A Compiler Framework for Proactive UAV Regulation Enforcement

Huaxin Tang John Henry Burns Alexander Strong Yu David Liu
{htang28, jburns11, astrong3, davidL}@binghamton.edu
SUNY Binghamton, Binghamton NY 13903

***Abstract*— In the rapidly evolving landscape of Unmanned Aerial Vehicles (UAVs), regulation enforcement is critical. Unfortunately, existing practices are largely manual and reactive in nature. We present THEMIS[1], a novel compiler-directed approach for automated and proactive regulation enforcement. By expressing regulations through a specification language and integrating their enforcement into the compilation process, THEMIS enables safe and regulation-compliant UAV flights by enforcing prohibited and restricted areas, avoiding flights over humans, and managing maximum limits of altitude and speed. Our framework features a bidirectional interface that allows the concrete algorithms used for enforcement to be customized. Our evaluation shows THEMIS-compiled autopilots can adhere to regulatory constraints amidst complex flight conditions, while significantly reducing the burden of UAV operators.**

I. INTRODUCTION

In recent years, Unmanned Aerial Vehicles (UAVs) have found diverse applications such as in agriculture, media production, and delivery services. For example, Amazon [1] and Google's Wing [2] demonstrate their potential in reshaping our future by piloting drone delivery services. One primary hurdle against broader adoption of UAVs in public domains is *safety*, i.e., how UAVs can safely fly over public space, and how safe they are perceived to be. To address the safety need and gain confidence in the general public, a wide variety of regulations have been implemented in place. For example, US Federal Aviation Administration (FAA) [3] and European Union Aviation Safety Agency (EASA) [4] are regional governing bodies for UAV regulation. In addition, local governments, businesses, and non-profit entities may further regulate the operation of UAVs over their space.

Despite the importance of UAV regulations, their enforcement primarily relies on the individual operator's responsibility and integrity. In this manual process, the operator needs to comprehend the regulations and be constantly reminded of their enforcement. In other words, the regulations *de facto* serve as "best practice" guidelines whose unintentional violations may pose safety hazards. These regulations are most effective for *reactive* compliance analysis: they can help law enforcement conduct post-mortem "whodunit" analysis *after* the violation. By this time however, the potentially catastrophic outcome of the violation has occurred.

A. Automated and Proactive Compliance

In this paper, we describe THEMIS, a compiler-based approach for *automated* and *proactive* enforcement of UAV regulations. As seen in Fig. 1, our framework consists of several components:

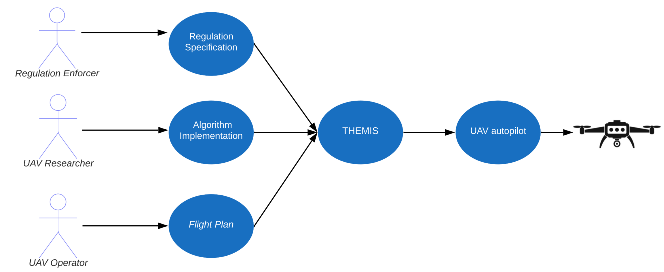


Fig. 1: THEMIS for UAV Regulation Enforcement

- a *regulation specification language* to encode UAV regulations, so that a diverse set of common regulations — such as altitude limits, speed limits, and Prohibited and Restricted Areas (PRAs) — can be written as *programs*. For example, FAA regulations can be written as one program, whereas the regulations defined by Yellowstone National Park can be written as another.
- a *bidirectional algorithm interface* to allow UAV researchers (algorithm developers) to *modularly* customize different UAV algorithms relevant for the enforcement of UAV regulations. For example, one may customize the controller for altitude limit enforcement, with choices such as Proportional-Integral-Derivative (PID) controller [5], or Incremental Nonlinear Dynamic Inversion (INDI) controller [6].
- a *compiler transformation* that combines the regulation specification, the flight plan, the developer-customized algorithm implementation into the autopilot, ready for UAV deployment.

The design philosophy of THEMIS is *regulation-by-default*, i.e., *regulation compliance should be the rule not the exception* of UAV software systems. With THEMIS, the burden of regulation compliance is reduced to a matter of making a program *compile*: all generated target code is guaranteed to comply with the regulations it compiles against. Through judiciously selecting the default regulation specification for compilation — e.g., a US-based UAV operator by default compiles her program against the FAA specification if none is provided — THEMIS can completely eliminate several classes of regulation non-compliance. This process is *proactive*, i.e., compliance is established before the target code is generated, let alone being deployed to a UAV

[1]Themis is a Titaness symbolizing divine law and justice, representing the social order and customs essential in Greek mythology.

for flights. This process is also *automatic*: the intricacies of modifying the autopilot source code to accomplish the compliance is delegated to the compilation process.

