Efficient Control Recovery for Resilient Control Systems

Shuo Zhang
School of Mathematics and Information Security
Royal Holloway University of London
Egham UK TW20 0EX
Email: MYVA375@live.rhul.ac.uk

Stephen D. Wolthusen
School of Mathematics and Information Security
Royal Holloway University of London
Egham UK TW20 0EX
Email: stephen.wolthusen@rhul.ac.uk

***Abstract*—Resilient control systems should efficiently restore control into physical systems not only after the sabotage of themselves, but also after breaking physical systems. To enhance resilience of control systems, and given an originally minimal-input controlled linear-time invariant(LTI) physical system, we address the problem of efficient control recovery into it after removing a known system vertex by finding the minimum number of inputs. According to the minimum input theorem, with a digraph embedded into LTI model and involving a precomputed maximum matching, this problem is modeled into recovering controllability of it after removing a known network vertex. Then, we recover controllability of the residual network by efficiently finding a maximum matching rather than recomputation. As a result, except for precomputing a maximum matching and predetermining the removed vertex, the worst-case execution time of control recovery into the residual LTI physical system is linear.**

I. Introduction

Control systems [1] can effectively sense, compute, communicate and control physical systems automatically [2] [3], which are also called process control systems, cyber-physical systems, distributed control system and supervisory control and data aquision systems in different applications [4]. Although breaking control systems damages its control into physical systems [5] [2] [6] [7], control systems would also lose its control due to sabotage of the physical system. Therefore, to maintain a resilient control system against malicious attack or random failure on a single system vertex of a physical system, it is desirable to recover the controllability of the residual physical system with high efficiency.

In this paper, we therefore address the problem of efficiently recovering control into a LTI physical system after removing an already known vertex, which is initially conrtrolled by the minimal inputs. To model this problem, a large, sparse, *Erdős-Rényi* random digraph in LTI model and also including a precomputed maximum matching, is considered as an input network. By the minimum input theorem [8], since the minimum number of inputs to fully control a digraph in LTI model is determined by the maximum (cardinality) matching, our problem is modeled into efficient recovery of controllability of an input network after removing a known vertex. And this problem is solved by identifying a maximum matching of this residual network so that the minimum number of inputs and the system vertices directly driven by those inputs would be clarified finally. Besides, the removed vertex can be predetermined by nodal importance in some aspects, such as the vertex degree [9] [10]. In particular, we do not find a maximum matching of the residual digraph caused by a nodal removal, instead, we efficiently find a maximum matching of the input digraph in linear time, which contains the minimum number of arcs incident to the following removed node. Since such a maximum matching is proved to directly determine the maximum matching of the residual network, recomputation of a maximum matching of residual network is effectively avoided.

For our contribution, with a minimal-input controlled LTI physical system and a known system vertex, control recovery into this system after removing that known vertex is executed in linear time by efficiently finding a maximum matching of a relative digraph after a nodal removal. Besides, given a general digraph, we also proved that the maximum matching of it after a known nodal removal can be efficiently identified through another maximum matching of it before the nodal removal.

Remaining paper is outlined as follows: Section II introduces control theory and the minimum input theorem; Section III reviews previous work of control recovery; Section IV shows our recovery method and section V concludes this paper.

II. Control Theory & Minimum Input Theorem

In modern control theory [11] [12] [13], a linear-time invariant (LTI) system can be described by a differential equation:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a matrix, representing the interaction among N system vertices; matrix $\mathbf{B} \in \mathbb{R}^{N\times M}$ represents the impact of M external inputs on the N system vertices. Also, vector $x(t) = (x_1(t), x_2(t), \dots, x_N(t))^T$ captures the state of every system vertex at time t; while vector $u(t) = (u_1(t), u_2(t), \dots, u_M(t))^T$ indicates the M external inputs at time t, which directly drive M system vertices of N. When system described by equation 1 is controllable, if and only if the matrix $\mathbf{C} \in \mathbb{R}^{N\times NM}$, where $\mathbf{C} = [\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \dots, \mathbf{A}^{N-1}\mathbf{B}]$, has the full rank[11], noted by $rank(\mathbf{C}) = N$ and called the controllability rank condition.

However, value of each non-zero entry of matrix $\mathbf{A}$ and $\mathbf{B}$ is known by approximation in parctice [14]. Besides, calculating the rank of matrix $\mathbf{C}$ from $\mathbf{A}$ and $\mathbf{B}$ is also computationally massive in $O(2^N)$ time [8]. To avoid those two constrains and still verify whether a LTI system is controllable or not, Lin et al. [14] [15] put forward the structural controllability, which is defined:

Definition 1 (**Lin's Structural Controllability** [14]). *A system of equation 1 is structurally controllable if a completely controllable system with the same structure as it exists.*

