OSSで動作するUAVの悪用防止プロセッサによる侵入禁止区域での強制軟着陸の基礎検証

○福田 杜和（芝浦工大） 安孫子 聡子（芝浦工大） 佐藤 大祐（都市大） 辻田 哲平（防衛大）

Recently, the widespread adoption of OSS (Open-Source Software), which permits the free distribution and enhancement of programs, has significantly facilitated the development of UAVs (Unmanned Aerial Vehicles). However, there is growing concern that UAVs equipped with OSS may be exploited for inhumane purposes, such as terrorist activities and other criminal acts unintended by OSS developers. In response to these concerns, this study developed and validated an anti-abuse system designed to seize control of a UAV upon its entry into a restricted zone, thereby enforcing a controlled soft landing. The results indicated that the UAV initiated an appropriate altitude descent immediately after the entry into the restricted zone was detected.

1. 緒言

近年，無人航空機 (UAV: Unmanned Aerial Vehicle) の急速な発展により，その産業利用は多岐にわたっている．その市場規模 (主に商業用，娯楽用) は 2030 年までに約 558 億 US ドルまで増加すると言われている [1]．その一方で，UAV がテロ行為などの非人道的な用途に利用される可能性が強く懸念されている．国際連合では，非人道的目的のための UAV の取得，武器化，配備方法についての詳細な分析をまとめ，各国が現在直面するセキュリティ上の課題を明らかにした [2]．Heubl は，一般販売されている民生用ドローンの悪用事例，それに対する対応策と課題について示した [3]．また，テロ行為以外にも，盗撮や密輸などでも悪用されているという問題がある [4]．

これらの問題に対して，DJI などの民生用ドローンでは，プロプラエタリなシステムとして，悪用防止対策を進めている [5]．一方，ソースコードを無償で配布，改良，再配布を行うことができるオープンソースソフトウェア (OSS: Open-Source Software) を用いて開発したロボットや UAV に対しての対策はいまだ確立されていない．米国 CISA(Cybersecurity & Infrastructure Security Agency) は政府機関や重要なインフラシステム等においても OSS に大きく依存していることの危険性を分析し，OSS を適切に理解し保護することの重要性を示している [6]．

UAV の開発においては，OSS の存在のみならず，Pixhawk [7] 等のハードウェアとソフトウェアの両方がオープンソースとなるフライトコントローラ (FC: Flight Controller) も存在しており，高校生レベルの技術力でも開発できてしまうほど容易となっている．これらの問題に対して，OSS 開発制度がもたらす社会的有用性を阻害することなく，テロ行為等に悪用される可能性を低減する枠組みが必要である．

現在，オープンソースの開発者は Github 等のソースコード公開サイトに自作コードを公開し，ユーザは同コードをそのまま，または一部改変して使用できる．ソースコードを公開しているため，プログラム中に悪用防止機能を持たせても，その部分を除外するだけで簡単に悪用されてしまう．そこで，橋本らは，プロセッサ内部の外部からの改ざんを防止した領域に悪用防止機能を持たせ，OSS によるプログラムは読み書き許可領域でのみ実行可能とするロボット開発のフレームワークを提案した [8]．また，具体的な悪用防止機能を搭載したセキュアプロセッサシステムを考案し，移動ロボットに搭載し，その有用性を評価した [9]．伊藤らは，よりテロ行為に悪用されやすい，かつ，二次被害の発生が懸念される UAV に悪用防止機能を搭載し，制御系奪取による強制軟着陸の模擬検証を行った [10]．