For end users, THEMIS comes with several advantages. First, it significantly reduces the impact of human errors by shifting the UAV regulation compliance practice from human-dependent enforcement to automated enforcement. Second, as the number of UAVs in operation grows, the automated approach adopted by THEMIS is essential for *scalability*. Third, our compiler naturally decouples the orthogonal interests of the UAV *regulator*, the UAV *operator*, and the UAV *researcher*. With THEMIS, each role can focus on one of three programming interfaces — the regulation, the flight plan, and the algorithm customization — and it is the compiler that "ties the knot" to bring all concerns into one organic whole. This modular design streamlines the integration of updated regulations and cutting-edge algorithms, ensuring that UAV operators remain in sync with the latest regulatory mandates and algorithmic advancements.

THEMIS has a complementary role with existing algorithm-centric efforts. For example, geofencing [7] is a well-known technique for supporting PRAs. There is also a large body of work on altitude and speed control [8], [9], [10]. With THEMIS, these algorithms can "plug and play" for regulation enforcement. Upfront, we should also make clear that the motivation of our work is safety *not* security. Our goal is to endow a good-intentioned UAV operator with tools for proactive and automated regulation enforcement, not to thwart a malicious UAV operator from circumventing regulations. As safety and security often go hand in hand, we speculate our compiler-based approach may serve as a base to further design UAV security mechanisms when THEMIS becomes a part of the Trusted Computing Base (TCB).

B. Contributions

To the best of our knowledge, THEMIS is the first systematic framework to study UAV regulation enforcement through a compiler-centric approach. It is a novel instance of legal-computational integration in the domain of UAVs. The key contributions of this paper are as follows:

1) the design philosophy of regulation-by-default for UAV autopilots, enabled by a compiler-based approach for proactive and automated UAV regulation enforcement, and manifested by a programming model that divides the responsibility of UAV programming among the regulator, the operator, and the researcher,
2) the design of the regulation specification language, where important classes of UAV regulations are programming abstractions, and the composition of multiple regulations is supported.
3) the compiler implementation that generates the autopilot control loop based on the combined programs of the regulation, the flight plan, and the algorithm implementations.
4) default algorithm implementations for PRA enforcement, no flying over humans, speed limit, and altitude limit.

Regulation	Description
§ 107.39	**Operation over human beings.** No person may operate a small unmanned aircraft over a human being unless that human being is: 1) Directly participating in the operation of the small unmanned aircraft; 2) Located under a covered structure or inside a stationary vehicle that can provide reasonable protection from a falling small unmanned aircraft; or
§ 107.45	**Operation in prohibited or restricted areas.** No person may operate a small unmanned aircraft in prohibited or restricted areas unless that person has permission from the using or controlling agency, as appropriate.
§ 107.51	**Operating limitations for small unmanned aircraft.** A remote pilot in command and the person manipulating the flight controls of the small unmanned aircraft system must comply with all of the following operating limitations when operating a small unmanned aircraft system: (a) The ground speed of the small unmanned aircraft may not exceed 87 knots (100 miles per hour). (b) The altitude of the small unmanned aircraft cannot be higher than 400 feet above ground level.

TABLE I: Key FAA Operation Regulations (Some clauses are shortened in this presentation.)

THEMIS is an open-source project. The source code of our compiler, together with default algorithm implementations, can be found at https://github.com/sick-py/Themis.

II. BACKGROUND AND MOTIVATIONS

A. UAV Regulations

In this section, we summarize UAV regulations, an emerging area of policy-making and legalization. We start with US regulations as a representative example, followed by a brief discussion on the broader international community.

The Federal Aviation Administration (FAA) is the primary governing body for US-based UAV regulations. The current Advisory Circular (AC) [11] of the FAA categorizes aviation safety [12], [3] into three primary domains: equipment (part A), operations (part B), and personnel (part C). While Part A and Part C are beyond the scope of computing, their ultimate objective is to ensure that UAVs adhere to the operational rules outlined in Part B.

A subset of Part B regulations [13] are outlined in Table I. For example, regulation § 107.39 establishes limitation on operation over human beings, UAV shouldn't fly over a human being unless certain conditions are met. Regulation § 107.45 forbids UAV operations within PRAs unless express permission has been granted. Regulation § 107.51 establishes speed limit and altitude limit. Together, these regulations ensure the safe and responsible operations of UAVs.

In addition to FAA-enforced regulations, state/local governments and organizations can further impose restrictions. For example, while FAAs have strict enforcement of airports as PRAs, individual entities — from corporation campuses to national park areas — may further define additional PRAs, commonly referred to as "no-fly zones".

Outside the US, UAV regulation is also common. The European Commission and the European Aviation Safety Agency formulates a risk-based regulatory framework for UAVs [14]. For example, UAVs in the open category are restricted to altitudes below 120 meters. Lee et al. [12] conducted an analysis of UAV regulations worldwide.

