車載向けハードリアルタイムシステムにおける性能品質保証

○鈴木 貴広*1　　名越 朱梨*1　　鈴木 雅彦*1　　桑原敦*1　　松尾谷 徹*2

概要

背景: 車載向け組込みソフトウェアでは、高度な信頼性をハードリアルタイム特性で提供する必要がある。エンジン ECU（Electronic Control Unit）などの車載製品では、複数の周期内で複数のタスクが動作し、トルク制御や排ガス制御などの複雑な処理が実装されている。性能品質保証とは、このハードリアルタイム特性を製品が動作する様々な環境で保証するための技術や活動を指す。現在の課題は、機能追加などの派生開発における性能品質保証活動と納期のバランスを取ることである。我々の目指す性能品質保証技術は性能品質を合理的に評価できる方式の構築である。

提案手法と成果: 我々は周期タスクの確実性に対する「ばらつき」を、周期タスクの発行から実行までの遅延時間（ジッター時間）のばらつきと、周期タスクの処理時間のばらつきに分け、動作環境の変化による確率分布の変化としてモデル化した。この方式を実現するためには、μs 単位のジッター時間を計測できる環境と、複数の周期タスクの観測値が混合した時系列の観測値から特性を抽出する技術が必要である。今回はエンジン ECU を事例としたジッター時間の計測と解析について報告する。これらの計測と分析から、派生開発における性能品質保証技術の実現性を確認した。

*1 株式会社デンソー

*2 有限会社デバッグ工学研究所

1　まえがき

車載向け組み込みソフトウェアへの要求性能の一つとして、リアルタイム性が挙げられる。リアルタイム性とは、システム外部からの入力を受けて、それに対応した処理を一定時間内に行う性質のことであるが、エンジン制御など高エネルギー制御のリアルタイム性には特別な制約がある。この特性を区別するため、ハードリアルタイムシステムと呼ばれている [4]。

ハードリアルタイムシステムは、非常に短い時間間隔で、制御対象であるエンジンなどの状態を確認し、燃料や排ガスなどの制御をおこなう。制御そのもののアルゴリズムは、モデルベース開発などが担い、組み込みソフトウェアは、その I/O を確実に行う使命を持っている。制御理論におけるサンプリング制御の実装部分に相当する。

車載ソフトウェアの分野における組み込みソフトウェアの構造は、AUTOSAR（Automotive Open System Architecture）に基づいて設計されている [1]。AUTOSAR は 2003 年に発行され、その後、大幅な機能強化が行われているが、基本的な構造は、アプリケーション層、ランタイム環境（RTE）、および基本ソフトウェア（BSW）という 3 つの主要なレイヤーで構成されている。

ハードリアルタイムの仕組みは、BSW における特別なタスクの実行管理によって機能する。その方式は、周期タスク制御と呼ばれ、小さなタスク群を短い正確な時間周期で、起動と終了を繰り返す特別なループで実現している。周期タスクには、実行の再現性が求められる。再現性とは、同じ入力に対して

同じ結果を返すことだが、実行時間についても再現性が必要である。このハード再現性の要件から、一般的に周期タスクでは再帰処理（Recursion）やスタック操作や動的メモリ管理など処理時間に変動が生ずる手法は禁止されている。

周期タスクの実行管理は 2 種類の制御があり、高優先度周期タスクと低優先度周期タスクである。高優先度周期タスクでは、定められた周期内で起動と終了が完結する。低優先度周期タスクは、プリエンプトと呼ばれる高優先度周期タスクの割込み処理を許すタスク実行方式である。周期タスクの発行から実行までの遅延時間をジッターと呼んでいる。

アプリケーション層では、一般的なタスク処理である POSIX インターフェースが使える。但し、AUTOSAR 4.x(2011 年) 以降をサポートする BSW に限られる。POSIX インターフェースを用いないアプリケーションタスクは、アプリケーション層と BSW の周期タスクとの間で保護機能が無く、同じメモリー空間になる [3]。

