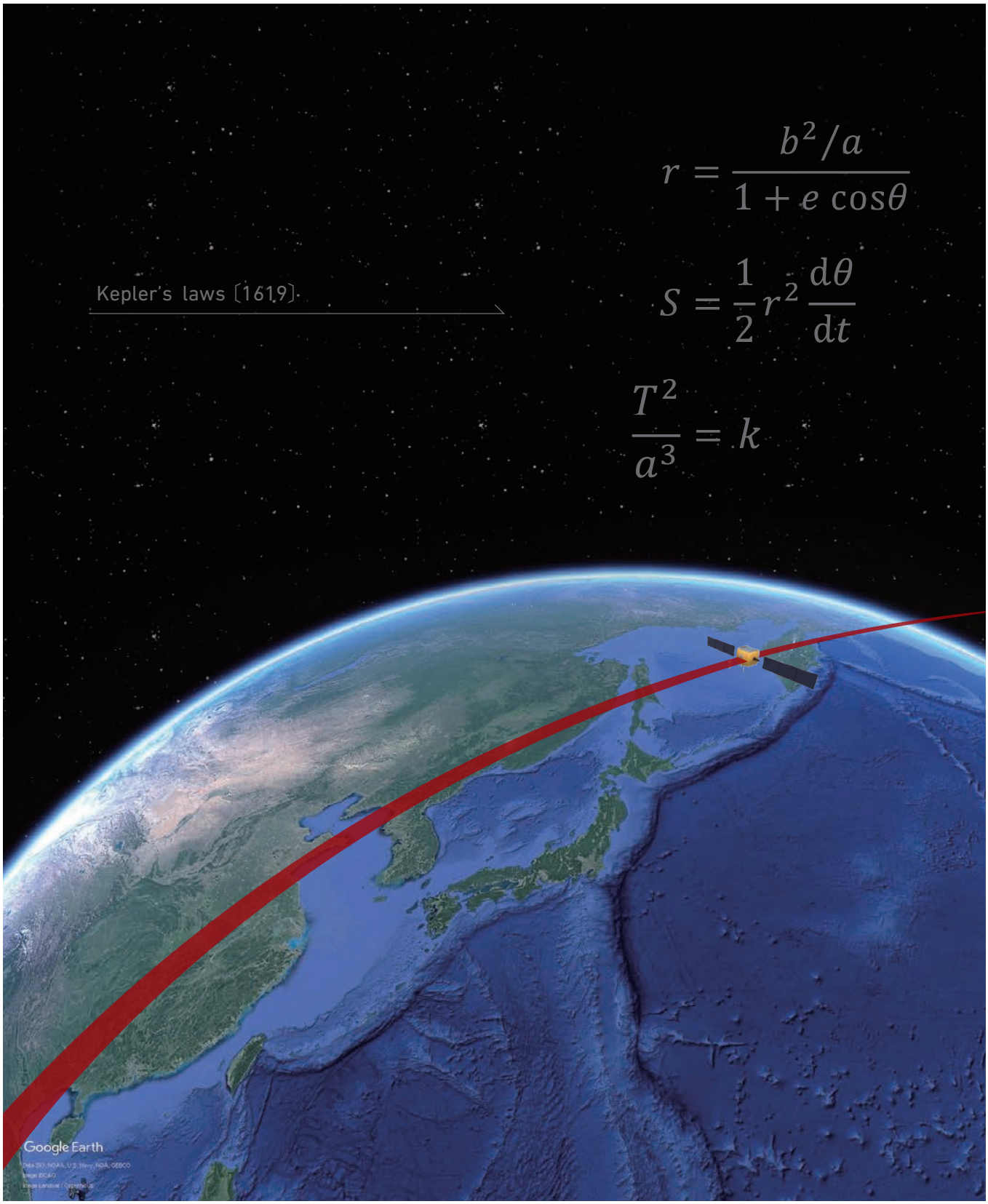
ISSN 2433-1597

Homura Heavy Industries Technical Review

炎重工技報

Vol.1
2017



Feature

汎用自律移動コントローラの紹介

PROCESSWARPと群ロボット

生体群制御とロボット養殖

Homura Heavy Industries Corporation.

炎重工技報

Homura Heavy Industries
Technical Review

Vol.1

2017

「炎重工技報」創刊の挨拶

代表取締役　古澤 洋将

平素は格別のご高配を賜り、厚く御礼を申し上げます。当社は、東日本大震災からの復興を胸に抱き、頭上に独自の発想を、両手には誠意と技術を持ち、2016 年 2 月に岩手県滝沢市にて設立されました。前身は、代表取締役が 2004 年に興した農業組織 NEXTFOODS の開発部門であります。本技術機関誌「炎重工技報」の創刊にあたり、挨拶を申し上げます。

科学技術の歴史を紐解くと、1837 年にフランス人の経済学者ジェローム＝アドルフ・ブランキが、1700 年代後半のイギリスで興った蒸気機関による工業化を産業革命と初めて名付けました。それ以来、日本も産業革命とは無縁でなく、1862（文久 2）年に江戸幕府老中の小栗上野介忠順は、駐日フランス公使レオン・ロッシュらと共に近代的な製鉄所の建設案を作成し、1865（慶応元）年には横須賀製鉄所を建設しました。次いで釜石製鉄所、八幡製鉄所など日本各地に製鉄所が作られ、明治の造船産業、大正の航空機産業、 昭和の自動車産業などを支えることになりました。 近現代史を振り返ると、世界の国々の勃興は例外なく科学技術によって行われております。

さて、2011 年 3 月に東日本大震災（死亡・行方不明者 18,446 名）が発生してから、東北地方の太平洋沿岸を中心に地域経済は破壊されてしまいました。近現代史に習えば、このような大震災からの復興に必要なのは、21 世紀を切り開く新しい科学技術であると言えます。ここで、今日の私どもを取り巻く環境を眺めてみますと、1990 年代から飛躍的に発展したコンピュータとネットワーク技術を基礎に、2000 年から自律移動型ロボットや自動車が広がりを見せ、汎用人工知能の実現をも期待されています。また、人々の通信手段も飛躍的に発展して、いつでもどこでも電話や電信（メール）ができ、SNS といったコミュニティ・ネットワークが人間の社会生活の一端を担っています。人類の歴史から例を探すのが困難なほど、様々な技術と情報に溢れているのが現代の特徴である、と言えるでしょう。

このような背景のもと、21 世紀の現代において新しい価値を創造するには、日々の研究開発を積み重ねる方法のほかに、過去の技術や産業を現代の技術でもう一度見つめ直すという方法もあるのではないかと思います。異業種交流という言葉が生まれて久しく、例えば金融工学や生体医工学など、異分野との協業から生まれた技術や製品、サービスは数多あります。これに加えて当社は、創造的な未来を切り開き人々の喜びを広げるために、ロストテクノロジーを見つめ直し、創造的改善にも取り組んで参ります。

創刊号は、屋外ロボット開発キットに使われている汎用自律移動コントローラと、ソフトウェア・プロセスのマイグレーションができる PROCESSWARP、魚介類等の水中生物群を自由に誘導できる生体群制御の技術について、それぞれご紹介いたします。ぜひ御高覧頂き、忌憚のないご意見をお寄せ頂ければ幸甚の至りに存じます。今後、炎重工技報は年 1 回発行していきます。より一層の御支援・御鞭撻を賜りますようお願い申し上げ、挨拶といたします。

目 次

巻頭言

技術報告

その他

炎重工株式会社
HOMURA HEAVY INDUSTRIES

Feature
1

汎用自律移動コントローラの紹介

古澤 洋将 *1　伊藤 祐司 *1

Multi-purpose autonomous controller for outdoor machinery

Yosuke Furusawa*1, Yuji Ito*1

Abstract - This paper presents the outline of the multi-purpose autonomous controller and the philosophy of development. All components, including the microprocessor, autonomous software, self-diagnostic system and actuators, were specially designed, and many robust technologies were adopted. A detailed account is given regarding the systems and devices developed to meet such requirements, along with the specifications.

