

リアルタイム Linux を利用した 振動モータの同期制御

福本 隆浩 *¹ 山田 健仁 *² 百田 正広 *²

Synchronous Control for Vibrating Motors using Real-Time Linux

Takahiro FUKUMOTO, Takehito YAMADA and Masahiro MOMOTA

Abstract

Synchronous control for vibrating motors has been realized using a microcomputer (PC) with a real time operating system. Vector inverters drive the vibrating motors that are the AC motors with centrifugal weights. The PC calculates the velocity reference value, based on the digital PID control theory, for controlling the vector inverters. The vibrating motors consisted of two axes were precisely controlled by the PC based digital control on a RT-Linux which is a real time operating system. Moreover, the PC system was achieved miniaturization and rich functionality using PC-AT compatible small-sized single board microcomputer. The digital control real time processing was achieved in the function of the RT-Linux. The RT-Linux is the extended function of the Linux operating system, so the developed systems can realize a remote control by using the network function in the Linux.

Key Words : Synchronous Control, PI Control, Digital Control, Vibrating Motor, Real-Time Linux

1. まえがき

様々な生産工程において、ホッパー、フィーダ、ふるいなどの振動を利用した生産機器が使われている。振動を発生させる機械装置をその加振方法に分けると、電磁式、遠心重式、カムまたはクランク式、空気式、液圧式の5種類になる。いずれにしても物体に振動変移、あるいは振動力を与えるものであるが、加振方法、振動系の相違によりそれぞれ特徴があり、用途も異なる。本研究の対象である遠心重式振動機（以下振動モータ）は、振動数が高く、間欠運転に適していて、設置や取り扱いが容易で、加振力調整が可能であると言

う特徴を持っている¹⁾。

実験の対象とした振動モータは2軸で構成されているもので、ACモータを2個用いることによって様々な振動を発生させることができる。しかし、2個のモータの各軸間の角度制御、すなわち位相制御がうまくいかないと所望の振動モードを得ることはできない。

これまで本研究室では、PC-AT互換機（パーソナルコンピュータ）やSHワンボードマイコン（日立製SHマイコン使用）による2軸モータの同期制御をこれらのコンピュータシステムで実現した²⁾。しかし、PC-AT互換機はこのような制御システムには大形で機能が過剰であり、SHワンボードマイコンによる

*¹ 情報電子工学専攻

*² 情報電子工学科

制御システムは小形ではあるが通信機能などの汎用性に欠ける、と言う欠点があった。

最近では、汎用のコンピュータシステムにフリーソフトのリアルタイムOSを利用した高精度なコンピュータによる機器制御が実用的になってきた³⁾。リアルタイムOSとは、処理をリアルタイムに実現することを重視し、その為の機能を実装したOSのことである。必要な処理時間の予測を行う機能を備え、複数の処理要求が同時に発生した場合でも目的の時間内に完了させることができる。この機能を備えたものに Real time Linux (以下 RT-Linux) がある。フリーのOSとして多く使用されている従来の Linux では、カーネルのスケジューラ機構の調整によるバックグラウンド処理で、擬似的なリアルタイム処理を実現しているが、RT-Linux では完全なリアルタイム処理が実現できる。

本研究は、2軸モータの同期制御を PC-AT 互換のシングルボードマイコン (以下 PC) と RT-Linux を利用して実現することにより、振動モータのより高精度な制御の実現とコンピュータ制御におけるネットワーク利用などの汎用性を高めることを目的とする。

2. 振動モータの機構部の構成

遠心重式の振動モータでは、モータの軸部にアンバランス・ウェイトを取り付ける。このモータを2個用いることにより、アンバランス・ウェイトの位置関係や回転方向によって任意の振動を作り出すことができる。図1に機構部の構成概念図を示す。また、図2にモータの回転速度や各モータの回転方向、各軸に取り付けられたアンバランス・ウェイトの位置関係によって任意の振動が発生する状態を示す。