Broadly, UAV regulations are only a portion of regulations in aviation safety. For example, FAA has basic and stringent requirements on collision avoidance [15] and other real-time guarantees for certified systems [3], beyond the scope of this paper. More broadly, software solutions to improve the safety and reliability of UAVs [16], [17], [18], [19], [20], [21] is a growing area of pursuit.

B. Motivations And State of the Art

Ubiquitous enforcement of UAV regulations is critical for a number of reasons. First, UAVs can pose significant *public safety* risk if not operated safely; regulations like § 107.39 and §107.51 can help mitigate these risks. Second, UAVs equipped with cameras or other surveillance equipment can invade *privacy* if misused; regulations such as § 107.45 can help prevent such misuse by setting restricted areas, ultimately protecting privacy by setting limits on when, where, and how drones can be used for surveillance purposes. Finally, regulations can help clarify *legal liability* in the event of accidents or incidents, offering protection and fairness to involved parties such as drone operators and victims.

While these regulations are crucial, the state of the art for regulation enforcement is primitive and manual. The FAA's regulations are primarily self-enforced, relying heavily on the experience and responsibility of UAV operators. This is *undesirable for both operators and regulators*. For operators, understanding and complying with all regulations is onerous and time-consuming, and it can be daunting if one considers the variations between countries, states, and locales. The consequence is that a well-intentioned UAV operator could unintentionally breach regulations and threaten safety. For regulators, monitoring these regulations is also a daunting task due to the vast number of UAVs in operation. Although the FAA has mechanisms to report violations, it is challenging to identify violators unless an accident occurs or a complaint is lodged [22].

As we described in § I, the state of the art for UAV regulation enforcement is largely *reactive*. A draconian "just say no" approach — i.e., not allowing any UAV to fly unless authorized — may indeed be trivially proactive, but it is unrealistic. For example, one may in theory rely on the Low Altitude Authorization and Notification Capability (LAANC) of FAA — an approach currently only used for controlled spaces — to authorize all operators. This may introduce delays and does not scale.

Geofencing [7] is one of the few successful examples where automated computer technologies meet regulation enforcement (see § 107.45). At its essence, this GPS-based technology is also *reactive*: a UAV may indeed turn around upon the encounter of a geofence, but this often comes with the cost of wasted resources: if the UAV had been aware of the regulation in place, a more optimal flight plan could have been computed and avoided the geofenced area all together.

```
<!DOCTYPE regulator SYSTEM "regulator.dtd">
<regulator name = "FAA" max_alt="121.92" max_speed="5" >
<PRA color="red" name="keepout">
<waypoint name="c1" x="68.1" y="136.4"/>
<waypoint name="c2" x="116.2" y="142.0" />
<waypoint name="c3" x="123.1" y="109.1"/>
<waypoint name="c4" x="110.0" y="92.9"/>
</PRA>
<stationaryPRP name="personS" x="0" y="0" radius="3"/>
<dynamicPRP name="personD" radius="3"/>
</regulator>
```

Fig. 2: A Sample Regulation Specification

Algorithm 1 THEMIS Bi-Directional Interface

```
import type COORD // coordinate
import type WID // waypoint ID
export type WP struct { w : WID; c : COORD } // waypoint
export type SEG struct { s : WID; d : WID } // flight segment
export type FPLAN LIST{SEG } // flight plan
export type PRA struct {bound : SET{WID}; band : FLOAT}
        // prohibited or restricted areas
export type PRP struct { p : WID; band : FLOAT } // prohibited or restricted point
export type MAP struct { bounds : SET{PRA}; ss : SET{SEG} }
export type PATH : LIST{WID}
export INIT(fp : FPLAN, nf : SET{PRA}, np : SET{PRP}) : MAP
        // initialize the map with declared plan fp, PRA nf, PRP np
export SAFEBANDPRA(b : FLOAT, z : PRA): PRA
        // PRA safety band
export SAFEBANDPRP(b : FLOAT, p : PRP): PRP
        // PRP safety band
import FINDPATH(s : WID, d : WID, m : MAP): PATH
        // compute the shortest path from start s to destination d in map m
import UPDATEMAP(m : MAP, nf : SET{PRA}) : MAP
        // dynamically update map m given the updated PRA nf
export GETCUR : COORD // return current position
export NAVPATH(now: WID, p: PATH)
        // follow path p by one segment, starting at waypoint now
import LOCATEDYNAMICPRP(w: WID) : WP
        // detect and update a dynamic PRP
export UNITSYS(s : ENUM(metric, imperial)) // unit system
```

Fig. 3: THEMIS Interface for PRA and PRP Enforcement

III. Design

A. Regulation Specification

A core component of THEMIS is a language for regulation specification. We choose to embed our language grammar in XML for its downstream interoperability. An example regulation specification is shown in Fig. 2.

