



Multi-Mode Guidance for an Independent Multicopter Geofencing System

Mia N. Stevens* and Ella M. Atkins†

University of Michigan, Ann Arbor, Michigan, 48109, U.S.A.

As small unmanned aircraft systems (UAS) become more popular and the low altitude airspace becomes increasingly crowded, small UAS operations become increasingly complicated. One solution to organize airspace is with electronic geofencing systems. Current autopilots are capable of geofencing, but these geofences are implemented within the single-string autopilot that relies on the same hardware and software for all aspects of flight guidance, navigation, and control. This paper examines a geofence system with independence from the autopilot system. Specifically, different guidance modes defined to assure the UAS is safely kept within a designated test range. In simulation, the independent geofence system is shown to exhibit capabilities offered from geofence systems that are integrated into the autopilot plus an additional shared control guidance mode while supporting redundancy and resilience to nominal autopilot component failure.

Nomenclature

e	Error value passed to controller, $e = r' - \hat{x}$
LL	Local loiter
NAS	National Airspace System
r	Output from default autopilot guidance
\bar{r}	Output from geofence guidance
r'	Selected guidance command
RTL	Return to launch
u	Controller output
UAS	Unmanned aircraft system
W	Pilot input in the form of waypoints
x	Current aircraft state
\hat{x}	Current autopilot estimate of aircraft state
\bar{x}	Current geofence estimate of aircraft state
x_p	Pilot commands in a form that are not waypoints

I. Introduction

Geofencing holds promise to safely contain small unmanned aircraft systems (UAS) in approved low or high altitude operating volumes without the need for system-wide detect and avoid. The geofence itself, however, must be guaranteed to function properly, otherwise it cannot be certified or trusted. Emerging geofencing systems rely on single-string avionics (e.g., software add-ons to existing autopilots). If any of the components or software in that autopilot fail so will the geofencing system. This is particularly concerning for small UAS that may rely on open-source or uncertified avionics and software solutions. An alternative is

*PhD student, Robotics Program, University of Michigan, Ann Arbor, MI.

†Associate Professor, Aerospace Engineering Dept, University of Michigan, Ann Arbor, MI, Associate Fellow.

to separate the geofencing system from the nominal platform by creating an independent geofencing module with separate sensors, processor, and software, thus providing a redundant backup to the nominal system. Such a solution will add to the cost of the aircraft, but it may not be required in all applications as regulatory requirements for commercial, public, and hobby uses are still in flux.

Is it the responsibility of the airframe manufacturer and vendor, the autopilot team, or a third party to develop, certify, and assure geofencing capability? Some UAS companies are already taking steps to incorporate geofencing systems into their products. DJI currently offers geofencing capabilities for its newer lines of UAS (www.dji.com/flysafe/geo-system), and ArduPilot offers geofencing (ardupilot.com) with hardware such as Pixhawk¹ (ardupilot.org/plane/docs/common-autopilots.html). However, to the best of our knowledge, there is no geofencing system available today that addresses issues with single-string avionics and software. This paper investigates the capabilities of an independent geofencing system, extending our previous work,^{2,3} and specifically investigates a family of geofencing guidance modes that are compared and contrasted in simulation.

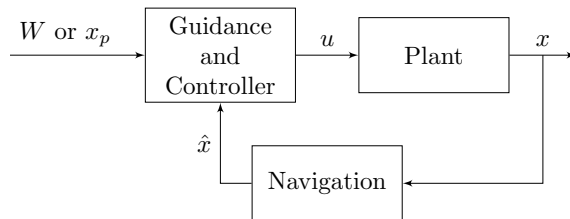


Figure 1. System level diagram of single-string geofence-autopilot.

A general single-string geofencing and autopilot system is depicted in Figure 1, where the plant and navigation blocks function as they would in any normal multicopter system. The guidance and controller blocks are changed by the addition of the geofence. Figure 2 is a diagram of the internals of the guidance and controller blocks in Figure 1. The inputs to the blocks are the current position estimate (\hat{x}) and the pilot input which can be in the form of waypoints, W , or other pilot commands, x_p . The current position estimate (\hat{x}) feeds into the block that checks for a geofence violation and acts as a switch between the default autopilot guidance and the geofence guidance. If no geofence violation is detected, the pilot commands are passed through the default guidance block. If a geofence violation is detected, the pilot commands are passed to the geofence guidance block, where it is modified to correct the geofence violation. The output of the selected guidance block is then combined with the current position and passed to the controller to calculate the output (u).

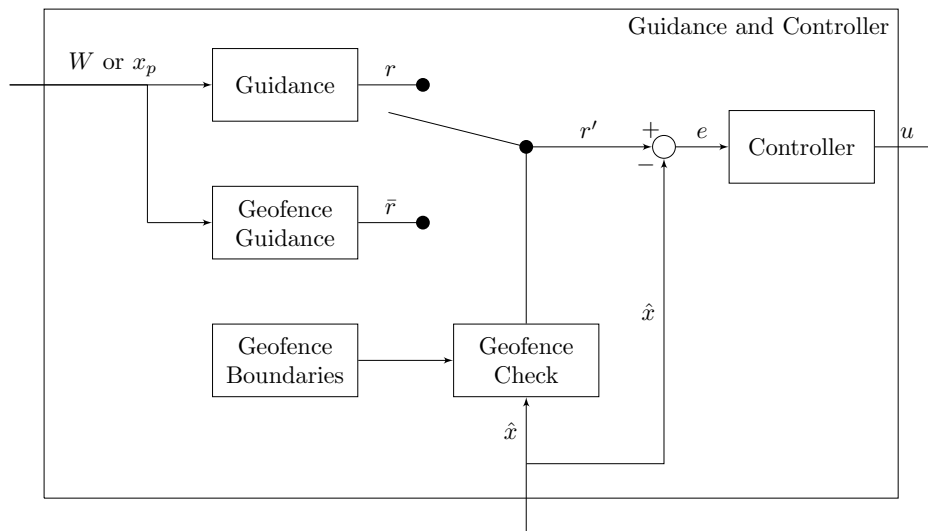


Figure 2. Diagram of single-string geofence-autopilot guidance and control block.

