

理 工 学 研 究 科

2020年1月

修 士 論 文

電流兆候解析に基づく
DCモーターのオンライン故障検知

47018305 大 窄 直 樹

(情 報 科 学 専 攻)

電流兆候解析に基づく DC モーターのオンライン故障検知

大窄 直樹

内容梗概

本論文では, DC モーターのオンライン故障検知を電流兆候解析により行う. モーターの保守運用は時間基準方式から状態保全方式に移行され, オンライン故障検知及び劣化に応じたハードウェア制御が求められている. 電流兆候解析は, 個々のモーター関連設備にセンサーを設置することなく故障検知できるため実装及び管理のコストを削減できるが, 故障検知の過程で周波数解析を行う必要があり, 計算量が多くオンライン化が困難という課題がある. この課題を解決するため本論文では, 電流兆候解析をハードウェア併用することでオンライン故障検知を試みる. 電流兆候解析の手法として, 故障したモーターの場合にはモーターに流れる電流の低周波数帯の高調波成分が増加することを利用して故障の検知を行う. 実装はプログラマブル SoC で行い, 計算量の多い周波数解析をハードウェア実装し, 他の計算量の少ない処理をソフトウェア実装する. 提案手法に基づき実験を行った結果, PWM 信号の duty を変更しても故障検知が可能で, 周波数解析を高速化することでオンライン故障検知が可能になった.

キーワード

電流兆候解析, 故障検知, ハードウェア設計

目次

第1章 序論	1
第2章 モーターの故障検知	3
2.1 DC モーターとその制御	3
2.1.1 モーターの種類	3
2.1.2 モーターの構造と故障	3
2.1.3 PWM 制御	4
2.2 モーターの故障検知	4
2.2.1 故障検知の手法	4
2.2.2 DC モーターの故障検知	6
第3章 予備実験	9
3.1 DC モーターの周波数成分	9
3.1.1 周波数成分の取得	9
3.1.2 DC モーターの周波数成分	10
3.2 PWM 信号と周波数成分の関係	11
第4章 オンライン故障検知のプログラマブル SoC による実装	15
4.1 オンライン故障検知の方式	15
4.2 プログラマブル SoC	15
4.3 実装	16
4.3.1 概要	16
4.3.2 信号の取得	17
4.3.3 データの固定小数点化	17
4.3.4 ソフトウェアとハードウェア間の通信	18
4.3.5 FFT のハードウェア実装	18
4.3.6 周波数解析	18
4.4 実験結果	19
4.4.1 負荷を与えていないモーターでの実験結果	19
4.4.2 負荷を与えたモーターでの実験結果	20
4.5 考察	20

第 5 章 結論	23
謝辭	25
参考文献	29

第1章 序論

モーターは冷蔵庫, エアコン, 新幹線等, 多岐に渡る機器に内蔵されている。これらの常時作動している設備用電気機器によって, 暮らしの安全や快適さがサポートされている。このような機器が故障し動作を停止すると人々の生活に影響を与えるため, それらを未然に防ぐ対策が求められる。

従来行われてきた故障の対策として時間基準保全がある。時間基準保全方式の多くの場合には, 保全作業を行う時期が経験則に基づいており, 実際の個々の電気機器に対してその時期が適切であるかどうかの判断はできていない。そのため, 交換のサイクルが短い場合にはコストが増大し, 長い場合には安全に対するリスクが高まる。また, 想定外の事態によって引き起こされる故障は防止できない欠点もある。そこで近年, 状態基準保全という使用機器への日常的な診断により機器状態を推定する方法が採用されている。故障や寿命の時期を詳細に予測して整備, 修理, 交換を行えるためコストを削減することができる [1]。

モーターの点検は, 作業員の感覚に依存するため, 故障の判断が一意ではない。また, 多数のモーターを使用する設備では, 点検が容易でなく多大なコストがかかる。これらの課題は, 作業員の感覚に頼った診断からシステム化された診断基準に統一化で解決できる。

機器の状態を推定する手法には振動解析, 温度解析, 電流兆候解析等がある [2]。振動解析, 温度解析は振動・温度データから診断するため, 振動・温度センサーをモーター関連設備に新たに取り付けるための導入・保守コストを必要とする [3]。電流兆候解析は, 個々のモーター関連設備に振動・温度センサーを設置する必要なく, 電気室にある制御盤内に設置された電流センサーによるセンシングデータをもとに遠隔で分析を行えるため, この手法が故障検知の主流になりつつある。

電流兆候解析は, 固定子と回転子のエアギャップの変化, 回転子の抵抗値の変化などが空間時速線に影響を与え, 逆起電力を通して固定子に流れる電流に反映しているという理論に基づいたもので, 回転子バーの破損, 回転子の静的/動的偏心などの故障を検知できる。また, 検出信号に信号処理を施すことで, 回転子バーの故障, 軸受の故障などの故障も検知できる [4]。しかし信号処理で行う周波数解析の計算量が多いため, 故障検知のオンライン化及び劣化に応じたモーター制御が困難という課題がある。これに対し, 電流兆候解析のオンライン化に対する研究が行われている。

文献 [5] は, 電流兆候解析によりモーターのローターバーの故障検知を行っている。文献 [6] は, ブラシレス DC モーターの電流兆候解析による故障検知を行っている。しかし, これらの手法は電流兆候解析の処理の一部である周波数解析の計算量が多く, 時間を要するためオフラインでの実装である。さらに, ブラシレス DC モーターの故障検知では, PWM 信号の duty を変更した故障検知を行っていない。また, 文献 [7] は, 電流兆候解析の高速化のためモーターの電流信号をハードウェアに送信しているが, オンライン故障検知を実装するまでには至っていない。

そこで, 本論文では電流兆候解析に基づく DC モーターのオンライン故障検知を試みる。電流兆

候解析の手法として、モーターに流れる電流の低周波数帯の高調波成分が、故障したモーターの場合に増加することを利用して故障を検知する。実装はプログラマブル SoC で行い、計算量の多い周波数解析をハードウェア実装し、他の計算量の少ない処理をソフトウェア実装する。提案手法に基づき実験を行なった結果、PWM 信号の duty を変更しても故障検知が可能で、周波数解析を高速化することでオンライン故障検知が可能になった。

第2章 モーターの故障検知

2.1 DC モーターとその制御

2.1.1 モーターの種類

モーターは、AC (Alternating Current) モーターと DC (Direct Current) モーターに大別することができる。

AC モーターは交流の電源で駆動し、周波数に応じて一定の回転速度を保つことができる。また、DC モーターと異なり構成部品にブラシが無いため、耐久性のある装置の作成ができる。しかし周波数に応じた一定の回転速度しか得られないため、過負荷による停止、周波数が高い、起動しない等の場合には速度制御をする動作に向いていない。