1) PRAs: THEMIS features a new abstraction with the XML tag `PRA`. Each `PRA` defines a polygon area, the corners of which are delineated by the `waypoint` clause. Such areas typically include vicinities around airports, government establishments, military installations, and other critical infrastructures, in alignment with regulation 107.45 [13]. In the example, one `PRA` named `keepout` is defined, with 4 corners defined by waypoints `c1`, `c2`, `c3`, and `c4`. Semantically, the `PRA` abstraction captures the requirement of PRAs: UAVs should not fly into the defined areas.

2) Human Avoidance: THEMIS introduces two tags: `stationaryPRP` and `dynamicPRP`. The `stationaryPRP` tag delineates a waypoint which represents either a stationery person or object over which flight is prohibited. The `dynamicPRP` tag establishes a potentially mobile person/object that should not be flown over. In the provided example, a `stationaryPRP` named `peopleS` represents stationary individual, while a `dynamicPRP` named `peopleD` denotes a moving individual. Semantically, the `PRP` abstraction specifies that the UAV should avoid flying over the specified points. The `dynamicPRP` is required for all human beings in the proximity of the UAV, except the UAV operators with explicit permissions as defined in Regulation §107.39. How to automate human detection is an implementation issue we detail in § III-E.

3) Speed Limit and Altitude Limit: THEMIS introduces two tags, `max_alt` and `max_speed`, to proactively adhere to attitude/speed limits. These tags define the maximum allowable altitude and speed in accordance with FAA regulations, mitigating the risk of violations.

B. Regulation Composition and Regulation-by-Default

THEMIS allows multiple regulation specifications to be composed together. Given two specifications r_1 and r_2, an autopilot generated with the composed regulation conforms to these rules:

- the PRAs it avoids are the union of that of r_1 and r_2;
- the human beings it avoids are the union of those defined/tracked in r_1 and r_2;
- the altitude limit is the lower of that of r_1 and r_2;
- the speed limit is the lower of that of r_1 and r_2;

THEMIS is a "regulation-by-default" compiler: a flight plan can only be compiled when a regulation specification is provided. In addition, based on the geographical location of the UAV, a default regulation is chosen, with the FAA specification for US-based UAVs, and the EU specification for EU-based UAVs [2]. When THEMIS users explicitly defines a regulation, our compiler always considers the regulation to enforce as the composition of the default geographically determined regulation and the user-provided regulation. The composition rules above ensure that a UAV operator does not accidentally override a "default", say, the FAA regulation, by compiling her flight plan with less restrictions.

C. Bi-Directional Algorithm Interface

THEMIS features a *bi-directional* interface design: a data type or function can be either declared as an **import** or an **export**. An **import** declaration is aligned with our goal of developing a *customizable* compiler framework: developers and researchers with new algorithms for implementing a specific form of regulation enforcement can "plug in" their implementations, including both concrete data structure representations and the algorithms defined over them. As a standalone system, THEMIS also provides a default implementation for each **import**, detailed in the next subsection. An **export** declaration is a functionality already implemented by THEMIS. They can be used in two purposes. First, the algorithm designers who implement the **import** functionalities can use them as *callbacks* to our framework. Second, the **export** functions serves as the *interface between the parser and the code generator* for the THEMIS compiler.

Fig. 3 defines the bi-directional algorithm interface related to PRA and PRP enforcement. Two examples of **import** data types are the representation of the coordinates (COORD) and the ID of the waypoints (WID). The former is needed because different coordination systems are popular, such as LTA, UTM, and relative coordinates. The latter is often represented as a mnemonic string or a numeric ID, but leaving it as **import** further allows the algorithm designers to keep additional algorithm-specific metadata. We leave the discussion of the main **import** functions — UPDATEMAP, FINDPATH and LOCATEDYNAMICPRP — to § III-E.

The WP data type serves as an example to show the **export** data types at work. It defines the type for a waypoint, which consists of an ID and a coordinate. Despite being simple, this example shows the fundamental *dependency* between the **import** and **export** declarations, where the latter is defined through the former. As an example of the interface between parsing and code generation, note that INIT is defined as an **export**. After the regulation specification is parsed, code is generated to initialize the PRA and PRP objects based on the declarations (as well as the flight plan). With that, the code generator only needs to generate a call to INIT to complete the initialization process of the autopilot runtime.