Keywords: robot autonomous

1.　まえがき

米国西海岸のカリフォルニア州に、日系三世が経営する国府田農場がある [1]。農場の面積は約 3,000ha で、そのうち約 1,600ha にもち米、約 800ha にうるち米を作り、残りの約 600ha には綿や小麦などを作っている。農場の区画は約 64ha（800×800m）で、そこには広漠たる真平らの水田がある [2]。

しかし、実は 1960 年代までは、米国といえどもこのような大規模な区画を画一に栽培することはできなかった。区画内部に高低差がありすぎて均一に灌水をすることができず、等高線に沿って畦畔を作るしかなかった。その後、1970 年代に入りレーザー均平（レーザーレベラー）技術の登場により、広大な農地を水平に均平できるようになった。そのため、等高線に沿った畦畔を作らずとも均一に灌水ができるようになり、区画面積が次第に大きくなっていった。古くは畜力農機具（農耕馬による犁など）から始まった工業化農業も、このような耕作・区画面積の増大と共に、多頭による大型畜力機械化を経て、内燃機関を用いた大型の農業機械へと発展していった [3]。

大型トラクタ＋アタッチメント＋大型コンバインといった大型の汎用機械の開発が進んだ欧米型農業に対して、日本では 1894 年に小岩井農場が輸入した蒸気式プラウをきっかけに [4]、稲作を中心とした中型の専業機械の開発が進んだ [5]。この日本型機械の発展は、第二次世界大戦後の農地改革により、日本農業の経営規模が小さくなってしまったことにも起因している [6]。例えば、先ほどの国府田農場では、早くも 1929 年には飛行機による種籾の散布に取り組んでいたが [2]、日本で普及したのは育苗箱を用いた田植機であった [7]。とかく、このような歴史的背景により、大型の汎用機械を主力にした欧米メーカと、中型専業機械を主力にした日本メーカに分かれることになった。近年では中国メーカも台頭してきており [8]、世界的な食糧需要に相まって、農業機械は多様化が進んだ成長市場になっている。

本稿では、汎用自律移動コントローラの概略、及びそれを適用した開発車両を紹介し、自律移動の具体例を示す。将来は、屋外作業の自律化を進めることによって、生産高の拡充や労働時間の短縮、3K 労働の低減を目指していく。

2.　汎用自律移動コントローラ

2.1　開発の狙い

屋外作業に目を向けると、農業以外にも林業や土木・建設作業、冬期の除雪作業など、実に様々な業種がある。作業現場は、農地や砂地、雪上が多く、これら作業機械の多くは内燃機関を用いた装軌（クローラ）車両であり、農業機械と類似した構造と移動方式を採っている。日本国内で発売されている作業機械のうち、最も小型・軽量なものは、ガソリンエンジン 100cc 程度、車両重量 100kg 程度の運搬車のようである。大型のものは枚挙に暇がないが、農業用であればディーゼルエンジン 3,000cc 程度、車両重量 4,000kg 程度の汎用コンバインが挙げられる。

当社の汎用自律移動コントローラは、農業機械だけでなく、建設機械や除雪機などの装軌車両に対しても自律移動機能を付加することを狙っている。また、既存の車両に適用することで、今まで培われた既存車両のノウハウを活用し、開発期間を短縮することが可能になる。

図 1　屋外ロボット開発キット（開発車両）

*1: 炎重工株式会社 研究開発部
*1:R&D Dept, Homura Heavy Industries Corporation.

2.2　コントローラ構成

汎用自律移動コントローラは、ECU 系を中心に、センサ系、通信系、インターフェース系のモジュールに分けられる。また、緊急停止系とアクチュエータ系のモジュールは、車両の仕様に合わせて構成される。

これらは独自にバス接続され、ECU と各モジュール間で通信が行われる（図 2）。

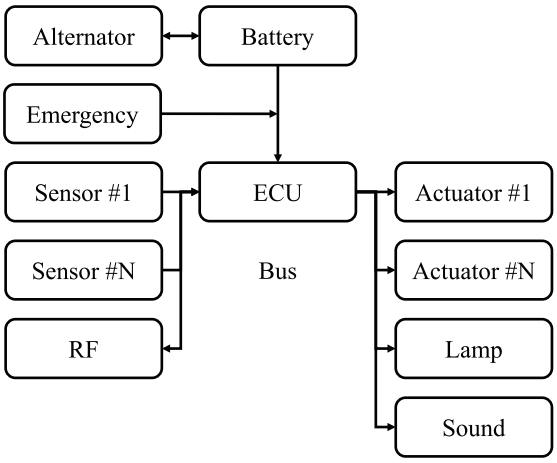


図 2　システム概要

(1) ECU 系

ECU 系は、内部に 2 つのプロセッサを持ち、リアルタイム性が求められる低レイヤの制御と、移動経路の計算など高レイヤの制御をそれぞれ分担して行う。

(2) センサ系

センサ系は、電圧・電流・温度センサやモーションセンサ、GNSS など、車両や環境、位置情報を取得する役割を持つ。一部のセンサは、ECU の外部に配置される。本稿では、人工衛星を用いた位置情報のシステム（GNSS）を総称して GPS と呼ぶ。

(3) 通信系

通信系は、外部と通信をするためのアンテナを含んだ通信システムである。標準では、WiFi と RC 受信機が搭載される。

(4) インターフェース系

インターフェース系は、利用者に対するランプと警報装置である。ウィンカーやブレーキランプの機能も、含まれている。

(5) 緊急停止系

緊急停止系は、物理的な緊急停止スイッチの押下とガードバンパによるインターロック、通信系のモニタリングによる OR 条件によって作動する。緊急停止が作動すると、直ちにエンジンが停止する。

(6) アクチュエータ系

アクチュエータ系は、車両の物理的なインターフェースに取り付けられ、車両の動きを制御する。例えば、エンジンのスロットル、クラッチ、ミッション、ブレーキなどである。

(7) その他

本システムから独立して、ドライブレコーダが取り付けられる。システムが通電されると、車両の動作状態を撮影し続ける。

表 1　コントローラ仕様概要

項目	仕様
品番	RCBA2000C
基本制御 OS	F - OS/PIC24F 3.4 - RELEASE
	CPU 16MIPS
	RAM 8KB
	ROM　128 KB
基本機能	アクチュエータ出力　4ch
	RC プロポ入力　6ch
	6 軸モーションセンサ
	電圧・電流・温度センサ
	リレー出力　2ch
	ランプ出力　6ch
拡張制御 OS	Linux 4.4
	ARM　1GHz (約 2000 MIPS)
	RAM 1GB
	SSD 32GB
拡張機能	WiFi 2.4GHz
	GPS (QZSS)　3ch
	USB　2.0
	GPIO 3ch
	A/D 3ch
	I2C　1ch
操作 I / F	RC プロポ
安全機能	非常停止スイッチ
	ガードバンパ
	ドライブレコーダ
動作環境	- 20 ～ 40 ℃
	防沫形（IPX4 級）
電源	DC 9 ～ 28V
	15W（アクチュエータを除く）

図 3　コントローラ全景

2.3　車両構成

下表に、屋外ロボット開発キットに用いられているベース車両の仕様を示す。以降、ベース車両について述べるが、汎用自律移動コントローラと車両は独立した構成であり、ユーザは用途に合わせて自由に車両を選定することができる。

(1) 拡張性

ベース車両には、最大 300kg まで積載することができる。無積載の状態で使用する場合、車両重心が後方になるため、40kg 程度のフロントウェイトを取り付ける場合がある。また、オルタネータを内蔵しているため、外部に DC 12V の電源を供給することができる。

(2) 走破性能

ベース車両は、基本的な移動機能として、前進 2 速と後進 2 速、左右の旋回を行うことができる。また、防滴構造のため、雨天、泥路、降雪、雪上など様々な環境下での走行ができる。そのほか、路面勾配 ±8% までの段差乗り越えと傾斜横断を行うことができる。

表 2　ベース車両主要諸元

項目	仕様
品番	RCBA0000A
移動方式	無限軌道（クローラ）
外観寸法	1640 × 650 × 875mm
最低地上高	85mm
進行方向	前進・後進・左右旋回・スピンターン
最小回転半径	1200mm
走破性能	水深 20cm 以下
変速機	前進 2 速
	後進 2 速
最高速度	3.6km/h
路面勾配	± 8%（4.6 度）
車両重量	150kg
最大積載量	300kg
バッテリ	12V, 28Ah（鉛蓄電池 40B19L）
排気量	無鉛ガソリンエンジン, 125cc
最大出力	3.1kW / 3800rpm
最大トルク	15.3Nm / 1800rpm

図 4　雨天走行

図 5　雪上走行

図 6　泥路走行

図 7　段差乗り越え

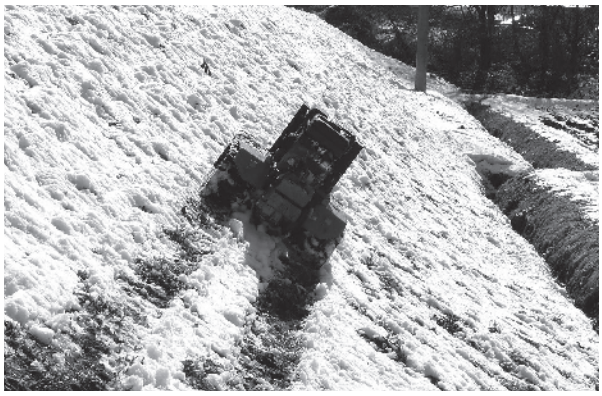


図 8　傾斜横断

2.4　自律移動の例

汎用自律移動コントローラは、GPS による位置情報を元に、予め指定された経路情報に沿って自律移動（Waypoint Navigation）を行う。航空機の用語に習い、計器走行とも呼ぶ。ECU 内部に経路情報を複数持つことができ、外部（操作 I/F）からの指示で経路を切り替えることができる。図 9 と図 10 に、自律移動の走行例を示す。

