情報処理学会論文誌　データベース　Vol.12　No.1　11–16 (Jan. 2019)

テクニカルノート

# シミュレータによる動的ホワイトリストを用いたサイバーフィジカル型攻撃検知方式

立床 雅司[1,a)]　樋口 毅[1]　河内 清人[1]　永谷 達也[2]　古澤 康一[2]　米田 健[1]

受付日 2018年6月8日, 採録日 2018年7月30日

**概要**：ネットワーク経由で工場どうしが協調連携することで柔軟な製造を行う「つながる工場」が未来の製造システムの主流となると想定されている．一方，工場どうしがネットワークを通じてやりとりすることにより，つながる工場に対するサイバー攻撃の懸念は高まる．本稿では，つながる工場の製造コマンドに対する攻撃に着目し，つながる工場での適用が想定されるイベントドリブン型製造方式において，製造コマンドの宛先ごとに製造コマンドの順序が保証される性質を利用した攻撃検知方式を提案する．提案方式に基づき同一の生産計画から得られた実際の製造コマンドとシミュレータの予測結果とを比較するシステムを検討し，フィージビリティを検証した．

**キーワード**：つながる工場，マスカスタム生産，スマート製造，攻撃検知，ファクトリセキュリティ

# A Cyber Physical Attack Detection Method Using Dynamic Whitelist by Simulator

Masashi Tatedoko[1,a)]　Tsuyoshi Higuchi[1]　Kiyoto Kawauchi[1]
Tatsuya Nagatani[2]　Koichi Furusawa[2]　Takeshi Yoneda[1]

Received: June 8, 2018, Accepted: July 30, 2018

***Abstract:*** It is assumed that the “Connected Manufacturing”, which enables flexible manufacturing by collaborative cooperation between factories through a network, will be the main stream of the future manufacturing system. However, since factories exchange with each other through a network, the risk of cyber attack increases. In this work, we focus on tampering with manufacturing commands and propose an attack detection method using the order of manufacturing commands is guaranteed for each destination when event driven manufacturing method is applied. Based on the proposed method, we examine a system that compares the actual production commands obtained from the same production plan with the simulation result of simulator, and verified the feasibility.

***Keywords:*** connected manufacturing, mass customization, smart manufacturing, attack detection, factory security

## 1. はじめに

多様化，細分化される消費者の嗜好に応えるために，未来の製造システムは，同じ製品を大量に生産する形態から，より柔軟な製造が可能な形態に変化していくと考えられている [1]．本稿では，未来のスマートな製造システムにおける工場をマスカスタム生産工場と呼び，それらがネットワークを通じて連携する形態をつながる工場と呼ぶ [2]．

つながる工場は，柔軟な製造を効率良く，低コストで実現できる．一方，工場どうしがネットワークを通じて製品の製造に必要な情報をやりとりすることにより，つながる工場に対するサイバー攻撃の懸念が高まってくる．2010 年には，イランの核燃料施設がウィルスに感染し，ウラン濃縮用遠心分離機が破壊されるという被害があった [3]．身

[1] 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 247–8501, Japan
[2] 三菱電機株式会社先端技術総合研究所
Advanced Technology R&D Center, Mitsubishi Electric Corporation, Amagasaki, Hyogo 661–8661, Japan
[a)] Tatedoko.Masashi@ap.MitsubishiElectric.co.jp

© 2019 Information Processing Society of Japan

近な例では，冷凍食品や缶詰，ファストフード店等における異物混入による健康被害が問題となっており，工場に対するサイバー攻撃により人命および経済が危機にさらされる懸念が高まってきている．

工場における異常検知について整理した文献 [4] によれば，つながる工場における異常は製造不良によるもの，攻撃によるもの，およびセンサの不良等その他のものに分類できるとされている．本稿では，つながる工場におけるマスカスタム生産工場の製造コマンドに対する攻撃に着目し，改ざんを検知するための攻撃検知方式を提案する．

