

PDF issue: 2025-06-23

# 船舶機関室におけるドローン自動巡視システム構築 の検討

油木,代一 内田,誠

三輪, 誠

(Citation) 神戸大学大学院海事科学研究科紀要,16:62-71

(Issue Date) 2019

(Resource Type) departmental bulletin paper

(Version) Version of Record

(JaLCDOI) https://doi.org/10.24546/81011875

(URL) https://hdl.handle.net/20.500.14094/81011875



# 船舶機関室におけるドローン自動巡視システム構築の検討 Fundamental Study on Drone Auto Patrol System for Ship Engine Room

油木代一\* 内田誠\*\*\* 三輪誠\*\*\*\* Daiichi ABURAGI, Makoto UCHIDA, Takashi MIWA

(令和元年7月11日受付)

## Abstract

In this paper, we investigate the possibility of a drone auto patrol system for a ship engine room using a small indoor drone without GPS function controlled by PC with Wi-Fi connection.

A monitoring operation of equipment meters and screens on the engine control console in the MEPS (Marine Engine Plant Simulator equipped in Kobe University) control room was incorporated into a drone automatic patrol system and trial operation was carried out.

This fundamental study in the simulator environment is intended to the development of technologies related to monitoring and operation management via satellite communication in MASS (Maritime Autonomous Surface Ships) for the future.

(Received 11 July, 2019)

## 1. はじめに

昨年は地震や豪雨、また台風や高潮など、自然被害が多発した。高潮による神戸港へのコンテナの流 出事故<sup>[1]</sup>など、海事産業に関連する影響も大きく、我々の記憶に新しい。一般的にドローンは、プログ ラム制御の無人飛行による撮影ができることから、これら自然災害が発生した後、被災状況の把握手段 の一つとして期待される。既にドローンの活用は様々な分野で進んでおり<sup>[2]</sup>、高所構造物の測量の場面 では、従来、人による点検・計測が行われていたが、ドローンによる撮影・取得映像の画像解析により、 現場の労力軽減と測量技術の向上に貢献している<sup>[3]</sup>。今後益々、ドローンの需要が高まることが予想さ れる。

本稿では、GPS 機能を持たない屋内向け小型ドローンを用いて、ドローン巡視プログラムを作成、PC からドローンに Wi-Fi 接続し、船舶機関室環境で自動巡視させるシステムの構築を検討する。

本研究科の教育研究設備である MEPS (Marine Engine Plant Simulator)の模擬制御室にて、システ ムの動作検証を行い、ドローンによる各機器の監視状況を確認することで、システムの課題や発展性を 探る。そして、将来の自律航行船において、ドローンが、機関室の様子をモニタリングした映像データ を、セキュリティの確保された衛星通信を介して送信、共有することで、陸上からの船舶運航管理支援 <sup>(4)</sup>などに貢献できる可能性に期待する。

\*\* 大学院海事科学研究科 技術部 \*\*\* 大学院海事科学研究科 マリンエンジニアリング講座 \*\*\*\* 海洋教育研究基盤センター

# 2. システム構成と主な構成要素

本システムを構成するハードウェア及びソフトウェアを紹介する。ただし、ソフトウェアは、使用するハードウェアによって異なることに注意する必要がある。システム構成を図1に示す。



図 1. システム構成

# 2-1 ドローン (Tello EDU)

Tello EDU は、ビジョンポジショニングシステムおよび前方カメラを搭載する屋内の飛行に適した小型軽量かつ低価格のクアッドコプターである。ビジョンポジショニングシステムの主なコンポーネントは、下方カメラと 3D 赤外線モジュールとなっており、その場で正確にホバリングが可能になっている。

SDK 2.0 を搭載する Tello EDU は、DJI 社のフライトコントロール技術を採用し、専用操縦アプリ だけでなく様々なプログラミング言語からも操縦することが可能となっている<sup>[5]</sup>。EIS(電子式映像ブレ 補正)にも対応しているためカメラ映像も滑らかに表示できる。機体写真を図 2、3、主な仕様を付録 A に示す。