同期運転にはPI制御を用いる。比例要素 (P要素) は入力が増加すると出力がすぐに対応量だけ増加する要素であり、積分要素 (I要素) は、偏差 (動作信号) のある間、過去の偏差の積分値によって修正動作を行う要素である。従って、PI制御は、基本的にはP要素によって、制御系の安定度や速応性を補償し、これにI要素を付加することによって、外乱に対するオフセット (定常位置偏差) を除こうとするものである。しかし、I要素を強くすると安定度が悪くなり、これを補償するためにP要素を弱めると速応性が悪くなるというトレードオフの関係がある。

また、一定の回転速度になるようにモータを回転させたいが、外乱の影響によって回転速度が変化してしまう。そこで、速度フィードバック制御をかけることで、モータの安定状態を保つようにする。

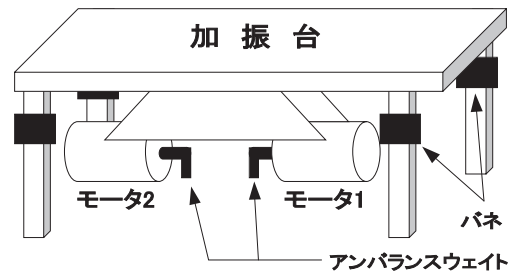


図1 振動モータによる加振器の構成概念図

逆方向回転			同方向回転		
アンバランスウェイトの位置	振動モード	方向性	アンバランスウェイトの位置	振動モード	方向性
		上下			最大
		右上			中間
		水平			ゼロ
		左上			中間

図2 アンバランス・ウェイトによる振動モード

3. 同期制御システムの構成

本研究で用いるコンピュータの仕様を表1に示す。また、同期運転のシステム構成を図3に示す。モータで振動を作るには、モータ軸部にアンバランス・ウェイトを取り付けて回転させればよいが、軸の角度を制御しないと所望の振動は得られない。そこで、モータの位相制御が必要となる。ベクトルインバータはモータの軸に取り付けられたエンコーダからの回転パルスを基に速度フィードバックをかけて制御系の安定化をはかっている。このループをマイナーループとし、メインループはPCのデジタル制御で実現する。メインループはエンコーダからの回転パルスをインクリメンタルエンコーダで受けて、サンプリング時間毎にその値を読み取り、モータの軸角度 (位置) の検出を行う。そして目標値 (位置指令値) と軸角度の偏差を求め、偏差から目標速度を演算してD/A変換器でアナログ信号にした後、インバータに速度指令値として与える。プログラムの大まかな流れを図4に示す。

表1 制御用コンピュータの仕様

シングルボード コンピュータ	PCM-5820 (Advantech 社製)
OS	Silicon-Linux (RT-Linux) イーサネット接続
コンパクトフラッシュ メモリ	32MB
ハードディスク	サイズ: 2.5インチ 容量: 2GB

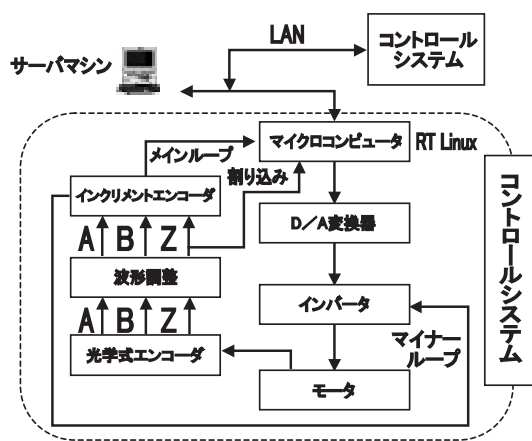


図3 2軸モータ制御システムの構成図

4. 制御系の構成

目標値が変化しない制御系であればP制御（比例制御）のみでよい。しかし、ステップ関数やランプ関数に追従させようとすると定常偏差が生じる。これを改善する為にI制御（積分制御）が必要である。本実験の制御システムはモータが回転している状態で2軸の位相を合わせるため、制御目標がランプ関数で変化していると考えられる。従って、P制御とI制御を合わせたPI制御が必要であり、内部モデル原理よりI要素は2重積分型を必要とする。