D. THEMIS *Common Runtime Features*

The autopilot generated by THEMIS shares several features in its runtime, transcending individual algorithm-specific customizations.

a) Uniform Treatment for `PRAs` and `PRPs`: An insight of THEMIS is that the enforcement of regulations such as "no flying over human beings" can be reduced to a dynamic variant of PRA enforcement. Concretely, we treat a PRP as a PRA bounded by a circle (approximated by a polygon) centered at the specified no-flying waypoint. As a result, Fig. 3 also serves as the common interface for PRP enforcement. With this unified treatment, we are able to apply the same path planning algorithm – regardless how it is customized by the algorithm developer — to both PRAs and PRPs. The implication here is that despite the PRAs are statically defined, PRPs may be either `stationery` or `dynamic`. The latter necessities a path planning algorithm that can dynamically update the waypoints to consider.

b) Safety Band Support: In THEMIS, the PRA comes with a *safety band*: there should be a "buffer" distance so that the UAV does not fly too close to the PRA. The specific size of the safety band can be customized, through the SAFEBANDPRA function. In a similar vein, one may also customize the radius of the circle we construct around the PRP, through the SAFEBANDPRP function.

[2] As an imperfect solution, THEMIS selects the FAA specification for non-US non-EU UAVs, which we plan to refine as future work.

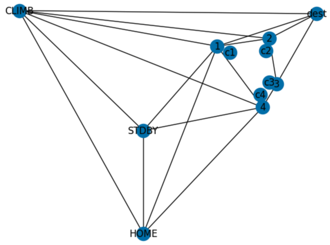


Fig. 4: A Visibility Graph Example

c) Dynamic Maps: THEMIS defines the data structures of MAP and PATH, essential for path planning algorithms. While the specific algorithms used for MAP and PATH are up for customization, what remains in the common runtime is that PRA enforcement and PRP enforcement at its heart is a dynamic path planning problem. To see the dynamic nature of this, the UPDATEMAP must be implemented by the algorithm developer. The LOCATEDYNAMICPRP addresses the detection of the potential movement of each `dynamicPRP`. Whenever the THEMIS common runtime identifies a new position for the target object, it updates PRP and subsequently calls UPDATEMAP to recalculate the path.

d) Control-Based Speed and Altitude Limit: A naive solution for speed limit (or altitude limit) regulation is to add a runtime check to the control loop and adjust the speed (or altitude) when the limit is reached. This approach however is flawed: by the time the limit is reached, the momentum of the UAV may well push the UAV to exceed the limit.

THEMIS speed/altitude limit enforcement is control-based: the UAV must proactively adjust its speed/altitude well before the limit is reached. The specific controller choice can be customized (see § III-E). We elide the interface specification for setting the controllers and tuning their parameters.

E. Algorithm Default Implementations

The default algorithm implementation of THEMIS for PRA and PRP enforcement employs visibility graphs [23], [24]. Consider the regulation specification in 2 as an example. The resulting visibility graph is illusrated as a MAP in 4. Here, waypoints labelled 1, 2, 3, and 4 represent the waypoints modified from the original declared waypoints `c1`, `c2`, `c3`, and `c4`, after the safety bands are applied. Upon the construction of the MAP, THEMIS employs the FINDPATH function to discern the most efficient route from the source to the destination, subsequently transmitting the resulting PATH to NAVPATH, the routine defined within the common runtime of THEMIS to navigate a segment on the PATH. The current implementation of FINDPATH is A-Star. For mobile PRPs, our default implementation for LOCATEDYNAMICPRP utilizes the aggregate count of orange pixels to detect the object. The onboard camera of the UAV continuously captures images at a rate of 20 FPS. Once the aggregate orange pixel count reaches a threshold in the image, our implementation interprets it as the presence of a person. The radius of the safety band is set at 3 meters. In our default implementation, the PID (Proportional-Integral-Derivative) controller [5] is used for speed and altitude limit regulation. It continuously calculates an error value as

Listing 1: Flight Plan

```
<!DOCTYPE flight_plan SYSTEM "flight_plan.dtd">
<flight_plan alt="600" ground_alt="380" lat0="
    37.2109800" lon0="-113.4567800"
    max_dist_from_home="500"
name="PRATest" qfu="270" security_height="25">
<waypoints>
<waypoint name="HOME" x="0" y="0"/>
<waypoint name="STDBY" x="0" y="75"/>
<waypoint name="CLIMB" x="-114.5" y="162.3"/>
<waypoint name="dest" x="160" y="160"/>
</waypoints>
<blocks>
<block name="Fly">
<go wp="STDBY"/>
<go wp="dest"/>
<go wp="CLIMB"/>
</block>
<block name="GoHome">
<go wp="HOME"/>
</block>
</blocks>
</flight_plan>
```

Fig. 5: A Flight Plan Example

the difference between a desired setpoint (in this case, the altitude or speed limit) and a measured process variable (the UAV's current speed/altitude). The controller then seeks to minimize this error by adjusting the UAV.

