Adding RF Situational Awareness to Robotic Simulation Systems

Sydney Gibbs[a], Mitch Thornton[a], Darrell Young[b*]
[a] Southern Methodist University, Dallas, Tx; [b] Raytheon Intelligence and Space, Richardson, Tx

ABSTRACT

Various tools are now available to assist the roboticist in developing autonomy algorithms for tasks such as path planning or collision avoidance. Many tools support the integration of live or simulated RGB cameras, LIDAR, radar, and IMU sensors. This paper will describe adding an RF sensor. The proposed RF sensor detects radio and locates emitters in the environment for the purpose of collision avoidance. We outline an approach to share data to help locate and avoid collisions. The protocol is designed to maximize safety, privacy, security, timeliness, and other desirable properties discussed in the paper. Preliminary results are shown to illustrate the concepts.

Keywords: Robotics, Simulation, Situation Awareness, V2X, V2I, I2V, Digital Twin, Collision Avoidance, Cybersecurity

1. INTRODUCTION

Sensing Radio Frequency (RF) emitters can help robots navigate and avoid collisions. RF-Situational Awareness Sensors (RF-SAS) can detect energy in day or night conditions, in all weather, and beyond line-of-sight. Cooperative avoidance is when entities share their state and intent information, such as when a pedestrian broadcasts his position to nearby vehicles[1]. The city or factory of the future may have cooperative information-sharing zones where all entities report their position, thus significantly reducing collision risk, like controlled airspace. However, in the foreseeable future, there will be a mix of RF-emitting entities in the environment, with only some reporting locations. The proposed RF-SAS module detects RF energy for the purpose of collision avoidance (CA). The primary use case discussed in this paper is for automotive vehicle technology; however, the concepts generally apply to other robotic domains.

RF-SAS can augment or assist in autonomous CA decisions in different ways. Decisions can range from quick reaction maneuvers to avert an accident to more general policy decisions to slow down based on current and past sensing of RF activity. RF-SAS can act as a cue to steer and refine other perception sensors such as radar, lidar, or cameras.

Collisions can be avoided by sensing RF energy due to the ubiquitous presence of cellular phones, smartwatches, and other Bluetooth, Wi-Fi, or cellular devices. The detection and location of these personal devices present technical and policy issues worth solving due to the tremendous benefit of reducing vehicular accidents. Each year, 1.35 million people are killed on roadways worldwide [2]. Every day, almost 3,700 people are killed globally in crashes involving cars, buses, motorcycles, bicycles, trucks, or pedestrians. More than half of those killed are pedestrians, motorcyclists, or cyclists [2]. 38,824 people died, and an estimated 2,282,015 people were injured on U.S. U.S. roads in 2020[3]. Over 91% of the people in the world have a cellphone; however, they are not currently integrated into any kind of CA system. Soon it may be possible for RF-SAS, combined with other sensors and V2X technology, to help avoid collisions with non-reporting devices. In the future, devices may report their state for purposes of CA.

A major technical challenge is quickly and securely obtaining and disseminating sufficiently accurate situational awareness information to support taking proper action. A major policy issue is how to ensure the collected information on personal devices does not violate the individual's privacy rights. Engineering simulations can help solve both policy and technical issues. The simulations can provide the necessary information for technologists and policymakers to make implementation decisions.

The U.S. Department of Transportation (USDOT) has estimated that Vehicle-to-Vehicle (V2V) communications based on Digital Short-Range Communications (DSRC) can address up to 82% of all crash types involving unimpaired drivers in the U.S. and approximately 40% of all crashes occurring at intersections[4]. Detecting RF signals to help prevent collisions until a fully cooperative system where all entities report position may be a useful interim measure.

*dlyoung@smu.edu, 703 582 4124, This work was not performed by Dr. Young as a Raytheon Technologies employee.

2. DEVELOPMENT OF AN RF-SAS MODULE

Many robotic and simulation tools support a variety of sensors shown in Figure 1. Most simulation tools support a Python gym or gym-like module to support the development of reinforcement learning algorithms.

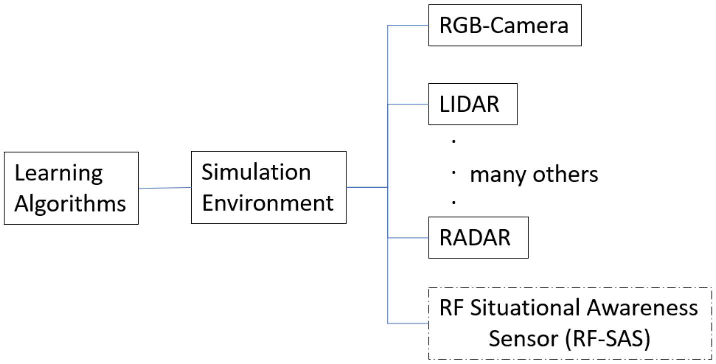


Figure 1 RF Situational Awareness Sensor added to existing autonomy simulation and learning environments.

Table 1 provides a partial sampling of autonomy development tools. The tools support various types of sensors. Almost all of them support Python gym or have a gym-like interface for machine learning. Our approach is to develop an RF-SAS Python gym environment to maximize integration interoperability.

Table 1. Autonomy Simulation Tools

Simulation Environment	RGB-Camera	LIDAR	RADAR	Event Camera	GPS/GNSS	Stereo Photogrammetry Depth	Accelerometer	Magnetic Compass	Gyroscope	Python Gym Support
CARLA[5]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VISTA[6]	✓	✓		✓	✓	✓	✓			✓
NVIDIA[7] Isaac	✓	✓				✓	✓		✓	✓
ROS Tools[8]	✓	✓	✓		✓	✓	✓		✓	
SUMMIT[9]	✓	✓			✓	✓				✓
Unity 3D[10]	✓				✓	✓	✓	✓	✓	✓
AWS RoboMaker[11]		✓			✓		✓		✓	✓
SVL[12]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PGDrive[13]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SUMO[15]										✓
Gazebo[14]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Our approach to developing a Python Gym RF-SAS environment is to first use MATLAB to perform a high Level-of-Detail environment characterization to investigate system parameters such as detection range, bearing angle accuracy, triangulation covariance error, etc. and then transfer the result to the initial Python Gym module parametrically. Low, Medium, and High levels of simulation detail is described in **Table 2**.

Table 2. RF-SAS State Simulation Level-of-Detail

Low	Medium	High
State parametrically assigned	State estimated using a kinematic model with assigned values for measurement properties	State estimated from measurements simulated using a physical model that includes signal, environment, and noise properties

3. RF SITUATIONAL AWARENESS ENVIRONMENT

Each of the following steps to develop the operational parameters is described below. The steps are the outline for this section. The next section discusses how operational parameters are implemented in a Python gym environment.

1. Scenario laydown of receivers and transmitter
2. Estimate warning delay from the composition of its constituents.
3. Compute SNR at receivers based on two-ray path loss and distance.
4. Compute angle measurement standard deviation for the computed SNR
5. Use the noisy angles to compute the estimated location and error covariance.
6. Use error covariance to compute the Probability of Collision
7. Compute impact speed to compute risk.
8. Privacy and cybersecurity concerns

3.1 Scenario laydown of receivers and transmitter

The vehicle and collision object state can be modeled as,

$$\hat{\boldsymbol{x}}(\boldsymbol{t}+\Delta\boldsymbol{t}) = \begin{pmatrix} \boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{\Delta s}\cdot\mathbf{cos}\,(\boldsymbol{\theta}(\boldsymbol{t}) + \Delta\boldsymbol{\theta}/\mathbf{2} \\ \boldsymbol{y}(\boldsymbol{t}) + \boldsymbol{\Delta s}\cdot\mathbf{sin}\,(\boldsymbol{\theta}(\boldsymbol{t}) + \Delta\boldsymbol{\theta}/\mathbf{2} \\ \boldsymbol{\theta}(\boldsymbol{t}) + \Delta\boldsymbol{\theta} \end{pmatrix}, \qquad (1)$$

where Δs and Δθ are the incremental longitudinal and rotational motion computed from odometers or an inertial measurement. The probability density function (pdf) of $x=(x,y,\theta)^T$ having $\boldsymbol{\Sigma}$ covariance matrix and an $\hat{\boldsymbol{x}}$ mean:

$$p(\boldsymbol{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^3\sqrt{\det(\boldsymbol{\Sigma})}}e^{-\frac{1}{2}\left((\boldsymbol{x}-\hat{\boldsymbol{x}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\hat{\boldsymbol{x}})\right)} \qquad (2)$$

The covariance matrix can result from a filter process such as the Kalman Filter. We define it directly by,

$$\boldsymbol{\Sigma} = E\left((\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^T\right). \qquad (3)$$

The scenario shown in Figure 2 defines the uncertainty of the vehicle and the collision obstacle, which could be a pedestrian, bicyclist, or another vehicle. The origin is taken as the center of the intersection. The initial speed of the vehicle is u_v. The vehicle and collision obstacles are measured from their centers. The vehicle and obstacle shape is characterized by the length and width of an enclosing boundary box. The error covariance and the bounding box are used to compute the probability of collision, as described in a later section.