To investigate conditions of structurally controllable systems, a system described by equation 1 is mapped into a digraph noted by $G(\mathbf{A}, \mathbf{B}) = (V_1 \cup V_2, E_1 \cup E_2)$. In detail, with a bijection λ, for any non-zero entry noted by $a_{ij} \in \mathbf{A}$, there is $\lambda : a_{ij} \to \overrightarrow{\langle v_j, v_i \rangle}$, where $\overrightarrow{\langle v_j, v_i \rangle} \in E_1$ $v_i, v_j \in V_1$. For any non-zero entry of $\mathbf{B}$ noted by $b_{ij} \in \mathbf{B}$, there is $\lambda : b_{ij} \to \overrightarrow{\langle u_j, v_i \rangle}$, where $\overrightarrow{\langle u_j, v_i \rangle} \in E_2$ and $u_j \in V_2$, $v_i \in V_1$. Then, $G(\mathbf{A}, \mathbf{B})$ is used to give conditions of structural controllability with following items:

Definition 2 (**Inaccessibility** [14]). *In $G(\boldsymbol{A}, \boldsymbol{B})$, any $v_i \in V_1$ is inaccessible if it can not be approached from any vertex of V_2.*

Definition 3 (**Dilation of Digraphs** [14]). *In $G(\boldsymbol{A}, \boldsymbol{B})$ with vertex sets $S \subseteq V_1$, and $T(S) = \{v_j | \overrightarrow{\langle v_j, v_i \rangle} \in E_1 \cup E_2, v_i \in S\}$, $G(\boldsymbol{A}, \boldsymbol{B})$ contains a dilation iff $|S| > |T(S)|$.*

Definition 4 (**Stem, Bud** [14]). *In $G(\boldsymbol{A}, \boldsymbol{B})$, a stem is a directed path. A bud is a directed cycle pluse an arc such as $\{\{\overrightarrow{\langle v_1, v_2 \rangle}, \overrightarrow{\langle v_2, v_3 \rangle}, \ldots, \overrightarrow{\langle v_j, v_1 \rangle}\}, \overrightarrow{\langle v_{j+1}, v_j \rangle}\}$, where $\overrightarrow{\langle v_{j+1}, v_j \rangle}$ is called a distinguished edge.*

Definition 5 (**Cactus** [14]). *By definition 4, any stem is a cactus. Given a stem S_0 and buds $B_1, B_2, \ldots, B_l$, then, $S_0 \cup B_1 \cup B_2 \cup \ldots B_l$ is a cactus if the tail of the distinguished edge of any B_i $(1 \leq i \leq l)$ is not the top vertex of S_0 but is the only common vertex of $S_0 \cup B_1 \cup B_2 \cup \ldots B_{i-1}$. A set of vertex-disjoint cacti is called a cactus.*

After this, conditions of structurally controllable systems are given below:

Theorem 1 (**Lin's Structural Controllability Theorem** [14]). *The following three statements are equivalent:*

1) *A system of equation 1 is structurally controllable.*
2) *The digraph $G(\boldsymbol{A}, \boldsymbol{B})$ contains neither inaccessible nodes nor dilation.*
3) *$G(\boldsymbol{A}, \boldsymbol{B})$ is spanned by a cactus.*

Based on the controllability rank condition, almost all structrually controllable systems are also controllable except for some pathological cases with certain constrains [14], [15]. As an instance of a pathological case, a system of equation 1 includes five system vertices $\{v_1, v_2, v_3, v_4\}$, one external input u_1, and system edges: $\{\overrightarrow{\langle u_1, v_1 \rangle}, \overrightarrow{\langle v_1, v_2 \rangle}, \overrightarrow{\langle v_2, v_3 \rangle}, \overrightarrow{\langle v_3, v_2 \rangle}, \overrightarrow{\langle v_3, v_4 \rangle}\}$. Obviously, such a system is structurally controllable because it contains neither inaccessible nodes nor dilation according to the theorem 1. But this system can not be completely controllable if weight of any edge is positive one, because the rank of matrix $\mathbf{C}$ is less than four. Besides, whether the numebr of inputs is minimal in a system of equation 1 was still pending over decades. In recent years, Liu et al. [8] effectivly solved this problem via the maximum matching of digraphs, which is generalised by the following theorem:

Theorem 2 (**The Minimum Input Theorem** [8]). *The minimum number of inputs to fully control a digraph such as $G(\boldsymbol{A}) = (V_1, E_1)$ is one, if there is a perfect matching. Otherwise, it equals the number of unmatched nodes according to a maximum matching.*

In general digraphs, a matching is a set of edges without sharing heads or tails. A maximum matching is a matching out of any other ones. A digraph may contain multiple maximum matchings. The head of an arc involved into a maximum matching is a matched node, otherwise, it is unmatched. When the number of matched nodes is same as the number of vertices, this digraph has a perfect matching [16]. Nonetheless, by contrast, in a directed bipartite graph, a maximum matching is a set of maximum number of completely vertex-disjoint edges in this paper.