# 2-2 タブレット PC

主に、ドローンに自動巡視させるためのプログラムの作成、実行を行い、ドローンの前方カメラからの映像をリアルタイムに表示、録画するために用いる。本タブレット PC は、折り畳み部分で切り離し、タッチパネル対応タブレットだけで動作することも可能であるため、取扱いや持ち運びに便利である。 さらに長時間のバッテリー駆動が可能で、ドローンと Wi-Fi 接続するので、システム全体をコードレス 化できるメリットがある。本体写真を図 4、主な仕様を付録 B に示す。



図 2. Tello EDU 上面



図 3. Tello EDU 下面



図 4. タブレット PC

#### 2-3 ミッションパッド

Tello EDU1 台に付属しているフレキシブルなミッションパッド は、計4枚で、各両面に1~8のいずれかの番号がID として大きく 白印字されている。ミッションパッドは、Tello EDU を操縦する上 で必須ではないが、機体が画像認識、反応可能な物理的ウェイポイ ントであるため、飛行精度を向上させる効果がある。図5が示すよ うにミッションパッドには座標系が定義されている。ロケットの絵 が示す向きがX軸正の向き、パッド上方がZ軸正の向きとなり、右 手座標系から必然的にY軸が決まる。これを基にドローンの下方カ



メラによってミッションパッドを画像で捉え、Intel Myriad 2 VPU 図 5. ミッションパッドと座標系 による画像処理を経て機体位置の誤差をパッド上で自動修正することが可能となっている<sup>[6]</sup>。

#### 2-4 Go 言語 (Ver 1.12.3)

Go 言語は、Google とオープンソースコミュニティからの協力者によって開発されたオープンソース のプログラミング言語<sup>[7]</sup>であり、Windows、Linux マシン、Mac など複数のプラットフォームで動作 し、コンパイル・実行速度が早い、安全性が高い、並列処理の記述が容易などの特徴を有している<sup>[8]</sup>。

#### 2-5 Gobot (Ver 1.11)

Gobot は、ロボット工学、フィジカルコンピューティング及びモノのインターネット (IoT) 向けの Go 言語で記述されたフレームワークである。また、Gobot は、低レベルの Arduino や Raspberry Pi のさ まざまな物理デバイス、さらにはドローン、玩具などのデバイスを制御するためのドライバとアダプタ が提供されている<sup>[9]</sup>。

## 2-6 MPlayer (r38119)

マルチプラットフォームで動作するオープンソースのメディアプレーヤーである<sup>[10]</sup>。ドローンより送 られてくるビデオストリームをタブレット PC で受信し、このプレーヤーに送り込むことでカメラ映像 を表示する。本システムではプログラムよりコマンドラインを実行させることで MPlayer を動作させ ている。詳細は付録 D のドローン巡視プログラムに示す。

#### 3. システムに関する準備

Tello EDU は、購入した段階でプロペラ及びプロペラガードなどが組立てられた状態になっている。 フライトバッテリーを機体に挿入して充電しておく必要がある。タブレット PC ではドローン巡視プロ グラムを作成し実行できるようソフトウェア環境を準備する。ソフトウェア環境セットアップ手順を付 録 C に示す。

ドローンから取得する映像の視認性、飛行の再現性など基本的事項を確認することを目的に、実際の 船舶機関室よりも立体物が少なく、実験室レベルでの検証が可能な本研究科の MEPS 模擬制御室を巡 視対象空間とし、ドローンに各種機器を自動巡視させる。

#### **3-1 MEPS 模擬制御室**<sup>[11]</sup>

船舶運航シミュレータ (Marine Simulator) は、2010 年 3 月に神戸大学大学院海事科学研究科へ導入された教育研究設備であり、船橋における操船を模擬するナビゲーションシミュレータ (Navigation Simulator) と推進動力機関プラントの運用管理を模擬する舶用機関プラントシミュレータ (MEPS) から構成される。それらは、連携して使用することも、それぞれ単独で使用することも可能である。船 舶運航シミュレータ全体図を図 6 に記す。

