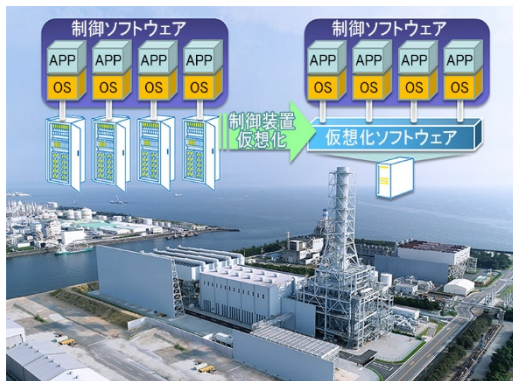


プラント制御装置への仮想化技術の適用

Application of Virtualization Technology to Plant Controllers



川嶋 大樹*¹
Hiroki Kawashima

舟橋 聡*¹
Satoru Funahashi

中出 実*²
Minoru Nakaide

戸田 信一*³
Shinichi Toda

プラント制御システムは、制御する対象や機能ごとに制御装置を分けて制御する分散構成であるため、装置を減らすことによるコスト削減が難しい。三菱重工業株式会社(以下、当社)では、その課題を解決するため IT システムで普及が進んでいる仮想化技術をプラント制御装置へ適用することを進めている。仮想化技術を適用することで、一台の物理ハードウェア上に複数の制御装置を集約して実装することができるため、大幅なコスト削減とともに、制御装置自体のエネルギー消費低減、CO₂ 排出量低減が期待できる。今回、仮想化した制御装置を実用化すべく、制御装置に必要な①リアルタイム性の向上と②冗長化構成における切替え手法の開発を行った。制御周期を 50msec とした試作機において評価を行った結果、プラント制御装置に要求されるリアルタイム性と可用性を満たすことを確認したので本報にて報告する。

1. はじめに

プラント制御システムは、[図1](#)に示すように制御する対象や機能ごとに制御装置を分けて制御する分散構成であるため、制御装置を減らすことによるコスト削減は難しい。加えて、制御装置はハードウェアに合わせてソフトウェアの開発を行っているため、制御装置を構成する電子部品の生産中止等に伴い仕様の異なるハードウェアを用いる際には、都度ソフトウェアの更新・改修が必要となり、保守コストがかかっている。これらの課題を解決し、設備・開発の合理化により CO₂ 排出量低減に寄与するため、IT システム分野で普及が進んでいる仮想化技術に注目し、プラント制御装置への仮想化技術の適用に取り組んでいる。

本報では、プラント制御装置へ仮想化技術を適用することによる利点と課題について説明し、その課題に対する解決策であるリアルタイム性の向上と冗長化構成の切替え手法について述べ、解決策ごとに有効性を評価した結果を示す。

*1 デジタルイノベーション本部 CIS 部

*2 デジタルイノベーション本部 CIS 部 主席技師 技術士(情報工学部門)