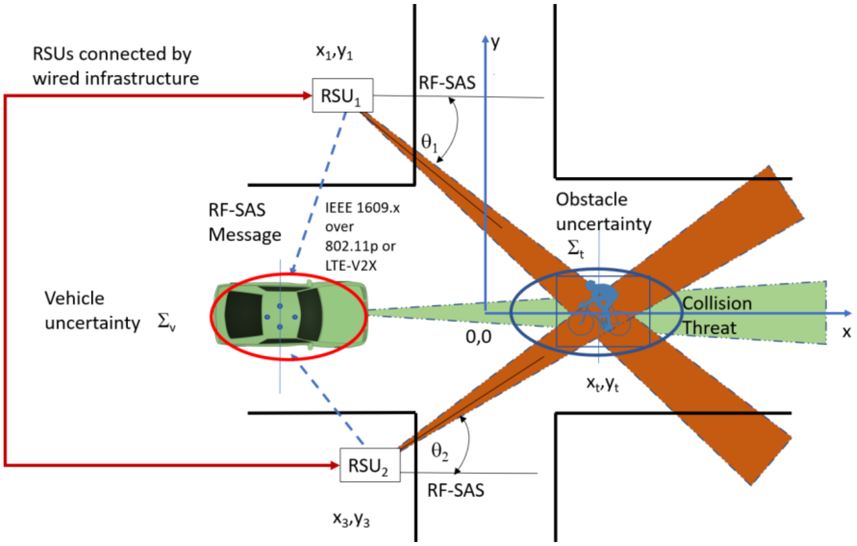


Figure 2 Scenario of moving vehicle with uncertainty covariance Σ_v approaching collision obstacle with uncertainty Σ_t.

In the simulations described below, the starting location is -100 meters in the x-direction from the intersection. RSU RF-SAS sends a message with bearing information with a short delay after the start of the simulation. The OBU unit makes its own observations and estimates the location of the obstacle. After some total reaction delay, $\tau_{D,}$ the vehicle applies the braking action to avoid the collision. The collision is avoided if the vehicle comes to a stop before impact. If it does not come to a stop, the braking action deceleration reduces the impact speed, $v_{v,}$ and the resulting risk, R, of a severe accident. In all the simulations, the braking deceleration is set to a constant 0.47g.

3.2 Estimate warning delay from the composition of its constituents.

Timeliness is an important aspect of RF-SAS measurement quality because of the short timelines to avoid a collision. Scanning channels take time. The problem is compounded by the fact that the emitter is not guaranteed to be emitting during the observation scan time.

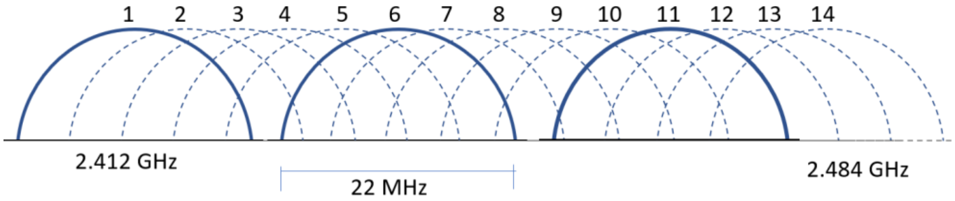


Figure 3 A single receiver may require up to 4 seconds to scan 40 Wi-Fi channels. Shown are the 14 channels in the 2.4 GHz band. Not shown are channels in the 5 GHz band.

The time to first Probability-of-Intercept (POI) can be estimated using a calculation of overlapping observation and activity windows described by Self [17] et al., as shown in Figure 4

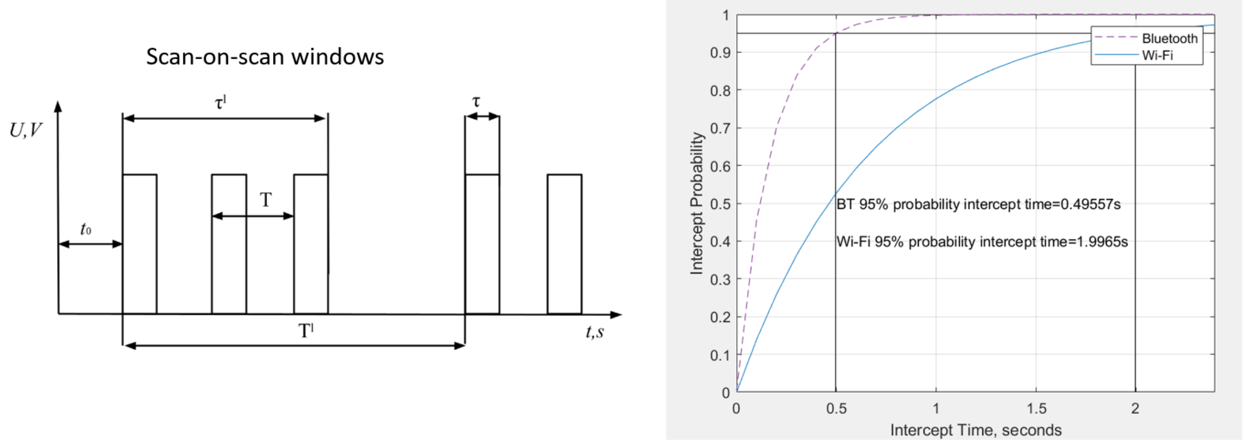


Figure 4 Left: overlapping observation and activity windows; Right: Time-to-first intercept for Bluetooth and Wi-Fi when using six scanning receivers.

The allowable maximal delay requirement for safety applications that have been commonly utilized, for instance, the warning of intersection collision or the warning of forward collision, is 100 milliseconds.

The overall reaction delay, $\tau_{D,}$ is a combination of scan-on-scan, message transmission, and calculation time. We explore the range of allowable reaction delay by investigating its relationship to fatality risk as a function of the initial speed.

3.3 Compute SNR at receivers based on two-ray path loss and distance

The Signal-to-Noise (SNR) ratio for Wi-Fi is computed using a combination of transmit power, $P_{t_{dBm}} = 27\ dBm$ and transmit gain, G_r=3 *dBm*, so that the Equivalent Isotropic Radiated Power (EIRP) is less than the legal limit of +30 dBm. The Path Loss (PL) is computed using the line-of-sight (LOS) distance from the transmitter, and the transmitter and receiver heights, ht and $h_{r,}$ set to 1-meter in the two-ray path loss equation,

$$Path\ Loss: PL = P_{t_{dBm}} - P_{r_{dBm}} = 40log_{10}(d) - 10log_{10}(Gh_t^2 h_r^2)\ , \qquad (4)$$

where G is the combined antenna gain long the line-of-sight path. The two-wave path loss model is derived by considering the effect of the radio wave reflecting from the ground, as shown in rays 1 and 2 in Figure 5. We intend to include 3D Geometry Based Scattering Models (GBSM) from multiple surfaces in future work.

