Bisimulation as Congruence in the Behavioral Setting

A Agung Julius
University of Pennsylvania, agung@seas.upenn.edu

A J. van der Schaft
University of Twente

Follow this and additional works at: https://repository.upenn.edu/ese_papers

Recommended Citation

Suggested Citation:

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/ese_papers/553
For more information, please contact repository@pobox.upenn.edu.
Bisimulation as Congruence in the Behavioral Setting

Abstract
We cast the notion of bisimulation in the Willems' behavioral setting. We show that in this setting, bisimulation is also a congruence, as it is known in the field of concurrent processes. Bisimulation is a congruence means if $A$ and $A'$ are bisimilar systems, then $A \parallel B$ and $A' \parallel B$ are also bisimilar. Here, the operator $\parallel$ denotes systems with composition, and $B$ is any other system that is composed with $A$ or $A'$.

Disciplines
Engineering

Comments
Suggested Citation:

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This conference paper is available at ScholarlyCommons: https://repository.upenn.edu/ese_papers/553
Bisimulation as congruence in the behavioral setting

A. A. Julius  
Dept. Electrical and Systems Engineering  
University of Pennsylvania  
200 S 33rd Street, Philadelphia PA19104  
USA  
Email: agung@seas.upenn.edu

A. J. van der Schaft†  
Dept. Applied Mathematics  
University of Twente  
PO Box 217, Enschede 7500AE  
The Netherlands  
Email: a.j.vanderschaft@math.utwente.nl

Abstract—We cast the notion of bisimulation in the Willems’ behavioral setting. We show that in this setting, bisimulation is also a congruence, as it is known in the field of concurrent processes. Bisimulation is a congruence means if $A$ and $A'$ are bisimilar systems, then $A \parallel B$ and $A' \parallel B$ are also bisimilar. Here, the operator $\parallel$ denotes systems composition, and $B$ is any other system that is composed with $A$ or $A'$.

I. INTRODUCTION

In this paper we study bisimulation of dynamical systems using the behavioral systems theory setting. Bisimulation as a notion of systems equivalence originates in the field of concurrent processes in theoretical computer science [1], [2]. The recent development in systems theory, particularly in the branch of hybrid systems, has seen an increase in the application of ideas from the theoretical computer science to the traditional systems theory. Bisimulation is one of the ideas picked up by researchers and applied to, for example, hybrid systems [3], [4], [5] and some classes of continuous time dynamical systems [6], [7], [8].

Bisimulation is a notion of systems equivalence. One of its idiosyncrasies, as it is known in theoretical computer science, is that bisimulation is a congruence. By this we mean the following. Let $A$ and $A'$ be bisimilar processes. We denote synchronization of processes with the operator $\parallel$. Then for any other process $B$, we have that $A \parallel B$ and $A' \parallel B$ are also bisimilar.

Obviously, congruence is an essential property if a notion of systems equivalence is to be applied in the context of systems composition. Consider a complex system composed of many subsystems. If we replace a subsystem with another system that is equivalent to it, then the congruence property guarantees that the altered complex system is equivalent to the original one. This motivates us to study the congruence property of bisimulation in the behavioral framework [9], [10].

See Figure 1 for an illustration of the discussion above.

One potential benefit that we can exploit from this property is that when we want to analyze the complex system, we can replace the subsystems with other systems that are less complex, and thus make the analysis easier. By less complex, we typically mean systems with smaller state space.

The physical interpretation for lesser complexity can be illustrated as follows. For discrete event systems, this means fewer states, which in terms of computer programs can mean a program that takes less space in the memory. For physical systems, typically the state of a system is associated with the energy of the system. Thus, a state space with smaller dimension means a system with less way to store energy, which usually implies that the system is simpler.

Naturally, we also need to make sure that the system properties that we want to analyze is preserved under the systems equivalence. For bisimulation, it is known that temporal logic formula in LTL and CTL are preserved [11], [12].

II. SYSTEMS AND DYNAMIC MAPS

Now we shall (re)introduce some basic concepts of the behavioral systems theory.

Definition 1: [13] A dynamical system $\Sigma$ is defined as a triple $(\mathcal{T}, \mathcal{W}, \mathcal{B})$, where $\mathcal{T}$ is called the time axis, $\mathcal{W}$ is called the signal space, and $\mathcal{B} \subset \mathcal{W}^\mathcal{T}$ is called the behavior of the system.

