O-032

実時間制御システムのための設計モデルとその実現

A Design Model for Real-Time Control Systems and Its Implementation

小山 恭平† **島川 博光**†
Kyohei Koyama Hiromitsu Shimakawa

１．はじめに

下水処理プラントやトンネル換気制御システムなどの実時間制御システムに対して，新たな要求が起こっている．現在，実時間制御システムの開発は非効率であると言わざるをえない．そこで，効率的な実時間制御システムの開発を行うための開発環境が必要とされている．

また，実時間制御システムには，我々の生活に密接に関連するものが多く，より安全に制御を行い，多くの人々の生活を守らなければならないという社会的義務も大きい．そこで，実時間制御システムにおける安全システムの要求も高まってきている．制御システムでは，正確な制御を行っていてもトンネル火災などのような災害が起こりえる．そのような例外的なことが起こっても，被害を最小限に抑え，また事前に予防することができる安全システムが必要である．

このような要求に応えるべく，本論文では実時間制御システムのための設計モデルと，そのモデルに基づいた開発を可能にするための開発・実行環境を提案する．

２．実時間制御システムへの要求

2.1 実時間制御システムの開発効率化

実時間制御システムの構成は図１に示されるような構成をしている．下水処理プラントやトンネルなどの制御対象に設置されているセンサからデータが獲得され，情報システムはそのデータを基に演算を行い，制御命令を導き出す．このとき，制御命令を導き出すにあたっては，厳しい実時間制約が課せられる．

このような実時間制御システムの開発現場では，実時間の専門家と制御の専門家が連携を取って開発を行っている．しかし，このように２つ以上の専門家が連携を取って開発を行うことは煩雑であり，開発効率は非常に悪い．このような問題を解決するには，実時間制御システムを効率良く開発することを支援する環境や設計モデルが必要である[1].

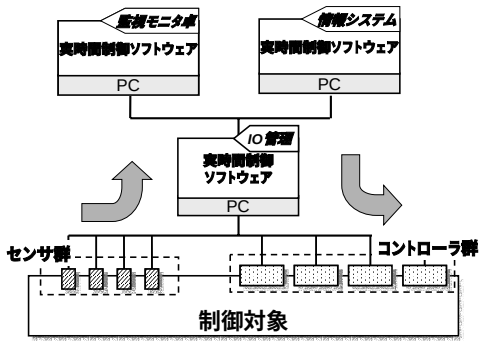

図１：実時間制御システムの構成

† 立命館大学 大学院 理工学研究科

2.2 監視システムへの新たな要求

図１に示したように，実時間制御システムの中には監視モニタ卓が存在する．これは，獲得されたデータを基に制御対象の状況を表示し，監視員に状況を伝えるものである．トンネル火災などの例外的なことが起こったときには，監視員は監視モニタ卓から得られる情報を基に最適な処理方法を判断する．このとき，監視モニタ卓に必要とされるのは，より早くより正確に監視員へ現在の状況を伝えることである．そのため，監視モニタ卓にはセンサから獲得された数値だけを表示するのではなく，アニメーションなどを用いて監視員に直感的に訴える機能が必要となる．また，過去に起こった状況を即座に参照し，処置方法の判断材料にするということも有効であると考えられる．さらに，過去と現在のデータから未来の状態を予測することができれば，例外的な事象が起きることを抑制することも可能となる．このように過去，現在，未来のデータをアニメーションとしてシームレスに監視モニタ卓に表示し，それを監視員が任意のタイミングで再生，巻戻し，早送り，コマ送りなどができれば，状況分析などにも非常に有効となる．

2.3 目標

上記のような実時間制御システムへの要求を満たすために，新たな設計モデルが必要となる．まず，制御システムが，周期的動作を保証するための RMA(Rate Monotonic Analysis)[2]を適用できるように構築されなければならない．普通，RMA の適用を行うには実時間の専門知識が必要となるが，それなしでも RMA を適用可能な形に実時間制御システムの開発を導く設計モデルがあれば，より効率的なシステム開発が可能となる．また，2.2 で述べたような時間的自由度の高い監視画面を実現するには，監視画面表示用タスクが任意のタイミングでデータを獲得できる必要がある．過去のデータをデータベースから獲得する場合は問題ないが，実時間制約の厳しい現在のデータを，各タスクが任意のタイミングで獲得する場合には新たな設計モデルが必要となる．