2軸同期のPI制御のブロック図を図5に示す。PI制御器はデジタル制御器で構成する。また、モータ1, 2の軸角度を同じ目標値に追従させる。PI制御器をデジタル制御器で構成する手法としては、連続系で制御器を設計し、特性をボード線図等で確認した後、離散系に変換する方法を使った。実験で用いた2次の積分器を持つPI制御器を式(1)に示す。

$$H(s) = \frac{100s^2 + 20s + 1}{10s^2} \quad (1)$$

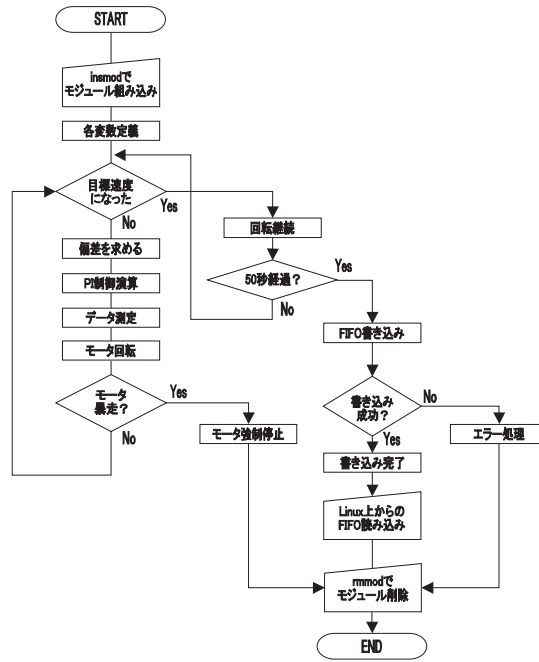


図4 モータ制御全体のフローチャート

これは、これまでの研究において求めた2重積分型のPI制御器である²⁾。

PI制御器をプログラムで実装するために離散化する。数値計算ソフトMATLAB (MathWorks社製) のbilinear関数(双一次S-Z変換関数)により、連続系から離散系に変換して離散化した係数を求める。これによりZ領域の伝達関数を求めると式(2)となる。

$$H(z) = \frac{10.00200z^2 + 20.000z + 9.9980}{1.000z^2 - 2.000z + 1.000} \quad (2)$$

(但し、サンプリング周波数: 1 kHz)

これをプログラムに組み込むには、式(2)を時間領域の演算式に変換する必要がある。

式(3)はZ領域での一般的な伝達関数である。なお、a, bは係数である。

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N a_k z^{-k}}{1 + \sum_{k=1}^N b_k z^{-k}} \quad (3)$$

Z^{-1} は単位遅延の演算子であるので、式(3)より入出力間の時間領域表現は、次のような差分方程式になる。

$$y(nT) = \sum_{k=0}^N a_k x(nT - kT) - \sum_{k=1}^N b_k y(nT - kT) \quad (4)$$

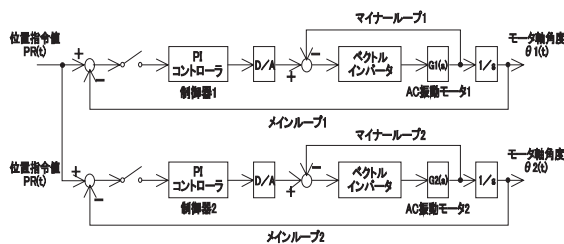


図5 2軸同期のPI制御ブロック図

従って、プログラムに組み込む式は、式(2)より

$$y(nT) = 10.0100x(nT) - 20.0000x(n-1)T + 9.9900x(n-2)T + 2.0000y(n-1)T - 1.0000y(n-2)T \quad (5)$$

となる。但し、 x はその時々を入力値で y は出力値である。今回の実験ではパルス数で算出された偏差(位相差)をラジアンに変換し(パルス数 $\times 2\pi \div 1000$:エンコーダ1回転で1000パルス)、これを入力値 x とした。デジタルPI制御器で計算された値は1秒当りの回転角度がラジアンで出てくるが、次回の計算でこの値を出力 y として用いなければならない。