ハードリアルタイムシステム向けの周期タスク制御は、市販されている RTOS（Real-Time Operating System）である VxWorks、QNX や OSS である FreeRTOS などでも実装されており、仕様レベルでは μ ITRON でも定義されている。Linux 系の RTOS Linux-RT や携帯電話向けの Android も周期タスク制御を含んでいるが、アプリケーション側の機能を充実させている。ハードリアルタイムで要求される周期タスクの時間制約と、アプリケーション側の機能とのバランスで、RTOS の選択や制御の設定が行われている。

ビジネスの側面において、車載向け組み込みソフトウェアにはリアルタイム性の保証が求められる一方で、例えばエンジンであれば燃費向上や排ガス規制への対応、走行安全や他機器との連携によるセキュリティ対応など、機能の高機能化開発が続いている。一般的に機能の増加は ECU リソースを消費し、リアルタイム性の保証に必要な応答性能を悪化させる。そのため、機能強化と性能維持のバランスを達成するには、要件として求められるリアルタイム時間制約の特性と、対象のシステムが対応できる処理量とリソースとが釣り合うように、設計を行い、その検証を確実に行わなければならない。製品開発における応答時間に関する設計は、新規に機能を開発する局面で負荷状況、一定の時間 (デッドライン) 以内に入力に対して応答の必要があるハードリアルタイムタスクの数、デッドラインの時間を決定し、その要件を達成できるよう設計し評価を行っている [4]。システムとしての検証においては、最も過酷な環境下での動作試験で異常が生じないことを確認し、設計通りのリアルタイム動作を評価している。

製品開発には新規開発に対峙させた概念として派生開発と呼ばれる、一部の機能変更や追加を行う場合も含まれており、派生開発においても性能設計と性能検証が必要である [5]。車載向け組み込みソフト開発の現場では、活動の多くが派生開発によって機能の追加変更に対応しており、本論で対象とする性能課題に対する品質保証は、主に派生開発を対象としている。

派生開発における既存機能への想定外の影響は、デグレード問題として多くの研究が報告がある。リアルタイム特性への影響、特にハードリアルタイム特性に対する検証方法や品質保証について論じた報告は見当たらない。考えられる理由は、対象とする性能のデグレードが数十 μs（10 万分の 1 秒）単位であり、通常のソフトウェア技術では取り扱いが困難であることが考えられる。

従来から行ってきたリアルタイム性能品質保証は、当該の派生開発が完了後において、有識者が最も過酷な環境下を想定して積上げで判断している。また実機環境で計測をしている。この方法で、性能要件を満たさないケースは検出できるが、継続的に行う派生開発の連鎖の中で、リアルタイム性能品質がどのように変化し、どの部分のマージンが劣化しているのか、などの性能品質情報を収集できていない課題がある。

我々が課題としたのは、継続的に続く派生開発の連

鎖のなかで、周期タスクそれぞれのリアルタイム特性に対して、あたかも履歴管理の diff 情報のように変化を把握し、派生開発の要件定義や構造設計段階で、派生製品として致命的な性能リスクを防ぐ性能品質保証の構築である。

この課題を解くための手順を以下に示す。

1. 性能計測における対象のモデル化
2. 対象モデルの係数を算出
3. 派生開発間で生じた変化や異常を数量化
4. 今後の派生開発の性能リスク見積りに活用

2 章では、対象のモデル化について統計モデルを使った方法について示し、3 章では事例において具体的な係数を求め変化や異常を検出する方法について示す。4 章では、モデルの検証について評価し、5 章ではこの方式の活用について示し、6 章でまとめる。

2　提案する性能品質保証

この章では、提案する性能品質保証でハードリアルタイム特性を把握するための統計モデルと、そのモデルの係数を対象製品から計測し分析する方法について論ずる。そのモデルを実際に構築するプロセスについて示し、次章で実際の例とその検証示す。

2.1　性能特性の統計モデル化

ハードリアルタイム特性は、時間の単位で計測できる部分が多々あるが、値に「ばらつき」があるので、統計的な手法が考えられる。統計手法の基本は、分布の型を検定し、型を決することにある。分布の型が同定されれば、その平均や分散を推測し、統計モデルから振る舞いの予測が可能になる。

