ドローン利活用の安全性の確保における企業情報システム開発技術の適用

細金万智子[†1]　青木善貴[†1]　星野隆之[†1]

安心安全かつ快適な企業情報システムを実現するため，我々はドローンをはじめとする移動型デバイスを効果的に制御する方法を研究してきた．移動型デバイスの制御プログラムにモデル検査を用いて網羅的な検証を行うことにより，高精度な自律制御と共に制御プログラムの品質向上と安全性が確保できると考えられる．本項では，ドローンなど，3 次元空間を移動するデバイスの安全性をモデル検査により確保する手法について提案する．

Applying Development Technology of Enterprise Systems to Ensuring Safety of Drones

MACHIKO HOSOGANE[†1]　YOSHITAKA AOKI[†1]
TAKAYUKI HOSHINO[†1]

This manuscript is a guide to apply development technology of enterprise systems to ensuring safety of drones. It is thought to ensure high level safety and quality improvement of the control program by applying 'model checking' to the control program of Drones. I suggest how to secure the safety of a device moving the three-dimensional space by 'model checking'.

1. はじめに

ドローンには墜落という危険性が常に付きまとうため，安全性の確保は非常に重要である．ユニシスはミッションクリティカルなシステムを構築してきた経験から，企業情報システムの構築で活用検討しているモデル検査をドローンの安全性確保に適用することを提案する．

ドローンは，3 次元空間を自由に移動し，あらゆる環境情報をセンシングできるというメリットをもち，エンターテイメント領域のみならず，様々な分野における利活用が進んでいる．しかしその一方で，ドローンを取り巻く環境は厳しい．安全面への配慮から，米国では，FAA（米連邦航空局）が，商業ドローンの運用基準となる規制試案（NRP : Notice of Proposed Rulemaking）[1]を発表した（表 1-1）．

表 1-1　米連邦航空局による商業ドローン規制案骨子

1	商業ドローンの重量は 55 ポンド（25kg）まで，遠隔パイロットによる日中の飛行を認める．
2	FAA による商業ドローン機体の認可は必要なし．オペレータの自己責任でドローンの安全確認，保守を行うこと．
3	飛行エリアは VLOS（Visual Line-of-sight，視線の届く範囲）に限定する．
4	飛行高度は原則 500 フィート（約 150m）以下，速度は時速 100 マイル（時速約 161km）以下とする．
5	飛行に際しての気象条件は，コントロール・センターから視界 3 マイル以上．
6	商業航空機の飛行エリア（Class A : 1 万 8,000～6 万フィート/5.5km～18km）の飛行は認めない．Class B～E（飛行場近辺）については ATC（Air Traffic Control，航空管制センター）の許可が必要
7	小型ドローン（micro UAS，重量 2kg 以下）は，クラス G（700～1,200 フィート，約 210m～365m）の飛行を認めるオプションを検討
8	商業ドローンの操縦は 17 歳以上で飛行（オペレータ）免許が必要．ただし，オペレータ免許は（パイロット免許と違い），実飛行時間および健康診断チェックは不要．免許は 2 年毎に更新．
9	オペレータは TSA（Transportation Security Administration）のチェックを受ける必要がある．
10	アクシデント（物損，人身事故）発生時には 10 日以内に FAA への報告を義務化．

この規制のため，米 Amazon 社（以下，Amazon）が実証実験中のサービスである「Amazon Prime Air」を初めとするドローンの商用利用は実現できず，狭い範囲での利用（農薬散布やインフラ設備の点検，自然環境調査など）に制限された．ドローンの自動飛行や遠隔によるセンシングも実現しにくくなった．それでも 2015 年 3 月に Amazon は，FAA から米国内での飛行実験の認可を取得し，一部サービス実施に向け進展した．しかし操縦者は医師の診断書と自家用操縦士の免許を取得している必要があり，未だ商用利用には多くの課題が残されている．

国内においては，東京都で都立公園での全面飛行禁止が条例で決定した．また，住宅密集地や空港周辺，国家重要施設上空の飛行禁止，それ以外の区域でも夜間の飛行を認めないこと等などを定めた航空法改正案が閣議決定された．

†1 日本ユニシス株式会社
Nihon Unisys, Ltd.