DC モーターは直流の電源で駆動し、電圧に対しての回転特性が安定していて扱いやすく低価格である。重ねて、低電圧で動作するため装置の小型化に向いている。しかし構造にブラシ、整流子といった部品が含まれているためノイズや回転にムラがあるのと同時に壊れやすいといった課題がある。また、DC モーターには、DC モーターからブラシを取り去ったブラシレス DC モーターがある。ブラシがなくなったことにより耐久性が向上し、電氣的ノイズも小さくなった。

2.1.2 モーターの構造と故障

ブラシ付き DC モーターの構造を図 2.1 に示す。整流子とブラシが、DC モーターの特徴となる部品である。整流子はコイルに流れる電流の向きを切り替え、磁極の向きを逆転させることにより回転子は右回転する。軸とともに回転する整流子にはブラシから電力が供給されている。整流子がブラシと当たって給電し、整流子とブラシが回転により触れる箇所が変更することで電流の向きが変わる。コイルは磁界の中に置かれ、流れる電流によりコイルが片方の磁極に反発し、同時に反対側が別の磁極に引かれることで回転する。

モーターの主な故障は下記の 4 つに分類できる [2]。

1. 軸受の故障

軸受の故障は、雑音、振動の増大を引き起こす。モーターの故障の 40-50% を占める。

2. 固定子、電機子の故障

固定子、電機子の故障は絶縁故障を引き起こす。モーターの故障の 30-40% を占める。

3. 回転子の故障

回転子の故障は平均トルクの減少及び、ホットスポットの生成を引き起こす。モーターの故障の5-10%を占める。

4. 静的／動的な偏心

偏心は固定子と回転子間の摩擦の原因となり、固定子、回転子の故障を引き起こす。

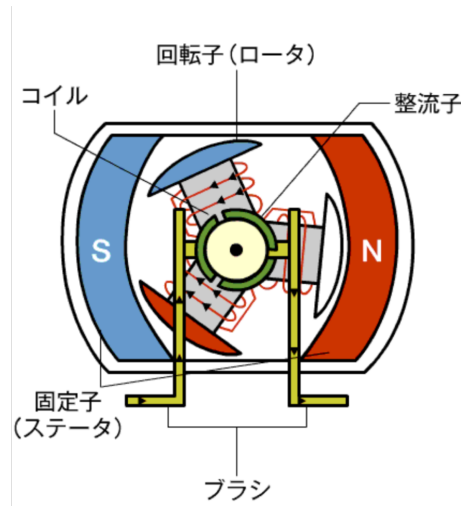


図 2.1: DC モーターの構造 [8]

2.1.3 PWM 制御

従来の制御方式は、電圧を上下させてモータの速度を変えている。この方式は電圧の強さを変えるために専用の回路を用意する必要がある。付け加えて、電圧の上昇／下降により発熱するためエネルギー効率が非常に悪い。

これらの課題を解決するために、最近ではPWM (Pulse Width Modulation) 制御が用いられる。PWM 制御では、パルス幅の duty を変更することによりモーターを制御する。図 2.2 は、PWM 制御を用いた信号の例である。上から duty が 25%, 50%, 75% の信号である。周期の high の割合を duty という。

