

企 業 名 : 株式会社 アドテックス

研究代表者 : システム生体工学科
准教授 松本 浩樹

研究テーマ : 「自己学習機能を持つ、予測制御を用いた小型モータを含めた汎用自動制御装置の開発」

自己学習機能をもつ、予測制御を用いた
小型モータを含めた汎用自動制御装置の開発

株式会社アドテックス

武田 徹

千川 達也

前橋工科大学工学部システム生体工学科

松本 浩樹

(1) 開発の概要

自己学習機能をもつ汎用自動制御装置「インテリジェントインバータ」の研究開発について、平成 27 年度では、低速で比較的制御しやすい大型モータ（3,600 回転/分）での動作を確認している。本年度は、この開発成果を拡張し、高速で制御の難しい小型モータ（25,000 回転/分）での動作を実現する。

(2) 開発内容

本開発で扱うインテリジェントインバータは、コンピュータによる自己調整機能をもつ適応制御を用いることで、適切な制御パラメータへと自動的に調整する汎用インバータである。

PID 制御は比例(Proportional)、積分(Integral)、微分(Differential)の 3 つの制御を組み合わせた制御法で、構造が単純で優れた収束の早さを実現できることから、現在のモータ制御において幅広く利用されている。しかしながら、P・I・D の 3 つのパラメータが互いに干渉するため厳密な調整が難しく、稼働環境の変化や製品誤差によって異なる適切なパラメータに自動的に対応できないという難点がある。

一方で、適応制御はアルゴリズムを組むことにより、現在の状態から要求された状態になるまでパラメータを繰り返し少しずつ増減させるという、単調な微調整をコンピュータに自動で処理させる制御法である。適切なパラメータになるまで自動的に調整するため、モータの特性に依存しない汎用性の高さがある。しかしながら、モータ制御における適応制御の実績がないため、アルゴリズムの確立やモータ制御における課題点の調査が必要となる。

そこで本開発では、様々なモータに対して適応制御を実施し、そのアルゴリズムの検討や汎用性の確認を行う。本課題を達成する手順を以下に示す。以降、この手順に倣い順次解決していく：

- ① シミュレーションで不可欠となる，モータの同定
- ② 比較用としてPID制御を数式化し，動作をシミュレーションで確認
- ③ 適応制御のアルゴリズムを作成，数式化し，動作をシミュレーションで確認
- ④ それぞれの制御を実機で試験し，それらの動作を確認

(3) シミュレーション

(3-1) モータの同定

シミュレーションを行うためには，加えた電圧に対してモータがどのような挙動をするかを明らかにするため，モータの同定が必要となる．今回使用する送風ファン付き小型モータは公表できないため，明示されている仕様を Table 1 に示す：

Table 1: Motor specification

項目	仕様	備考
モータの種類	センサレス DC モータ	駆動回路分離タイプ、三相モータ
軸受	単列深溝軸受	$\phi 8 \times \phi 4 \times 5$
回転方向	時計回り	
公称電圧	DC12V	
最大電流	2.5A 以下	
回転数範囲	500~25,000rpm (参考値)	

Table 1 で示されるモータに対し，1.0[V]の電圧をかけたときの回転数，いわゆるステップ応答からモータの同定を行う．ここでは，ステップ応答を 10 回取得し，時刻ごとに平均化した．その結果を Fig. 2 に示す：

