

Donkey Car（自動運転車両）製作記

中里武彦

1 はじめに

近年の自動車技術の進化に伴い自動運転車両が一般社会に現れるのもけして遠いはなしではなくてきたように思われる。日々のニュースの中からも自動運転に係る新技術が次々と各自動車メーカーや、IT 関係の会社より聞こえてくる。

新たに登場すると思われる自動車は既存の自動車の持つイメージとは異なりドライバーの技量に頼るばかりではなく自動車が自ら何らかの制御により安全に走行できるロボットのようなイメージに移り変わりつつある。その制御にはコンピュータープログラムが大きく関係し、今まではパソコンの中だけの世界が我々の目の前に現れようとしている。

自動運転を行うモデルカーの存在を知り、製作に挑戦することにより自動運転への理解を深めたいと願い研究を開始した。

2 目的

自動車整備に係る者として、新技術への知見を深めることを目的とし本研究では、Donkey Car（以下ドンキーカーとする）を用いた自動運転車両を制作、走行することによりプログラミングなど新しい技術にふれ、車両制御に及ぼす影響を調査し報告する。

3 ドンキーカーとは

駆動装置を持つモデルカーを自動走行（自動運転）できるようにマイクロコンピューターボードをはじめ、各種のセンサーを取り付けプログラミングした車両である。現在世界的に多くの愛好家により開発が進められており、大枠としてはラジコンカー（以下 RC カーとする）と同様の機能を持つが、コントロールを受け持つ部品がマイクロコンピューターボードに置き換えられ、現在ラズベリーパイが主流を占めるが高性能機種のコンピューターも散見される。ラジコンカーでは人間による操作で走行を制御するのに対し、ドンキーカーではマイクロコンピューターボードによる AI 操作となり、車両本体の性能（ハードデバイス）とそれをコントロールするプログラミングにより制御される。

本研究では、公開されているドンキーカープログラムを使用し数年来保管されていた RC カー

を改造しドンキーカーを制作した。(以下 NAC ドンキーカーとする)

以下にドンキーカーと RC カーの制御比較を表 1 に表す。

表 1 ドンキーカー・RC カー比較

	ドンキーカー	RC カー
車両	モデルカー	モデルカー
原動機	モーター	モーター
駆動電圧	制御 5V, モーター別電圧	7.2V
送受信方式	Wi Fi	72MHz 電波
操作方法	AI による制御	プロポによる遠隔操作
認識方法	カメラ, 他センサー各種	目視

図 1-1 に NAC ドンキーカーを, 図 1-2 に RC カーの制御機構を示す。

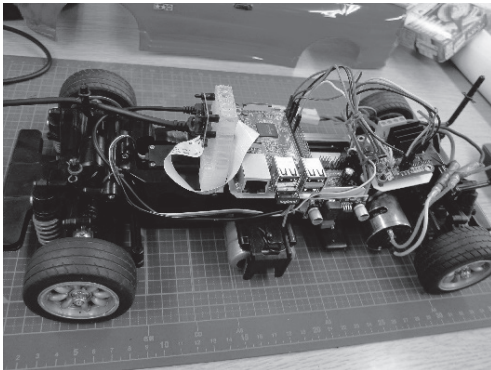


図 1-1 NAC ドンキーカー

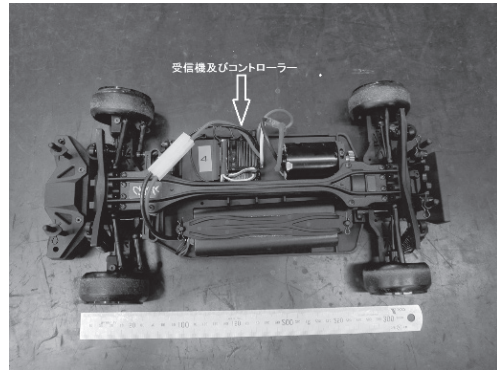


図 1-2 RC カー

図 (1-1, 1-2) から見て取れるように RC カーは構造がシンプルなのに対し, NAC ドンキーカーは電子基板が複数取り付けられ複雑なことがわかる。この電子基板と, プログラムにより, RC カーの矢印で示す受信機及びコントローラーと操縦者の働きを担うのである。

