**Macintosh HD:Users:Dependable:Shared:Documents:Templates:DCi.logo.pdf**

Sponsor: AFRL

Project No.: 144

Contract: ASTRA IST

This document is for authorized use only. Any information contained herein should be considered proprietary unless otherwise specified.

© 2024 Dependable Computing. All Rights Reserved

Dependable Computing

V0.2 1/19/2024

Behavior Description

Zone Alert Service

Abstract

This document provides behavior descriptions of the zone alert service designed to inform the end user and stakeholders of the services behavior with respect to its provided services.

.

Contents

1 Introduction 1

2 Basic Behaviors 1

3 Complex Behaviors 4

4 Satisfactory Parameters 4

1. Introduction

The Zone Alert Service is a stand-up service that can be activated as part of an OpenUxAS instance. Its purpose is to alert subscribers to when vehicles associated with the UxAS instance in one of two states:

1. Are in current violation of a zone
2. Are in danger of violating a zone within a future time window.

These cases are more carefully defined in the next sections.

1. Service Parameters

The service is parameterized by

* Vehicle collision radii, **RV**
* Path deviation tolerance **ED**
* Lookahead time window **TW**

These values are defined, more specifically as follows:

## Vehicle Collision Radii, RV

*Vehicle Collision Radii*: The radius of a circle, defined in meters, for a given vehicle, defining a circle in the horizontal plane (of OpenUxAS) centered on the vehicle’s reported position. The radius of the circle is the distance around the aircraft within the vehicle is circumscribed on the horizontal plane. In other words, when looked at from above, the vehicle must always “fit” in the circle based on the vehicle’s reported position.

**VP**

**RV**

The figure to the right shows a vehicle at position **VP** with collision radius **RV**.

*Source*: Vehicle configuration record in OpenUxAS. It is assumed that this radius is currently defined in the OpenUxAS vehicle configuration data sent out by OpenUxAS as the beginning of operation during vehicle announcement.

This radius forms the basis for determining how close a vehicle can get to the edges of a keep-in or keep-out zone without violating it. See the figure to the right.

## Path Deviation Tolerance, ED

*Path Deviation*: The **closest distance,** defined in meters, from a vehicle’s reported position to its ***currently*** assigned path.

**ED**

*Path Deviation Tolerance*: A **maximum path deviation** for a vehicle relative to the vehicle’s path and current position.

An example maximum path deviation is defined to the right. It defines a corridor around a vehicle’s planned path, see the figure to the right, that if a vehicle position remains within, projected zone violations within the defined time window are not reported.

**path**

Given that vehicles may follow paths in performance of their mission, and it is a given of path finding services of OpenUxAS that computed paths for vehicles will obey all keep-in and keep-out zones, then it is useful to not alert when a vehicle is approaching a violation but is currently following a path. This allows paths to properly approach zone boundaries without resulting in superfluous warnings.

Note, if a vehicle DOES violate a zone, an alert will occur regardless of path distance.

## Lookahead Time Window, TW

*Lookahead Time Window*: A time window, defined in seconds, within which the service will project the last reported linear trajectory of the vehicle into the future to see if it violates keep-in/out zones.

*Linear Horizontal Velocity*: The velocity of a vehicle projected into the horizontal plane of OpenUxAS.

The service is based on an iterative projection of the current linear horizontal velocity of a vehicle from its reported position to determine if the vehicle violate keep-in/out zone definitions in the horizontal plane.

1. Basic Behaviors

## Case 1: Report Potential Zone Violations Based on Current Linear Trajectory

### The service will alert subscribers when a violation is predicted to occur in a future time window based on the current position and linear velocity of each aircraft. This basic algorithm is meant to match the linear projection models used for DIADLUS [cite].

### Projected Violation Model

Violations are detected from the last received vehicle state report, Ts, to a future time limit Tw, in the parametric range for time, t, as **Ts**<=**t**<=**Ts**+**TW**.

Ts+Tw

The model of projected violation is based on the linear projected trajectory of the vehicle from its state-reported position to the final position in the time window along vector **Pv**, which has length **TW**\***VS** and where **Sv** is the current speed of the vehicle with direction **Vn** (the normalized velocity vector), both from the state report at time **Ts**.

Projected Future

Violation

Tv

Keep-Out Zone

Tp

Ts

A projected violation is reported if any point of the configured ‘collision radius’ around a vehicle (described above for violation detection) would be on or inside a keep-out zone or on or outside a keep-in zone for any given projected future position of the vehicle along the vector of vehicle position, described above along the projected position vector for time **TS**<**Tp**<=**t**<=**Ts**+**TW** as defined by variable time t. If a violation is projected to occur, the time of projected first occurrence is included in the notification sent to all subscribers.

Below are illustrated two example cases. In the first, no future violation is reported as none is projected in the time window, in the second, a violation is reported as projected to occur.

### Past Projected Violations Are Not Reported

Note that this vector does not include positions along the vector for **t**<**Tp**. We will not report projected violations that only occur in the past relative to the current iteration time of the service. Below is a figure showing two example projections of violation, but one is in the past from **Ts**<=**t**<**Tp**. Because it occurred before this iteration only, it is not reported.