図 9　雪上走行の例

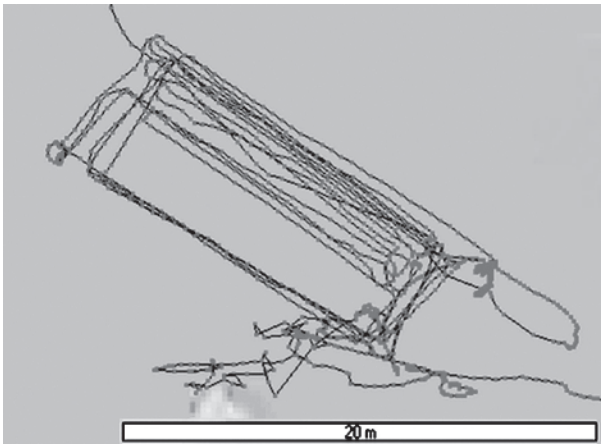


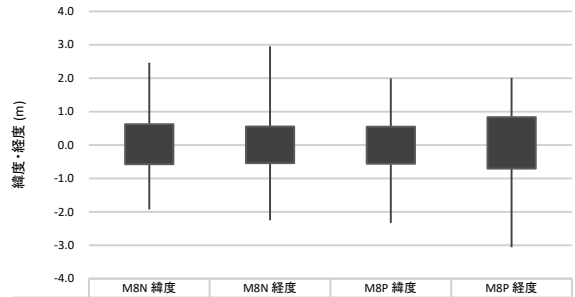
図 10　GPS の軌跡の例

2.5　GPS の精度と評価の例

汎用自律移動コントローラは、外部に任意の GPS モジュールを接続することができ、用途に応じて選定が可能である。図 11 に GPS の評価例を示し、評価に用いた GPS の仕様概要を表 3 に示す。いずれも、GPS 単体（Standalone）で計測を行った。

表 3　GPS の仕様概要 [9]

Specifications	ublox NEO - M8N	ublox NEO - M8P
Receiver	72ch	
	GPS , GLONASS , BeiDou	
	QZSS	-
	Galileo	-
Sensitivity	- 148 ～ - 167dB	- 148 ～ - 160dB
Rate	～ 10 Hz	
Position accuracy	2.5m	



	M8N 緯度	M8N 経度	M8P 緯度	M8P 経度
第3四分位	0.623	0.554	0.549	0.835
最大値	2.465	2.954	1.992	2.012
最小値	-1.929	-2.253	-2.335	-3.058
第1四分位	-0.575	-0.542	-0.560	-0.704

図 11　GPS の要約統計量の例

ublox NEO-M8N、及び NEO-M8P の評価結果、仕様上の水平精度 2.5m に対して、実測上は 1.1 ～ 1.5m 程度の精度が得られた。

3.　自律移動制御

3.1　はじめに

本稿では屋外ロボット開発キットの開発車両を例に、汎用自律移動コントローラが実装している自律移動制御のうち、2.4 節で述べた Waypoint Navigation（計器走行）の一部を紹介する。

3.2　車両の制御とモデル

開発車両は、一般的な自律移動型のロボットと異なり、車両の動力源としてガソリンエンジンを使用している。エンジンの出力軸は、スロットル開度に応じて常時 700 ～ 3000rpm 程度で回転しており、前進・後進の 2 段変速機を経て、左右のクローラへと出力が伝えられる（図 12）。エンジンと変速機の間、変速機内部と左右クローラの間には、合計 3 つのクラッチ（図 12 の矢印）が存在し、エンジンの出力を接続または切断することができる。

汎用自律移動コントローラは、これらクラッチに取り付けられた 3 つのモータを使用して、発進・左旋回・右旋回・停止など車両の走行を制御している。一般的な自律移動型のロボットと走行の出力系が異なること以外は、独立二輪駆動型のアーキテクチャに類似している。

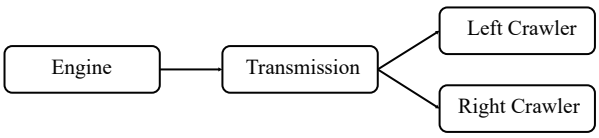


図 12　原動機と動輪の関係

図 13 は開発車両のモデルを表し、v は開発車両の走行速度、ω は開発車両の回転角速度、ω_L は左クローラの回転速度、ω_R は右クローラの回転速度、T はトレッド（車体幅）、R_w はクローラ動輪の半径を表す。重心位置の表記は、無積載時のものであり、ガソリンエンジンに近い後部中央に位置する。しかし実際には、状況に応じて、中央、前方、右寄り、左寄りなどと激しく変化する。

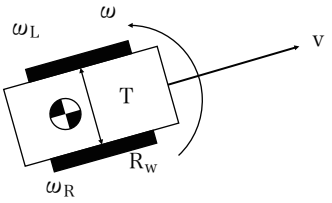


図 13　開発車両のモデル

3.3　自己位置の算出

汎用自律移動コントローラは、屋外での自己位置を算出するために、GPS による位置情報を用いている。GPS から直接得られる位置情報は、1960 年に米国国務省が定めた世界測地系（World Geodetic System）を元に、現在は 1984 年に改訂された世界測地系 84（WGS-84）に規定された計測値である [11]。WGS-84 は、地球中心直交座標系（ECEF：Earth-Centered Earth-Fixed）とも呼ばれ、地球の中心に原点を置き、グリニッジ子午線と赤道面の交わる方向を x 軸、地軸方向を z 軸、x 軸及び z 軸と右手直交系をなす方向を y 軸と定めている（図 14）。

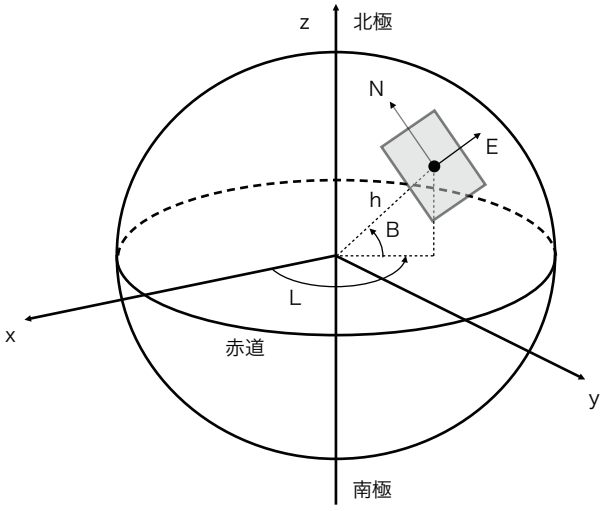


図 14　ECEF 直交座標系

また、緯度を B、経度を L、楕円体高を h、卯酉線曲率半径を n、離心率を e とすると、GPS から得られる ECEF 直交座標系と緯度・経度の測地座標系は、よく知られた数式 1 の関係式で表される [10]。なお本稿では、ECEF 直交座標系から測地座標系の算出について、これ以上の詳述は行わない。

数式 1　ECEF 直交座標系と測地座標系の関係式

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (n+h)\cos B \cos L \\ (n+h)\cos B \sin L \\ \{(1-e^2)n+h\}\sin B \end{bmatrix}$$

汎用自律移動コントローラの自律移動制御では、平面での移動に近似するため、局地座標系ではなく ENU 座標系（East-North-Up）を用いている（図 15）。ENU 座標系を用いると、任意の原点 O からの現在位置 P が 3 次元の直交座標として得られる。

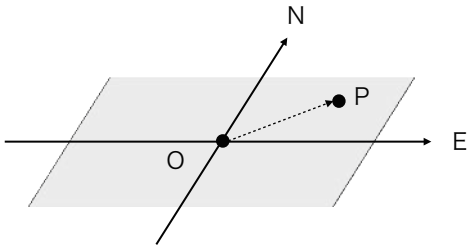


図 15　ENU 座標系

ここで任意の原点 O の ECEF 直交座標を O_{ECEF}、自己位置 P の ECEF 直交座標を P_{ECEF} とすると、ENU 座標系における P_{ENU} の位置は、数式 2 の関係式で表される。