統計分析を使用し、検査対象に対して取得できたデータから、どの程度の確率で性能品質に関するクライテリアを超える可能性があるのかを算出する。以下本論文で使用した正規分布を使った基礎統計分析の手法と手順についての概要を記載する。

2.2　基礎統計分析手法

組込みシステムのリアルタイム処理は、周期タスクによって決まった複数の周期で実行される。その実行時間の分布を分析する。例えばある i 番目の周期タスク T^i が起動して終わるまでの時間 $t^i_{start-end}$ を測り、値の分布（ヒストグラム）を分析対象とする。

1. ヒストグラム
   対象となるデータについて、ある範囲で区切って階級に分け、階級別に数え上げた数をバーとしてまとめたものであり、データ密度を与える [2]
2. 正規分布
   左右対称で釣り鐘型の曲線で表現できる。その曲線は平均 μ と標準偏差 σ で表現することができる。連続量としての確率密度関数は以下の通り [2]

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \qquad (1)$$

3. 正規 Q-Q プロット
   対象データの平均 μ と標準偏差 σ から算出された正規分布に従う場合の期待値を Y 軸に設定、観測対象データを X 軸に設定したグラフプロットで、直線に並ぶ程度によって、正規分布との適合性が解る [6]。ここでは、飛び値の検出にも用いる。

2.3　提案プロセスの詳細

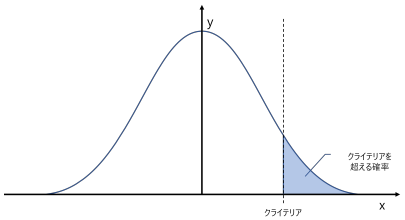
性能品質保証のための技術的なプロセスについて説明する。

1. 測定
   対象データを従来の評価設計に従い計測する。計測後、**2. ヒストグラム確認**へ進む。
2. ヒストグラム確認
   計測したデータの特徴を把握し、対象のデータがどのような分布になっているのかをグラフの形から推測する。推定したグラフが正規分布に

従わない場合には **3．対象範囲の解析**に進み、推定したグラフが正規分布に従う場合は **4．グラフの検定**に進む。もしデータ数が不足している場合には **1．測定**に戻る。

3. 対象範囲の解析
   グラフが正規分布に従わない場合、クライテリアに影響する範囲（ジッター時間が長い箇所）がどのような分布に従うのか解析を行う。正規分布となっている対象の範囲のみをフィルタして抽出する。解析後、再度 **2．ヒストグラム確認**に戻る。
4. 分布の型を推定
   対象データが推定したグラフと言えるか検定を実施する。正規分布に従うかどうかを正規 Q-Q プロットにて検定する。正規 Q-Q プロットが直線になれば、正規分布であると言える。ヒストグラムの確認とグラフの検定をもって正規分布であることを決定する。
5. クライテリアを超える確率を計算
   正規分布に従う場合には、下記計算式で確率を求めることができる。各タスク群における制限時間のクライテリアを x とし、求めた正規分布の特性値から、x を超える確率 Px を求める。算出は確率密度の累積計算から求め、製品ライフサイクルにおいて十分に影響が低いことを確認する。

$$P = \int_{x}^{\infty} f(x)dx \tag{2}$$



3　実証事例

この章では、性能品質保証のもとになる統計モデルに具体的な値を設定する方法について、エンジン制御の組込みソフトを事例に説明する。分析対象は 2 つ、高優先度タスクと低優先度タスクについて示す。

3.1　対象

ハードリアルタイム特性を決定する周期タスクの振る舞いは、設定された周期の繰り返しとしてとらえることができる。周期タスクの周期は複数あり、一番短い周期を T_{min} とすると、長い周期は $N * T_{min}$ と整数倍で設定される。

