CC-001

安全検証を利用したセキュリティ機能の実現
― セキュリティと機能安全の両立のために ―

Implementation of Security Function Utilizing Safety Verification Function for Collaboration of Security and Safety

金川 信康†
Nobuyasu Kanekawa

1. はじめに

ディープ・ラーニングを初めとする人工知能は人知を超えた最適解を提供してくれるが，その安全性は必ずしも保証されたものではない。そこで人工知能の動作の安全性を検証，保証する技術を付加することにより，人知を超えた安心・安全な最適解を得られることが期待される。

報告者が属する研究グループ（以下「報告者ら」とする）が提案する安全検証型適応制御[1-4]によれば，AI に安全検証機能を付加することにより，より安全に制御に適用できるが，安全とセキュリティの両立も今後，重要な課題となる。なお安全とセキュリティの両立に関して，IEC (International Electro-technical Commission)では安全とセキュリティに関して TC65/WG20 にて TR (Technical Report)を策定中である[5, 6]。

2. 安全検証型適応制御[1]

2.1 知能化制御と安全性

深層学習に代表される知能化機能は人知を超えた最適解が期待できるが，内部状態が不明であるか，たとえ情報を得ることができても人間が理解できる形での表現は困難で，そのままでは安全性は保証されていない。そこで知能化制御 (Intelligent Control)に安全検証機能 (Safety Verification)を付加することにより人知を超えた安全な最適解を得られることが期待できる(図１)。

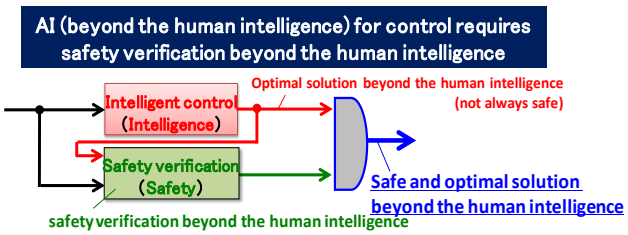


図１ 安全検証の必要性

自動運転にかかわる過去の事故事例を分析すると，日照条件による画像認識誤り，人間―機械双方による「だろう運転」，不適切な人為的介入（オーバーライド）が主な原因としてあげられる。
これらの内自動運転システムに起因する要因，すなわち，日照条件による画像認識誤り，機械による「だろう運転」は先にあげた，知能化制御に対する安全検証機能(Safety―Verification)により回避可能で，さらに入力データ補完により，機械側に起因する危険事象を回避可能である(図２)。

さらに人為的要因，すなわち，人間による「だろう運転」，不適切な人為的介入（オーバーライド）も人為的操作に対する安全検証機能により回避可能で，さらに安心志向制御により人間に不必要な不安感を抱かせないようにでき，不適切な人為的介入（オーバーライド）の根本原因となる不必要な人為的介入の機会を減らすことができる。