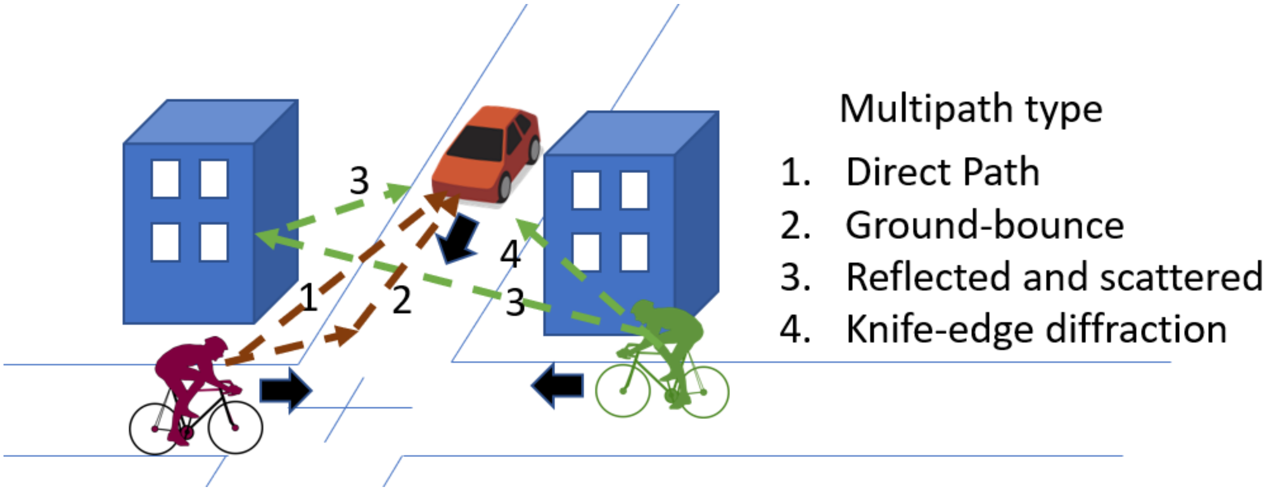


Figure 5 Outdoor multipath scenario. 1) Direct path; 2) Ground-bounce; 3) Reflected; 4) Diffracted

Table 3 shows the parameters used in the two-ray path model for 25 and 100-meter line-of-sight distances.

Table 3. Estimated Signal-to-Noise (SNR) for Wi-Fi and Bluetooth

Type	D_{los}	Pr (dBm)	PT (dBm)	Gt (dBi)	Gr (dBi)	PL 2-ray (dB)	Margin (dBm)	Threshold (dBm)	SNR (dB)
Wi-Fi	100	-75.802	27	3	3	93.802	15	-90.89	15.088
BT	100	-95.802	7	3	3	93.802	15	-103.9	8.0979
Wi-Fi	25	-51.72	27	3	3	69.72	15	-90.89	39.17
BT	25	-71.72	7	3	3	69.72	15	-103.9	32.18

Figure 6 shows the two-ray model estimated SNR over the range of distances in the simulation. The following section uses the SNR estimates to compute direction-of-arrival angle errors.

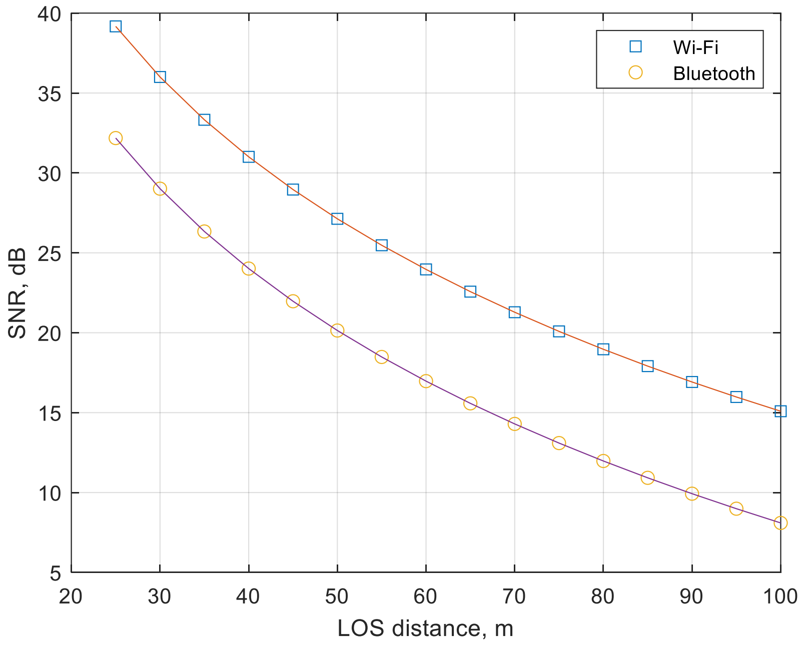


Figure 6 Outdoor multipath scenario. 1) Direct path; 2) Ground-bounce; 3) Reflected; 4) Diffracted

3.4 Compute angle measurement standard deviation for the computed SNR

Techniques for RF direction finding amplitude techniques compare the received signal strength to estimate the Angle-of-Arrival (AOA), as shown in Figure 7A. Phase techniques estimate the delay between antennas, as shown in Figure 7B. Phase techniques require at least one pair of coherent receivers.

The performance of both the phase and amplitude techniques depends on the received Signal-to-Noise (SNR) ratio estimated in the previous section.

The monopulse amplitude error also depends on the beamwidth and slope of the difference beam. The estimated beamwidth of the proposed disk-and-rod depends on its length. The antenna creates a pencil beam in azimuth and elevation which a Gaussian beam can approximate. The slope of a Gaussian pencil beam at the 3-dB point is km=1.386.

The phase interferometer accuracy is proportional to the baseline separation distance between antennas assuming direction ambiguities can be managed. Due to limited space, the deployment of RF-SAS on mobile vehicles limits both the monopulse and the interferometer approach.

Table 4. RF-SAS Physical Layer Location-Related Measurements

Detection only, Pd=Probability of detect, Pfa=Probability of false alarm	Amplitude-based DOA	Phase-based DOA	Time-of-Arrival (TOA or TDOA)
Only detects the presence of a signal, not direction or location	Monopulse ratio between sum and difference beams gives a bearing estimate	Interferometric phase gives a bearing estimate	Time measurements taken at two or more stations. Derivative effects exploit Doppler effects.
Depends on system sensitivity, signal strength, environment, and noise	Bearing error depends on signal, environment, and noise	Bearing error depends on signal, environment, and noise	Measurement error depends on signal, environment, and noise properties

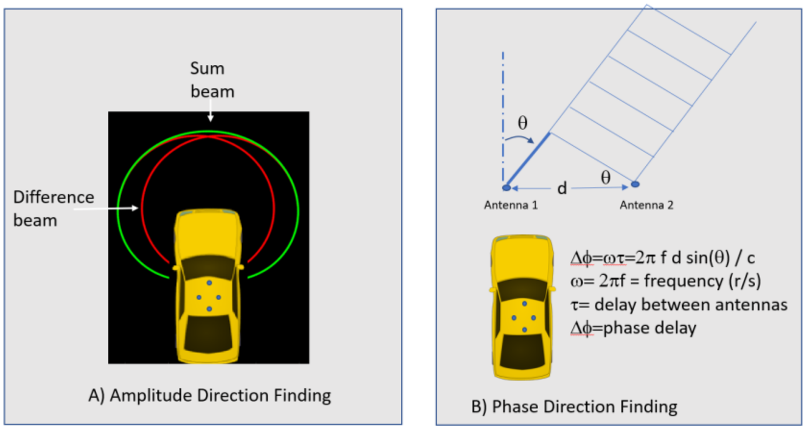


Figure 7 A) Direction finding by comparing RF amplitudes; B) by comparing RF received signal phases.

The estimated SNR along with system parameters of antenna spacing, D=0.25 meters, radio frequency wavelength λ=0.12 meters to estimate the angular Root-Mean-Square (RMS) error of the phase-based interferometer is shown in Table 5.