We note that there exist alternative path planning algorithms [25], [26], [27], [28], alternative human/object detection algorithms [29], [30], [31], [32], [33], [15], [34], and alternative controllers [8], [9], [10]. THEMIS is complementary to these algorithms by allowing them to be plugged into our framework for regulation enforcement.

F. Flight Plan

Instead of developing a flight plan language from scratch, THEMIS extends from an existing XML-based language, Paparazzi flight plans [35], for defining the navigation path.

The flight plan example in Fig. 5 defines the `waypoints` to be used. Each `block` defines a stage of in-flight behavior. For instance, in the provided example, the UAV first `Fly` and then `GoHome`. Within the `Fly` stage, the UAV traverses the `STDBY`, `dest`, and `CLIMB` waypoints.

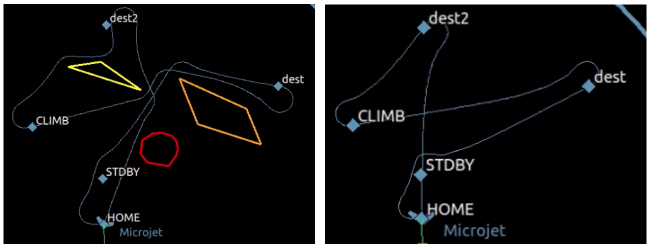
IV. EVALUATION

In this section, we report our experience in leveraging THEMIS for UAV regulation. Our experiments are constructed over the Paparazi simulator, with human detection supported through Gazebo.

A. Prohibited and Restricted Areas

We first evaluate the effectiveness of THEMIS over PRA enforcement. We define a flight plan that traverses a series of pre-defined waypoints, from `HOME`, to `STDBY`, to `dest`, to `CLIMB`, to `dest2`, and finally back to `HOME`. In our regulation specification, we define 3 PRAs, each as a polygon. The flight plan consists of 42 lines of code and the regulation specification consists of 27 lines of code. Their source code can both be found in our repository.

Figure 6 shows the result of simulation over a fixed-wing craft, where the actual flight path is indicated by



(a) With PRA regulation (b) Without PRA regulation

Fig. 6: Enforcing PRAs with THEMIS

thin lines. Without any regulation, the UAV directly traverses all waypoints, as shown in the left subfigure. When the regulation specification is in place and compiled with THEMIS, the flight path is shown on the right. Two important observations can be made. First, the THEMIS-regulated flight can successfully avoid all PRAs. Second, the regulated flight largely follows a similar flight path as the no-PRA one, faithfully carrying out the mission defined in the flight plan.

With THEMIS, the programmer's task of flight planning is significantly simplified. Her original flight plan does not require *any* change. The regulation specification is likely written by some regulators. How to avoid PRAs is the responsibility of the compiler.

B. Human Avoidance

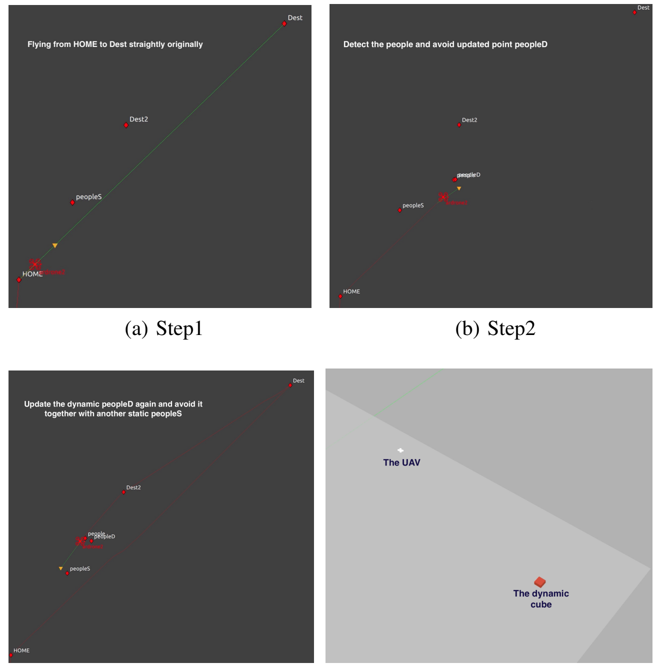
To assess the capability of THEMIS in human avoidance, our flight plan consists of starting from `HOME`, traversing two designated waypoints (`Dest` and `Dest2`), and subsequently returning to `HOME`. We introduce two humans: one stationary `PeopleS` and the other mobile `PeopleD`. `PeopleD` remains unknown until captured by the camera of the UAV, first located between `HOME` and `DEST`. As the UAV redirects towards `Dest2`, `PeopleD` moves to block the UAV's return path to `HOME`. The flght plan consists of 33 lines of code, and the regulation specification has 10 lines.