## 2. 問題設定

### 2.1 未来のマスカスタム生産工場

本稿では，つながる工場におけるマスカスタム生産工場を図 1 に示すように，スケジューラ，MES [*1] サーバ [5]，設備コントローラ，AGV [*2] コントローラ [6]，ジョブショップ（作業台），AGV から構成される工場と想定する．ジョブショップは，複数のロボットおよび製造装置から構成され，食品の詰め込み等の作業を実行する．AGV は，ジョブショップ間の搬送を担当する．スケジューラは，生産計画に基づき，各ジョブショップに割り付ける仕事のスケジューリングを行い，スケジュールを MES サーバに通知する．MES サーバは，スケジュールとジョブショップの製造能力および AGV の搬送能力に基づき，設備コントローラおよび AGV コントローラに依頼する作業内容を記載した製造コマンドを発行する．製造コマンドには，送付すべきコントローラを示す宛先，製造対象となる製品の ID，製造コマンドの名前，製造コマンドのパラメータといった情報が含まれている．設備コントローラおよび AGV コントローラは，製造コマンドに基づきジョブショップおよび AGV を制御することで製品の製造を行っている．MES サーバの実現方式の一例として，MES サーバに製造コマンドを蓄積するための RDBMS [*3] を用意し，各コントローラからの SQL 問合せにより製造コマンドの受け渡しを実現する方法が知られている [5]．

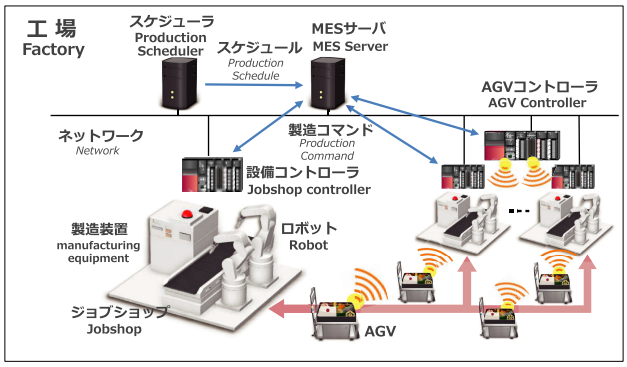


図 1　つながる工場におけるマスカスタム生産工場の基本構成
**Fig. 1**　The future mass customization factory structure.

*1 Manufacturing Execution Systems
*2 Automated Guided Vehicle
*3 Relational Database Management System

### 2.2 製造コマンドによる製造指示

MES サーバから各コントローラに製造コマンドを受け渡すための製造指示方式には，製造指示のタイミングに応じて，タイムドリブン型製造指示方式とイベントドリブン型製造指示方式の 2 種類の製造指示方式がある [7]．タイムドリブン型製造指示方式は，製造装置が実施する作業について作業時間を正確に見積もり，作業ミスが起こらないように事前に調整可能な場合に現実的な時間でスケジューリングを行うことが可能である．一方，イベントドリブン型製造方式は，作業時間の見積りに幅がある場合および作業ミス等の外乱が発生しうる場合でもロバストなスケジューリングを行う．想定している未来のつながる工場では，作業時間の見積りおよび事前調整が困難と想定されるため，イベントドリブン型製造方式が適用されると考えられている [8]．イベントドリブン型製造指示方式では，各コントローラは，製造コマンドを取得していない場合に MES サーバから製造コマンドを取得し，製造コマンドに基づき作業可能となるまで待機し，作業可能と判断すると作業を開始して，作業終了後に MES サーバから製造コマンドを取得するという流れを繰り返すことで柔軟性を確保している．

### 2.3 製造コマンド改ざんの課題

MES サーバから発行される製造コマンドは，ネットワークを経由して設備コントローラおよび AGV コントローラに送信される．そのため，ネットワークを流れる製造コマンドを改ざんされる脅威が存在する．たとえば，ARP キャッシュポイズニング [9] 等の手法を用いた中間者攻撃により改ざんが行われる，不正に入手された管理パスワードを用いて MES サーバを操作される，等の攻撃が考えられる．製造コマンドが改ざんされた場合，食品アレルギーを引き起こす物質の混入等の不正な製品の出荷による人命に関わる事態や，製造不良による歩留りの悪化，ロボットの故障による経済的損失等を引き起こす可能性がある．MES サーバは攻撃される可能性があるため，MES サーバを信頼して製造コマンドを直接取得することは困難であり，ネットワーク上を流れる製造コマンドをキャプチャしてその内容から攻撃検知する方法が考えられる．しかしながら，ネットワーク上を流れる製造コマンドはジョブショップおよび AGV の稼働状況により順序が変化する可能性があるため事前にホワイトリストを用意しておくことが困難である．また，動的に問題のある製造コマンドを網羅的に予測してブラックリストを作成することも困難である．