A behavior is a collection of trajectories, which are functions mapping the time axis to the signal space. We don’t require the trajectories to be total functions, as they can

†A. J. van der Schaft is also affiliated with the Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 800, Groningen 9700AV, The Netherlands, Email:A.J.van.der.Schaft@math.rug.nl

§A. J. van der Schaft is also affiliated with the Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 800, Groningen 9700AV, The Netherlands, Email:A.J.van.der.Schaft@math.rug.nl
be partial functions as well. The behavior of a dynamical system is the collection of all possible trajectories of the system. A trajectory is possible if it is consistent with the (mathematical) laws describing the system. Thus,

\[ \mathcal{B} := \{ w : T \to \mathbb{W} \mid w \text{ is compatible with the laws of } \Sigma \}. \]  

(1)

The description of possible trajectories above is rather abstract. To make the exposition clearer, we provide the following examples of dynamical systems.

**Example (Newton’s laws):** Newton’s law of motions stipulates that the force \( F \) needed to accelerate a physical body is proportional to the mass \( m \) and the acceleration \( a \).

This is epitomized in the well-known relation

\[ F = ma. \]  

(2)

A dynamical system that describes this theory can be written as a triple \((T, \mathbb{W}, \mathcal{B})\). If we denote the position of the center of gravity of the body as \( x \), then the behavior of the dynamical system that describes the relation between the force and the position of the body is

\[ \mathcal{B} := \{ (F, x) \in \mathbb{R} \to \mathbb{R}^3 \times \mathbb{R}^3 \mid \forall t \in \mathbb{R}, F(t) = m \cdot \frac{d^2 x}{dt^2}(t) \}. \]  

(3)

The trajectories of this system are trajectories of the force and the position variables. Each variable is vector valued, so in this case we take \( \mathbb{W} \) as \( \mathbb{R}^3 \times \mathbb{R}^3 \). The evolution of the variables takes place continuously, so we take \( T \) as, for example, \( \mathbb{R} \).

**Example (finite state automaton):** A finite state automaton can be associated with the collection of strings that it executes [14]. Consider as an example, a printer machine that operates in the following way. It can receive a printing job, this event is annotated by \texttt{job}. It must print out the job that it has received and feed out the output before it can receive another job. Suppose that the printing is associated to an event annotated by \texttt{print}, and the feeding out the printout is associated to \texttt{feed}. The behavior of this system can be defined as the collection of all strings such that \( \texttt{job}.\texttt{print}.\texttt{feed} \) always occurs in this order. Any string that has, for example, \( \texttt{print.\texttt{job.\texttt{feed}}} \) as a substring is not an element of the behavior as it corresponds to the acceptance of a new job before the printout is fed out. The behavior has \( \mathbb{Z}_+ \) as its time axis, and \( A = \{ \texttt{print}, \texttt{job}, \texttt{feed} \} \) as its signal space.

In general, for a dynamical system \( \Sigma = (T, \mathbb{W}, \mathcal{B}) \), we assume that there exists a totally ordered commutative group \( G \) such that either

- (i) \( T = G \) or,
- (ii) There is a \( t_0 \in G \) such that \( T = \{ t \in G \mid t \leq t_0 \} \).

This assumption gives the structure of the general time axis \( T \).

**Definition 2:** [9], [10] Given a dynamical system \( \Sigma = (T, \mathbb{W}, \mathcal{B}) \). Any surjective map with \( \mathcal{B} \times T \) as its domain is called a **dynamic map** of \( \Sigma \). We denote the class of dynamic maps of \( \Sigma \) by \( \mathcal{D}(\Sigma) \).

We equip \( \mathcal{D}(\Sigma) \) with a partial ordering \( \preceq \), which is defined as follows.

