ホワイトリスト運用のための 複数の制御機器情報に基づいた運転状態の推定

秦康祐[1] 望月明典[1] 澤田賢治[1]
中井綱人[2] 山口晃由[2] 小林信博[2]

概要：産業用制御システム (ICS: Industrial Control System) をサイバー攻撃から防御する手法として，近年ホワイトリストが注目されている．先行研究では ICS の運転状態に応じてホワイトリストを切り替えることで高度な攻撃を検出する手法が提案されている．本稿では ICS を構成する複数の制御機器から取得した情報をもとに運転状態を推定する手法を提案する．これにより，ICS で生じた異常な動作の検出性能を向上させることができる．また，本稿では提案手法の有用性を確認するために行った数値実験の結果も示す．

Driving State Estimation via Multiple Control Device Information For Whitelist Operation

KOSUKE HATA[1] AKINORI MOCHIZUKI[1] KENJI SAWADA[1]
TSUNATO NAKAI[2] TERUYOSHI YAMAGUCHI[2] NOBUHIRO KOBAYASHI[2]

1. はじめに

近年，工場や発電所，ガスプラントなどの産業用制御システム (ICS: Industrial Control System) を対象としたサイバー攻撃による被害が報告されている．フィールド機器の破壊を目的とした Stuxnet，データ収集を目的とした Flame，そしてデータ破壊を目的とした Shamoon などのマルウェア被害が挙げられる [1]. また，2015 年 12 月にはウクライナの電力会社 3 社に対してサイバー攻撃が行われ，数時間にわたる停電が発生した [2]．さらに ICS の制御器である PLC (Programmable Logic Controller) を標的とした感染拡大型マルウェア PLC-Blaster が研究室レベルで開発されている [3]. このように，ICS を取り巻く環境は年々厳しさを増している．

ICS の防御手法として，近年ホワイトリストが注目されている [4]. ICS の正常な動作を登録したホワイトリストを PLC 等の制御器やネットワーク監視装置(以下, 監視装置)に組み込むことで，異常な動作の停止や抑制，検出が期待できる [5]. また, ICS はシステム環境の変化が少ないため，ホワイトリストの更新頻度を下げることができるほか，ブラックリスト型の異常検出機能より未知の攻撃に対応しやすい．そこで，本稿ではホワイトリストを監視装置に実装し，ICS で生じた異常な動作の検出を行う．

ICS には「停止中」や「立ち上げ」などの運転状態が存在し，各運転状態の通信・動作パターンが異なる．このため運転状態毎にホワイトリスト（以下，運転状態別ホワイトリスト）を作成し，高度な攻撃を検出する手法が提案されている [6][7]．先行研究[6]では通信プロトコル Modbus/TCP に着目した運転状態別ホワイトリストを監視装置に実装し，運転状態に関わる異常通信の検出を実現している．また，先行研究[7]ではマルウェアに感染した機器により ICS の運転状態に関する情報が改ざんされた場合に備え，運転状態の遷移方向（以下，運転サイクル）の数理モデルを利用した運転状態推定機能を与えている．本機能と文献[6]の方法を協調させることで，より信頼性の高いホワイトリスト型異常検出アルゴリズムを実現している．

一方で先行研究[7]は状態推定に 1 つの機器情報のみを利用しているため，複数の機器情報の改ざんに対応できない課題があった．そこで，より信頼性の高い異常検出アルゴリズムのために本稿では複数の機器情報に基づく運転状態の推定手法を提案する．このとき，ホワイトリストの監視対象となる機器の運転状態とホワイトリストの切り替えに利用する運転状態の推定結果が異なる場合がある．これは状態推定用のトリガー信号を生成するまでに要する時間が制御システムを構成する機器によって大きく異なるためである．そこで，本稿では複数の機器情報と連携したホワイトリストの適用方法も提案する．また，本稿では，気液プラントシステムのシミュレーションモデル利用し，模擬マ

1 電気通信大学，〒182-8585 東京都調布市調布ヶ丘 1-5-1.
UEC, 1-5-1, Chofugaoka, Chofu, Tokyo, 182-8585, Japan.
2 三菱電機株式会社，〒247-8501 神奈川県鎌倉市大船 5-1-1.
Mitsubishi Electric, 5-1-1, Oofuna, Kamakura, Kanagawa, 247-8501, Japan

ルウェアに対する提案手法の有効性を示す.