単純化するため、2 つの種類の周期タスクに着目してその挙動を Figure1:「周期タスクの時間特性」に示す。図中のジッター時間とは、低優先度周期タスクの起動タイミングから、実際に周期タスクが起動するまでの時間を示す。図は低優先度タスクの 1.5 周期分を示している。高優先度タスクと低優先度タスクの周期比率 N は、単純化して 2 として図示している。

Figure1 の最上位にある EVENT タスクは、周期タスクの制御外で、ハード割込みから直接起動されるような特別な処理を示す。短い処理で計測できないが、観測値としては高優先度タスクの実行時間変動に含まれる。

1 つの周期タスクの実行キューには、複数のタスクが登録されている。周期タスクの設計基準は絶対的な再現性（実行時間を含め変化しない）を保っていること、実行時間の変動は、キャッシュの影響などでランダムと考えられる。外乱としての EVENT タスクもランダムに発生することなどから、周期タスクの実行時間分布は正規分布と考える。

3.2　計測環境と計測

計測は HIL（Hardware-In-the-Loop）評価環境で行った。HIL 評価環境は、組み込み開発において非常に重要な手法の一つで、実際のハードウェアを使ってシステムやソフトウェアの動作をシミュレートし、テストするための環境である。計測する時系列データは、BSW の実行制御から信号を計測している。主な項目を示す。

- ソフト上の処理：

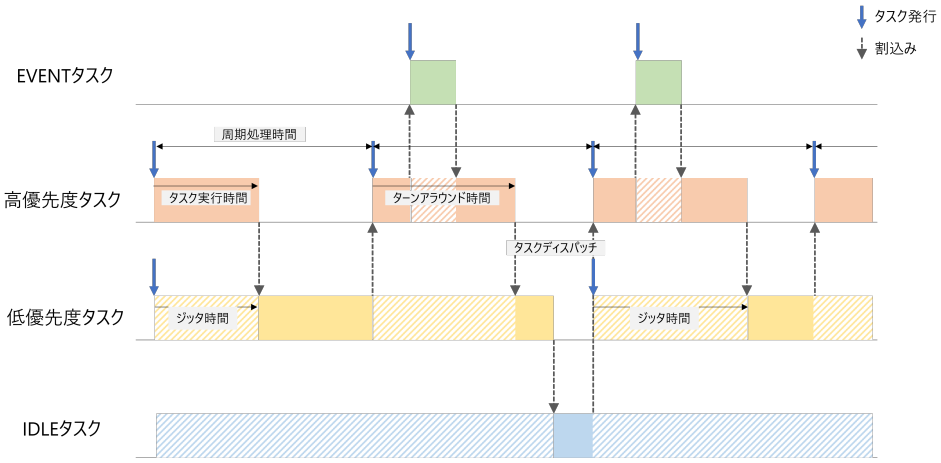


Figure1: 周期タスクの時間特性

タスク起床要求である、ハード割込みからの要求から、実際要求された処理が動き出すまでの時間（タスク起床遅延時間、割込み遅延時間）を計測し、処理が動き出したタイミングでジッター時間計測信号として時間を更新する。

- ハード上の計測処理：
  ジッター計測時間信号を $1\mu s$（100 万分の 1 秒）間隔で計測する。
- 計測 HIL 環境；
  エンジン ECU の利用環境として想定される上限に限りなく近い負荷状況を HIL 環境を使用して作成し、データ計測期間中は環境を固定し、複数の負荷状況で繰り返す。環境の変化による影響を見る場合には、シングルポイントで入力信号を操作し、一因子以外は同一環境とするなど配慮した。

3.3 高優先度周期タスク群

1. 計測
   HIL 環境を使用して高優先度周期タスク群によるジッター時間を計測した。

2. ヒストグラム確認
   まずは高優先度周期タスク群のデータについて確認する。ジッター時間 (μs) が X 軸、出現回数が Y 軸、ヒストグラムの区分け範囲は $1\mu s$ でグラフを示す。Figure 2 のグラフではジッター時間が $0\mu s$ 付近の実測データが大半を占めていることが分かる。このグラフは正規分布ではないが、性能上の問題にはならないので無視する。データを拡大して観測すると $600\mu s$ 近辺にも観測値があり、その範囲のデータがクライテリアを超えるか確認していく。