**Definition 3:** Given a dynamical system \( \Sigma = (T, \mathbb{W}, \mathcal{B}) \). Let \( \phi \) and \( \gamma \) be dynamic maps in \( \mathcal{D}(\Sigma) \). We say that \( \phi \preceq \gamma \) if for all \( (w_1, t_1), (w_2, t_2) \in \mathcal{B} \times T \) the following implication holds.

\[ \gamma(w_1, t_1) = \gamma(w_2, t_2) \Rightarrow \phi(w_1, t_1) = \phi(w_2, t_2). \]  

(4)

Since we assume that the dynamic maps are surjective but not necessarily bijective, their inverse function do not necessarily exist. Intuitively, this means that some information may be lost when \( \mathcal{B} \times T \) is passed through a dynamic map. That is, it can happen that two different elements of \( \mathcal{B} \times T \) are mapped to the same point in the codomain. The fact that \( \phi \preceq \gamma \) can be interpreted that \( \phi \) retains less information than \( \gamma \). Indeed, we can prove that \( \phi \preceq \gamma \) if and only if there exists a surjective map \( \kappa \) such that \( \gamma = \phi \circ \kappa \). Furthermore, we can also prove that \( \mathcal{D}(\Sigma) \) and the partial ordering \( \preceq \) form a lattice. By this we mean that the greatest lower bound and least upper bound of any pair of elements in \( \mathcal{D}(\Sigma) \) exist.

Given a dynamical system \( \Sigma = (T, \mathbb{W}, \mathcal{B}) \), there are a few properties of dynamic maps in \( \mathcal{D}(\Sigma) \) that are of interest in this paper. These properties are:

**Past inducedness.** A dynamic map \( \phi \in \mathcal{D}(\Sigma) \) is said to be past induced if for any \( w_1, w_2 \in \mathcal{B} \) and \( \tau \in T \),

\[ w_1(t)_{\leq \tau} = w_2(t)_{\leq \tau} \Rightarrow \phi(w_1, \tau) = \phi(w_2, \tau). \]  

(5)

**Markovian.** A dynamic map \( \phi \in \mathcal{D}(\Sigma) \) is said to be Markovian if for any \( w_1, w_2 \in \mathcal{B} \), \( \tau_1, \tau_2 \in T \), and \( \tau'_1 > \tau_1 \),

(i) \( \phi(w_1, \tau_1) = \phi(w_2, \tau_2) \) and

(ii) \( (w_1(t)_{|\tau_1 < t \leq \tau'_1} = w_2(t)_{|\tau_2 < t \leq \tau_2 - \tau_1 + \tau'_1}) \) implies

\[ \phi(w_1, \tau_1') = \phi(w_2, \tau_2 - \tau_1 + \tau'_1). \]  

(6)

In words, a dynamic map is Markovian if whenever two trajectories that are not distinguishable by the dynamic map at a certain time and they proceed with the same segment of trajectory, they should remain indistinguishable.

**State property.** A dynamic map \( \phi \in \mathcal{D}(\Sigma) \) is said to be a **state map** if it satisfies the following state property. For any \( w_1, w_2 \in \mathcal{B} \), \( \tau_1, \tau_2 \in T \),

\[ \phi(w_1, \tau_1) = \phi(w_2, \tau_2) \Rightarrow (w_1 \wedge_{\tau_1} w_2) \in \mathcal{B}. \]  

(7)

The concatenation operation \( \wedge_{\tau_2} \) is defined as

\[ (w_1 \wedge_{\tau_2} w_2)(t) := \begin{cases} w_1(t), & t \leq \tau_1, \\ w_2(t - \tau_1 + \tau_2), & t > \tau_1. \end{cases} \]  

(8)

The past inducedness and state properties are related to the partial ordering \( \preceq \) through the following result.

**Lemma 4:** [9] Let \( \phi \) and \( \gamma \) be elements of \( \mathcal{D}(\Sigma) \) such that \( \phi \preceq \gamma \), then

(i) If \( \gamma \) is past induced, so is \( \phi \),

(ii) if \( \phi \) is a state map, so is \( \gamma \).

The codomain of a state map is called its **state space**, and the elements of the state space are called **states**.

To make the exposition on the concept of state maps clearer, we shall present a few examples. First, we introduce some notations we shall use hereafter.
Notation. (i) The class of locally integrable functions \( f : \mathbb{R} \to \mathbb{R}^v \) that are continuous from the left is denoted as \( L_y^{loc} \).
(ii) The class of all \( g \times q \) matrices whose entries are real polynomials with indeterminate \( \xi \) is denoted by \( \mathbb{R}^{g\times q}[\xi] \).

