

宇宙機の自律化を目的とした搭載コマンド計画・状態監視機能の実装について

福島洋介 宇宙航空研究開発機構宇宙科学研究所<fukushima@isas.jaxa.jp>

三田信, 大槻真嗣(ISAS/JAXA)

なぜ自律化にコマンド計画・状態監視が重要なのか

こんにちは、福島洋介です。

宇宙研で助教をしています。メンバも全員助教です。理論とソフトウェアは福島が、ハードウェア試作は三田と福島で、屋外でのプール実験は福島と三田、海岸実験は福島と大槻という分担で活動しています。

このポスターでは、従来の宇宙システムで実用が検討されたことがなく、また今後しばらくは難しいであろう技術（テクノロジー）をご紹介します。

この研究の目的は、宇宙機を自律化することです。自律化といつてもいろいろなことが含まれますので、ここでは宇宙機の搭載ソフトウェアによって実現しうる自律（しているかのように見える機能）とします。

まず、自律とは何でしょうか。自律化や知能化といった用語は、いろいろな研究者がいろいろな提案をしていて、誰かが定義をするともめる原因となります。ここでは次のような「ばやっ」としたものにします。

自律：人が介在しないで、状況に応じた行動をとること
知能：どうしたらいいかわからない（明示されていない）ときにも、適切に振る舞うことができる能力

異論がある方もいらっしゃるかもしれません、お許しください。

ところで、このような言葉を決めて物事は進みません。搭載ソフトウェアにそった言葉に書き換える必要があります。現実的な意味で宇宙機における自律の定義を次のようにしようよ、と提案します。

自律：事前に運用者が準備した行動選択肢の集合から、目的を満足させ、かつ、システムの自滅につながらない適切なものを自ら選択する機能

宇宙システムができることは、宇宙システムのハードウェアで実現できることであり、それは設計で決定されています。その後できることは、それらを「どのように組み合わせて使うのか」だけです。それは自律を目指した宇宙システムであっても同様です。

自律という言葉から何か創造的な行動をとる発想される方も多いかもしれません。が、宇宙という極限環境で生き延びための選択の幅は非常に狭い。創造的な行動はトライ・アンド・エラーを伴うものです。しかし、それでは確実に自滅してしまうでしょう。死んでしまっては何もなりません。

自律の定義をもう一度確認してみましょう。「行動の選択肢」とはコマンドです。「目的を満足させ」るよう「選択する」とは計画することです。そして、「システムの自滅にならない」は状態監視をしています。

というわけで、「宇宙機の自律化を実現する」とはつまり「搭載コマンド計画・状態監視機能を実装する」と言い換えられるわけです。では、次に考えることは「どうすればそれが実現できるのか」です。

1. どのように実現させるのか

まずは、搭載ソフトウェアの基本的な文節と組み立て（アーキテクチャ）を決めます。ここでは一般的な方法である「三層構造」を採用します（図1）。要するに、リアルタイムが必要なものはdriver層、そうでないものはdeliberator層、その両者をつなぐものをexecutor層と考えておけば十分です。

従来の意味での制御をはじめとした機能の大部分はdriver層になり、コマンド計画や状態監視という機能がdeliberator層に割当られます。最近のRTOSを使えば機能のグループ分けと層単位の実行は簡単。

問題はdeliberator層をどうするかです。ここでは、従来あまり搭載ソフトには用いられなかったスクリプトエンジンを採用しています。要するに、JavaScriptをつかってコマンド計画・状態監視を実現しました。

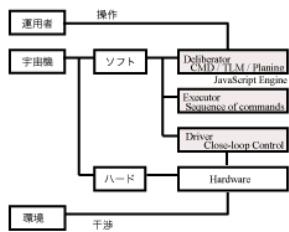


図1 三層構造アーキテクチャ

ミッションを実現するためにコマンドは計画されます。従来の衛星では計画されたコマンドを運用者かあるいは時刻指定で送信・実行されていました（図2）。運用者ではいろいろなことが起きますが、運用者はそれにあわせてコマンド計画を変更し、ミッションを遂行させていました。これを搭載ソフトウェアで実施すればよいわけです。