MEPS は、外航船舶である大型コンテナ船(機関定格出力 43,840 kW)の機関プラントをモデルにしており、主機関、発電機、ボイラー、排ガスエコノマイザーなどの主要船舶機器の動作を、コンピュー

タシステムにより仮想的に実現する。MEPS は、2 室で構成されて、一方には模擬機関室、他方には模 擬制御室とインストラクターコンソールが配置されている。

模擬機関室では3画面のタッチパネルディスプレイが配置され、主機、補機、各種配管系統等がイラ ストと共にそのディスプレイ上に表現される。画面上のタッチ操作により各機器の始動や停止操作が可

能であり、各部配管温度や流体圧 カの選択表示と不表示が行える。 機関プラントを構成する機器に おいて異常が生じると、模擬制御 室では警報とともに異常内容が エンジンモニタへ表示される。ま た模擬機関室のパトロールラン プが連動して点灯する仕組みに なっている。

模擬制御室は、ECC (Engine Control Console)、GSP (Group Starter Panel)、主配電盤、電力



制御盤、エンジンモニタから構成 図 6. 船舶運航シミュレータ全体図(赤枠内は模擬制御室) されており、ECCは、実機と同等のハードウェア形態となっているが、その他 GSP 等は設置空間にお ける制約によりタッチパネル操作式のコンパクトな装置構成となり、実機とはハードウェア形態で大き く異なる。エンジンモニタは、模擬制御室とインストラクターコンソールに同じ構成のものが配置され ており、模擬機関室で計測するデータを数値やトレンドグラフとして参照することや警報発生状況等も 確認できる。

#### 3-2 自動巡視ルート

自動巡視ルートを図7に示す。各直線ルートに①~⑧の番号を付与している。模擬制御室の装置全体の左半分が ECC、中央の PC モニタがエンジンモニタ表示部、右端上部にあるタッチパネルが電力制御盤等の表示部となっている。次に、ミッションパッドの配置を図8に示す。床に3枚、卓上に2枚、計5枚をウェイポイントとして配置している。そして、IDが1番のミッションパッドをホームポジションとし、ドローンを離陸させてから、ECC、エンジンモニタ、電力制御盤の順で自動巡視させる。巡視を終えればホームポジションに着陸させる。



図 7. 自動巡視ルート

図 8. ミッションパッド配置

#### 3-3 ドローン巡視プログラム

自動巡視ルートに基づいて作成したプログラムと直線ルートに関するプログラムコード解説を付録 Dに示す。タブレット PC から Tello EDU に Wi-Fi 接続し、プログラムは、タブレット PC 上で実行す る。Tello EDU に送るコマンドやビデオストリーム等のデータは、UDP (User Datagram Protocol) で 通信される。タブレット PC にインストールした Gobot にはフライトに必要な API (Application Programming Interface) 関数が用意されている。さらに、Tello EDU に関するドライバを初期化する プログラムや MPlayer でカメラ映像を表示するプログラムが公開されている<sup>[9]</sup>ので、それらを組み合わ せて雛形として活用し、ユーザーは、操縦部分の処理だけ追加コーディングすることでプログラムを作 成することができる。操縦部分の処理は、飛行空間、監視対象物及び飛行ルートによって異なる。

#### 4. システム動作検証

付録 D で示したドローン巡視プログラムを用いて自動巡視の動作検証を行った。記録映像<sup>[17]</sup>から、 ECC、エンジンモニタ、電力制御盤を監視している場面について、順番に切り出した画像を図 9、 10、11 に示す。システム動作手順は付録 E に示す。

ドローン取得映像に関して、ECCにおける計器類の針の位置、インジケータランプの状態が確認できた。エンジンモニタ、電力制御盤では、表示データの項目名、数値といった小さな文字も読み取れた。 本検証実験の結果から映像の視認性は高く、実用上支障ない映像品質と考える。