こういった規制が強化されるのはドローンの安全性が確保さていないためである．

ドローンの安全性が確保されれば，活用の場はさらに広がり，利便性の高く，より高精度な情報を扱うシステム構築が可能になる．

そこで本稿では，企業情報システムの品質を保証するための開発技術である「モデル検査」を利用して，ドローンの振る舞いの検証を行い，モデル検査によるドローンの振る舞いの検証の有効性について考察する．

2. ドローン利活用の意義と課題

ドローンとは，自律飛行する無人の飛行物体である．ドローンは，もともと軍事目的で発展したが，Amazon が 2013 年に宅配サービス目的での利用を発表したことをきっかけに，商用利用が急速に進み始めた．

最近では，ドローンで撮影した空撮映像と VR ヘッドセットを組み合わせたリアルな操縦体験や，ドローレースの開催，パレードやショーでの活用など，エンターテイメントにおける活用も進んでいる．ドローンの利点は，3 次元空間を自由に移動することで，ダイナミックな全方位の空撮ができることはもとより，カメラ以外にも多種多様なセンサーを搭載することによって，従来は取得できなかった 3 次元空間の環境情報の取得が可能となる．

更には，ドローンは単なるラジコンヘリコプターの範疇を超え，「空飛ぶロボット」として位置づけられつつある．具体的には知能を持ったドローンである．GPS などの位置情報をもとに自律制御する時代は終え，人間と同じように自ら考え，周囲の環境情報をセンシングし，その時々でその環境に合った適確な動作を判断する，人工知能を持ったデバイスとなることが予想され，これからのデジタル化された社会システムに欠かすことのできない存在になることは間違いない．

ドローンの活用の場が広がり，社会システムを構成する要素として認知される中で，ドローン本体の小型化・高性能化が進むとともに，安全・安心に関わる技術も重視されている[2]．具体的には以下のようなものがある．

- ドローンが何かに衝突し，墜落しそうになった時のためのパラシュート（着脱と再利用が可能）を備えた「Hubsan X4 Pro」
- 障害物回避システム「SkySpecs」
- ドローンが着陸するだけで充電が始まるマット型充電器

こうした周辺機器の進化により，ドローンの用途や可能性はさらに広がり，益々，ドローンの安全性の確保が急務かつ必須となる．

ドローンを安全に社会システムで利用するためにはドローンの振る舞いを漏れなく検証する必要がある．近年，システム開発の上流工程においてシステムが取りうる様々な状態を満たすかを検査する「モデル検査」が注目されている．このモデル検査により，ドローンが取りえる様々な状態（振る舞い）をドローンの制御コードが満たすかどうか，また意図しない状態（振る舞い）に陥る可能性がないかどうかを網羅的に検査することにより，ドローン飛行の安全性の向上が望めると考える．

3. モデル検査について

3.1 モデル検査の概要

形式手法は，数学的に厳密に意味付けられた言語を用いて情報システムの要求や設計などを記述し，情報システムがユーザの要求が満たされているかを論理的に検査する手法である．形式手法はシステムが持つ性質そのものを検証するため論理的に不具合が存在しないことの保証ができる．そのため欧米を中心に実際のシステムへの適用事例は増加している[3].

形式手法は，二つのアプローチに分けられる．ひとつはモデル検査であり，システムの振る舞いを状態遷移系のモデルに変換し，それが満たすべき性質を時相論理式で表して充足関係を検証する．モデル検査は状態遷移を全自動で網羅的に探索して検証を行ってくれる．検査した性質が満たされない場合はその満たされない状態に至る状態遷移の経緯を出力する．また検査対象の状態は有限である必要がある．代表的なモデル検査ツールは SPIN [4], NuSMV [5], Java Pathfinder(JPF)[6]，UPPAAL[7]である．

もうひとつは検証したい性質が定義された仕様によって満たされることを推論規則に基づいて証明する演繹法である．代表的な手法は VDM [8] や B-method [9]などである．検査対象の状態が有限である必要はない．検証には人による作業が必要であり，全自動での検証はできない．

本稿ではモデル検査を利用して全自動で検証ができるモデル検査を利用する．

3.2 モデル検査をドローンの検査に使う利点

作成するシステムの規模が大きくなるほどシステムが取りうる振る舞いの全てを開発者が理解することは難しくなる．開発者は特定の機能や，特定のフローのみの理解しかできなくなってしまうためテスト作業はシステムのあるべき振る舞いをテストするのではなく，個々の手続きをテストして，それを積み重ねてシステム全体の正しさを確認することが多くなる．そしてテストの期待結果は主に正常終了のケースとなるので，あらかじめ想定困難な不具合をテストケースとして作成することは難しく，また起こり得るケースを全て網羅してテストすることも不可能である．

