産業技術短期大学誌　第 56 巻(2023)

Bull. Coll. Ind. Tech.
Vol.56, pp. 19 - 25 (2023)

研究論文

全方向移動車の速度制御方法の比較検討

畑迫　健一*

Study of speed control method for omni-directional moving vehicle

Kenichi HATASAKO*

Synopsis: An omni-directional vehicle was fabricated and the speed control method of the omni-directional vehicle was evaluated and examined. The speed was calculated by attaching a rotary encoder to the motor and measuring the number of revolutions of the motor. Two types of evaluation were performed: evaluation for a constant speed command value and evaluation when the command speed was changed. In both evaluations, feed forward (FF) control, FF + proportional (P) control, and FF+P integral (I) control improved in the order of fidelity and response speed to command values. We were able to confirm the effect of feedback control on omni-directional vehicles.

Regarding the experiment and evaluation, I wrote a part of the content that the students implemented in the class of the graduation training in the 3rd year of Reiwa in this paper.

(Received Sep. 22, 2022)

Key words: PID control, speed control, omni-directional moving vehicle, autonomous driving, ROS

1．緒　　言

安全性の向上や利便性の向上を目的に運転支援や自動運転のできる自動車の研究や開発が進められている [1]．自動運転を行うためには自動車の走行速度や進行方向を自動にて制御する必要がある．自動で制御するためには，それぞれの指令値に対して，実際の値（実測値）ができる限り近いことが求められる．

今回，前後や左右だけでなく斜め方向やその場旋回など任意の方向に移動することのできる全方向移動車 [2,3] を製作した．製作した全方向移動車は受動車輪の軸をらせん状に配置したメカナムホイールを使用したものである．この全方向移動車を使用し，進行速度を制御する実験を実施した．

速度の制御方法はあらかじめ取得したデータにて指示するフィードフォワード制御のみのときとエンコーダによりモータの回転数を検知し，指示した速度の回転数と比較を行うフィードバック制御を実施した．フィードバック方法は PI 制御について評価を行った．PI 制御は比例制御（Proportional），積分制御（Integral），を合わせた制御方法であり，今回の評価では 1 つずつ制御方法を追加して評価を実施した．PI（D）制御の有効性については既に知られているところである [4, 5]が，今回の評価においても PI（D）制御の有効性を確認する結果となった．

なお，本論文の内容である全方向移動車の速度制御の評価は令和３年度の本学電気電子工学科２年生が卒業研修の授業時間に実施した内容の一部をまとめたものである．本論文では紹介にとどめるが卒業研修では自動運転や地図の作成といった機能も実現した．

＊　産業技術短期大学教授　博士（工学）電気電子工学科

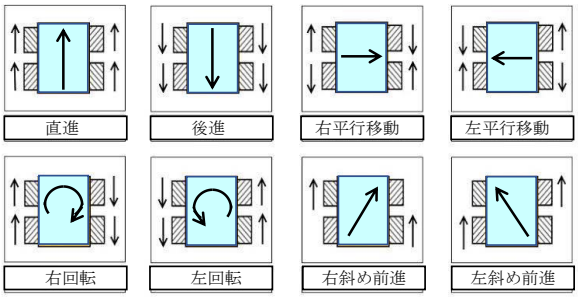
2．全方向移動車の製作

2.1 主な使用部品

(1) メカナムホイール

全方向への移動を実現するために Fig. 1 (a)に示すメカナムホイールを使用した．メカナムホイールでは車輪に対して 45° の角度でローラが付いている．このローラによりタイヤを回転させると車輪を取り付けた向き 45° 方向に力が発生するので 4 輪の回転方向を Fig. 1 (b)の様に組み合わせることで 8 方向へ移動することができる．

(a) Photograph of used Mecanum Wheel.



(b) The way to move in 8 directions. The direction of movement of the wheels (filled in for each wheel) and the direction of movement of the vehicle body (filled in the center) are shown.

Fig. 1 Photograph of the Mecanum wheel and the way how to move in 8 directions.

