Supervisory Control Based on Vector Synchronous Product of Automata

Yangzhou Chen

School of Electronic Information
and Control Engineering
Beijing University of Technology
100022 Beijing, P.R. China
yzchen@bjut.edu.cn

W. M. Wonham

Department of Electrical
and Computer Engineering
University of Toronto
M5S 3G4 Toronto, Ontario, Canada
wonham@control.toronto.edu

Abstract: The paper deals with the supervisory control problem based on the vector synchronous product (VSP) of automata. A necessary and sufficient condition for the existence of such a controller is given which depends on a new concept of vs-controllability. A construct called vector synchronous product with communication is proposed. In addition, isomorph and homomorph of two VSPs are defined. Simplified traffic examples are provided in illustration.

Keywords: discrete-event system, supervisory control, vector synchronous product, vs-controllability.

Yangzhou CHEN received the Bachelor degree in Mathematics from Hubei University (China) in 1984, and the Ph.D. degree in System Control from the St. Petersburg University (Russia) in 1994. From 1996 to 1998, he held a postdoctoral position in the School of Astronautics, Harbin Institute of Technology (China). Since 1998 he has been a faculty member with the Faculty of Electronic Information and Control Engineering of Beijing University of Technology, and became a professor in 2001. In 2005, he was a visiting professor with the Department of Electrical and Computer Engineering at the University of Toronto, Canada.
His present research interests include hybrid dynamic systems, discrete-event systems, and cooperative control of multiple vehicles, and intelligent transportation systems.

W. M. Wonham received the B. Eng. degree in engineering physics from McGill University in 1956, and the Ph.D. in control engineering from the University of Cambridge (U.K.) in 1961.
From 1961 to 1969 he was associated with several U.S. research groups in control. Since 1970 he has been a faculty member in Systems Control, with the Department of Electrical and Computer Engineering of the University of Toronto.
Wonham's research interests have included stochastic control and filtering, geometric multivariable control, and discrete-event systems. He is the author of "Linear Multivariable Control: A Geometric Approach" (Springer-Verlag: 3rd ed. 1985) and co-author (with C. Ma) of "Hierarchical Control of State Tree Structures" (Springer-Verlag: 2005).
Wonham is a Fellow of the Royal Society of Canada, a Life Fellow of the IEEE, and (2005) a Foreign Associate of the (U.S.) National Academy of Engineering. In 1987 he received the IEEE Control Systems Science and Engineering Award and in 1990 was Brouwer Medallist of the Netherlands Mathematical Society. In 1996 he was appointed University Professor in the University of Toronto, and in 2000 University Professor Emeritus.

1. Introduction

The supervisory control problem for discrete-event systems (DES) modeled as automata was introduced in the work of Ramadge and Wonham [3, 4]. In their theory, based on a new controllability concept, a supervisory controller is designed to confine the behavior of the closed-loop system to within specified legal bounds. In order to deal with more complex practical DES, e.g., multiple subsystems or multiple agents, a variety of architectures such as hierarchical, decentralized, and heterarchical, have been proposed. In most of this work, attention is paid mainly to the case where events occur serially. However, some important practical real-life DES exhibit simultaneous event occurrences, or consist of subsystems that operate in strict concurrency. Accordingly, DES models with concurrent event or vector event occurrences have been introduced and their supervisory control problems studied [1,2,5,6]. Li and Wonham introduced a supervisory control problem for concurrent DES with language-based specifications [2]. In their formulation, a concurrent DES is modeled by the *concurrent* synchronous product of two automata with disjoint event sets. Subsequently, Takai and Ushio considered a more general class of concurrent DES than the concurrent synchronous product [6]. In their model simultaneous events are represented as an unordered set, suitable to describe the evolution of the system with subsystems operating independently. On the other hand, Hubbard and Caines proposed a different framework called multi-agent product of automata [1]. Based on this framework, a supervisory control problem was solved by Romanovski and Caines [5]. There, simultaneous events are represented by a vector whose components are events occurring in the corresponding subsystems. In the two-agent case, both subsystems must make a transition at every step. If one of the automata has no available transition from its current state, the product will also have no available transition from its (vector) state. Similarly labeled events are necessarily synchronized when these are defined for both component models. This product is suitable for modeling a system such that all its subsystems evolve in lockstep. In [1] Hubbard and Caines also mentioned a product of automata called vector synchronous product (VSP), considered as an extension of the scalar synchronous product. The main purpose of the present paper is to develop the VSP. We will first provide a definition of VSP for multiple agents. Here a simultaneous event is represented as a vector with possibly empty events in some of its components. This means that the VSP is suitable to describe both cases: where some of its subsystems operate independently, or where all its subsystems evolve simultaneously. On the basis of VSP we propose and solve a supervisory control problem. The result depends on a new definition of vs-controllability. We also define isomorphism and homomorphism between VSPs. Furthermore, we propose a new VSP with communication. The latter (VSPC) is motivated by the existence in practical systems of subsystems which evolve synchronously, driven not by the occurrence of shared events, but by events linked through a communication relation, for instance the relation between the changes of a traffic light and the behavior of a group of vehicles. It is shown that VSPC is a generalization of VSP. Throughout the paper we use a simple example of traffic control to illustrate the ideas.