モデル検査はシステムを有限の状態遷移で表わしたモデルに対し，要求する性質を時相論理式で記述し，モデル

がこの論理式を満せるかを網羅的に検査できるので，検査モデルさえ構築できれば，テストケースの検討をせずとも検査モデル上でシステムの振る舞いはすべて検査できるため，通常のテストでは見逃されてしまう不具合を発見できる．モデル検査の利用により通常のテストではカバーできていない部分を補完できる．

ドローンはその振る舞いの複雑さと外部環境から受ける影響が相まって信頼性の確認が難しい．近年発生している諸所の問題を踏まえると，ある程度自立性を持つドローンの安全性の担保には障害許容性,回復性が重要と考える．障害許容性とは「障害にもかかわらず，システムが意図したように運用操作できる度合い」であり，回復性とは「障害時にシステムが受けた影響を回復しシステムを希望した状態に復元できる度合い」である．

ドローンを制御するソースコードとドローンが取りうる振る舞いを関連付けてモデル化できれば上記の信頼性の性質が保障できるかを確認できると考える．不具合が発見されればそれを修正モデルとして検証し最終的にドローンに反映することによりドローンの安全性の向上が望める．発見されない場合でもモデル検査の網羅的な検証を通したことにより，不具合がないことについて一定の保障ができる．

4. 適用方法

ドローンの制御プログラムを制御フローにそってモデル化する（例：図 4-1）．ソースコードのステートメントを一つの状態として解釈してモデル化する．条件分岐の構造（例：IF 文　図 4-1①）はその条件式に基づく分岐をモデル上の分岐条件に反映する．繰り返しの構造（例：For 文 図 4-1②）の場合は繰り返す状態遷移とそれを抜け出す条件分岐の構造をモデル上に反映する．

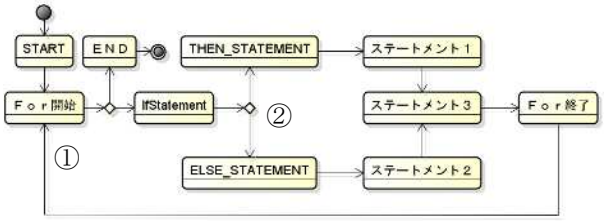

図 4-1　ソースコードのモデル例

外部環境のモデル化も行う．ドローンは各種センサーを介して外部環境から影響を受けるため，はセンサーの振る舞いをモデル化して対応する(例図 4-2)．センサーが取得する値は非決定性を用いて設定する．モデル上で設定できる型は int 型もしくは boolean 型となるため，センサー値をそのまま設定できない場合がある．その場合，モデル上のドローンの振る舞いを制御する境界値となるようにセンサー値を設定する．図 4-2 はセンサーモデルの例である．センサー値を取得する状態を繰り返す．センサー値の種類が増えれば，センサー値を取得する状態の種類が増える．

図 4-2 センサーのモデル例

これらをソースコードのモデルと結合させて検査モデルを作成する．そうすることによりドローンが実装している振る舞いをモデル化したことになる．センサー類はドローンにより装備が異なるが，一度モデル化すれば部品化して流用が可能となる．

次に仕様書などを分析して開発者がドローンにとらせたい振る舞いを抽出する．基本的に仕様は自然言語で記述されている．ここから状態を抽出してドローンの仕様のモデルを作成する．図 4-3 は例である．

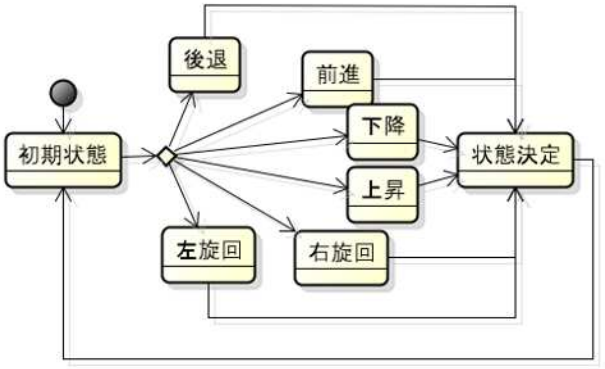

図 4-3 仕様上で取りうる振る舞いのモデル例

このドローンの取りうる振る舞いもモデル化し，先にあげたドローンが実装している振る舞いのモデルと機能を接点として接合する．もし実装している振る舞いと仕様上で取りうる振る舞いの間に不一致があれば，この検査モデル上であり得ない振る舞いの状態として検出して不具合を発見することができる．