By theorem 1 and the minimum input theorem [8], efficient control recovery of an originally minimal-input controlled LTI physical system after a known vertex removal, can be modeled into efficiently recovering the controllability of a digraph in LTI model with a precomputed maximum matching after removing a known vertex. And efficiently identifying a maximum matching of the residual digraph can be as a solution.

III. Related Work

Control recovery of complex networks attracts increasing attentation. People do it via the maximum matching or via the power dominating set [17]. On the one hand, Ding et al. [18] designed a scenario by adding extra nodes or edges to the residual digraph after malicious attack or random failure on network structure, involving recomputing a maximum matching of the residual graph. On the other hand, because structurally controllableity is the necessary but not sufficient condition of complete controllability, there are also scenarios recovering structural controllability [14]. Alwasel et al. in [19] [20] [21] recovered the structural controllability of the *Erdős-Rényi* random digraph in LTI model after removing vertices by maintaining an approximated power dominating set [17] with the average-case time complexity $O(nc^k)$ according to original research in [22] [23], where c is a constant number, with the remaining digraph, n is the number of nodes and k is the number of partitioned subgraphs. Besides, Alcaraz et al. [24] still relies on the power dominating set to recover structural controllability [14] of general power-law and scale-free digraphs in LTI model without recomputation of a new power dominating set. Three repair strategies are given against deletion of nodes and edges respectively, which are re-linking existing dominating nodes with unobserved nodes; re-linking with constrained network diameter; and using pre-computed instances. Particularly, the first two strategies are executed in $O(|U||V|^2)$, in which U is the set of nodes out of remaining dominating set and V is the set of all vertices. From those previous work, control recovery can be effectively executed after malicious attack or random failure on system vertices via several ways in polynominal time. By contrast, we obtain the controllability of networks in LTI model by precomputing a maximum matching so that the inputs would be minimal. And out control recovery into the network after a known nodal removal is executed by finding a maximum matching efficiently to confirm the minimal inputs rather than recomputation of a maximum matching.

Reviewing existing maximum matching algorithms is indispensable for us. To identify a maximum matching of a general bipartite graph, the Hopcroft-Karp algorithm [25] is well-known. With n and m as the number of nodes and edges of a given bipartite graph, the worst-case execution time of Hopcroft-Karp algorithm is $O(m\sqrt{n})$. When the bipartite graph is dense, time complexity is improved into $O(m\sqrt{n/logn})$ [26]. By contrast, for computing a maximum matching of general undirected graphs, Micali and Vazirani [27] also approached the same time complexity as the Hoproft-Karp algorithm. Until now, for the worst-case execution time to computate an exact maximum matching of general graphs via the deterministic algorithms, these two algorithms are the best [28]. Nevertheless, concerning the time complexity, these two algorithms should be used as few times as possible in multiply identifying a maximum matching of networks. In consequence, efficiently identifying a maximum matching of a digraph after a node deletion rather than recomputation is worth of being considered as a way to efficiently recover controllability of a resilient physical system. To avoid recomputation of a maximum matching of the residual network, we firstly identify a maximum matching containing the minimum number of edges incident to the following removed node through the known one. Then, we conclude that a maximum matching of the residual digraph is determined by it without recomputation.

IV. Control Recovery via Maximum Matching

A. Preliminaries

Firstly, an input digraph is defined:

Definition 6 (**Input Digraph**). *A large, sparse* Erdős-Rényi *random digraph $D = (V, A)$ in LTI model, including a precomputed maximum matching M_D by using algorithm [25] or [27], and excluding parallel arcs, isolated nodes and selfloops, is determined as an input digraph, where $V = \{v_i | 1 \leq i \leq N\}(N > 1)$, $A = \{\overrightarrow{\langle v_i, v_j\rangle} | 1 \leq i, j \leq N, i \neq j\}$.*

The removed vertex is noted by u and $u \in D$, and by the minimum input theorem [8], problem of control recovery into a $D - u$ that models control revocery into a LTI physical system after a system nodal removal, is solved by efficiently identifying a maximum matching of $D - u$.

Secondly, a maximum matching noted by $MM \subseteq D$ involving the minimum number of edges incident to u, where $l \in \{0, 1, 2\}$ represents that minimum number. Then, following lemmas concludes how MM of D determines the maximum matching of $D - u$. In notation, $\overrightarrow{\langle v_g, u\rangle}$ and $\overrightarrow{\langle u, v_f\rangle}$ note any two arcs incident to u. $M_{(v_g,v_f)}$ notes a matching incident to v_g, v_f and out of MM, while $MM_{(v_g,v_f)}$ notes the matching of MM adjacent to $M_{(v_g,v_f)}$ in $D - u$.