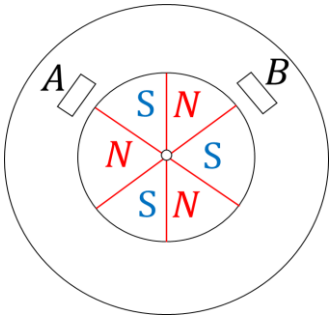
(2) 磁気式ロータリーエンコーダ

磁気式ロータリーエンコーダ（以下，エンコーダ）はモータの回転軸に永久磁石の円盤が取り付けられていて（Fig. 2 (a)），モータが回転すると永久磁石も回転し 2 つのホールセンサ A，B が永久磁石の磁極を検出する（Fig. 2 (b)）ことでモータの回転数を検出できる．モータが 1 回転すると 2 つのセンサが磁極を半分ずらして 6 極/周を検出するので，Fig. 2 (c)に示すように A，B の組み合わせで 12 カウント/周のパルスを検出することができる．モータのギヤ比は 75:1 であるため，タイヤ 1 回転でモータは 75 回転することになり，タイヤ 1 回転で得られるエンコーダのカウント数は

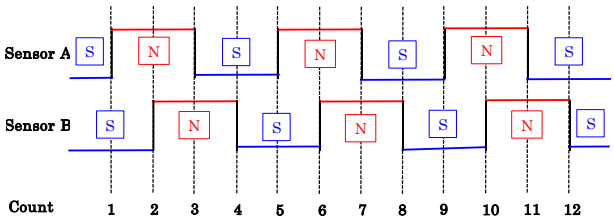
$$12[\text{カウント/周}] \times 75[\text{周}] = 900 \quad [\text{カウント}]$$

となる．

(a) Photograph of magnetic rotary encoder.



(b) Structure of a permanent magnet disk attached to the motor shaft and two Hall sensors (A, B) that detect it.



(c) Signal detection by A and B Hall sensors during one rotation of the motor.

Fig. 2 Photograph of a magnetic rotary encoder and the structure of the Hall sensors that detect rotation.

タイヤの周速度 v [m/s]とモータの回転数 n [1/s]との関係は，タイヤの直径を D [m]とすると，

$$v = \pi D n$$

であるので，使用したタイヤの直径 $D = 0.08$ [m]と $n = \frac{N}{900}$ の関係を代入すると

$$N = \frac{11250}{\pi} v \qquad (1)$$

N ：エンコーダのカウント数 [/s]

となり，(1)式よりエンコーダのカウント数 N とタイヤ速度 v との関係が得られる．

(3) Arduino マイコンボード
(Espressif Systems 社製 ESP32-DevKitC)

プログラムを作成してモータ制御を行うためにマイコンとして，Arduino マイコンボード（ Espressif Systems 社製 ESP32-DevKitC ）を使用した（Fig. 3）．プログラムは Arduino IDE より書き込みを行なった．各モータに対して，PWM 出力 2 本，エンコーダの入力信号 2 本が必要なためモータ 4 個で 16 ピンが必要であり，他のセンサやマイコンとの通信等を入れてもピン数は対応可能であるものを選択した．性能は最大クロック周波数:240MHz，RAM:520KB であり，Wi-Fi と Bluetooth が内蔵されている．

全方向移動車の製作に使用した装置や操作用のタブレット，使用した部品の仕様について Table 1 に示す．

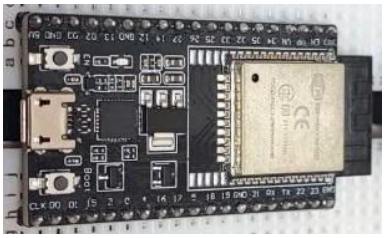


Fig. 3 Photograph of Arudino microcomputer board (Espressif Systems' ESP32 - Dev Kit C).

2.2 ハードウエアの製作

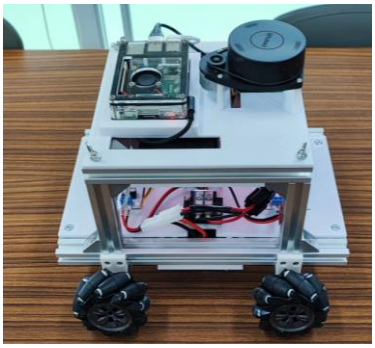
全方向移動車の完成写真を Fig 4 に示す．Fig. 4 (a)は上からの写真であり，1 階に Arudino マイコンやモータドライバ等を実装したプリント基板を配置している．2 階には Rasberry Pi と LiDAR（light detection and ranging）センサを配置した．

Table 1 Main equipment used and parts list.