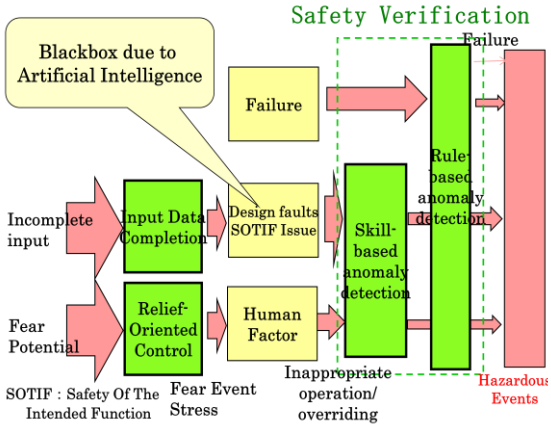


図２危険事象発生メカニズム

そこで，報告者らは知能化制御[2]，人為的操作に対する安全検証機能[3]，入力データ補完[4]，安心志向制御を備えた安全検証型適応制御(図３)を提案した。

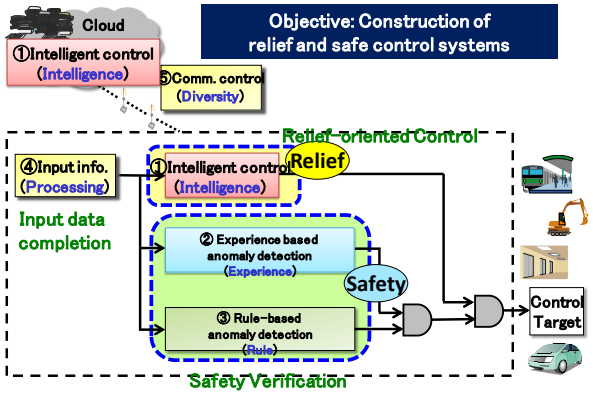


図３ 安全検証型適応制御

† （株）日立製作所, Hitachi, Ltd.

3. 安全性とセキュリティの両立

3.1 技術課題と対応策

安全性とセキュリティの関係は図 4 に示すように，
・両立する(Compatible：両者が独立している)場合
・相反する(Conflictive：一方を改善すると他方が悪化する)場合
・相互協調する(Collaborative：一方を改善すると他方も改善する)場合
の 3 つの場合が考えられ，常に両立するわけではない。

相反する場合の例として，脆弱性が発見された場合，セキュリティの見地から速やかに脆弱性を解消するためのバージョンアップ（応急的にはセキュリティパッチ）が必要とされるが，安全の見地からはバージョンアップ（セキュリティパッチ）の安全性の検証が必要である場合が挙げられる。そこで本研究では，脆弱性を解消するための速やかなバージョンアップとバージョンアップの安全性の検証の両立をさせ，さらには相互協調させることを目的とする。

報告者らによって提案されている人工知能を導入した制御システムに安全検証機能を備えて，自動制御機能安全性を検証した上で，実際に制御をする技術によれば，人工知能を導入した制御システムの安全性確保に加えて，該安全検証機能を有効活用することによりセキュリティ機能（サイバーアタックに対する耐性）を強化できる可能性がある。

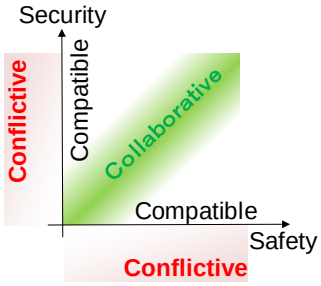


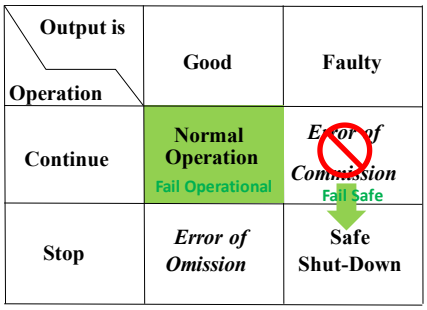
図 4　セキュリティと安全

上記目的を達成するために，以下の手段からなる安全検証機能活用セキュリティ方式を提案する。
(1) 制御システムに安全検証機能を備える。
(2) 脆弱性が検出された場合，またはセキュリティパッチを実施した場合には，安全検証を通常よりも強化する
(3) セキュリティパッチの検証が完了した場合，安全検証の強化を解除し，通常の安全検証に戻す。

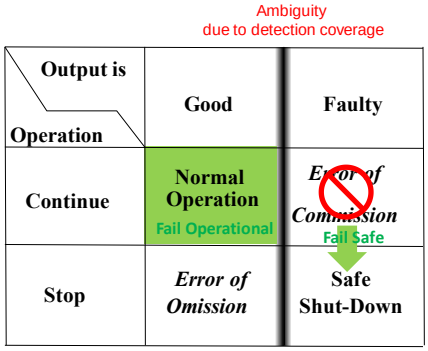
また動作中に経験ベース安全検証機能の学習をしている場合には，
(4) 脆弱性が検出された場合，またはセキュリティパッチを実施した場合には，経験ベース安全検証機能の学習を停止する。
(5) セキュリティパッチの検証が完了した場合，経験ベース安全検証機能の学習を再開する。