*3 株式会社 MHI パワーコントロールシステムズ Netmation 開発室 Netmation 開発部 主席技師

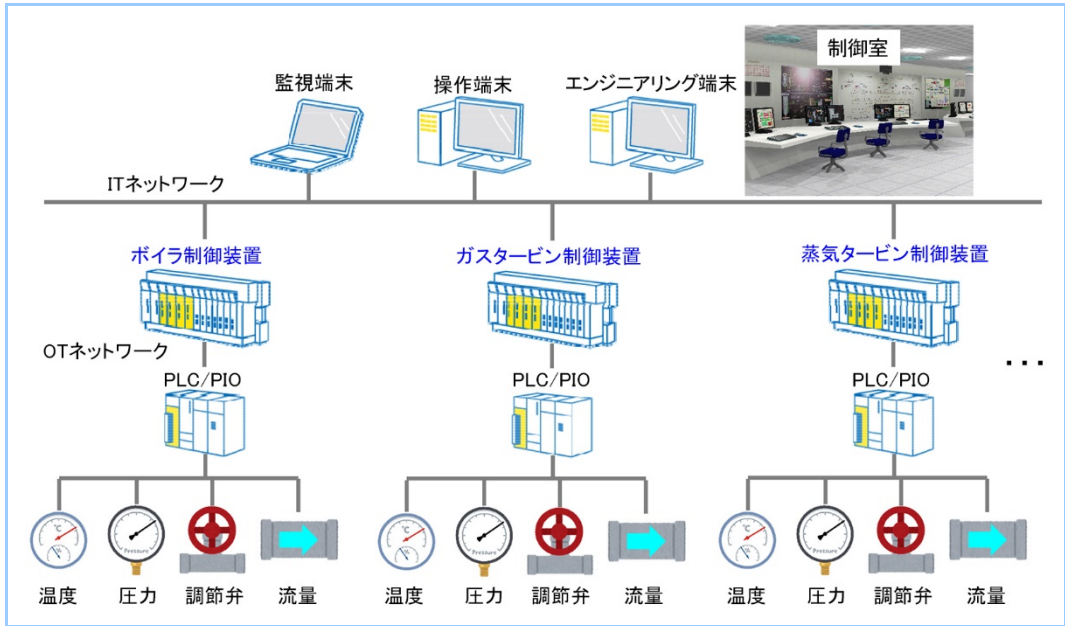


図1 プラントにおける分散型制御システムの構成例

2. 仮想化技術の制御装置への適用

2.1 仮想化技術とは

仮想化技術とは、物理的なハードウェアをソフトウェアにより抽象化し、物理リソース(CPU、メモリ、ハードディスク、ネットワーク等)を論理リソースとして扱うことで、物理的な制限にとらわれず、ソフトウェア的に統合・分割等のリソース制御を可能にする技術である。この技術により、一台の物理ハードウェア上で複数の仮想的な計算機を動作させることで、ハードウェアリソースを無駄なく有効活用でき、コストの削減が可能となることから、クラウドサービスをはじめとするITシステムで普及が進んでいる。ITシステムにおける仮想化技術の適用例を図2に示す。

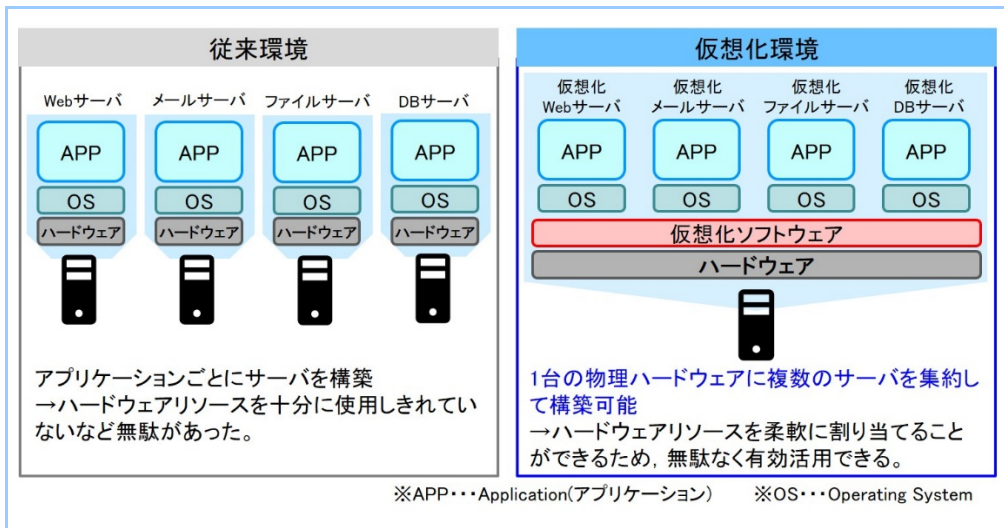


図2 ITシステムにおける仮想化技術の適用例

2.2 仮想化技術をプラント制御装置へ適用することによる効果

仮想化技術を制御装置へ適用した場合の構成図を図3に示す。制御装置の集約によるコスト削減のメリットが得られることに加えて、ハードウェアを抽象化して扱うため、ハードウェアの変更が生じたとしてもその上で動作するソフトウェアが認識するハードウェア情報は変化しない。そのため、電子部品の生産中止等に伴ってハードウェア改版・変更となっても、ソフトウェアは変更不要で動作するため、ソフトウェアの更新・改修コストの削減が可能となる。