Lemma 1. *Within $D = (V, A)$ of definition 6, when either $l < 2$, or $l = 2$ and $\nexists M_{(v_g,v_f)}$. The maximum matching of $D - u$ is the remaining matching of MM.*

Proof. When $l = 0$, $u \notin MM$, removal of u can not influence MM, so MM is also a maximum matching of $D - u$. When $l = 1$, after removal of u, matching either $\{MM - \overrightarrow{\langle v_g, u\rangle}\}$ or $\{MM - \overrightarrow{\langle u, v_f\rangle}\}$ is obtained. If it is not maximal in $D - u$. At least an arc not incident to this matching could exist; or, a matching out of MM incident to either v_g or v_f and share heads or tails with this remaining matching could exist, which contradicts with maximality of M_D or $l = 1$. Thus, $\{MM - \overrightarrow{\langle v_g, u\rangle}\}$ or $\{MM - \overrightarrow{\langle u, v_f\rangle}\}$ is a maximum matching of $D - u$. When $l = 2$ and $\nexists M_{(v_g,v_f)}$, $\{\overrightarrow{\langle v_g, u\rangle}, \overrightarrow{\langle u, v_f\rangle}\} \subseteq MM$. For the same reason of $l = 1$, $\{MM - \overrightarrow{\langle v_g, u\rangle} - \overrightarrow{\langle u, v_f\rangle}\}$ is a maximum matching of $D - u$. □

Lemma 2. *Within $D = (V, A)$ of definition 6, when $l = 2$ and $\exists M_{v_g,v_f}$, $\{\{MM - MM_{(v_g,v_f)}\}, M_{(v_g,v_f)}\}$ is a maximum matching of $D - u$.*

Proof. When $l = 2$, and $\exists M_{(v_g,v_f)}$, MM includes $\overrightarrow{\langle v_g, u\rangle}$ and $\overrightarrow{\langle u, v_f\rangle}$. In $D - u$, since $|M_{(v_g,v_f)}| - |MM_{(v_g,v_f)}| = 1$, we can thus obtain a matching: $\{\{MM - MM_{(v_g,v_f)}\}, M_{(v_g,v_f)}\}$. For the same reason of $l = 1$, based on the maximality of M_D or the condition of $l = 2$, this matching is a maximum matching in $D - u$. □

By lemma 1, 2, a maximum matching of $D - u$ can be obtained without recomputation, while MM should be identified in D. To do this, a bipartite graph $B = (V_B, E_B)$ mapped from D is used and defined:

Definition 7 (**Bipartite graph** B). *With two bijection α, β and $D = (V, A)$ of definition 6, a bipartite graph $B = (V_B, E_B)$, $|E_B| = |A|$, $V_B = V_B^+ \cup V_B^-$ is obtained. For any $\overrightarrow{\langle v_i, v_j\rangle} \in A - M_D$, $\alpha : \overrightarrow{\langle v_i, v_j\rangle} \to \overrightarrow{\langle v_i^+, v_j^-\rangle}$, while for any $\overrightarrow{\langle v_p, v_q\rangle} \in M_D$, there is $\beta : \overrightarrow{\langle v_p, v_q\rangle} \to \overrightarrow{\langle v_q^+, v_p^-\rangle}$, where $\overrightarrow{\langle v_q, v_p\rangle} \in M_B$, $\overrightarrow{\langle v_i^+, v_j^-\rangle} \in E_B - M_B$, and $v_i^+, v_q^+ \in V_B^+$, $v_j^-, v_p^- \in V_B^-$.*

By definition 7, M_B must be a maximum matching of B, and $\nexists \overrightarrow{\langle v_i^-, v_i^+\rangle}$ or vice-versa. Otherwise, maximality of M_D and D excluding selfloops would be contradicted. Given any $\{\overrightarrow{\langle u, v_f\rangle}, \overrightarrow{\langle v_g, u\rangle}\} \subseteq M_D$, there are $\{\overrightarrow{\langle v_f^-, u^+\rangle}, \overrightarrow{\langle u^-, v_g^+\rangle}\} \subseteq M_B$. Thus, identifying $MM \subseteq D$ can be solved by finding a maximum matching noted by MM_B of B that contains the minimum number of edges incident to u^+ and u^-. To find MM_B, few kinds of defined matching sets of [29] are re-defined and used by us in following:

Definition 8 (**Replacing Link**). *In $B = (V_B, E_B)$ of definition 7, with respect to M_B, any single edge $e_i = \overrightarrow{\langle v_i^+, v_j^- \rangle}$ is a replacing link if either $v_i^+ \in M_B$ and $v_j^+ \notin M_B$ or vice-versa.*

Definition 9 (**Uncrossed Replacing set**). *In $B = (V_B, E_B)$ of definition 7, with $t(1 \leq t \leq |M_B|)$ distinct edges $\{m_1, m_2, \ldots, m_t\} \subseteq M_B$, a set of distinct t edges $\{e_1, e_2, \ldots, e_t\} \subseteq E_B - M_B$ is an uncrossed replacing set, if either $e_i \cap m_i \in V_B^+ \cap M_B (1 \leq i \leq t)$, $e_j \cap m_{j+1} \in V_B^- \cap M_B (1 \leq j < t)$, and $e_t \cap \{V_B^- \cap M_B\} \in \emptyset$. OR, $e_i \cap m_i \in V_B^- \cap M_B (1 \leq i \leq t)$, $e_j \cap m_{j+1} \in V_B^+ \cap M_B (1 \leq j < t)$, and $e_t \cap V_B^+ \cap M_B \in \emptyset$.*