Table 5. Interferometer Phase and Monopulse Amplitude Angle-of-Arrival (AOA) Estimation Error

	Interferometer Angle Error	**Monopulse Angle Error**
	$\sigma_{phase} = \frac{1}{\frac{D}{\lambda}\cos(\theta)\sqrt{SNR}}$	$\sigma_{amp} = \frac{\beta}{k_m\sqrt{SNR}}$
RMS (deg) @ SNR=27 dB	σ_{phase} =1.2285°	σ_{amp}=1.4503°

The monopulse amplitude technique takes advantage of Received Signal Strength Indicator (RSSI) measurements. RSSI is used by Wi-Fi, Bluetooth, and Cellular systems to measure the quality of the received signal. This enables the AOA to be estimated and associated with specific emitter MAC addresses using existing receivers to decode the Wi-Fi, Bluetooth, or Cellular protocol. The association of angular measurements using some digital identifier is necessary for triangulation to avoid false ambiguity ghosting of intersecting bearing rays when multiple emitters are detected. The MAC address privacy of the detected emitter is protected using SHA-256 with a secure, time-varying nonce, as described in the next section.

The RSSI values must be measured on both channels simultaneously to avoid amplitude channel imbalance. RSSI-reported values by Wi-Fi receivers in monitor mode are quantized, which introduces additional angle error. The calibration response of the monopulse function is computed and stored using a calibration factor K_D as described by Gomez et al. [18] ,

$$\boldsymbol{\psi_D(\theta)} = \frac{\boldsymbol{\Delta_{RSSI}(\theta)}}{\boldsymbol{\Sigma_{RSSI}(\theta)}} = \frac{\boldsymbol{RSSI_1(\theta) - K_D\, RSSI_2(\theta)}}{\boldsymbol{RSSI_1(\theta) + K_D\, RSSI_2(\theta)}} \tag{5}$$

The Direction-of-Arrival (DoA) is estimated using a measured monopulse ratio obtained from the RSSI readings on each antenna,

$$\boldsymbol{\psi_{RSSI}} = \frac{\boldsymbol{\Delta_{RSSI}(\theta)}}{\boldsymbol{\Sigma_{RSSI}(\theta)}} = \frac{\boldsymbol{RSSI_1 - K_D RSSI_2}}{\boldsymbol{RSSI_1 + K_D RSSI_2}} \tag{6}$$

Numerical search to minimize the absolute difference between the measured $\boldsymbol{\psi_{RSSI}}$ and pre-stored calibration table values of $\boldsymbol{\psi_D(\theta)}$ gives an estimate of the DoA.

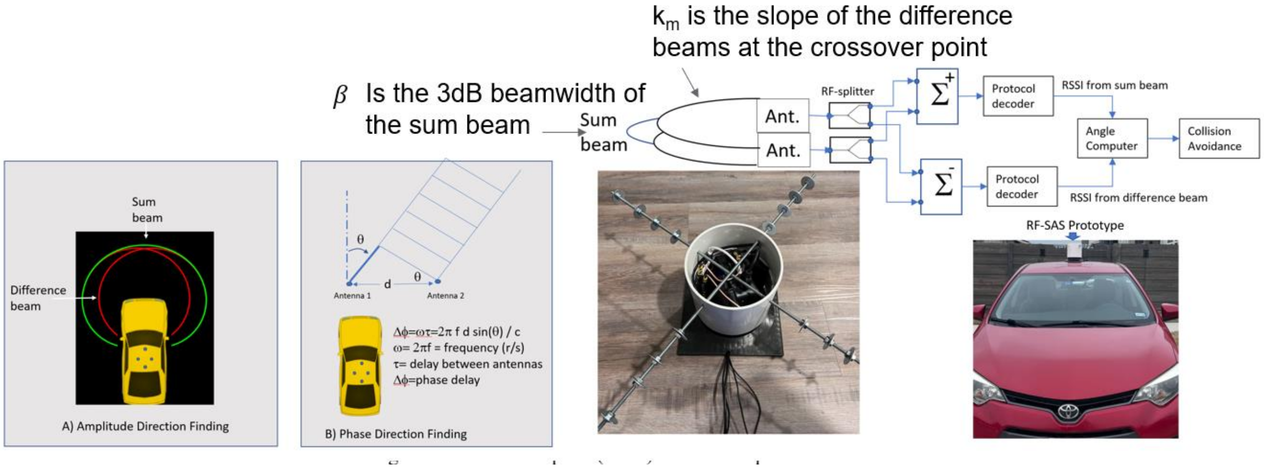


Figure 8 Left) RSSI Monopulse DF technique; Middle) Prototype under development; Right) Prototype RF-SAS Deployment

Figure 8 shows an RSSI monopulse ratio prototype direction finding (DF) prototype.

3.5 Use the noisy angles to compute the estimated location and error covariance.

Figure 9 shows an approach described by Torreri[19] to linearize the bearings-only triangulation problem to estimate the location of the emitter and its error covariance.

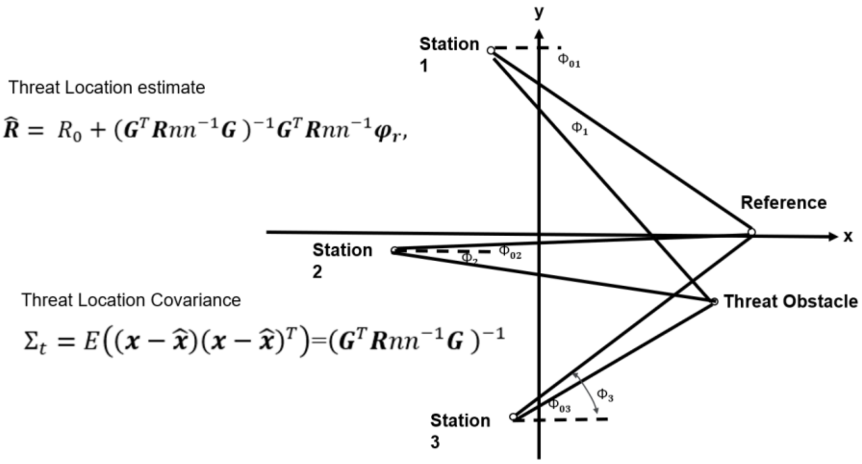

Threat Location estimate
$\widehat{\boldsymbol{R}} = R_0 + (\boldsymbol{G}^T \boldsymbol{R}nn^{-1}\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{R}nn^{-1}\boldsymbol{\varphi_r},$
Threat Location Covariance
$\Sigma_t = E\big((\boldsymbol{x}-\widehat{\boldsymbol{x}})(\boldsymbol{x}-\widehat{\boldsymbol{x}})^T\big) = (\boldsymbol{G}^T\boldsymbol{R}nn^{-1}\boldsymbol{G})^{-1}$
y
x
Station 1
Station 2
Station 3
Reference
Threat Obstacle
Φ_{01}
Φ_1
Φ_2
Φ_{02}
Φ_3
Φ_{03}

Figure 9 Linearization of triangulation gives location and covariance error estimate of collision threat emitter.

A reference point is established $R_o=(x_o,y_d)$ to linearize around. We choose the center of the intersection as the reference point. The reference quantities in the following equation are computed based on the relative location of the reference point to the N receiving stations,

$$\sin(\phi_{0i}) = \frac{y_0 - y_i}{D_{0i}};\ \cos(\phi_{0i}) = \frac{x_0 - x_i}{D_{0i}}\ \ i = 1,2,\dots N\ , \tag{7}$$

where,

$$D_{0i} = \sqrt{[(y_0 - y_i)^2 + (x_0 - x_i)^2]}\ . \tag{8}$$

The estimated location of the emitter $\hat{R}$,

$$\hat{\boldsymbol{R}} = R_0 + (\boldsymbol{G}^T \boldsymbol{R}nn^{-1} \boldsymbol{G}\ )^{-1} \boldsymbol{G}^T \boldsymbol{R}nn^{-1} \boldsymbol{\varphi_r}, \tag{9}$$

where *Rnn* is taken to be a diagonal matrix of angular error variance at each of the receiving stations estimated from the previous section.

$$G = \begin{bmatrix} -\sin(\phi_{01}) & -\cos(\phi_{01}) \\ \dots & \dots \\ -\sin(\phi_{0N}) & -\cos(\phi_{0N}) \end{bmatrix}\quad . \tag{10}$$

As depicted in Figure 9, the ith component of $\boldsymbol{\varphi_r}$ is the difference between the measured angle and the angle to the reference point at the ith receiver station,

$$\emptyset_{ri} = \emptyset_i - tan^{-1}\left(\frac{y_0 - y_i}{x_0 - x_i}\right) \tag{11}$$