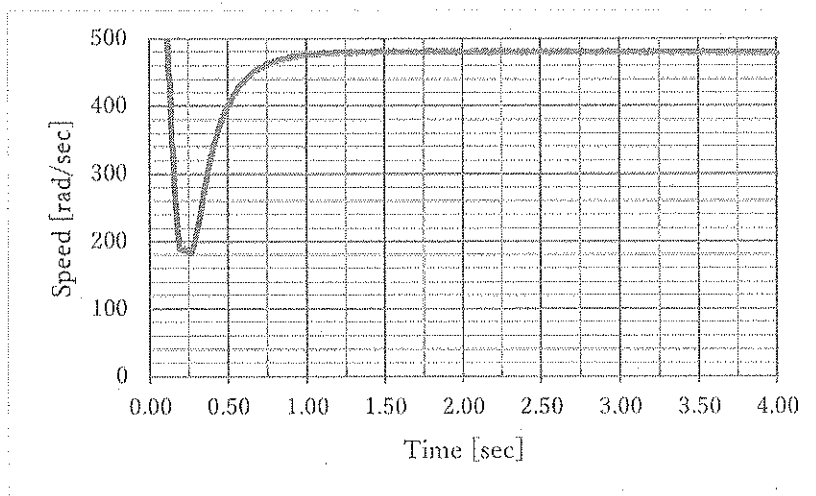


Fig. 2: Step response of motor

ここで、縦軸は rpm (1 分あたりの回転数) ではなく rad/sec (1 秒あたりの回転角、 $360^\circ = 2\pi \text{ rad}$) である。モータの駆動直後は正確な回転数が推定できないため、同定にはモータの回転数が徐々に上がる約 0.25 [sec] 以降を利用する。この波形から、このモータは 1 次遅れ系に近似が可能であり、定常値 479.3897 [rad/sec], 時定数 0.16 [sec] となる。以降、この 2 つの値を用いてシミュレーションを行う。

(3-2) PID 制御

ここでは、PID 制御のシミュレーションについてまとめる。その目的は、①PID 制御を数式化すること、②PID ゲインを決定すること、③モータを目標回転数 25,000 [rpm], すなわち目標回転速度約 2617.994 [rad/sec] で回転させることの 3 つである。

構成する制御系を Fig. 3 に示す：

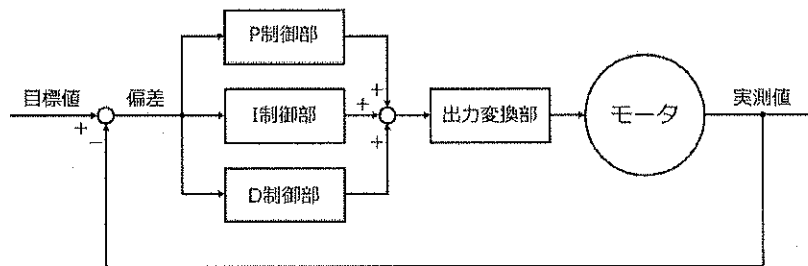


Fig. 3: PID control system

Fig. 3 の制御系のうち、モータは(3-1)で同定したモータである。目的①を達成するために必要な数式化は、P、I、D 制御部と出力変換部が該当する。そこで、Python 言語を用いた計算ツール「spyder」を使用し、Fig. 4 のように PID 制御のプログラムを作成した。Fig. 4 により、①の目的を達成した。

つづいて目的②について検討する。PID 制御ゲインは、(3-1)で得たモータのステップ応答を基に、PID ゲイン調整法の一つであるジエグラ・ニコルスのステップ応答法を参考にした結果、 $K_p = 0.0019$, $K_i = 0.00002$, $K_d = 0.0002$ を得る。以上により、目的②を達成した。

最後に目的③を解決する。上記で得た PID ゲインを用いて Fig. 4 のプログラムを実行すると、Fig. 5 が得られる。ここで、破線は目標回転速度、実線はモータの回転速度を表している。Fig. 5 から、Fig. 4 のプログラムを用いた PID 制御による速度制御が達成できていることがわかる。