2. Vector Synchronous Product of Automata

In this section we define Vector Synchronous Product (VSP) of automata. A simple example is presented in illustration.

Consider a group of discrete-event systems described by automata

$$G_i = (Q_i, \Sigma_i \cup \{\upsilon_i\}, \Gamma_i, \delta_i, q_{oi}, Q_{mi}) \tag{1}$$

where i = 1,2,…,N. For $i \in \{1, \cdots, N\}$, Q_i, Σ_i, Q_{mi} are the sets of states, events, and marked states, respectively; q_{oi} is the initial state; the set-valued map $\Gamma_i : Q_i \to 2^{\Sigma_i \cup \{\upsilon_i\}}$ specifies the set of active events at each state q_i, according to

$\sigma \in \Gamma_i(q_i) \Leftrightarrow \delta_i(q_i, \sigma)!$;

the map $\delta_i : Q_i \times \Gamma_i \to Q_i$ is the transition function; and the special empty event $\upsilon_i \notin \bigcup_{k=1}^{N} \Sigma_k$ represents the null transition

$\delta_i(q_i, \upsilon_i) = q_i\ \ \forall q_i \in Q_i)$.

Definition 1: Given a group of automata (1), their Vector Synchronous Product (VSP) is defined as the new automaton $G := G_1 \|_{vs} G_2 \|_{vs} \cdots \|_{vs} G_N$, as follows.

$$G = (Q, \Sigma, \Gamma, \delta_{vs}, \vec{q}_o, Q_m) \tag{2}$$

where

$Q = Q_1 \times Q_2 \times \cdots \times Q_N$,

$\Sigma = (\Sigma_1 \bigcup \{\upsilon_1\}) \times (\Sigma_2 \bigcup \{\upsilon_2\}) \times \cdots \times (\Sigma_N \bigcup \{\upsilon_N\})$,

$\vec{q}_o = (q_{o1} \quad q_{o2} \quad \cdots \quad q_{oN})$,

$Q_m = Q_{m1} \times Q_{m2} \times \cdots \times Q_{mN}$,

$\Gamma : Q \to 2^{\Sigma}, \vec{\sigma} \in \Gamma(\vec{q}) \Leftrightarrow \delta_{vs}(\vec{q}, \vec{\sigma})!$,

For $\vec{q} \in Q$, $\vec{\sigma} \in \Sigma$ the transition function is specified according to

$$\delta_{vs}(\vec{q}, \vec{\sigma}) = \begin{cases} \left(\delta_1(q_1, \sigma_1) \quad \cdots \quad \delta_N(q_N, \sigma_N)\right), \\ \qquad \text{if } [\forall\ 1 \le i \le N,\ \sigma_i \in \Gamma_i(q_i)] \\ \qquad \wedge [\forall\ 1 \le i,\ j \le N, j \ne i, (\sigma_j = \sigma_i) \\ \qquad \vee (\ \sigma_i \notin \Gamma_j(q_j) \wedge \sigma_j \notin \Gamma_i(q_i))] \\ \\ \text{undefined}, \qquad \text{otherwise} \end{cases} \tag{3}$$

In the case of two agents, the "if" condition in (3) can be written in the simple form

$$[\sigma_1 \in \Gamma_1(q_1) \wedge \sigma_2 \in \Gamma_2(q_2)]$$
$$\wedge [(\sigma_1 = \sigma_2) \ \vee (\sigma_1 \notin \Gamma_2(q_2) \wedge \sigma_2 \notin \Gamma_1(q_1)]$$

The VSP can describe the simultaneous occurrence of events in Σ_1 and Σ_2, i.e., the case when $\sigma_1 \in \Sigma_1,\ \delta_1(q_1, \sigma_1)!$ and $\sigma_2 \in \Sigma_2,\ \delta_2(q_2, \sigma_2)!$. In particular, when an event belongs to the intersection of the two active event sets, it should be active in both agents, i.e., two components of the event vector must be identical. That means the two agents evolve synchronously. On the other hand, it can also describe the asynchronous occurrence of events in Σ_1 and Σ_2, i.e., the case when one of the events is empty, e.g., $\delta_{vs}((q_1, q_2), (\sigma_1, \upsilon_2))$ is defined whenever $\delta_1(q_1, \sigma_1)$ is defined.

Example 1: Consider a simplified traffic intersection. Let the behaviors of vehicle platoons in the left, middle and right lanes of a direction, say from west to east, be described by automata

$G^i := (Q^i, \Sigma^i, \Gamma^i, \delta^i, q_o^i, Q_m^i)$, where

$i = \mathrm{l}\,(\text{left lane}), \mathrm{m}\,(\text{middle lane}), \mathrm{r}\,(\text{right lane})$,

$Q^i := \{stopping^i, going^i\}$,

$\Sigma^i := \{stop^i, go^i\}$,

$Q_m^i = \{q_o^i\} = \{stopping^i\}$,

$\Gamma^i(q) = \Sigma^i \cup \{\upsilon^i\}\ (\forall q \in Q^i)$.