高信頼システムのあるべき挙動は，理想的には図 5 (a)に示すように，システムの出力が正しい場合には動作を継続し，正しくない場合には動作を停止させることである。なおここで，システムの出力が正しくない場合には確実に動作を停止させることに重点を置いた設計，もしくはその特性をフェイルセーフと呼び，システムの出力が正しい場合には動作を継続することに重点を置いた設計，もしくはその特性をフェイルオペレーショナルと呼ぶ。また，システムの出力が正しいにもかかわらず，動作を継続できないことを Error of Omission，システムの出力が正しくないにもかかわらず，動作を継続し，結果的に正しくない，さらには危険な動作をしてしまうことを Error of Commission と呼ぶことにする。

しかし現実には誤り・異常検出のカバレッジの限界から，図 5 (b)に示すように，システムの出力が正しい／正しくないという判断の境界に曖昧さが出てくる。



(a) Ideal



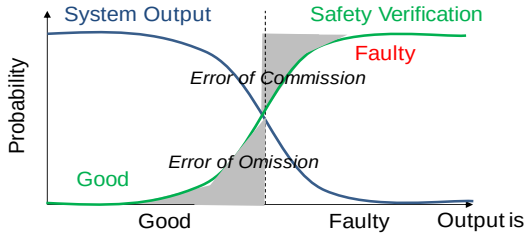
(b) In reality

図 5　ディペンダブルシステムの動作

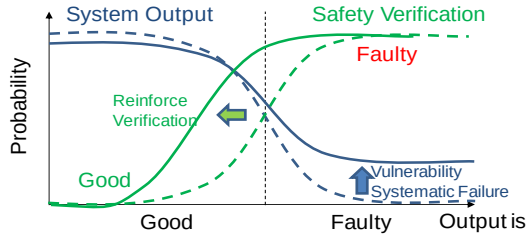
続いて，脆弱性やシステマティック故障（バグ）が存在する場合の高信頼システムのあるべき挙動を図 6 に示す。システムが正常な場合には図 6 (a)に示すように，システムの出力は正常である確率が異常である確率よりも高い。また安全検証機能は所定のカバレッジ（比率）をもって正常，異常を判断するため，異常であるのに正常と看做し，システムが動作を継続し，結果的に正しくない，さらには危険な動作をしてしまうた場合には Error of Commission が発生する。これとは逆に，正常であるのに異常と看做し，システムが動作を継続する機会を逃してしまう場合には Error of Omission が発生する。実際のシステムを設計する際には，アプリケーションの特質に応じて，Error of Commission と Error of Omission のリスクのトレードオフを図って，安全検証（正常，異常の判断）の閾値を設定するのが望ましい。

脆弱性やシステマティック故障（バグ）が存在する場合には図 6 (b)に示すように，システムが異常な出力を出す確率が高くなる。また，図 6 (c)に示すように，安全検証機能

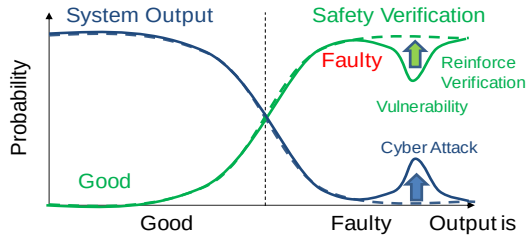
自体に脆弱性がある場合には脆弱性を突いたサイバーアタックが増えることが懸念される。