以上により、③の目的も達成でき、PID 制御のシミュレーションにおけるすべての目的を達成した。

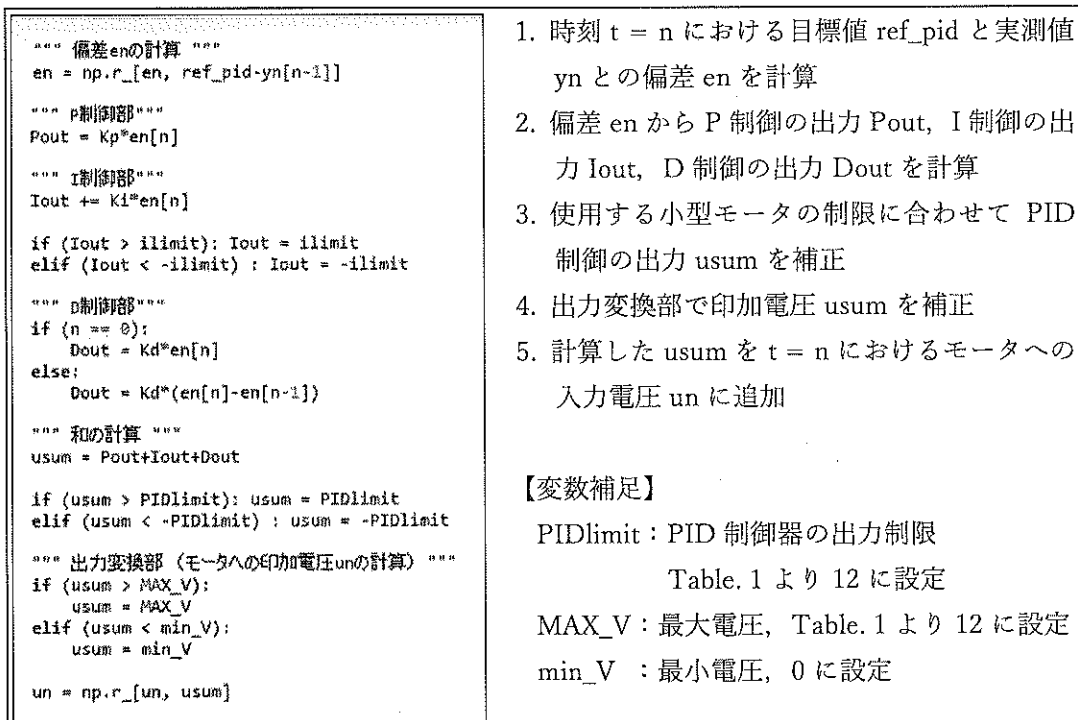


Fig. 4: PID control on spyder

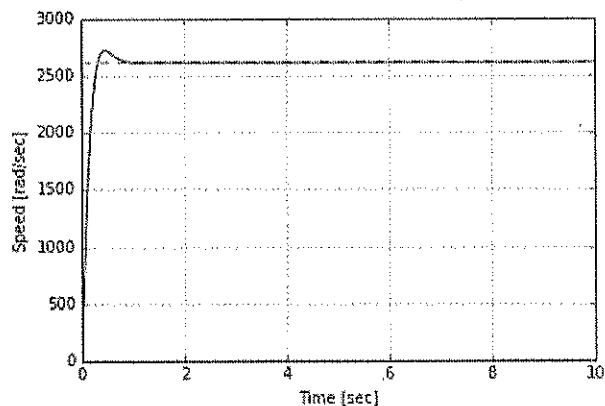


Fig. 5: Response of PID control

(3-3) 適応制御

ここでは、適応制御のシミュレーションについてまとめる。その目的は、①モータ制御における適応制御部のアルゴリズムを組むこと、②アルゴリズムを数式化すること、③ステップサイズを決定すること、④モータを目標回転速度約 2617.994 [rad/sec]で回転させることの4つである。

構成する制御系を Fig. 6 に示す：

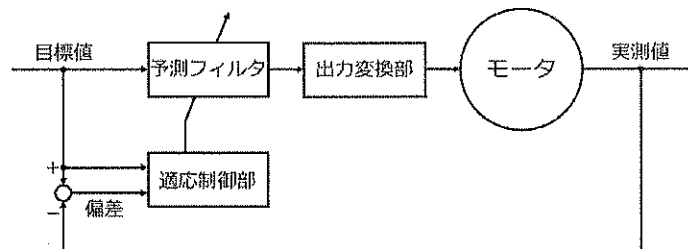


Fig. 6: Adaptive control system

Fig. 6 の制御系のうち、モータは(3 - 1)で同定したものを、出力変換部は(3 - 2)で設計したものを用いる。まず、目的①のアルゴリズムとして、本開発では LMS(Least Mean Square：最小二乗平均) アルゴリズムを採用する。LMS アルゴリズムは、任意の評価関数の二乗平均が最小になるように関数内の値を反復的に修正していくアルゴリズムである。本開発ではこの評価関数を Fig. 6 の偏差とし、偏差が小さくなるように予測フィルタのフィルタ係数を次式に基づいて反復的に修正する；

$$h(n+1) = h(n) + \mu e(n)r(n) \quad (1)$$

ここで、 $h(n)$ は予測フィルタのフィルタ係数、 μ はフィルタ係数の修正幅を調整するステップサイズ、 $e(n)$ は偏差、 $r(n)$ は目標値である。以上で、①の目的を達成した。