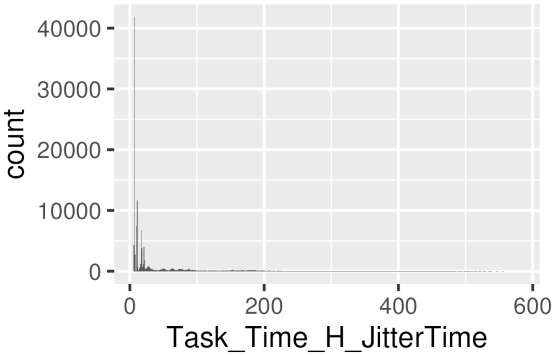


Figure2: 高優先度周期タスク群のヒストグラム

3. 対象範囲の解析
   Figure 3 は対象範囲として 300(μs) 以上のデータのみ抽出したものである。340(μs) 付近を頂点として片側正規分布となっているように見える。

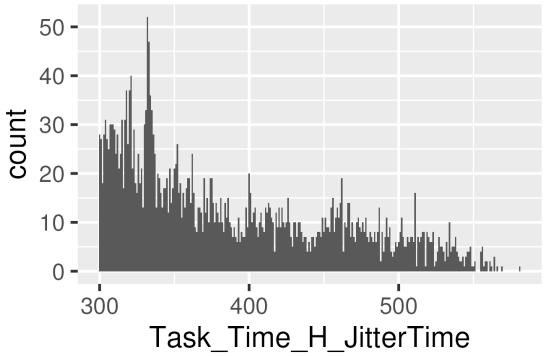


Figure3: 300 μ s 以上の範囲を拡大した図

4. グラフの検定
   対象データについて Q-Q プロットで確認した結果を以下に示す。直線（理論値）と近いところを通っていることが分かる。問題となる 3 σ近辺では、理論分布よりかなり下回っているので、正規分布による検定で問題ないと言える。

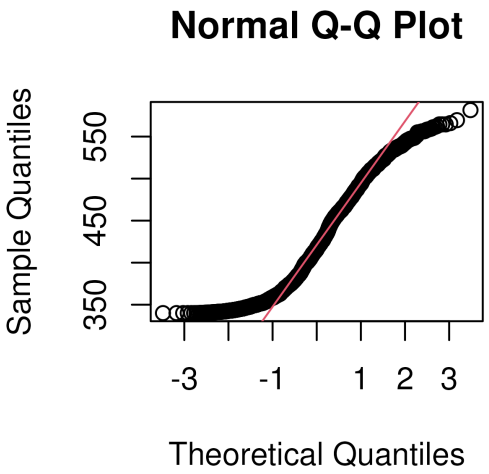


Figure4: 観測値の Q-Q プロット

5. クライテリアを超える確率を計算
   算出したクライテリアを超える確率を求める。
   1 秒間でクライテリアを超える確率 $P_{<1sec}$ は、

$$P_{<1sec} = \frac{1.240e^{-118}}{10[s] * 5[回] * 3[回]} = 4.685e^{-54} \tag{3}$$

   製品ライフサイクルでクライテリアを超える確率は $P_{<1sec}$ にライフサイクル時間を掛ける P_{life} とする。製品ライフサイクルを 10 年（315360000 秒）として、