## 3. 攻撃検知方式

製造コマンド改ざんの課題は，ネットワーク上を流れる

© 2019 Information Processing Society of Japan

製造コマンドを事前に予測することが困難なことが原因であった．本稿では，イベントドリブン型製造方式において，製造コマンドの宛先に着目すると製造コマンドの順序が保証される性質を利用し，シミュレータにより動的作成したホワイトリストをキャプチャした製造コマンドと比較することで製造コマンドの改ざんを検知するサイバーフィジカル型攻撃検知方式を提案する．単純にシミュレータにより予測したネットワーク上を流れる製造コマンドと実工場から取得した製造コマンドを比較する場合，製造遅延等により設備コントローラおよび AGV コントローラから MES サーバへの問合せタイミングがずれることがあるため順序が一致することはまれであり，製造コマンドが改ざんされているかどうかの判断を正確に行うことが困難である．一方，本方式では，製造コマンドの宛先ごとに到達順を記憶しておき，到達順が同じ製造コマンドについて製品 ID，製造コマンド名，製造コマンドのパラメータが一致することを確認することで即座に製造コマンドが改ざんされているかどうかの判断を行うことができる．

本方式は，**図 2** に示すように，生産計画サーバ，実工場，仮想工場，攻撃検知システムから構成される．生産計画サーバは，仮想工場と実工場に同一の生産計画を入力する．実工場は，2 章で述べたマスカスタム生産工場であり，生産計画に基づきスケジューラが作成したスケジュールに従い MES サーバが製造コマンドを発行することでネットワーク上に製造コマンドが流れる．仮想工場は，実工場の各構成要素をモデル化したものであり，生産計画に基づきシミュレーションを行うことでネットワーク上を流れる製造コマンドを予測する．攻撃検知システムは，実工場のネットワーク上を流れる製造コマンドをキャプチャした結果得られる実製造コマンドと，仮想工場のシミュレーション結果により得られた予測製造コマンドとを比較することで，製造コマンドの改ざんを検知する攻撃検知方式を実現したシステムである．

本方式が正しく動作するためには，仮想工場におけるシミュレーションの健全性，攻撃検知システムの健全性，仮想工場と攻撃検知システムとの間の通信経路の健全性が保証されていることが前提となる．ここで，仮想工場および攻撃検知システムは，侵入を受けやすい工場ネットワークからのアクセスを厳しく制限でき，物理的なアクセスも他の設備と比べて厳格に制限できる．そのため，仮想工場および攻撃検知システムのセキュリティを担保することは工場ネットワークと比較して容易であると考えられる．

攻撃検知のアルゴリズムの流れを**図 3** に示す擬似コードに基づき説明する．$C_a$ は実製造コマンド，$C_v$ は予測製造コマンドであり，それぞれ，$m$，$i$，$n$，$p$ の組がネットワーク上で直列化された順序が保存されている順序集合である．$M$ は製造コマンドの宛先の集合，$I$ は製品の ID の集合，$N$ は製造コマンド名の集合，$P$ は製造コマンドのパラメータの集合である．$C_a$，$C_v$ に対して，製造コマンドの宛先ごとに宛先数 $|M|$ 個分の初期値 $\emptyset$ を持ち，要素を追加した場合に順序が保存される順序集合 $S_a$，$S_v$ を定義しておく．

まず，ステップ 1 では，$C_v$ について先頭から順に取り出し，宛先 $m$ ごとに到達順を付与する．5 行目の $push(S_v[m], c_v)$ は，$c_v$ を $S_v[m]$ の末尾に追加する処理である．ここで，$S_v[m]$ は，$S_v$ より宛先 $m$ の到達順の順序集合を取得する処理である．この処理により，$S_v[m]$ は，**図 4** に示すような宛先 $m$ ごとに到達順に整列された予測製造コマンドの

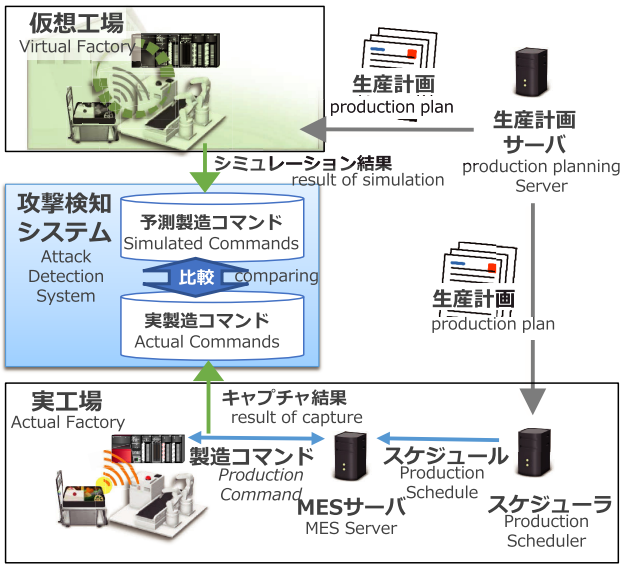


図 2　攻撃検知システム
**Fig. 2**　The attack detection system.