D/A変換器への出力は、速度指令値として与える必要がある。正転で考えるとD/A変換値2048~4095の2048段階の間にモータの回転数0~60 rps (60 rps = 3600 rpm)を比例させて割り振るのだが、60 rpsを $2\pi \times 60 = 120\pi$ [rad]とし、これをD/A変換器の2048段階に近似させて 1π ラジアン当り17をD/Aに出力することにした。

5. RT-Linuxによる実装

RT-Linuxは、Linuxを使用して計測や制御向けのハードウェアリアルタイム処理を実現するための拡張機能として開発されたものである。

5.1 RT-Linuxの特徴

RT-Linuxは、フリーソフトであり、Linuxが動作しているコンピュータのカーネルをRT-Linuxに修正することで、リアルタイム処理用のシステムが実現できる。RT-LinuxはLinuxの拡張機能であり、Linuxの持つ各種のソフトウェアをそのまま利用することができる。LinuxにRT-Linuxをインストールすることで、Linuxと

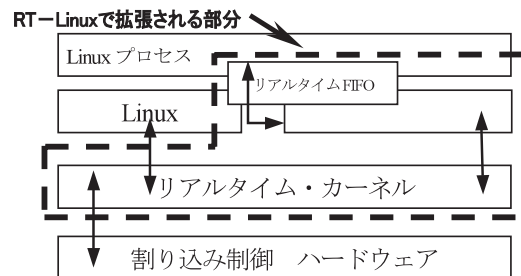


図6 RT-Linuxのデータ処理フロー

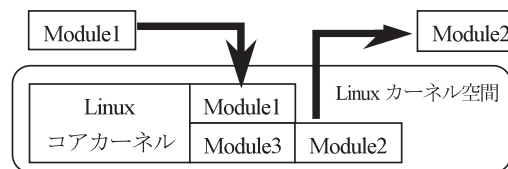


図7 モジュールの組み込みと削除

RT-Linux間のデータ処理の流れは、図6のような形になる。図中の点線で示される部分がRT-Linuxで拡張される部分である。

ハードウェアからの割り込みをRT-Linuxのリアルタイムカーネルで受け付け、Linuxまたはリアルタイムタスクに処理を渡す。リアルタイムタスクとLinuxのプロセス間の情報交換機能として、リアルタイムFIFO (First-In-First-Out) が設けられている。RT-Linuxのユーザアプリケーションプログラムは、Linuxのカーネル空間で動作させる。この機能を用いれば、制御プログラムのようにリアルタイム性の必要な処理のみをRT-Linux上のリアルタイムプログラムとして動作させ、観測データ処理のようにリアルタイム性の必要がない処理をLinux上のプログラムとして動作させることによって、制御処理を実行しながら、観測データをLANに流すようなシステムを簡単に作ることができる。また、これによりLANを通じてのモータの遠隔制御も実現できると考えられる。

5.2 リアルタイムプログラムの実行環境

Linuxカーネルは基本部分(コア)と、自由に組み込みと削除が可能なモジュール(Linuxカーネルに機能を追加するためのプログラム)群で構成されている。図7にモジュールカーネル空間への読み込みと削除の概念図を示す。モジュールの組み込みは、“insmod”、削除は“rmmod”を使用して自由にカーネルの機能

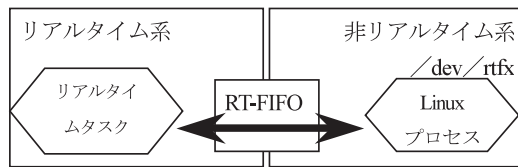


図8 データフローの概念図

拡張や不要な機能の削除ができるようになっている。リアルタイム処理が必要なデジタル制御演算をこのモジュールで実現した。

RT-LinuxのFIFOドライバはLinux上のユーザプロセスとリアルタイムタスク間のデータ伝達用に提供されている。Linuxプロセスからは、標準的なデバイス“dev/rtfx” (xは0, 1, 2...)として、open/close/read/writeインターフェースで利用できる。