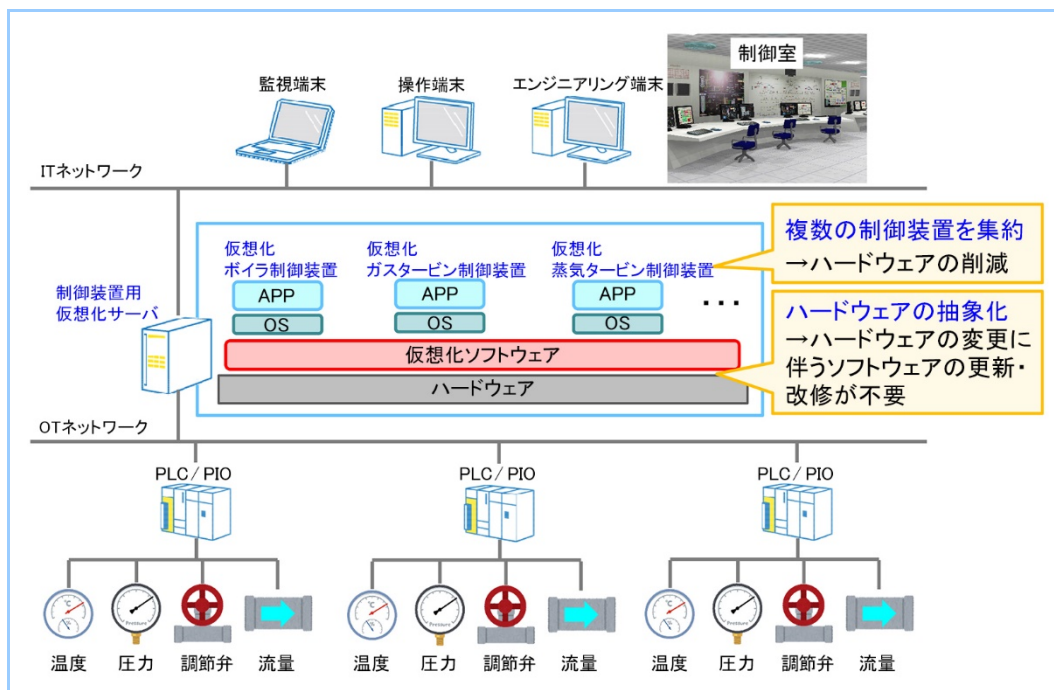


図3 仮想化技術を制御装置へ適用した場合の構成例

2.3 仮想化したプラント制御装置の実用化に向けた課題

仮想化技術の適用による利点がある一方で、実用化するためには仮想化した制御装置（以下、仮想化制御装置）においても、従来通りの性能とシステムが停止することなく稼働し続ける能力（可用性）が必要である。

プラント制御装置は、制御対象を安全に制御するため、リアルタイムに制御信号を送信する必要がある。しかし、制御装置へ仮想化技術を適用すると、仮想化特有の処理（ハードウェアエミュレータや仮想化ソフトウェアの処理）が発生し、リアルタイム性の低下が予想される。

また、プラント制御装置は高い可用性が必要なため、“稼働側”と“待機側”の2重系で構成された冗長化構成を採っている。仮想化した場合においても冗長化が必要であり、異常時には稼働側から待機側に速やかに切り替わり、正常な制御信号を制御対象へ出力することが求められる。しかし、仮想化制御装置における系統切替え手法の開発と切替え時間の性能評価ができていない。

そこで我々は、仮想化制御装置の実用化を目指し、①仮想化制御装置のリアルタイム性向上と②冗長化構成における系統切替え手法の開発に取り組んだ。

3. 仮想化制御装置の試作

仮想化ソフトウェア KVM (Kernel-based Virtual Machine)⁽¹⁾を用いて物理ハードウェア1台に仮想化制御装置を2台構築し、物理ハードウェア2台を用いて二重化した冗長化構成を構築した。構成図と環境を図4、表1に示す。初期状態は、物理ハードウェア1側の仮想化装置（稼働側）が稼働し、物理ハードウェア1側で異常が発生した場合に物理ハードウェア2側の仮想化装置（待機側）に切り替えることを想定している。