Example (continuous time LTI): Consider a continuous time linear time invariant (LTI) system
\[
P \left( \frac{d}{dt} \right) y = Q \left( \frac{d}{dt} \right) u, \tag{9}
\]
where \( P(\xi) \in \mathbb{R}^{y\times y}[\xi] \) and \( Q(\xi) \in \mathbb{R}^{y\times u}[\xi] \). We assume that \( P^{-1} \) exists as a rational matrix and that \( P^{-1}Q \) is a proper rational matrix. The variables \( y \) and \( u \) can be regarded as output and input of the linear system. The behavior of this system can be defined as
\[
\mathcal{B} := \{ (y, u) \in \tilde{L}_y^{loc} \times \tilde{L}_u^{loc} \mid (9) \text{ is satisfied weakly} \}. \tag{10}
\]
It is known that this system admits an observable state space representation of the form
\[
\begin{align*}
dx &= Ax + Bu, \\
y &= Cx + Du.
\end{align*} \tag{11}
\]
It can be proven that:
(i) The state variable in this observable representation can be regarded as a dynamic map of the dynamical system, as we can find polynomial matrices \( T_1 \) and \( T_2 \) such that the following relation holds:
\[
x(t) = T_1 \left( \frac{d}{dt} \right) y + T_2 \left( \frac{d}{dt} \right) u. \tag{12}
\]
(ii) The dynamic map \( x \) defined in (12) is indeed a state map. Moreover, it is also past induced and Markovian. \( \square \)

Example (discrete time LTI): Consider the discrete time counterpart of the previous example. We deal with a system of the form
\[
P(\sigma) y = Q(\sigma) u, \tag{13}
\]
where \( P(\xi) \in \mathbb{R}^{y\times y}[\xi] \) and \( Q(\xi) \in \mathbb{R}^{y\times u}[\xi] \). We assume that \( P^{-1} \) exists as a rational matrix and that \( P^{-1}Q \) is a proper rational matrix. The symbol \( \sigma \) signifies unit time shift operation, that is,
\[
\sigma y(k) := y(k + 1), \text{ for all } k \in \mathbb{Z}. \tag{14}
\]
We can prove that this system admits an observable state space representation of the form
\[
\begin{align*}
x(k + 1) &= Ax(k) + Bu(k + 1), \\
y(k + 1) &= Cx(k) + Du(k + 1). \tag{15}
\end{align*}
\]
It can be proven that:
(i) The state variable in this observable representation can be regarded as a dynamic map of the dynamical system, as we can find polynomial matrices \( T_1 \) and \( T_2 \) such that the following relation holds:
\[
x(k) = T_1(\sigma) y + T_2(\sigma) u. \tag{16}
\]
(ii) The dynamic map \( x \) defined in (16) is indeed a state map. Moreover, it is also past induced and Markovian. \( \square \)

Example (deterministic automata): Consider a deterministic automaton\(^1\) \( A \) as depicted in the figure below.

The language generated by this automaton is given by the following regular expression.
\[
L(A) = (aab + aba)^* \tag{17}
\]
The overbar denotes the prefix closure operation. We associate the language generated by the automaton with the behavior of the system. Each string in the language, which is a partial function from \( \mathbb{Z}_+ \) to \( \{a, b\} \), is considered as a trajectory of the system.

We can associate the state reached by a certain string at a certain time with a state map. Denote this state map as \( x \), then we have that, for example:
\[
\begin{align*}
s_1 :&= aba, \quad x(s_1, 0) = 1, x(s_1, 1) = 2, x(s_1, 2) = 4, \cdots \\
s_2 :&= aab, \quad x(s_2, 0) = 1, x(s_2, 1) = 2, x(s_2, 2) = 3, \cdots
\end{align*}
\]
It is trivial to see that \( x \) indeed has the state property. Moreover, \( x \) is also Markovian and past induced. \( \square \)

III. Bisimulation in the Behavioral Setting

Hereafter we assume that the signal space of the dynamical systems we discuss can be factored as follows.
\[
\mathbb{W} = \mathbb{V} \times \mathbb{D}. \tag{18}
\]
The interpretation being that \( \mathbb{V} \) is the external signal space and \( \mathbb{D} \) is the internal signal space. Thus, any trajectory can be written as a pair \( w = (v, d) \), where \( v \) and \( d \) denote the external and internal component of the trajectory.

