

# 創作ゼミナール 報告書

平成 20 年 1 月 16 日 作成

学籍番号：ソ 1 7 0 1 8

氏名：草階 司（矢萩研究室）

テーマ名：「自立型走行ロボットの走行用プログラム作成」

## 1. はじめに

創作ゼミナールにおける研究テーマは『自立型走行ロボットの走行用プログラム作成』である。これまで製作されてきた H8(3664)マイコンを搭載した自立型走行ロボットが、加速 - 定速 - 減速を主体とした基本走行ができるような走行用プログラムを開発することを目的として研究を進めることにした。

この研究テーマの目標は、ロボットの動作する仕組み、要するにプログラムや回路図などを頭に入れつつ、さまざまな動きをシミュレートしながらロボットの構造やそれに関する知識を学んでいき、その経験を生かし走行用プログラムを開発することである。

図 1 が今回使用した自立型走行ロボットの写真である。

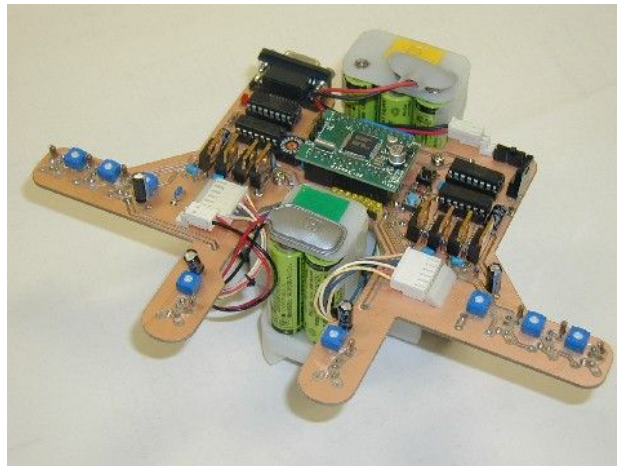


図 1 H8(3664)マイコンを搭載した自立型走行ロボット

## 2. 研究内容

今回使用しているロボットに搭載されている H8(3664)マイコンは 32kbyte のメモリを積んでいる。従来、このロボットが加速-定速-減速から成るスムーズな走行を行うために、sin 関数を元にした走行用プログラムを乗せていた。図 2 に示したモータ駆動回路のモータ駆動 IC(PMM8713)に図 3 のようなパルスを入力することにより、スムーズなモータの回転が行われる。しかし、これまでの方法では全体のメモリの容量に対して、高い容量を用するという欠点があった。もし、このままこのプログラムを使用すると H8(3664)マイコンのメモリ容量で使える 32kbyte の部分のかなりの部分を、加速-定速-減速の走行用プログラムが占拠してしまう。これでは、後々センサ用プログラムなどの部分などを追加するとき、容量不足でメモリ内に入りきらな

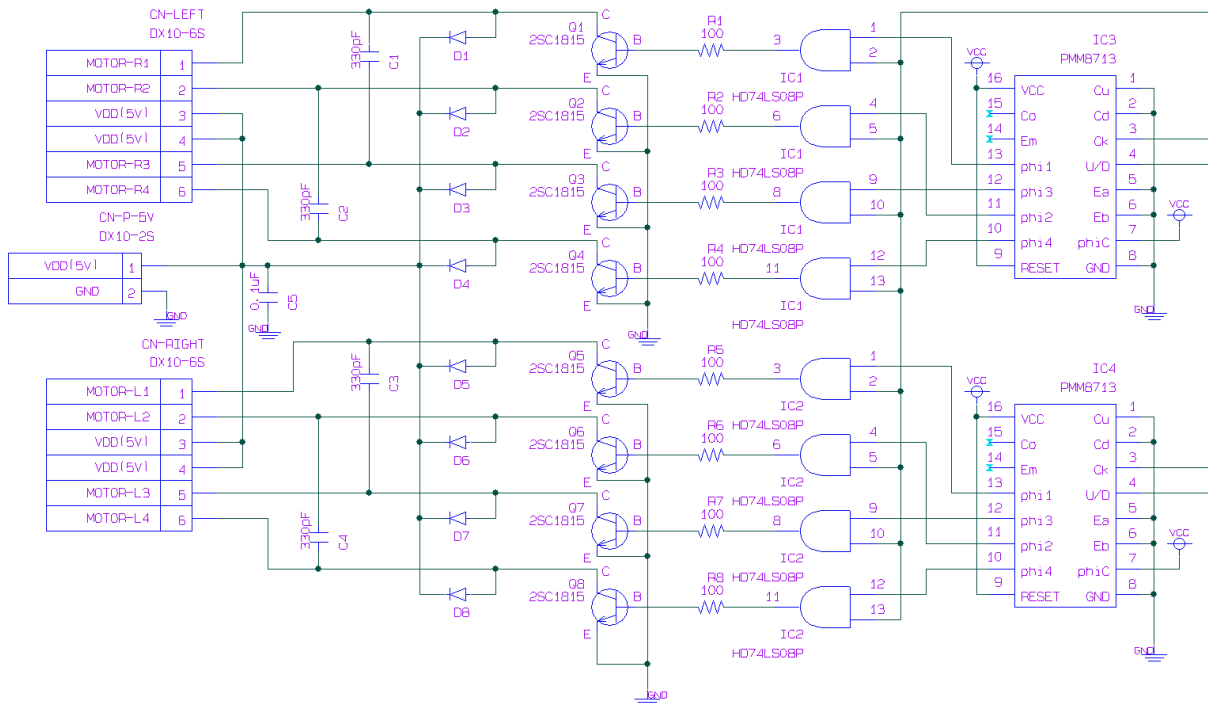


図2 モータ駆動回路

くなる恐れが出てくるので、どうにかしてその部分のメモリ消費量を軽量化する必要があった。

さて、この問題を解決するため、プログラム内の sin 関数を多項式として近似することができないか検討の必要がある。プログラム内で使用される計算方法を変更することによってメモリの軽量化を図ろうという狙いである。