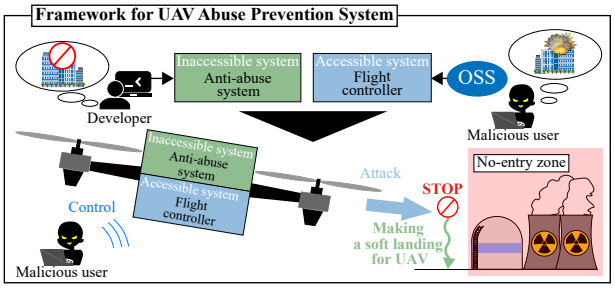


図 1: Prevention of UAV abuse

本稿では，UAV が侵入禁止区域に侵入した場合の制御系奪取による強制軟着陸を実現する悪用防止プロセッサの開発とその基礎検証結果について述べる．ここでは，基礎検証としてモーションキャプチャシステムによる模擬飛行環境下での強制軟着陸実験を実施する．

2. OSS 搭載 UAV の悪用防止手法の提案

本研究で提案する UAV の OSS 悪用防止システムを Fig. 1に示す．悪用防止システムは，エンドユーザによってアクセス不可能なプロセッサ内部領域に実装することで，エンドユーザによるソースコード削除や物理的な信号線の切断などによる悪用防止システムの無効化を防止するとともに，OSS の FC による制御を監視する．一方，OSS 開発者によって開発された制御プログラムはエンドユーザらにアクセス可能な領域内に実装できるようにプロセッサを設計し，エンドユーザはアクセス可能領域内では自由に制御プログラムを開発することができる．以上より，エンドユーザ程度の技術力では悪用が困難となる一方で，プロセッサレベルの改良・改ざんが可能な専門的知識を有する人物の

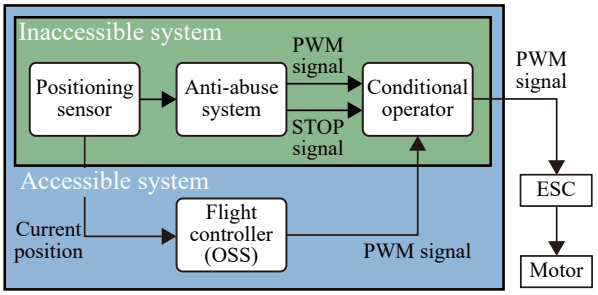


図 2: Abuse prevention methods

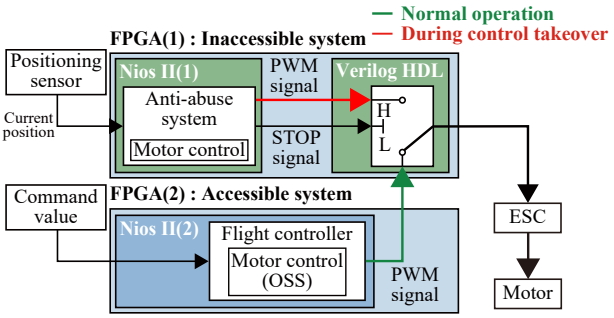


図 3: Prototype of abuse prevention processor

特定はエンドユーザよりも絞り込みやすいと考えられ，抑止効果が高くなると考えられる．ここでは，具体的な悪用防止システムとして，UAV の侵入禁止区域への侵入を検知した場合，機体の制御を OSS 制御プログラム側から悪用防止システム側が奪取し，強制的に軟着陸させるシステムを構築する．

Fig. 2に本稿で提案する悪用防止システムを搭載したプロセッサの概要図を示す．OSS の制御プログラムより生成されたモータの制御信号 (PWM 信号) は，悪用防止システムの条件演算子を介して出力される．UAV の悪用を検知した場合，OSS 制御プログラム側からの制御信号を悪用防止システムが遮断し，悪用防止システムにより生成された制御信号に置き換えモータを制御することで制御を奪取する．

3. 悪用防止プロセッサの設計要件

本章では，前章で述べた悪用防止機能を実現する悪用防止プロセッサの設計要件を示す．

要件 1：OSS の有用性を阻害せずに悪用を防止

OSS の悪用防止の際に，OSS 利用のためのライセンス制度の実装などによって OSS の利用に制限をかけるという対策は，OSS 本来の利点である研究・開発の促進を妨げてしまう．そのため，OSS の悪用のみを防止する手法を提案する必要がある．