(a) Normal Operation



(b) with Vulnerability/Systematic Failure



(c) with Vulnerability in Safety Verification

図６　ディペンダブルシステムの動作（脆弱性，システマティック故障あり）

そのため提案方法では，脆弱性が発見された場合には手段(2)により，セキュリティパッチを実施以前には安全検証の強化によりサイバーアタックによる異常動作を検出できる確率を高める。さらに，セキュリティパッチを実施以降にもセキュリティパッチの検証が未完の場合には安全検証の強化を継続させてセキュリティパッチのバグによる異常動作を検出できる確率を高める。続いてセキュリティパッチの検証が完了した後には，手段(3)により安全検証の強化を解消して通常の安全検証に戻すことにより安全検証のフォルスポジティブ（正常であるのに異常であると誤検出してしまうこと）による Error of Omission の確率を下げる。

また手段(4)により，セキュリティパッチを実施以前には経験ベース安全検証機能の学習の停止によりサイバーアタックによる誤った学習（有害な学習）を防止し，セキュリティパッチを実施以降にはセキュリティパッチのバグによる誤った学習（有害な学習）を防止することができる。

3.2 安全検証機能活用セキュリティ方式

図 7 は提案方式による状態遷移の例である。初期状態では安全検証緩和状態 S0 にあり，自動制御機能に脆弱性が発見された場合には脆弱性による誤動作を検出するために安全検証を強化する安全検証強化状態 S1 に遷移する。脆弱性を解消するための自動制御機能にセキュリティパッチを実施した後に，セキュリティパッチのバグによる誤動作を検出するために安全検証の強化を継続する安全検証強化継続状態 S2 に遷移する。その後セキュリティパッチを実施した自動制御機能について安全検証を網羅的に完了するか，形式検証を完了して脆弱性が解消した後に再び安全検証緩和状態 S0 に戻る。

なお，脆弱性が発見されたという事象は，制御システム自らが安全検証機能により検出した異常動作からサイバータックとそれに対する脆弱性を検出する場合と複数の制御システムを管理する管理センタを有し，センタから通信路を介して通知される場合が考えられる。後者の場合，管理センタでは，管理する複数の制御システムからの誤動作情報からサイバータックとそれに対する脆弱性を検出する。

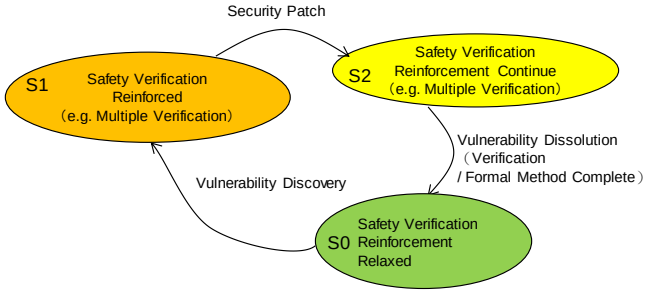


図７　安全検証機能活用セキュリティ方式

図８は経験ベース安全検証をオンボードでも学習する場合の状態遷移の例である。初期状態の安全検証緩和状態 S0 では経験ベース安全検証学習を開始し，安全検証強化状態 S1 では脆弱性による誤った学習を防止するために経験ベース安全検証学習を停止する。安全検証強化継続状態 S2 では，脆弱性はセキュリティパッチにより対策されたものの，セキュリティパッチの検証が済んでいないので，セキュリティパッチのバグによる誤った学習を防止するために経験ベース安全検証学習を停止したままとする。その後，安全検証緩和状態 S0 では脆弱性対策のためのセキュリティパッチの検証が済んでいるので経験ベース安全検証学習を再開する。