An independent geofence added to an autopilot is depicted in Figure 3, where the guidance, controller, plant, and navigation blocks are the respective components of any autopilot (with or without geofencing capabilities). The geofence navigation block denotes the geofence sensors and state estimation system. The geofence check block denotes the geofence violation check. The geofence guidance block modifies the pilot commands to avoid geofence boundary violation. These four left-most blocks of Figure 3 correspond to the blocks shown within the guidance and controller blocks in Figure 2. A Simulink implementation of this system is shown in Figure 5.

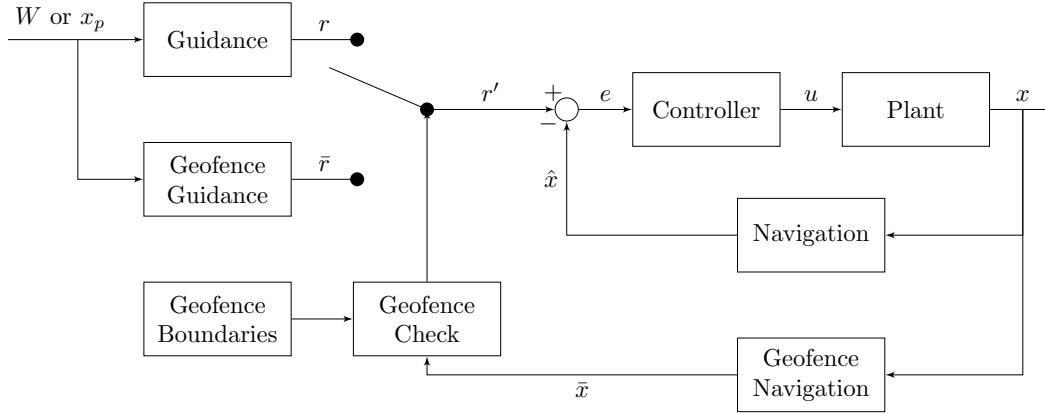


Figure 3. System level diagram of proposed independent geofence added to a generic autopilot.

This work will build on our previous work to develop and validate a multicopter keep-in geofencing system.² A series of simulations will be used to demonstrate and discuss three different geofence modes: shared control, return to launch (RTL), and local loiter (LL). When a geofence boundary is encountered while using the shared control geofence mode, the geofence system modifies the pilot commands to eliminate the elements of the commands that would result in a geofence boundary violation. When a geofence boundary is encountered while using the return to launch (RTL) geofence mode, the geofence system commands a return to the launch location (or some other user-defined waypoint). The local loiter (LL) geofence mode is similar to RTL, but when geofence boundary violation is imminent the aircraft is commanded to loiter a small distance from the local geofence boundary rather than returning to the launch point. This prevents potentially large excursions to RTL followed by a return trip to the intended operating area (just inside the fence).

In the next section (Section II), currently available geofence systems are reviewed to provide background for this work. Section III defines the proposed independent geofencing system and the three geofencing guidance modes. Section IV summarizes the implementation of the independent geofence system. Then, Section V presents simulation results from all three guidance modes, followed by discussions and conclusions in Section VI.

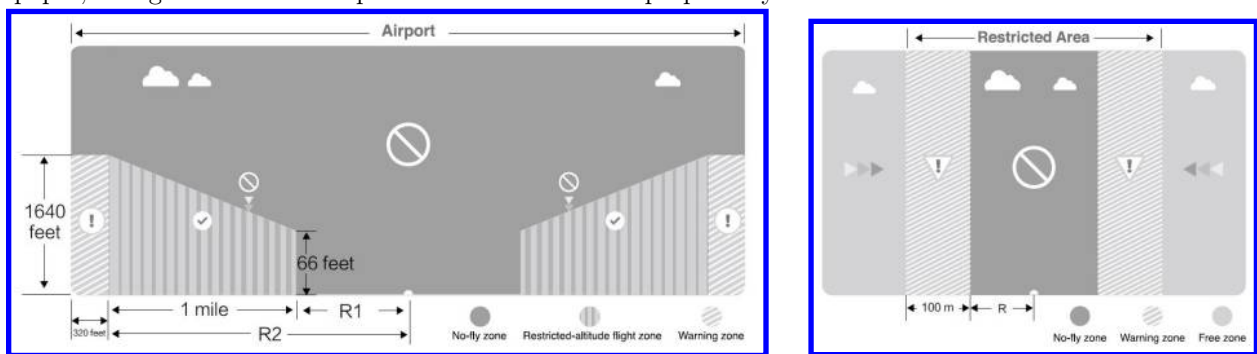
II. Geofencing Background

Geofencing is a promising method for the integration of small UAS into the existing airspace system. The idea of using geofencing for airspace safety has been introduced previously,³⁻⁶ and our previous work took steps to develop an initial hardware implementation of keep-in geofencing.² The two types of geofences considered in this paper are keep-out and keep-in. Keep-out geofencing prevents UAS from flying over sensitive and high-risk areas such as government buildings, military bases, airports, cities, and stadiums. Keep-in geofencing confines UAS operations to within the flight volume specified by the operator. Both types of geofencing are necessary for safe operation of UAS. While sUAS regulations are largely under flux, there are numerous proposals of how to ensure the safe operation of UAS and how to segregate low-altitude airspace. Other options that are being developed to ensure safe UAS operation include detect and avoid, assured containment, and redundant systems.^{7,8} Some proposals for small UAS integration segregate the airspace below the National Airspace System (NAS) based on lateral boundaries, such as neighborhood and park boundaries.⁹ Other proposals, including the Amazon Prime Air proposal, recommend dividing the

airspace based on speed of UAS travel and the communication capabilities of each aircraft which assumes even the lowest altitude airspace is free to occupy regardless of property owner preferences.^{10,11} Specific geofencing functions available in popular autopilot systems are described below.