要件 2：開発者自身が悪用の基準を設定可能

本研究で述べる OSS の悪用の基準とは，OSS を搭載した UAV をテロ行為等の非人道的な行為に使用することを示す．本研究では，技術者が防止できる基本機能として，重要政府機関やインフラシステム等の侵入禁止区域に対し，UAV がその区域に侵入した際に制御を奪うなど，悪用防止システムの開発者が OSS を実装した UAV を悪用されないように事前に悪用の基準を設定可能なシステムとする必要がある．なお，非人道的手段を有するペイロードを搭載するような，研究者・技術者のみでは防止困難な状況は多数存在する．これら問題は，法律家やその他の関連組織・団体との継続的な議論・対策の提案は必要ではある．

要件 3：ユーザによって無効にすることができない

OSS の悪用を防止するにあたり悪用防止システムを OSS 上に搭載した場合，その部分のソースコードを削除することで簡単に悪用防止システムを無効化できてしまう．そのため，エンドユーザによって簡単に悪用防止システムを無効化できないような設計を行う必要がある．

4. 悪用防止プロセッサの試作

前述した設計要件をもとに悪用防止システムを搭載した悪用防止プロセッサの試作を行う．悪用防止プロセッサの開発では，信号線の切断やプログラムの消去による悪用防止システムの無効化を防ぐために，ハードウェア上に悪用防止システムを設計する．最終的には，一つのプロセッサにユーザのアクセス不可能領域とアクセス可能領域を設け，アクセス可能領域にのみユーザが OSS を含むプログラムを書きこむことが可能なプロセッサとする．ここでは，UAV の悪用防止システムの枠組みを検証するため，二つの FPGA(Field-Programmable Gate Array) を用い，一方を悪用防止システム，もう一方を OSS 搭載可能領域とみなし，プロトタイプを製作する．製作した悪用防止プロセッサの構成図を Fig. 3に示す．同図に示すように FPGA (1) を悪用防止システム搭載領域，FPGA (2) を OSS 書き込み領域とする．それぞれのシステムの詳細を以下に示す．

4.1 OSS による単一ロータ制御システム

OSS の UAV 飛行制御プログラムの実装を想定し，FPGA(2) のユーザアクセス可能領域に単一ロータの高度制御システムを仮想的な OSS による飛行制御プログラムとして実装する．外部からの指令値により，同制御システムは高度制御に必要な制御指令値を決定し，出力する．このとき，単一ロータ制御システムから出力される制御信号線は，ESC(Electronic Speed Controller) に直接接続せず，Fig. 3に示すように悪用防止システムを経由して，ESC へ出力する．プロトタイプでは，FPGA を二つ使用しているため，物理的な信号線が露出しているが，最終的には一つのプロセッサ内にアクセス不可能領域と可能領域を設けるため，プロトタイプの段階ではこの点は問題としない．エンドユーザは，ユーザアクセス可能領域のみを認識して開発を行うため，エンドユーザは出力信号が悪用防止ステムを経由した後に出力されているという認識は生じない．

4.2 強制軟着陸を可能とする悪用防止システム

悪用防止システムは FPGA(1) のユーザアクセス不可能領域に実装する．同領域はプロセッサ開発者のみがアクセスすることが可能であり，同領域内に開発者によって任意の悪用防止機能を実装する．本研究では，

最も基本的な侵入禁止区域への侵入を悪用とみなした悪用防止機能を実装する．

悪用防止システムには，悪用の監視機能と OSS 側の制御信号の奪取による強制軟着陸制御機能を搭載する．ここでは，監視機能として，位置情報を監視する．ここでも，前節同様に位置計測センサはユーザアクセス不可能領域の外部より接続しているため，物理的な信号線が露出しているが，プロトタイプの段階ではこの点は問題としない．同図より悪用防止システムでは，上位制御と判断をインテル社製ソフトコアマイクロプロセッサである Nios II に実装し，悪用防止機能の起動を担う根幹部をハードウェア記述言語である Verilog HDL で記述する．ハードウェア上で直接悪用防止機能を実行するため，仮にユーザアクセス不可能領域の改ざんを試みようとしても通常のエンドユーザでは，その実現は困難となる．強制軟着陸制御システムには，PID 制御をここでは実装する．