つぎに、目的②について検討する。上記のアルゴリズムを計算ツール「spyder」に記述した結果を Fig. 7 に示す：

<pre> """ 偏差enの計算 """ en = np.r_[en, ref_pid-yn[n]] """ 更新則に従い予測フィルタの係数を更新 """ for fi in range(fltsize): LMSflt[fi] += LMS_step*en[n]*ref_pid """ 予測フィルタの出力を計算 """ usum = 0 for fi in range(fltsize): usum += LMSflt[fi]*ref_pid """ 出力変換部 (モータへの印加電圧unの計算) """ if (usum > MAX_V): usum = MAX_V elif (usum < min_V): usum = min_V un = np.r_[un, usum] </pre>	<ol style="list-style-type: none"> 1. 時刻 $t = n$ における目標値 ref_pid と実測値 yn との偏差 en を計算 2. ステップサイズ LMS_step、偏差 en、目標値 ref_pid で予測フィルタ $LMSflt$ を修正 3. 予測フィルタ $LMSflt$ の係数と目標値 ref_pid から予測フィルタの出力を計算 4. 出力変換部で印加電圧 $usum$ を補正 5. 計算した $usum$ を $t = n$ におけるモータへの入力電圧 un に追加
---	---

Fig. 7: Adaptive control on spyder

つづいて、目的③について検討する。(1)式から、ステップサイズにかかる変数は偏差 $e(n)$ と目標回転速度 $r(n)$ である。今回は目標回転速度が大きいため、フィルタを少しずつ調整するために $\mu = 5.0 \times 10^{-15}$ と与える。

最後に目的④を解決する、上記のステップサイズを Fig. 7 のプログラムに適用して実行すると、Fig. 8 が得られる。ここで、破線は目標回転速度、実線はモータの回転速度を表している。Fig. 8 から、数式化したアルゴリズムによってモータの回転速度が目標回転速度に収束していることがわかる。したがって、目的④も達成でき、適応制御のシミュレーションにおけるすべての目的を達成した。

以上により、PID 制御と適応制御のシミュレーションが完了した。

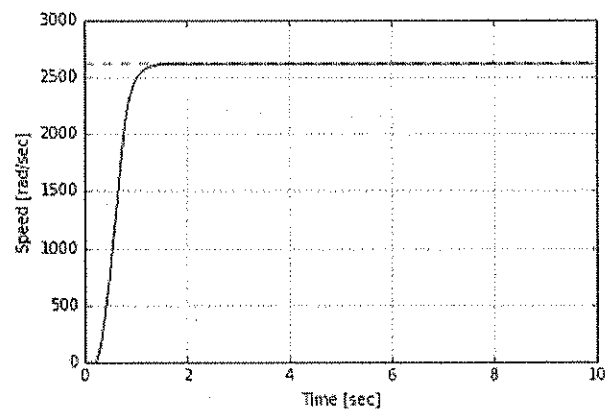


Fig. 8: Response of adaptive control

(4) 実機試験

ここでは、シミュレーションで作成したプログラムとそれに用いたパラメータをもとに、実際にモータを目標回転速度に合わせて動かせるかどうかを確認する。まず、本実機試験を行うための試験環境を簡単に紹介する。つぎに、PID 制御、適応制御の順に試験を行い、モータが目標回転速度に合わせて動くことを確かめる。