A. DJI Geofencing

DJI currently offers a keep-in and a keep-out geofence system available for at least seven of their systems. For those aircraft, the functionality of the geofence system is determined by the flight mode of the aircraft. DJI aircraft have up to three flight modes: P-mode (Positioning), S-mode (Sport), and A-mode (Attitude). P-mode is the default flight mode, uses all available sensors, and applies both keep-out and keep-in geofences. S-mode is designed for high-performance flight and the geofence capabilities of this mode were not stated in the user manuals currently available from DJI. A-mode is the simplest flight mode, does not utilize GPS capabilities, and only enforces a keep-in altitude geofence. The keep-in geofence system for flights using P-mode supports a maximum altitude geofence (within the range of 20m to 500m) and a maximum radius geofence (within the range of 15m to 500m).^a The keep-in geofence causes the aircraft to stop before breaching the boundary.¹² This geofencing behavior is similar to the shared control mode defined in this paper, though certain DJI implementation details are proprietary.



(a) DJI Airport graphic. (dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual.en.v1.2-160328.pdf)

(b) DJI Restricted Area graphic. (dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual.en.v1.2-160328.pdf)

Figure 4. DJI keep-out geofence graphics.

The keep-out geofence is active during P-mode flight, which is the default flight mode. When the aircraft approaches a No Fly Zone while in P-mode, it will be stopped at the boundary. If the system is powered on within a No Fly Zone, the system will not be allowed to take-off. If the system is flown in A-mode (Attitude), the GPS system is not active, so the No Fly Zones are not enforced. If the system is flown in a No Fly Zone using A-mode, then is switched into P-mode, the aircraft is automatically landed followed by powering down of the motors. The DJI No Fly Zones include restricted areas such as borders and government property (Figure 4(b)) and select airports (Figure 4(a)). The airports have a default No Fly Zone radius set by the government, then an additional one mile radius beyond that with altitude restrictions to allow for the takeoff and descent of aircraft. The altitude restrictions begin at the edge of the No Fly Zone at 20 meters (65ft) then increases linearly at 15° to the edge of the one mile radius where the altitude limit is 500 meters (1640ft). If an aircraft is flown into an altitude-restricted zone while in A-mode, then switched into P-mode, the aircraft will descend until it is no longer in violation of the geofence, then the aircraft will descend to hover 15ft below the Restricted Altitude Zone limit.¹²

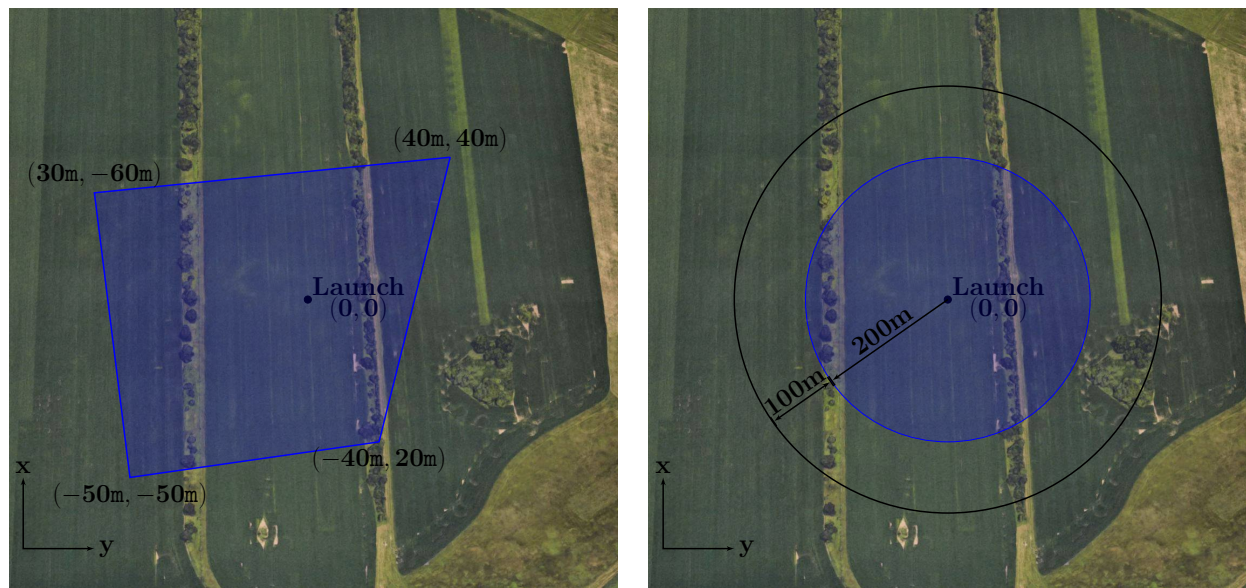
DJI maintains that the No Fly Zone list that they provide and enforce is strictly advisory and is not a complete list, and that the pilot is responsible for the legality of each flight (www.dji.com/flysafe/no-fly). The No Fly Zone list enforced by DJI updates whenever the system is connected to the internet and is enforced independent of whether an internet connection is available during flight. Airspace data, including temporary flight restrictions, for DJI is provided by AirMap, but "AirMap.io ... is not representative of what

^aAccepted ranges for DJI keep-in geofence were found within the DJI Mobile SDK documentation at: developer.dji.com/iframe/mobile-sdk-doc/android/reference/dji/sdk/FlightController/DJIFlightLimitation.html, developer.dji.com/iframe/mobile-sdk-doc/ios/Classes/DJIFlightLimitation.html.

data is used .. or how DJI has implemented that data. DJI may also add to the data provided by AirMap.” (www.dji.com/flysafe/geo-system/details)

B. Ardupilot

Ardupilot is one of the most widely utilized open source autopilots, with hardware options including Pixhawk (3dr.com/kb/pixhawk-system-components/) and Erle-Brain Linux Autopilot (ardupilot.org/plane/docs/common-autopilots.html). Ardupilot implements keep-in geofencing with a return to launch (or waypoint) guidance mode. The Ardupilot geofence does not currently implement keep-out geofencing. In contrast to the DJI system, the Ardupilot system only activates after the geofence boundary is breached, so a buffer area around the geofence is recommended to account for momentum. When the geofence is violated, the aircraft returns to the designated waypoint or rally point within the boundaries. A different keep-in geofence design is implemented in the Ardupilot Plane and Copter systems.