数式 2　ECEF 座標系と ENU 座標系の関係式

$$P_{ENU} = \begin{bmatrix} -\sin L & \cos L & 0 \\ -\cos L \sin B & -\sin L \sin B & \cos B \\ \cos L \cos B & \sin L \cos B & \sin B \end{bmatrix} [P_{ECEF} - O_{ECEF}]$$

$$P_{ENU} = \begin{bmatrix} P_E \\ P_N \\ P_U \end{bmatrix}, \quad P_{ECEF} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \quad O_{ECEF} = \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix}$$

3.4　経路と自律移動

開発車両 M が、開始位置 S から、予め指定された位置（Waypoint）を経由して、終了位置 E まで移動すること考える（図 16）。このとき、開発車両 M の移動経路は、開始位置 S から終了位置 E までの座標リスト WPn という形で表現できる（表 4）。ただし、本稿では簡易的に説明するため、経路はすべて 2 次元の直線移動とし、途中の障害物や傾斜などは考慮しない。

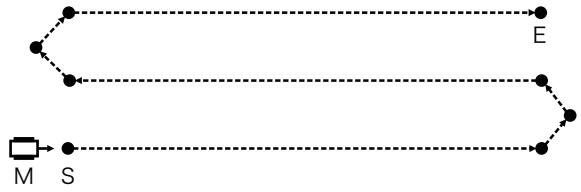


図 16　移動経路の例

表 4　座標リストの例

List	E	N	U
S	S_E	S_N	S_U
WP n	WPn_E	WPn_N	WPn_U
E	E_E	E_N	E_U

ここで、3.2 節で示した開発車両 M のモデルを用いて、座標リストの $WP1_{ENU}$ から $WP2_{ENU}$ へ直線移動する場合を考える（図 17）。

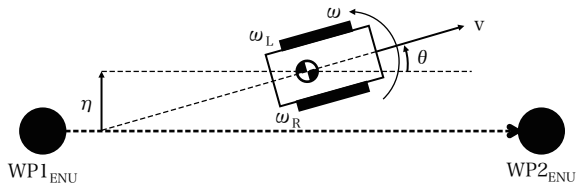


図 17　直線に沿う制御

3.2 節の通り、開発車両のクローラ型の車両は、独立二輪駆動型のアーキテクチャとして取り扱えるため、$WP1_{ENU}$ から $WP2_{ENU}$ の直線（目標値）に沿う制御を行うとき、開発車両 M の速度 v を一定とすると、数式 3 として表すことができる [13]。

数式 3　開発車両の拘束式

$$\left(\frac{d\omega}{dt}\right) = -k_\omega\omega - k_\theta\theta - k_\eta\eta$$

$$s^3 + k_\omega s^2 + k_\theta s + vk_\eta = 0$$

$$\omega(\mathrm{t}+\Delta\mathrm{t}) = \omega(\mathrm{t}) - (k_\omega\omega + k_\theta\theta + k_\eta\eta)\Delta t$$

ここで、開発車両 M の回転角速度 ω と直線に沿うための左右のクローラ回転速度 ω_L、ω_R の関係式を求めると、数式 4 になる。

数式 4　クローラの回転速度

$$\begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix} = \begin{bmatrix}\frac{1}{R_w} & \frac{T}{2R_w}\\ \frac{1}{R_w} & -\frac{T}{2R_w}\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$

したがって、左右のクローラ回転速度 ω_L、ω_R を制御すれば、$WP1_{ENU}$ から $WP2_{ENU}$ へ直線移動することができる。同様に、WP3 … WPn と直線移動を繰り返していけば、任意の経路を移動できる。

3.5　自己位置と制御の実際

GPS の位置情報には、様々な誤差が含まれている。自己位置をより正確に算出するには、3.3 節の式に対して、振動子変動、電離層遅延、対流圏遅延など様々な補正式を加え、GPS ごとに較正する必要がある [14][15][16][17]。

また、屋外ロボットが走行する環境は、路面とクローラの接地抵抗が理想的とは限らず、スリップの発生しやすい泥や雪の場合がある [18]。加えて、車両の姿勢や積載の状況によっては、重心位置も激しく変化してしまう。汎用自律移動コントローラは、このような条件下でも経路走行が出来るように制御を行っている。

4.　おわりに

本稿では、汎用自律移動コントローラを用いた屋外ロボット開発キットと、そのベース車両について紹介を行った。これは、屋外作業機の市場に新しい波紋を巻き起こすべく、意欲的な商品として設計されたものである。今後、皆様からの御意見・御批判を頂き、更に改良を重ねていく所存である。将来は、屋外作業の自律化を進めることによって、生産高の拡充や労働時間の短縮、3K 労働の低減を目指していく。

謝辞

本研究の一部は、総務省平成 27 年度異能 vation によった。

参考文献

[1] Koda Farms: http://www.kodafarms.com/

[2] 七戸長生 , et al: 農学基礎セミナー農業の経営と生活 ; 社団法人農山漁村文化協会 , pp.2-3, (2000).

[3] 石井敦 : 米国巨大水田見聞記 ; 農業土木学会誌 , No.73 Vol.4, pp.335-338, (2005).

[4] 小岩井農場の歴史 : https://www.koiwai.co.jp/story/kaikon.html

[5] 農林水産省 : 農業機械化促進法 ; (1953).

[6] 中村政則 , et al: 占領と改革 ; 岩間書店 , (1995).

[7] 保木元利行 : 日本農業機械市場の歴史的展開過程とその分析 ; 山形大学紀要（農学）, 第 13 巻 第 2 号 , pp.117-143, (1999).

[8] 中華人民共和国 国務院 : 中国製造 2025; (2015).

[9] ublox NEO-M8 シリーズ https://www.u-blox.com/ja/product/neo-m8-series

[10] 坂井丈泰 : GPS 技術入門 ; 東京電機大学出版局 , (2003).

[11] Misra P, et al: GLOBAL POSITIONING SYSTEM signal measurements and performance; Gange-Jamuna Press, (2001).

[12] 坂井丈泰 : GPS のための実用プログラミング ; 東京電機大学出版局 , (2007).

[13] 米田完 , et al: はじめてのロボット創造設計 ; 講談社 , (2001).

[14] Hofmann-Wellenhof, et al: GPS 理論と応用 ; シュプリンガー・フェアラーク東京 , 2005.

[15] Klobuchar, J: Ionospheric Effects on GPS; Global Positioning System: Theory and Applications, vol.1, pp.485-515, (1996).

[16] Spilker, J. J. Jr.: Tropospheric Effects on GPS; Global Positioning System: Theory and Applications, vol.1, pp.517-546, (1996).

[17] 内藤勲夫 , et al: GPS 気象学 ; 気象研究ノート , 日本気象学会 , No.192, (1998).

[18] 吉田和哉 , et al: スリップを考慮したクローラ型移動ロボットの軌跡追従制御の実現 ; 日本機械学会ロボティクス・メカトロニクス講演会講演論文集 , pp.2P1-L01, (2007).

Feature
2

PROCESSWARPと群ロボット

伊藤 祐司 *1　古澤 洋将 *1

Distributed system with PROCESSWARP applicable to swarm robotics

Yuji Ito*1, Yosuke Furusawa*1

Abstract - We are developing a system that creates a virtual machine on multiple computers connected through the Internet. Our product makes easier to use the distributed system. It also makes possible to handle computer resources flexibly. We will describe the system architecture and the idea of the swarm robotics control with the system in the technical report.