2. 問題設定

2.1 対象システム

本研究では, 先行研究[6][7]で作成した気液プラントシステムの MATLAB/Simulink モデル[8]を利用して提案手法の有効性を確認する. MATLAB/Simulink はモデル設計・解析・シミュレーション用の CAD ツールである.

対象システムはオペレータのインターフェースとなる HMI (Human Machine Interface), 弁や計測機器などの制御を行う PLC, 弁や計測機器などのフィールド機器, そしてネットワーク監視装置で構成される. 対象システムの構成図と HMI 上でオペレータが変更及び確認可能なパラメータをそれぞれ Fig. 1, Table 1 に示す. 制御目的はタンク内の圧力を目標値に追従するように制御することである. HMI, PLC, 監視装置はそれぞれ Ethernet ケーブルでスイッチングハブと接続している. 監視装置はハブのミラーリング機能によって, ハブを経由する通信情報を取得できる. PLC とフィールド機器は電気信号線で接続されている.

流入弁と流出弁は PLC から受信した指令値に基づき 10 秒かけて開閉を行う. また, 対象システムの運転状態は Table 2 に示すように流入弁と流出弁の開閉状態に応じて 4 種類定義されている. ここで「制御中」とは対象システムが目的の動作を行っている状態を示し,「立ち上げ」,「立ち下げ」はそれぞれシステムの運転状態が「停止中」から「制御中」もしくは「制御中」から「停止中」へと遷移する過程である. 対象システムの運転状態は HMI が管理する運転サイクルに基づき, HMI からの指令によって Fig. 2 に示すように遷移する.

ネットワーク機器間の通信プロトコルは, 通信仕様が一般に公開されている Modbus/TCP を利用する. Modbus/TCP の構成を Table 3 に示す. HMI から PLC への送信命令は, Function code によって分類することができる. Table 4 に運転状態毎の HMI から PLC への送信命令仕様を示す. ただし, 表中の FC は Function code を意味し, ○は命令内容の送信が許可されていること, ×は命令内容の送信が許可されていないことを示す.

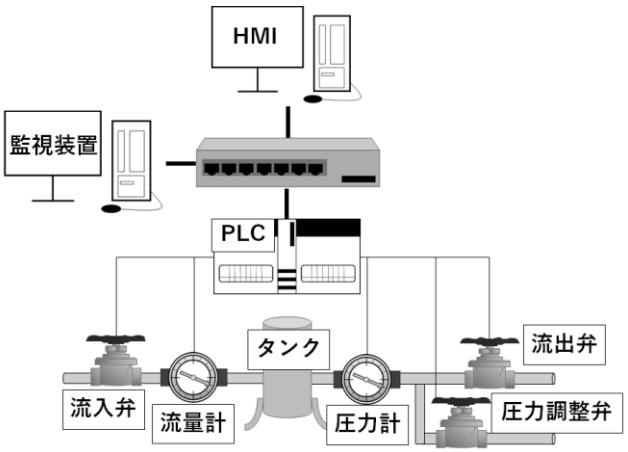

Fig. 1 対象システムの構成

Table 1 オペレータが変更及び確認可能なパラメータ

種類	データ	単位
変更可能	システムの運転状態 (停止中, 制御中)	-
	タンク内の圧力制御設定値	KPa
確認可能	流量の計測値	L/min
	圧力の計測値	KPa
	タンク内の圧力制御設定値	KPa
	弁の開閉状態	-
	システムの運転状態 (停止中, 立ち上げ, 制御中, 立ち下げ)	-

Table 2 弁の開閉状態とシステムの運転状態の関係

運転状態	流入弁	流出弁
停止中	閉	開
立ち上げ	開	開
制御中	開	閉
立ち下げ	閉	閉

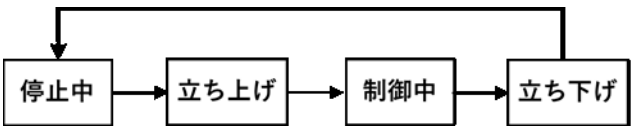

Fig. 2 対象システムの運転サイクル

Table 3 Modbus/TCP の構成

Data Type	TCP Header	Transaction ID	Protocol ID	Length
Data Size	-	16bit	16bit	16bit