5. 適用事例

5.1 提要事例概要

使用したドローンは仏 Parrot 社の AR Drone2.0[10]である(図-5-1)．ドローンとしては中国の DJI 社に続き，トップクラスのシェアを誇る．比較的低価格で量販店で購入可能であるため，入手し易いモデルである．ホビー用のドローンではあるが，SDK が付属しており，スマートフォンによるアプリ操作だけでなく，制御プログラムを独自に開発することができる．飛行時間は約 12 分で，制御プログラムを搭

載した端末とは Wi-Fi で通信し，動作する．ドローンには，HD カメラの他，3 軸加速度センサー，気圧センサー，3 軸ジャイロセンサー，3 軸磁気センサー，高度計測用の超音波センサーを内蔵しており，OS は Linux2.6.32 を搭載している．

図 5-1　AR.Drone

今回の適用事例の機器構成は次のとおりである(図 5-2)．人の動作を Kinect[11]で読み取り，それを端末（今回は Windows8.1 搭載 Surface Pro3）から Wi-Fi 経由でドローンへ命令を伝える．

図 5-2 機器構成

ドローンの機体制御のソースコードは C++で作成されている(図 5-3 NUL Apps)．端末側アプリケーションの構成は図 5-3 のとおりである．開発は Visual Studio 上で行っており，AR Drone 2.0 SDK を CVDrone[12]でラップしてサンプルアプリケーションより操作を行う．Kinect から取り込んだ画像はプログラム内で扱えるように OpenCV (Open Source Computer Vision Library)[13]で処理を行っている．

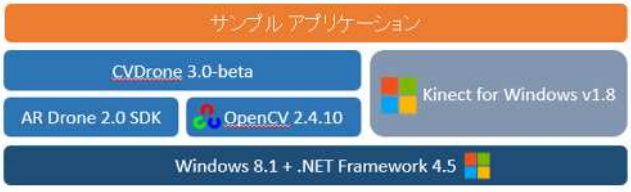

図 5-3 端末側アプリケーション構成

デモ用に作成したドローンの制御プログラムについて提案手法を適用した．

5.2 振る舞いの分析

今回デモ用に作成したプログラムであるため機能はシンプルで，操作できる機能は，離陸・着陸・左旋回・右旋回に 4 つである．仕様は次のとおりである．

(1)右手首を肩より上にあげると離陸
(2)右手首を左右肩の中央より左側に置くと左旋回
(3)右手首を右肩より肩幅の半分以上右に置くと右旋回
(4)右手首を臀部より下にさげると着陸

ドローンの振る舞いをデシジョンテーブル[14]で表わすと表 5-1 になる．デシジョンテーブルとは，プログラムの仕様を整理する際に，複数の判定条件の組み合わせと，それに対応する判定結果をまとめた表のことである．

表 5-1 デシジョンテーブル

項目	種別	仕様(1)	仕様(2)	仕様(3)	仕様(4)
離陸する	アクション	実行する	－	－	－
右旋回する	アクション	－	実行する	－	－
左旋回する	アクション	－	－	実行する	－
着陸する	アクション	－	－	－	実行する
着陸中	条件	満たす	－	－	－
離陸中	条件	－	満たす	満たす	満たす

デシジョンテーブルからドローンの振る舞いが読み取れる．想定されない振る舞いは「着陸中に左旋回もしくは右旋回することはない」である．ソースコードを制御フローに基づきモデル化すると共に，ドローンの飛行状態もモデル化しこれらを結合した検査モデルを作成し検査を行う．

5.3 ソースコードモデル

ソースコードをモデルに変換する．ステートメントを状態と置きかえる．条件分岐(IF 文)や繰り返し(FOR 文)の制御フローを反映したモデルとなる(図 5-4)．「Kinect の骨格データ読込」「離陸」「着陸」「右旋回」「左旋回」の命令は図 5-4 上の各コメントが指している状態が該当する．