Keywords: distributed system, distributed computing, virtual machine, network system, swarm robotics

1. はじめに

我々の周りには、スマートフォンやパーソナルコンピュータ、家電など様々なコンピュータが存在する。現在、殆どの家電はインターネットに接続されず、独立・単体で動作している。一方で、スマートフォンやパーソナルコンピュータは、インターネットに接続され、世界中の情報へのアクセスや、多人数のコミュニケーションの媒介に利用されている。

複数のコンピュータがネットワークを通して連携し、動作する構造・仕組みを分散システムという。分散システムには様々な種類があり、その種類や実装により可用性やスケーラビリティ、性能向上などの面で様々なメリットがある [1]。例えば、データベース製品の一部は、複数のコンピュータを用いてクラスタリングを行うことで、スケーラビリティと可用性を向上させる [2][3][4]。また、SETI[5] や BOINC[6] は、クライアント・サーバーとグリッド・コンピューティングを組み合わせることで、一般家庭にあるコンピュータの処理能力を統合し、大量の科学技術計算を処理する。

分散システムは、複数のコンピュータがあれば実現できるわけではない。通常のプログラムは、単体のコンピュータでのみ実行されるように作られているため、そのままでは分散システムとして動かすことはできない。また、複数のコンピュータを用意して、プログラムを単に並列に実行しても、処理時間が短縮する、又は処理できるデータが増えるということはない。

分散システムでは、複数のコンピュータの CPU・主記憶・外部記憶などのリソースを相互又は一方的に連携している。例えば、データベースのクラスタでは、複数のコンピュータ上で主記憶・外部記憶に記録されているデータに齟齬が生じないように相互に連携している。

そのため分散システムは、個々の目的に合わせた専用のアルゴリズムを実装する必要がある。例えば、リレーショナルデータベースのクラスタでは 3 層コミット [7]、Key Value Store やドキュメントデータベースではリーダ選出のアルゴリズム [8][9] を実装している。

より汎用的な分散システムの実装のアプローチとして、ストレージや主記憶へのデータの保存に着目したシステムがある。Hadoop[10] は、MapReduce というプログラミングモデル [11] に従い、計算の処理過程や結果を分散ファイルシステムに保存することで、一貫したデータをシステムで共有し、並列計算を行う分散システムである。

また、松本らの MBCF[12] および SSS-PC[13] は、分散共有メモリおよびプロセスのライブマイグレーションを用いて、分散サーバを実現するシステムである。

これらのシステムは、専用のアルゴリズムの実装に比較すればプログラムの開発が容易だが、十分なメンテナンスが可能なサーバや、特定の環境を要求している。

我々は、スマートフォンやパーソナルコンピュータ、IoT ゲートウェイのようなヘテロジニアス環境において、汎用性の高い分散システムを実現するため、PROCESSWARP というシステムの開発を行っている。PROCESSWARP は、ネットワークを介して接続された複数のコンピュータ上に、仮想的な単一のプログラム実行空間を作成するシステムである。仮想的なプログラム実行空間を用いることで、通信機能を隠蔽し、CPU・主記憶の柔軟な連携が、容易に可能な環境を提供する。加えて、連携するコンピュータが動的に変化するような不安定な屋外のネットワークでの利用も想定しているため、今まで開発が難しかった複数のロボットの連携をとるようなシステムへの利用が期待できる。

本稿では、PROCESSWARP の特徴とアーキテクチャについて述べる。また、PROCESSWARP の利用例として、群ロボット制御の構想について述べる。

*1: 炎重工株式会社 研究開発部
*1:R&D Dept, Homura Heavy Industries Corporation.

2. PROCESSWARP

2.1 はじめに

インターネットを始めとするネットワーク技術の進歩により、複数のコンピュータ上でデータの柔軟な連携・利用が可能になった。また World Wide Web（以下「WWW」という。）の登場により、人間が可読・利用可能な情報やデータの共有・アクセス可能になった。

それに対し PROCESSWARP の目的は、コンピュータの計算リソースの結合および、それを容易かつ柔軟に利用可能にすることである。WWW が HTML ファイルの転送により情報の柔軟な利用を可能にしたように、PROCESSWARP は実行中のプログラムを転送することで、計算リソースの柔軟な利用を可能にする（表 1）。

表 1　WWW と PROCESSWARP の比較

		方法
WWW	データ・情報	ファイル・データの転送
PROCESSWARP	計算リソース	実行中のプログラム・機能の転送

PROCESSWARP での計算リソースの結合・利用は、同程度の計算リソースをもつ互換性のあるコンピュータ間だけではなく、豊富な計算リソースをもつコンピュータと、そうでないコンピュータ間や、CPU の命令や OS が異なるコンピュータ間などでの利用も想定する。例えば、スマートフォンとパーソナルコンピュータのような、利用シチュエーションやハードウェアアーキテクチャが異なるコンピュータが混在するような環境である。PROCESSWARP は、そのような異なるアーキテクチャのコンピュータ間においても、計算リソースを結合・利用するために、実行環境に依存しない仕組みになっている。

PROCESSWARP は、複数のコンピュータにインストールした専用アプリケーションを用いて、仮想的な 1 つのプログラム実行空間を作る（図 1）

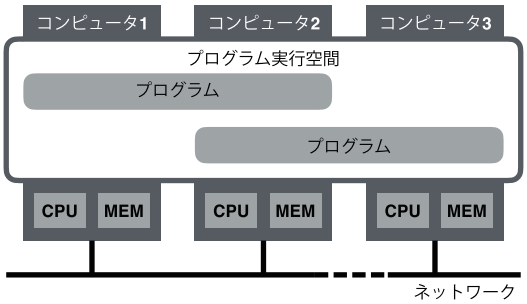


図 1　PROCESSWARP のプログラム実行空間

他のコンピュータの計算リソースを借りるということは、仮想的なプログラム実行空間内で、プログラムが他のコンピュータに移動するということである。その動作は、プログラムの中断やデータの保存を意識することなくシームレスに行われる。

PROCESSWARP は、プロセスを構成するスレッドを、その動作状態を保ったまま、他のコンピュータへ移動することで、実行中のプログラムを移動し、計算リソースの結合・利用を実現している。

2.2 ネットワーク構造

インターネットで利用可能なネットワーク構造は、大きくクライアント／サーバ型と P2P 型に大別される。

クライアント／サーバ型は、WWW などで一般的に使われており、データの集中管理に適しているが、クライアントの数が増えるに従いサーバへの通信・処理が集中し、パフォーマンスが低下する。

P2P 型は、Skype[14] などで使われており、クライアント同士が直接通信を行う構造になっている。パフォーマンスの低下が起こりにくいが、データの集中管理は難しい。Skype では、音声通話機能は P2P 型、アカウントの管理などはクライアント／サーバ型と、組み合わせて利用している。

PROCESSWARP では、個々のコンピュータの持つ計算リソースが柔軟に利用できる環境を提供することを目的としているため、各コンピュータ同士で直接通信を行う方が適している。また、後述するメモリなどの機能の実現には、低レイテンシかつ頻繁な通信がの方が適しているため、P2P 型である。また、ログイン機能や web を通した通信、コンピュータ同士の直接通信を行うまでの接続情報の交換などの機能はサーバを利用して実現するため、Skype と同様に、クライアント／サーバ型とP2P 型の両方を併用している。

2.3 仮想マシン

PROCESSWARP では、異なるアーキテクチャのコンピュータ間においても同一のプログラムを実行する環境を提供するため、プログラム実行環境を仮想マシンという形で実装している。仮想マシンとは、コンピュータの中に別のコンピュータの動作をエミュレートするソフトウェアである。仮想マシンを利用することにより、実コンピュータのアーキテクチャに依存しない仮想的なプログラム実行環境を作ることが可能である。そのため、スマートフォンとパーソナルコンピュータが混在するヘテロジニアス環境でも、プログラムに一定の環境を提供することが可能になる。

仮想マシンには、VMware 社製品 [15] のようなハードウェア仮想マシンと、Java 仮想マシンのようなプロセス仮想マシンの 2 種類がある。ハードウェア仮想マシンはコンピュータそのものをエミュレートする仕組みである。ハードウェア仮想マシンのインスタンス内では、実在するコンピュータと同様に、ゲスト OS・ミドルウェア・アプリケーションを実行することが可能である。プロセス仮想マシンは単一のプログラム実行環境として動く仮想マシンである。仮想マシン上でプログラムが直接動くので、ハードウェア仮想マシンに比べ軽量だが、専用の形式へコンパイルを行う必要がある（2.4 節で詳述する）。

PROCESSWARP では、分散システムを軽量に実現するためにプロセス仮想マシンとして実装している。プロセス仮想マシンは軽量なため、動作可能な環境のハードルを下げると共に、分散システム固有の機能を API という形で提供できるメリットがある。

一方で、ハードウェア仮想マシンを分散システム上に実現する場合、ゲスト OS を実行するため既存の OS 上で動くアプリケーションが流用可能というメリットがある。しかし、割り込みや I/O のエミュレーションなど、分散システム上のプログラムのためではなく、ゲスト OS のために必要な機能を実装し、ゲスト OS を動かすためのリソースが必要になる。

2.4 実行可能なプログラム形式

プロセス仮想マシンは、バイトコードと呼ばれる形式の、仮想マシンで解釈可能な命令セットを含むファイルを読み込み実行する。バイトコードは、それぞれの仮想マシンに最適化されており、人が直接編集するには適さない。そのため、人が理解しやすいプログラムからコンパイルを行って生成する。例えば Java の場合、Java 仮想マシンで実行可能なバイトコードを、Java 言語で書かれたプログラムから専用のコンパイラを利用してコンパイルする。また、プロセス仮想マシンが実行するのはバイトコードであるため、対応するバイトコードにコンパイルできれば、プログラム言語は問わない。