2.2 モーターの故障検知

2.2.1 故障検知の手法

故障検知手法には下記のような方法がある [2]。

1. 電磁場解析

固定子、回転子間の電磁場を解析することにより故障検知を行う。

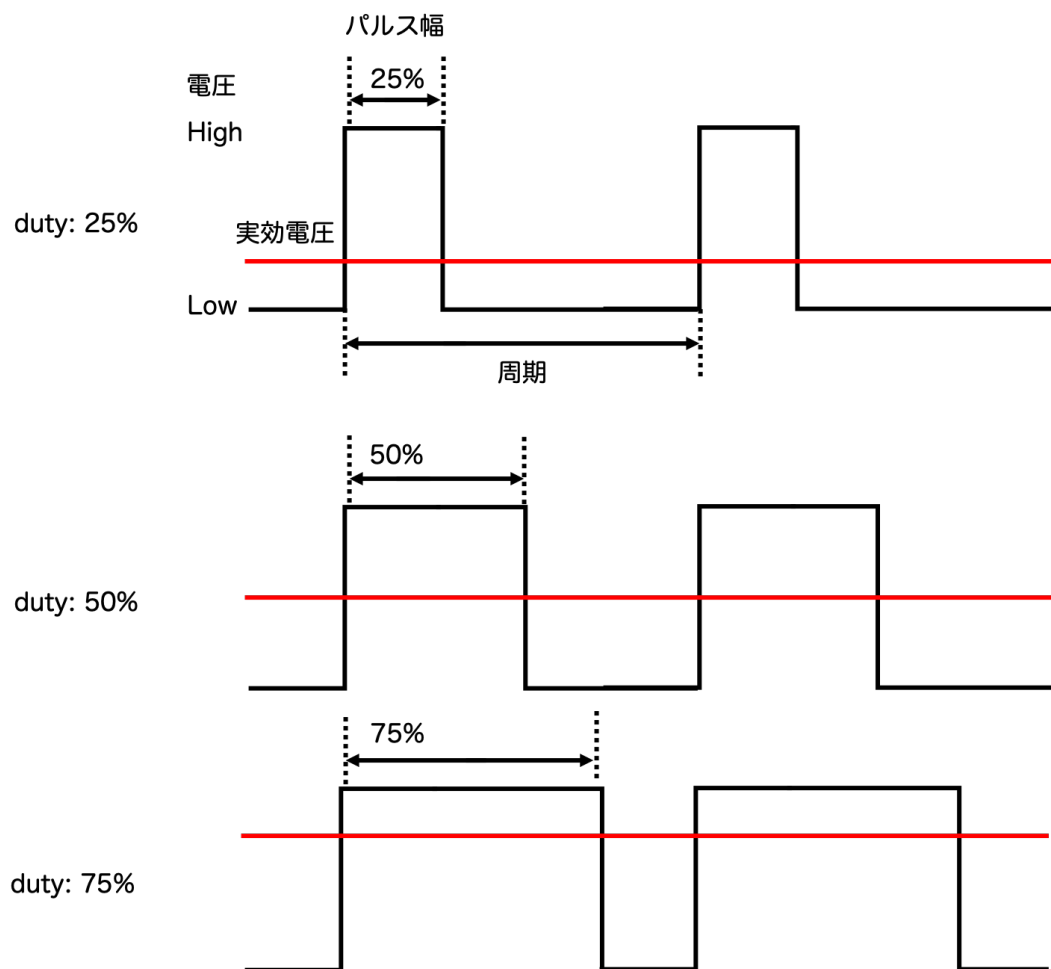


図 2.2: PWM 信号

2. 温度解析

モーターの駆動時のコイルの温度変化を解析することにより故障検知を行う。

3. 振動解析

モーター振動を解析し、抽出した特徴に基づいて故障検知を行う。

4. 音響解析

駆動時の異常音や突発的に発生する異音から故障検知を行う。

5. 電流兆候解析

駆動時の電流信号の変化から故障検知を行う。

この中で故障検知において、低コスト、高精度という点から近年注目されているのが電流兆候解析である。電流兆候解析は、固定子と回転子のエアギャップの変化、回転子の抵抗値の変化などが空問時速線に影響を与え、逆起電力を通して固定子に流れる電流に反映しているという理論に基づいたものである [4]。電気室にある制御盤内に設置された電流センサーによるセンシングデータをもとに遠隔で分析を行えるため、初期コストが低い特徴がある。回転子、固定子、静的/動的な偏心の故障などを検知することができる。また、検出信号に信号処理を施すことで軸受の故障も検知できる。電流兆候解析は、AC モーター、サーボモーター、BLDC モーター等広い範囲で適用され効果をあげている。

しかし DC モータは、回転数が一定でないため適用が難しく適用されていない。また、信号処理で周波数解析を行う必要があり処理に時間がかかりオンライン故障検知が困難という課題がある。

2.2.2 DC モーターの故障検知

文献 [7] では、電流兆候解析に基づいてブラシレス DC モーターのオフライン故障検知を行っている。対象とする故障は歯車の歯こぼれで、歯が 1 つ欠けた故障及び、歯が 2 つ欠けた故障を対象にしている。正常なモーターと故障したモーターの電流信号に与える影響調べるため比較実験を行う。それぞれで電流信号の周波数解析を行い周波数表現にし、そのデータからパワースペクトルを生成する。

0-50Hz 帯の最大周波数成分を表 2.1 に示す。Amplitude は最大周波数成分、Frequency は最大周波数成分の周波数、% Change In Amplitude は正常なモーターの最大周波数成分に対して故障したモーターの最大周波数成分の増加割合である。Health は正常なモーター、G1 fault は歯車の歯が 1 つ欠けたモーター、G2 fault は歯車の歯が 2 つ欠けたモーターである。故障したモーターは正常なものとは比べ最大周波数成分が高いことが分かる。しかし、この研究では duty を変化させた故障検知及びオンライン故障検知を行っていない。

表 2.1: 周波数解析結果 [7]

Peak	Health	G1 fault	G2 fault
Amplitude	28.50	33.47	86.18
Frequency (Hz)	16.97	16.97	16.97
% change in amplitude	0	+17.44	+202.38

第3章 予備実験

電流兆候解析により DC モーターの故障検知を行うため、DC モーターの周波数成分の特徴を調べる予備実験を行った。正常なモーターと故障したモーターそれぞれで電流信号データの低周波数帯を解析し電流兆候解析により故障検知可能か調べる。また、DC モーターは duty により回転速度が調整が可能のため、duty による周波数成分の変化についても調べる。

3.1 DC モーターの周波数成分

3.1.1 周波数成分の取得

文献 [7] のブラシレス DC モーターの故障検知ではブラシレス DC モーターが故障した場合、電流信号データの低周波数帯の高調波成分影響を与えた。そこで、DC モーターの場合には周波数成分にどのような影響を与えるのか調べた。

周波数成分の取得は、図 3.1 の構成で行った。実験に用いた DC モーターは、小型直流モーターのマブチモーター RS-380PH である。この DC モーターを正常なモーター 1 個、故障したモーター 1 個の合計 2 個の利用した。故障した DC モーターには、あらかじめ過電流・過電圧で一定時間駆動させるものを利用した。また、この DC モーターを PWM 制御するためにモーターコントローラーボードである MXA033 を用いた。電流信号の解析に利用するスペクトラムアナライザには Analog Discovery2 を用いた。矢印の向きは電流の向きである。

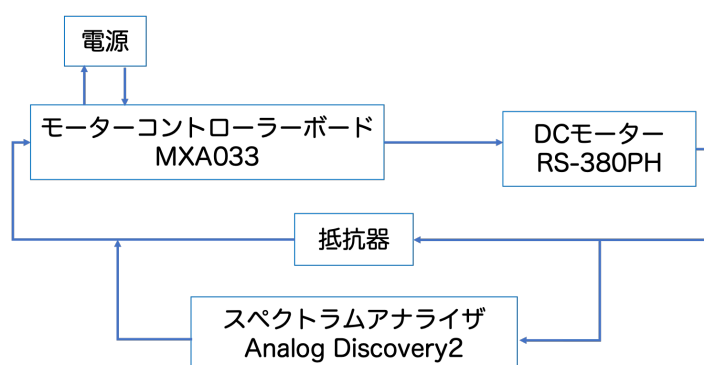
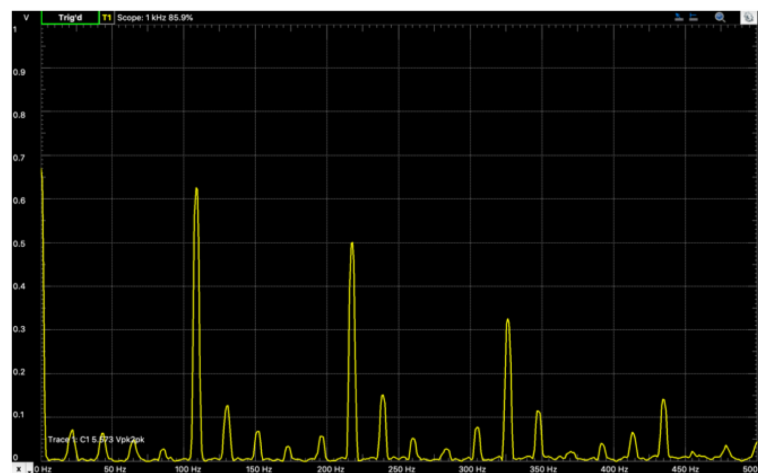