３．パントリ・コック・モデル

3.1 パントリ・コック・モデルの概要

我々は実時間制御システムの設計モデルとして，パントリ・コック・モデル(Pantry-Cook Model)を提案する．図２に示すように，このモデルは実時間制御システム内でのデータの流れを，厨房における食材の流れになぞらえたものである．

料理の注文が伝えられたとき，料理人(Cook)は必要な食材を食材貯蔵庫(Pantry)に自ら取りにいき，調理を行い，料理を完成させる．このとき，料理人は「食材調達，調理，料理完成」の一連の流れを独立に行う．また，注文内容が変更になっても調理方法，または料理人を変更するだけで，食材貯蔵庫の構造を変更する必要はない．

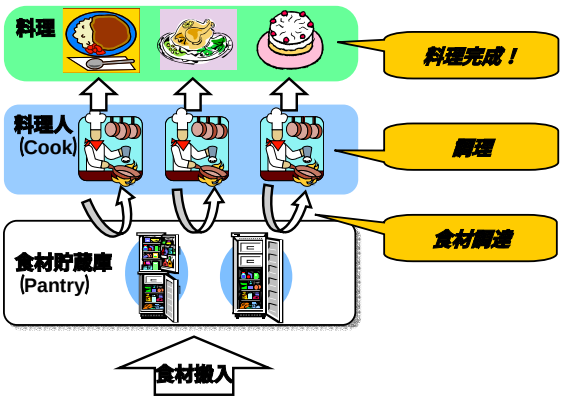

図２：パントリ・コック・モデル

実時間制御システムでのデータの流れを食材の流れと同様に考える．制御命令の導出や監視画面の作成を行っている制御タスク群は，センサから獲得されたデータを用いる．図３に示すように，センサから獲得されたデータがデータ格納庫に収められている．制御タスクは必要なデータをデータ格納庫から獲得し，演算を行い，制御命令を導き出す．このとき各タスクは「データ獲得，演算，制御命令送出」の一連の流れをそれぞれ独立に行う．

このパントリ・コック・モデルに基づいて実時間制御システムを設計する利点は大きく３つある．

まず，１つ目に実時間制御システムの構成を固定基盤と対象依存部に分けられることである．実時間制御システムにおいて固定基盤をくくり出すことができれば，実時間制御システムの開発者は対象依存部のみを開発すればよく，開発効率は飛躍的に向上する．

２つ目に，各タスクの独立性を確保できることが挙げられる． RMA では，各タスクの最悪計算時間が計測できる必要がある．よって，各タスクが独立であるか，タスク間の相互干渉による閉塞時間の最悪値が見積もれる必要がある．図３におけるデータ格納庫は制御タスク群により共有されるメモリ空間であり，制御タスク間の相互干渉はここへのアクセス・メソッドに限定される．データ格納庫は固定基盤であり，それへのアクセス・メソッドは不変であるので，このメソッドにより引き起こされる閉塞時間の最悪値は見積もり可能である．

３つ目の利点としては，各タスクが任意のタイミングで実時間性を保証しながらデータを獲得できることである．よって，時間的自由度の高い監視画面を作成することが可能となる．これは，各タスクが独立であり，タスク自らデータを獲得するというモデルによってもたらされる利点である．

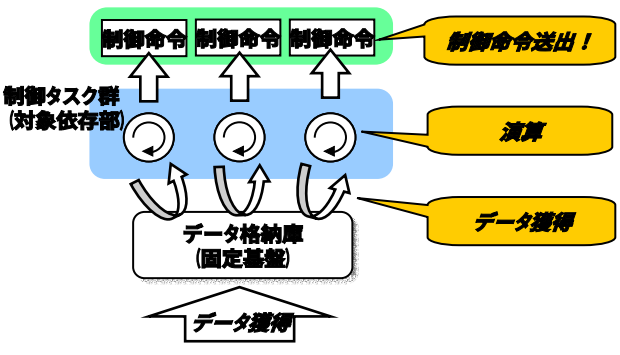

図３：パントリ・コック・モデルによるシステム設計

3.2 固定基盤：RT-Platform

実時間制御システムを開発する上での固定基盤として RT-Platform を提供する．RT-Platform はパントリ・コック・モデルに基づいた実時間制御システムのための開発・実行環境である．そのため，RT-Platform を適用すれば，システム開発者は対象依存部，すなわち，制御タスク群の開発のみに専念でき，システムの開発効率は飛躍的に向上する．また，RT-Platform には，RMA による実時間性の保証を行う機構も設けられており，システム開発者は，実時間の専門知識なしでもシステムの開発が可能となる．

