash (override) intervention. In the second, we will overview work toward an emergency flight management system capable of landing safely when mechanical failures or structural damage occurs. This last case is particularly challenging for aircraft since guidance and control must be maintained throughout a potentially long and difficult approach to a safe landing site. Issues in translating aircraft override to ground vehicle systems are discussed, followed by a brief summary.

VERRIDE DECISION-MAKING PROTOCOL

The task of the override system is to recognize and react in time to avert situations that would result in an accident. The fundamental requirements of the override system, therefore, are to *unambiguously recognize* such situations and their real-time constraints, and to *correctly react* to avoid an accident. Given a human operator, override should also minimally disrupt the operator’s nominal course of action. Figure 1 illustrates the high-level matrix of potential vehicle and override states with respect to safety, highlighting both the appropriate override case to be targeted as well as the inappropriate decisions to be avoided, particularly “improper override”.

At the top level, the actual vehicle state is either safe or unsafe. Safety state can be classified by the automation system as safe, marginal, unsafe, or unknown. Note that we group marginal safety with safe operation in this diagram since override is reserved for very high-risk circumstances. For cases the automation classifies as unknown, the automation would clearly but unobtrusively indicate this unknown status to the operator, but would initiate neither warnings nor override.

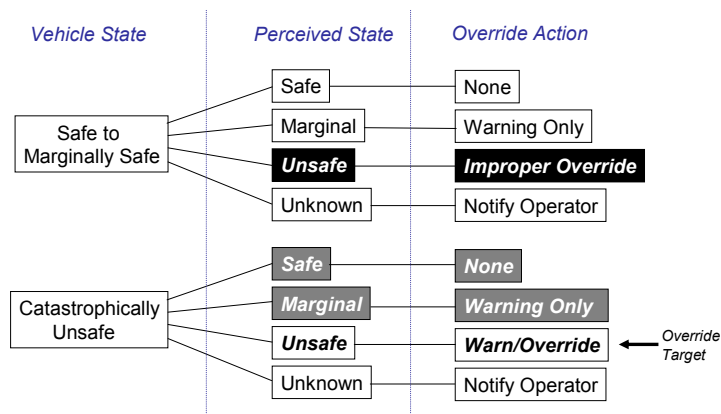


Figure 1: Safety Override Decision Cases.

When actual vehicle state is safe to marginally safe, the appropriate override action is to do nothing or warn the operator of marginal conditions as is done in air and surface vehicles today. When actual vehicle state is projected to become catastrophically unsafe, or result in a crash, the override system must correctly detect the unsafe condition and execute the warn-then-override sequence outlined above.

Perhaps the primary use for Figure 1 is to highlight incorrect automation decisions that must be avoided in any override implementation. The most egregious error is for the override system to improperly assume control when the vehicle is actually operating safely. This would occur if the automation incorrectly perceives an unsafe condition and acts, as indicated by the black-shaded Figure 1 state. It is also undesirable for the override system to improperly classify a truly unsafe condition as safe or marginal, as indicated by grey shading in Figure 1. However, note that these cases would revert to today's standard in which the operator must detect and resolve compromising situations without override and with warnings only in well-understood cases.

Given vehicles operate capably today and are becoming increasingly networked, successful override systems will inevitably build upon existing technologies and rely on a large suite of information resources. To avoid "improper override" cases, override systems must err toward unknown or even marginal classifications given incomplete or inconsistent information. However, the override system can be built-up over time to accommodate an increasing suite of high-risk/unsafe state classes, each of which will typically require a specific set of models, data, and override procedures.

Figure 2 provides an overview of the sense-act operations an override system would handle. The indicated high-risk/unsafe states span environment, vehicle, and communications problems, focusing primarily on scenarios where mitigation translates to vehicle maneuvering to avoid an accident. The leftmost Figure 2 column lists the sources of data an override

system would use to detect an unsafe scenario. The center column lists the broad classes of unsafe states to be managed, while the rightmost column indicates the recovery strategy(ies) appropriate for each unsafe state class. The high-risk state classes are discussed below in the context of the override sense-act strategies required for each.

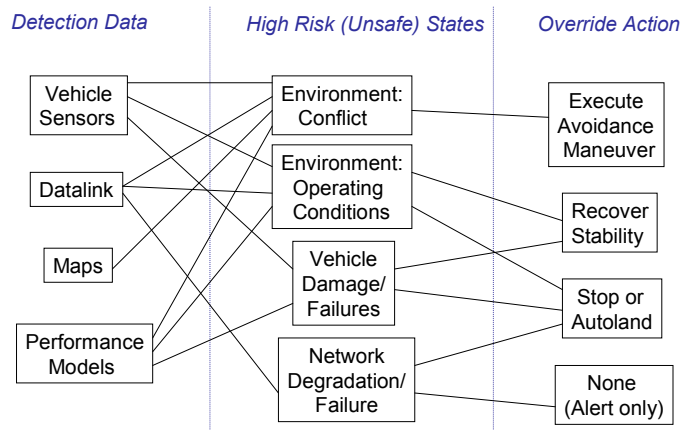


Figure 2: Override Sense-Act Matrix.

Environment Conflict: An environmental conflict can result either from operator error or from other vehicles/obstacles impeding a nominally-open passage. The operator error case would typically occur when the vehicle is instructed to drive or fly in a manner that will impact mapped or sensed terrain/buildings. Conflicts with other vehicles or obstacles (e.g., debris on a roadway) would be detected through sensors or datalink to the other vehicles or central (air) traffic control. Conflict scenarios are resolved by executing an avoidance maneuver to prevent collision. The maximum delay between detection/warning and override response will range from under a second in the case of swerve-to-avoid to over a minute in cases where the current planned vehicle trajectory is expected to conflict with another vehicle or terrain. An aircraft Controlled Flight into Terrain (CFIT) override case will be discussed in more detail below.

Environment Operating Conditions: The environment can impose hazardous operational conditions that also require override action. For example, degraded weather conditions ranging from icing on roads and airframes to high winds can introduce very high risk if not handled properly. In some cases, control will be lost temporarily, as for a car sliding on ice or an aircraft upset by violent wind gusts. Such states would prompt an override control system to automatically reestablish a stable operating condition (e.g., equilibrium/trimmed flight, minimal car wheel slippage). In other cases (e.g., substantial airframe icing, impassable road or bridge damage) the override must take stronger action to