図 3.1: 周波数成分の取得

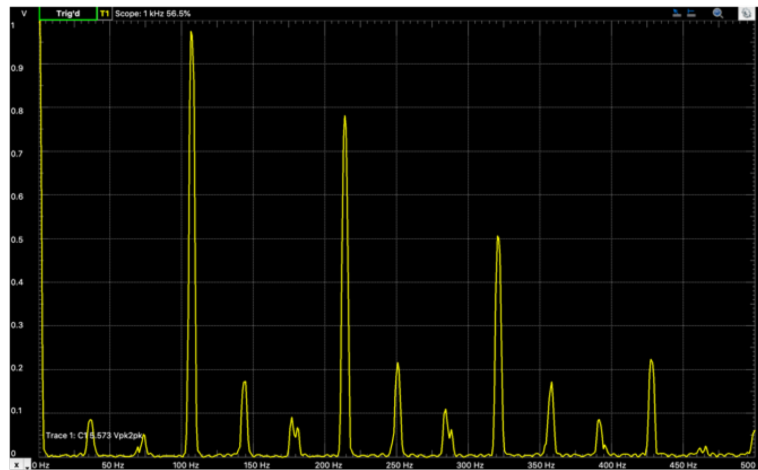
3.1.2 DC モーターの周波数成分

取得する周波数成分のサンプル数は 512 点, DC モーターの duty は 50%, 周波数帯は 0-500Hz のデータを取得する. 図 3.2 は取得した周波数成分をパワースペクトルにしたものである. (a) が正常なモーター, (b) が故障したモーターのパワースペクトルで, 縦軸はパワーで単位は V, 横軸は周波数で単位は Hz である. 正常なモーター, 故障したモーターともに最大周波数成分は約 100Hz に位置していて, 正常なモーターの最大周波数成分が約 0.65V で, 故障したモーターの最大周波数成分は約 0.98V である.

この実験により, 文献 [7] と同様に DC モーターでも故障により低周波数帯の最大周波数成分が増加することが確認できた. したがって, 正常なモーターと故障したモーターは低周波数帯の最高周波数成分の大きさにより判別できることがわかった.



(a) 正常時



(b) 故障時

図 3.2: DC モーターのパワースペクトル

3.2 PWM 信号と周波数成分の関係

PWM 信号を変化をさせることにより、低周波数帯の最大周波数成分にどのような影響を与えるか調べる。

DC モーターを正常なモーター 10 個、故障したモーター 5 個の合計 15 個の利用する。PWM 信号の duty が 25%, 50%, 75% の場合について調べる。正常なモーターは各モーター、各 duty に対して 5 回最大周波数成分を取得し、故障したモーターは 10 回最大周波数成分を取得する。すなわち、正常なモーター及び故障したモーターの最大周波数成分を 50 個取得する。

取得した duty ごとの最大周波数成分を図 3.3, 図 3.4 に示す。

図 3.3 は離散確率分布図で、横軸が最大周波数成分での単位は dbV, 縦軸は確率での単位は%である。オレンジが正常なモーターの最大周波数成分、青が故障したモーターの最大周波数成分である。duty は図の上から 25%, 50%, 75% である。正常なモーターの最大周波数成分は、故障したモーターの最大周波数成分と比べ低い値を示している。例えば duty が 50% の時、正常なモーターの最大周波数成分は約 -6.0dbV から -7.8dbV の間に集中しているのに対し、故障したモーター最大周波数成分は約 -0.5dbV から -5.0dbV に集中している。

故障したモーターの最大周波数成分が正常なものとは比べ大きい関係は、duty が 25%, 50%, 75% のとき全てで成り立っている。そのため、duty が変化しても低周波数帯の最高周波数成分を調べることで正常と故障を分別できることがわかった。

また、図 3.3 から duty が小さいほど、正常時及び、故障時ともに最高周波数成分が増加することがわかった。

図 3.4 は箱ひげ図で、縦軸は最大周波数成分で単位は dbV である。オレンジが正常なモーターの最大周波数成分、青が故障したモーターの最大周波数成分である。duty は図の上から 25%, 50%, 75% である。duty の増加に対し正常なモーター及び故障したモーターの最大周波数成分の四分位範囲が大きくなっている。