Unit ID	Function code	Data
8bit	8bit	-

Table 4　HMI から PLC への送信命令仕様

命令内容	FC	対象システムの運転状態			
		停止中	立ち上げ	制御中	立ち下げ
・弁の開閉指令値	1 (Read)	○	○	○	○
・弁の開閉状態	2 (Read)	○	○	○	○
・圧力制御設定値	3 (Read)	○	○	○	○
・圧力の計測値 ・流量の計測値	4 (Read)	○	○	○	○
・運転状態 ・弁の開閉指令値	5 (Write)	○	○	○	○
・圧力制御設定値	6 (Write)	×	×	○	×

2.2 想定する攻撃内容

本稿では HMI 及び PLC がマルウェアに感染し，3 種類の攻撃がそれぞれ実行されたことを想定する．攻撃内容 1 では，マルウェアに感染した HMI が PLC に対して通信仕様に違反するコマンドを送信する．Table 4 に示すように「立ち上げ」時に本来許容されていない圧力制御設定値の変更を実施する．攻撃内容 2 では，HMI に感染したマルウェアが通信コマンドを改ざんし，本来の運転サイクルとは異なる運転状態の遷移を実行する．本稿では対象システムの運転状態を「制御中」から「立ち上げ」に不正遷移させる．攻撃内容 3 では，PLC に感染したマルウェアが PLC 内の情報を改ざんする．本稿では対象システムの運転状態が「制御中」のときに流入弁と流出弁の開閉状態情報をそれぞれ「閉」,「開」に改ざんする．

3. ホワイトリスト型異常検出アルゴリズム

2.2 節で示した 3 種類の攻撃を検出するために，本章では運転状態を推定する機能と運転状態別ホワイトリストの切り替え手法を述べる．本稿での新規性は複数の機器情報を利用した運転状態推定機能の実現及び運転状態別ホワイトリストの切り替えである．

まず，運転状態推定機能について述べる．本稿では HMI の不正な運転状態の遷移を検出するためにペトリネット[9][10]を利用して正常な運転サイクルの数理モデルを作成する．ペトリネットとは離散事象システムのモデル化に利用される有向グラフの一つである．ペトリネットはプレース及びトランジションと呼ばれる 2 種類のノードとそれらを接続するアークで構成される．ペトリネット (PN) は

$$PN = (P, T, F, W, M_0) \quad (1)$$

と表される．$P = \{P_1, P_2, \dots, P_m\}$はプレースの有限集合を意味し，$T = \{T_1, T_2, \dots, T_n\}$はトランジションの有限集合，$F$はアークの集合，$W$は重み付け関数，$M_0$は初期マーキングである．ペトリネットの動作の特徴を以下に示す．

- プレースはトークンを一つ以上保持することができ，このトークンの有無がシステムの状態を示している．
- アークはトークンの移動方向を示している．
- イベントが発生するとトランジションが発火し，トランジションに接続されるアークの方向と付記された重みに従いトークンは移動する．

ペトリネットをグラフで表現する場合はプレースを丸”○”，トランジションを棒”▮”，アークは”→”，トークンは点”・”で表す．

本稿ではプレースが対象システムの運転状態を示し，$x_i(i=1,2,3,4)$と表記する．同様にトランジションは対象システムの運転状態が遷移するイベントを示し，u_iと表記する．Fig. 3 に対象システムの運転サイクルのモデルを示す．x_1を停止中，x_2を立ち上げ，x_3を制御中，x_4を立ち下げとする．また，u_1を停止中から立ち上げ，u_2を立ち上げから制御中，u_3を制御中から立ち下げ，u_4を立ち下げから停止中へと変更するためのイベントとする．Fig. 3 において，x_1にトークンが 1 つあることはシステムが停止中であることを示す．

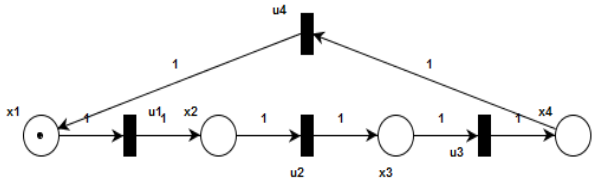

Fig. 3 ペトリネットを利用した運転サイクルモデル

Fig. 3 より，対象システムの運転サイクルを離散事象システムの線形状態方程式で表現する．$\boldsymbol{x}[k] \in \boldsymbol{R^n}$，$\boldsymbol{u}[k] \in \boldsymbol{R^r}$，$\boldsymbol{y}[k] \in \boldsymbol{R^m}$とすると，

$$\begin{cases} \boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] \\ \boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k] \end{cases} \tag{2}$$