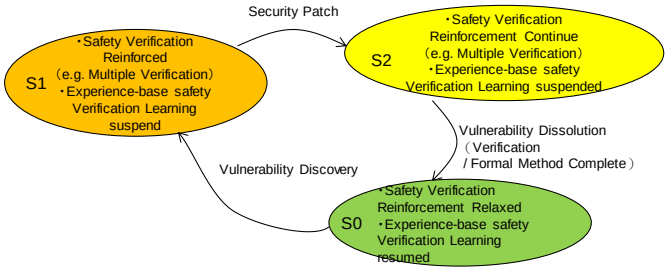


図８　安全検証機能活用セキュリティ方式（オンボード学習あり）

なお，安全検証強化はの実現手段としては次の２つが考えられる。

・安全検証強化時には，リミッタ型出力制限方式を AND 型出力制限方式に切り替えるか，制限出力許容範囲を狭める。

・安全検証強化時には，安全検証を図９に示すように多段に重ねる。

図９の例では多段に重ねた安全検証機能それぞれの出力により制御される ANS ゲート（または出力制限回路）により制限してスイッチ SW2 により安全検証強化状態 S1，安全検証強化継続状態 S2 には安全制限出力を選択して制御出力として出力する。

多段とした安全検証機能に同じ判定論理を実装した場合には，安全検証機能は冗長系として機能し，いずれかが故障した場合でも，制御出力を安全のために制限する機能を確保することができる。また，安全検証機能に異なる判定論理を実装した場合には，設計多様化の効果により判定論理に依存する検出漏れを防ぐことが可能となる。特に，Figure 3(c)に示すように，安全検証機能自体に脆弱性がある場合には脆弱性を補うために有効な手段である。また，安全検証機能のうち少なくとも１つに深層学習などの人工知能による判定論理，さらに少なくとも１つにルールに基づく判定論理をそれぞれ実装することで，人工知能による人知を超えた異常（危険事象）検出と，確実なルールに基づくアカウンタビリティ（説明責任，説明性）を両立させることが可能となる。なお，スイッチ SW1 を上に倒すことにより安全検証強化緩和状態 S0 を実現することができる。

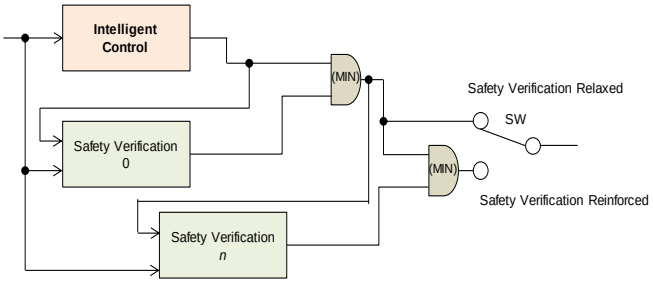


図９　安全検証の強化：多段化

図１０は状態遷移のトリガーとなる事象が動作不良可能性発見である動作例である。動作不良可能性発見という事象には，脆弱性の発見に加えて，運用開始後の不良箇所の発見などの事象が含まれる。

なお，動作不良可能性発見という事象は脆弱性発見と同様に，制御システム自らが安全検証機能により検出した異常動作から動作不良可能性発見を検出する場合と複数の制御システムを管理する管理センタを有し，センタから通信路を介して通知される場合が考えられる。後者の場合，管理センタでは，管理する複数の制御システムからの誤動作情報から動作不良可能性発見を検出する。

初期状態では安全検証緩和状態 S0 にあり，自動制御機能に脆弱性が発見された場合には脆弱性による誤動作を検出するために安全検証を強化する安全検証強化状態 S1 に遷移する。脆弱性を解消するための自動制御機能にセキュリティパッチを実施した後に，セキュリティパッチのバグによる誤動作を検出するために安全検証の強化を継続する安全検証強化継続状態 S2 に遷移する。その後セキュリティパッチを実施した自動制御機能について安全検証を網羅的に完了するか，形式検証を完了して脆弱性が解消した後に再び安全検証緩和状態 S0 に戻る。

なお，脆弱性が発見されたという事象は，制御システム自らが安全検証機能により検出した異常動作からサイバーアタックとそれに対する脆弱性を検出する場合と複数の制御システムを管理する管理センタを有し，センタから通信路を介して通知される場合が考えられる。後者の場合，管理センタでは，管理する複数の制御システムからの誤動作情報からサイバーアタックとそれに対する脆弱性を検出する。