図2 小型科学衛星でのコマンド送信

どのようにコマンド計画を自動でつくるのか。自律化の定義のところで確認しましたが、ここでは「コマンドの選択肢から適切なものを選択すること」を行えばよいわけです。コマンドの選択肢はあらかじめ与えられていますから、宇宙機の状況にあわせて選択のやり直し、選択結果の並び替えを行えばよいわけです。これを図4のように3ステップで実施します。

この処理のときのコマンドの選択肢は「プラン」という基本要素にまとめられています。このプランには、送信するべきコマンドだけでなく、送信する前の確認事項を含んだ「状態監視」を行ためのアルゴリズムを含んでいます。プログラミング言語の言葉で言えば、プランはオブジェクトです。この研究でのプランはJavaScriptで記述してみます。この方法で実装できそうです。では、実験してみましょう。

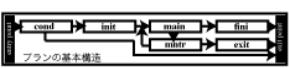


図3 ロジックコマンドの基本要素「プラン」

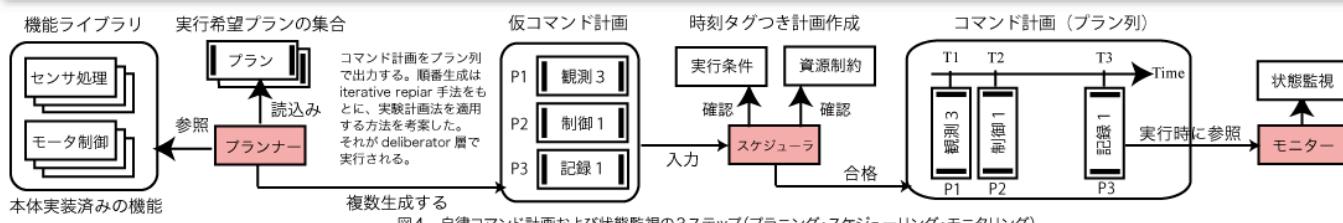


図4 自律コマンド計画および状態監視の3ステップ(プランニング・スケジューリング・モニタリング)

2. AUVに実装し、その実現性を実験で確認する

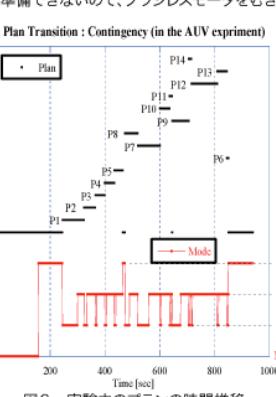
何事も言葉だけならばどうでも言えます。そこで、提案するソフトウェアを遠隔システムに実装して動作するか確認してみました。対象はAUV（自律潜水艇）。水中では手軽に通信できないので本当に自律でないと実験できないから選びました（not because they are easy, but because they are hard）。

ハードウェアは我々で試作しました（図5）。全長70cm、重量20Kg。材料は東急ハンズと秋月電子で購入した部品が主な部材です。モーターも防水ケーブルを準備できないので、ブラシレスモーターをむき出しでっています。エレクトロニクスもモータードライバチップから半田付けするレベルから試作しています。

図6に水中での動作結果を示します。このケースではP1からP14までのプランが順に実行されるようにしてAUVを水中に沈めました。P5で水深異常を検知、コマンド計画を自律で変更し、順番を変更して最後まで実行したことが示されています。つまり、コマンド計画と状態監視が想定通り機能したのです。

同様の実験をISAS内のプールや浅い海で実験しています。海の実験では、海水が船内に浸水し、電子機器を完膚なきまでに腐食させたこともあります。

これまで9個のモーターと姿勢センサを使った移動を中心としたミッションばかりでした。現在は自作した広視野を装着を行っています。また、それ以外のセンサを搭載し、一般的な移動体研究領域にも踏み込んで行きます。



AUVを試作して実験したせいか、この研究の評判はよろしくありません。「宇宙研でやる内容か?」とか「まだ潜水艦やっている?」というものがかりです。ですので、研究継続ために衣替えも検討しています。もしも来年発表する機会がありましたら、別の遠隔システムを使った実験になっているかもしれません。

なお、この研究成果については、第16回ロボティクスシンポジウム（第53回自動制御連合講演会、第54回宇宙科学連合講演会）でも一部発表されています。より詳しい情報はこれらの学会のプログラムを著者名で検索されるか、ポスター左下のURLまでアクセスください。

最後まで目を通していただいた方にはお礼を申し上げます。ありがとうございました。