The error covariance of the linearized estimate is given by,

$$\Sigma_t = E\big((\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^T\big) = (\boldsymbol{G}^T \boldsymbol{R}nn^{-1} \boldsymbol{G}\ )^{-1} \tag{12}$$

The elements of Σ_t are extracted by direct calculation as,

$$\sigma_1^2 = E[((\hat{x} - x)^2)] = \frac{\mu}{\mu\lambda - \upsilon^2} \qquad , \tag{13}$$

$$\sigma_2^2 = E[((\hat{y} - y)^2)] = \frac{\mu}{\mu\lambda - \upsilon^2} \qquad , \tag{14}$$

$$\sigma_{12} = E[((\hat{x} - x)^2)((\hat{y} - y)^2)] = \frac{\upsilon}{\mu\lambda - \upsilon^2} \qquad . \tag{15}$$

Where,

$$\mu = \sum_{i=1}^{N} \frac{cos^2\ \phi_{0i}}{D_{0i}^2 \sigma_{\phi i}^2} \qquad , \tag{16}$$

$$\lambda = \sum_{i=1}^{N} \frac{sin^2\ \phi_{0i}}{D_{0i}^2 \sigma_{\varphi i}^2} \qquad , \tag{17}$$

and,

$$\upsilon = \sum_{i=1}^{N} \frac{sin\ \phi_{0i} cos\ \phi_{0i}}{D_{0i}^2 \sigma_{\phi i}^2} \qquad . \tag{18}$$

The estimated error covariance matrix of the collision threat obstacle, Σ_t and the approaching vehicle, Σ_v are used in the next section to compute the probability of collision.

3.6 Use Error covariance to compute the Probability of Collision

The error covariance of the vehicle, Σ_v, is typically much less than the collision object due to the vehicle's onboard GPS and inertial system. We arbitrarily fix the vehicle estimated position standard deviation to be 3 meters in the direction of motion and 1 meter in the cross-axis to focus on the parameters of the threat object.

Lambert[20] et al. give a Monte Carlo (MC) algorithm to estimate the probability of collision that includes the shape of the objects to solve the problem of underestimating the collision probability of point distributions that do not include the size of the objects. A summary of Lambert algorithm 1 is shown below.

Algorithm 1: Probability of collision between the vehicle (v) and threat object (t)

```
1. $P_{collision(v,t)} \leftarrow 0$
2. *For j←1 to N do*
   $x_v \leftarrow$ randn($\hat{x}_v, \Sigma_v$)
   $x_t \leftarrow$ randn ($\hat{x}_t, \Sigma_t$)
   ***If*** 2D area is occupied by the vehicle, $A_v$ intersects the area occupied threat $A_t$:
           $P_{collision(v,t)} \leftarrow P_{collision(v,t)} + 1$
   ***End if***
3. End for
4. $P_{collision(v,t)} \leftarrow P_{collision(v,t)} /N$
```

3.7 Compute impact speed to compute risk

Rosen and Sander[21] estimate the risk of pedestrian fatality as a function of vehicle impact speed measured in km/h, as,

$$P_f(v) = \frac{1}{1+e^{(6.9-0.090v)}}. \qquad (19)$$

Figure 10 left shows how their risk nonlinearly increases with vehicle impact speed measured in m/s.

Impact Speed

$$v_v = \sqrt{u_v^2 - 2a_b D_{vt}}$$

Rosen Fatality Risk

$$P_f(v) = \frac{1}{1+e^{(6.9-0.090v)}}.$$

Our Collision Risk

$$R = \ P_{\text{collision}}(\text{v},\text{t})\ P_f(v).$$

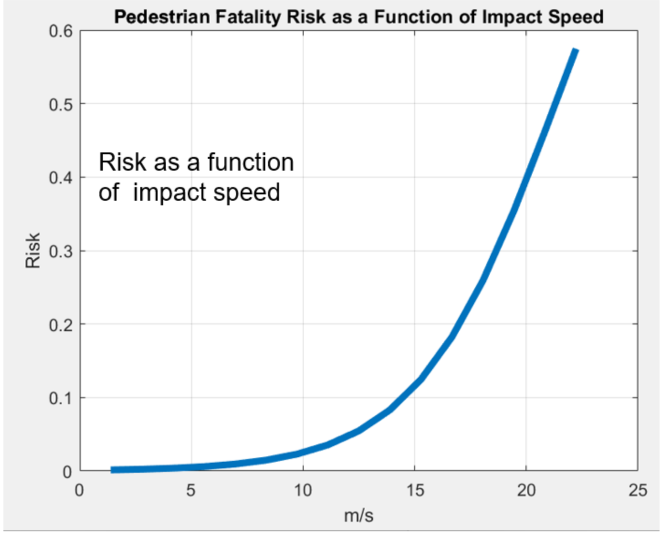


Figure 10 Right: Pedestrian fatality risk as a function of impact speed in m/s (adapted from Rosen[21])

After the vehicle applies the braking action, the vehicle comes to a stop at a braking distance of $D_{stop}= -\frac{u_v^2}{2a_b}$. If the collision threat is less than the stopping distance, the impact speed at the center of the intersection is,

$$v_v = \sqrt{u_v^2 - 2a_b x_b} \qquad (20)$$

Where x_b is the distance to the object, and a_b is the deceleration. We use Rosen's equation combined with Lambert's to define collision risk, with R as the probability of collision multiplied by the risk of fatality,

$$R = \ \mathrm{P}_{\mathrm{collision}}(\mathrm{v},\mathrm{t})\, P_f(v). \tag{21}$$

Section 4 below uses reinforcement learning to minimize the collision risk, *R.*

3.8 Privacy and cybersecurity concerns

IEEE Std 1609.2 specifies security mechanisms that may be used to provide confidentiality, authenticity, integrity, non-repudiation, replay protection, and relevance checking. However, the security mechanisms of IEEE Std 1609.2 do not on their own provide privacy. We propose the following technique to protect the MAC address of detected wireless energy. The detected MAC is protected using SHA-256 with a time-varying, securely distributed nonce as part of the overall safety message sent by the RSU to the OBU, including the channel number, measured angle, and date timestamp.

The OBU RF-SAS tunes to the commanded channel and makes an angle measurement. It hashes its detected MAC address using a nonce that was previously distributed over the one-to-one encrypted channel. The two MAC hashes are compared to ensure the two measured lines of bearing should be used in the triangulation calculation.

In this way, the observed MAC addresses are never sent over the air without being hashed using the secure nonce. Hashing the observed MACs with the secure nonce raises the bar to protect against eavesdroppers from building activity patterns o because the hash is always changing. Eavesdroppers would need to register with the system, receive the secure nonce, and detect and measure their own angle to a transmitter to record estimated observed MAC locations.

Low latency of message reception is paramount. Collision avoidance safety applications are particularly vulnerable to Denial-of-Service (DoS) attacks. DoS can also arise from spectrum congestion.

Regardless of the source of the DoS delay, it can be monitored so that alternative measures can be taken. One alternative measure is to use an out-of-band signaling mechanism such as LTE-V2X. IEEE 1609.3 Standard for Wireless Access in Vehicular Environments (WAVE) Networking Services describes a variety of services distributed across multiple 802.11 10 MHz channels[22]. Appendix M of 1609.3 shows how these services can be provided when a single 20-MHz channel of LTE-V2X is the underlying communications technology. The following section shows the importance of low latency in reducing the probability of collision and impact speed.