Notation. We denote the projection of the trajectories with respect to the external and internal signal space as \( \pi_e \) and \( \pi_d \) respectively.

Consider the following definition.

Definition 5: A state system is an ordered pair \( (\Sigma, x) \), where \( \Sigma \) is a dynamical system and \( x \in \mathcal{D}(\Sigma) \) is a state map of \( \Sigma \).

Bisimulation is relation defined between the states of two state systems.

Definition 6: [10] Given two dynamical systems \( \Sigma_1 = (T, \mathbb{V} \times \mathbb{D}_1, \mathcal{B}_1) \) and \( \Sigma_2 = (T, \mathbb{V} \times \mathbb{D}_2, \mathcal{B}_2) \), with state maps \( x_1 \) and \( x_2 \) respectively. Notice that the systems share the same external signal space. We denote the state space of

\(^1\)For excellent introductory material on discrete event systems and automata, we refer the reader to [15], [14].
the state maps as $X_1$ and $X_2$ respectively. A bisimulation relation $R \subseteq X_1 \times X_2$ is a relation with the following property. If we take any $(\xi_1, \xi_2) \in R$, then given any $w_1 := (v_1, d_1) \in \mathcal{B}_1$ and $t_1 \in T$ such that $x_1(w_1, t_1) = \xi_1$, the following holds. If $t_2 \in T$ is such that there exists a $w' := (v', d') \in \mathcal{B}_2$ such that $x_2(w', t_2) = \xi_2$, then there exists a $d_2 \in \pi_d \mathcal{B}_2$ such that if we define
\begin{align}
v_2 := &
\langle v' \land t_2 \rangle v_1,
\quad (19)
\end{align}
\begin{align}w_2 := &
(v_2, d_2),
\quad (20)
\end{align}
we have that
\begin{align}w_2 \in &
\mathcal{B}_2,
\quad (21)
x_2(w_2, t_2) = \xi_2,
\quad (22)
d_2(\tau) = d'(\tau), \forall \tau \leq t_2,
\quad (23)
\end{align}
and for all $\tau > t_2$,
\begin{align}(x_1(w_1, \tau - t_2 + t_1), x_2(w_2, \tau)) \in &
R.
\quad (24)
\end{align}

Conversely, given any $w_2 := (v_2, d_2) \in \mathcal{B}_2$ and $t_2 \in T$ such that $x_2(w_2, t_2) = \xi_2$, the following holds. If $t_1 \in T$ is such that there exists a $w' := (v', d') \in \mathcal{B}_1$ such that $x_1(w', t_1) = \xi_1$, then there exists a $d_1 \in \pi_d \mathcal{B}_1$ such that if we define
\begin{align}v_1 := &
\langle v' \land t_1 \rangle v_2,
\quad (25)
\end{align}
\begin{align}w_1 := &
(v_1, d_1),
\quad (26)
\end{align}
we have that
\begin{align}w_1 \in &
\mathcal{B}_1,
\quad (27)
x_1(w_1, t_1) = \xi_1,
\quad (28)
d_1(\tau) = d'(\tau), \forall \tau \leq t_1,
\quad (29)
\end{align}
and for all $\tau > t_1$, (24) holds. Furthermore, we require that
\begin{align}\forall \xi_1 \in X_1, \exists \xi_2 \in X_2 &
such that (\xi_1, \xi_2) \in R,
\quad (30)
\forall \xi_2 \in X_2, \exists \xi_1 \in X_1 &
such that (\xi_1, \xi_2) \in R.
\quad (31)
\end{align}

Although this definition looks cumbersome, it can be summarized as follows. The bisimulation requires that from any two bisimilar states it is possible to proceed with equal external trajectories while visiting states that are bisimilar. This definition is similar to the original definition by Milner [1], [2], and also similar to the extension defined for continuous-time dynamical systems in, for example, [6], [7], [8].

The additional requirements (30) and (31) are to make sure that all states are involved in the bisimulation.