今回制作した車両はボディーを取り付けた乗用車を模した形としたが, ドンキーカーの多くはボディーを持たず基板がむき出しの車両が多い。

4 NAC ドンキーカー諸元

NAC ドンキーカー制作にあたり Web ページを参考にし, 製作実績のある電子部品を多く取り入れた, シャーシ及びモーターに係る部分では, 旧式ではあったが動作確認したところ実働したのでサーボ及びモーターは学内に保管されていた部品である。バッテリーに関しても当初は保

表2 NAC ドンキーカー諸元

項目	型式, 性能
シャーシ	タミヤ Mシャーシ (1/10スケール)
モーター	ストックモーター
サーボ	Futaba 53003
バッテリー	7.2v 3300mA
マイクロコンピューターボード	Raspberry Pi 3 Model B+ 1.4GHz
メモリー	Micro SD カード 128G
サーボドライバーモジュール	Hi Letgo PCA9685
モータードライバーモジュール	Hi litandgok 5c7 bxui
カメラモジュール	Raspberry Pi camera V 2
DC コンバータ (7.2v to 5v) モジュール	LM2596 DC-DC HW-411

管品であったが古いようで長時間使用することができないことが判明し新規に購入した。

NAC ドンキーカー制作にあたり使用した部品及び諸元を表2に示す。

5 ドンキーカー制作

5-1 環境準備

NAC ドンキーカー制作にあたり GitHub (github.com) 内の Donkey Car (https://coolerking.github.io/donkeycar_jpdocs/#donkey-cardonkey2) ページを参考に製作を開始した。

この GitHub とはソフトウェア開発のプラットフォームであり、様々な開発者がプログラムを公開し、プログラム開発に多くの人に参加することにより開発スピードを上げることを目的としている Web ページである。その中に Donkey Car の製作方法及びプログラムが記載されている。

Donkey Car プログラムはオープンソースであり、自由に使用可能なプログラムである為そのまま使用することとし、マイクロコンピューターボードも同ページで紹介されている Raspberry Pi 3 Model B+ (以下ラズパイとする) とした。ドンキーカーのプログラムはラズパイにインストールされるが、ラズパイにはハードディスクが無いため、Micro SD カードにプログラムをインストールしそのままラズパイのハードディスクとして使用する。

他の部品についても細部まで記載されているが、サーボドライバーは、Hi Letgo PCA9685 と指定があり指示通り購入することとした。

ドンキーカー制作には車両を準備するほかに、プログラムやドンキーカーを制御するためのパソコン (以下 PC とする) が別途必要になる。他にもドンキーカーと PC を通信させるための Wi Fi 環境も必要になるなど準備を進めた。NAC ドンキーカー使用部品構成を図2に示す。