(4-1) 試験環境

実機試験には、Fig. 9 に示す株式会社デスクトップラボ様の基板を使用した：

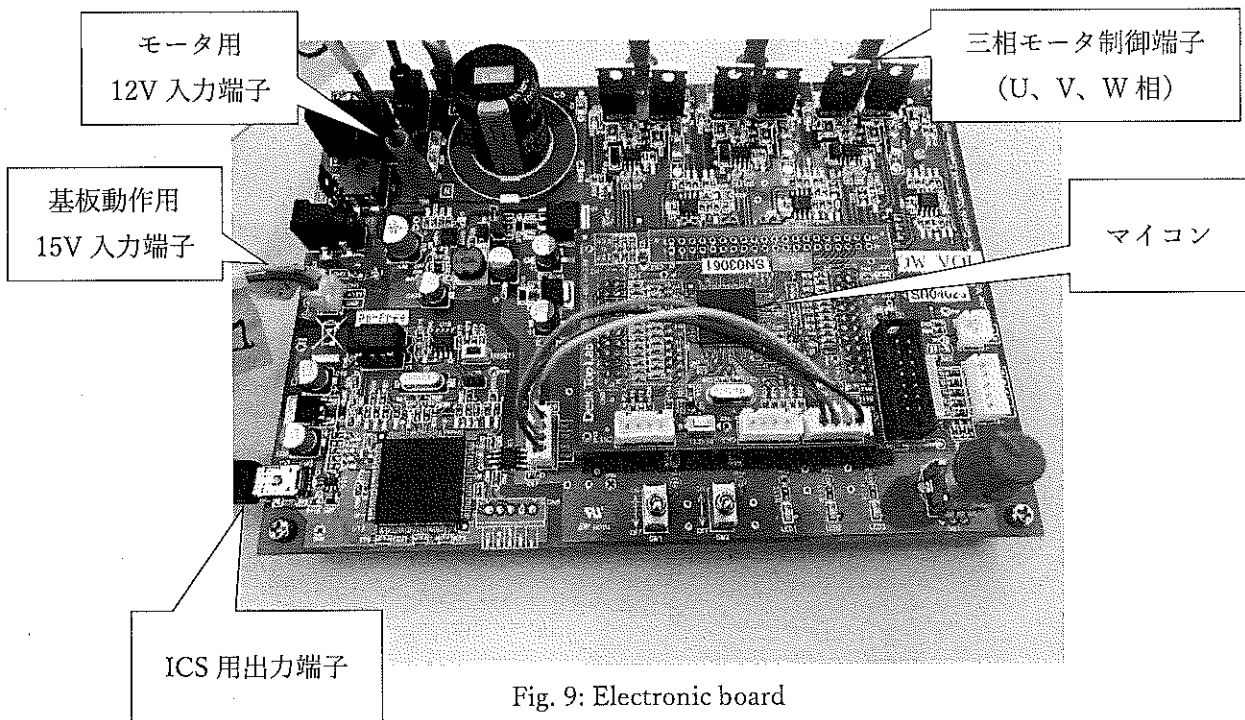


Fig. 9: Electronic board

Fig. 9 の基板に備え付けられたマイコンに、Fig. 4 や Fig. 7 のプログラムを C 言語に変換して書き込む。また、Fig. 4 や Fig. 7 の電圧 u_n を実際の電圧に変換する出力変換部 (Fig. 3, Fig. 6 参照) には PWM 制御を使用する。

また、Fig. 9 の ICS 用出力端子は、ルネサスエレクトロニクス株式会社様の ICS (InCircuit Scope) というツールを使用するための端子である。このツールはマイコン内で使われている変数をリアルタイムで取得でき、Fig. 10 のようにオシロスコープの画像を描くことができる：

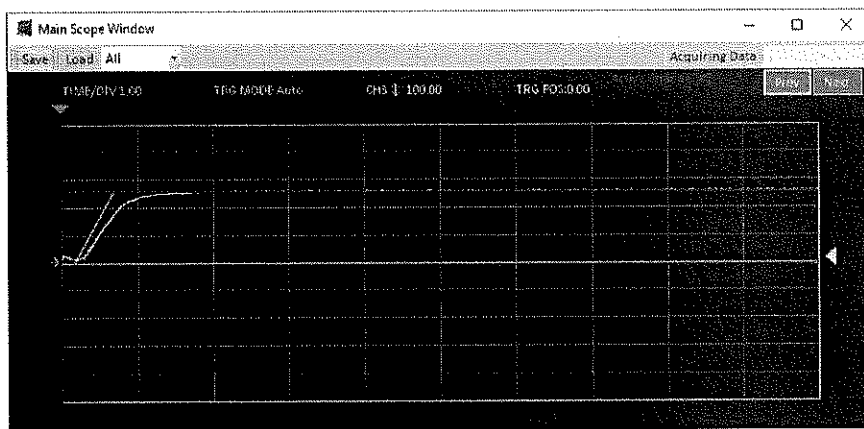


Fig. 10: Example of Main Scope Window of ICS

取得結果は cvs 形式で出力でき、PID 制御と適応制御それぞれで 100 回ずつ計測し、集計を経てそれぞれの制御を評価する。ただし、今回は横軸の時間が最大 10 秒であるため、PID 制御も適応制御も 10 秒以内に目標値に収束させるよう制御する。