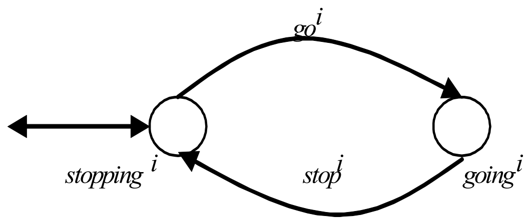

Figure 1. Diagram Describing the Transitions in G^i

The transitions are defined as in Figure1 (Note: self-loops are usually omitted in the figures).
If one requires that the vehicles in the left and middle lanes be controlled by different traffic lights, one can treat the events in Σ^l and Σ^m differently. Thus, $G^l \parallel_{vs} G^m$ is obtained as in Figure 2 (a). If the vehicles in the left and middle lanes are controlled by the same traffic light, one can treat the events in Σ^l and Σ^m as the same. Thus, $G^l \parallel_{vs} G^m$ is obtained as in Figure 2 (b).

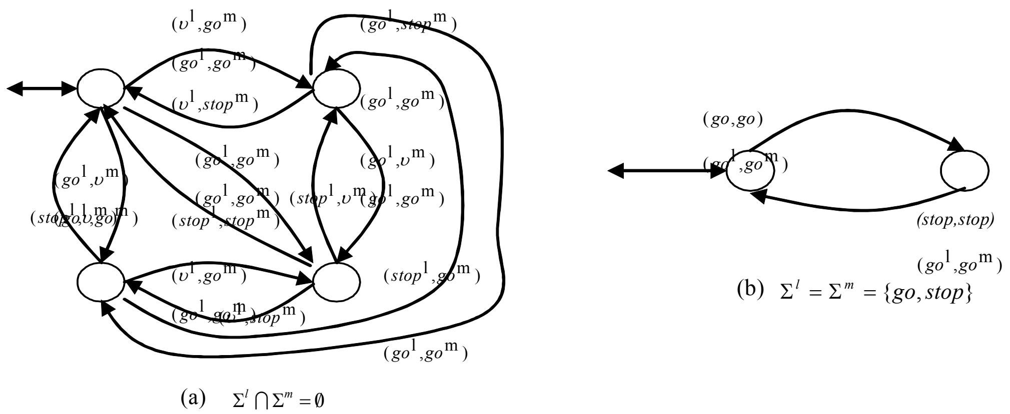

Figure 2. Diagram Describing the Transitions in $G^{\mathrm{L}} \parallel_{vs} G^{\mathrm{m}}$

Remark 1: One can see from Definition 1 that $\neg\,\delta_{vs}\left(\vec{q},\vec{\sigma}\right)!$ if and only if either $\neg\delta_i\left(q_i,\sigma_i\right)!$ for some $i\in\{1,\cdots,N\}$, or else for all $i\in\{1,\cdots,N\}$, $\delta_i(q_i,\sigma_i)!$ but there exist $\sigma_i\neq\sigma_j$ such that either $\sigma_i\in\Gamma_j(x_j)$ or $\sigma_j\in\Gamma_i(x_i)$.

Remark 2: The associative law for the VSP can be easily verified.
To compute the VSP an inductive definition is sometimes more convenient.
Definition 2 (inductive version): Given discrete-event systems described by automata as in (1), their vector synchronous product (VSP) is inductively defined as follows.

1) For two agents $G_1\,\|_{vs}\,G_2$ is given by Definition 1.

2) Suppose

$$\begin{aligned}G^M &:= G_1\,\|_{vs}\,G_2\,\|_{vs}\cdots\|_{vs}\,G_M\\ &:=(Q^M,\Sigma^M,\Gamma^M,\delta_{vs}^M,\vec{q}_o^{\,M},Q_m^M)\end{aligned}$$

has been defined. Then

$$\begin{aligned}&G^{M+1}:=G^M\,\|_{vs}\,G_{M+1}\\ &:=(Q^{M+1},\Sigma^{M+1},\Gamma^{M+1},\delta_{vs}^{M+1},\vec{q}_o^{\,M+1},Q_m^{M+1})\end{aligned}$$

is defined as follows:

$$Q^{M+1}:=Q^M\times Q_{M+1},$$

$$\Sigma^{M+1}:=\Sigma^M\times(\Sigma_{M+1}\cup\{\upsilon_{M+1}\}),$$

$$\vec{q}_o^{\,M+1}=(\vec{q}_o^{\,M},q_{o(M+1)}),$$

$$Q_m^{M+1}:=Q_m^M\times Q_{m(M+1)},$$

For any state

$$\vec{q}=(\vec{q}^{\,M},q_{M+1})\in Q^M\times Q_{M+1}$$

and event

$$\vec{\sigma}=(\vec{\sigma}^M,\sigma_{M+1})\in\Sigma^M\times(\Sigma_{M+1}\cup\{\upsilon_{M+1}\}),$$