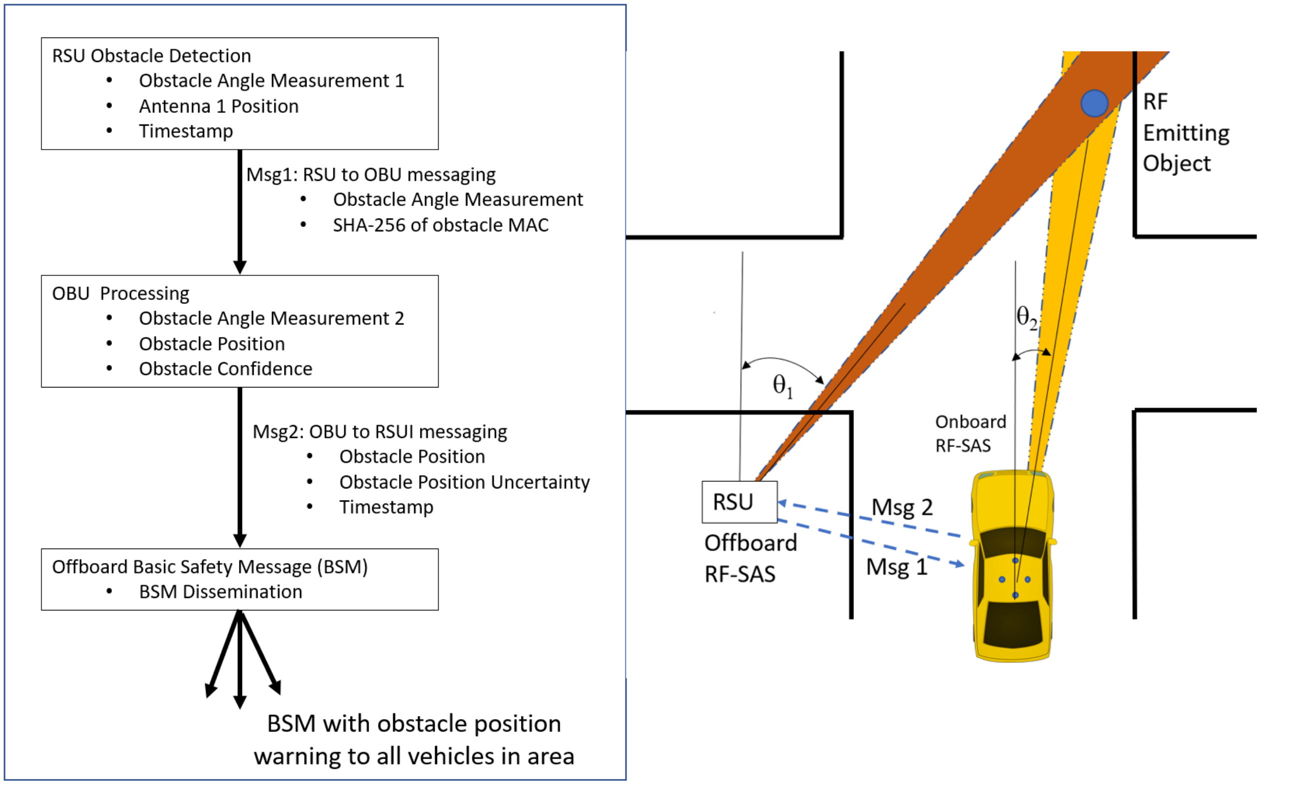
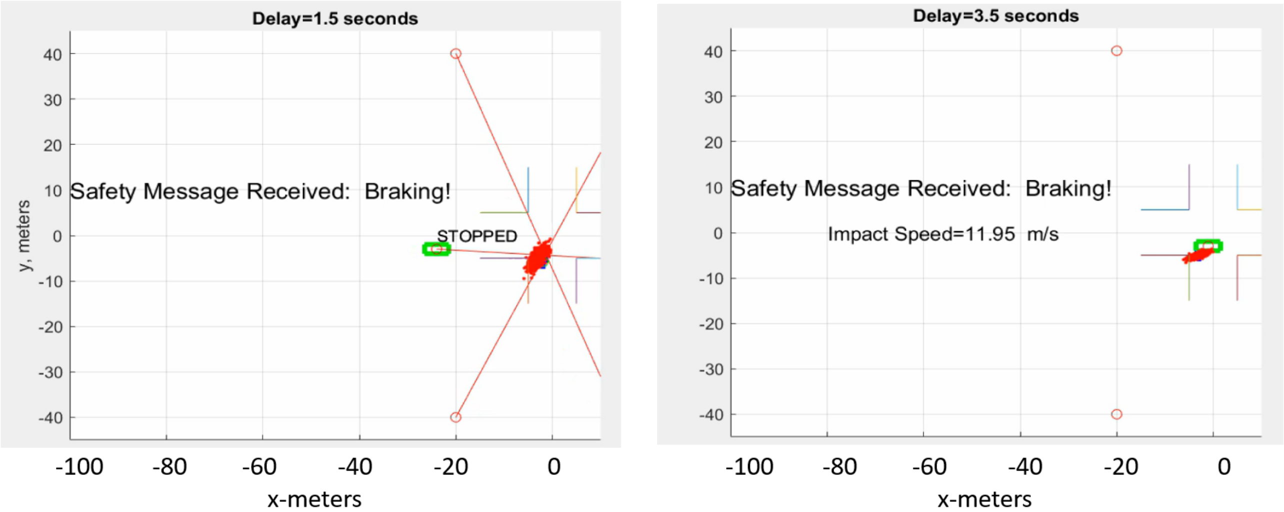

RSU Obstacle Detection
Obstacle Angle Measurement 1
Antenna 1 Position
Timestamp
Msg1: RSU to OBU messaging
Obstacle Angle Measurement
SHA-256 of obstacle MAC
OBU Processing
Obstacle Angle Measurement 2
Obstacle Position
Obstacle Confidence
Msg2: OBU to RSUI messaging
Obstacle Position
Obstacle Position Uncertainty
Timestamp
Offboard Basic Safety Message (BSM)
BSM Dissemination
BSM with obstacle position warning to all vehicles in area
RF Emitting Object
θ1
θ2
Onboard RF-SAS
RSU
Offboard RF-SAS
Msg 2
Msg 1

Figure 11 Offboard unit sends angle measurement for onboard obstacle location estimation.

3.9 Results

Figure 11 shows the effect of a reaction delay of 1.5 seconds versus 3.5 seconds. For both scenarios, the initial speed is 20 meters per second. The left panel shows early warning enables the vehicle to come to a complete stop. The right panel shows late warning results in a probability of collision of 0.43 and an impact speed of 12.3 meters-per-second. The risk of fatality at the reduced speed is 0.052 compared to the risk of fatality at the initial speed of 20 meters-per-second of 0.40. The reduction 87% reduction in fatality risk for a 38% reduction in speed shows the value of applying the braking action even if a collision cannot be avoided.



3.10

Figure 12 MATLAB simulation of RF-SAS system parameters. Left: delay of 1.5 seconds; Right: delay of 3.5 seconds results in a collision with high impact speed.

The Probability of Collision multiplied by the fatality risk is used in the next section to train a reinforcement learning agent.

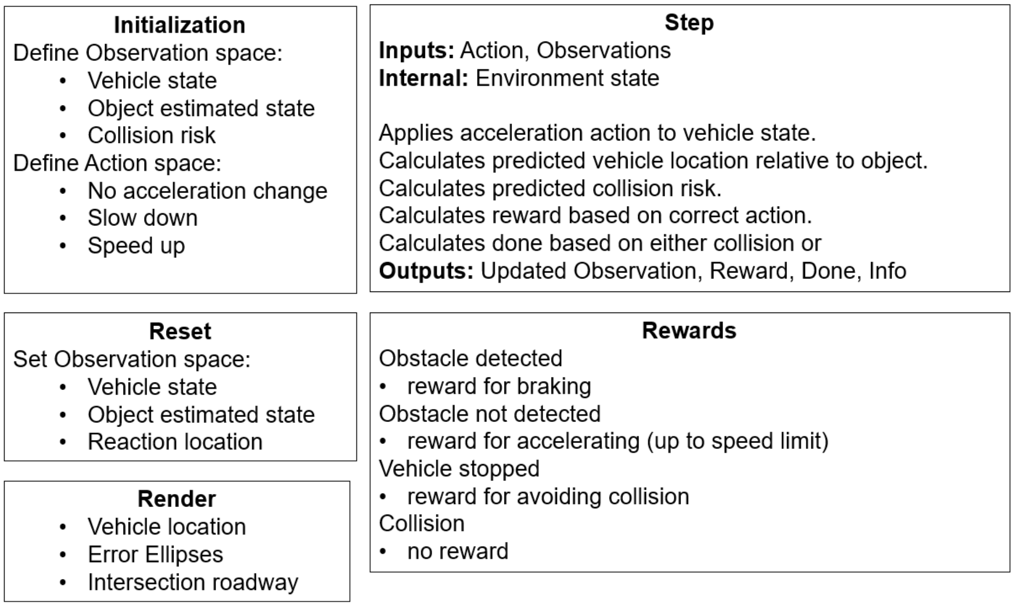
4. DEVELOPMENT OF RF-SAS GYM MODULE

The information gained from the high-level detail modeling of the operation environment is applied to develop a Python gym RF-SAS module shown in Figure 13. The gym environment initialization function defines the min and max range of the observation space using the spaces.Box function. The observations include the vehicle state and the estimated risk, defined as the probability of collision multiplied by the fatality risk based on the predicted impact speed. The 3-dimensional action space is defined using spaces.Discrete. The three actions are to 1) set acceleration to zero, 2) set deceleration to -4.7 m/s^2; or 3) speed up by setting acceleration to 2 m/s^2. The environment clamps the maximum velocity to 20 m/s^2 and prevents deceleration from resulting in a backward motion.