また、実際の制御システム構成に基づいて、制御装置と制御対象間で制御信号等のデータをやり取りすることを目的とした OT (Operational Technology) ネットワークと、制御装置とエンジニアリング端末間で運転員が監視や管理を行うために必要なデータをやり取りすることを目的とした IT ネットワークを構築した。仮想化制御装置は、制御信号送信プログラムを実行し、制御対象へ定周期で制御信号を送信する。本評価ではプラント制御装置を想定し、送信周期を 50msec とした。

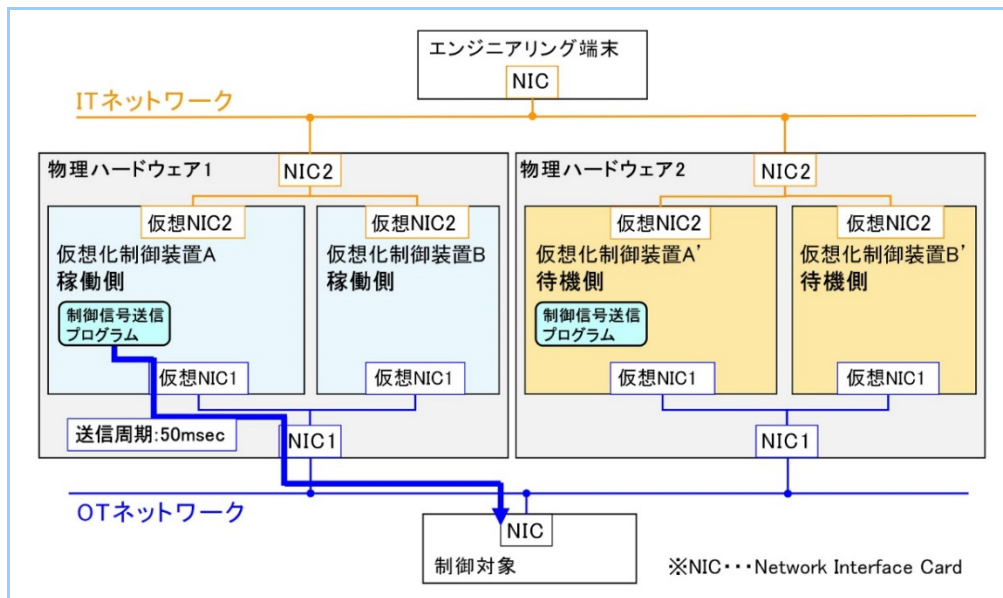


図4 冗長化した仮想化制御装置の評価構成

表1 評価環境

システム	項目	情報・バージョン
物理ハードウェア	OS	Ubuntu 18.04 64bit (Linux Kernel 5.4.54)
	CPU	Core i5-9500vPro 3GHz
	MEM	8Gbyte
	PREEMPT-RT patch	5.4.54-rt33
	仮想化ソフトウェア (ハイパーバイザ)	KVM (Kernel Virtual Machine)
仮想化制御装置	OS	Ubuntu 14.04 32bit (Linux Kernel 3.12.74)
	CPU コア数	2 コア
	MEM	2Gbyte
	PREEMPT-RT patch	3.12.74-rt99

4. 仮想化制御装置のリアルタイム性向上

4.1 リアルタイム性向上手法の適用

仮想化制御装置のリアルタイム性を向上させる手法として、参考文献⁽²⁾に記載の手法である以下(a)~(c)の手法を適用し、リアルタイム性の向上を図った。

(a) 排他制御の変更によるタスク切替えオーバーヘッド低減

リアルタイム性を確保するためには、実行されているタスクよりも優先度が高いタスクが実行される際、遅延なく速やかに実行されることが必要である。通常のカーネルでは排他制御を担うスピロックによりプリエンプション禁止時間(オーバーヘッド)が発生する。そこで、カーネルにPREEMPT_RT patch⁽³⁾を適用することでプリエンプティブルな排他制御を行い、タスク切替え時のオーバーヘッド低減を行った。