本研究では、このFIFOを利用してモータの性能評価などのデータ収集を行うこととした。最初にモジュールを組み込んだ際には、リアルタイムプログラムが実行される。このリアルタイムプログラムでモータの同期制御を行う。その際に、この制御システムの性能評価を行うために、エンコーダの値を測定し、記録する。この処理を2軸の同期が取れて安定するまでの約50秒間実行する。約50秒間の実行が終了した時点で、測定データをFIFOに書き込む。そして、LinuxプログラムからFIFOに書き込まれたデータを読み出し、別のファイルに書き込む。そこまで完了すれば、動作終了となりモジュールを削除する。このようにして、FIFOへの書き込み、読み出しを使用して得られた測定データからシステム評価を行った。なお、この機能は実際の装置では機器の状態監視などに使用できる。

6. 実験

実験では、制御システムのサンプリング周波数を1 kHz (サンプリング周期1ms)、目標回転速度720rpmとし、1軸の安定回転の確認実験、2軸の同期制御実験を行った。

現在のエンコーダの値と過去(1サンプル前)のエンコーダの値と目標位置から偏差を求める。この偏差より式(5)に従ってPI制御演算を行う。これにより求めた値を速度指令値に変換し、D/A変換器に送る。この間、エンコーダの値を測定しておき、制御実験後FIFOを介してLinux側へ観測データを転送し、実験値を評価する。

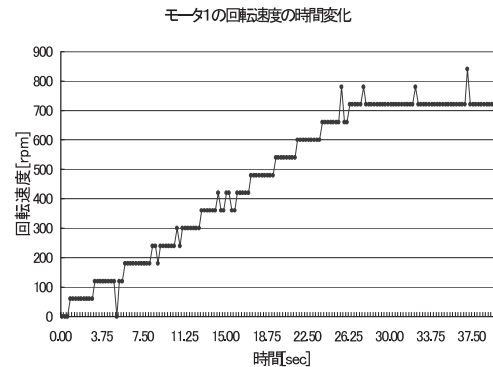


図9 モータ1の回転速度の時間変化

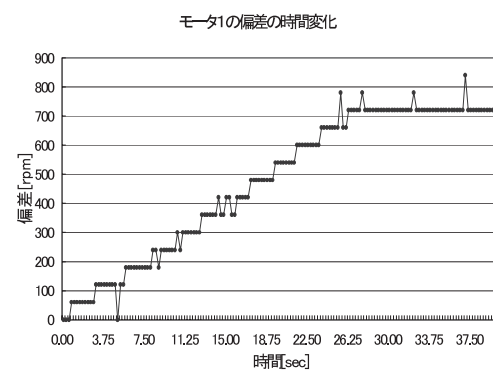


図10 モータ1の偏差の時間変化

6.1 1軸回転評価実験

1軸で運転したときのモータの回転状態の評価を行った。目標値を720rpmに設定し、目標値への追従制御における、回転速度の時間変化、及び、目標速度に対する偏差を測定した。

図9, 10は、モータ1に関する回転速度の時間変化、及び、偏差の時間変化である。図より、720rpm(安定状態)になるまで約20秒かかり、それ以後は常に一定回転をしていることが分かる。モータ2に対しても同様の結果を得た。

6.2 2軸同期制御実験

2軸での同期制御を行うには、2つのモータの位相を合わせなければならない。このため、最初に低速度回転で2つのモータ軸の角度合わせ(位相合わせ)を行い、それから2軸同期制御を行うという手法を採った。各モータからのエンコーダの値の差が、5パルス