品名	型名	仕様
3Dプリンター（パーツ作成用）	ANYCUBIC社 Chiron	最大加工寸法：400mm×400mm×450mm
タブレットPC（操作用）	東芝製 Dynabook V714/K	CPU:Intel i5-4210Y RAM:4GB OS:Ubuntu:18.04
Raspberry Pi4 ModelB	LANZHI社製 8 GB	Broadcom BCM2711、 Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz × 4 (4コア) 、 GPU:500MHz (2コア)
Arduinoマイコン	Espressif Systems社製 ESP32-DevKitC	・WiFi：802.11 b/g/n、 ・Bluetooth v4.2 ・デュアルコア Tensilica LX6、 ・SRAM：520KB ・電源電圧：2.2V~3.6V ・静電容量タッチインターフェース：x10 ・32kHzオシレータ：x1、・GPIO：x21 ・UART：x3、・SPI：x2
LiDARセンサー	SLAMTEC社 RPLiDAR A1M8	・測定周波数：2000Hz ・スキャン周期：標準5.5Hz (1～10Hz可変) ・測定距離（半径）：0.15～6m

Fig. 4 (b)は完成車の下からの写真であり，バッテリーを中央部に配置し，モータ，車輪等を確認できる．

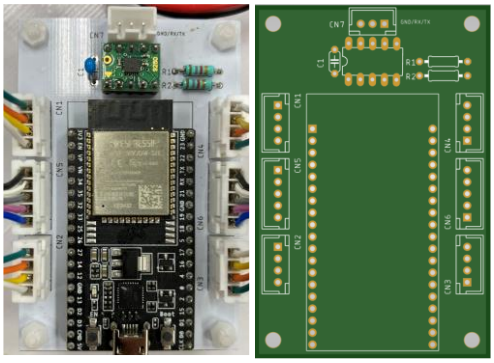
Arudino マイコンやモータドライバは信頼性向上のため，抵抗やコンデンサ等と一緒にプリント基板に実装した．



(a) Photograph from above.

(b) Photograph from below.

Fig. 4 Photograph of the completed omni-directional vehicle.



(a) Frontside photograph. (b) Backside photograph.

Fig. 5 Photographs of microcomputers mounted on a printed circuit board.

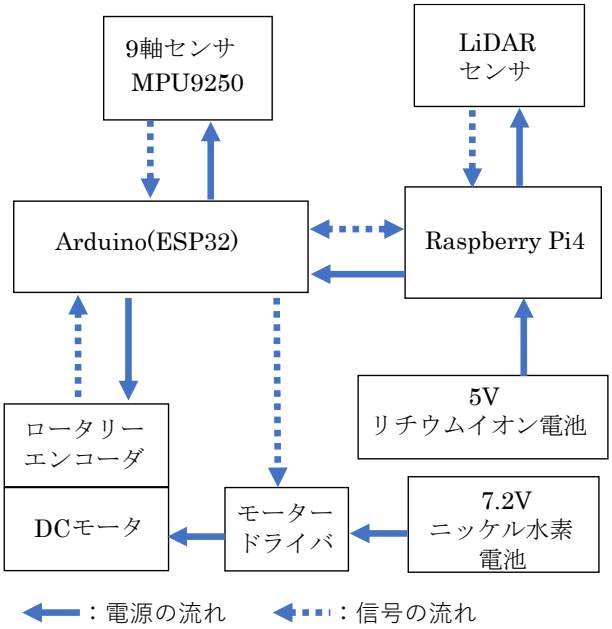


Fig. 6 Power supply and signal flow in the omni-directional vehicle.

Arudino マイコンボードを基盤へ実装した写真を Fig. 5 に示す．プリント基板の設計からはんだ付け等の製作まで全て学生たちが実施し完成させた．

Fig. 6 に完成車内での電源と信号の流れを示す．

2.3 プログラムの作成

Fig. 6 に示した信号の流れを実現するために，Arudino マイコンの中で「C 言語」と「C++」をベースにしたプログラミング言語を使用してプログラムを作成した[6]．プログラムの流れを示すフローチャートを Fig. 7 に示す．