(b) ハードウェアエミュレータ(QEMU)のスケジューリング優先度変更による処理遅延低減

仮想化制御装置はCPUや通信インタフェースなどのハードウェアリソースにアクセスする際、ホスト計算機上で動作するハードウェアエミュレータ(QEMU)を介して処理する。ハードウェアエミュレータの実行に待ち時間が発生すると仮想化制御装置の処理も遅延する。そこで、ハードウェアエミュレータの優先度を、リアルタイム性が要求されない他タスクよりも高優先度に設定し、待ち時間の低減を図った。

(c) CPU コアの静的割付けによるリソース競合回避

仮想化制御装置が CPU にアクセスする際と同じ CPU コアを他が使用した場合(競合)、待

機による処理遅延が発生する。そこで、各仮想化制御装置が使用する CPU コアを別々に割り付けることで競合を回避し、処理遅延を解消した。

4.2 仮想化制御装置のリアルタイム性評価

4.2.1 評価方法

一般にリアルタイム性とは、ある制約時間以内に処理が実行できる性質のことである。当社のプラント制御装置では、制御対象が制御装置からの制御信号の受信間隔を計測し、受信間隔が制約時間(制御周期の2倍。本報では、100msec)を超えた場合に制御信号が途絶したと判断し、異常として検知する。そこで本評価では、制御対象で定周期制御信号の受信間隔を計測し、図5に示すように仮想化制御装置の送信周期との差を“受信遅延”として評価した。また計測時は、システム負荷をかけるためファイル転送・書き込みの処理を印加した。

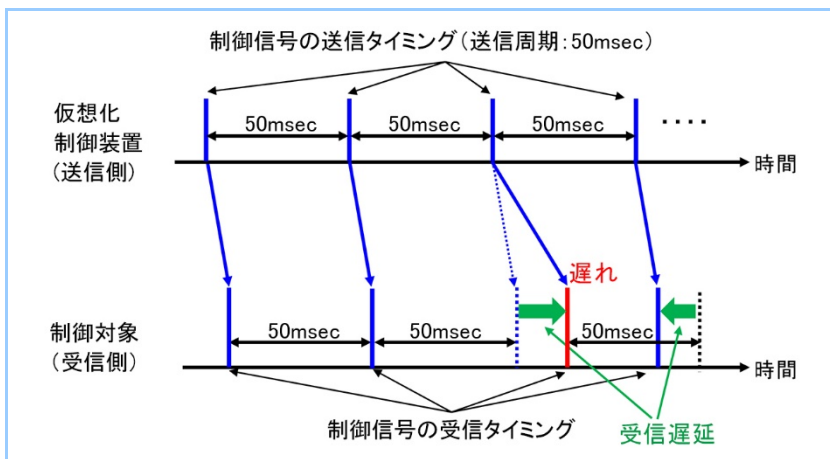


図5 受信遅延

4.2.2 評価結果

制御信号を 144 万回(連続 20 時間)制御対象に送信し、リアルタイム性向上手法の適用前後における受信遅延を計測した。計測の結果を図6に示す。リアルタイム性向上手法を適用することで受信遅延の平均値, 最大値, ばらつきが大幅に減少し、仮想化制御装置のリアルタイム性が向上することを確認した。