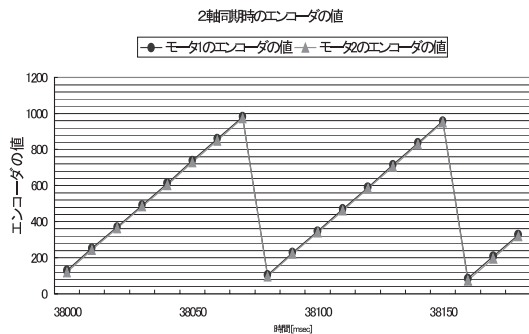


図 1.1 2軸同期制御時の各モータのエンコーダ値

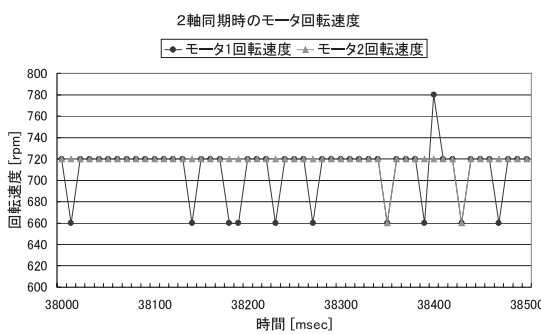


図 1.2 2軸同期制御時の各モータの回転速度

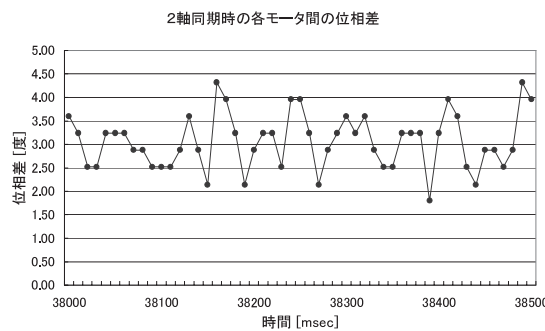


図 1.3 2軸同期制御時のモータ間の位相差

以内になるまで低速度回転から位相を合わせ、その後、モータの目標速度まで回転速度を上げる。

同期を取った後の結果を、図 1.1, 1.2, 1.3 に示す。

図 1.1 は 2 軸同期制御後の各モータのエンコーダの値である。各エンコーダの値（軸角度）が、ほぼ一致していることから同期が取れていることが分かる。

図 1.2 は同期を取った際の回転速度である。同期の初期状態でエンコーダの値に誤差が生じたために、モータ 1 の速度を変化させ同期を修正している様子が分かる。

また、図 1.2 よりモータ 1 の回転速度が 660～780 rpm の範囲で変化していることが分かる。

図 1.3 は、2 軸同期後の各モータ間の位相差（軸角度のずれ）である。最大で約 5 度の位相差が生じているが、ほぼ同期を取って回転していることが分かる。なお、同期は各モータのパルス数を基に実現しているため、パルス数に補正量を加減することによって位相をずらして、図 2 のような位相のずれに伴う各種振動モードを設定することも可能である。

7. まとめ

本研究では、RT-Linux を用いた振動モータの制御システムを開発し、RT-Linux 下のデジタル P I 制御器を用いての一定回転速度運転、及び、2 軸同期制御運転の性能評価実験を行った。

RT-Linux のモジュールとしてデジタル P I 制御器を構成し、モータの一定速度回転を実現した。また、制御データの測定法として RT-Linux の FIFO 領域への書き込み、また Linux 側からの FIFO 領域からの制御データ読み出し、制御データのファイルへの書き込み、などを実現して制御データの測定手法を確立した。

2 軸同期制御では、起動時に各モータの位相を合わせてから目標速度までの制御を行い、それから 2 軸同期を行う手法を実現した。しかし、この方法では 2 軸同期が取れるまでに時間がかかってしまうという欠点がある。その改善方法に関しては検討課題とする。

RT-Linux を利用する中で、RT-Linux は Linux の機能を拡張しているために容易にネットワーク機能が利用でき、技術的に遠隔制御が実現可能であるということも分かった。今後は、ネットワークを用いての遠隔制御の実現を目指す。

文献

- 1) 原田香織: “振動モータの 2 軸同期制御方式の検討”, 平成 11 年度徳山工業高等専門学校特別研究論文集, PP. 61-66 (2000)
- 2) J. H. Ruwan V Perera: “Linux を用いた誘導電動機の二軸同期制御”, 平成 11 年度徳山工業高等専門学校卒業研究論文集, PP. 125-128 (2000)
- 3) 森友一朗, 薬師輝久, 馬場秀忠: “RTLinux リアルタイム処理プログラミングハンドブック”, (株) 秀和システム (2000)

(2002. 9. 2 受理)