the transition function is determined as

$$\delta_{vs}^{M+1}\left(\vec{q},\vec{\sigma}\right)=\begin{cases}\left(\delta_{vs}^M(\vec{q}^{\,M},\vec{\sigma}^M)\quad\delta_{M+1}(q_{M+1},\sigma_{M+1})\right),\\ \qquad\text{if }[\delta_{vs}^M(\vec{q}^{\,M},\vec{\sigma}^M)!\\ \qquad\wedge\delta_{M+1}(q_{M+1},\sigma_{M+1})!]\\ \qquad\wedge[\forall 1\le i\le M,\ \ (\sigma_{M+1}=\sigma_i^M)\\ \qquad\vee(\vec{\sigma}^M|_{i\to M+1}\notin\Gamma^M(\vec{q}^{\,M})\\ \qquad\wedge\ \sigma_i^M\notin\Gamma_{M+1}(q_{M+1}))]\\ \text{undefined,}\quad\text{otherwise}\end{cases}$$

and the active event set as

$$\Gamma^{M+1}: Q^{M+1} \to 2^{\Sigma^{M+1}}, \vec{\sigma} \in \Gamma^{M+1}(\vec{q}) \Leftrightarrow \delta_{vs}^{M+1}(\vec{q}, \vec{\sigma})!.$$

Here σ_i^M denotes the ith component of $\vec{\sigma}^M$, and $\vec{\sigma}^M|_{i \to M+1}$ is derived from $\vec{\sigma}^M$ by replacing its ith component σ_i^M with σ_{M+1}.

The VSP of two VSPs is defined similarly.

3. Isomorph and Homomorph of VSPs

In the section we give definitions of isomorph and homomorph of two VSPs.

Definition 3: Let

$$G^i := (Q^i, \Sigma^i, \Gamma^i, \delta_{vs}^i, \vec{q}_o^i, Q_m^i), i = 1,2 \tag{4}$$

be two VSPs. G^1 and G^2 are said to be isomorphic, denoted by $G^1 \cong G^2$, if there exist bijective maps $F: \Sigma^1 \to \Sigma^2$ and $H: Q^1 \to Q^2$ with the following properties:

1) $F(\vec{\sigma}) = \vec{\sigma}$ whenever $\vec{\sigma} \in \Sigma^1 \bigcap \Sigma^2$;

2) $H(\vec{q}_o^1) = \vec{q}_o^2$;

3) for any $\vec{\sigma}^1 \in \Sigma^1$ and $q^1 \in Q^1$,

$$\delta_{vs}^2(H(\vec{q}^1), F(\vec{\sigma}^1))! \Leftrightarrow \delta_{vs}^1(\vec{q}^1, \vec{\sigma}^1)!$$

and moreover

$$H(\delta_{vs}^1(\vec{q}^1, \vec{\sigma}^1)) = \delta_{vs}^2(H(\vec{q}^1), F(\vec{\sigma}^1))$$

when $\delta_{vs}^1(\vec{q}^1, \vec{\sigma}^1)!$.

4) $\vec{q}^1 \in Q_m^1$ if and only if $H(q^1) \in Q_m^2$.

Remark 3: If the two VSPs in (4) are both reachable, the definition 3 can be simplified as follows: G^1 and G^2 are isomorphic if there exists a bijective mapping $F: \Sigma^1 \to \Sigma^2$ with the following properties 1) $F(\vec{\sigma}) = \vec{\sigma}$ whenever $\sigma \in \Sigma^1 \bigcap \Sigma^2$; 2) for any $\vec{s}^1 \in (\Sigma^1)^*$,

$$\delta_{vs}^2(q_o^2, F(s^1))! \Leftrightarrow \delta_{vs}^1(q_o^1, s^1)!$$

and

$$\delta_{vs}^2(\vec{q}_o^2, F(\vec{s}^1)) \in Q_m^2 \Leftrightarrow \delta_{vs}^1(\vec{q}_o^1, \vec{s}^1) \in Q_m^1.$$

In fact, in this case one can naturally extend the mapping $F: \Sigma^1 \to \Sigma^2$ to $F: (\Sigma^1)^* \to (\Sigma^2)^*$ and define a mapping

$H{:}Q^1 \to Q^2:\ \delta^1_{vs}(\vec{q}^{\,1}_o,\vec{s}^{\,1}) \mapsto \delta^2_{vs}(\vec{q}^{\,2}_o, F(\vec{s}^{\,1}))$.

Remark 4: Let

$G^1 := (Q^1, \Sigma^1, \delta^1_{vs}, q^1_o, Q^1_m)$

be a VSP. Suppose G^2 is derived from G^1 by permuting the order of components in states and events in the same way. Then it is easy to check that $G^2 \cong G^1$. In other words, given two automata G_1 and G_2, one has

$G_1 \parallel_{vs} G_2 \cong G_2 \parallel_{vs} G_1$.

Example 2: Consider a simplified traffic intersection. Let G^i be the automaton describing the traffic behavior of the vehicles in a lane (see Example 1), and G the automaton describing the traffic light (see Figure.3).