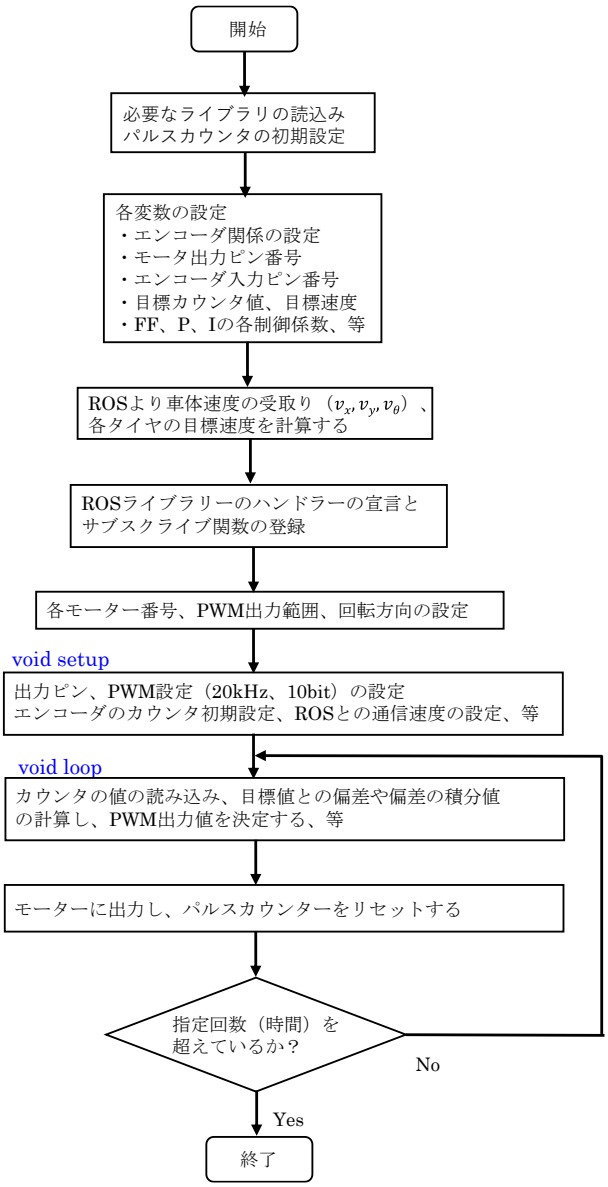


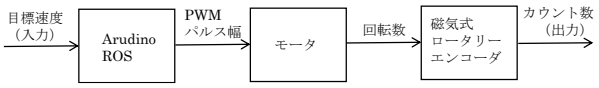
Fig. 7 Program flow chart created with Arduino microcomputer.

車体の速度は ROS（ Robot Operating System ）に車体速度を送信し，ROS にて直進方向(x, y)，回転方向(θ)の速度を計算した後，逆運動学を使用して各タイヤの設定速度を算出している[7, 8]．各タイヤの実際の速度はエンコーダによりモータの回転数を測定して(1)式を使用して求め，設定値との偏差や積分値を算出し各モータの PWM 出力パルス幅にフィードバックをかけた．このフィードバック部分は Fig. 7 の void loop の中で実施していて，繰り返しフィードバックを行うようになっている．

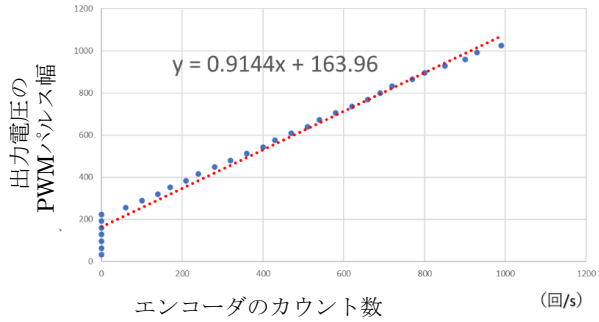
3.　速度の制御方法

3.1 フィードフォワード制御

フィードフォワード制御（以下，FF 制御）は Fig. 8（a）のブロック線図の構成になる．目標値は車体の速度であり，操作量は Arduino マイコンからの出力電圧の PWM(Pulse Width Modulation)パルス幅，出力はモータの回転数で決まるエンコーダのカウント数となり，一方向に順次指令が伝わる開ループ制御になる．モータが無負荷時の出力電圧の PWM パルス幅とエンコーダのカウント数の関係は Fig. 9 (b)となる．Fig. 9 (b)より PWM パルス幅と出力のカウント数は直線関係にあることがわかる．この関係を使用して目標速度の PWM パルス幅を指示した．



(a)　Feedforward system block diagram.



(b)　Relationship between voltage pulse width of PWM control and encoder count.

Fig. 8 Feedforward control system.