逆に、特定の言語で書いたプログラムを複数のプラットフォーム向けに変換することも考えられる。プロセス仮想マシンではないが、GCC[16] は、C/C++ 言語で書かれたプログラムを、複数のアーキテクチャ・プラットフォーム向けに出力可能である。

しかし単純に考えると、バイトコードとプログラム言語の組み合わせだけコンパイラが必要になり、新しいプログラミング言語や新しいプログラム実行環境を開発するコストが高くなる。

プログラミング言語とプログラム実行環境の組み合わせの問題を解決する方法として、プログラム言語を一旦共通の中間表現に置き換え、中間表現をそれぞれのプラットフォーム向けに変換する方法がある。LLVM [17] は、LLVM－IR という共通の中間表現を利用し、様々なプログラム言語とプラットフォームでの、コンパイラ基盤の実現を目的としたプロジェクトである。例えば、新しいプログラミング言語を開発する場合、その言語から LLVM-IR へ変換する仕組みを作ると、LLVM-IR から実行可能形式への変換は LLVM-IR を処理可能な既存のツールを使い実現される。逆に、新しいハードウェアや CPU を開発する場合、LLVM-IR からターゲットとなる命令へ変換する仕組みを作ると、既存のプログラミング言語からのコンパイルが可能になる。

PROCESSWARP の仮想マシンは LLVM-IR をロード・実行するため、既存のプログラミング言語で開発されたアプリケーションの実行が可能である（図 2）。

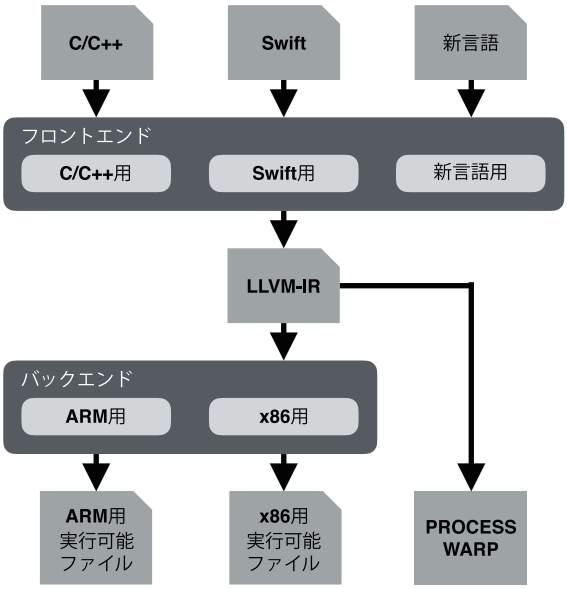


図 2　LLVM-IR と PROCESSWARP の関係

2.5 分散共有メモリ

PROCESSWARP では、分散共有メモリを利用することで、個々のプログラムにおいて通信機能を実装しなくとも、他のコンピュータと値を共有することが可能である。

一般的なプログラムが動作する時、プログラムの命令によりレジスタや主記憶などのメモリにアクセスし、値の読み出しや変更を行う。PROCESSWARP の仮想マシンは、プログラムがメモリアクセスする際、他のコンピュータ上で動く仮想マシンと協調し、分散共有メモリの値を読み書きする。そのため仮想マシン上で動くプログラムは、複数コンピュータ上で分散して動作するときも通信について考慮する必要はない。複数コンピュータ上で、複数のスレッドをそれぞれ動作させ、通常のプログラムと同じように変数を書き換えると、それが他のコンピュータ上の仮想マシンのメモリにも反映される。通信の前後でプログラム・モジュールを分割する必要がなくなるため、プログラム開発の難易度を抑えられる。

また、分散共有メモリを利用することで、1 台のコンピュータの主記憶容量以上のメモリを利用することが可能である。複数のコンピュータにメモリのデータを分散して配置することが可能なため、複数の少ない主記憶を持つコンピュータを連携させて、大きなデータを取り扱うことが可能になる。

PROCESSWARP で実装している分散共有メモリのデメリットとして、メモリへのアクセスがネットワーク越しに行われるため、レイテンシが増加する。通信によるレイテンシの増加は、プロセス間通信を行う他の分散システムでも発生するが、PROCESSWARP の分散メモリではメモリアクセスのたびに通信が発生する可能性があるため、その影響が大きい。

PROCESSWARP では、通信回数の削減とレイテンシの改善のため、キャッシュを用いている。複数の仮想マシンにキャッシュがあるため、同時に同じ変数に対して読み書きを行った場合、齟齬が発生する可能性がある。そこで、PROCESSWARP の分散共有メモリでは、メモリの整合性を取るためにリーダ選出アルゴリズムとロックを用いて、単一のコンピュータ上で動くマルチスレッドプログラムと同様の書き込み順番を再現する（図 3、図 4）。

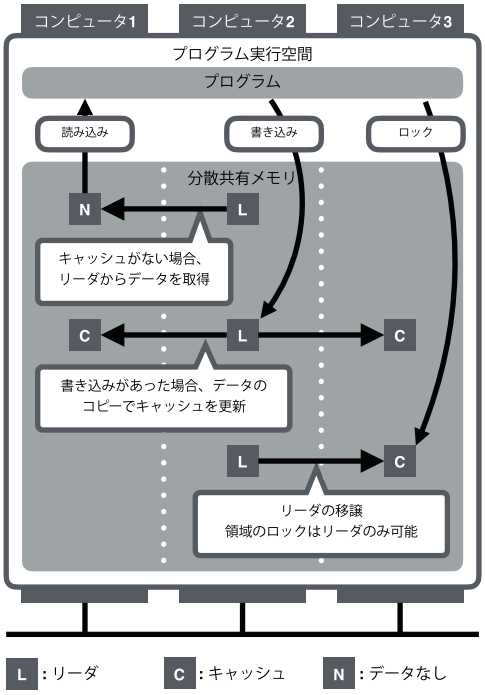


図 3　リーダとキャッシュの同期

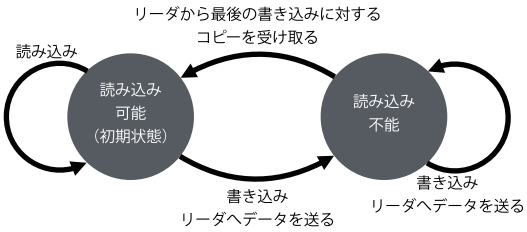


図 4　キャッシュに対するメモリアクセスの状態遷移

2.6　スレッドの移動

Linux や FreeBSD などのマルチタスク OS では、プログラムを実行すると、1 つ以上のスレッドをもつプロセスを作る [18]。スレッドはプログラムの実行の単位であり、1 つのプロセスに複数のスレッドを持つ場合もある。複数のスレッドを持つことで、同時に複数の作業を実行して、CPU を効率よく利用できる可能性がある。例えば、動画の編集を行いながら動画の書き出しを行う場合、実装にもよるが、2 つ以上のスレッドを持つことで入力と書き出しを同時に行い、書き出し時間を短縮する。

マルチタスク OS は、プロセスごとの実行状態やメタデータを、プロセス制御ブロックという領域に保存している。プロセス制御ブロックには、プログラムの名前などのメタデータ、各スレッドがプログラムのどこを実行しているか、スレッドが実行・休止状態など、プログラムの実行に重要なデータが保存されている。また、関数の呼び出し情報はスタックという領域に保存している。

PROCESSWARP では、プロセス制御ブロックやスタック領域を分散共有メモリに格納し、どの仮想マシン上でスレッドを実行するかをコントロールすることで、スレッドの移動を実現している。より詳細に説明すると、仮想マシン間で以下のような制御が行われている（図 5）。

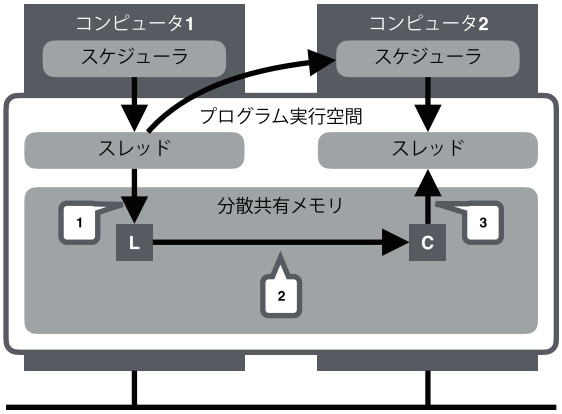


図 5　スレッドの移動を行う時の制御

1. 移動元の仮想マシン上でスレッドが動作しているとき、仮想マシンがスレッドに対応したプロセス制御ブロックをロックし、領域を優先的に利用している。
2. スレッドが移動するとき、移動元の仮想マシンでキャッシュを書き込み、ロックをリリースする。
3. 移動先の仮想マシンでは、プロセス制御ブロックの情報を仮想メモリから読み込んで、スレッドをアクティブにする。