RT-Platform を適用した実時間制御システムの構成を図４に示す．RT-Platform を全てのクライアントに導入することにより，各クライアントの実時間性を保証することができる．結果としてシステム全体の実時間性を保証することができる．

3.3 対象依存部の開発

既に述べたように，RT-Platform を導入すれば，システム開発者は対象依存部の開発のみに専念することができる．また，対象依存部の開発は RT-Platform の導入により簡単化される．センサから獲得されたデータは RT-Platform に蓄えられている．よって，システムの開発者は RT-Platform から必要なデータを取得し，望む出力データに加工するという流れで制御タスク群を作成する．これは一方向のデータ加工処理であり，きわめて簡単である．

また，固定基盤である RT-Platform は，制御対象が変更になっても変更を加える必要はない．これを実現するために，センサから獲得されたデータの表現型として XML を導入し，固定基盤と対象依存部のインタフェースとする．通常，実時間制御システムでは，センサから獲得されるデータ型は制御対象に依存する．制御対象の相違によりデータ型が変わり，プログラムの変更を余儀なくされ，開発効率が低下するという問題が現実に存在した．しかし，XML の導入で制御対象の相違によるデータ型の違いを最小限に抑えることができ，より汎用性の高い固定基盤となる．

4．RT-Platform の構成要素と設計

4.1 RT-Platform の設計とデータの流れ

今回，我々は RT-Platform のプロトタイプを RT-Linux[3] 上で作成した．RT-Platform 内部設計の概要を図５に示す．

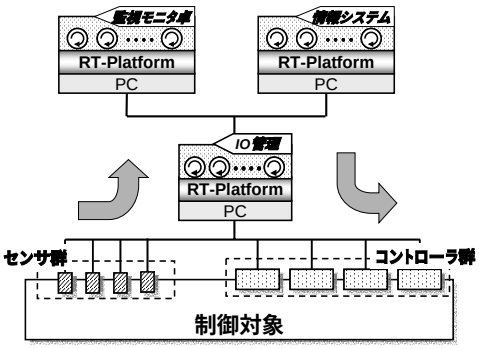

図４：RT-Platform を適用した実時間制御システム

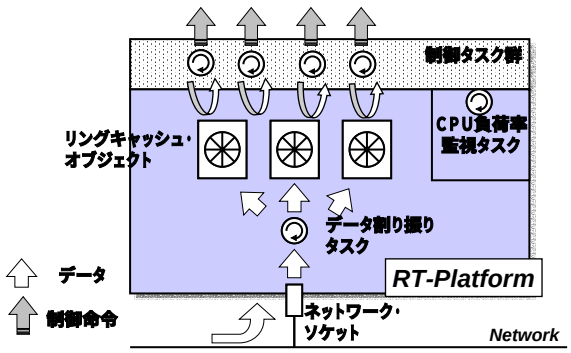

図５：RT-Platform の構成

センサから獲得されたデータはネットワークを通じて RT-Platform のネットワーク・ソケットに到着する．そのデータはデータ割り振りタスクによってリングキャッシュ･オブジェクトに割り振られる．ここでリングキャッシュ・オブジェクトとは，論理的にリング状の形をしたメモリ空間とそれへの書き込み，読み出しの機構を隠蔽化した部品である．リングキャッシュ・オブジェクトはデータの周期毎に設けられている．RT-Platform の上に作成されている制御タスク群は，このリングキャッシュ・オブジェクトから必要なデータを各タスク任意のタイミングで獲得し，制御命令を導き出す．