また、duty が大きいほど最大周波数成分の四分位範囲が大きいことを確認した。

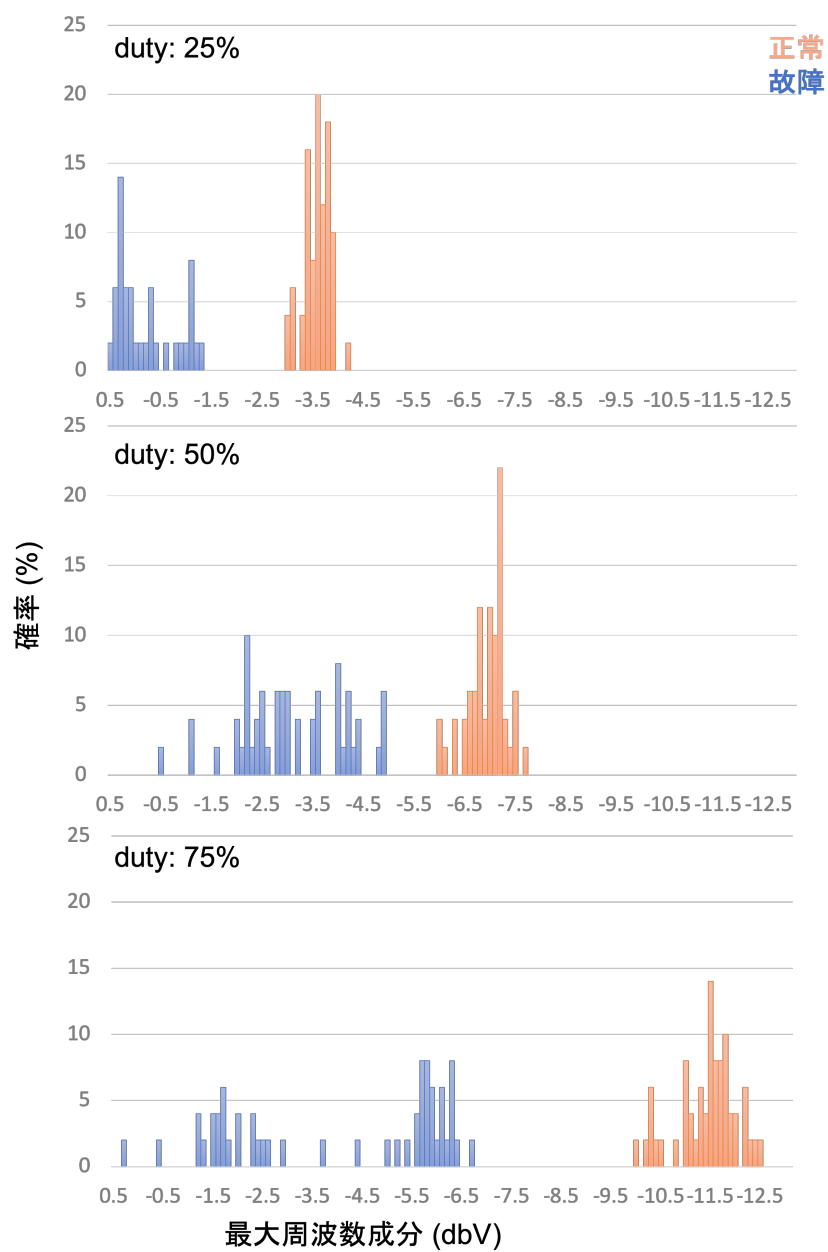


図 3.3: 最大周波数成分の離散確率分布図

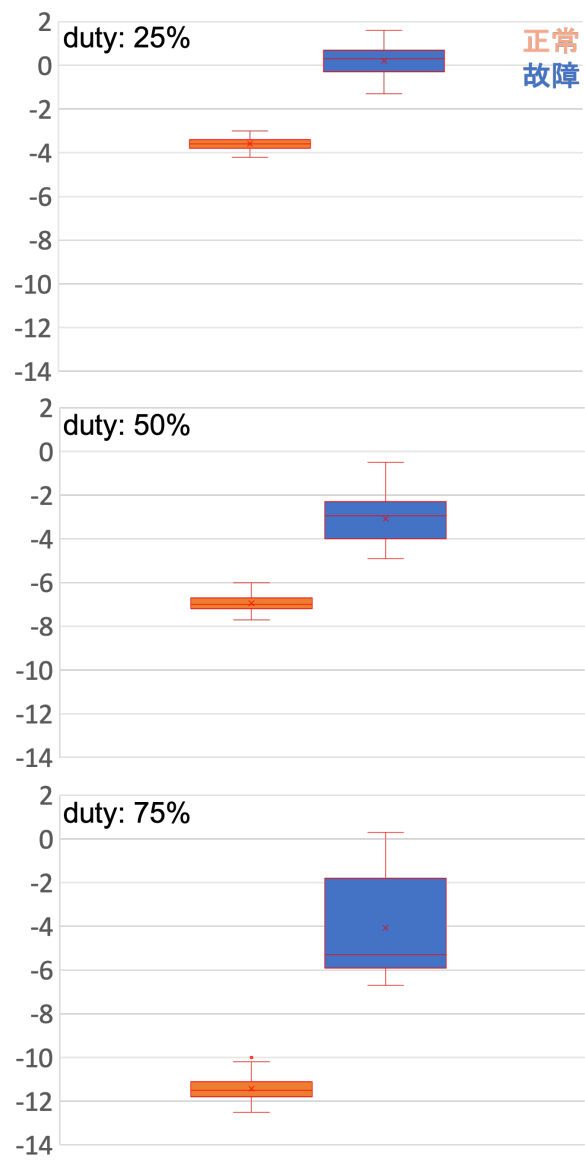


図 3.4: 最大周波数成分の箱ひげ図

第4章 オンライン故障検知のプログラマブル SoC による実装

4.1 オンライン故障検知の方式

本論文では電流兆候解析に基づく DC モーターのオンライン故障検知を行う。従来電流兆候解析の処理の周波数解析で用いられるフーリエ変換の処理の計算量が多くボトルネックとなってるため、オンライン故障検知ができなかった。そこで提案手法では、フーリエ変換のハードウェア化を行うことで電流兆候解析のオンライン故障検知を試みる。

フーリエ変換はハードウェア実装に向いていて高速化な固定小数点高速フーリエ変換 (固定小数点 FFT) を用いる。

故障検知の方法には電流兆候解析を用いる。モーターに流れる電流の低周波数帯の高調波成分が、故障したモーターの場合に増加することを利用して故障を検知する。PWM 信号の duty が増加に従い、モーターの最高周波数成分は減少することを利用することで、duty が変化しても故障検知を行うことができる。

提案手法による実装を図 4.1 に示す。計算量の多いフーリエ変換の処理をハードウェア化し、他の計算量の少ない処理であるデータ取得、データの前処理、スペクトラム解析をソフトウェアで行う。まず、ソフトウェアでモーターから電流信号を取得し、ハードウェアで効率的に計算させるためにデータの固定小数点化を行う。次に、ハードウェアで固定小数点 FFT を行う。最後に、得られた周波数からスペクトル解析を行うことで故障検知を行う。

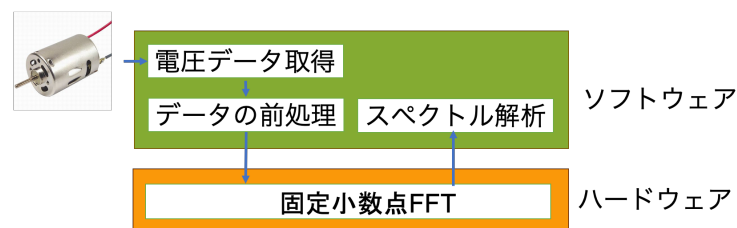


図 4.1: 電流兆候解析のオンライン故障検知

4.2 プログラマブル SoC

本論文ではプログラマブル SoC を用いて実装を行う。プログラマブル SoC とは、ハードマクロのプロセッサと FPGA を 1 チップ上に搭載した SoC のことである。特徴として、プロセッサと

FPGA の接続がチップ上に実装されるため、両者の高速な通信が可能であることが挙げられる [9]. ハイエンドの組み込みシステムを対象としたプログラマブル SoC である Zynq-7000 は、プロセッシングシステム (PS) およびプログラマブルロジック (PL) の 2 つで構成されている. 前者はデュアルコアの ARM Cortex-A9 プロセッサを中心として、キャッシュ、オンチップメモリ、外部メモリインタフェース、DMA コントローラ、および、入出力ペリフェラルで構成されている. 後者は構成可能論理ブロック、RAM、デジタル信号処理ブロック、プログラマブル入出力ブロック、シリアルトランシーバ、および、A-D コンバータで構成されている. PS の ARM Cortex-A9 プロセッサの最大動作周波数は 600MHz から 1GHz となっている. これは、FPGA 上にソフトコアとしてプロセッサを合成した場合のものと比べて、数倍高速なものとなっている. また、PS と PL は複数層の ARM AMBA (Advanced Microcontroller Bus Architecture) AXI (Advanced eXtensible Interface) を使用して接続されており、両者間の高速な通信を可能にしている.

プログラマブル SoC に ZYBO Zynq-7000 Development Board を用いて実装を行う. ZYBO Zynq-7000 Development Board は Zynq を搭載した評価ボードである. Zynq は、FPGA とプロセッサを搭載した Xilinx 社の IC チップ製品群の総称であり、CPU をベースとした PS (Processing System) と従来の FPGA に相当する PL (Programmable Logic) が一つのチップの中に入っている [10]. そのため、Zynq を用いることにより FPGA によるハードウェア開発だけでなくプロセッサを用いたソフトウェア開発や既存のソフトウェア資源の活用が可能となる. また、PS 部と PL 部はチップ内部で接続されているため、PS 部と PL 部間での高速なデータ転送を行うことができる.