Definition 10 (**Crossed Replacing Set**). *In $B = (V_B, E_B)$ of definition 7, with $t(1 \leq t \leq |M_B|)$ distinct edges $\{m_1, m_2, \ldots, m_t\} \subseteq M_B$, a set of distinct t edges $\{e_1, e_2, \ldots, e_t\} \subseteq E_B - M_B$ is crossed replacing set, if either $e_i \cap m_i \in V_B^- \cap M_B (1 \leq i < t)$, $e_j \cap m_{j+1} \in V_B^+ \cap M_B (1 \leq j < t)$, and $e_t \cap m_1 \in V_B^+ \cap M_B$, $e_t \cap \{V_B^- \cap M_B\} \in \emptyset$. OR, $e_i \cap m_i \in V_B^+ \cap M_B (1 \leq i < t)$, $e_j \cap m_{j+1} \in V_B^- \cap M_B (1 \leq j < t)$, and $e_t \cap m_1 \in V_B^- \cap M_B$, $e_t \cap \{V_B^+ \cap M_B\} \in \emptyset$.*

Tassa $et\ al.$ [29] proved that any matching set out of M_B that are used to produce a different maximum matching from M_B in involved nodes by edge replacement, must be classified into these three kinds of edge sets of definition 8-10. Besides, we deduce the distribution of these matching sets:

Lemma 3. *In $B = (V_B, E_B)$ of definition 7, with respect to M_B, given any two valid replacing sets of definition 8-10 incident to any $v_i^+ \in \{V_B^+ - V_B^+ \cap M_B\}$, $v_j^- \in \{V_B^- - V_B^- \cap M_B\}$ respectively. Then, they can not share any vertex.*

Proof. Assuming there is a vertex shared by any two valid replacing sets of definition 8-10, incident to $v_i^+ \notin M_B$ and $v_j^- \notin M_B$ respectively. Then, in $B = (V_B, E_B)$ with respect to M_B, there must be a path starting from this shared vertex and ending at a node out of M_B and alternatively involving arcs out of and of M_B, while there must be another path ending at this shared vertex and starting from V_B^- or V_B^+ out of M_B and alternatively contains arcs of and out of M_B. Both of these two alternating pahts would produce a matching out of M_B and adjacent to a submatching of M_B, which is also bigger than this submatching of M_B in cardinality by one. In other words such matching can replace this submatching of M_B to obtain a matching bigger than M_B in cardinality by one. However, since M_B is already the maximum matching, a contradiction would exist otherwise. Consequently, any shared vertex by such two valid replacing link of definition 8 or replacing sets of definition 9, 10 can not exist. □

B. Find a Replacing Link or a Replacing edge Set

Following algorithm finds a replacing link, or a replacing sets incident to v_f^-, when $\overrightarrow{\langle v_f^-, u^+ \rangle} \in M_B$. In notation, $H(v_f^-)$ represents a set of arcs whose head is v_f^-, and any arc of it is noted by $a_i \in H(v_f^-)$. P_0 notes an arc set, $P(P_0)$ represents a set of arcs pointing arcs of P_0, in which any arc of it is noted by $a_j \in P(P_0)$. Also, P_n notes a traversed path of P_0 ending at v_f^- starting from a node of M_B. And $M_{sub} \subseteq M_B$ that is adjacent to the replacing sets at nodes of both V_B^- and V_B^+.

Algorithm 1 Find a replacing link or set incident to v_f^-

Input: $B = (V_B, E_B)$, M_B, $P_0 = \emptyset$, $v_f^- \in M_B$, u^+, u^-
Output: A replacing link or set incident to v_f^-
1: Label nodes involved into M_B **and** $V_B = V_B - u^+ - u^-$
2: **while** $H(v_f^-) \neq \emptyset$ **and** $a_i \in H(v_f^-)$ **do**
3: $\quad H(v_f^-) = H(v_f^-) - a_i$
4: $\quad P_0 = P_0 + a_i$ **and** Label tail of a_i with v_f^-
5: $\quad$**if** Tail of a_i out of M_B **then**
6: $\quad\quad$**return** a_i
7: $\quad$**while** $P(P_0) \neq \emptyset$ **and** $\exists a_j \in P(P_0)$ **do**
8: $\quad\quad E_B = E_B - a_j$
9: $\quad\quad$**if** Tail of a_j is not labeled with v_f^- **then**
10: $\quad\quad\quad$Label tail of a_j with v_f^- **and** $P_0 = P_0 + a_j$
11: $\quad\quad\quad$**if** a_j pointing P_n **and** tail of a_j out of M_B **then**
12: $\quad\quad\quad\quad$**return** $\{P_n, a_j\} - M_{sub}$