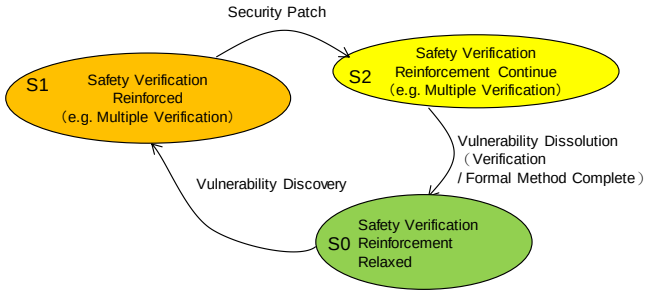


図１０ 安全検証機能活用システマティック故障対策

3.3 提案方式による制御システムの動作

提案方式による制御システムの動作の例（机上検証シナリオ）を表に纏め，図１１－１５に個々に示す。

表　机上検証シナリオ

Case	Scenario		Results	
	Event	When	with Safety Verification Reinforce	without Safety Verification Reinforce
1	Tolerable Fault	any time	No Hazardous Events	No Hazardous Events
2	Cyber Attack	before Security Patch	No Hazardous Events	Hazardous Events
3		after Security Patch	No Hazardous Events	No Hazardous Events
4	Security Patch Bug Exteriorizad	after Security Patch	No Hazardous Events	Hazardous Events
5	Cyber Attack to Safety Verificatiion	before Security Patch	No Hazardous Events	Hazardous Events
6		after Security Patch	No Hazardous Events	No Hazardous Events

夫々の動作例において，時刻 t1 に脆弱性が発見され，安全検証強化状態 S1 に遷移し，時刻 t2 にセキュリティパッチを実施した後に安全検証強化継続状態 S2 に遷移し，時刻 t3 にセキュリティパッチの検証が完了して安全検証強化緩和状態 S0 に戻る。提案方式により安全検証を強化（カバレッジは 100%と仮定する。）した場合のシステムの動作を実線で，安全検証を強化しなかった場合のシステムの動作を破線で示す。全ての場合において，安全検証を強化することにより，危険事象が発生しないことがわかる

・Case 1

図１１は te1，te1’においてシステムが許容できる故障１が発生し，te2 においてシステムが許容できない故障２が発生した場合の動作例である。この場合には，te1 においては安全検証が強化された場合には念のため制御動作を停止して出力を安全状態にする。ここで安全状態の出力はシステムの用途に依存し，例えば鉄道制御においては動力を切って，ブレーキをかけることにより安全状態を維持できる。自動運転に代表されるように自動車においては，緩やかにブレーキをかけて，緩やかに減速しながら停止するか，人為的オーバーライドの優先度を上げるか，全面的に人為的オーバーライドに切り替える。

一方，安全検証が強化されなかった場合には破線で示すように出力を継続することができるが，次の図１２の破線に示す動作例のようにサイバーアタックを受けた場合には安全検証が強化されていないために危険事象発生の恐れがある。

時刻 te2 においてシステムが許容できない故障２が発生した場合には，通常の（緩和された）安全検証により移動が検出され，制御動作を停止して出力を安全状態にする。

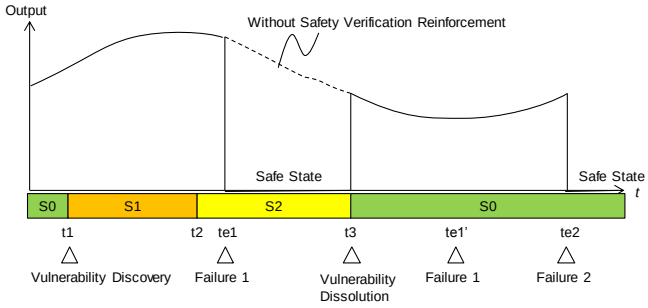


図１１　提案方式による制御システムの挙動（サイバーアタック無，故障のみ）

・Case 2

図１２は，時刻 ta1（セキュリティパッチ実施する時刻 t2 以前）においてサイバーアタックがあった場合の動作例で，安全検証が強化された場合には異常を検出して制御動作を停止して出力を安全状態にすることができるが，破線で示すように安全検証が強化されなかった場合には異常を検出できずに危険事象が発生する。