Xillynx は Xillybus 社がリリースした Ubuntu 12.04LTS に基づく Linux ディストリビューションである. Zynq に Xillynx を導入することで、Linux を動作及び、Xillybus を用いて PS-PL 間の通信を実現できる.

4.3 実装

4.3.1 概要

本論文では ZYBO Zynq-7000 Development Board を用いて電流兆候解析のオンライン故障検知を行った. また Linux を動作及び、PS-PL 間との通信を実現のために OS に Xillynx を用いた. システムの全体像を図 4.2 に示す. これは ZYBO Zynq-7000 Development Board を用いて実装した. 図の深緑の実装をが PS 部で行い、オレンジの実装を PL 部で行った. 本手法では、駆動しているモーターから電流信号を取得した. その後取得した電流信号を加工しデバイスドライバを通して PL 部にデータを送った. データの送信が終わり次第、PS 部から PL 部にデータの送信終了信号をデバイスドライバを通して送り PL 部分で 512 点の固定小数点 FFT を行った. その後、PS 部から PL 部のデータを取得しスペクトル解析を行うことでモーターの劣化及び故障を検出した.

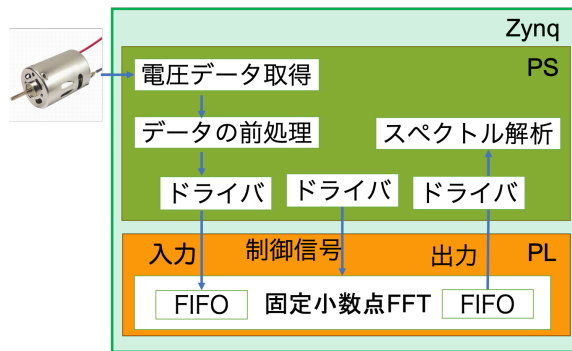


図 4.2: プログラマブル SoC を用いた実装

4.3.2 信号の取得

本手法では, Analog Discovery2 を用いてオンラインに電流信号を取得した. Analog Discovery2 は専用ソフトである Waveforms を用いて javascript で制御を行うことができる. Analog Discovery2 のオシロスコープは非常に高速で USB のフルスピードの 12Mbps を超える速度でデータを取得できる. Waveforms でオシロスコープのサンプリング時間を 500ms, サンプリング数を 512 点に設定を行った. その後, 図 4.3 のプログラムを実行することでオンラインに電圧を取得できた. `Scope1.Channel1.data` は設定を行って取得したデータで, `data.txt` に保存した.

```

1   while (true) {
2       var data=Scope1.Channel1.data
3       File("~/data.txt").write(data)
4   }

```

図 4.3: オシロスコープの制御プログラム

4.3.3 データの固定小数点化

ハードウェアの浮動小数点の計算は固定小数点の計算と比べると, 計算速度, 消費電力などの面で後者が優ってる. 取得した電流信号は電圧データのためほとんどのデータが 0-1.0V の間で, 浮動小数点のデータであった. そのため, 取得した電圧データの固定小数点化を行った. 実装は PS 部で C 言語を用いて行った.

本論文では浮動小数点データを 100 倍し, 小数点以下を切り捨てることで固定小数点化を行う. この処理は PS 部で行う. 固定小数点化の具体例を図 4.4 に示す. 図の左がモーターから取得した電圧データ, 右がデータを加工して固定小数点化したデータである. 例えば 0.9938883564469071 の場合, 100 倍すると 99.38883564469071 になり小数点以下を切り捨てることで 99 となることで固定小数点になる.

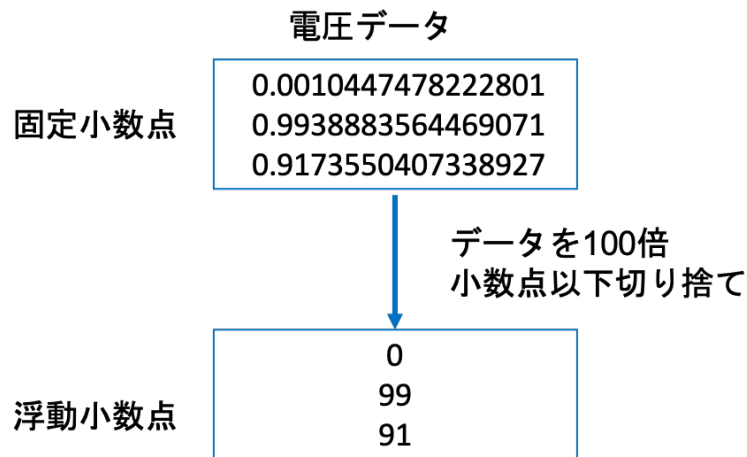


図 4.4: データの固定小数点化

4.3.4 ソフトウェアとハードウェア間の通信

Xillynux では PS 部と PL 部の通信を高速に行うためにデバイスドライバが提供されている。書き込み用のドライバドライバである xillybus_write_32 を用いて PL 部にデータを書き込みを行った。また、読み込み用のドライバドライバである xillybus_read_32 を用いて PL 部のデータを読みを行った。

4.3.5 FFT のハードウェア実装

ハードウェア設計言語 VHDL を用いて PL 部に FFT のハードウェア実装を行った。FFT は 512 点の固定小数点高速フーリエ変換を実装した。PS 部と PL 部でデータの読み書きを行うため、PL 部に入力の FIFO と出力の FIFO の合計 2 個の FIFO を設置した。また、PL 部の制御のためにデバイスドライバを作成した。PL 部での FFT 実装を下記のような流れで行った。まず、PS 部から固定小数点の電圧データを PL 部に書き込んだ。書き込み終了後、PS 部からデバイスドライバを通して計算開始の制御信号を送信した。制御信号を受け取ったら FFT の処理を始め、計算結果を出力用 FIFO にデータを格納させた。この動作は 4,656 cycle 処理でき、LUT は 17,600 である。動作周波数を 100MHz とすると $46.5 \mu s$ で実行できる。これは十分にオンライン故障検知可能な範囲である。

4.3.6 周波数解析

PL 部から FFT の結果を読み取り周波数解析を行った。まず、受け取った周波数スペクトルを二乗しパワースペクトルに変換した。その後、パワースペクトルの値のうちもっとも大きい値、すなわち最大周波数成分を求めた。また、元の電圧信号から duty を求めた。これは電圧信号 512 点の