ただし、スレッドの移動を行うためには、関数呼び出しを含めた一切の情報を移動先に渡し、移動前の状態を復元する必要がある。一般的なプロセス仮想マシンでは、仮想マシン上の関数呼び出しを実行する際に、ネイティブ関数を入れ子状に利用するものがある (Python ではコルーチンなどの機能を実装するために、C スタックを使わない実装が別に存在する [19])。例えば、バイトコードに CALL 命令があった場合、CALL 機能を実現する C の関数が呼ばれ、その関数がオペランド（バイトコードにおける命令であるオペコードに対する操作対象・引数）を元に仮想の関数を実行する。しかし、C のスタック状態は実際のコンピュータのアーキテクチャに依存するため、他のコンピュータでその状態を再現することが難しい。PROCESSWARP の仮想マシンでは、関数呼出し時に C の関数が入れ子状に呼ばれないように作られているため、オペコードとオペコードの間にスレッドの移動を行えば、他の仮想マシン上でも関数呼び出しの状態が復元される（図 6、図 7）。

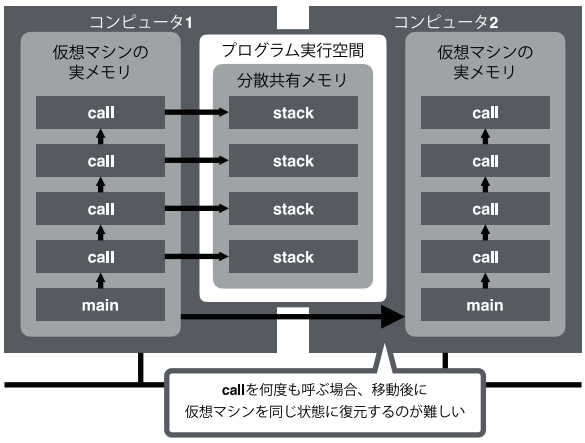


図 6　スタックを利用する場合の仮想マシンのメモリ

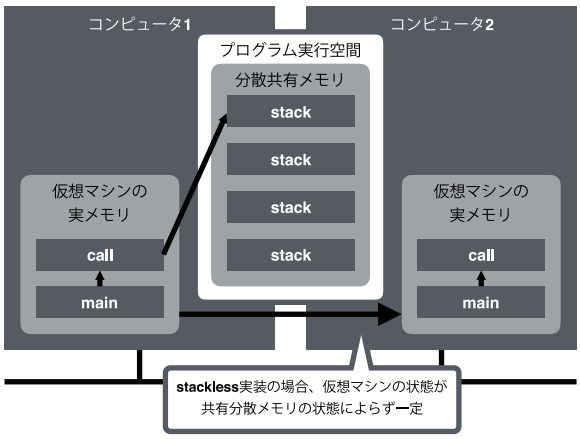


図 7　スタックを利用しない場合の仮想マシンのメモリ

3.　群ロボット

3.1　はじめに

コンピュータの利用により、計算・文書作成・ワークフローなどの情報を対象とした作業の効率化は進んでいる。一方、物理世界での作業の効率化は、作業を自動的に行うためのロボットが、その役割を担うことになる。ロボットには、工場などで製品の組み立てを行うロボット、コミュニケーションロボット、移動ロボットなど様々な種類がある。

群ロボットとは、複数の個体が連携を取りながら動作するロボットである [20]。複数のロボットが近くにあり、特にそれが移動している場合、互いの作業・目的を達成しつつ、衝突を回避する必要がある。また、複数のロボットを連携させることで、単体のロボットより効率的に作業を行える場合がある。例えば、災害時の生存者の探索を行うドローンを考えると、複数のドローンを利用することで、単体のドローンより短時間で広範囲を探索することが可能である。

単一で高機能なロボットを用いて作業を行う場合、そのロボットが故障すると作業ができなくなるが、群ロボットの場合は複数のロボットで構成されるため、１台が故障しても全体では作業可能というメリットがある。また、さまざまな規模の作業に対応しなければならない場合、ロボットの大きさや機能ではなく、台数を増やすことで対応が可能なため、柔軟性な運用が可能である。

例えば AutoStore[21] は、群ロボットを用いた倉庫システムを展開している。規模や間取りに合わせて調整することで、異なる倉庫でも同一のロボットを利用することが可能である。

複数のロボットを用いて１つの大きな荷物を運ぶ研究も行われている [22]。運搬する荷物の大きさや重さに合わせて運搬するロボットの台数を調整できれば、運搬する荷物の大きさが異なる場合でも同一のロボットを用いることが可能になる。ロボットを動かすためのエネルギーの面で、大きなロボットで全てを運搬する場合に比べ効率的である。また、メンテナンスの面で、大きさの異なるロボットを利用する場合に比べ容易である。倉庫での運搬以外にも、災害時の救助活動、屋外や宇宙空間での探索などでの活用が期待され、研究が行われている。

3.2　群ロボットと制御

群ロボットは、中央集中制御に頼らない制御を実装することが理想である。群制御では、ロボットは自分の周辺のセンシングによって得られた情報（例えば、近くの障害物など）と、自分の周囲の別のロボットからの情報（ロボット同士の位置関係、単体でセンシングできない障害物など）により、次の行動を決定する。このとき、特定のロボットに機能を依存、または処理が集中しないアルゴリズムを利用することで、耐障害性やスケールなどのメリットを実現している。

しかし、全体の大きな目標の決定や停止、利用者からの入力などのために、集中制御と組み合わせて利用する場合がある。集中制御では、ロボットの情報をサーバやリーダとなるロボットに集め、そこから制御を行う。分散制御と異なり、全ロボットが集中制御を行うサーバやリーダと通信可能な状態を維持する必要がある。また、サーバやリーダが故障すると群の制御ができなくなるため、単一障害点となってしまう。さらに、ロボットがサーバやリーダから通信経路的に遠い場合や計算リソースが十分にない場合、応答時間が長くなってしまうため、即応性が求められる処理には向かない。

分散制御が一方的に優れているように思えるかもしれないが、分散制御では実装が難しい機能もあるため、目的や実現したい機能・要件により両方の制御方法を組み合わせる必要がある。

3.3 群ロボットへの PROCESSWARP の適用

PROCESSWARP を群ロボットへ適用し、個別のロボットが持つCPUとメモリを統合することで、仮想的なプログラム実行環境で集中制御を行うことが可能になる。群を構成するロボットが存在すれば、仮想的なプログラム実行環境が維持されるため、単一障害点となるサーバやリーダなどを利用せずに集中制御を行うことが可能となる。また、ロボットの数が増えるに従い、利用可能なCPUとメモリが増える。

プログラムの構造に目を向けると、集中制御のプログラムが実行できるだけでなく、命令の送信などを行わずにロボットのコントロールが可能になる。各ロボット上では PROCESSWARP の仮想マシンが動くため、ロボットへの命令を行う API を用意することにより、PROCESSWARP 環境の中で動くプログラムから直接 API を呼び出すことが可能になる（図 8）。そのため、単純な集中制御のプログラムを、より容易に実装できる。

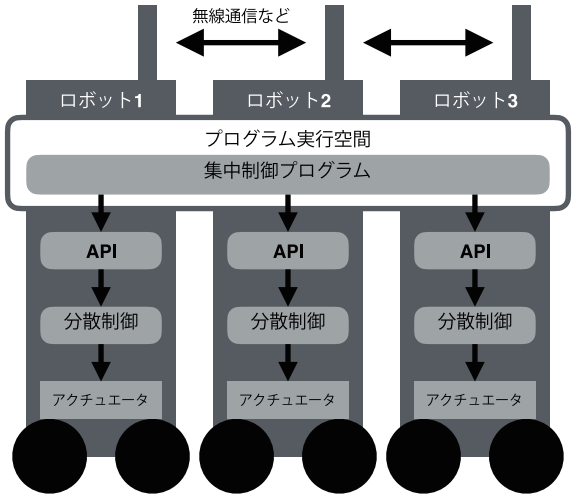


図 8　集中制御プログラムからの API の呼び出し

4. おわりに

本稿では、計算リソースの柔軟な利用を可能にする分散システムである PROCESSWARP の特徴と、その仕組について述べた。また、PROCESSWARP の利用例として、群ロボットへの適用の可能性を示した。

今後、PROCESSWARP の研究を進め、より汎用的なプログラム記述方式への対応や高速化、安定性の向上などを行う。また、実証実験などを行い、実用化を目指す。

謝辞

本研究の一部は、総務省 平成 27 年度 異能 vation によった。

参考文献

[1] Andrew S. Tanenbaum, Maarten van Steen: 分散システム原理とパラダイム ; ピアソン・エデュケーション , (2009)

[2] Oracle Corporation: Oracle RAC の概要 ; https://docs.oracle.com/cd/E16338_01/rac.112/b56290/admcon.htm

[3] IBM: DB2 データベース・クラスタリング ; https://www.ibm.com/support/knowledgecenter/ja/SSZJPZ_11.5.0/com.ibm.swg.im.iis.productization.iisinfsv.ha.install.doc/topics/wsisinst_pln_ha_xmeta_db2.html

[4] Oracle Corporation: MySQL Cluster 機能と特徴 ; https://www.mysql.com/jp/products/cluster/features.html

[5] UC Berkeley: SETI@home; http://setiathome.berkeley.edu/

[6] UC Berkeley: BOINC; http://boinc.berkeley.edu/

[7] Dale Skeen, et al: A Formal Model of Crash Recovery in a Distributed System; IEEE Transactions on Software Engineering, pp.219-228, (1983).