$$P_{life} = 4.685e^{-54} * 315360000 = 1.477e^{-45} \tag{4}$$

   上記の通り、製品ライフサイクル上でクライテリアを超えることはないと十分に言える。

3.4 低優先度周期タスク群

次にプリエンプト機能を持つ低優先度周期タスク群についてジッター時間（起動）を確認していく。この例は高優先度周期が 1ms の場合である。

1. 計測
   HIL 計測環境を使用して低優先度周期タスク群の実行時間を計測した。
2. ヒストグラム確認
   実行時間 (μs) が X 軸、出現回数が Y 軸、ヒストグラムの区分け範囲は 1(μs) であるグラフは以下の通り。このグラフでもジッター時間の分布は、複数のグループがあり、250(μs) 付近に最大の分布、それ以上で 2 つの分布が観測された。第三のグループ（1000(μs)〜1300(μs)）には、約 1 %の件数（135/14468）が存在している。その範囲のデータについて確認していく。
3. 対象範囲の解析
   Figure 6 に 1000(μs)〜1300(μs) の間の観測値を示す。観測値の 1 %程度であった。この分布について正規分布と仮定した場合、2 σ以上の振舞いについて評価する。1150(μs) 付近を中心とした正規分布になっているように見えるため、Q-Q プロットの検定に進む。

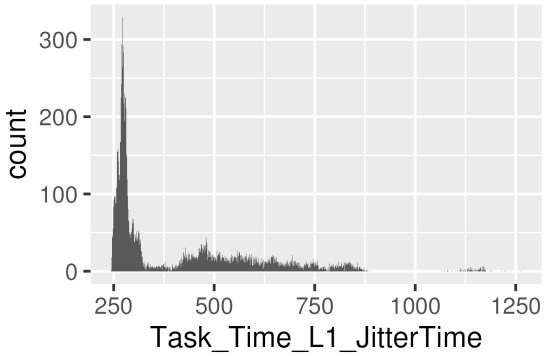


Figure5: 低優先度周期タスク群のヒストグラム

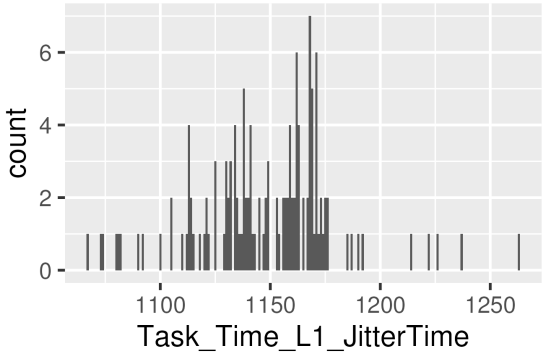


Figure6: ジッターヒストグラムの拡大

4. グラフの検定
   Figure 7 に Q-Q プロットを示す。直線（理論値）の上部の 2 σを超えた部分で理論値より大きな値が観測されている。統計的には外れ値に属する値であり、動作的には 1150(μs) 付近の分布とは異なると想定される。つまり、4 番目の分布が考えられる。

以上、ハードリアルタイムシステムの性能を実現する周期タスクに着目して、性能品質をモデル化する事例を検証と共に示した。2 例とも分布の型が正確に正規分布か否かを利用するのではなく、裾野の広がりを評価するために利用している。低優先度周期タスクの事例では、分割した分布の裾野の部分に着目することで、飛び値が存在することを検出できた。飛び値を分析する統計手法は、極値分布の手法があるが、ここでは飛び値があることを検出し、そ

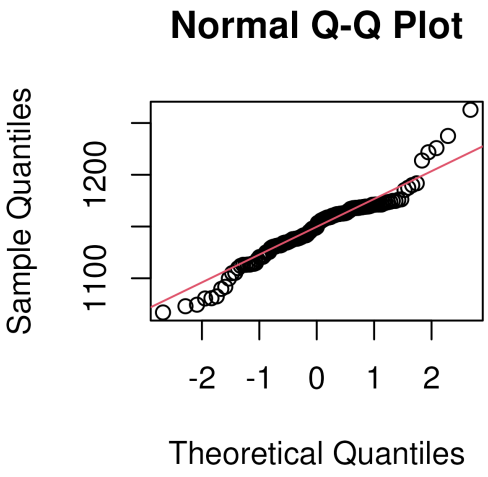


Figure7: Q-Q プロットによる評価

の部分の原因を探るトリガーとした。

4 評価と考察

前章の事例分析から、このモデルについて評価と考察を行う。

4.1 分布の型は正規分布

得られた観測値は、複数の分布の集まりであり、部分に着目すれば正規分布として扱うことができた。手順に従いデータの分布を観測し、性能上、問題になる上限値に対して、モデルの係数を推測すれば、計測した時間データからクライテリアを超える確率を計算し推測が可能と言える。派生開発毎で、データを蓄積すれば、連続的な派生開発リスクについて推測が可能と思われる。