うち、0.01V より大きい値の割合を計算することにより求めた。この二つの特徴量、最大周波数と duty からモーターの劣化及び故障を検出した。

4.4 実験結果

4.4.1 負荷を与えていないモーターでの実験結果

本実装で正常なモーター 1 個と故障したモーター 1 個それぞれでデータを 100 個取得した。duty 変更時でも故障検知できるか確かめるために、それぞれのデータの取得間隔を 1 秒とし duty を変更させながらデータを取得した。取得したデータを表したものが図 4.5 である。縦軸が最大周波数で単位は V^2 、横軸は duty で単位は%である。オレンジの点が正常なモーターから取得したデータ、青の点が故障したデータである。また正常なモーター、故障したモーターのデータを最小二乗法により線形予測したものがオレンジの直線、青の直線である。

故障したモーターの最大周波数成分は、正常なものとは比べ、任意の duty においてほとんどのデータが大きなものとなっている。また、線形予測の直線により、両方のモーター共に duty の増加に伴い最大周波数成分が小さくなった。

これらの結果から、故障したモーターの線形予測を下回っていたら正常、上回っていたら故障とすることで故障検知を行うことができる。

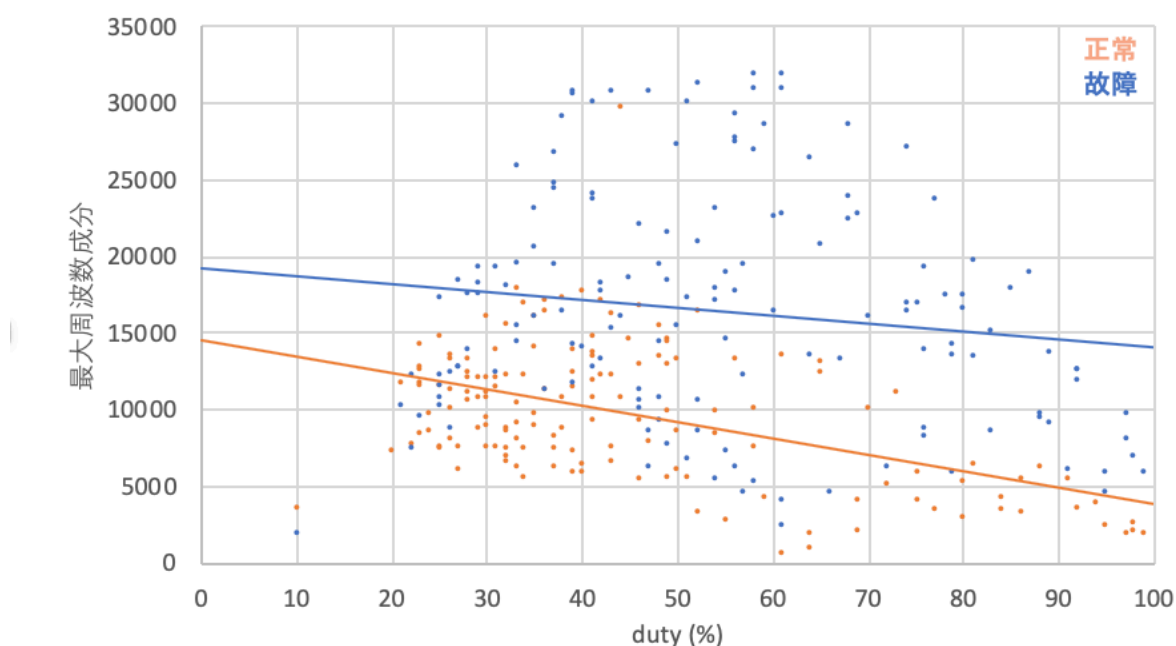


図 4.5: 負荷を与えていないモーター実験結果

4.4.2 負荷を与えたモーターでの実験結果

負荷を与えた状態で正常なモーターと故障したモーターそれぞれでデータを 100 個取得した。モーターに一定の負荷を与えるため、計測用モーターとは別のモーターを負荷用モーターを準備した。計策用モーターと負荷用モーターのシャフト同士を接着し、負荷用モーターを計測用モーターに対して逆回転させることで一定の負荷を与えた。負荷を与えた状態で計測を行い取得したデータを図示したのが図 4.6 である。

正常なモーターは負荷ありの場合と負荷なしの場合で大きな変化が見られないのに対し、故障したモーターは大きな変化があった。故障したモーターは負荷を与えることにより最大周波数成分が低くなった。その結果、正常なモーターと故障したモーターのデータに有意な差は無くなった。

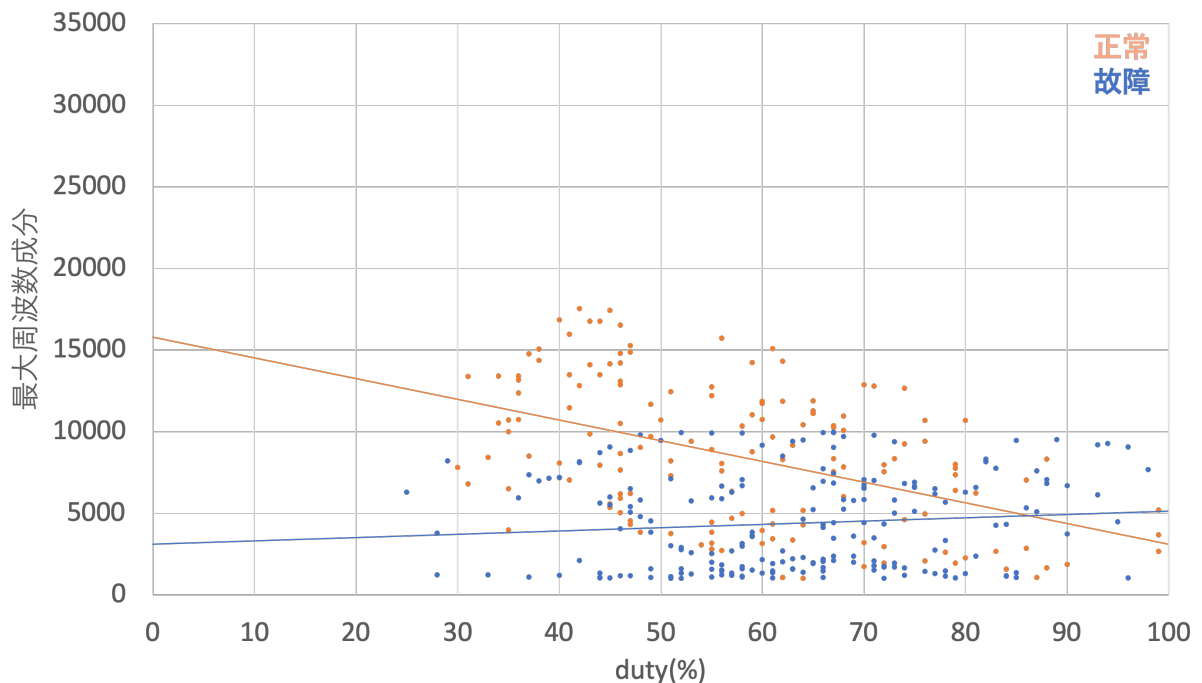


図 4.6: 負荷を与えたモーターの実験結果

4.5 考察

提案手法により、電流兆候解析に基づく DC モーターのオンライン故障検知を実現できた。

データ取得、データの前処理、スペクトラム解析の計算量が $O(N)$ ののに対し、FFT の計算量は $O(N \log N)$ でボトルネックとなっていた。そこで、FFT をハードウェア実装を行い高速化するとこにより故障検知のオンライン化を行った。