**Algorithm 1**
**Input**: $C_a, C_v \subset \{\{m, i\ n, p\} | m \in M, i \in I, n \in N, p \in P\}$
**Output**: true when arrival order matchs , otherwise false.
1: $S_a[m], S_v[m] = \{\emptyset\}, m \in M$
2:
3: // step1: assign arrival order for each controller
4: **for** each $c_v \in C_v$ **do**
5:　$push(S_v[m], c_v)$
6: **end for**
7:
8: // step2: compare commands
9: **for** each $c_a \in C_a$ **do**
10:　$push(S_a[m], c_a)$
11:　$l = length(S_a[m])$
13:　**if** $l > length(S_v[m])$ **then**
14:　　**return** false
15:　**end if**
16:　$c_v = S_v[m][l]$
17:　**if** $c_a \neq c_v$ **then**
18:　　**return** false
19:　**end if**
20: **end for**
21: **return** true

図 3　攻撃検知のアルゴリズム
**Fig. 3**　The algorithm for attack detection.

© 2019 Information Processing Society of Japan

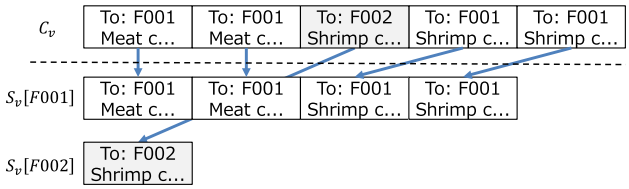


図 4　宛先別の到達順に整列された予測製造コマンド $S_v$ の例

**Fig. 4**　An example for ordered commands.

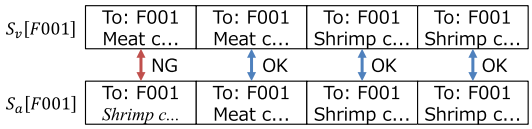


図 5　攻撃検知の例

**Fig. 5**　An example for attack detection.

順序集合となる．

次に，ステップ 2 では，$C_a$ について先頭から順に取り出し，宛先 $m$ ごとに到達順を付与し，**図 5** に示すように同じ到達順の予測製造コマンドと比較することで攻撃検知を行う．10 行目の $push(S_a[m], c_a)$ は，$c_a$ を $S_a[m]$ の末尾に追加する処理である．この処理により，$S_a[m]$ は，宛先 $m$ ごとに到達順に整列された実製造コマンドの順序集合となる．11 行目の $length(S_a[m])$ は，$S_a[m]$ に含まれる製造コマンドの数を取得する処理であり，$l$ は $c_a$ の到達順となる．この値を 12 行目で同様に取得した $S_v[m]$ に含まれる製造コマンドの数と比較することで，予測された製造コマンドよりも実際に多くの製造コマンドが到達した場合に攻撃により製造コマンドが追加されたことを検知できる．16 行目では，$c_v$ に $S_a[m][l]$ を代入することで，$c_a$ と同じ到達順の予測製造コマンドを取得している．ここで，$S_a[m][l]$ は，$S_a[m]$ より到達順 $l$ の製造コマンドを取得する処理である．17 行目で $c_a$ と $c_v$ とを比較し，$m$，$i$，$n$，$p$ の組が異なる場合は，攻撃により製造コマンドが書き換えまたは削除されたことを検知できる．図 3 ではバッチ処理として実現する流れを示しているが，製造コマンドの宛先ごとの到達順をデータベース等の外部記憶に記憶させておくことで，実工場から製造コマンドがキャプチャされるたびに逐次処理することも可能である．