The step function updates the state of the vehicle using the input action and returns a new observation. The step function returns a reward of +1 if the action is braking when the estimated risk is greater than a threshold. A positive reward is also returned if the risk is less than the threshold and action is to accelerate. The step function returns the done flag as true when the vehicle either reaches the intersection or the vehicle comes to a stop.

The reset function resets the state of the vehicle back to the beginning and sets the reaction location. The reaction location is where the vehicle is alerted to the presence of the collision object near the center of the intersection. When the reaction location is near the center of the location, there is no time for the vehicle to slow down. When the reaction location is near the vehicle's starting location, there is the sufficient distance for the braking action to bring the vehicle to a stop. Intermediate locations result in slowing which reduces the risk. The reaction location depends on reaction delay and on the buildings blocking the intersection line-of-sight of cross-traffic.

The RF-SAS render function uses Pygame to draw the location of the vehicle and the collision object relative to the intersection. It applies an oriented ellipse around each one to depict the state covariance matrix.



Initialization

Define Observation space:
- Vehicle state
- Object estimated state
- Collision risk

Define Action space:
- No acceleration change
- Slow down
- Speed up

Step

Inputs: Action, Observations
Internal: Environment state

Applies acceleration action to vehicle state.
Calculates predicted vehicle location relative to object.
Calculates predicted collision risk.
Calculates reward based on correct action.
Calculates done based on either collision or
Outputs: Updated Observation, Reward, Done, Info

Reset

Set Observation space:
- Vehicle state
- Object estimated state
- Reaction location

Render
- Vehicle location
- Error Ellipses
- Intersection roadway

Rewards

Obstacle detected
- reward for braking

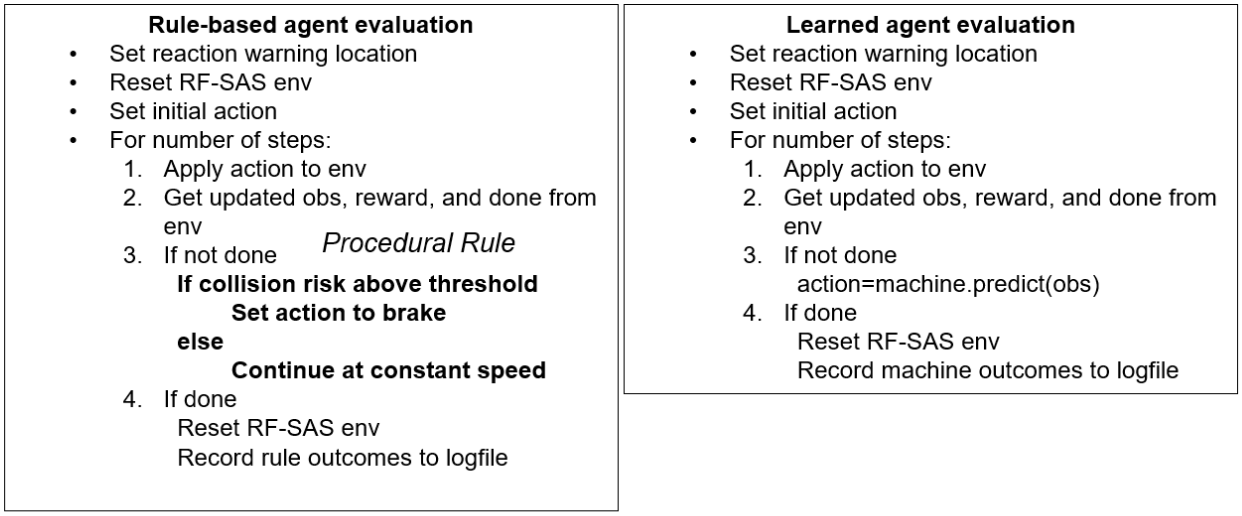
Obstacle not detected
- reward for accelerating (up to speed limit)

Vehicle stopped
- reward for avoiding collision

Collision
- no reward

4.1 Figure 13 Python Gym RF-SAS environment

The RF-SAS agent is trained using Stable Baselines 3, Proximal Policy Optimization (PPO) [23] using the RF-SAS environment with default parameters of policy=MlpPolicy, seed=0, batch_size=64, ent_coef=0.0, learning_rate=0.0003, n_epochs=10, n_steps=64. Once trained, the agent is evaluated against a rule-based procedure that sets the action to braking whenever the estimated collision risk is above a threshold, as shown in the figure below. The right side of the figure shows how the action is obtained from the trained agent.



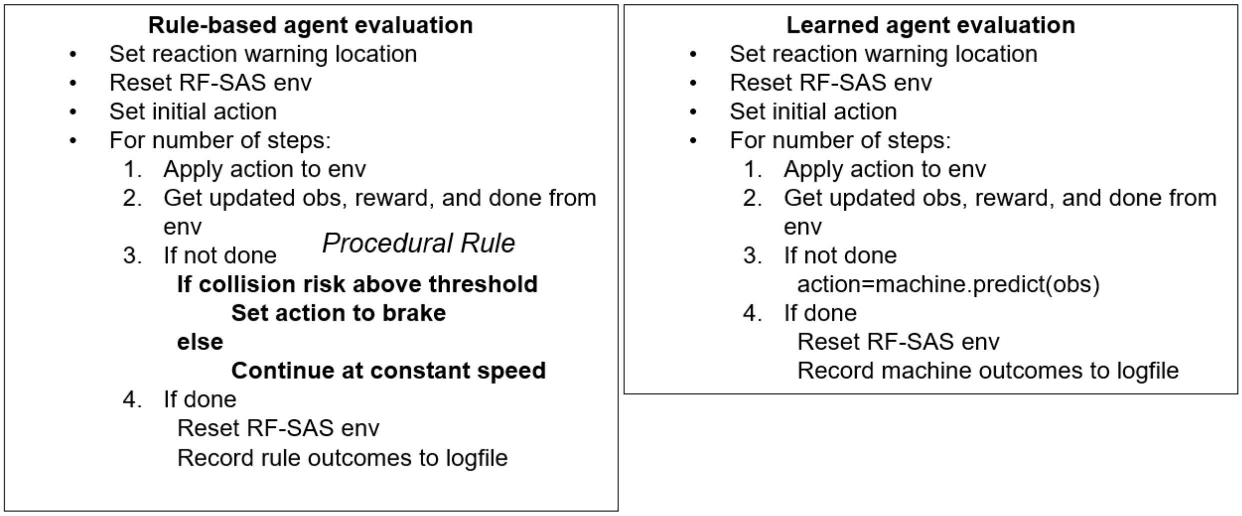
Rule-based agent evaluation

```
• Set reaction warning location
• Reset RF-SAS env
• Set initial action
• For number of steps:
    1. Apply action to env
    2. Get updated obs, reward, and done from
       env
    3. If not done
         If collision risk above threshold
              Set action to brake
         else
              Continue at constant speed
    4. If done
         Reset RF-SAS env
         Record rule outcomes to logfile
```

Learned agent evaluation

```
• Set reaction warning location
• Reset RF-SAS env
• Set initial action
• For number of steps:
    1. Apply action to env
    2. Get updated obs, reward, and done from
       env
    3. If not done
         action=machine.predict(obs)
    4. If done
         Reset RF-SAS env
         Record machine outcomes to logfile
```