本研究では、負荷を与えていないモーターと負荷を与えたモーターで電流兆候解析による故障検知を行った。負荷を与えていないモーターで行った実験では、故障したモーターと正常なモーターで duty に応じた最大周波数に有意な差があった。モーターの最大周波成分は故障が増加するほど、

最大周波数成分もまた増加する。これは、モーターがどの程度劣化しているかの目安になり、ある一定の値より大きくなることで故障と判断することができる。またモーターに負荷を与えて計測した場合、正常なモーターにはほとんど影響を与えないが、故障したモーターには影響を与え最大周波数成分が小さくなった。その結果、本手法で故障検知を行う場合、正常なモーターと故障したモーターともに正常なモーターと判断される。故障したモーターの最大周波数成分が減少した理由は、故障が原因でトルク小さく外部からの負荷が弱いためだと考える。

第5章 結論

本論文では、電流兆候解析に基づく DC モーターのオンライン故障検知を提案した。電流兆候解析をハードウェア併用することでオンライン故障検知実装した。電流兆候解析の手法として、故障したモーターの場合にはモーターに流れる電流の低周波数帯の高調波成分が増加することを利用して故障の検知を行った。実装はプログラマブル SoC で行い、計算量の多い周波数解析をハードウェア実装し、他の計算量の少ない処理をソフトウェア実装した。提案手法に基づき実験を行った結果、PWM 信号の duty を変更しても故障検知が可能で、周波数解析を高速化することでオンライン故障検知が可能になった。

本手法では、一定時間過電圧・過電流で駆動させたモーターを故障したモーターと定義して故障検知できることを確認したが、他の故障も検知できる可能性がある。

今後の課題は、負荷を与えたモーターでの故障検知、及び劣化を抑制するようなモーター制御が重要な課題となる。

謝辞

本研究に際し、多くの方々から御助言、御助力を賜りました。ここに感謝の意を表します。本研究の機会を与えていただき、数々の御助言、御指導を賜りました石浦菜岐佐教授に心より感謝いたします。本研究に関する有益な議論の場を提供していただき、様々な視点から御指導いただきました京都高度技術研究所の神原弘之氏に感謝いたします。本分野について深い知見をもち、多くの御助言、御指導いただきました立命館大学の富山宏之教授に感謝いたします。技術的な面で多くの御助言をいただきました元立命館大学の中谷嵩人氏に感謝いたします。本研究に関して御協力、御討議頂いた関西学院大学石浦研究室 HLS チームの諸氏に深く感謝いたします。研究にかぎらず私生活の面でも多くの御支援を賜りました石浦研究室の諸氏に感謝いたします。

関連発表文献

1. 大窄直樹, 石浦菜岐佐, 神原弘之: “電流兆候解析に基づく DC モーターのオンライン故障検知,” 電子情報通信学会技術研究報告, (Mar. 2020 to appear).
2. N. Osako, S. Ota, S. Yura and N. Ishiura: “High-level synthesis of side channel attack resistant RSA decryption circuit” in *Proc. Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI 2018)*, pp. 159–160 (Mar. 2018).
3. 大窄直樹, 石浦菜岐佐: “モンゴメリ法及び並列化を適用した耐サイドチャネル攻撃 RSA 復号回路の高位合成,” 電子情報通信学会ソサイエティ大会, A-7-1 (Sept. 2017).

参考文献

- [1] 田邨優人, 高井勇志, 加藤丈和, 松山隆司: “電流波形パターン解析に基づく電気機器の異常検出・劣化診断,” in *Proc. the Institute of Electronics, Information and Communication Engineers (IEICE 2013)*, pp. 135–140 (Nov. 2013).
- [2] S. Nandi, H. A. Toliyat, and X. Li: “Condition monitoring and fault diagnosis of electrical Motors-A review,” *IEEE Transactions on Energy Conversion*, vol. 20, no. 4, pp. 719–729 (Dec. 2005).
- [3] 山本豊樹, 和田和実, 鹿子慎太郎, 樋口暁: 製鉄設備における診断技術の現状と今後 (online), 日鉄住金技報第 402 号 (2015), <https://www.nipponsteel.com/tech/report/nssmc/pdf/402-02.pdf> (accessed 2020-01-08).
- [4] 豊田 利夫: 電流兆候解析 MCSA による電動機駆動回転機の診断技術 (online), https://www.takada.co.jp/~takada-co-jp/giho/pdf/gihou20/kikou_1.pdf (accessed 2019-08-27).
- [5] N. Bessous, S. Sbaa and A. Toumi: “A detailed study of the spectral content in the stator current of asynchronous machines under broken rotor bar faults using MCSA technique,” in *Proc. 2018 6th International Conference on Control Engineering & Information Technology (CEIT 2018)* (July 2018).
- [6] G. Sreedhar Babu, A. Lingamurthy, and A. Seshadri Sekhar: “Condition monitoring of brushless DC motor-based electromechanical linear actuators using motor current signature analysis,” *International Journal of Condition Monitoring*, vol. 1, no. 1, pp. 20–32 (June 2011).
- [7] S. Sarkar, P. Kumar Hembram, P. Purkait, and S. Das: “Acquisition and pre-processing of three phase induction motor stator current signal for fault diagnosis using FPGA, NI Compact-RIO Real Time Controller,” in *Proc. Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON 2016)*, pp. 110–114 (Dec. 2016).
- [8] Renesas Electronics Corporation: BLDC モーターとは? (online), <https://www.renesas.com/jp/ja/support/technical-resources/engineer-school/brushless-dc-motor-01-overview.html> (accessed 2020-01-15).

- [9] 東遼平, 高瀬英希, 高木一義, 高木直史: “プログラマブル SoC のためのシステム設計環境の検討と SW-HW インタフェース生成手法の実装,” 情報処理学会研究報告, 2014-SLDM-164 (Jan. 2014).
- [10] 大西 裕二: “プロセッサ搭載 FPGA による新しいデータ取得システムの開発,” 大阪大学大学院理学研究科物理学専攻修士論文 (Feb. 2019).