## 4.　フィージビリティスタディ

本章では，製造コマンドが書き換えられる場合，製造コマンドが追加される場合，製造コマンドが破棄される場合の各具体例に基づき本方式のフィージビリティスタディを行う．**表 1** に示す予測製造コマンドの例を用いて，次節以降の説明を行う．表中の address は製造コマンドの宛先，pid は製品 ID，command は製造コマンド名，parameter は製造コマンドのパラメータ，order は宛先ごとの到達順を示している．簡単のため到達順の付与は，図 3 のアルゴリズムを用いて行っているものとする．

表 1　予測製造コマンドの例

**Table 1**　An example of predicted manufacturing commands.

| address | pid | command | parameter | order |
|---|---|---|---|---|
| F001 | P001 | start | Meat croquette, ... | 1 |
| F001 | P001 | end | Meat croquette, ... | 2 |
| F002 | P001 | start | Shrimp croquette, ... | 1 |
| F001 | P002 | start | Shrimp croquette, ... | 3 |
| F001 | P002 | end | Shrimp croquette, ... | 4 |

表 2　製造コマンドが書き換えられた実製造コマンドの例

**Table 2**　An example of rewrited manufacturing commands.

| address | pid | command | parameter | order |
|---|---|---|---|---|
| F001 | P001 | start | *Shrimp croquette*, ... | 1 |
| F001 | P001 | end | Meat croquette, ... | 2 |
| F002 | P001 | start | Shrimp croquette, ... | 1 |
| F001 | P002 | start | Shrimp croquette, ... | 3 |
| F001 | P002 | end | Shrimp croquette | 4 |

表 3　製造コマンドが追加された実製造コマンドの例

**Table 3**　An example of pushed manufacturing commands.

| address | pid | command | parameter | order |
|---|---|---|---|---|
| F001 | P001 | start | Meat croquette, ... | 1 |
| F001 | P001 | end | Meat croquette, ... | 2 |
| F002 | P001 | start | Shrimp croquette, ... | 1 |
| F001 | P002 | start | Shrimp croquette, ... | 3 |
| F001 | P002 | end | Shrimp croquette, ... | 4 |
| *F001* | *P001* | *start* | *Shrimp croquette, ...* | *5* |

### 4.1　製造コマンドが書き換えられた場合の検知

**表 2** に示す製造コマンドが書き換えられた実製造コマンドの例を用いて，製造コマンドが書き換えられた場合の検知について検証する．書き換えられた実製造コマンドの 1 行目 {F001, P002, end, “Shrimp croquette, ...”} と対応する予測製造コマンド {F001, P002, end, “Meat croquette, ...”} を比較すると，パラメータがミートコロッケからエビコロッケに変化していることが分かる．このとき，図 3 の 17 行目の比較結果が真になるため，製造コマンドの書き換えが検出できる．

### 4.2　製造コマンドが追加された場合の検知

次に，**表 3** に示す製造コマンドが追加された実製造コマンドの例を用いて，製造コマンドが追加された場合の検知について検証する．6 行目に追加された製造コマンド {F001, P001, start, “Shrimp croquette, ...”} の到達順は 5 となるが，表 1 を見ると予測では到達順 4 が最大である．よって，図 3 の 13 行目の比較結果が真になるため，製造コマンドの追加が検出できる．また，末尾ではなく途中に挿入された場合は，到達順が同じ予測製造コマンドと比較することで書き換えと同様に検知できる．

© 2019 Information Processing Society of Japan

表 4　製造コマンドが破棄された実製造コマンドの例
**Table 4**　An example of deleted manufacturing commands.

| address | pid | command | parameter | order |
|---|---|---|---|---|
| F001 | P001 | end | Meat croquette, ... | *1* |
| F002 | P001 | start | Shrimp croquette, ... | 1 |
| F001 | P002 | start | Shrimp croquette, ... | *2* |
| F001 | P002 | end | Shrimp croquette, ... | *3* |

### 4.3　製造コマンドが破棄された場合の検知

最後に，**表 4** に示す製造コマンドが破棄された実製造コマンドの例を用いて，製造コマンドが破棄された場合の検知について検証する．1 行目の実製造コマンド {F001, P001, end, “Meat croquette, ...”} と製造コマンドの宛先が同一で到達順が 1 の予測製造コマンド {F001, P001, start, “Meat croquette, ...”} とを比較すると製造コマンドの名前およびパラメータが異なることから製造コマンドの破棄が検出できている．