ドローンの飛行に関して、卓上でのホバリング時は、機体は安定した姿勢を保てず動揺している様子 が確認できた。機体から卓上面までが近くなっており、プロペラから生み出される空気の流れが跳ね返 されるため、機体はその影響を受けやすくなったものと推察する。また、模擬制御室にてドローン自動 巡視を繰り返し試みたところ、飛行距離の誤差により指定ルートから少し外れ、ミッションパッドを検 出できず、機体がその場でホバリングし続けるような制御不能に陥ることがあった。実船での自動巡視 を想定すると、飛行に関する精度、再現性は、非常に重要であると考えられる。今後、ドローンの飛行 速度、時間に関するパラメーターの調整などプログラムの改良で飛行に関する精度、再現性を高めるこ とが課題である。

飛行時の機体騒音は、実際の船舶機関室の騒音環境では聞き取れないほどの低レベルの騒音と思われる。船舶機関室で巡視させたとしても特段の支障はないものと考える。

模擬制御室は、実際の制御室とハードウェア形態がよく似ており、模擬制御室のエンジンモニタでは、 ECC アナログ計器の指示値も含め、他の計測データも数値データとして表示、管理可能である。通信技 術を使えば遠隔サイトでも情報共有が容易であることから、ドローンで実際の制御室をモニタリングす る必要性は低いと考える。一方、実際の機関室では、それら計測データだけでは判別できない配管から の液漏れ、装置からの発煙、固縛から解かれた移動物、塗装剥がれ、サビの進行といった物体の性質、 表面形状、色の変化などの異状を視覚的に発見することが可能であり、乗組員による五感を駆使した高 度な巡視には及ばないものの映像取得できるドローンの活用は機関室巡視に有効と考える。



図 9. ECC 計器類

図 10. エンジンモニタ

図 11. 電力制御盤

## 5. おわりに

本システムにおける巡視プログラムの開発環境は、短時間で構築できる上に SDK2.0 コマンドをはじめ Gobot における API 関数が充実しているので、様々な飛行アクション、ルートをプログラムで表現し、実行可能である。システム拡張に関して、現在は PC とドローンの1対1接続であるが、ブロードバンドルーターを介して、1対多接続を行い、同時に複数のドローンを制御することで、巡視対象を拡大し、巡視の効率化についても探求したいと考える。

従来における映像取得による監視は、ドローンではなくとも、壁や天井に取付けられたビデオカメラ を用いても可能であったが、特に、実際の機関室のように様々な装置、機器等の立体物が数多く存在す る空間では、監視箇所が多ければ、その都度設置するカメラの台数は多くなり、監視箇所の変更、増加 に伴い、カメラの移設、増設が予想される。しかしながら、ドローンであれば、大きなシステム変更や コスト負担を伴わず、立体物に対する回り込みによって監視を行うなど、柔軟かつ多角的な監視が可能 になり、その特徴がドローンを用いた自動巡視システムにとって大きな長所になりえるものと考える。

機関室、機関制御室の有人運用管理環境において、巡視ドローンは、乗組員の業務に新たな負担を与 える存在になり得るかも知れないが、機体の衝突防止、室内停電時における赤外線カメラ若しくは局部 照明などの機能追加、バッテリー持続時間、人手を要しないバッテリー充電、飛行に関する精度向上な どの性能改善、予備機の配備による冗長性を持たせたシステム耐障害性の確保や機体の自動・手動操作 切替え対応などにより、巡視の役割を担い船舶の安全な運航に貢献できる期待がある。特に、将来予想 される自律航行船の無人運用管理環境下では有益な存在になる可能性があると考える。

#### 参考文献

[1] https://www.kobe-np.co.jp/news/bousai/201809/0011659204.shtml

- [2] <u>http://www.fddi.jp/business/example</u>
- [3] https://www.kensetsu-plaza.com/kiji/post/13925
- [4] http://www.mlit.go.jp/common/001215815.pdf
- [5] https://www.ryzerobotics.com/jp/news/announcing-the-programmable-tello-edu-drone-page
- [6] <u>http://drone-next.jp/special/new\_item/tello-2/</u>
- [7] <u>https://golang.org/project/</u>
- [8] <u>http://golang.jp/about\_go</u>
- [9] <u>https://gobot.io/documentation/getting-started/</u>
- [10] <u>http://mplayerwin.sourceforge.net/</u>
- [11] 油木代一、内田誠、三輪誠、MEPSの概要と高度な設定自由度、神戸大学大学院海事科学研究科紀 要第 12 号、P23-29(2015)
- [12] <u>https://www.ryzerobotics.com/jp/tello-edu/specs</u>
- [13] <u>https://online.univ.coop/catalog/files/item/80194072.pdf</u>
- [14] https://qiita.com/hfujikawa77/items/28f30791b32fc909006d
- [15] <u>https://aviutl.info/game-dvr/</u>