と表せる．本稿では，

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

である．(2)式において k は状態遷移の回数を意味し，$\boldsymbol{x}[k]$は対象システムの運転状態（各プレース内のトークン数），$\boldsymbol{y}[k]$は観測可能な対象システムの運転状態，$\boldsymbol{u}[k]$は運転状態が遷移するためのイベントを示す．対象システムでは 1 つのイベントに対して運転状態が 1 つしか遷移しないため，$\boldsymbol{x}[k]$，$\boldsymbol{y}[k]$，$\boldsymbol{u}[k]$は単位ベクトルである．このとき，$\boldsymbol{B}$，$\boldsymbol{C}$ はそれぞれ

$$\boldsymbol{B} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

である．(2)式から運転状態推定機能を構成すると，

$$\begin{cases} \tilde{\boldsymbol{x}}[k+1] = \tilde{\boldsymbol{x}}[k] + \boldsymbol{B}\boldsymbol{u}[k] + \boldsymbol{K}(\tilde{\boldsymbol{y}}[k] - \boldsymbol{y}[k]) \\ \tilde{\boldsymbol{y}}[k] = \boldsymbol{C}\tilde{\boldsymbol{x}}[k] \end{cases} \tag{3}$$

と表せる．ただし，$\tilde{\boldsymbol{x}}[k]$，$\tilde{\boldsymbol{y}}[k]$はそれぞれ$\boldsymbol{x}[k]$及び$\boldsymbol{y}[k]$の推定量を示す．また行列$\boldsymbol{K} \in \boldsymbol{R^{n \times m}}$は運転状態の推定値と実値の誤差修正用のオブザーバゲインである．このとき，誤差$\boldsymbol{e}[k]$を

$$\boldsymbol{e}[k] = \tilde{\boldsymbol{y}}[k] - \boldsymbol{y}[k] \tag{4}$$

とすると，(2)式と(3)式の誤差系は

$$\boldsymbol{e}[k+1] = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C})\boldsymbol{e}[k] \tag{5}$$

と表せる．ここで，誤差$\boldsymbol{e}[k]$を収束させるために$(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{C}) = \boldsymbol{0}$を満たすようにオブザーバゲイン $\boldsymbol{K}$ を設計すると，$\boldsymbol{e}[k+1] = 0$，すなわち 1 回のイベント発生で推定誤差が 0 に収束することになる．そのようなオブザーバゲインは

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

である．本手法では，正常時には(5)式が成立すると仮定する．すなわち，(5)式が成立しないとき，対象システムにおいて異常な運転状態の遷移が発生したものとみなす．

HMI が PLC に送信する運転状態を$\boldsymbol{y}[k]$，HMI が PLC から取得する流入弁及び流出弁の開閉状態情報を$\boldsymbol{u}[k]$として，監視装置は運転状態推定機能(3)を運用する．$\boldsymbol{y}[k]$及び$\boldsymbol{u}[k]$は Modbus/TCP の Data 部に格納されている．弁の開閉状態と$\boldsymbol{u}[k]$の関係について Table 6 に示す．

Table 6 弁の開閉状態と$\boldsymbol{u}[k]$の関係

弁の開閉状態		$\boldsymbol{u}[k]$			
流入弁	流出弁	u_1	u_2	u_3	u_4
閉	開	0	0	0	1
開	開	1	0	0	0
開	閉	0	1	0	0
閉	閉	0	0	1	0

次に，運転状態別ホワイトリストの運用を実現する．先行研究[6]ではホワイトリスト条件として，送信元情報や送信先情報，データ長，Function code や設定値の範囲などを定義したペイロード条件を設定しているが，本稿ではフィールド機器の動きに着目したホワイトリスト運用のために Table 4 に示した Function code のみをホワイトリストに登録する．また，HMI から PLC に通信コマンドを送信することで運転状態に関わる弁の開閉指令値が送信されるため，HMI と PLC，監視装置のもつ情報が常に同期されているわけではない．また，本推定機能のトリガーとなる PLC の持つ弁の開閉状態情報は弁の開閉完了後に反映されるため，HMI の運転状態遷移時に限り運転状態と運転状態の推定結果は異なる．したがって，先行研究[7]と同様に運転状態の推定結果のみに基づいて運転状態別ホワイトリストを切り替えると，システムが正常な場合であってもホワイトリストにない通信コマンドを検出する可能性がある．そこで，HMI から取得した運転状態$\boldsymbol{y}[k]$と運転状態の推定結果$\tilde{\boldsymbol{y}}[k]$よりそれぞれ選択した運転状態別ホワイトリストを協調する手法を提案する．本稿では，$\boldsymbol{y}[k]$より選択したホワイトリストに無い通信コマンドの検出かつ$\tilde{\boldsymbol{y}}[k]$より選択したホワイトリストに無い通信コマンドの検出を確認した際に異常と判定する．