今回、その問題のプログラムを改良するに当たって Igor(Graphing and Data Analysis from WaveMetrics)というデータ解析用ソフトを使用した。このソフトは、次式

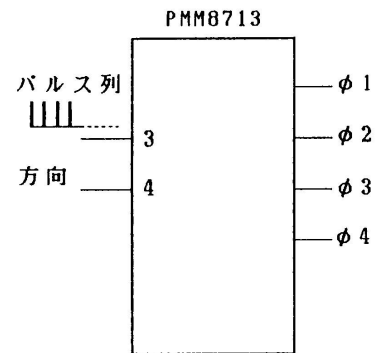


図3 モータ駆動 IC

$$W(i) = \frac{1}{V_{\min} + (V_{\max} - V_{\min}) \times \frac{1}{2} \left\{ 1 + \sin \pi \left( \frac{i}{AccPulse} - 0.5 \right) \right\}}$$

$$V_{\max} = 1.0, V_{\min} = 0.2, AccPulse = 70$$

$V_{\max}$  : 最大速度     $V_{\min}$  : 最小速度     $AccPulse$  : 加速に必要なパルス数

の sin カーブの曲線を 70 等分にする。その 70 個の点から計算して、その点のできるだけ近くを

通過するような実線を描く多項式を算出してくれる。

このソフトでは、算出するときはこちら側でその線を再現するときを使う項数を指定でき、指定する項数が多い式の方が元の線に限りなく近い滑らかな線を導き出せる。今回は 3 項から次第に条件を広げていって多項式の項が 6 項になったところで、ほぼ完全に元の sin カーブに近い実線が描けたので 6 項の多項式を使用した。

次式が

$$W(i) = 4.9888 + 1.6084 \times 10^{-2}i - 1.5563 \times 10^{-2}i^2 + 7.0595 \times 10^{-4}i^3$$

$$- 1.4344 \times 10^{-5}i^4 + 1.4189 \times 10^{-7}i^5 - 5.5325 \times 10^{-10}i^6$$

ヒッティングして得られた近似曲線の多項式である。

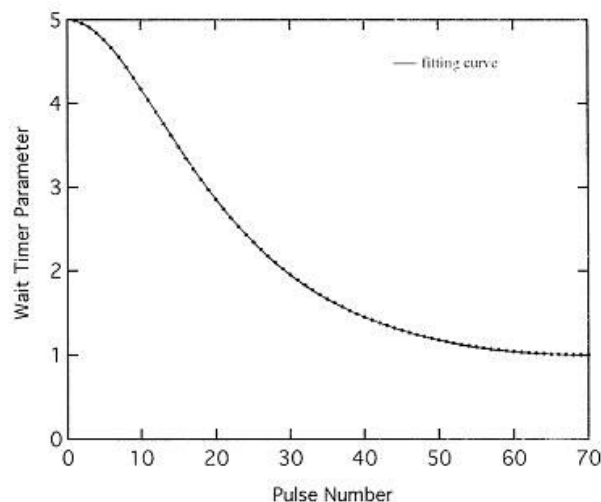


図4 ヒッティング曲線

図4が元のプログラム内にあった sin 関数をもとにして得られる曲線 (70 等分されて得られた点線部分) である。実線部分がヒッティング(近似)後の 6 項の多項式にあたる部分である。このように別の式に置き換えた後でも、グラフに刻まれる軌跡は近似前、近似後と殆ど変わらないことがわかり、計算方法の変更がうまく働いていることがわかる。

さて、このように計算方法を変更し、計算方法を置き換えることによりデジタルコンピュータが苦手とする計算方法、従来の sin 関数を元に加速-減速を行う走行用プログラムという円周率等で発生する複雑な積分を回避することに成功している。これにより大幅にメモリ消費容量を削減しつつ、人間の目では違いがわからないほど同じ動きをするプログラムができあがった。

外見的には違いはでないが、改良したプログラムの方はメモリ消費量を 3 分の 1 にまで減らしても、従来の sin 関数をもとに加速、減速を行う走行用プログラムとほぼ同じ動きを再現できた。

### 3.まとめ

今回はプログラム部分に集中して取り組んだが、まだまだ手付かずの部分も多く、特に本体部分にはまったく手を加える余裕が無かった。自立型走行ロボットの完成度をさらに高めるにはプログラムの改良だけでなく、本体部分の改良が必要だと思われるので、今回の作業で空いたメモリ部分なども使用して、更なる改良、強化を施していきたいと思う。

### 参考文献

瀬戸芳明、平成 17 年度卒業研究論文「自立型走行ロボットの設計・製作」(青森大学工学部電子システム工学科)