[16]<u>https://support.xbox.com/ja-JP/xbox-on-windows/system-requirements/pc-hardware-</u>requirements-for-game-dvr

[17] <u>http://www2.kobe-u.ac.jp/~aburagi/kiyou/cr2.mp4</u>

# 付録A Tello EDU の主な仕様<sup>[12]</sup>

Aircraft	
Weight	87g (Propellers and Battery Included)
Dimensions	98×92.5×41 mm
Propeller	3 inches
Built-in Functions	Range Finder, Barometer, LED, Vision System, 2.4 GHz
	802.11n Wi-Fi, 720p Live View
Flight Performance	
Max Flight Distance	100m
Max Speed	8m/s
Max Flight Time	13min
Max Flight Height	30m
Battery	Detachable Battery: 1.1Ah/3.8V
Camera	
Photo	5MP (2592x1936)
FOV	82.6°
Video	HD720P30
Format	JPG(Photo); MP4(Video)

# 付録 B タブレット PC の主な仕様<sup>[13]</sup>

Manufacturer	Microsoft
Product Name	Surface Book 2
Product Number	HNQ-00012
OS	Win10 Pro(64bit)
CPU	Intel® Core <sup>TM</sup> -i7
Main Memory	16GB (On-board)
Storage	SSD 1TB
LCD	13.5 Inch PixelSense <sup>™</sup> Display
Graphic Device	NVIDIA GeForce GTX 1050 discrete GPU (2GB GDDR5)
Wireless LAN	802.11a/b/g/n/ac
Battery Run-time	Capable of continuous video playback for up to about 17 hours
Dimensions	312mm×232mm×23mm
Weight	1642g

# 付録 C ソフトウェア環境セットアップ手順[14]

インターネット接続済みのタブレット PC に、次に示す手順にてセットアップを行う。ただし、本 セットアップ手順は、本稿でのシステム構成要素の組み合わせが前提になっていることに注意する必 要がある。

- 1) Go 言語 (go1.12.1.windows-amd64.msi) のインストール
- インストール先は変更せず既定の C:¥Go とする。2) Windows の環境変数 Path に C:¥Go¥bin を追加

3) Git\*(Git-2.21.0-64-bit.exe) のインストール\*Gobot を導入するためのバージョン管理ソフト

- Gobot のインストール
   Windows コマンドプロンプトにて >go get -d -u gobot.io/x/gobot/... を実行する。
- MPlayer (mplayer-svn-38119-x86\_64.7z) のインストール 解凍したフォルダ内のファイルを C:¥Go¥bin 直下に配置する。
- 6) ウィルス対策ソフトは Windows Defender を、ファイアウォールは Windows Defender ファイ アウォールを用いる。後に作成するドローン巡視プログラムを実行する時に Windows Defender ファイアウォールでドローンとの通信を許可する。あるウィルス対策ソフトのファイ アウォールでは、通信を許可する設定にしても PC とドローンが通信できないことがあり、ア ンインストールしてはじめて通信できるという状況を確認した。
- 付録D ドローン巡視プログラムおよび直線ルートに関するプログラムコード解説

```
package main
import (
    "fmt"
    "os/exec"
    "time"
    "gobot.io/x/gobot"
    "gobot.io/x/gobot/platforms/dji/tello"
)
func main() {
    drone := tello.NewDriver("8890")
    work := func() {
        //カメラ、MPlayer 設定・起動
mplayer := exec.Command("mplayer", "-fps", "30", "-")
         mplayerIn, _ := mplayer.StdinPipe()
        if err := mplayer.Start(); err != nil {
             fmt.Println(err)
             return
        }
        drone.On(tello.ConnectedEvent, func(data interface{}) {
             fmt.Println("Connected")
             drone.StartVideo()
             drone.SetVideoEncoderRate(4)
            gobot.Every(100*time.Millisecond, func() {
    drone.StartVideo()
             3)
        })
        drone.On(tello.VideoFrameEvent, func(data interface{}) {
             pkt ≔ data.([]byte)
             if _, err := mplayerIn.Write(pkt); err != nil {
                 fmt.Println(err)
            }
        })
        //ミッションパッド検知機能有効化
        gobot.After(1*time.Second, func) {
                    drone.SendCommand("mon")
                    drone.SendCommand("mdirection 0")
          })
        //操縦部分の処理(飛行アクション・ルートの指定など)
        gobot.After(23*time.Second, func) {
                    drone.TakeOff()
          })
                                                                                         ① 直線ルート
         gobot.After(27*time.Second, func() {
                    drone.Up(20)
          })
```