(a) Ardupilot Plane Geofence.

(b) Ardupilot Copter Geofence.

For the Plane system (Figure 5(a)), the lateral geofence is designated by a minimum of 4 and a maximum of 18 edge points, with the same first and last waypoint. A minimum and maximum altitude above ground level can be defined as well. If both geofence altitudes are set to zero, it is equivalent to not having an altitude geofence, and when the aircraft returns to the waypoint after a breach, the aircraft will simply maintain its current altitude (ardupilot.org/plane/docs/geofencing.html).

For the Copter system (Figure 5(b)), a maximum lateral radius and a maximum altitude can be set by the pilot. Upon geofence breach, the pilot can elect to either be notified or the vehicle will be ordered to RTL. If the pilot decides to override the RTL command after a geofence breach, the geofence system will set another geofence boundary 20m farther out than the original geofence boundary, where it will command an RTL. If the pilot overrides the RTL command after breaching the new geofence boundary, the geofence system will continue erecting boundaries at additional 20m increments until 100m farther than the original geofence boundary, at which point the vehicle will be commanded to land. This is the procedure for both lateral and altitude geofence boundaries (ardupilot.org/copter/docs/ac2_simple_geofence.html).

III. Independent Geofence System

The proposed geofence system is independent of the autopilot, so that it has its own software, processor, and sensors (see Figure 3). This system independence provides a layer of redundancy usually lacking in small UAS. The keep-in geofence is defined by the user. As geofencing becomes more common and a central information hub becomes available, the keep-in geofence could be defined based on the vehicle’s GPS location as the known property lines or designated flight areas. Similarly, the keep-out system needs to be defined

by a secure database to protect No Fly Zones from entrance by unauthorized vehicles. Access to No Fly Zones needs to be specified per flight or per pilot basis. DJI has taken steps to create a system of this type with their Advanced Unlocking system, which requires the pilot to *unlock* a zone with either a credit card or cell phone number (www.dji.com/flysafe/geo-system/details#unlocking). The DJI system creates a first level accountability system, but a more formal transparent system that incorporates all No Fly Zones is needed for the system to be considered complete. A geofence system with a complete No Fly Zone list would limit incursions into restricted airspace by uninformed pilots, removing some of the responsibility from the pilot. Our current implementation assumes that the geofence keep-in boundaries do not violate any keep-out geofence boundaries. Once a keep-out geofence database is available for integration into the system, the keep-in geofence boundaries can be checked against the database before flights begin.

In pursuit of a geofence system that pilots are likely to use and that can be compared/contrasted over time, we define three distinct operational modes for a geofence system: shared control, return to launch (RTL), and local loiter (LL). The shared control mode boundary response is similar to the DJI geofence boundary response: the command components that would result in a boundary breach are ignored while the other command components are used without modification. The RTL mode is similar to the behavior implemented by Ardupilot, where the vehicle returns to the launch position (or another waypoint) after a geofence boundary violation. The LL mode is a variation on the RTL mode; after a geofence boundary violation, the vehicle moves a set minimum distance from all geofence boundaries before returning control to the pilot. Each operating behavior or mode has distinct rules for enforcing the geofence boundaries, and certain modes may be better suited than others for a given flight objective.

In the current implementation of the independent geofence, the mode of operation of the geofence is held constant throughout the flight. As the system progresses, it might make sense to allow manual or automatic switching between modes based on sensor data accuracy, wind conditions, or other factors. Similarly, there is no geofence automatic landing function or kill switch decision. These functions could be added in future iterations to address more serious conditions that might arise, such as if the vehicle is not responding as expected to commands from the geofence.

A. Shared Control

When using shared control, the original pilot/autopilot commands for motion are modified to eliminate components of the commands that would result in a geofence boundary breach. For a hover capable UAS, the components of commands that would result in a fence breach are modified to be executed within the geofence boundaries, the limit conditions result in the system maintaining a hover state. Algorithm 1 summarizes the procedure for calculating the shared control commands for a system using position control. The advantage of this geofence mode is that the pilot's commands are minimally changed from the default commands. The disadvantage of shared control is that without a clear ground station indication that the geofence is active, the pilot could overcompensate for the geofence blocking a direction of travel, which might then make the UAS appear unresponsiveness to pilot commands.

Algorithm 1 Calculate shared control for position control and a rectangular geofence area

Ensure: $x_p \in bounds$

$x_p \leftarrow$ Position control pilot commands

$bounds \leftarrow$ Rectangular geofence corners

for all x_p **do**

if $x_p < \min bounds$ **then**

$x_p \leftarrow \min bounds$

else if $x_p > \max bounds$ **then**

$x_p \leftarrow \max bounds$

end if

end for

B. Return to Launch

An imminent geofence breach for a hover-capable vehicle with RTL engaged can transition to hover, then fly directly to the launch location, where it can hover until the pilot reasserts control of the system. The

advantages of the RTL geofence mode is that the activation of the geofence system is clear, and the aircraft is returned to a central location so assertion of control by the pilot can be simple. The disadvantage of this mode is that in large flight areas, a return to home could consume a significant amount of time and onboard energy. RTL mode impacts the default flight path enough to justify providing the pilot with the capability to turn the geofence system off and on to interrupt the command sequence.