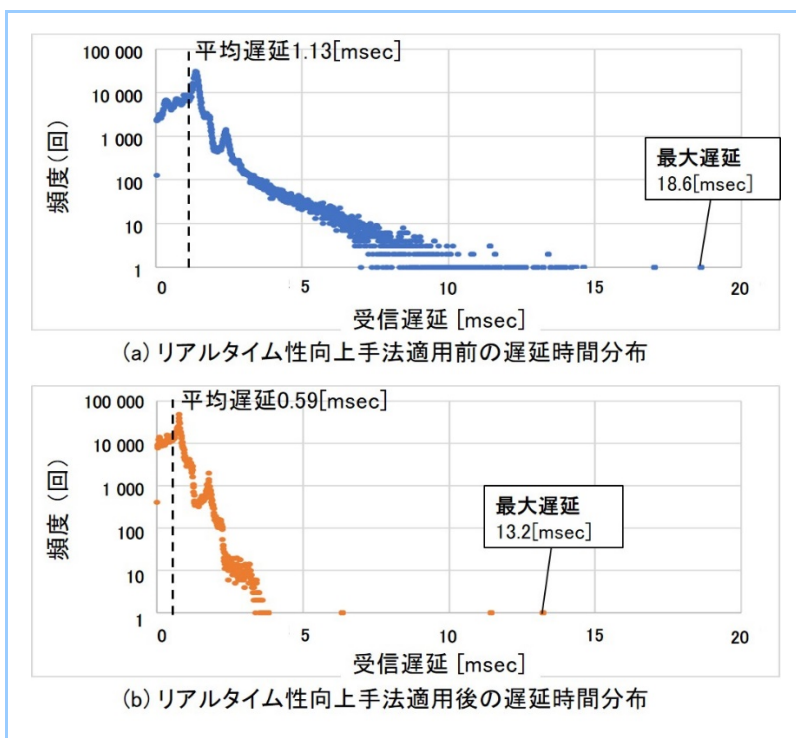


図6 受信遅延分布 144 万回測定(20 時間)

5. 仮想化制御装置の冗長化構成における系統切替え手法の開発

5.1 系統切替え手法

冗長化構成において異常発生時に待機側へ自動的に切り替えるためには、システムが異常発生を自動で検知し切り替える必要がある。系統切替え手法として、二重化した仮想化制御装置が確認信号と呼ばれる一定周期の信号を互いに送り合い、相手の確認信号を受信できなくなった場合に相手が停止したと検知し、自分が稼働状態に切り替わる方法を採用した。図7に切替え手法の概要図を示す。

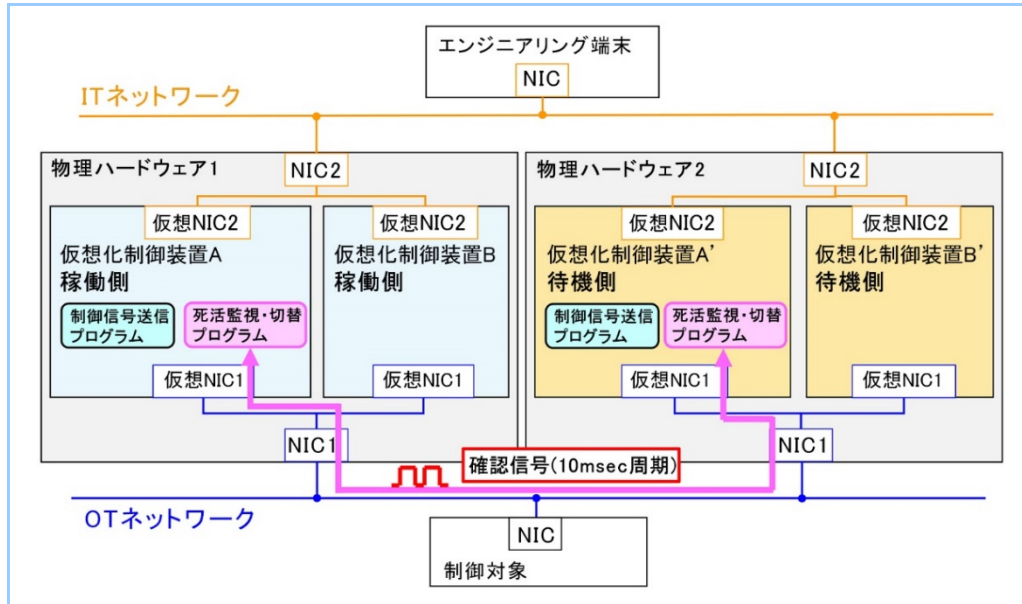


図7 仮想化制御装置の冗長化構成における系統切替え手法

系統切替え手法の手順は以下の通りである。

- ・ 初期状態は物理ハードウェア1の仮想化制御装置Aを稼働側、物理ハードウェア2を待機側とする。
- ・ 稼働側からのみ、制御信号を送信する。
- ・ 稼働側と待機側の仮想化制御装置は、10msec周期で確認信号をお互いにOTネットワークを介して送受信する。
- ・ 待機側が自分の確認信号を3回送信する間に稼働側の確認信号を受信しない場合、待機側から稼働へ切替え、制御信号の送信を開始する。