Figure 14. The right side of the figure shows how the action is obtained from the trained agent.



```
Rule-based agent evaluation
• Set reaction warning location
• Reset RF-SAS env
• Set initial action
• For number of steps:
    1. Apply action to env
    2. Get updated obs, reward, and done from
       env
    3. If not done            Procedural Rule
         If collision risk above threshold
             Set action to brake
         else
             Continue at constant speed
    4. If done
         Reset RF-SAS env
         Record rule outcomes to logfile
```

```
Learned agent evaluation
• Set reaction warning location
• Reset RF-SAS env
• Set initial action
• For number of steps:
    1. Apply action to env
    2. Get updated obs, reward, and done from
       env
    3. If not done
         action=machine.predict(obs)
    4. If done
         Reset RF-SAS env
         Record machine outcomes to logfile
```

4.2 Figure 14 Rule-based versus learned actions

Table 6 compares the rule-based to machine learning outcomes. The rule-based outcomes fail when there is not enough warning. The machine learning outcomes, in effect, learn to slow down near the intersection, regardless of the presence of a warning. Both approaches reduce the collision risk by applying braking action to reduce impact speed. The initial vehicle speed was 20 m/s for all simulations. The RL agent learns to apply some braking as it approaches the intersection regardless of detecting a collision threat. One interpretation of this result is that autonomous vehicles will learn to slow down near intersections with historical RF activity.

Table 6. Comparison of rule-based to RL collision avoidance

	Learned Braking Policy			Procedural Braking Policy			
Reaction Distance (m)	Probability of Collision	Fatality Risk	Risk	Probability of Collision	Fatality Risk	Risk	Learned Risk Reduction (%)
40	0.00125	0.001007	1.26E-06	0.010	0.001007	1.01E-05	800%
50	0.0446875	0.001007	4.50E-05	0.355	0.001007	0.000358	795%
60	0.053125	0.00173	0.000354	0.415	0.006773	0.002811	795%
70	0.0965625	0.004681	0.001622	0.415	0.030318	0.012582	776%
80	0.0965625	0.008746	0.003309	0.415	0.062747	0.02604	787%
90	0.13944444	0.071723	0.029488	0.415	0.234801	0.097442	330%
100	0.13944444	0.089713	0.036953	0.415	0.396517	0.164554	445%

5. FUTURE WORK

Southern Methodist University Darwin Deason Institute for Cybersecurity (SMU-DDI) has recently established the Cybersecurity Autonomy Range (CAR) research facility[24]. Future work intends to address various aspects of autonomy security and information processing.

6. SUMMARY

The components of the RF-SAS module discussed above are:

- Compute SNR at receivers -> Two-ray path loss model
- Compute bearing angle standard deviation for the computed SNR -> monopulse RSSI[18]
- Use the noisy angles to compute estimated location and error covariance -> linearized least-squares of Torreri[19]

- Error covariance and object sizes to compute Probability of Collision, Pcoll-> Monte Carlo approach of Lambert[20]
- Compute collision risk -> impact speed-based fatality risk of Rosen[21] multiplied by Pcoll of Lambert[20]
- Privacy, and cybersecurity concerns-> secure nonce protects MAC, Out-of-band for DoS
- Python Gym RF-SAS environment -> Initial runs indicate ML can improve over procedural rules

The purpose of the RF-SAS module is to provide radio frequency observations to help the autonomous systems avoid collisions with entities emitting radio frequency energy but not reporting location to the autonomous system. In the future, more and more entities will cooperatively report location for collision avoidance. The RF-SAS approach is hoped to help reduce collision for older, non-integrated devices. This paper describes preliminary research which should not be used without independent verification. As with all safety-critical applications, extreme care and extensive testing must be done before deployment.

7. ACKNOWLEDGEMENTS

The authors would like to thank SMU-DDI staff Mark Bigham, Dr. Eric Larson, and Mark Bradbury for their review comments. We would also like to thank Raytheon for their review comments and suggestions.

REFERENCES

[1] Kabil, A., Rabieh, K., Kaleem, F., & Azer, M. A. (2022). Vehicle to Pedestrian Systems: Survey, Challenges and Recent Trends. IEEE Access, 10, 123981-123994.

[2] World Health Organization (WHO). Global Status Report on Road Safety 2018. Geneva, Switzerland: The World Health Organization (WHO); December 2018.

[3] U.S. Department of Transportation, Bureau of Transportation Statistics, Transportation, Statistics Annual Report 2022 (Washington, DC: 2022). https://doi.org/10.21949/1528354

[4] Wu, X., Miucic, R., Yang, S., Al-Stouhi, S., Misener, J., Bai, S., & Chan, W. H. (2014, September). Cars talk to phones: A DSRC based vehicle-pedestrian safety system. In 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall) (pp. 1-7). IEEE.

[5] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017, October). CARLA: An open urban driving simulator. In Conference on robot learning (pp. 1-16). PMLR.

[6] Amini, A., Wang, T., Gilitschenski, I., Schwarting, W., Liu, Z., Han, S., Karaman, S., & Rus, D. (2022). VISTA 2.0: An Open, Data-driven Simulator for Multimodal Sensing and Policy Learning for Autonomous Vehicles. 2022 International Conference on Robotics and Automation (ICRA). IEEE.

[7] Liang, J., Makoviychuk, V., Handa, A., Chentanez, N., Macklin, M., & Fox, D. (2018). GPU-Accelerated Robotic Simulation for Distributed Reinforcement Learning. Conference on Robot Learning (CoRL) 2018.

[8] Stanford Artificial Intelligence Laboratory et al. (2018). Robotic Operating System.

[9] Cai, P., Lee, Y., Luo, Y., & Hsu, D. (2020, May). Summit: A simulator for urban driving in massive mixed traffic. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4023-4029). IEEE.

[10] Haas, J. K. (2014). A history of the unity game engine.

[11] Amazon RoboMaker. https://docs.aws.amazon.com/robomaker/latest/dg/chapter-welcome.html.

[12] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke,M. Mozeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta ˇ et al., "Lgsvl simulator: A high fidelity simulator for autonomous driving," arXiv preprint arXiv:2005.03778, 2020.

[13] Li, Q., Peng, Z., Zhang, Q., Liu, C., & Zhou, B. Pgdrive: Procedural generation of driving environments for generalization.

[14] Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566) (Vol. 3, pp. 2149-2154). IEEE.

[15] Krajzewicz, D. (2010). Traffic simulation with SUMO–simulation of urban mobility. Fundamentals of traffic simulation, 269-293.

[16] Xu, R., Xiang, H., Xia, X., Han, X., Li, J., & Ma, J. (2022, May). Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In 2022 International Conference on Robotics and Automation (ICRA) (pp. 2583-2589). IEEE.

[17] Self, A. G., & Smith, B. G. (1985, July). Intercept time and its prediction. In IEE Proceedings F (Communications, Radar and Signal Processing) (Vol. 132, No. 4, pp. 215-220). IET Digital Library.

[18] Gómez-Tornero, J. L., Cañete-Rebenaque, D., López-Pastor, J. A., & Martínez-Sala, A. S. (2018). Hybrid analog-digital processing system for amplitude-monopulse RSSI-based MIMO WiFi direction-of-arrival estimation. IEEE Journal of Selected Topics in Signal Processing, 12(3), 529-540.

[19] Torrieri, D. J. (1984). Statistical theory of passive location systems. *IEEE transactions on Aerospace and Electronic Systems*, (2), 183-198.

[20] Lambert, A., Gruyer, D., Saint Pierre, G., & Ndjeng, A. N. (2008, October). Collision probability assessment for speed control. In 2008 11th International IEEE Conference on Intelligent Transportation Systems (pp. 1043-1048). IEEE.

[21] Rosén, Erik, and Ulrich Sander. "Pedestrian fatality risk as a function of car impact speed." Accident Analysis & Prevention 41, no. 3 (2009): 536-542.

[22] IEEE Intelligent Transportation Systems Committee. (2005). IEEE P1609. 3 Standard for Wireless Access in Vehicular Environments (WAVE)–Networking Services. Draft IEEE Standard. IEEE P1609, 3, D18.

[23] Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., & Dormann, N. (2019). Stable baselines3.j

[24] Young, Thornton, et al., (2022, October). SMU-DDI Cyber Autonomy Range. In 2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). IEEE.