4. 対象システムのモデル化

4.1 HMI

本装置のモデルを作成するにあたり，Table 1 に示した情報の入出力機能を備える必要がある．そこで，本研究では Simulink に多数用意されているライブラリのうち，dashboard ライブラリを利用した．本ライブラリにはスイッチやボタン，ゲージ等の機能を有したブロックがあるため，入出力情報の可視化に役立つ．Simulink ではこのブロックを組み合わせることで，モデルの作成を行う．Fig. 4 に作成した HMI のインターフェース部のモデルを示す．

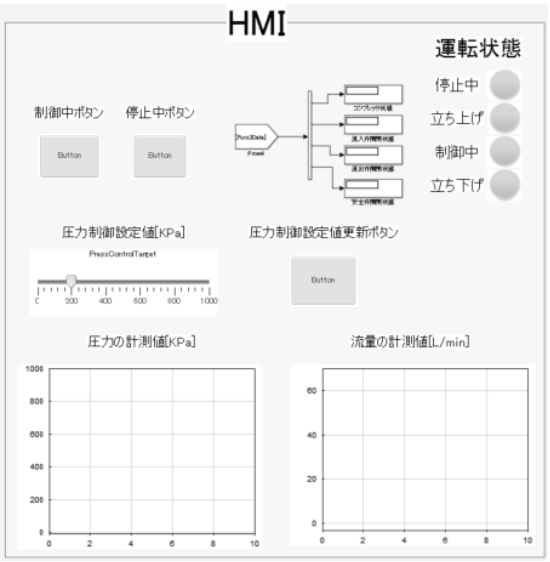

Fig. 4 HMI のインターフェースモデル

4.2 PLC

本装置は HMI からの命令に基づきフィールド機器の制御を行うほか，タンク内の圧力を制御するために圧力計から取得したタンク内の圧力値と HMI から受信した圧力制御設定値を利用して，圧力調整弁の開度を調整する．なお，圧力調整弁の開度は PID を利用して制御されており，本研究では Discrete PID Controller ブロックを利用した．

4.3 フィールド機器

本研究では流入弁と流出弁，圧力調整弁，流量計，圧力計，タンクのシミュレーションモデルを作成した．(6) 式に示した物質収支の関係が満たされるように本シミュレーションモデルを作成した．本研究ではタンク内に流入した水の質量とタンク内から流出した水の質量，そしてタンク内に蓄えられた水の質量の関係が満たされたとき物質収支の関係が満たされたとする．Table. 5 に (6) 式で使用する変数の意味を示す．

Table. 5　(6) 式の変数の説明

変数名	意味	単位
ρ	水の密度	$[\mathrm{Kg/m^3}]$
$V_0(t)$	タンクの体積	$[\mathrm{m^3}]$
$V_1(t)$	タンク内の空気の体積	$[\mathrm{m^3}]$
$R(t)$	水の流入流量	$[\mathrm{m^3/sec}]$
$Q(t)$	水の流出流量	$[\mathrm{m^3/sec}]$

$$\rho(V_0(t) - V_1(t)) = \rho \int (R(t) - Q(t))dt \qquad (6)$$

4.4 対象システムにおける通信

厳密な Modbus/TCP 通信のシミュレーションモデルを作成するのは困難なため，本研究では Modbus/TCP 通信を模擬したモデルを作成した．具体的には Table 3 に示した TCP Header を除くデータ配列を HMI 及び PLC モデルの内部でそれぞれ作成し，これを HMI と PLC のモデル間で受け渡しできるようにモデルを構成した．ここで，HMI から PLC に送信されるデータ配列の一例を Fig. 5 に示す．ただし，Fig. 5 に示した長方形の枠に示している数値の型は uint8 型である．

監視装置はハブによってミラーリングされた HMI と PLC 間の通信内容を取得する．本モデルでも同様にハブのモデル内部で複製した通信内容を監視装置のモデルに送信する．