市販の冗長化サーバには、死活監視用の確認信号を専用線やITネットワークでやり取りするものがある。一方、プラント制御では制御対象に誤った信号を送信しないことが重要である。ネットワークの断線が発生すると、二重化した仮想化制御装置の両方が稼働状態となって制御信号を出力することとなる。専用線やITネットワークを用いる場合、制御対象は両方の制御信号を受信することになり誤動作するリスクが発生する(図8(a))。

そこで、本開発では、OTネットワークを介して確認信号をやり取りする方法(図8(b))とした。この場合、ネットワークの断線が発生し、二重化した仮想化制御装置の両方が稼働状態になったとしても両方の制御信号が制御対象へ到達することはないので、制御対象の誤動作を回避することができる。

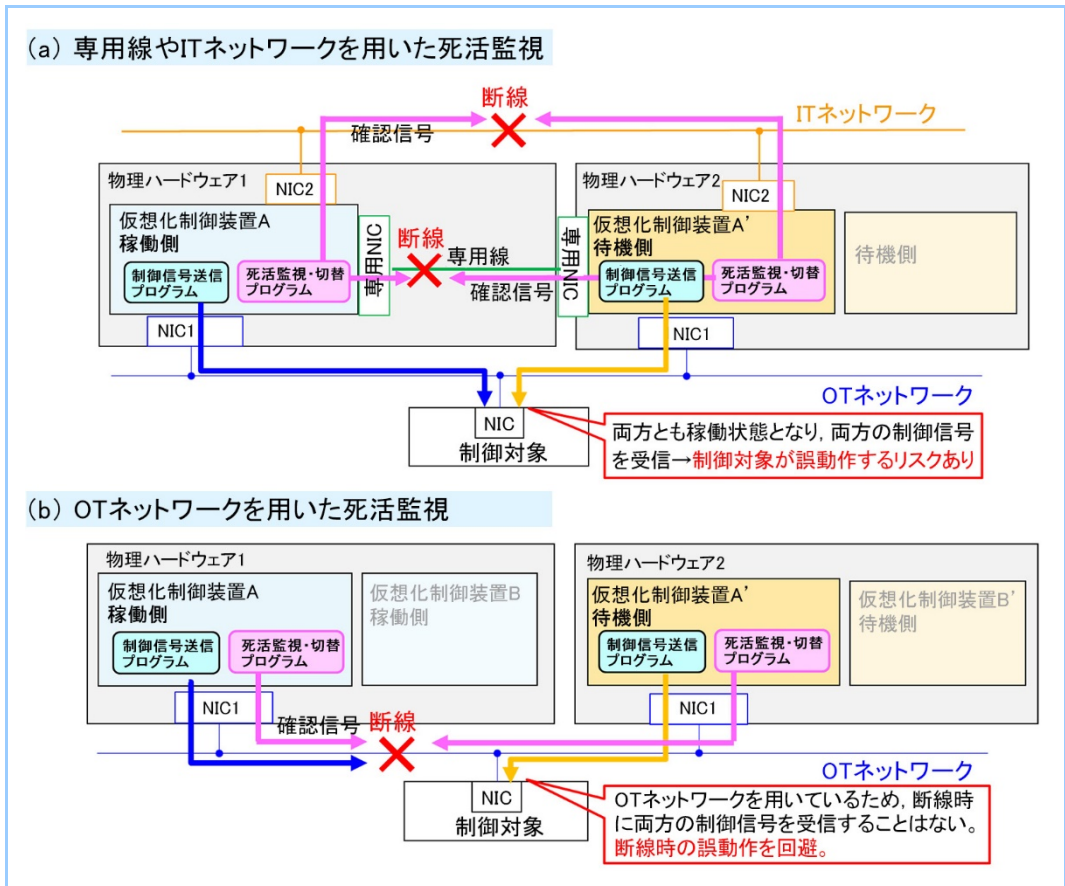


図8 ネットワーク断線障害発生時の挙動

5.2 稼働・待機切替え時間の評価

5.2.1 評価方法

二重化した冗長化構成において、制御対象で受信する制御信号の送信元が切り替わる直前と直後の受信間隔を稼働・待機切替え時間(図9)として評価した。4.2.1 節の通り、制御信号の受信間隔が制御周期(本報では 50msec の2倍)を超えた場合に異常として検知する。よって本評価では、切替え時間の目標値を 100msec 以下とし評価した。また、負荷条件として、定周期制御信号送信プログラムが動作する仮想化制御装置にファイル転送・書き込みの処理負荷を印加した。