4.2 RT-Platform の構成要素

RT-Platform 構成要素の中で最重要なものとして，リングキャッシュ・オブジェクトと CPU 負荷率監視タスクが挙げられる．

リングキャッシュ・オブジェクトとは，既に述べたように，論理的にリング状の形をしたメモリ空間とそれに付随する機構を隠蔽化した部品である．この部品の中には，RMA の適用を考慮した排他制御のための機構[4]が含まれている．実時間制御システムの開発過程で排他制御の部分は難解であるとされている．実時間制御システムの開発者はメモリ空間の設計を行う必要はなく，この部品をいくつ配置するかを決め，それを配置するだけでよい．そのため，システムの開発効率は向上する．

CPU 負荷率監視タスクは，実時間性の保証を行うための機構である．このタスクは RT-Platform 内とその上の全タスクを常に監視し，各タスクの最悪計算時間を元に RMA の演算を行う．この CPU 負荷率監視タスクにより，タスクの動的な追加や変更を，実時間性を保証しながら実行することが可能となる．

5．他モデルとの比較

5.1 Publish-Subscribe パターン

システムの設計を行うにあたり，デザインパターンを適用することがある．デザインパターンの１つに Publish-Subscribe パターン，別名 Observer パターンがある[5]．Publish-Subscribe パターンでは，Publish（出版者）役が送信するデータを，Subscribe（購読者）役が受動的にデータの到着を待ち受けることになる．もし，このパターンを制御システムの構成に適用するならば，データ格納庫が Publish 役，制御タスク群が Subscribe 役ということになり，データ格納庫が指定された制御タスクにデータを送信するという形になる．しかしながら，制御タスクが実行周期を動的に変動させたい場合には，Subscribe 役が，能動的にデータを取りに行く，すなわち，データ獲得の決定権を制御タスクに持たせる方が好都合である．

5.2 シミュレータによる実時間制御システム開発

実時間制御システムを開発するにあたり，非効率なシステム開発という問題点を解決するべく，シミュレータベースによる実時間制御システムの開発環境が研究されている[1][6]．これは，GUI(Graphical User Interface)を用いて，制御システムに必要な部品を組み合わせていくことにより，設計を行うものである．また，設計を行った部品の組み合わせをプログラムソースに変換する機構も設けられている．現在，Matlab/Simulink[7]などのようなシミュレータを用いた実時間性を意識しない制御システムの開発が，制御の専門家の間では広く使われている．よって，シミュレータを用いたシステム開発を，実時間性を意識したものに拡張することの意義は大きい．しかし，シミュレータベースによる実時間制御システムの開発では，任意の場合での実時間性の検証はできても，実時間性の保証はできない．

6．おわりに

本論文では，実時間制御システムの新たな設計モデルであるパントリ・コック・モデルと，そのモデルに基づいた開発・実行環境である RT-Platform を提案した．この設計モデルならびに開発・実行環境は実時間制御システムに対する新たな要求に対応するためのものである．

現在，RT-Platform は試作段階であり，今後さらに実装を進め，その有用性を検証していく予定である．これを発展させ、我々は過去，現在，未来のデータをシームレスに統合し，時間的自由度の高い監視モニタ卓を作成する．

参考文献

[1] Johan Eker，Anton Cervin：“A Matlab Toolbox for Real-Time and Control Systems Co-Design”，Proc. of 6th International Conference on Real-Time Computing Systems and Applications pp.320-327（1999）

[2] 白川 洋充，竹垣 盛一：“リアルタイムシステムとその応用”，朝倉書店 システム制御情報ライブラリ（2001）

[3] FSMLabs-The RTLinux Company, http://www.fsmlabs.com/

[4] 島川 博光，水沼 一郎，竹垣 盛一：“プラント履歴データの実時間獲得・提供システム”，電子情報通信学会論文誌，Vol.J78-D-Ⅰ No.8，pp.798-806（1995）

[5] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides, “Design Patterns”, Addison Wesley Longman, （1995），本位田 真一，吉田 和樹 監訳：“オブジェクト指向における再利用のためのデザインパターン 改訂版”，ソフトバンク パブリッシング（1999）

[6] G.Quaranta, P.Mantegazza ： “ USING MATLAB-SIMULINK RTW TO BUILD REAL TIME CONTROL APPLICATIONS IN USER SPACE WITH RTAI-LXRT”，Proc. of Real time Linux Workshop（2001）

[7] Matlab ホームページ，http://www.cybernet.co.jp/matlab/