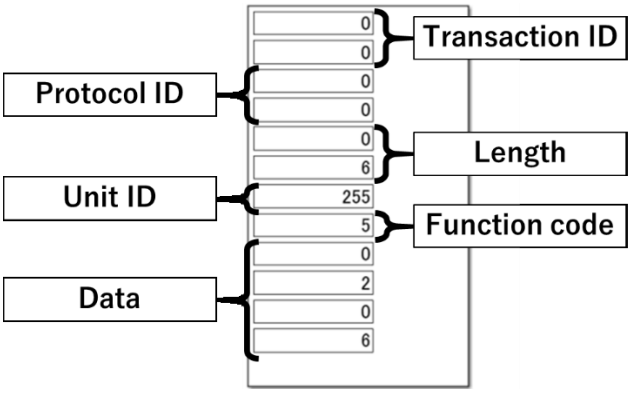

Fig. 5 HMI から PLC に送信されるデータ配列

5. 数値実験

5.1 実験方法

2.2 節に示した攻撃内容 1, 2, 3 をそれぞれ実行する．本数値実験では作成したマルウェアのモデルを HMI 及び PLC のモデルに予め実装しておき，シミュレーション中に任意のタイミングで対象システムに対して攻撃を実行できる仕様とした．

5.2 実験結果

攻撃内容 1, 2, 3 を実行した結果を示す．ただし，図には対象システムの運転状態の遷移状況とマルウェアが不正な動作を引き起こした時間，そして監視装置が異常を検出した時間を示す．なお，対象システムの運転状態と図中に示す数値の関係は Table. 7 に示した通りである．また，マルウェアによって不正な動作が発生しているとき及び監視装置が異常を検出したとき，図中では 1 を示す．

Table. 7 対象システムの運転状態と数値の関係

運転状態	数値
停止中	0
立ち上げ	1
制御中	2
立ち下げ	3

攻撃内容 1 を実行した結果（実験結果 1）を Fig. 6，7，8 に示す．Fig. 6，7 より，16 秒付近で対象システムの運転状態が「停止中」から「立ち上げ」に遷移し，22 秒から 26 秒付近まで複数回にわたって圧力制御設定値変更の不正な指令を HMI から PLC に送信したことがわかる. その結果，監視装置が 22 秒付近で異常を検出したことを Fig. 8 に示した．これにより，運転状態別ホワイトリストの適切な切り替えができていることがわかる．

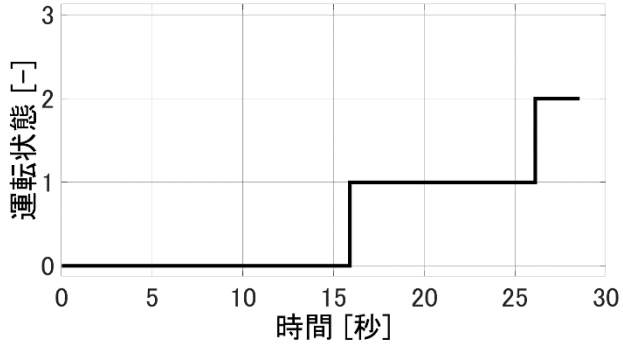

Fig. 6 攻撃内容 1 の実行結果 (a)

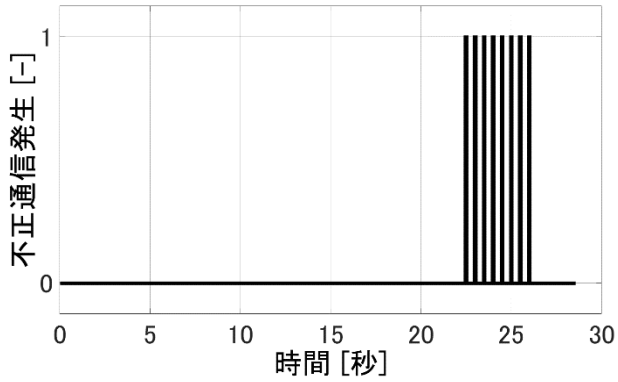

Fig. 7 攻撃内容 1 の実行結果 (b)