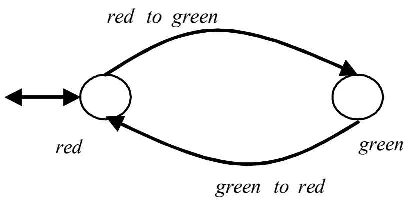

Figure 3. Diagram Describing the Transitions in G

It is easy to verify that $G^i \cong G$ under the following mappings:

$F: stop^i \mapsto green\ to\ red,\ go^i \mapsto red\ to\ green$

$H: stopping^i \mapsto red,\ going^i \mapsto green$

Definition 4: Given two VSPs (4), G^2 is said to be a homomorphic image (homomorph) of G^1, denoted by $G^1 \succeq G^2$ or $G^2 \preceq G^1$ if there exist two surjective maps $F: \Sigma^1 \to \Sigma^2$ and $H: Q^1 \to Q^2$ with the following properties:

1) $F(\vec{\sigma}) = \vec{\sigma}$ whenever $\vec{\sigma} \in \Sigma^1 \cap \Sigma^2$;

2) $H(\vec{q}^{\,1}_o) = \vec{q}^{\,2}_o$;

3) for any $\vec{\sigma}^2 \in \Sigma^2$, $\vec{\sigma}^1 \in F^{-1}(\vec{\sigma}^2), \vec{q}^{\,2} \in Q^2$, $\vec{q}^{\,1} \in H^{-1}(\vec{q}^{\,2})$:

$\delta^2_{vs}(\vec{q}^{\,2}, \vec{\sigma}^2)! \Leftrightarrow \delta^1_{vs}(\vec{q}^{\,1}, \vec{\sigma}^1)!$,

and

$H(\delta^1_{vs}(\vec{q}^{\,1}, \vec{\sigma}^1)) = \delta^2_{vs}(\vec{q}^{\,2}, \vec{\sigma}^2)$

when $\delta^2_{vs}(\vec{q}^{\,2}, \vec{\sigma}^2)!$.

4) $\forall \vec{q}^{\,1} \in H^{-1}(\vec{q}^{\,2}): \vec{q}^{\,1} \in Q^1_m$ if and only if $\vec{q}^{\,2} \in Q^2_m$.

Remark 5: If both VSPs are reachable, Definition 4 can be simplified as follows: G^2 is a homomorphic image of G^1 if there exists an injective map $\hat{F}:\Sigma^2 \to 2^{\Sigma^1}$ with the properties

1) $\hat{F}(\vec{\sigma}) = \{\vec{\sigma}\}$ whenever $\vec{\sigma} \in \Sigma^1 \cap \Sigma^2$;

2) $\hat{F}(\vec{\sigma}_1^2) \cap \hat{F}(\vec{\sigma}_2^2) = \emptyset$ whenever $\vec{\sigma}_1^2 \neq \vec{\sigma}_2^2$, and $\bigcup_{\vec{\sigma}^2 \in \Sigma^2} \hat{F}(\vec{\sigma}^2) = \Sigma^1$;

3) for any $\vec{s}^{\,2} \in \Sigma^{2*}$ and $\vec{s}^{\,1} \in \hat{F}(\vec{s}^{\,2})$,

there holds

$\delta_{vs}^2(\vec{q}_o^{\,2}, \vec{s}^{\,2})! \Leftrightarrow \delta_{vs}^1(\vec{q}_o^{\,1}, \vec{s}^{\,1})!$

and

$\delta_{vs}^2(\vec{q}_o^{\,2}, \vec{s}^{\,2}) \in Q_m^2 \Leftrightarrow \delta_{vs}^1(\vec{q}_o^{\,1}, \vec{s}^{\,1}) \in Q_m^1$.

In this case one can naturally extend the map $\hat{F}:\Sigma^2 \to 2^{\Sigma^1}$ to $\hat{F}:(\Sigma^2)^* \to 2^{(\Sigma^2)^*}$
and define the two maps as follows: for $\forall \vec{\sigma}^2 \in \Sigma^2, \vec{\sigma}^1 \in \hat{F}(\vec{\sigma}^2)$

$F:\Sigma^1 \to \Sigma^2 : \vec{\sigma}^1 \mapsto \vec{\sigma}^2$,

and for $\forall \vec{s}^{\,2} \in (\Sigma^2)^*, \vec{s}^{\,1} \in \hat{F}(\vec{s}^{\,2})$

$H: Q^1 \to Q^2 : \delta_{vs}^1(\vec{q}_o^{\,1}, \vec{s}^{\,1}) \mapsto \delta_{vs}^2(\vec{q}_o^{\,2}, \vec{s}^{\,2})$.