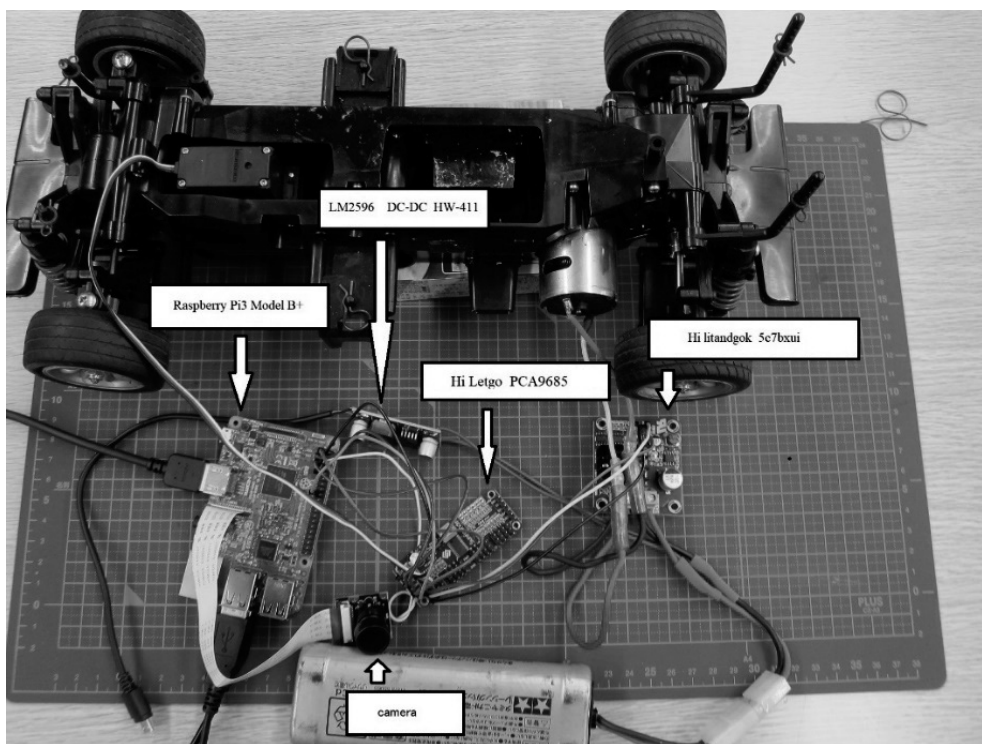


図2 NAC ドンキーカー構成

5-2 ドンキーカープログラム

ドンキーカーを制御するためには複数のプログラムを使用する必要がある。図3に概略を示す。

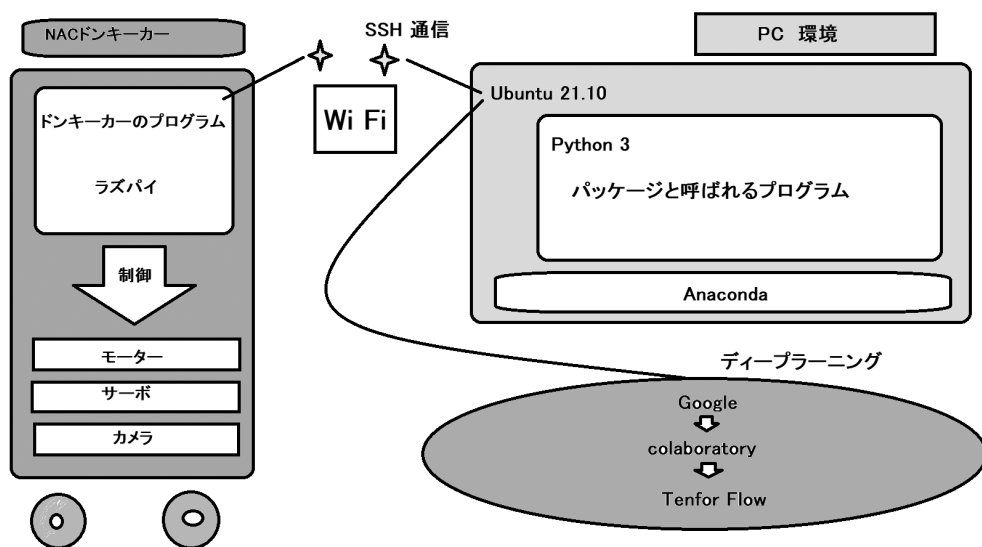


図3 プログラム概略

ドンキーカーは自動運転を目的としているが、電源を入れればすぐに自動運転が開始されるわけではなく運転開始までにいくつかの手順を踏み作動開始、終了などは別のPCより行う、通信方法はSSHと呼ばれるプロトコルにより行い直接ドンキーカーのプログラムを操作するのである。

ドンキーカーのプログラムはPython プログラム言語で記述されておりPCにもPython 3をインストールしドンキーカーに搭載されたサーボドライバーやカメラの設定プログラムを変更し図3に示す環境すべてで認識させるのである。この作業もSSHを使用する為SSH環境が重要となり、これらの設定を行うにはOSがWindowsよりもLinuxが最適である。Linuxには数種類のOSがあり実績があるUbuntu 21.10をインストールした。ほとんどの設定作業はUbuntuの端末と呼ばれる機能を使用しコマンドを入力し行い、モーターの直進、停止、後退やステアリング左右の切れ割合調整など走行に影響する部分を調整できる。図4にステアリング調整を示す。

図4は、Ubuntu 端末画面を示し、表示されているプログラムはドンキーカー内のプログラムである。SSHにて直接プログラムを変更することができ中央付近の文字色が変わっている部分がステアリング調整となり入力数値を変更し切れ角を調整する。