When two state systems $(\Sigma_1, x_1)$ and $(\Sigma_2, x_2)$ are such that there exists a bisimulation relation $R$ between their state spaces, we say that $(\Sigma_1, x_1)$ and $(\Sigma_2, x_2)$ are bisimilar. In shorthand notation, we write $(\Sigma_1, x_1) \approx_{\text{bis}} (\Sigma_2, x_2)$. The following result establish $\approx_{\text{bis}}$ as an equivalence relation.

**Remark 7:** In some literature, a bisimulation relation $R$ does not necessarily satisfy (30) and (31). However, two systems are said to be bisimilar if there exists a bisimulation relation that satisfies (30) and (31).

---

**Lemma 8:** [10] Let $(\Sigma_i, x_i)$, $i \in \{1, 2, 3\}$, be state systems. Moreover, assume that $x_i$, $i \in \{1, 2, 3\}$, are past induced Markovian state maps. The following relations hold.

(i) $(\Sigma_1, x_1) \approx_{\text{bis}} (\Sigma_1, x_1)$,

(ii) $(\Sigma_1, x_1) \approx_{\text{bis}} (\Sigma_2, x_2)$ implies $(\Sigma_2, x_2) \approx_{\text{bis}} (\Sigma_1, x_1)$,

(iii) $(\Sigma_1, x_1) \approx_{\text{bis}} (\Sigma_2, x_2)$ and $(\Sigma_2, x_2) \approx_{\text{bis}} (\Sigma_3, x_3)$ implies $(\Sigma_1, x_1) \approx_{\text{bis}} (\Sigma_3, x_3)$.

Notice that although in Lemma 8 we assume that the state maps are past induced and Markovian, this is not too restrictive. For example, all the examples that we discuss in the previous section have this property.

**Assumption.** In the remaining of this paper, we shall assume that state maps are past induced and Markovian.

---

**IV. INTERCONNECTION OF STATE SYSTEMS**

In the previous section we see how bisimulation is formulated as a notion of equivalence among state systems. In order to study bisimulation as congruence, we need to formalize the notion of interconnection of state systems.

**Definition 9:** Let $\Sigma_i = (T, \mathbb{V} \times D_i, \mathcal{B}_i)$, $i = 1, 2$, be dynamical systems with their respective state maps $x_1$ and $x_2$. The interconnection of the state systems $(\Sigma_1, x_1)$ and $(\Sigma_2, x_2)$ is defined as follows.

\begin{align}(\Sigma, x) := (\Sigma_1, x_1) \parallel (\Sigma_2, x_2),
\end{align}

where
\begin{align}\Sigma := (T, \mathbb{V} \times D_1 \times D_2, \mathcal{B}),
\mathcal{B} := \{(v, d_1, d_2) \mid (v, d_1) \in \mathcal{B}_1 \text{ and } (v, d_2) \in \mathcal{B}_2\},
\end{align}
\begin{align}x(v, d_1, d_2) := \left[\begin{array}{c}x_1(v, d_1, t) \\
x_2(v, d_2, t)
\end{array}\right].
\end{align}

Thus, interconnection of state systems are defined for systems that have the same external signal space. Notice that the state space of the composed state system is a subset of the product of the state spaces of the individual state systems.

**Example:** Consider two continuous time LTI system
\begin{align}q_1 \left(\frac{d}{dt}\right) y(t) = p_1 \left(\frac{d}{dt}\right) u(t) + r_1 \left(\frac{d}{dt}\right) d_1(t),
\end{align}
\begin{align}q_2 \left(\frac{d}{dt}\right) u(t) = p_2 \left(\frac{d}{dt}\right) y(t) + r_2 \left(\frac{d}{dt}\right) d_2(t),
\end{align}
where $p_i / q_i$ and $r_i / q_i$ are strictly proper fractions. Consider $u$ and $y$ as external variables, and $d_1$ and $d_2$ as internal variables. It is known (see, for example, [10]) that there exist $X_1 \left(\frac{d}{dt}\right)$ and $X_2 \left(\frac{d}{dt}\right)$ such that if we define
\begin{align}x_1(t) := X_1 \left(\frac{d}{dt}\right) \left[\begin{array}{c}u \\
d_1
\end{array}\right],
\end{align}
\begin{align}x_2(t) := X_2 \left(\frac{d}{dt}\right) \left[\begin{array}{c}u \\
d_2
\end{array}\right],
\end{align}
we can have the following state-space representation.
\begin{align}\frac{d}{dt} x_1 = A_1 x_1 + B_1 u + F_1 d_1,
\end{align}
\begin{align}y = C_1 x_1,
\end{align}
V. BISIMULATION AS CONGRUENCE

Now that we have covered the necessary preliminary materials, we can state the main result of this paper.