Example 3: Consider the simplified traffic intersection. Let $G^{\mathrm{m}} \parallel_{vs} G^{\mathrm{r}}$ be the VSP describing the traffic behavior of the vehicles in the middle and right lanes. If it happens that $\Sigma^{\mathrm{m}} \cap \Sigma^{\mathrm{r}} = \emptyset$, then $G^{\mathrm{m}} \parallel_{vs} G^{\mathrm{r}}$ has transitions similar to those $G^{\mathrm{l}} \parallel_{vs} G^{\mathrm{m}}$ as shown in Figure 2. (a). Let G be the automaton describing the traffic light (see Figure 3). When one considers the empty event of G and allows the vehicles in the right lane not to be controlled by the traffic light, one can define a map $\hat{F}:\Sigma \to 2^{(\Sigma^{\mathrm{m}} \cup \{\varepsilon^m\}) \times (\Sigma^{\mathrm{r}} \cup \{\varepsilon^r\})}$ as follows:

$red\ to\ green \mapsto \{(go,*) : * \in \Sigma^r \cup \{\upsilon^r\}\}$;
$green\ to\ red \mapsto \{(stop,*) : * \in \Sigma^r \cup \{\upsilon^r\}\}$;
$\upsilon \mapsto \{(\upsilon^m,*) : * \in \Sigma^r \cup \{\upsilon^r\}\}$

Then, it is readily seen that $G^{\mathrm{m}} \parallel_{vs} G^{\mathrm{r}} \succeq G$.

Examples 2 and 3 should clarify the concepts of isomorph and homomorph of VSPs.

4. Supervisory Control of VSP

Given a group of discrete-event systems described by automata (1), let
$\Sigma_i = \Sigma_{ic} \cup \Sigma_{iu}, \quad i = 1,2,\cdots,N$,
where Σ_{ic} is the controllable event set and Σ_{iu} the uncontrollable event set. Assume that

$\Sigma_{ic} \bigcap \Sigma_{ju} = \emptyset, i, j = 1, 2, \cdots, N, i \neq j$

Namely any shared event is either controllable or uncontrollable in both components.

Definition 5: The controllable event subset of the VSP (2) is defined as $\Sigma_c := \{\vec{\sigma} \in \Sigma \mid \exists i \in \{1,2,\cdots,N\} : \sigma_i \in \Sigma_{ic}\}$ and its uncontrollable event subset is defined as $\Sigma_u := \{\vec{\sigma} \in \Sigma \mid \forall i \in \{1,2,\cdots,N\} : \sigma_i \in \Sigma_{iu} \text{ or } \sigma_i = \upsilon_i\}$,

where σ_i denotes the ith component of $\vec{\sigma}$.

Definition 6: Given a VSP (2), let Σ_c, Σ_u be its controllable and uncontrollable event subsets, respectively. A centralized supervisory controller for the VSP G is a map $f : L(G) \to 2^{\Sigma_c}$, i.e., for each string $s \in L(G)$, the event $\vec{\sigma} \in f(\vec{s})$ is disabled by disabling some controllable components of $\vec{\sigma}$. The closed behavior, denoted $L(f/G)$, and the marked behavior, denoted $L_m(f/G)$, of the controlled system under the supervisory control are defined inductively as follow:

$\vec{\varepsilon}_G \in L(f/G)$, where $\vec{\varepsilon}_G$ is the empty string of G; and

$\forall \vec{s} \in L(f/G), \forall \vec{\sigma} \notin f(\vec{s}) : \vec{s}\vec{\sigma} \in L(G) \Rightarrow \vec{s}\vec{\sigma} \in L(f/G)$

$L_m(f/G) = L_m(G) \bigcap L(f/G)$.

It is clear that an uncontrollable event is never disabled. Moreover, when an event $\vec{\sigma} \in f(\vec{s})$ is disabled by disabling some of its controllable components, other events (if any) with the same controllable components will be disabled too. Thus, we need a new definition of controllable language, for which extra notation is needed as follows.

Given a VSP (2), let $I_c(\vec{\sigma})$ denote the index subset of the controllable components of an event

$$\vec{\sigma} = (\sigma_1, \cdots, \sigma_M, \sigma_{M+1}, \cdots, \sigma_N) \in \Sigma.$$

Given an arbitrary fixed string

$$\vec{s} = (s_1, \cdots, s_M, s_{M+1}, \cdots, s_N) \in L(G),$$

let $j_1, \cdots, j_l \in \{1,2,\cdots,N\}$ and

$$\hat{G} := G_{j_1} \|_{vs} \cdots \|_{vs} G_{j_l}$$

Thus $P_{\hat{G}}$ denotes the component-wise projection, i.e.,

$$P_{\hat{G}}(\vec{s}) = (s_{j_1}, \cdots, s_{j_l}) \in L(\hat{G}),$$

For any sub-vector

$$\vec{\sigma}|_{j_1,\cdots,j_l} := (\sigma_{j_1}, \cdots, \sigma_{j_l})$$

of $\vec{\sigma}$ and any state $\vec{q} \in Q$, define a set

$$P_{\hat{G}}^{-1}(\vec{q}; \vec{\sigma}|_{j_1,\cdots,j_l}) \\ := \{\vec{u} \in \Sigma \mid P_{\hat{G}}(\vec{u}) = \vec{\sigma}|_{j_1,\cdots,j_l} \text{ and } \delta_{vs}(\vec{q}, \vec{u})!\}.$$

Definition 7: Let $G := (Q, \Sigma, \Gamma, \delta_{vs}, q_o, Q_m)$ be a VSP, and Σ_c, Σ_u be its controllable and uncontrollable event subsets, respectively. A prefix-closed sub-language $K \subseteq L(G)$ is said to be vs-controllable with respect to G if