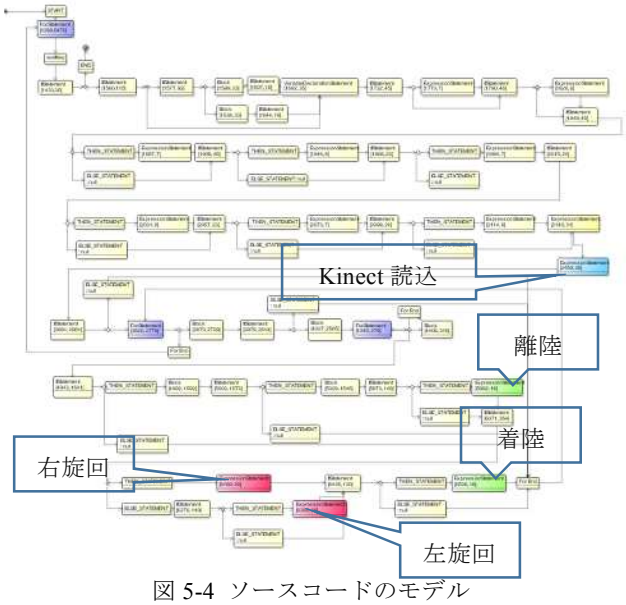

図 5-4 ソースコードのモデル

5.4 センサーのモデル

今回，センサーは Kinect である．Kinect の振る舞いをモデル化する．Kinect が取得する値は非決定性を使用して設定する．Kinect は人の骨格を 20 の部位で計測する．今回はプログラム内で使用している計測ポイントのみモデル化している(図 5-5)．計測しているポイントは右肩，左肩，肩中央，臀部中央，右手首の５ポイントである．

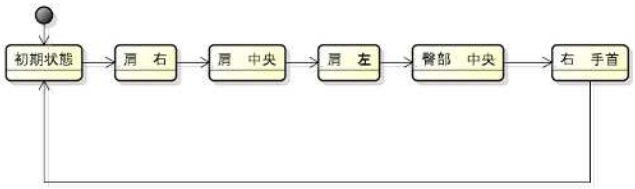

図 5-5 kinect のモデル

これらのポイントについて個々に X 軸，Y 軸の位置の値を指定範囲内で非決定に設定する．X 軸は体の中央を起点として左をー右側を＋としている．Y 軸は Kinect の設置位置の高さ 80cm と仮定し，そこを基点として基点より上方をー，下方側を＋としている．このモデルは値の取得後に初期状態に戻り値の取得を繰り返す．

5.5 仕様上で取りうる振る舞いのモデル

ドローンが取りうる振る舞いを仕様上で取りうる振る舞いのモデルとして作成する．今回ドローンが取りうる振る舞いは，「着陸中」，「離陸中」，「静止」，「右旋回」，「左旋回」である．振る舞いは離着陸系と旋回系の２種類あるので，モデルも２種類作成する．離着陸系のモデルが図 5-6 である．着陸中，離陸中の状態を交互に遷移するモデルである．状態遷移のきっかけは，着陸の命令(landing())，離陸の命令(takeoff())である．