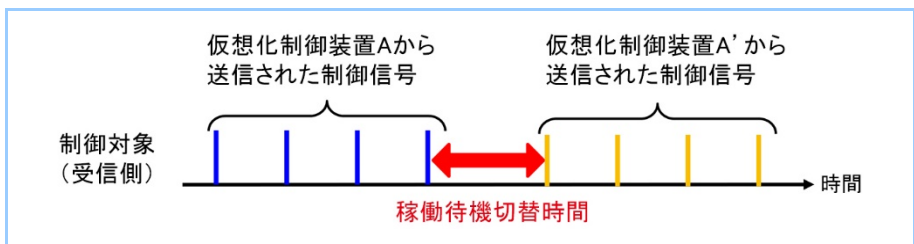


図9 稼働・待機切替え時間

5.2.2 評価結果

物理ハードウェア1の仮想化制御装置Aが稼働状態のときに、①仮想化制御装置Aのシャットダウン、②物理ハードウェア1のシャットダウンの2つ切替え条件を発生させ、稼働・待機切替え時間を各 10 回計測した。結果を図 10 に示す。

両条件において目標値 100msec 以下であることを確認し、仮想化制御装置においてもプラント制御装置の要求時間以内に切り替えられることが分かった。

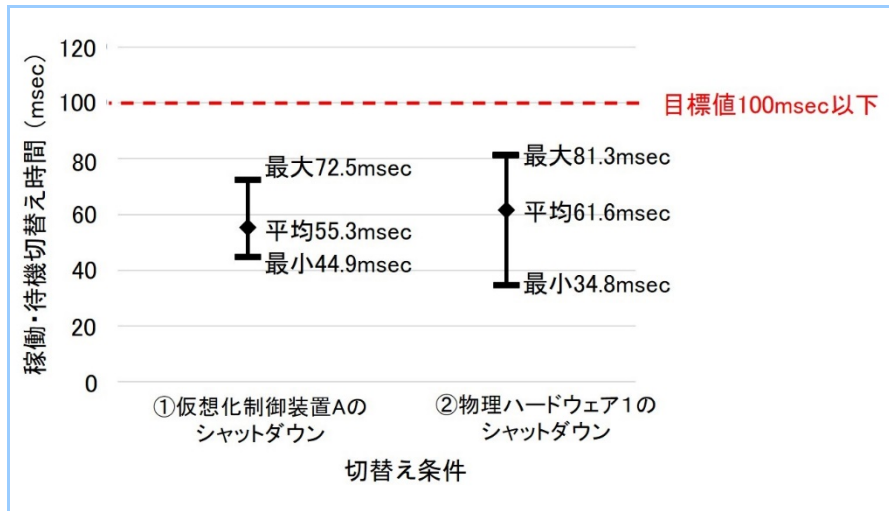


図 10 稼働・待機切替え時間の計測結果(両条件で各 10 回計測)

6. まとめ

プラント制御装置への仮想化技術適用によるコスト削減と CO₂ 排出量低減を狙いとして、実用化に向けて①仮想化制御装置のリアルタイム性の向上と②冗長化構成に適用可能な系統切替え技術を開発した。その結果、制御装置へ仮想化技術を適用した試作機において、プラント制御装置に要求されるリアルタイム性と可用性を満たすことを確認した。

今後、火力発電所や工場などで用いられる当社制御装置 Netmation 等への適用に向けて研究を進め、製品の高度化を図っていく。

参考文献

- (1) Kernel Virtual Machine
<https://www.linux-kvm.org>
- (2) Reghenzani, Federico, Giuseppe Massari, and William Fornaciari. "The real-time linux kernel: A survey on PREEMPT_RT." ACM Computing Surveys (CSUR) 52.1 (2019)
- (3) Kiszka, Jan. "Towards linux as a real-time hypervisor." Proceedings of the 11th Real-Time Linux Workshop. 2009