1) $\overline{K}\Sigma_u \bigcap L(G) \subseteq \overline{K}$;

2) For $\forall\vec{s}\in K,\forall\vec{\sigma}\in\Sigma_c,\vec{q}\in Q$, if
$\delta_{vs}(\vec{q}_o,\vec{s})=\vec{q},\vec{s}\,\vec{\sigma}\in L(G),\vec{s}\,\vec{\sigma}\notin K$
then

$\exists j_1,\cdots,j_l\in I_c(\vec{\sigma}):\vec{s}P_G^{-1}(\vec{q};\vec{\sigma}|_{j_1,\cdots,j_l})\bigcap K=\emptyset$.

***Theorem 1:* Let G be a VSP given in (2), and $K\subseteq L(G)$ be a prefix-closed sub-language. There exists a centralized supervisory controller $f:L(G)\to 2^{\Sigma_c}$ of G such that $L(f/G)=K$ if and only if K is vs-controllable with respect to G.**

Proof: (Sufficiency) Define a map $f:L(G)\to 2^{\Sigma_c}$ according to

$$f(\vec{s})=\begin{cases}\{\vec{\sigma}\in\Sigma_c\mid\vec{s}\,\vec{\sigma}\in L(G),\vec{s}\,\vec{\sigma}\notin K\}, & \\ & \text{if } \vec{s}\in K\\ \text{undefined}, & \text{otherwise}\end{cases}.$$

Using the vs-controllability of K, it is easy to verify that $L(f/G)=K$. In fact, for any $\vec{s}\,\vec{\sigma}\in L(f/G)$, i.e., $\vec{s}\in L(f/G)$ and $\sigma\notin f(s)$, if $\vec{s}\in K$ and $\sigma\in\Sigma_u$ then $\vec{s}\,\vec{\sigma}\in K$ according to the first property of the vs-controllability of K; otherwise, if $\vec{s}\in K$ and $\vec{\sigma}\in\Sigma_c\setminus f(\vec{s})$ then $\vec{s}\,\vec{\sigma}\in K$ according to the definition of f. So, one has that $L(f/G)\subseteq K$. On the other hand, since $K\subseteq L(G)$ and $L(f/G)=K$ is obtained from $L(G)$ by disabling only those events of $L(G)$ which generate strings not belonging to $K\subseteq L(G)$, we have $L(f/G)\supseteq K$ according to the second property of the vs-controllability of K. (Necessity) Assume there exists a centralized supervisory controller $f:L(G)\to 2^{\Sigma_c}$ such that $L(f/G)=K$. Then the first requirement of the controllability of K is satisfied because f cannot disable uncontrollable events. The second requirement is met because of the fact that $L(f/G)=K$**.** The proof of Theorem 1 is complete.
Let

$$S:=(X,\hat{\Sigma},\xi,x_o,X_m)\tag{5}$$

be any VSP that implements the supervisory control, i.e., $L(S)=L(f/G)$ and $L_m(S)=L_m(f/G)$. Then it is easy to verify that
$P_G(L(G\|_{vs}S))=L(f/G)$,
$P_G(L_m(G\|_{vs}S))=L_m(f/G)$.
It is also clear from the definition of VSP that the behavior of the plant G is affected only by those events of the controller S which are in their shared event set. Evidently if a VSP (5) is chosen to be the supervisory controller, it is required that the event sets Σ and $\hat{\Sigma}$ have the same dimension, say N, and their components satisfy the inclusion relation $\hat{\Sigma}_i\subseteq\Sigma_i, i=1,2,\cdots,N$. On the other hand, in practice situations often arise where distinct events in two different agents should synchronize via some communication transition. To capture this notion we give a more general definition of VSP in the next section.

5. Vector Synchronous Product with Communication

The following definition can be extended to several systems. For clarity of exposition we focus on just two agents.

Definition 8: (VSP with communication). Consider two discrete-event systems described by automata

$G_i = (Q_i, \Sigma_i \cup \{\upsilon_i\}, \Gamma_i, \delta_i, q_{oi}, Q_{mi}), i = 1,2$

Here $Q_i, \Sigma_i, q_{oi}, Q_{mi},\ i = 1,2$ are the sets of states, events, initial states and marked states, respectively; the active sets are defined as

$$\Gamma_i : Q_i \to 2^{\Sigma_i \cup \{\upsilon_i\}},$$
$$\sigma \in \Gamma_i(q_i) \Leftrightarrow \delta_i(q_i, \sigma)!, i = 1,2$$

two special *empty* events $\upsilon_i \notin \Sigma_1 \cup \Sigma_2, i = 1,2$ are introduced, to represent the null transitions

$\delta_i(q_i, \upsilon_i) = q_i\ \forall q_i \in Q_i),\ i = 1,2$;

and $\delta_i,\ i = 1,2$ are the transition functions. Finally, to support the communication feature, bring in a subset $H \subseteq \Sigma_1 \times \Sigma_2$