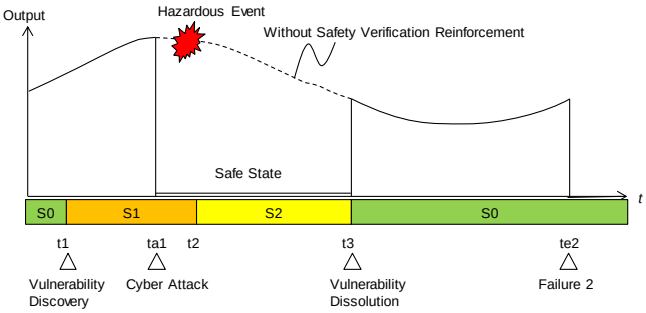


図１２　提案方式による制御システムの挙動（セキュリティパッチ前にサイバーアタック）

・Case 3

図１３に示すように時刻 ta1（セキュリティパッチ実施する時刻 t2 以降）においてサイバーアタックがあった場合にはセキュリティパッチが実施されており，サイバーアタックの影響を受けないので，危険事象は発生しない。

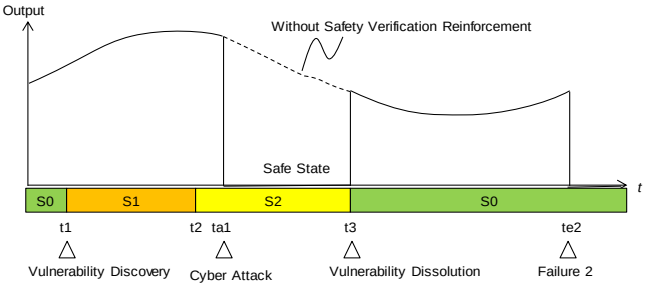


図１３　提案方式による制御システムの挙動（セキュリティパッチ後にサイバーアタック）

・Case 4

図１４は時刻 tx1 においてセキュリティパッチのバグが顕在化した場合で，安全検証が強化された場合には異常を検出して制御動作を停止して出力を安全状態にすることができるが，破線で示すように安全検証が強化されなかった場合には異常を検出できずに危険事象が発生する。

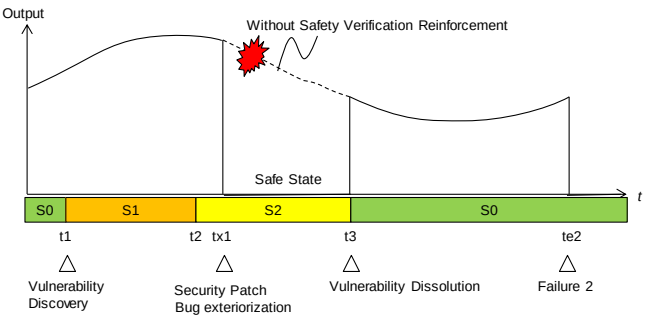


図１４　提案方式による制御システムの挙動（セキュリティパッチのバグ顕在化）

・Case 5

図１５は安全検証強化状態 S1，安全検証強化継続状態 S2 において安全検証を強化し，経験ベース安全検証の学習を停止した場合のシステムの動作を実線で，安全検証を強化せずに経験ベース安全検証の学習を停止しなかった場合のシステムの動作を破線で示す。本実施例では，時刻 te3 において発生した故障３が時刻 ta1 におけるサイバーアタックの結果発生した危険事象（有害な学習結果）に引き起こされる危険事象を発生する場合を想定している。