C. Local Loiter

The local loiter (LL) geofence mode is designed to appear as though the vehicle is bouncing off of the geofence boundary. LL commands the vehicle to fly a minimum distance from all geofence boundaries before returning control to the pilot. For a hover capable vehicle, an LL response results in the vehicle halting, then flying perpendicular to the fence for a specified distance, then hovering until the pilot retakes control. The specified distance, d_{LL} , from the geofence boundary is currently set to one tenth the minimum distance from the home waypoint to the geofence boundary. This method for calculating this distance will become better defined based on further simulation and flight tests.

Within the simulation, each geofence boundary is defined as a straight line between two waypoints, $(x_{b,1}, y_{b,1})$ and $(x_{b,2}, y_{b,2})$. The geofence boundary that has been violated is whichever has the minimum distance, d , from the boundary to the current vehicle position. Once the violated boundary has been identified, the point on the boundary that is closest to the current aircraft position is calculated by Eq. 2.

$$d = \frac{|a * \hat{x} + b * \hat{y} + c|}{\sqrt{a^2 + b^2}} \quad (1)$$

$$(x_b, y_b) = \left(\frac{b(b * \hat{x} - a * \hat{y}) - ac}{a^2 + b^2}, \frac{a(-b * \hat{x} + a * \hat{y}) - bc}{a^2 + b^2} \right) \quad (2)$$

$$\text{where: } a = y_{b,2} - y_{b,1}$$

$$b = -x_{b,2} + x_{b,1}$$

$$c = x_{b,2} * y_{b,1} - y_{b,2} * x_{b,1}$$

Using (x_b, y_b) , we calculate the slope, m , and y-intercept, B , for the line perpendicular to the violated boundary. With these values, it is possible to find the local loiter waypoint, (x_{LL}, y_{LL}) , using Eq. 5 and 6.

$$m = \frac{-x_{b,2} + x_{b,1}}{y_{b,2} - y_{b,1}} \quad (3)$$

$$B = y_b - m * x_b \quad (4)$$

$$x_{LL} = x_b \pm \sqrt{\frac{d_{LL}}{1 + m^2}} \quad (5)$$

$$y_{LL} = m * x_{LL} + B \quad (6)$$

The advantages of LL mode are that it is clear when the geofence system takes control of the aircraft and the geofence control duration is relatively short compared to RTL mode. The disadvantage of LL mode is that if the pilot is having trouble controlling the aircraft, an unintentional loop could occur where the geofence moves the vehicle away from a boundary, only for the pilot to immediately fly at the boundary again, resulting in a cyclic behavior. This cyclic behavior would also occur if the aircraft was attempting to achieve a waypoint in violation of the geofence boundary.

IV. Implementation

As stated above, there are autopilot systems that already incorporate geofence capabilities directly into the system. However, to provide redundancy against sensor and control software failures, the geofence system must exist as a separate entity. To incorporate a separate geofence system, the system can be implemented either before or after the autopilot. By implementing the geofence before the autopilot, the geofence system must be able to replace the default guidance system and operate at the minimum input rate required by the

autopilot. By implementing the geofence after the autopilot, the geofence must be able to replace the default controller system and operate at the rate expected by the motors. We have focused thus far on implementing the geofencing system before the autopilot.

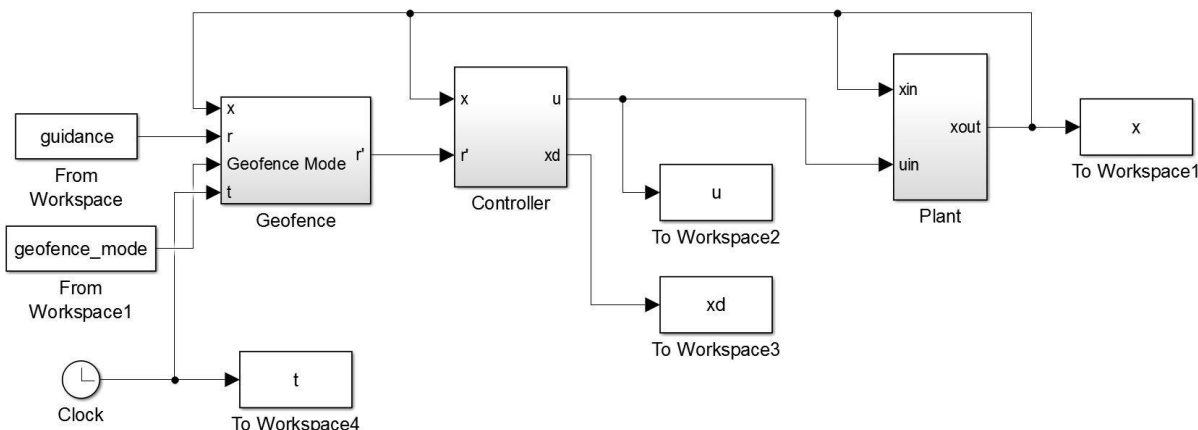


Figure 5. Simulink geofence implementation.

We have developed a basic quadrotor simulation in Matlab/Simulink. The simulation provides the opportunity to develop the geofence with complete knowledge of the system. Once the desired geofence behavior is demonstrated for the known case, the system can be modified to show that the geofence will also function on systems with incomplete knowledge of both the UAS and its environment. The Simulink model used to develop and demonstrate the independent geofence system is shown in Figure 5. The plant block of the model contains the dynamical model used in the simulation, which is based on the Beard model.¹³ The plant, controller, and guidance blocks can be replaced by the corresponding blocks of any multicopter system and the geofence block would still function properly.