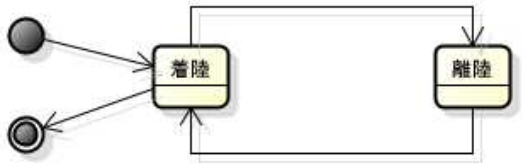

図 5-6 ドローンモデル 離着陸系

旋回系のモデルが図 5-7 である．静止，右旋回，左旋回の状態を状態遷移できるモデルである．状態遷移のきっかけは旋回の命令(move3D)である．

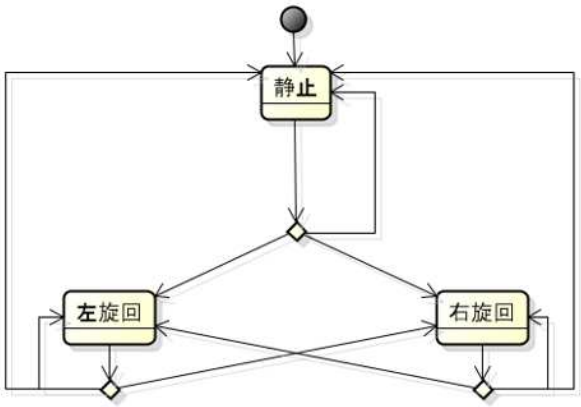

図 5-6 ドローンモデル　旋回系

5.6 検査モデルの構成

ここまでに作成した四つのモデルを表 5-1 にあるアクション(機能)を接点にして接合して検査モデルを構成する．ソースコードのモデルと Kinect のモデル，ドローンのモデルが同期される．同期で呼び出された側の処理が終了するまでソースコードのモデル側では状態遷移は行われない．

5.7 検査実施

5.2 節で判明した想定されない振る舞い，「着陸中に左旋回もしくは右旋回することはない」が発生することがないかを検査する．モデル検査では検査したい性質を検査式として与えて，検査を行う．検査式は「着陸中に左旋回もしくは右旋回することは決してない」となる．検査式は以下のとおりである．

□¬(着陸中の状態∧左旋回もしくは右旋回の状態)

この検査式で検査を行った．

検査結果は「満たされない」となり，着陸中に旋回する状態になることが判明した．

5.8 検査結果

検査式が満たされなかったためモデル検査ツールが反例を出力する．反例は満たされない状態になる状態遷移の過

程をしめしたものである．この反例を解析する．反例には多数の状態が記録されていて，そのまま見ても分かりにくいため特徴的な状態を選びその状態に任意の数値を割り当ててグラフ化して調査を行う．

最初にドローンの振る舞いを確認する．「離陸」「着陸」「右旋回」「左旋回」の命令を出しているステートメントの状態だけを抽出してグラフ化する(図 5-7)．縦軸は任意に設定した状態の数値，横軸は反例のレコード数である．何度か離着陸を繰り返した後，最後に左旋回とほぼ同時に着陸していることがわかる(図 5-7〇の部分)．

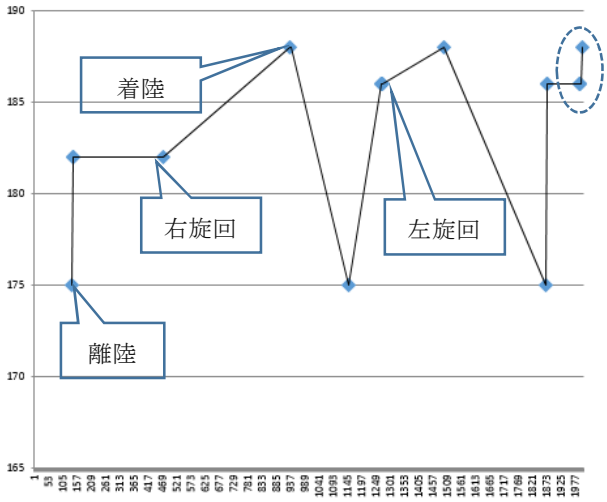

図 5-7 ドローンの振る舞いのグラフ

次に右手首と臀部中央のY軸方向の位置関係をグラフ化した(図 5-8)．右手首(wrist_right_y)が臀部中央(hip_center_y)より下にきていることがわかる(図 5-8〇の部分)．着陸の命令がでている．

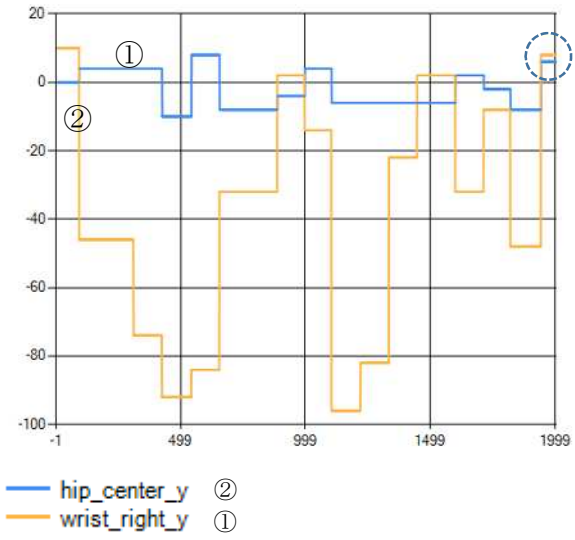

図 5-8 右手首と臀部中央 Y 軸方向のグラフ

次に右手首と肩中央のX軸方向の位置関係をグラフ化した(図 5-9)．右手首(wriset_right_x)が肩中央(shoulder_center_x)より左側にあることが分かる(図 5-9〇の部分)．左旋回の命令がでている．