Proof. Nodes of M_B are all labelled in $O(|V_B|)$ time to know if a tail of any arc ending at v_f^- is in or out of M_B without searching it in M_B. Then, procedure finds paths alternatively involving arcs out of and of M_B from v_f^- to a node out of M_B. Due to arcs of M_B are known, after removing $M_{sub} \subseteq M_B$ in a found path, remaining arcs are a replacing set by definition 9, 10. Firstly, $a_i \in H(v_f^-)$ is selected and added into P_0 to consider if it is a replacing link by definition 8, and its tail is labeled with v_f^-. If so, it is returned. Otherwise, procedure keeps searching proposed paths from a_i in the **while** loop from line 7 to line 12. For current $a_j \in P(P_0)$ is an edge of M_B, and tail of a_j is not labeled with v_f^-, it is added into P_0. For $P_0 = \{a_i, a_j\}$, there is not proposed alternating path by now, and procedure keeps searching arcs pointing a_j to obtain proposed paths. During this process, any tail of a traversed arc pointing P_0 is verified, if it is labelled already, this arc is not added into P_0, otherwise, its tail is labeled and added into P_0. If $P(P_0) = \emptyset$ caused by removing each arc of E_B pointing P_0 in line, all nodes able to approach a_i through a directed path have been traversed. Once tail of an arc of $P(P_0)$ is out of M_B, a directed path P_n with this arc is obtained, which is our proposed. After removing all involved arcs of M_B, remaining matching of this path is a replacing edge set. After this, another arc of $H(v_f^-)$ is chosen and examed as a_i, while it is the only one arc of P_0 possibly pointed by nodes out of P_0 at this moment. When $H(v_f^-) = \emptyset$ caused by removal of its arcs in line 3, this procedure terminates, and replacinging sets incident to v_f^- can be obtained. Without precomputing M_B, since edge of E_B is added into P_0 once at most, total running time of this procedure is thus represented by $O(|V_B| + |E_B|)$. $\square$

When $\overrightarrow{\langle u^-, v_g^+ \rangle} \in M_B$, this algorithm can be modified to find replacing links and sets incident to v_g^+ in $O(|V_B|+|E_B|)$ time. In detail, $H(v_f^-)$ is replaced with $T(v_g^+)$ to note all arcs whose tail is v_g^+, $P(P_0)$ now represents a set of arcs pointed by arcs of P_0, P_n now is a traversed path starting from v_g^+ and ending at node of M_B. Also, pointing should be replaced by pointed by in line 12, and tail should be replaced with head.

C. Obtain maximum matching MM_B of $B = (V_B, E_B)$

After obtaining a replacing link or set incident to v_f^- and v_g^+ by algorithm IV-B, next algorithm obtains MM_B in $B = (V_B, E_B)$ of definition 7, which involves the minimum number of edges incident to u^+ and u^- under all possible conditions. In notation, $R_{v_f^-}$, $R_{v_g^+}$ represent any two replacing sets or links of definition 8-10 incident to v_f^- and v_g^+ respectively. $Adj(R_{v_f^-})$ and $Adj(R_{v_g^+})$ also denote the set of edges of M_B adjacent to $R_{v_f^-}$ and $R_{v_g^+}$ in B.

Algorithm 2 Find a maximum matching MM_B of B

Input: $B = (V_B, E_B)$, M_B $R_{v_f^-}$, $R_{v_g^+}$
Output: MM_B
if $\exists R_{v_f^-}$ **and** $\exists R_{v_g^+}$ **then**
  return $MM_B = \{\{M_B - Adj(R_{v_f^-}) - Adj(R_{v_g^+})\}, R_{v_f^-}, R_{v_g^+}\}$
else if $\nexists R_{v_g^+}$ **and** $\exists R_{v_f^-}$ **then**
  return $MM_B = \{\{M_B - Adj(R_{v_f^-})\}, R_{v_f^-}\}$
else if $\exists R_{v_g^+}$ **and** $\nexists R_{v_f^-}$ **then**
  return $MM_B = \{\{M_B - Adj(R_{v_f^-})\}, R_{v_g^+}\}$
else if $\nexists R_{v_f^-}$ **and** $\nexists R_{v_g^+}$ **then**
  return $MM_B = M_B$

Proof. When $\overrightarrow{\langle u^+, v_f^- \rangle} \in M_B$ or $\overrightarrow{\langle v_g^+, u^- \rangle} \in M_B$, by operating algorithm IV-B, existence of replacing sets or links incident to v_f^- and v_g^+ can be confirmed respectively, based on which, following procedure shows all possibilities of obtaining MM_B through M_B. Referring to the lemma 3, for any obtained $R_{v_f^-}$ and $R_{v_g^+}$, they must be vertex disjoint. Besides, in particular, when either $\nexists R_{v_f^-}$ or $\nexists R_{v_g^+}$, it is just possible that either $u^+ \in M_B$ or $u^- \in M_B$. By contrast, once $R_{v_f^-} \notin \emptyset$ or $R_{v_g^+} \notin \emptyset$, there must be $u^+ \in M_B$ or $u^- \in M_B$. Thus, for $u^+ \in M_B$ or $u^- \in M_B$, there would be $u^+ \notin MM_B$ or $u^- \notin MM_B$. Except for the precomputation of M_B and execution of algorithm IV-B, according to definition 8-10, this procedure obtaining MM_B is executed by all known information, which leads the total rnning time of this algorithm is $O(1)$. $\square$