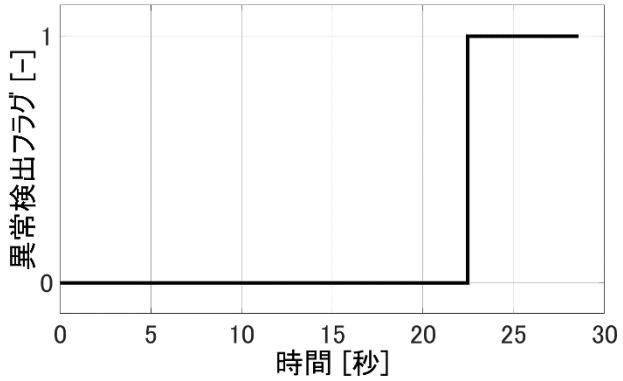

Fig. 8 攻撃内容 1 の実行結果 (c)

攻撃内容 2 を実行した結果（実験結果 2）を Fig. 9，10，11 に示す．Fig. 9，10 より，43 秒付近でマルウェアによって対象システムの運転状態が「制御中」から「立ち上げ」に不正遷移していることがわかる．この結果，監視装置が 10 秒程度遅延して異常を検出していることが Fig. 11 よりわかる. これは HMI の運転状態が不正遷移したことに伴い変更された弁の開閉指令値を PLC が受信し，そして流入弁や流出弁が実際に動作及び PLC の持つ弁の開閉状態情報に反映されるまでに時間を要したためである．先行研究[7]の手法では HMI から送信される弁の開閉指令値をトリガーとして利用しているため，異常検出までに時間を要しない. したがって, この点については今後の検討課題である．

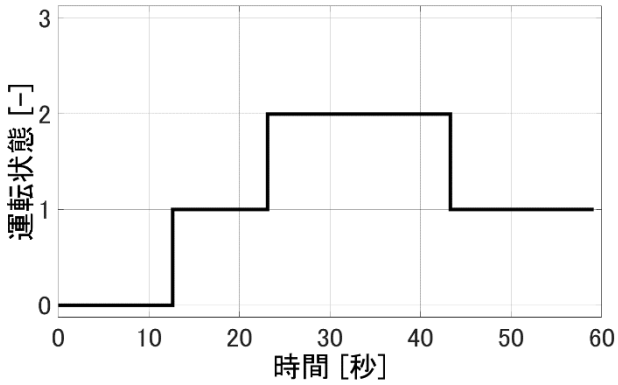

Fig. 9 攻撃内容 2 の実行結果 (a)

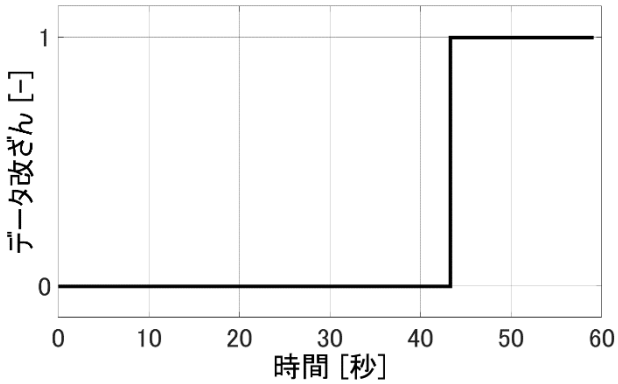

Fig. 10 攻撃内容 2 の実行結果 (b)

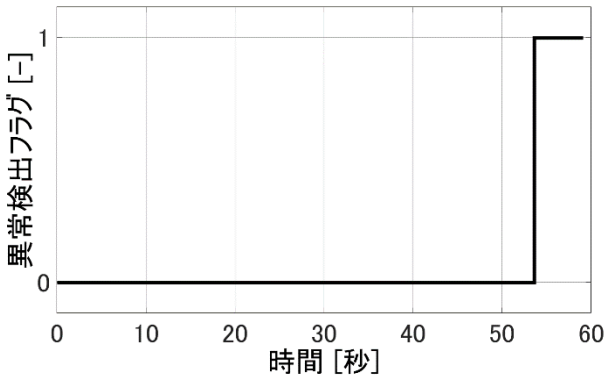

Fig. 11 攻撃内容 2 の実行結果 (c)

攻撃内容 3 を実行した結果（実験結果 3）を Fig. 12，13，14 に示す．対象システムが「制御中」である 42 秒付近で PLC が持つ弁の開閉状態情報を改ざんしたところ，監視装

置はほぼ同時刻に異常を検出したことがわかる．実験結果 2 と比較して監視装置の異常検出に要する時間が短かったが，これは運転状態推定機能のトリガーとなる$\boldsymbol{u}[k]$を直接改ざんしたためである．