Fig. 7 demonstrates that the THEMIS-compiled flight plan — without any change to its original definition — can successfully avoid both the stationary human being and the mobile human being. In Step 1, the UAV plans to fly directly from `HOME` to `DEST`. In Step 2, the UAV adjusts its path to circumnavigate the stationary human being. As the mobile human moves to obstruct the UAV in Step 3, the UAV avoids both human beings. A Gazebo simulation of the people detection can be found in the last subfigure.

C. Altitude Limit and Speed Limit

In final evaluation, we assessed the proactive capabilities of THEMIS with respect to speed and altitude limits.

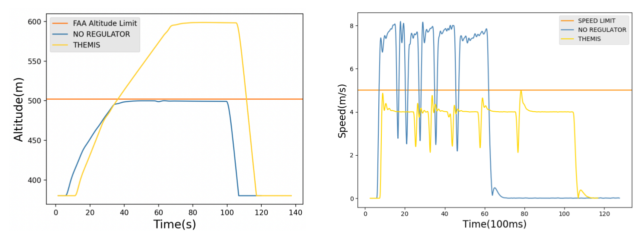
For altitude, we define a flight plan where the UAV is programmed to execute a circular maneuver at an altitude of 600 meters with and without the regulator. In the absence of the regulator, the UAV follows the original flight plan, as depicted in Fig. 8(a). In contrast, the UAV under THEMIS compilation is able to halt its ascent proactively and adhere to the altitude limit.



(a) Step1 (b) Step2

(c) Step3 (d) Gazebo Simulation

Fig. 7: Human Avoidance (The green line represents the projected flight path the UAV intends to follow, while the red line traces the UAV's actual flight path. The yellow triangle indicates the UAV's intended direction.)



(a) Altitude Regulation (b) Speed Regulation

Fig. 8: Enforcing Altitude and Speed Limits with THEMIS

For speed, we define a flight plan where the UAV performs a circular maneuver at a speed of 7 meters per hour with and without the regulator. Fig. 8(b) shows that the THEMIS-compiled autopilot is able to stay below the speed limit whereas the original flight plan without the regulation cannot.

V. CONCLUSION

THEMIS leverages the strength of the compiler to enforce compliance with UAV flight regulations in a proactive and automated manner. By implementing regulatory constraints with little burden to the operator, this framework alleviates the complexity of adhering to regulatory mandates. In the future, we plan to THEMIS to support dynamic regulation updates in the presence of multiple regulation zones.

Acknowledgements The project is sponsored by US NSF CNS-1823260.

REFERENCES

[1] (2016) Amazon drone delivery. Accessed: 2023-06-12. [Online]. Available: https://www.amazon.com/gp/help/customer/display.html?nodeId=T3jxhuvPfQ629BOIL4

[2] M. McFarland, "Google drones will deliver chipotle burritos at virginia tech," *CNN Money*, September 2016, accessed: 2023-06-12.

[3] Federal Aviation Administration. (2023, May) Unmanned aircraft systems (uas). Accessed: 2023-05-23. [Online]. Available: https://www.faa.gov/uas

[4] European Union, "European union aviation safety agency," May 2023, accessed: 2023-05-23. [Online]. Available: https://www.easa.europa.eu/

[5] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: Pid controller design," *Industrial Engineering Chemistry Process Design and Development*, Jan 1986.

[6] M. D. Pavel, P. Shanthakumaran, Q. Chu, O. Stroosma, M. Wolfe, and H. Cazemier, "Incremental nonlinear dynamic inversion for the apache ah-64 helicopter control," *Journal of The American Helicopter Society*, Apr 2020.

[7] P. Pratyusha and V. Naidu, "Geo-fencing for unmanned aerial vehicle," *International Journal of Computer Applications*, vol. 975, p. 8887, 2013.

[8] W. Koch, R. Mancuso, and A. Bestavros, "Neuroflight: Next generation flight control firmware," *arXiv: Robotics*, Jan 2019.

[9] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for uav attitude control," *ACM Transactions on Cyber-Physical Systems*, Feb. 2019.

[10] H. L. N. N. Thanh and S. K. Hong, "Robust dynamic sliding mode control-based pid–super twisting algorithm and disturbance observer for second-order nonlinear systems: Application to uavs," *Electronics*, Jul. 2019.

[11] FAA. (2021) Small unmanned aircraft systems (suas). Accessed: 2023-07-25. [Online]. Available: https://www.faa.gov/documentLibrary/media/Advisory_Circular/Editorial_Update_AC_107-2A.pdf

[12] D. Lee, D. J. Hess, and M. A. Heldeweg, "Safety and privacy regulations for unmanned aerial vehicles: A multiple comparative analysis," *Technology in Society*, 2022.