Algorithm 1に本研究で開発した悪用防止プロセッサの疑似コードを示す．Fig. 3に示すように，Nios II 内で位置情報を取得し，その情報に基づき侵入禁止領域への侵入の可否を判断する．ここでは，次章で示す強制軟着陸の基礎検証実験環境に従い，y 軸方向の侵入領域を指定し，侵入可否を判断する．なお，位置情報の取得方法として，GPS 情報の取得による実行も可能であったことは確認済みである．次に，侵入を検知したら，目標高度を徐々に下降させる PID 制御を実行し，悪用防止システムからの制御信号を出力する．同制御信号は，Verilog HDL で記述されたハードウェア側に入力される．また，侵入可否の判断結果および OSS の FC からの制御信号が入力され，侵入可否の判断に基づき，制御信号の出力を強制的に切り替える．

Algorithm 1 Anti-Abuse System Algorithm

Nios II

Input $Axis\_Data = X, Y, Z$
Output **PWM(from Anti-abuse)**
X_Data $\leftarrow Axis\_Data = X$
Y_Data $\leftarrow Axis\_Data = Y$
Z_Data $\leftarrow Axis\_Data = Z$
procedure MAIN
  while true do
    if $Y\_Data > Y\_ref$ **then**
      STOP_Signal $\leftarrow$ **HIGH**
      $Z\_ref \leftarrow Z\_Data$
      break
    else
      STOP_Signal $\leftarrow$ **LOW**
    end if
  end while
  while true do
    $Z\_ref \leftarrow Z\_ref - \Delta Z$
    $e \leftarrow Z\_ref - Z\_Data$
    $e\_s \leftarrow e\_s + e \times dt$
    $F\_Z \leftarrow k_P e + k_I \int edt + k_D \dot{e}$
    $F\_Z \leftarrow F\_Z + F\_H$
    $Convert\ from\ F\_Z\ to\ Duty\ of\ PWM$
    if $Z\_Data == Ground$ **then**
      PWM(from Anti-abuse) $\leftarrow 0$
    else
      PWM(from Anti-abuse) $\leftarrow Duty$
    end if
  end while
end procedure

Verilog HDL

Input **STOP_Signal**, **PWM(from OSS)**, **PWM(from Anti-abuse)**
Output **PWM**
procedure MAIN
  while true do
    if **STOP_Signal** == **HIGH** **then**
      PWM $\leftarrow$ **PWM(from Anti-abuse)**
    else
      PWM $\leftarrow$ **PWM(from OSS)**
    end if
  end while
end procedure

5. 強制軟着陸の基礎検証実験

本章では，位置情報計測センサを搭載した悪用防止プロセッサのシステムの動作検証を行う．ここでは，今後繰り返し基礎検証を実施するために，モーションキャプチャを用いた屋内実験を実施する．なお，前述の通り，本来の屋外飛行で必要となる GPS 情報取得等の基礎システムの動作検証は実施済みである．

5.1 実験装置

屋内検証実験で使用する実験装置を Fig. 4に示す．悪用防止プロセッサについては，Fig. 3に示す構成で開発したシステムを使用し，悪用防止システム内の Motor Control 内に前述の高度制御システムを実装する．検証実験では，UAV 飛行の前段階として単一ロータをリニアガイドに搭載し，簡易的な UAV とみなし，その高度制御と悪用防止システムによる強制軟着陸制御を行う．