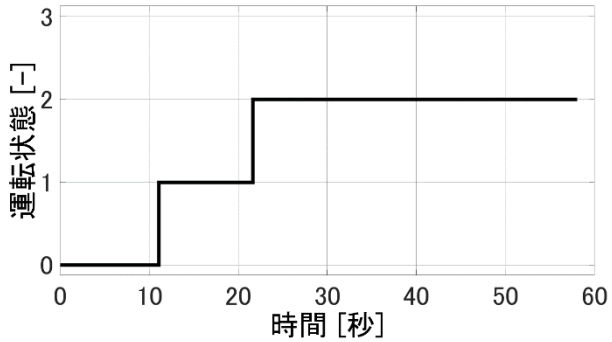

Fig. 12 攻撃内容 3 の実行結果 (a)

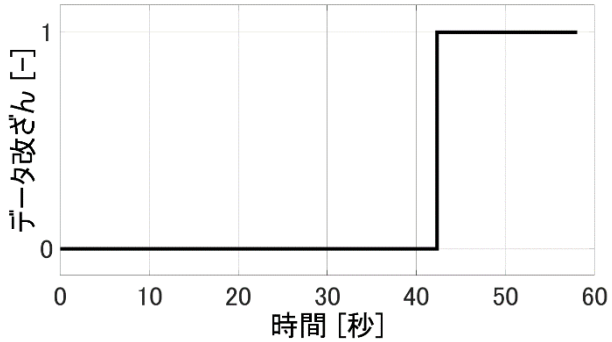

Fig. 13 攻撃内容 3 の実行結果 (b)

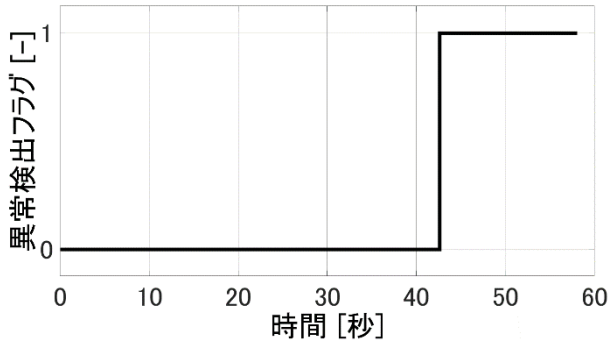

Fig. 14 攻撃内容 3 の実行結果 (c)

攻撃内容 3 は PLC 内の情報を改ざんしているため，先行研究[7]の手法では検出できない攻撃である．複数の機器情報を監視することで HMI と PLC の双方で生じた異常に対応できるため，本稿で示した手法の方が先行研究[7]と比較して優れているといえる．

6. 結論

本稿では複数の制御機器情報を利用して ICS の運転状態を推定する手法について提案した．そして，本機能と運転状態別ホワイトリストを協調させることで，高度な攻撃を検出できることを示した．今後の課題として，圧力や流量などの物理量を利用した異常検出手法の開発が挙げられる．これにより，ICS の実動作を異常検出の過程に反映させることができるため，より高度な攻撃の検出が期待できる．

参考文献

[1] S.Zhioua, “The middle east under malware attack dissecting cyber weapons,” International Confere-nce on Distributed Computing Systems Workshops, pp. 11-16

[2] G.Liang, “The 2015 Ukraine Blackout: Implications for False Data Injection Attacks” IEEE Transactions on Power Systems 2016

[3] https://www.blackhat.com/docs/asia-16/materials/asia-16-Spennebeg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf (2017 年 2 月閲覧)

[4] Jung, Woo-suk, et al. "Whitelist representation for FTP service in SCADA system by using structured ACL model." Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE, 2016

[5] 望月明典，澤田賢治，新誠一，細川嵩，“PLC のためのホワイトリスト式検知技術に関する検討,” MSCS2017，2017

[6] 中井綱人，山口晃由，清水孝一，小林信博“ライフサイクルに応じた産業用制御システム向けホワイトリスト型侵入検知システムのモデル化とシミュレーション評価,” SCIS2017，2017

[7] 秦康祐，佐々木翼，澤田賢治，中井綱人，山口晃由，小林信博，“プロセス制御システムのライフサイクルに基づいたホワイトリスト,” SCIS2017，2017

[8] https://jp.mathworks.com/products/simulink.html (2017 年 4 月参照)

[9] T. Murata, Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE, vol. 77, no. 4, 541/580 (1989)

[10]村田忠夫：ペトリネットの解析と応用(2011)