### No Re-Reports of Projected Violations

If a vehicle continues to project violation in successive iterations, those are not reported as additional notifications.

### Projected Re-violations Are Reported

If at least one iteration occurs where the projected violation would no longer occur, and then a future iteration occurs where it would again occur, this is reported to subscribers a second time. See the figures below for examples.

### Stale Reporting Errors

If the present time of the iteration of the service Tp is past the time window **Ts**+**Tw** for projection of the state vector, the service will report an error message to all subscribers. It will do this for any vehicle with a vehicle position report that is thus so stale. It will not repeat the message unless a new status of the vehicle is received. If that report is still stale beyond the time window, a new error report will be produced.

Thus, if continuous message lag exceeds **TW** error messages will be continuously delivered to all subscribers.

### Potentially Report While Following a Route

When a vehicle has filed a path route and is currently executing it, it will only have projected zone violation warnings sent out if it deviates too far from its planned route.

Keep-Out Zone

Ts+Tw

Tp

The figure below shows the expected behavior of this feature. A vehicle is approaching a keep-out zone, but is following its filed path route closely. So long as it stays on the route, the projected future keep-out zone violation will not be sent to subscribers, as illustrated.

Projected Violation

Ignored

Position at Ts is within Ed of Route

Ts

However, should the vehicle deviate too far from the line segment between its present waypoints, by tolerance distance to the line segment Ed, then notifications will be sent to the vehicle, as illustrated in the figure below. In that figure, a vehicle deviating within Ed does not report the future violation. To the right in the figure, if a projected violation occurs while deviating from route by distance Ed, or if distance Ed is exceeded while a projected violation is detected, subscribers will be notified. The notification will include a flag indicating that a route is filed but that the vehicle is exceeding the allowed distance tolerance.

Position at Ts is further from Route than Ed

Ts

Projected Violation

Reported

Tp

Ts+Tw

Keep-Out Zone

### Behaviors of Service with Noise in Vehicle Entity Report Measures

**TBD**

## Case 2: Violation of a Zone Boundary

If at any time the reported location of a vehicle is in violation of a zone boundary, the service will alert subscribers to the zone boundary violation. The above graphical example shows the classic examples of violating boundaries of a keep in our keep out zone, respectively. The service sends one notification for each zone that is violated. The service sends a separate notification for each vehicle for each zone that the vehicle is currently violating. It is assumed that there are a limited number of zones and vehicles, so that the number of simultaneous notifications to be sent and received will never burden OpenUxAS or services.

### Violation Region

Each vehicle defines a radius around it considered its ‘collision radius’ as part of its entity configuration in OpenUxAS. This radius in combination with the vehicle’s current reported position is used to compute violation. If any part of this defined volume intersects any part of keep-out zone’s interior or keep-in zone’s exterior, then a zone violation is occurring.

Keep-Out Zone

Violation

Vehicle

With Collision

Area

Velocity

Vector

### Notification Timing

No violation

reported

Notification occurs during the iteration of the service in which violation is detected by the latest state reports of the vehicles. It is **not** reported during an iteration when the vehicle will enter the violation before the next iteration. It **is** reported in the iteration in which vehicle position report satisfies violation conditions. For example, as seen in the figure, if a vehicle quickly passes through a keep-out zone between iterations of the algorithm, no violation will be reported. See the example of the figure to the right. This generally requires a vehicle to be moving very quickly against a very narrow keep-out zone, or brief escape from a keep-in zone.

Keep-Out Zone

Vehicle position at 2

successive iterations

Vector

Ti2

Ti1

### Case 2A: Don’t Report Again While Still the Same Violations

Not

Reported

Not

Reported

Reported

Ti4

The service will never repeat notifications for the same zone violation by the same vehicle while that violation continues to persist. If the violation terminates, and then the vehicle reviolates, then the notification will be resent. See the associated figure.

Keep-Out Zone

Reported

Ti3

Ti1

Ti2

### Case 2B: Report Again Later When It Reoccurs

Not

Reported

If a violation ends in an iteration and is detected again in iteration, a new notification will be issued.

Ti3

Not

Reported

If it ends and begins again between iterations (the change is ***not*** detected as it occurs between iterations), then renotification does not occur.

Ti2

Reported

Ti1

### Case 2C: Boundary-Skimming Behavior

If a vehicle’s flight path is erratic or a zone has a highly complex geometry, then it is possible that a vehicle skimming along a boundary may frequently enter and leave violation with a zone boundary. Such rapid oscillation of violation state in successive iterations is captured by this service and will result in a rapid succession of violation events.

### Case 2D: Position Noise Behavior

OpenUxAS is currently applied to the AMASE vehicle simulator. If OpenUxAS were to be used with real-world reporting and there is noise in vehicle position data over time, then the behavior of the violation alerting behavior is undefined.

## Case 3: No Reporting Regarding Operating Regions

The system does not respond or care about operating regions. Any vehicle entering or leaving an operating region or existing inside or outside of an operating region has no impact on the zone alert service.

1. Complex Behaviors

It is the intent for the service to always observe the above basic behaviors regardless of live state of vehicles in OpenUxAS. Therefore, no combinatorial behaviors are expected. Examples of this behavior should be demonstrated in this section.