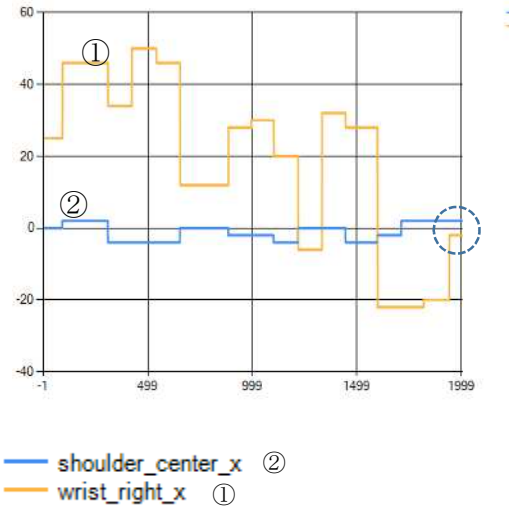

図 5-9 右手首と肩中央 X 軸方向のグラフ

5.9 検査結果ついての考察

ドローンの操作者の腕の動きは上下左右の 4 つのパターンを想定してプログラムを作成している．今回の反例より，右腕を左斜め下にむけて移動させると，左旋回と着陸の処理が同時に実行される可能性があることがわかった．この結果に基づきソースコードを確認したところ，IF 文の記述で ELSE の記述が適切ではなく着陸の命令と旋回の命令が同時に実行されることが判明した．

6. まとめ

単純な振る舞いに対する検査であったが今回ドローンの制御プログラムに対して，モデル検査により想定外の振る舞いが発生する不具合を発見した．複雑な振る舞いを持つ場合でも今回提案した手法を拡張することにより対応できると考える．例えばバッテリー切れによる墜落を防ぐ機能を作った場合，墜落する前に安全な場所に着陸することができるかの検証を行うことができ，障害許容性，回復性の向上が行える．

今後ドローンの規制が強まることが想定され，その規制をクリアするためにドローンに対して様々な作りこみが行われるはずである．それには様々なドローンの振る舞いを統合して検証する必要があり，実機による検証には限界がある．自動車のソフトウェアの品質を定義している ISO26262 では，クリティカルな機能については準形式手法による検証が求められている．同様に複雑な振る舞いをするドローンについても形式手法による網羅的な検証が有効と思われる．

今回の適用事例は単純な振る舞いしかしないが，ドロー

ンに対して入力される命令と状態を組み合せてあってはならない状態になることが確認できた．ハードウェアの問題による不具合を本提案手法で発見することは難しいが，想定外の命令の組合せや，意図しないタイミングでの複数機能の同時実行などで発生する不具合については発見できると考える．今後の予定としてはもっと複雑な振る舞いをするドローンに対して提案手法を適用していきたい．

参考文献

1)Overview of Small UAS Notice of Proposed Rulemaking, https://www.faa.gov/regulations_policies/rulemaking/media/021515_suas_summary.pdf, 2015.
2)離陸するドローン・エコシステム, http://blogos.com/article/103321/,2015.
3) 形式手法の実践ポータル, http://formal.mri.co.jp/db/fmtool/,2015.
4) SPIN, http://spinroot.com/spin/whatispin.html/, 2015.
5) NuSMV, http:// http://nusmv.fbk.eu/ ,2015.
6) Pathfinder,http://javapathfinder.sourceforge.net /,2015.
7) UPPAAL, http://www.uppaal.com/, 2015.
8) VDMTools, http://www.vdmtools.jp/,2015.
9)K. Lano;H. Haughton, “Specification in B: An Introduction Using the B Toolkit”, Imperial College Press, 1996.
10) Parrot AR.Drone 2.0, http://ardrone2.parrot.com/,2015.
11) Kinect for Windows, https://www.microsoft.com/en-us/kinectforwindows/,2015.
12) CVDrone
13) OpenCV.jp, http://opencv.jp/,2015.
14) B.Beizer: Software Testing Techniques, Computer Press, 1990