また、実機試験の注意点として、モータの初動の失敗と過電流が挙げられる。初動の失敗はモータの永久磁石の初期位置が毎回異なることにより、電磁石の NS 切り替えに永久磁石が追従できない場合に発生する。これを防ぐため、NS 切り替えの頻度が少ない低速回転のオープンループでモータを空回しした後に PID 制御や適応制御による速度制御に移る方針を取る。また、過電流は主にモータを急加速させた場合に生じる。特に起動直後は初速 0 [rad/sec] から目標回転速度 2617.994 [rad/sec] までの大きな変位があるため、加速度が大きくなる傾向にある。これを防止するため、本来の目標速度とは別に 5,000 [rad/sec²] で加速する速度制御用の目標速度を設けることで、加速度が大きくなり過ぎないように調整する。

以上の試験環境のもと、PID 制御と適応制御の実機試験を行う。

(4-2) PID 制御

ここでは、Fig. 4 のプログラムを C 言語化して PID 制御を行った結果を示す：

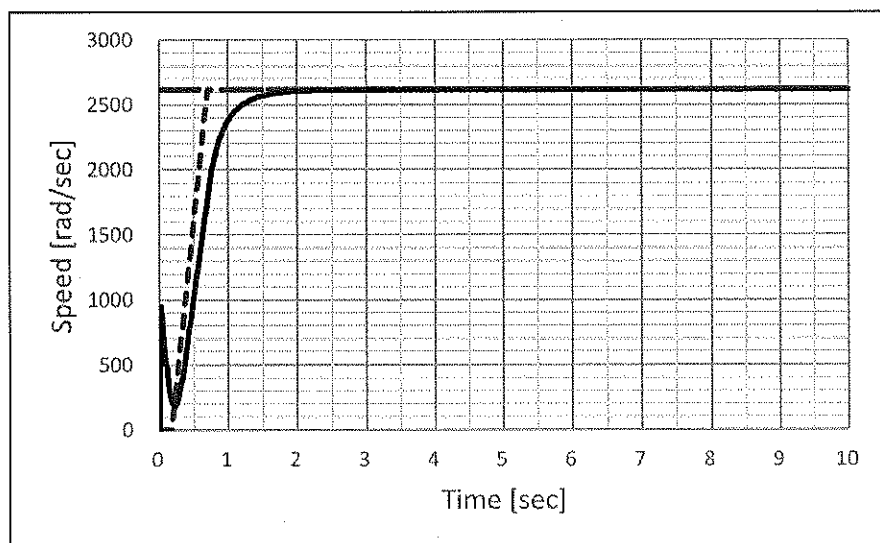


Fig. 11: PID control for the motor

ここで、破線が目標回転速度、点線が(4-1)で述べた制御用の目標回転速度、実線がモータの回転速度を表している。Fig. 11 から、作成した PID 制御のプログラムでモータの回転速度を目標回転速度に追従できていることがわかる。