[8] Leslie Lamport: The Part-Time Parliament; ACM Transactions on Computer Systems, pp.133-169, (1998).

[9] Diego Ongaro, et al: In Search of an Understandable Consensus Algorithm; Stanford University, 2014 USENIX Annual Technical Conference, pp.305-319, (2014).

[10] Apache Hadoop; http://hadoop.apache.org/

[11] Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters; Google, Inc., (2004)

[12] 松本尚 : A Study on Memory-Based Communication and Synchronization in Distributed-Memory System; 東京大学大学院理学系研究科 テクニカルレポート TR01-01, (2001)

[13] 松本尚 , 渦原茂 , 竹岡尚三 , 平木敬 : 汎用超並列オペレーティングシステムカーネル SSS-CORE; 第 17 回技術発表会論文集 , 情報処理振興事業協会 , pp. 175-188, (1998)

[14] Salman A. Baset and Henning Schulzrinne: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol; Columbia University, (2004)

[15] VMware: vSphere Hypervisor; http://www.vmware.com/jp/products/vsphere-hypervisor.html

[16] GCC, the GNU Compiler Collection; https://gcc.gnu.org/

[17] LLVM; http://llvm.org/

[18] 高橋裕和 , 小田逸郎 , 山幡為佐久 : Linux カーネル 2.6 解読室 ; SoftBank Creative, (2006)

[19] Christian Tismer: Continuations and Stackless Python

[20] Marco Dorigo, et al.: Swarm robotics; Scholarpedia, (2014); http://www.scholarpedia.org/article/Swarm_robotics

[21] AutoStore; http://autostoresystem.com/

[22] 宮田 なつき : 群ロボットによる異種作業割り付け型協調搬送 ; 電気学会論文誌 C , Vol. 120, (2000)

Feature
3

生体群制御とロボット養殖

古澤 洋将*1　佐々木 義一*2

Grand design of Robotic Aqua Farming by Electric Fish-cluster Control

Yosuke Furusawa*1, Gichi Sasaki*2

Abstract - Focusing on bio-electro signals, we developed a fish-cluster control technology and system that guides and fixes fish-cluster in water including fish and shellfish, crustacea, underwater mammals and the like to arbitrary positions in water using electricity. As an example of applying the fish-cluster control technology to a larger scale, we showed the idea of robotic aqua farming intended not to use a ship.

Keywords: automate aqua sea farming fish-cluster control guide robot

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

購入者限定

本ページはご購入後の本誌にてお楽しみください。

執筆者紹介

古澤洋将（ふるさわ ようすけ）　岩手県滝沢村出身／1982年生

大型・大特・牽引自動車運転免許、1級小型船舶操縦士、危険物乙種全類取扱者、第四級アマチュア無線技士などの資格を保有。筑波大学大学院システム情報工学研究科を修了し、CYBERDYNE株式会社に入社。ロボットスーツHAL福祉用及び医療用の電装系設計に従事。各種認証取得、製品上市を経験する。東日本大震災を機に退職・帰郷し、炎重工株式会社を設立。

伊藤祐司（いとう ゆうじ）　岩手県滝沢村出身／1984年生

岩手大学大学院工学研究科を修了し、日立東日本ソリューションズに入社。業務システムやグループ会社内での研究・新規事業立ち上げ時のプロトタイプの開発に従事。会社に務める傍らPROCESSWARPという分散処理システムの開発を行っており、総務省主催の異能vationプログラムに「PROCESSWARP分散処理システムの開発」というテーマで採択される。
2016年、炎重工株式会社に入社。

佐々木義一（ささき ぎいち）　岩手県山田町出身／1940年生

漁師。15歳の頃より漁に出て、60年以上のキャリアを持つ。東日本大震災で漁船を失うも、現在は新たな船と共に漁に出る日々。

炎重工株式会社
HOMURA HEAVY INDUSTRIES

会社概要

会 社 名 ：炎重工株式会社（Homura Heavy Industries Corporation.）

代表取締役：古澤 洋将（Yosuke FURUSAWA）

本社所在地：〒020-0633 岩手県滝沢市穴口 408-10

Ｔ Ｅ Ｌ ：050-7117-5702

Ｆ Ａ Ｘ ：019-618-7562

Ｍ ａ ｉ ｌ：info@hmrc.co.jp

Ｗ ｅ ｂ ：https://www.hmrc.co.jp/

資 本 金 ：3,400万円（資本準備金を含む）

取引銀行：三菱東京 UFJ 銀行 本店
岩手銀行 滝沢支店
盛岡信用金庫 本店

設 立 ：2016年2月25日

事業内容：(1) 製品の企画・開発・生産・販売・保守・賃貸・受託・輸出入・コンサルティング業
(2) 労働者派遣事業法に基づく一般労働者派遣事業、特定労働者派遣事業
(3) 書籍、記事等の執筆・出版・印刷業
(4) 上記に付帯し、または関連する一切の業務

沿 革

2013年1月　屋外ロボットの開発開始

2016年2月　炎重工株式会社を設立

2017年1月　生体群制御付き水槽　アクトリウムを発売

2017年3月　資本金を3,400万円に増資（資本準備金を含む）

表紙について

表紙の数式は、1619 年にドイツの天文学者ヨハネス・ケプラーによって発見された衛星・惑星の運動法則です。彼が生きていた時代の天文学は、天動説が主流であるものの、それに疑問を抱く学者もいる状況でした。同じ時代、ポーランドの天文学者ニコラウス・コペルニクスらは地動説を唱えていましたが、惑星は中心の星の周囲を完全な円軌道で運行すると考えていたため、惑星の動きを完全には説明することができていませんでした。その後、デンマークの天文学者ティコ・ブラーエによる膨大な天体観測記録をもとに、ヨハネス・ケプラーが法則を発見したことにより、地動説よりも惑星の動きを正確に説明できるようになりました。

ケプラーの法則
第 1 法則（楕円軌道の法則）：
惑星は、太陽をひとつの焦点とする楕円軌道上を動く。

第 2 法則（面積速度一定の法則）：
惑星と太陽とを結ぶ線分が単位時間に描く面積は、一定である。

第 3 法則（調和の法則）：
惑星の公転周期の 2 乗は、軌道の長半径の 3 乗に比例する。

また、現代の航空機や船舶、自動車、当社の屋外ロボットなどにも使われている GNSS（GPS とは本来、アメリカ合衆国によって構築された航法衛星システムのみを指します）は、複数の人工衛星を利用して地球上の現在位置を知るシステムです。つまり、私たちが GNSS を利用して位置を知るときは、ケプラーの法則を元にした計算を行っています。

本創刊号では、ヨハネス・ケプラーの発見によって天動説から地動説へ当時の科学界の常識が劇的に変化したことに敬意を表し、また現代のロボティクス・AI などによる社会・技術革新への期待を込めて表紙のデザインとしました。

炎重工技報

Homura Heavy Industries Technical Review

Vol.1
2017

2017 年 07 月 14 日　初版第 1 刷発行

発 行 人 ：古澤洋将

発 行 所 ：炎重工株式会社
〒020-0633 岩手県滝沢市穴口 408-10
TEL:050-7117-5702
FAX:019-618-7562
Mail:info@hmrc.co.jp
https://www.hmrc.co.jp/

デザイン：辻元気（株式会社ホップス）

印刷・製本：川嶋印刷株式会社

◆定価は表紙に表示してあります。
◆乱丁本・落丁本はお取り替えいたします。

炎重工株式会社
HOMURA HEAVY INDUSTRIES

定価：4,860円（税込）

炎重工株式会社
Homura Heavy Industries Corporation.
https://www.hmrc.co.jp/