A. Simulation Plant / State Update

The plant block of the simulation details the update of multicopter state in response to motor commands from the system controller. The first step is to convert the motor commands to the forces and torques that the motors will exert on the vehicle:

$$\begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 \\ \frac{\sqrt{2}}{2}lk_1 & -\frac{\sqrt{2}}{2}lk_1 & -\frac{\sqrt{2}}{2}lk_1 & \frac{\sqrt{2}}{2}lk_1 \\ \frac{\sqrt{2}}{2}lk_1 & \frac{\sqrt{2}}{2}lk_1 & -\frac{\sqrt{2}}{2}lk_1 & \frac{\sqrt{2}}{2}lk_1 \\ -k_2 & k_2 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} \delta_{f,l} \\ \delta_{f,r} \\ \delta_{b,r} \\ \delta_{b,l} \end{bmatrix} \quad (7)$$

where F is the total force exerted by the motors of the quadrotor, (M_x, M_y, M_z) are the torques exerted by the motors along the quadrotor's body axes, δ_i is the command sent to motor i , and (k_1, k_2, l) are constants that must be determined for each quadrotor. The equation above relates motor commands to forces and torques for quadrotors, but could easily be extended for hexacopters and octocopters.

Once the forces and torques acting on the vehicle are known, the rate of change of the state can be calculated using the following equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cos(\psi) & \sin(\phi) \sin(\theta) \cos(\psi) - \cos(\phi) \sin(\psi) & \cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi) \\ \cos(\theta) \sin(\psi) & \sin(\phi) \sin(\theta) \sin(\psi) + \cos(\phi) \cos(\psi) & \cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi) \\ \sin(\theta) & -\sin(\phi) \cos(\theta) & -\cos(\phi) \cos(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} r * v - q * w \\ p * w - r * u \\ q * u - p * v \end{bmatrix} + \begin{bmatrix} -g \sin(\theta) \\ g \cos(\theta) \sin(\phi) \\ g \cos(\theta) \cos(\phi) \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -F_b \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (J_y - J_z)/J_x * q * r \\ (J_z - J_x)/J_y * p * r \\ (J_x - J_y)/J_z * p * q \end{bmatrix} + \begin{bmatrix} M_x/J_x \\ M_y/J_y \\ M_z/J_z \end{bmatrix} \quad (11)$$

These rates are then integrated to update the state of the vehicle for a given time step. Equations 8, 9, 10, and 11 are the generic equations of motion derived by Beard.¹³

B. Simulation Guidance and Control

The guidance for the simulation is specified as a 5-dimensional set-point design and can operate in four different control modes. Only one control mode is utilized per simulation run, and the sequence of motion commands are generated prior to the execution of the simulation, indexed by execution time t . There is no confirmation that the previous command has been executed. Despite the entire motion command sequence being pre-planned, the system uses only the command designated for the current time step in order to simulate a real flight as closely as possible. The available guidance modes are as follows:

1. Angular Rate Control, $\mathcal{S}_1 = [t \ p_d \ q_d \ r_d \ \dot{z}_d]^T$

This guidance and control mode interprets the incoming command sequence as desired angular rates. This is a common mode for RC controlled vehicles. The first three terms of the command (p_d, q_d, r_d) are the desired roll rate, pitch rate, and yaw rate, and are passed straight to the controller. The fourth term of the command (\dot{z}_d) is the desired vertical velocity.

2. Angular Control, $\mathcal{S}_2 = [t \ \phi_d \ \theta_d \ \psi_d \ z_d]^T$

This guidance and control mode evaluates the incoming command sequence as desired roll, pitch, and yaw (ϕ_d, θ_d, ψ_d), and desired altitude (z_d).

3. Velocity Control, $\mathcal{S}_3 = [t \ \dot{x}_d \ \dot{y}_d \ \dot{z}_d \ \psi_d]^T$

This guidance and control mode interprets the command sequence as desired velocities in world coordinates ($\dot{x}_d, \dot{y}_d, \dot{z}_d$) and desired yaw angle (ψ_d).

4. Position Control, $\mathcal{S}_4 = [t \ x_d \ y_d \ z_d \ \psi_d]^T$

This guidance and control mode interprets the command sequence as a desired position in world coordinates (x_d, y_d, z_d) and a desired yaw angle (ψ_d). This control mode is similar to systems that operate using waypoint navigation or discrete path planning.

Regardless of which guidance mode is in use, the desired force (F) and angular rates (p_d, q_d, r_d) are passed from the guidance block to the controller block, then the controller outputs motor commands. The exact inputs and outputs of the controller and plant can be changed from the current setup to mimic a specific desired autopilot or vehicle format, or to introduce noise into the system.

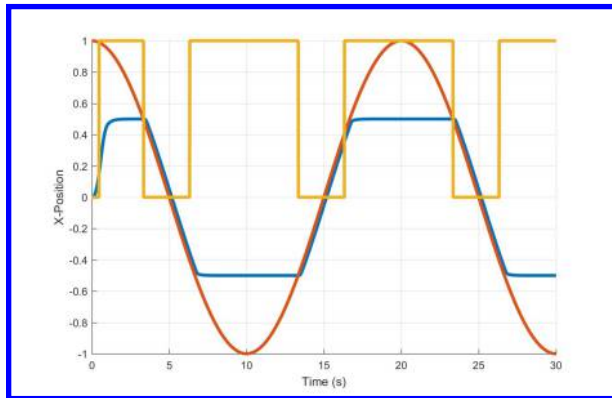
C. Simulation Geofence

The geofence block in Figure 5 contains all of the geofence blocks from Figure 3. Within the simulation, the geofence navigation is the same as the autopilot navigation. In future work, noise terms could be added to the autopilot navigation path and to the geofence navigation path to better approximate a flight system. The geofence boundaries are specified as a maximum altitude and a rectangular lateral area. If a geofence boundary violation is detected, the geofence guidance system replaces the default guidance system. Otherwise, the default guidance outputs are passed through the geofence block unchanged.