5.2 実験結果

実験時の様子を Fig. 5に示す．ここでは，y 軸方向が 3.0 m 以上を侵入禁止区域とみなす．まず，Fig. 4における FPGA (2) 内の OSS による FC によって単一ロータをホバリングさせる．その後，台車に乗せた実験装置を侵入禁止区域へと侵入させたところ，FPGA (1) の悪用防止システムが制御を奪い，模擬機体を強制軟着陸させることに成功した．

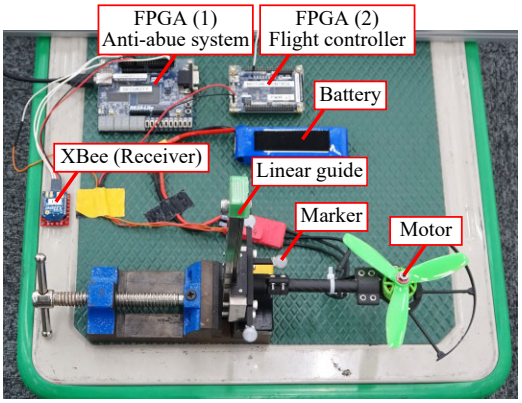


図 4: Experimental device (Motion capture)

Fig. 6にモーションキャプチャより取得した模擬機体の y 軸および z 軸の軌道を示す．高度方向の z 軸座標において，初期状態が約 0.24 m となっているのは台車の高さ情報を含むためである．同図より，約 8.4 s 付近で侵入禁止区域へ侵入し，その検知後，強制的に軟着陸を始めるまでの時間は約 0.24 s，機体が高度を下げ

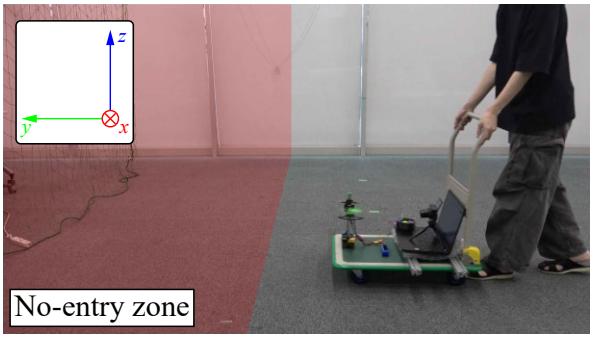
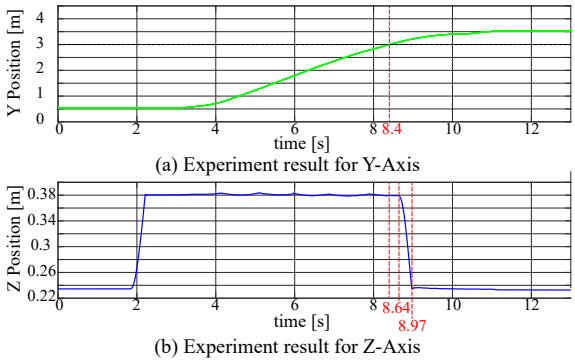


図 5: Verification experiment



(a) Experiment result for Y-Axis

(b) Experiment result for Z-Axis

図 6: Experimental result

始め着陸するまでの時間は約 0.33 s となった．機体がホバリング状態の時の高度は約 0.38 m，着陸時の高度が約 0.24 m であることから，機体が軟着陸する際の速度は約 0.42 m/s であることが分かる．

5.3 考察

モーションキャプチャによる屋内検証実験より，悪用防止プロセッサを用いて侵入禁止区域に侵入後にモータの制御を奪取し，強制的に軟着陸させる悪用防止システムの有効性について確認することができた．同結果より，あらかじめ飛行禁止としている区域内では UAV の飛行を制限可能であることを示すことができた．一方，先行研究 [10] では，スイッチ機能による悪用検知から制御を奪取するまでの時間は約 300 µs であったが，本実験では機体が侵入禁止区域に侵入後に悪用防止システムが制御を奪取するまでの時間は約 0.24 s ほどであった．同遅延の発生原因として，モーションキャプチャによる計測と Zigbee による通信，PID 制御の Nios II 内での計算時間，Nios II と Verilog HDL 間での通信時間等が考えられるが，詳細な検証分離には至っていない．今後，割込み処理や Nios II 側での制御系と Verilog HDL 側のハードウェアにて実現するコード整理を行い，検知後の制御奪取を高速に実現するための詳細検証と開発を実施する必要がある．