D. Identify the maximum matching MM of $D = (V, A)$

After obtaining the maximum matching MM_B that contains the minimum number of edges incident u^- and u^+ by applying algorithm IV-B and algorithm IV-C in $B = (V_B, E_B)$ of definition 7, MM_B would be used to identify MM of $D = (V, A)$ of definition 6 according to the bijections α, β of definition 7. And the value of $l \in \{0, 1, 2\}$ would also be confirmed. The related algorithm is below:

Algorithm 3 Identify a maximum matching MM of D

Input: $D = (V, A)$, M_D, u
Output: MM
if $u \notin M_D$ **then**
  $MM = M_D$ **and** $l = 0$
else if $u \in MM$ **then**
  Mapping $D = (V, A)$ with M_D into $B = (V_B, E_B)$ with M_B by definition 7.
  Obtain MM_B by operating algorithm IV-B, IV-C.
  Mapping MM_B into MM through bijection α, β of definition 7.
Search arcs incident to u involved into MM
return MM **and** value of l

Proof. Firstly, this procedure finds u in M_D in $\Theta(|V|)$ to known if $u \in M_D$ or not. If $u \notin M_D$, by the property of MM, there must be $MM = M_D$ and $l = 0$. Otherwise, $D = (V, A)$ with precomputed M_D is mapped into $B = (V_B, E_B)$ with M_B through bijection α and β. Obviously, running time of bijection is $\Theta(|A|)$. Then, by definition 6 and 7, obtaining MM of D can be solved by finding MM_B of B by using algorithm 1, 2 in $O(|V| + |A|)$ time due to $|V_B| \leq 2|V|$ and $|E_B| = |A|$. Since identifying MM via MM_B is virtually a part of reverse process of mapping D into B, given MM_B, MM can be identified in $O(|A|)$ time. Searching arcs incident to u in MM in $O(|A|)$ time is to confirm value of l. Therefore, except for precomputation of M_D and u, worst-case execution of identifying MM that involves the minimum number of edges incident to u is $O(|V| + |A|)$. $\square$

E. Control recovery via a maximum matching of $D - u$

Eventually, in accordance of the lemma 1 and 2, the process of obtaining a maximum matching based on returned MM and vlaue of l by algorithm IV-D is expressed by the following procedure, so that the minimum number of inputs to fully control $D - u$ can be identified.

Algorithm 4 Find a maximum matching of $D - u$

Input: $D = (V, A)$, MM, l
Output: A maximum matching of $D - u$
if $l = 0$ **then**
  return MM
else if $l = 1$ **then**
  return $MM - \overrightarrow{\langle v_g, u \rangle}$ **or** $MM - \overrightarrow{\langle u, v_f \rangle}$
else if $l = 2$ **then**
  if v_g^+ not labeled with v_f^- in algorithm IV-B **then**
    return $MM - \overrightarrow{\langle v_g, u \rangle} - \overrightarrow{\langle u, v_f \rangle}$
  else if v_g^+ labeled with v_f^- in algorithm IV-B **then**
    return $\{\{MM - \overrightarrow{\langle v_g, u \rangle} - \overrightarrow{\langle u, v_f \rangle}\}M_{(v_g, v_f)}$

Proof. According to lemma 1, and lemma 2, when $l = 1$, a maximum matching of $D - u$ can be obtained immediately. When $l = 2$, in $B = (V_B, E_B)$, there must be $\{\overrightarrow{\langle v_f^-, u^+\rangle}, \overrightarrow{\langle u^-, v_g^+\rangle}\} \subseteq M_B$. By algorithm IV-B, since any node approaching v_f^- via a directed path has been labelled with v_f^-. Thus, when $l = 2$, whether $M_{(v_g, v_f)}$ exists or not, can be confirmed by checking if v_g^+ is labelled or not in B. If $\exists M_{(v_g, v_f)}$ in D, $M_{(v_g, v_f)}$ can be obtained by mapping a directed path that has been traversed in algorithm IV-B, starting from v_g^+, and ending at v_f^- of B in $O(|A|)$ time. Thus, time complexity of this procedure is $O(|A|)$ except for computing MM and l. □

With such a maximum matching of $D - u$, based on the minimum input theorem [8], the minimum number of inputs to fully control $D - u$ is identified. And further, the control into physical system that involving the digraph $D - u$ is therefore restored with the minimum number of inputs. For the worst-case execution time of entire process except for precomputation of M_D and u, it is represented by $O(|V| + |A|)$ according to algorithm IV-D and IV-E.