3.2 フィードバック制御

フィードバック制御時のブロック線図を Fig. 9 に示す．このフィードバック制御はエンコーダのカウント数をフィードバックし，目標値の速度より算出したカウンタ数と比較を行ない，実績値に応じた補正を行って操作量にフィードバックする閉ループ制御になる．

P（比例）制御についてはエンコーダのカウント数の実測値 $y(t)$ と指令値 $r(t)$ との差分をとり，Fig 8 (b)のグラフから求まる係数 K_p をかけてフィードバックした．P 制御でのフィードバック量 $f_p(t)$ は以下である．

$$f_p(t) = K_p(y(t) - r(t)) \qquad (2)$$

t：経過時間［s］

K_p：P 制御係数

$r(t)$：指令速度より算出したカウント数

$y(t)$：実際に測定したカウント数

I（積分）制御についてはエンコーダのカウント数の実測値 $y(t)$ と目標値 $r(t)$ との差分 $(y(t) - r(t))$ の時間積分を行ったものに係数を掛けてフィードバックを行った．I 制御でのフィードバック量 $f_I(t)$ は以下である．

$$f_I(t) = K_p \times \frac{1}{T_I}\int\big(y(t) - r(t)\big)dt$$
$$\cong K_I \sum(y_i - r_i) \cdot \Delta t_i \qquad (3)$$

K_I　：I 制御係数

T_I：プログラム実行時の void loop 1 周期の時間

Δt_i　：カウント数を測定する間隔

FF 制御と PI 制御のフィードバック制御を合わせた全体の指令量 $f(t)$ は FF 制御の指令量と(2)，(3)式を合わせると以下のようになる．

$$f(t) = b(t) + K_p\big(y(t) - r(t)\big) + K_I \sum(y_i - r_i) \cdot \Delta t_i \qquad (4)$$

$b(t)$：FF 制御による指令量

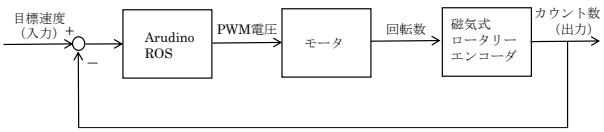


Fig. 9 Feedback system block diagram.

4．速度制御の評価結果

速度の指令を行い，実際の速度を評価した．実際の速度はエンコーダで測定した回転数より算出した．評価した速度は

・指令速度が一定のときの評価

・指令速度を変化させたときの評価

の 2 通りを実施した．速度制御については以下の 3 通りを評価した．

① FF 制御

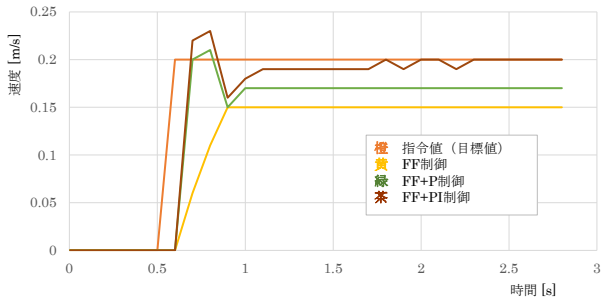
② FF＋P 制御

③ FF＋PI 制御

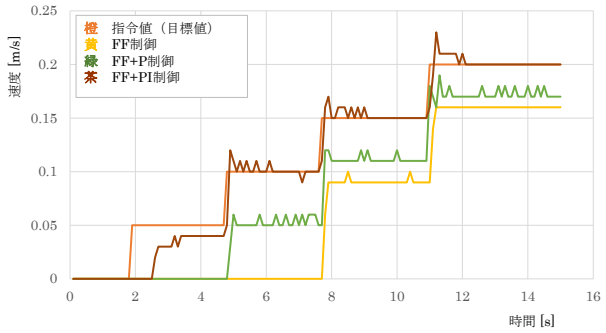
評価結果を Fig. 10 に示す．Fig. 10 (a)は一定速度に対する評価結果を示し，Fig. 10 (b)は指令速度を変化させたときの評価結果をそれぞれ示している．

Fig. 10 (a)の一定の指令速度に対する評価では，FF 制御は応答が遅く（遅延 0.38s），約 25%のオフセット（定常偏差）が発生し最終の値が大きくずれていることがわかる．FF＋P 制御では，FF 制御よりは応答は速くなりオフセットも小さくなったが，依然約 15%のオフセットが発生している．FF＋PI 制御では速度は最終的には指令値に一致しオフセットがなくなった．しかしながら，FF＋P 制御，FF＋PI 制御においても反応が遅い（遅延 0.20s）ことがわかった．

Fig. 10 (b)の指令速度を変化させたときの評価結果についても，Fig. 10 (a)とほぼ同じことが各評価について起こっている．異なる点は FF 制御，FF 制御＋P 制御の条件において，速度が遅いときには測定値が 0 になっていることである．



(a) Evaluation result for constant command speed.