しかし，現段階であっても，悪用検知後に制御を奪うまでに約 0.24 s の遅延があるが，UAV が時速 100 km/h であったとしても侵入禁止区域に約 6.67 m ほどの侵入で強制軟着陸は可能であることが分かる．侵入禁止区域の安全マージンを本来のものより数 m 程広く設定することで，本研究で開発した悪用防止プロセッサで十分侵入を防止することが可能だと考えられる．

6. 結言

本稿では OSS で動作する UAV の悪用防止を目的とした悪用防止プロセッサの試作とその基礎検証について述べた．UAV の悪用防止手法として，位置情報より，あらかじめ設定した禁止区域に侵入後に機体の制御の奪取により軟着陸を行うシステムを開発した．また，設計した悪用防止システムの有効性の検証として，単一ロータをリニアガイドに取り付けた模擬 UAV を用いて基礎検証実験を実施した．検証実験では，モーションキャプチャシステムより取得した位置情報をもとに侵入禁止区域での模擬機体の強制軟着陸の動作の実現を確認した．これにより，侵入禁止区域侵入後に機体の制御を奪い，強制軟着陸させることが十分可能であることを示した．

今後は，悪用検知から制御奪取までの時間遅延の軽減，実 UAV への搭載と屋外実験を実施し，より現実に近い検証を実施していくとともに，悪用の定義，技術者・研究者ができる防止策の議論を継続的に進めていく．

参 考 文 献

[1] E. Alvarado：“Drone Market Analysis 2022-2030,” Drone industry insight, accessed on July 10th, 2024, available from https://droneii.com/drone-market-analysis-2022-2030.

[2] United Nations Office of Counter-Terrorism：“Global Report on the Acquisition, Weaponization and Deployment of Unmanned Aircraft Systems by Non-State Armed Groups for Terrorism-related Purposes,” UNOCT AROS Programme and Conflict Armament Research, 2021.

[3] B. Heubl：“Consumer drones used in bomb attacks: Terrorist rebels are turning consumer drones into deadly weapons. ET investigates why it goes on and what can be done about it,” Engineering & Technology, vol.16, no.4, pp.1–9, 2021.

[4] 河：“ドローンの衝撃”, 株式会社扶桑社, 2015.

[5] DJI： “Drone Security White Paper,” accessed on July 10th, 2024, available from https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/trustcenter/DJIDroneSecurityWhitePaper.pdf

[6] Cybersecurity and Infrastructure Security Agency：“CISA Open Source Software Security Roadmap,” accessed on July 10th, 2024, available from https://www.cisa.gov/sites/default/files/2024-02/CISA-Open-Source-Software-Security-Roadmap-508c.pdf.

[7] Pixhawk： Pixhawk. Accessed on August 5th, 2023. Available from https://pixhawk.org.

[8] 橋本，石神，佐藤，辻田，安孫子：“オープンソフトウェアの悪用を防止可能なロボット開発フレームワークの提案”, 第 21 回システムインテグレーション部門講演会予稿集, pp. 24—28, 2020.

[9] 橋本，佐藤，辻田，安孫子：“オープンソースソフトウェアを用いたロボットの悪用を防止可能なセキュアプロセッサの試作”，ロボティクス・メカトロニクス講演会講演概要集, 1P1–D07, 2022.

[10] 伊藤, 安孫子, 福田, 辻田, 佐藤：“悪用防止プロセッサによるオープンソースソフトウェアで制御された UAV の強制軟着陸の基礎検証”, ロボティクス・メカトロニクス講演会概要集, 1P1-C04, 2024.