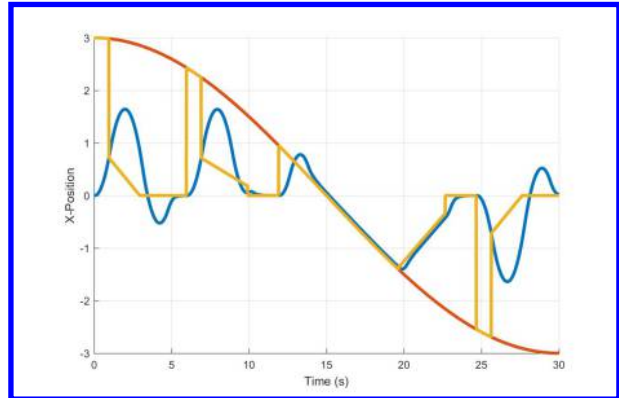
V. Results

A series of simulations demonstrate the different geofence modes: local loiter, return to launch, and shared control. Each of these geofence control modes can be used with all types of set-point control presented in Section IV-B. The results shown here all utilize position control because position control is the simplest control mode to visualize with relation to vehicle positions. The first set of results shown are the responses of each of the geofence modes to a continuous sinusoidal pilot command. The second set of results shows a flight path closer to what would occur if waypoint navigation were used.

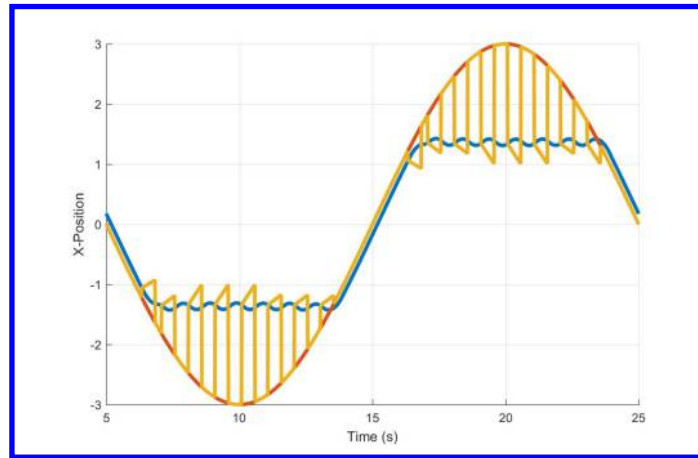
A. Sinusoidal Pilot Commands



(a) Shared Control. Blue - Vehicle Position. Red - Original Path. Yellow - Geofence On/Off



(b) Return to Launch. Blue - Vehicle Position. Red - Original Path. Yellow - Geofence Modified Path



(c) Local Loiter. Blue - Vehicle Position. Red - Original Path. Yellow - Geofence Modified Path

Figure 6. Geofence responses to a sinusoidal input.

Figure 6(a) displays the results of a position controlled simulation with geofence shared control (Section III-A). In this graph, the geofence boundaries are located at ± 0.5 . The default set-point commands are shown as the red line, the blue line is the vehicle position, and the yellow line indicates whether the geofence system is modifying commands (high value) or is passing through the default set-point commands.

Figure 6(b) displays the results of a position control simulation with an active RTL geofence (Section III-B). The geofence boundaries for the displayed plot are at ± 1.5 m along the x-axis. The red line shows the default set-point commands. While the vehicle position (the blue line) is within the geofence boundaries, the commands from the geofence system (the yellow line) match the default set-point. Once the vehicle's position crosses the boundary, the geofence commands change to a linear ramp from the vehicle's current position to zero over 3 seconds, then the geofence commands the vehicle to remain at the origin for an additional 2 seconds before returning to the default set-point commands.

Figure 6(c) shows an example of geofence LL behavior (Section III-C) as applied to a position controlled vehicle simulation. Once again, the geofence boundaries are located at ± 1.5 . The red line is the default set-point commands, which the vehicle position (the blue line) tracks until the geofence commands (the yellow line) change from the default set-point commands to prevent the vehicle from violating the geofence boundary. The effect of the LL geofence mode's short duration corrections can be seen clearly in the sections of the plot with the rapid toggling of geofence control, where the vehicle could be seen as "bouncing" off the geofence boundary.

B. Waypoint Flight Commands

Figures 7(a) and 7(b) show responses of shared control and local loiter to waypoints flight navigation. This example has the aircraft starting at $(0,0)$ then heading to the following waypoints in order: $(0.25, 0.25)$, $(0.75, 0.0)$, $(0.25, -0.25)$. Since this simulation does not confirm successful navigation, after the scheduled time, the system continues past the second waypoint without achieving it. If the geofence system was combined with a guidance system that does not skip waypoints that cannot be achieved, both local loiter and shared control would end up hovering at the geofence edge closest to the out-of-bounds waypoint. Results for return to loiter mode are not presented here because the behavior of that mode is sufficiently defined by the sinusoidal example.