4.2 正規分布に裾野が従わないデータ

観測されたデータをそのまま正規分布のモデルで用いるのではなく、複雑なシステムであり、多種多様な周期タスクが動作している実態に合わせて、分類をする必要があった。データに分類は、最大値近辺で行えば良く、困難な作業ではない。

低優先度周期タスクの事例のように、正規分布の裾

野に従わない場合には、追加の測定や分析が必要となる。方法としては、観測値を増やして統計モデルで進めるか、実際のタスク群の動作を解析して、タスク種を明らかにするか、が考えられる。

統計的な手法としては下記 2 点の方法が考えられる。

- 取得データ数
  ヒストグラムの形状から、データが不足していることが考えられ、追加でデータを取得・再度解析が必要である。取得するデータ数については、t 検定を用いて決定することができる。t 検定とは、2 つの母集団から無作為に抽出したデータの標本平均や標本標準偏差からその 2 つの母平均が等しいかを調べる方法である。取得するデータ数を増やしていき、そのたびに t 検定を適用、母平均から比較した 2 つの母集団が異なるとは言えない状態まで取得していく。そのようにすることで必要なデータ数を決定することができる。
- 正規分布以外の分布
  データ数を増やしてもヒストグラムが正規分布に従わない場合には、別の分布であることを推定する必要がある。そしてその分布に合わせた確率密度関数を導出、設定したクライテリアを超える確率を積分積算で算出することができる。しかし、大部分の観測値は正規分布で示せるので、当該周期タスクとは別のタスク動作によるデータ混入によって、混合型分布の可能性もある。

4.3 検討が必要な項目

- ジッター時間が正規分布にならないことについて
  高優先度周期タスク群と低優先度周期タスク群のジッター時間をヒストグラムにして確認したところ、どちらも山がいくつか存在し、複数の正規分布が含まれる分布となった。これは周期タスク群が複数のタスクで構成されており、タスクの組み合わせが単純ではないと思われる。
- フィルタリング
  また高優先度周期タスク群は最大値に近い箇所でフィルタすると正規分布に近いグラフとなった一方、低優先度周期タスク群は最大値に近い箇所でフィルタしても正規分布にはならなかった。これは、低優先度周期タスク群の方が自身の処理優先度より高優先のタスクをより多くの中断が発生して割り込まれるタスクの組み合わせがより複雑になっていると考えられる。
- 計測条件変更による計測結果の変動について
  今回の事例計測は、あるエンジン制御を例に選んだ。今回の製品事例では、エンジン回転数を 2000 回転に固定して試験実施した。設計上、エンジン回転数を増加させるとそれに伴い EVENT タスクの実行回数が増え、処理負荷が増大すると推測できる。しかしタスク群自体の実行時間はエンジン回転数によって大きく変動する要素がなく、計測条件を高回転に変えても実行時間に大きな影響が出ることはない。それとは対照的にジッター時間は高優先度 EVENT タスクの実行回数が回転数に伴って増加することが考えられる。このような、製品固有の環境要因を考慮して、実際の性能品質モデルは値を決定する必要がある。

5 応用と今後の課題

性能品質保証に用いる統計手法の応用と今後の課題について示す。

5.1 提案手法の有効性

ハードリアルタイム特性は、数十 μs（10 万分の 1 秒）単位の精度が必要であり、人手によるテスト技法やレビュー審査などで品質保証が困難である。そのため、ここで示した HIL 環境を使った実測を基にした手法が有効である。問題は、どんな状況であれば性能品質を達成しているのかを判定する方法であった。その解決策として、統計分布の裾野を Q-Q プロットで示し、判定する方法を提案した。