(b) Evaluation result for changing command speed.

Fig. 10 Speed control evaluation result.

5．考　　察

1) FF 制御，FF 制御＋P 制御においては，指令値に対してそれぞれ 25%，15%という大きなオフセットが発生した．また，速度が遅いときに測定値が 0 という結果も出た．これらは無負荷時のモータ特性により負荷のかかったモータを制御したことや 4 つのモータへの負荷のかかり方が一様ではないこと等が原因であると考えられる．

2) 一定速度に対する評価と指令速度を変化させたときの評価の 2 つの評価を実施し両方において（FF 制御）＜（FF＋P 制御）＜（FF＋PI 制御）の順に指令値に対する忠実性が良く，応答も速いことがわかった．フィードバック制御の有効性を確認することができた．

3) 指令値への応答の遅れが発生したが，今回時定数を考慮した係数の決め方や係数間の最適化などが十分に検討できていなかったので，継続して検討を進める．

4) PID 制御の D 制御が今回欠落していた．FF＋PI 制御において今回はほぼ指令値を満たしていたが，D 制御は外乱や急激な変動に対して効果を発揮する[4), 5)]ため，今後，上記 3)の応答性の問題と合わせ機会を見て評価を実施する．

6．結　　言

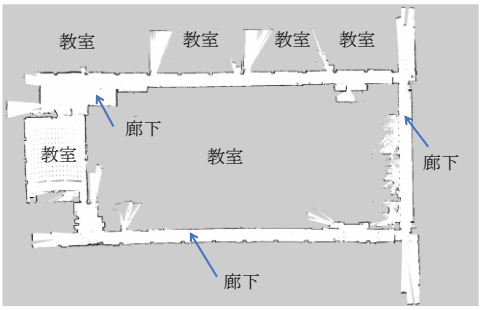
全方向移動車を作製し全方向移動車の速度制御方法について評価検討した．速度はモータにエンコーダを取り付けモータの回転数を測定して計算により算出した．指令速度が一定のときの評価と指令速度を変化させたときの評価の 2 通りを実施した．両方の評価において，指令値に対する忠実性と応答の速さにおいて，FF 制御，FF＋P 制御，FF＋PI 制御の順番に良くなった．全方向移動車に対するフィードバック制御の効果を確認することができた．

謝　　辞

本論文は令和 3 年度の卒業研修において得られた実験，実習結果の一部をもとにまとめたものである．卒業研修生（敬称略）は，砂田琢磨（JFE プラントエンジ（株）より派遣）がリーダとなり，今永亮，植野絢太，曽田悠太，松田 陽佑，山根 龍です．記して，各位に謝意を表します．

本論文の内容から外れますが，卒業研修では自動運転やマップの作成も実現した．その内容を Fig. 11 に示す．

(a) Autonomous driving that crosses obstacles.



(b) A map of a college building created using a LiDAR sensor.

Fig. 11 Functions realized other than this paper.

参考文献

1) 経済産業省：“自動運転に関する取組”，経済産業省ホームページ，<https://www.meti.go.jp/policy/mono_info_service/mono/automobile/Automated-driving/automated-driving.html>，（参照　2022-09-5）.

2) 多田隈建二郎：日本ロボット学会誌，20(2011)，516–519.

3) 榊正憲：2 輪駆動・オムニホイール・メカナムホイールの仕組みと制御，インプレス R&D，2019.

4) 斉藤制海，徐粒：制御工学，森北出版，2003.

5) 佐々木清吾：電気システムのための制御工学，オーム社，2017.

6) 平原真：実践 Arduino!，オーム社，2017.

7) 上田隆一：Raspberry Pi で学ぶ ROS ロボット入門，日経 BP 社，　2018.

8) 鹿賀悠多：Scamper による ROS & Raspberry Pi 製作入門，オーム社，2018.