以上により、本研究開発の目的であるモータの高速回転を PID 制御で達成できた。

(4-3) 適応制御

ここでは、Fig. 7 のプログラムを C 言語化して適応制御を行った結果を示す：

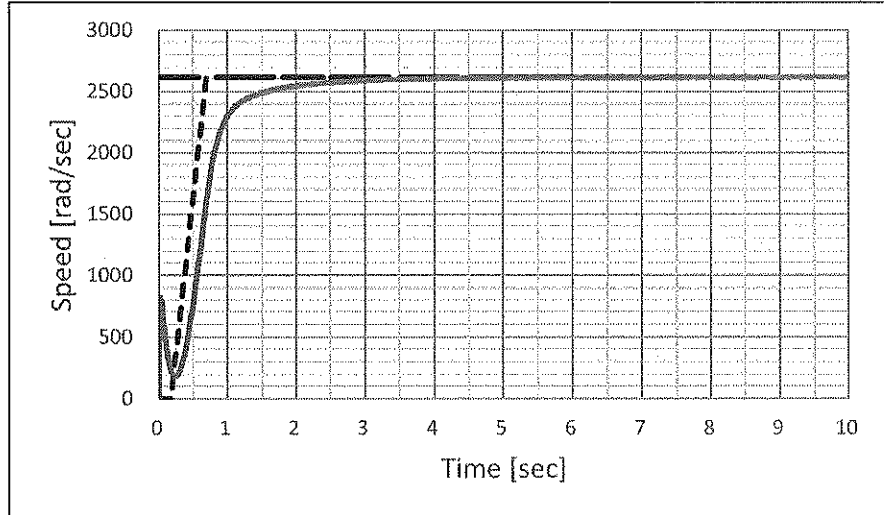


Fig. 12: Adaptive control for the motor

ここで、破線が目標回転速度、点線が制御用の目標回転速度、実線がモータの回転速度を表している。Fig. 12 から、作成した適応制御のプログラムでモータの回転速度を目標回転速度に追従できていることがわかる。

また、それぞれの制御の立ち上がりを比較するため、Fig. 11 と Fig. 12 の 1.0[sec] から 3.0[sec] までを拡大したものを Fig. 13 に示す：

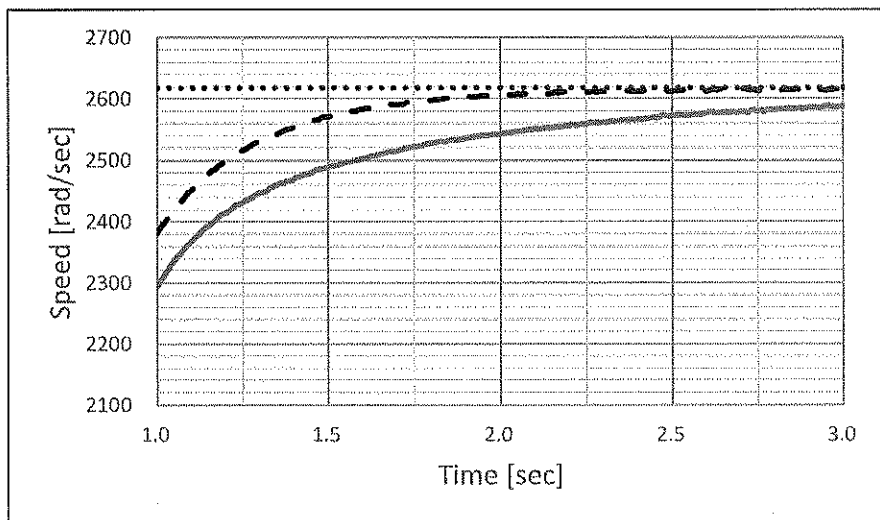


Fig. 13: From 1.0[sec] to 3.0[sec] of Fig. 11 and Fig. 12

ここで、点線が目標回転速度、破線が PID 制御の応答、実線が適応制御の応答を表している。Fig. 13 から、目標値への収束の早さは PID 制御が優れていることがわかる。一方、それぞれの制御の定常状態を比較するため、Fig. 11 と Fig. 12 の 8.0[sec]から 10.0[sec]までの回転速度からそれぞれの不偏標準偏差を計算し、目標速度付近での応答のばらつきを求める。計測数 100 回の不偏標準偏差を計測ごとに計算し、PID 制御と適応制御それぞれで平均した結果：

PID 制御 : 11.331[rad/sec]

適応制御 : 10.559[rad/sec]

となり、0.772[rad/sec]、つまり 1 分間あたり約 7.372 回転の差で適応制御の方がばらつきが小さい結果となった。これはすなわち、一定速度を維持する性能は適応制御の方が優れているといえる。

(5) まとめ

本研究開発では、適応制御による汎用自動制御を目的としたインテリジェントインバータについて、平成 27 年度に行った低速で比較的制御しやすい大型モータ（3,600 回転/分）での技術を拡張し、高速で制御の難しい小型モータ（25,000 回転/分）での動作を実現した。その過程で、シミュレーションと実機試験を行い、理論的に制御可能であること、そして実際に制御可能であることを確認した。また、それぞれの実測結果から、モータ制御における PID 制御と適応制御それぞれの特徴を比較し、立ち上がりの早さは PID 制御が優れ、定常状態では適応制御が優れることを確認した。