## 5.　おわりに

本稿では，つながる工場における製造コマンドの改ざんにより引き起こされる製造不良に着目し，改ざんを検知するための攻撃検知方式を提案し，具体例を用いてフィージビリティを確認した．今後は，具体的な工場を想定した評価を行い，製造不良により製造が停止した場合に行う再スケジューリングへの対応，設備の劣化・設備の内部パラメータ改ざん等製造コマンドだけでは検知が困難な事象についても検知可能な方式を検討していく．

**謝辞**　本研究は，総合科学技術・イノベーション会議が主導する革新的研究開発推進プログラム（ImPACT）の一環として実施したものです．

### 参考文献

[1] Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., Kim, B.H. and Noh, S.D.: Smart Manufacturing: Past Research, Present Findings, and Future Directions, *International Journal of Precision Engineering and Manufacturing-Green Technology*, Vol.3, No.1, pp.111–128 (2016).
[2] 社会リスクを低減する超ビッグデータプラットフォーム 2017 年度シンポジウム：パネル討論（オンライン），入手先 ⟨https://www.jst.go.jp/impact/hp_hharada/panel_discussion2017_03.html⟩（参照 2018-06-06）.
[3] Langner, R.: Stuxnet: Dissecting a Cyberwarfare Weapon, *IEEE Security & Privacy*, Vol.9, No.3, pp.49–51 (2011).
[4] Lopez, F., Saez, M., Shao, Y., Balta, E.C., Moyne, J., Mao, Z.M., Barton, K. and Tilbury, D.: Categorization of Anomalies in Smart Manufacturing Systems to Support the Selection of Detection Mechanisms, *IEEE Robotics and Automation Letters*, Vol.2, No.4, pp.1885–1892 (2017).
[5] Kletti, J.: *Manufacturing Execution System-MES*, Springer Science & Business Media (2007).
[6] Deroussi, L., Gourgand, M. and Tchernev, N.: A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles, *International Journal of Production Research*, Vol.46, No.8, pp.2143–2164 (2008).
[7] Lee, D.Y. and DiCesare, F.: Scheduling flexible manufacturing systems using Petri nets and heuristic search, *IEEE Trans. Robotics and Automation*, Vol.10, No.2, pp.123–132 (1994).
[8] Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T. and Lennartson, C.: An Event-Driven Manufacturing Information System Architecture for Industry 4.0, *International Journal of Production Research*, Vol.55, No.5, pp.1297–1311 (2017).
[9] 株式会社サイバー創研：ネットワークセキュリティ講義ノート（オンライン），入手先 ⟨https://www.ipa.go.jp/files/000018640.doc⟩（参照 2018-05-30）.

**立床 雅司**　（正会員）

2010 年京都大学大学院情報学研究科修士課程修了．同年三菱電機株式会社入社．データ分析技術，データベース技術の研究開発に従事．

**樋口 毅**　（正会員）

1990 年東北大学工学部卒業．同年三菱電機株式会社入社．情報技術総合研究所において情報処理技術の研究・開発に携わり現在に至る．機器の予防保全や最適化問題等の研究開発に従事．

**河内 清人**

1996 年慶應義塾大学大学院理工学研究科修了．同年三菱電機株式会社入社．現在，情報技術総合研究所で，情報セキュリティ技術に関する研究・開発に従事．

**永谷 達也**

2006 年東京工業大学大学院総合理工学研究科物理情報システム創造専攻博士前期課程修了．同年三菱電機株式会社入社．先端技術総合研究所研究員，現在に至る．産業用ロボット，FA 加工機，FA コントローラ等の研究開発に従事．計測自動制御学会，システム制御情報学会各会員．

© 2019 Information Processing Society of Japan

古澤 康一

1995 年大阪大学大学院物理系専攻情報工学分野博士前期課程修了．同年三菱電機株式会社入社．先端技術総合研究所，名古屋製作所，情報技術総合研究所において技術研究・開発に携わり，現在に至る．FA コントローラやエンジニアリング環境等の研究開発に従事．日本機械学会，電気学会各会員．

米田 健　（正会員）

1994 年慶應義塾大学大学院理工学研究科博士課程修了．博士（工学）．同年三菱電機株式会社入社．インフォメーションシステム事業推進本部情報セキュリティ推進センターセンター長，インフォメーションシステム事業推進本部連携事業推進センター DIGUARD 事業グループグループマネージャーを経て，2014 年より情報技術総合研究所情報セキュリティ技術部部長．情報セキュリティ，制御セキュリティの研究開発プロジェクトに従事．

（担当編集委員　石原 靖哲）

© 2019 Information Processing Society of Japan