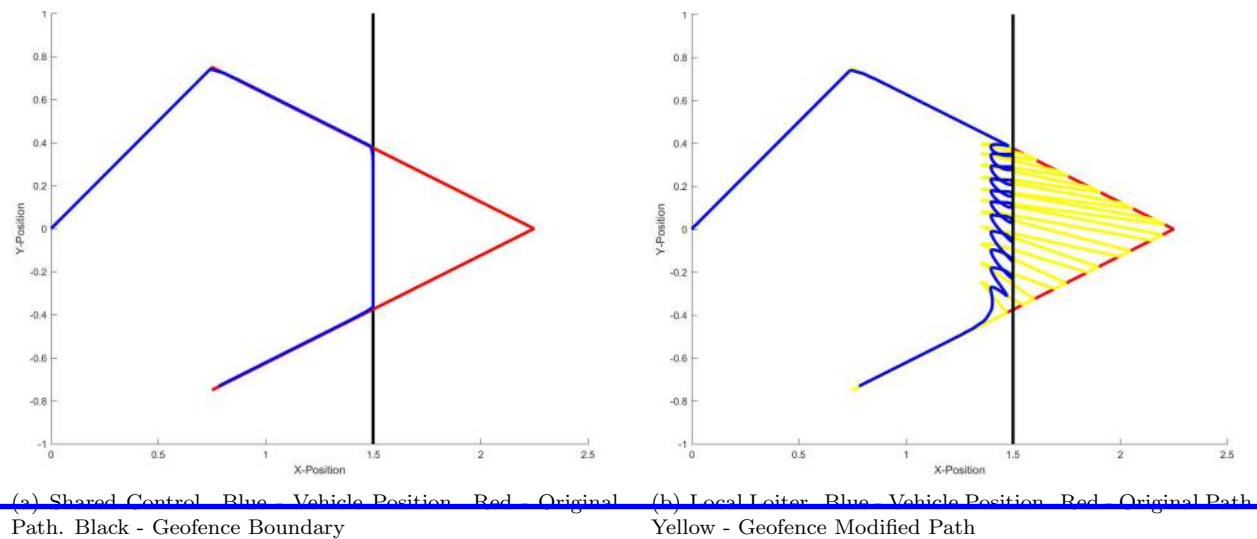


Figure 7. Geofence response during waypoint navigation.

VI. Conclusions

In conclusion, it is possible to use a geofence system that is independent of the aircraft autopilot to replicate the behavior of the geofence systems that are combined with the autopilot. The separation of the geofence system from the autopilot provides an opportunity for system redundancy, thus potentially increasing the safety of the flight vehicle. Three distinct modes of geofence guidance are implemented in simulation: shared control, return to launch, and local loiter. Each geofence mode reacts differently to a geofence violation. By developing several geofencing modes, it will be possible to provide pilots with a system that allows the selection of the preferred reaction to geofence boundary violations.

Moving forward, there are many avenues of work that need to be explored to further progress the independent geofence system. A secure No Fly Zone database needs to be developed, with the ability to be updated to include changes to zones and temporary flight restrictions. A system such as UAS traffic management (UTM) needs to extend flight permission to pilots to operate within specific areas with nominal restrictions. The geofence system needs to be configured in advance so that a pilot can fly in areas with limited or no data network. An analysis of sensor requirements and capabilities needs to be conducted to ensure that the precision and accuracy of the position estimate calculated by the geofence can be taken

into account when determining the activation point of the geofence. To make this system reasonable for large-scale implementation, the system needs to be able to learn/detect the aircraft characteristics so that the geofence system can activate before the aircraft has the chance to violate the geofence boundary. If the geofence can guarantee that the boundary will not be violated, then pilots can have a clear understanding of their flight area. Another development to be considered is communication from the geofence guidance to the autopilot guidance system, which would enable the autopilot guidance system to change the original flight plan to respond to and anticipate geofence system actions.

References

- ¹Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M., "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, Vol. 33, No. 1-2, 2012, pp. 21–39.
- ²Stevens, M. N., Coloe, B., and Atkins, E. M., "Platform-Independent Geofencing for Low Altitude UAS Operations," *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015, p. 3329.
- ³Atkins, E. M., "Autonomy as an Enabler of Economically-Viable Beyond-Line-of-Sight Low-Altitude UAS Applications with Acceptable Risk," *AUVSI North America*, 2014.
- ⁴Williams, B. P., Clothier, R., Fulton, N., Lin, X. n., Johnson, S., and Cox, K., "Building the Safety Case for UAS Operations in Support of Natural Disaster Response," *Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations Conference (AIAA Aviation 2014)*, American Institute of Aeronautics and Astronautics Inc., 2014.
- ⁵Anderson, R., Moncayo, H., Prazenica, R., Mirmirani, M., and Noriega, A., "Development of a Surrogate Autonomous Aircraft for Entry in the NASA Airspace Operation Challenge," *Proceedings of the AIAA Infotech@Aerospace Conference*, American Institute of Aeronautics and Astronautics Inc., 2015.
- ⁶Roadman, J., Elston, J., Argrow, B., and Frew, E., "Mission Performance of the Tempest Unmanned Aircraft System in Supercell Storms," *Journal of Aircraft*, Vol. 49, No. 6, 2012, pp. 1821–1830.
- ⁷Eaton, C. M., Chong, E. K., and Maciejewski, A. A., "Multiple-Scenario Unmanned Aerial System Control: A Systems Engineering Approach and Review of Existing Control Methods," *Aerospace*, Vol. 3, No. 1, 2016, pp. 1.
- ⁸Hayhurst, K. J., Maddalon, J. M., Neogi, N. A., and Verstynen, H. A., "A Case Study for Assured Containment," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
- ⁹Rule, T. A., "Drone Zoning," *96 North Carolina Law Review*, Forthcoming 2016.
- ¹⁰"Determining Safe Access with a Best-Equipped, Best-Served Model for sUAS," July 2015, https://images-na.ssl-images-amazon.com/images/G/01/112715/download/Amazon_Determining_Safe_Access_with_a_Best-Equipped_Best-Served_Model_for_sUAS.pdf.
- ¹¹"Revising the Airspace Model for the Safe Integration of sUAS," July 2015, https://images-na.ssl-images-amazon.com/images/G/01/112715/download/Amazon_Revising_the_Airspace_Model_for_the_Safe_Integration_of_sUAS.pdf.
- ¹²DJI, *Phantom 4 User Manual*, https://dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual_en_v1.2_160328.pdf.
- ¹³Beard, R., "Quadrotor Dynamics and Control Rev 0.1," 2008, Available online at: <http://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2324&context=facpub>.

This article has been cited by:

1. Mia N. Stevens, Hossein Rastgoftar, Ella M. Atkins. Specification and evaluation of geofence boundary violation detection algorithms 1588-1596. [[Crossref](#)]
2. Evan T. Dill, Steven D. Young, Kelly J. Hayhurst. SAFEGUARD: An assured safety net technology for UAS 1-10. [[Crossref](#)]