原理的に、対象とするシステムの実行方式によって、リアルタイム特性は変化する。ここで対象とした、周期タスクは、シングルタスクでの実行であることから、複雑な非線形特性ではないことが明らかになっている。この場合には、この方式が有効で、広い応用が考えられる。

5.2 変動要因に応じたモデル化

計測条件の変動に応じた応答時間のモデル化手法として以下の 2 つ提案する。

1. 机上で割込みされるタスクの種類と最大割込み回数を算出してモデル化する
2. 条件を変えて計測を実施、得られたデータからモデル化する

手法 1 を採用できれば応答時間のクライテリアを守ることができるか設計段階で検討に活用できる。また手法 2 を採用できれば計測データの信頼性を確認することができる。

このとき動作条件は実使用の範囲で検討することが多いが、その範囲を超えても問題なきことを確認するためにも活用可能である。

5.3 市場の流通量を踏まえた確率計算

今回の提案ではある 1 台における、クライテリアを超える確率を計算した。しかし実際には 1 つの製品を年間数万台市場に出荷する。そのため市場に流通するすべてについて問題なきことを導出する必要がある。

市場でクリテリアを超える確率 P_{field} の計算方法を以下に示す。

$$P_{field} = 1\text{台当たりの確率} \times \text{想定総出荷数} \quad (5)$$

5.3.1 ソフト開発現場における統計分析について

本論文では基礎統計分析を使用し、リアルタイムシステムにおける品質保証の手法を検証した。正規分布に従うような単純なデータ群であれば、グラフの検定を通してモデル化が可能であることが分かった。これによりソフトウェアの品質を検定により明らかにすることができると結論付けることができた。

一方、正規分布に従わない複雑なデータ群についてはより詳細な分析とモデル化が必要である。現在の組込みシステムは複雑化しているため、ソフトウェア開発者も専門的な知識が必要となる。ソフトウェアの品質保証のため、組織として専門家を作る必要がある。

5.4 今後対応しうる課題

今回の品質保障の手法はソフトウェア開発後に計測データを統計的に分析するものであった。しかし実際のソフトウェア開発現場では、納入前に品質に問題があることが判明した場合、手戻りが発生し、納期通り納入することが難しくなる。よって今後は追加・変更したロジックが性能にどれだけ影響があるのか、統計的検定モデルで事前に推察することができるように応用することも考えられる。

6 まとめ

本論文では、車載 ECU ソフトウェアにおけるリアルタイム性の性能品質保証の手段として、HIL 環境での測定と統計的分析の適用について検討を行った。正規分布に従うような単純なデータ群であれば、グラフの検定を通してモデル化が可能であることが分かった。これによりソフトウェアの性能品質を検定により明らかにすることができると結論付けることができた。

引用文献

[1] *AUTOSAR*. https://www.autosar.org/. (Accessed on 03/07/2024).

[2] David M. Diez, Christopher D. Barr, and 訳国友直人 Cetinkaya-Rundel Mine. **日本語訳** *OpenIntro statistics*. Vol. 4. OpenIntro Boston, MA, USA: 2012. URL: http://www.kunitomo-lab.sakura.ne.jp/2021-3-3Open(S).pdf.

[3] 中島達夫. “(第 5 回) 組込みオペレーティングシステム概論”. In: **映像情報メディア学会誌** 63.5 (2009), pp. 633–637. URL: https://www.jstage.jst.go.jp/article/itej/63/5/63_5_633/_article/-char/ja/.

[4] 南角茂樹. “組込みシステムにおける応答性能の保証 (組込みシステム特集号)”. In: **システム/制御/情報** 51.9 (2007), pp. 388–392. URL: https://www.jstage.jst.go.jp/article/isciesci/51/9/51_KJ00004684694/_article/-char/ja/.

[5] 清水吉男. **派生開発** *AFFORDD 2021*. ja. URL: https://affordd.jp/derivative-development/.

[6] **統計** *Web Q-Q* **プロット**. ja. URL: https://bellcurve.jp/statistics/glossary/2071.html.