V. Conclusion

Resilient control systems should recover control into physical systems efficiently not only after malicious attack or random failure on themselves, but also after them on the physical systems. In this paper, based on the minimum input theorem [8], efficiently recovering the controllability of a physical system after removing a known system vertex is modeled by control recovery into a digraph involved into the LTI system after a known vertex removal, which is further solved by efficiently identifying a maximum matching of the digraph rather than recomputation. As a result, except for the precomputation of a maximum matching and the following removed node of the input network, control recovery into the modified physical system by removing a vertex can be eventually executed in linear time, which thus enhances the resilience of control systems against vertex modification compared with recomputation. For the future work, we propose to enhance the resilience of control systems against edge removal of the controllable LTI physical systems.

References

[1] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM, 2011, pp. 355–366.

[2] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 911–918.

[3] E. A. Lee, "Cyber physical systems: Design challenges," in *Object oriented real-time distributed computing (isorc), 2008 11th ieee international symposium on*. IEEE, 2008, pp. 363–369.

[4] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." in *HotSec*, 2008.

[5] J. Wan, A. Canedo, and M. A. Al Faruque, "Security-aware functional modeling of cyber-physical systems," in *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, pp. 1–4.

[6] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach," in *Resilient Control Systems (ISRCS), 2012 5th International Symposium on*. IEEE, 2012, pp. 55–62.

[7] N. Rashid, J. Wan, G. Quirós, A. Canedo, and M. A. Al Faruque, "Modeling and simulation of cyberattacks for resilient cyber-physical systems."

[8] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[9] C.-L. Pu, W.-J. Pei, and A. Michaelson, "Robustness analysis of network controllability," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 18, pp. 4420–4425, 2012.

[10] C. Alcaraz, E. E. Miciolino, and S. Wolthusen, "Structural controllability of networks for non-interactive adversarial vertex removal," in *International Workshop on Critical Information Infrastructures Security*. Springer, 2013, pp. 120–132.

[11] R. Kalman, "On the general theory of control systems," *Automatic Control, IRE Transactions on*, vol. 4, no. 3, pp. 110–110, 1959.

[12] D. Luenberger, "Introduction to dynamic systems: theory, models, and applications," 1979.

[13] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control*. prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[14] C. T. Lin, "Structural controllability," *Automatic Control, IEEE Transactions on*, vol. 19, no. 3, pp. 201–208, 1974.

[15] R. Shields and J. Pearson, "Structural controllability of multiinput linear systems," *IEEE Transactions on Automatic control*, vol. 21, no. 2, pp. 203–212, 1976.

[16] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs & digraphs*. CRC Press, 2010.

[17] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, "Domination in graphs applied to electric power networks," *SIAM Journal on Discrete Mathematics*, vol. 15, no. 4, pp. 519–529, 2002.

[18] J. Ding, Y.-Z. Lu, and J. Chu, "Recovering the controllability of complex networks," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10 894–10 901, 2014.

[19] B. Alwasel and S. D. Wolthusen, "Recovering structural controllability on erdős-rényi graphs via partial control structure re-use," in *International Conference on Critical Information Infrastructures Security*. Springer, 2014, pp. 293–307.

[20] ——, "Recovering structural controllability on erdős-rényi graphs in the presence of compromised nodes," in *International Conference on Critical Information Infrastructures Security*. Springer, 2015, pp. 105–119.

[21] ——, "Structural controllability analysis via embedding power dominating set approximation in erdhos-rènyi graphs," in *Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on*. IEEE, 2015, pp. 418–423.

[22] A. Aazami and K. Stilp, "Approximation algorithms and hardness for domination with propagation," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 3, pp. 1382–1399, 2009.

[23] J. Guo, R. Niedermeier, and D. Raible, "Improved algorithms and complexity results for power domination in graphs," *Algorithmica*, vol. 52, no. 2, pp. 177–202, 2008.

[24] C. Alcaraz and S. Wolthusen, "Recovery of structural controllability for control systems," in *International Conference on Critical Infrastructure Protection*. Springer, 2014, pp. 47–63.

[25] J. E. Hopcroft and R. M. Karp, "An n^5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on computing*, vol. 2, no. 4, pp. 225–231, 1973.

[26] H. Alt, N. Blum, K. Mehlhorn, and M. Paul, "Computing a maximum cardinality matching in a bipartite graph in time o (n 1.5 mlog n)," *Information Processing Letters*, vol. 37, no. 4, pp. 237–240, 1991.

[27] S. Micali and V. V. Vazirani, "An o (v— v— c— e—) algorithm for finding maximum matching in general graphs," in *Foundations of Computer Science, 1980., 21st Annual Symposium on*. IEEE, 1980, pp. 17–27.

[28] M. Gupta and R. Peng, "Fully dynamic (1+ e)-approximate matchings," in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013, pp. 548–557.

[29] T. Tassa, "Finding all maximally-matchable edges in a bipartite graph," *Theoretical Computer Science*, vol. 423, pp. 50–58, 2012.