プログラムを書き換えるにはエディータを使用し、図4はエディータも同時に示し、図、下段の四角で表示されたコマンドがエディータ起動を表し、図3に示すAnaconda内のプログラムが行う。Anacondaについては他紙に譲る。

自動運転に先立ち学習走行を行う、操作としては、走行時にカメラより映し出された画像をデータとしてGoogleより提供されているColaboratoryとTensor Flowに送り、ドンキーカーに

```

GNU nano 3.2 myconfig.py
#DRIVETRAIN
#These options specify which chasis and motor setup you are using. Most are using SERVO ESC.
#DC_STEER_THROTTLE uses HBridge pwm to control one steering dc motor, and one drive wheel motor
#DC_TWO_WHEEL uses HBridge pwm to control two drive motors, one on the left, and one on the right.
#SERVO_HBRIDGE_PWM use ServoBlaster to output pwm control from the PiZero directly to control steering, and HBridge for a drive $
#PIGPIO_PWM uses Raspberrys internal PWM
#DRIVE_TRAIN_TYPE = "SERVO_ESC" #SERVO_ESC|DC_STEER_THROTTLE|DC_TWO_WHEEL|SERVO_HBRIDGE_PWM|PIGPIO_PWM|MN1|MOCK
#
#STEERING
#STEERING_CHANNEL = 1           #channel on the 9685 pwm board 0-15
#STEERING_LEFT_PWM = 460       #pwm value for full left steering
#STEERING_RIGHT_PWM = 290      #pwm value for full right steering
#
#STEERING FOR PIGPIO PWM
#STEERING_PWM_PIN = 13         #pin numbering according to Broadcom numbers
#STEERING_PWM_FREQ = 50        #frequency for PWM
#STEERING_PWM_INVERTED = False #If PWM needs to be inverted
#
#THROTTLE
#THROTTLE_CHANNEL = 0          #channel on the 9685 pwm board 0-15
#THROTTLE_FORWARD_PWM = 450    #pwm value for max forward throttle
#THROTTLE_STOPPED_PWM = 370    #pwm value for no movement
#THROTTLE_REVERSE_PWM = 290    #pwm value for max reverse throttle
#
#THROTTLE FOR PIGPIO PWM
#THROTTLE_PWM_PIN = 18         #pin numbering according to Broadcom numbers
#THROTTLE_PWM_FREQ = 50        #frequency for PWM
#THROTTLE_PWM_INVERTED = False #If PWM needs to be inverted
#
#DC_STEER_THROTTLE with one motor as steering, one as drive

```

図4 ステアリング調整

どこを走行すれば良いか学習させるためのデータを作成させる。Colaboratory は教育、研究用にインストール不要で使用できる機械学習、深層学習の環境を整えることができ、強力な GPU を持たない PC などでは有用なサービスである、Tensor Flow は写真や音声認識などに活用できるソフトウェアライブラリーであり、この2つのソフトを使用しディープラーニングを行い、データをドンキーカーに返し、自動運転ができる状態となるのである。

5-3 動作確認

各モジュールを配線し、プログラム設定を行い、動作確認を行ったが NAC ドンキーカーは、すぐには起動しなかった。トラブル内容について表 3 に示す。

表 3 トラブル内容

	トラブル	内容	対策
1	モーターが動かない	モータードライバーモジュールの電流不足	L298N から Hi litandgok 5c7bxui にモジュール変更
2	ドンキープログラム作動しない	カメラモジュール不適合	Saim Smart 1 mx219から Raspberry Pi cameraV2.1 に変更
3	後退しない	キャリブレーション不良	解決に至らず
4	プログラムエラー	ゲームコントローラー有効化に伴うプログラム誤記載	config を入れ替える

1のトラブルは初期に取り付けたモジュールではモーター端子には電圧は発生しているが回転させるための電流に対応していなかった為であるが、プログラムからの原因と混同し解決に時間を要した。

2のトラブルはドンキーカープログラム起動試験時にて発生し、プログラム要件は満たしていたが取り付けられたカメラモジュールが現プログラムに適合していないことが判明した。交換後、無事プログラムが起動したが、このトラブルではどちらのカメラも取り付け可能であり、外観としてはレンズのサイズが違う程度なのである。しかしモジュールとプログラムの適合が必要であることを学ぶ事となった。図5にカメラモジュールを示す。