[13] Federal Aviation Administration. (2023, May) Aeronautics and space part 107—small unmanned aircraft systems. Accessed: 2023-05-23. [Online]. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107

[14] E. Commission. (2019) European commission presents an aviation strategy for europe. Accessed:2023-09-06. [Online]. Available: https://transport.ec.europa.eu/news/european-commission-presents-aviation-strategy-europe-2019-06-27_en

[15] M. Liu, X. Wang, A. Zhou, F. Xiuyuan, M. Yiwei, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Sensors*, Apr. 2020.

[16] A. Czerniejewski, J. H. Burns, F. Ghanei, K. Dantu, Y. D. Liu, and L. Ziarek, "Jcopter: Reliable UAV software through managed languages," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*. IEEE, 2021, pp. 4282–4289.

[17] X. Liang, J. H. Burns, J. Sanchez, K. Dantu, L. Ziarek, and Y. D. Liu, "Understanding bounding functions in safety-critical UAV software," in *43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, Madrid, Spain, 22-30 May 2021*. IEEE, 2021, pp. 1311–1322.

[18] A. Di Sorbo, F. Zampetti, A. Visaggio, M. Di Penta, and S. Panichella, "Automated identification and qualitative characterization of safety concerns reported in uav software platforms," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, apr 2023. [Online]. Available: https://doi.org/10.1145/3564821

[19] J. H. Burns, X. Liang, and Y. D. Liu, "Adaptive variables for declarative UAV planning," in *COP '20: Proceedings of the 12th International Workshop on Context-Oriented Programming and Advanced Modularity, COP@ECOOP 2020, Virtual Event, 21 July 2020*. ACM, 2020, pp. 1:1–1:7.

[20] D. Wang, S. Li, G. Xiao, Y. Liu, and Y. Sui, "An exploratory study of autopilot software bugs in unmanned aerial vehicles," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021. Association for Computing Machinery, 2021, p. 20–31.

[21] Y. D. Liu and L. Ziarek, "Toward energy-aware programming for unmanned aerial vehicles," in *3rd IEEE/ACM International Workshop on Software Engineering for Smart Cyber-Physical Systems, SEsCPS@ICSE 2017, Buenos Aires, Argentina, May 21, 2017*. IEEE, 2017, pp. 30–33. [Online]. Available: https://doi.org/10.1109/SEsCPS.2017.8

[22] Federal Aviation Administration. (2023, May) Legal enforcement actions. Accessed: 2023-05-24. [Online]. Available: https://www.faa.gov/about/office_org/headquarters_offices/agc/practice_areas/enforcement/enforcement_actions

[23] T. Lozano-Pérez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[24] S. Ghosh and D. Mount, "An output-sensitive algorithm for computing visibility graphs," *SIAM Journal on Computing*, vol. 20, no. 5, pp. 888–910, 1991.

[25] D. M. K. K. V. Rao, H. Habibi, J. L. Sanchez-Lopez, and H. Voos, "An integrated real-time uav trajectory optimization with potential field approach for dynamic collision avoidance," *arXiv preprint arXiv:2303.02043*, 2023.

[26] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning," *IEEE Transactions on Industrial Informatics*, Feb. 2013.

[27] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav," *Aerospace Science and Technology*, Jan. 2012.

[28] M. D. Phung and Q. P. Ha, "Safety-enhanced uav path planning with spherical vector-based particle swarm optimization," *Applied Soft Computing*, 2021.

[29] H. Mliki, F. Bouhlel, and M. Hammami, “Human activity recognition from uav-captured video sequences,” *Pattern Recognition*, 2019.
[30] P. Doherty and P. Rudol, “A uav search and rescue scenario with human body detection and geolocalization,” *Springer Berlin Heidelberg eBooks*, Dec. 2007.
[31] P. Rudol and P. Doherty, “Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery,” *Proceedings*, Mar. 2008.
[32] E. Lygouras, N. Santavas, A. Taitzoglou, K. N. Tarchanidis, A. C. Mitropoulos, and A. Gasteratos, “Unsupervised human detection with an embedded vision system on a fully autonomous uav for search and rescue operations,” *Sensors*, Aug. 2019.
[33] C. Jiang, H. Ren, X. Ye, J. Zhu, H. Zeng, Y. Nan, M. Sun, X. Ren, and H. Huo, “Object detection from uav thermal infrared images and videos using yolo models,” *International Journal of Applied Earth Observations and Geoinformation*, vol. 112, p. 102912, 2022.
[34] J. Torres-Sánchez, F. López-Granados, and J. M. Peña, “An automatic object-based method for optimal thresholding in uav images: Application for vegetation detection in herbaceous crops,” *Computers and Electronics in Agriculture*, Jun. 2015.
[35] Alonsoac. (2023, May) Flight plans. Accessed: 2023-05-25. [Online]. Available: https://wiki.paparazziuav.org/wiki/Flight_Plans