この動作例によれば，提案方式に従い時刻 ta1 におけるサイバーアタック発生時に安全検証を強化し，経験ベース安全検証の学習を停止していれば，該サイバーアタックの結果発生した危険事象（有害な学習）を見逃すことも無く出力を安全状態にすることが出来る上，時刻 te3 において発生した故障３が生じる有害な学習に起因する危険事象を正常と判断して出力を継続して危険事象を発生してしまうこともない。一方，時刻 ta1 におけるサイバーアタック発生時に安全検証を強化せずに，経験ベース安全検証の学習も停止しない場合には。時刻 ta1 におけるサイバーアタックの結果発生した危険事象を経験ベース安全検証機能が正常な実績として学習してしまい，時刻 te3 において発生した故障３が生じる危険事象を正常と判断して出力を継続して危険事象を発生してしまう。

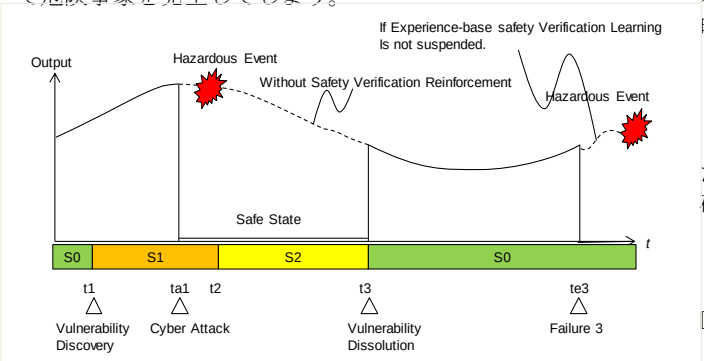


図１５　提案方式による制御システムの挙動.（オンボード学習あり）

・Case 6

なお，セキュリティパッチ実施(t2)以降にサイバーアタックがあった場合にはセキュリティパッチにより有害な学習が防止できるので危険事象が発生することがない。

4. おわりに

脆弱性を解消するための速やかなバージョンアップとバージョンアップの安全性の検証の両立をさせ，さらには相互協調させるために，以下の手段からなる安全検証機能活用セキュリティ方式を提案した。

(1) 制御システムに安全検証機能を備える。
(2) 脆弱性が検出された場合，またはセキュリティパッチを実施した場合には，安全検証を通常よりも強化する
(3) セキュリティパッチの検証が完了した場合，安全検証の強化を解除し，通常の安全検証に戻す。

また動作中に経験ベース安全検証機能の学習をしている場合には，

(4) 脆弱性が検出された場合，またはセキュリティパッチを実施した場合には，経験ベース安全検証機能の学習を停止する。
(5) セキュリティパッチの検証が完了した場合，経験ベース安全検証機能の学習を再開する。

上記提案方式により，全ての場合において，安全検証を強化することにより，危険事象が発生しないことを机上検討により示した。なお，机上検討ではセキュリティ対策のカバレッジを 100%と仮定しているが，カバレッジの定量的評価手法の確立が今後の課題である。

謝辞

FIT2017 においてご討論，貴重なコメントを下さいました方々，本研究の機会を与えてくださった(株)日立製作所研究開発グループ各位に心より感謝いたします。

参考文献

[1] 金川信康，人工知能の制御へのより安全な適用について，FIT2017, CC-003 (2017)
[2] 広津鉄平, 堀口辰也, 中村敏明, 田向権, 深層学習を活用した高精度知能化制御の提案, FIT2017, CF-007 (2017)
[3] 中川慎二, 組み込みシステム向け異常検知方式, FIT2017, F-013 (2017)
[4] 西田武央, 奥出真理子, 隠れマルコフモデルを用いた複数個体による高信頼環境情報の推定技術, FIT2017, CO-014 (2017)
[5] 石田茂, 組込みシステムセーフティ・セキュリティ検討 WG の取り組み, SEC journal Vol.12 No.3 pp.34-35 (2016).
[6] Subcommittee(s) and/or Working Group(s)>　TC 65/WG 20 http://www.iec.ch/dyn/www/f?p=103:14