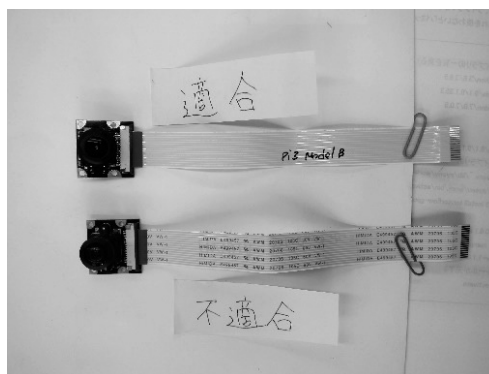


図5 カメラモジュール

3のトラブルはモーターコントロール適正化不良である。停止、前進、後退の極限值測定後プログラム出力数値を入力し記憶させることによりドンキーカーを制御する、プログラムから出力されるデジタル数値をモータードライバーにより PWM 制御されモーターに電気が伝わるのであるが、使用するモータードライバーの問題かプログラム制御の問題なのか現段階では解決に至

らないが前進、停止は可能のため研究を進めることとした。

4のトラブルはゲームコントローラーを用いドンキーカーをコントロールすることを試みたがプログラム変更に対する知識が乏しく config の記載を復旧できなくプログラム不動となる。バックアップファイルよりオリジナルプログラムに変更し、ゲームコントローラーは不使用にすることとした。

こうしたトラブルを乗り越え、完全ではないが走行できる状態にたどり着いた。

5-4 学習走行

自動運転を行うための学習走行を行った、学習走行によるディープラーニングの関係を図6に示す。

学習走行を行うとドンキーカー内にカメラからの画像とその時点でのステアリング角度とモーター出力が記憶され、それらをディープラーニングすることによりデータの最適化がされるのである。最適化されたデータをもとにカメラより入力される景色を判断しドンキーカー内のプログラムで出力を制御している。

Donkey Car のホームページには最低10000枚の画像データが必要と記されており、図7にNAC ドンキーカーの画像データを示す。画像が小さくわかりにくいパラパラ漫画のように走行状況が変化していることがわかる。この画像1枚1枚にステアリングやモーター出力データが対になり、すべてのデータを使用しディープラーニングをするのである。