直線ルート	プログラムコード解説
1)	プログラム実行 23 秒後に離陸、その4秒後に速さレベル 20(0~100) で2秒間上昇し
	てから 3 秒間ホバリングさせる。プログラム実行から離陸までの 23 秒間は、ドロー
	ンのカメラ映像を手動録画するための準備時間である。
	ミッションパッド1から2の位置に移動させるためのルートで、ミッションパッド1
2	の座標系における X 軸 0cm、Y 軸-95cm、Z 軸(高度)130cm を目標に速さレベル 15 で
	移動させる。移動中、ドローンのカメラ映像より ECC の計器類が確認できる。その
	後、ミッションパッド2を下方検知したところでホバリングさせる。
3	速さレベル 20 で 3 秒間前進させてから一旦ホバリングさせる。その後、ミッション
	パッド3を下方検知して高度 35cm の位置で機体をホバリングさせる。この位置でド
	ローンのカメラ映像よりエンジンモニタの表示内容を確認できる。
	ミッションパッド3から4の位置に移動させるためのルートで、ミッションパッド3
4	の座標系における X 軸 0cm、Y 軸-90cm、高度 45cm を目標に速さレベル 15 で移動
	させて、ミッションパッド4を下方検知したところでホバリングさせる。この位置で
	ドローンのカメラ映像より電力制御盤の表示内容を確認できる。
5	速さレベル 20 で 4 秒間後進、ホバリングさせる。
6	ミッションパッド5上の高度 100cm まで降下、機体をホバリングさせる。
Ţ	ミッションパッド5から1の位置に移動させるためのルートで、ミッションパッド5
	の座標系における X 軸 0cm、Y 軸 170cm、高度 50cm を目標に速さレベル 20 で移動
	させて、ミッションパッド1を下方検知したところでホバリングさせる。
8	ホバリングしているところからミッションパッド1に着陸させる。

# 付録 E システム動作手順

ミッションパッドが、ウェイポイントとして、所定の位置に、ドローンが、ホームポジションに配 置されていることを前提とする。

- 1) ドローンの電源を入れる。
- タブレット PC とドローンを Wi-Fi 接続する。
   一般のブロードバンドルーター(アクセスポイント)に Wi-Fi 接続する要領と同じである。
- 3) Windows コマンドプロンプトにて、ドローン巡視プログラムがあるディレクトリに移り、実行 する。

実行コマンド形式: >go run ソースファイル名

実行形式 exe ファイルを作成する場合は >go build ソースファイル名

4) ドローンのカメラ映像を録画する。

プログラムの実行が成功すると、コマンドプロンプト内で Connected というメッセージが表示 され、MPlayer が起動する。MPlayer にはドローンのカメラ映像がリアルタイム表示される。そ して、筆者が利用した MPlayer の録画方法は、Windows10 に標準装備されているデスクトップ キャプチャ機能 Game DVR<sup>[15]</sup>である。ただし、この録画方法を可能にするには、PC のビデオカ ードが一定のハードウェア要件<sup>[16]</sup>を満たしている必要がある。