The Vector Synchronous Product with Communication (VSPC) of the two agents is defined as a new automaton $G_1 \|_{vsc} G_2$, as follows.

$$\begin{array}{l} G_1 \|_{vsc} G_2 = (Q_1 \times Q_2, (\Sigma_1 \cup \{\varepsilon_1\}) \times (\Sigma_2 \cup \{\varepsilon_2\}), \\ \quad \Gamma, \delta_{vsc}, (q_{o1}, q_{o2}), Q_{m1} \times Q_{m2}) \end{array},$$

where $(q_{o1}, q_{o2}) \in Q_1 \times Q_2$ is the initial state of $G_1 \|_{vsc} G_2$, the transition function δ_{vsc} is defined as

$$\delta_{vsc}\left((q_1 \quad q_2), (\sigma_1 \quad \sigma_2)\right) = \begin{cases} \left(\delta_1(q_1, \sigma_1) \quad \delta_2(q_2, \sigma_2)\right), \\ \qquad \text{if } [\sigma_1 \in \Gamma_1(q_1) \wedge \sigma_2 \in \Gamma_2(q_2)] \\ \qquad \wedge [(\sigma_1, \sigma_2) \in H \\ \qquad \vee (\sigma_1 \notin P_1(H) \ \wedge\ \sigma_2 \notin P_2(H))] \\ \text{undefined}, \quad \text{otherwise} \end{cases},$$

The active set is given as $\vec{\sigma} \in \Gamma(\vec{q}) \Leftrightarrow \delta_{vsc}(\vec{q}, \vec{\sigma})!$. Here $P_i, i = 1,2$ denotes the component-wise projection, i.e., $P_i(\sigma_1, \sigma_2) = \sigma_i, i = 1,2$.

One can see that the VSP in Definition 1 for two agents is a special case of the VSPC in Definition 8. In fact, in Definition 1 one can specialize the subset H as

$$H = \left\{\left(\sigma \quad \sigma\right) \mid \sigma \in \Sigma_1 \cap \Sigma_2\right\}$$

Example 4: Consider a simplified traffic intersection. Let G^i be the automaton describing the traffic behavior of the vehicles in a lane (see Example 1), and G the automaton describing the traffic light (see example 2). Define the subset H as follows

$$H = \begin{Bmatrix} (green\ \ to\ \ red,\ \ stop^i), \\ (red\ \ to\ \ green,\ \ go^i) \end{Bmatrix}$$

Then $G \|_{vsc} G^i$ is a new automaton with the transition structure shown in Figure 4.

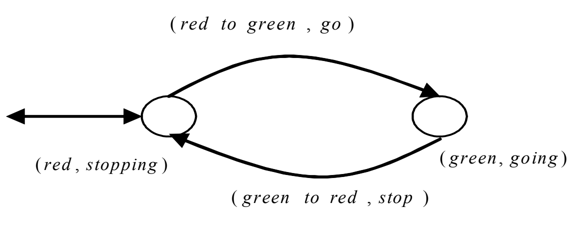

Figure 4. Diagram Describing the Transitions in $G\|_{vsc}G^i$

In this example, G may be thought of as a supervisory controller for the G^i. The behaviors of the G^i must follow those of G owing to the constraints represented by the subset H.
One can set up similar supervisory control problems based on the concept of VSPC.

6. Conclusion

We have investigated the supervisory control problem based on vector synchronous product of automata. Precisely, we first defined the VSP of multiple agents, and then proposed a definition of vs-controllability of a language with respect to a given VSP. The main result shows that vs-controllability of a given language is a necessary and sufficient condition for the existence of a supervisory controller that ensures that the controlled behavior of the system coincides with that language. We have also proposed a more general model based on the notion of vector synchronous product of automata with communication. The corresponding supervisory control problems remain for further investigation.

ACKNOWLEDGEMENT

This paper was supported in part by the National Natural Science Foundation of China (No.60374007) and the Beijing Natural Science Foundation (No.4042006).

REFERENCES

1. HUBBARD, P. and CAINES, P. E., **Initial Investigations of Hierarchical Supervisory Control for Multi-Agent Systems**, Proceedings of the 38th Conference on Decision & Control, Phoenix, Arizona USA, December 1999, pp. 2218-2223.
2. LI, Y. and WONHAM, W. M., **On supervisory control of real-time discrete-event systems**, Inform. Science., 46, no. 3, pp. 159-183, 1988.
3. RAMADGE, P. J. and WONHAM, W. M., **Supervisory control of a class of discrete event processes**, SIAM J. Control Optimization, 25, pp. 206-230, 1987.
4. RAMADGE, P. J. and WONHAM, W. M., **The control of discrete event systems**, Proc. IEEE: Special Issue on Discrete Event Systems, 77, pp. 81-98, 1989.
5. ROMANOVSKI, I. and CAINES, P. E., **On the supervisory control of multi-agent product systems**, Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA, December 2002, pp. 1181-1186.
6. TAKAI, S. and USHIO, T., **Supervisor Synthesis for a Class of Concurrent Discrete Event Systems**, Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, December 2003, pp. 2686-2691.