本研究ではこのディープラーニングの完成を見るまでに至らなかった、原因として考えられる

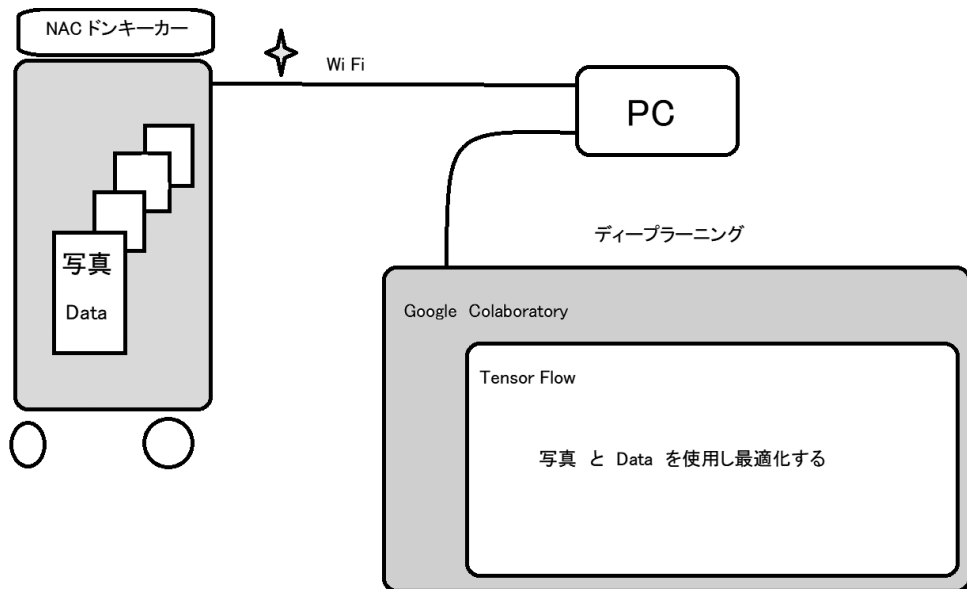


図6 ディープラーニング関係



図7 カメラ画像

項目は、プログラム設定不良、通信時間が長すぎる、学習データ不良などがあげられ今後解決を試みたい。

6 学生制作ドンキーカー比較

本研究とは別に学生の卒業研究として別のドンキーカーを制作した（以下学生ドンキーカー）、NAC ドンキーカーと学生ドンキーカーを図8に示す。図の左に示す車両が学生ドンキーカーであり右がNAC ドンキーカーとボディである。学生ドンキーカーは市販品であるがプログラムを書き換え自動運転可能に改造されている、カメラのモジュール適合でも述べたが学生もプログラムとの適合には多くの時間を要していた。学生ドンキーカーとNAC ドンキーカーの大きな違いとしてディープラーニングがあげられる。学生ドンキーカーではGoogleのTensor Flow を使用せず学生ドンキーカーと通信を行うPC内でディープラーニングを行う。その為にPCにはグラフィックボードを実装し計算速度向上に対応しており、使用するソフトウェアはChainerである。PC本体のグラフィックボードに大きく依存することになるがこの環境は自動運転車両に近く変化し続ける走行状況を車両内で最適化できることは今後の可能性に期待ができる。学生ドンキーカーは自動運転まで到達することができ見学する機会を得た、私感としては「ゆっくりと進むのが精一杯で、それ以上は処理が追い付かない」であった。

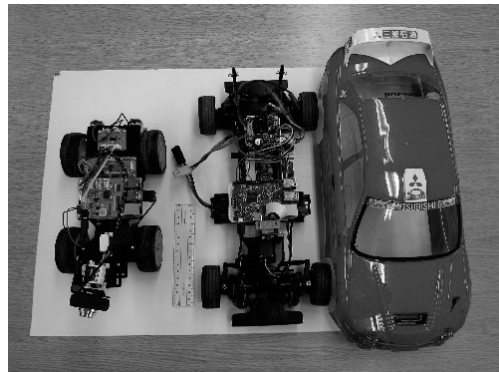


図8 学生・NACドンキーカー比較

7 自動運転では条件付運転自動化レベル

自動運転の研究としてドンキーカーを制作し、学生ドンキーカーによる走行を行うことができたが、この走行をSAEの定める自動運転レベルに置き換えてみたところレベル3の「条件付自動運転」に相応しいことが判明した。ドンキーカーでは入力画像に対する制御を出力する為、初めてのコースではコントロール不能になり学習済みのコースの中のみ自動運転が可能である、また、マイクロコンピューターボードの性能によるが高速処理が追い付かず、それに合わせてドンキーカーの速度も決して速くないなど照らし合わせればレベル3となった。

現在の自動運転を見れば開発段階であれレベル4車両も存在している、これを実現するためには多くのセンサーを実装し高速でデータを計算しながら車両を制御していると考えられ技術レベルの高さに改めて驚かされた。さらに上のレベル5ではすべての環境での自動運転と定義され、場所、環境すべてを制御するなど求められる条件は多く、自動運転実現の難しさを本研究を通じ実感し、同時にこれからの自動車が示す姿に期待が膨らむ。

8 ま と め

自動運転車両 ドンキーカー制作にあたり多くの問題に直面することになった。その多くはプログラム制御に関する問題であり今後実現するだろう自動運転車両にも通ずるところがあると考ええるモジュールの不適合では認識不足ではあるが型番にあわせプログラムを変更するなどの対応が必要であるなど多くの課題を残す結果となった。

しかし、今後の自動車には、こうした新技術が取り入れられる事になり一線で整備にあたる技術者にはさらなる知識が求められるであろう。この研究を止めることなく前に進め学生に還元できることを願う。

9 謝 辞

今回の研究開始時に、Linuxなど各ソフトウェアを理解している上でのドンキーカー製作であったが、私のソフトウェアへの知識が浅く度々困難に直面したが快く指導をしてくださった青木恒夫教授に感謝を述べたい。

同時期に別のアプローチにて学生ドンキーカーを制作してくれた学生諸君に謝意を述べる。現在彼らの研究が先行している状態であるが、常に刺激を受けながら研究を進めることができた。今回かなわなかったが同時走行を目指し共に歩みたいと願う。

参 考 文 献