**Theorem 10:** Let \( \Sigma_i = (T, \mathcal{V} \times D_i, \mathcal{B}_i), i = 1, 2, 3, \) be dynamical systems with their respective state maps \( x_1, x_2 \) and \( x_3 \). If \( (\Sigma_1, x_1) \approx_{bis} (\Sigma_2, x_2) \) then
\[
(\Sigma_1, x_1) \parallel (\Sigma_3, x_3) \approx_{bis} (\Sigma_2, x_2) \parallel (\Sigma_3, x_3). 
\] (45)

**Proof:** First we define
\[
(\Sigma, x) := (\Sigma_1, x_1) \parallel (\Sigma_3, x_3),
\]
(46)
\[
(\Sigma', x') := (\Sigma_2, x_2) \parallel (\Sigma_3, x_3).
\] (47)

Denote the state space of \( x_i \), \( i = 1, 2, 3 \). The state spaces of \( x \) and \( x' \) are given by (see Definition 9)
\[
\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_3,
\] (48)
\[
\mathcal{X}' = \mathcal{X}_2 \times \mathcal{X}_3.
\] (49)

We denote the behaviors of \( \Sigma \) and \( \Sigma' \) with \( \mathcal{B} \) and \( \mathcal{B}' \) respectively.

Since \( (\Sigma_1, x_1) \approx_{bis} (\Sigma_2, x_2) \), there exists a bisimulation relation \( R \subset \mathcal{X}_1 \times \mathcal{X}_2 \) as defined in Definition 6. We need to show that there also exists a bisimulation relation \( R' \subset \mathcal{X} \times \mathcal{X}' \) so that (45) holds. We can construct \( R' \) as follows. For any \( (\xi, \zeta) \in \mathcal{X} \) and \( (\xi', \zeta') \in \mathcal{X}' \),
\[
((\xi, \zeta), (\xi', \zeta')) \in R' \iff (\xi, \zeta) \in R \text{ and } \zeta = \zeta' \). (50)

We now have to prove that \( R' \) is indeed a bisimulation relation. First, we prove that for any \( (\xi, \zeta) \in \mathcal{X} \), there exists a state \((\xi', \zeta') \in \mathcal{X}'\) such that \((\xi, \zeta), (\xi', \zeta') \in R' \). We shall construct such a \((\xi', \zeta') \in \mathcal{X}'\). Since \( \mathcal{R} \) is a bisimulation relation, from Definition 6 we know that there exists a \( \xi' \in \mathcal{X}_2 \) such that \((\xi, \xi') \in R \). Moreover, from (49), we know that \((\xi', \zeta') \in \mathcal{X}' \). Therefore we can take \((\xi', \zeta') = (\xi', \zeta) \) and obtain \((\xi, \zeta), (\xi', \zeta') \in R' \). We also have to prove that for any \((\xi', \zeta') \in \mathcal{X}' \), there exists a state \((\xi', \zeta) \in \mathcal{X} \) such that \((\xi, \zeta), (\xi', \zeta') \in R' \). However, since this proof is analogous to the one we just constructed, we shall not display it.

We also need to show that if we take any \((\xi, \zeta), (\xi', \zeta') \in R' \), then, given any \( w := (v, d_1, d_3) \in \mathcal{B} \) and \( t_1 \in T \) such that \( x_1(v, d_1, t_1) = \xi \) and \( x_3(v, d_3, t) = \zeta \), the following holds. If \( t_2 \in T \) is such that there exists a \( w' := (v', d'_2, d'_3) \in \mathcal{B}' \) such that \( x_2(v', d'_2, t_2) = \xi' \) and \( x_3(v', d'_3, t_2) = \zeta' \), then there exists a \((d''_2, d''_3) \in \pi_d \mathcal{B}' \) such that if we define
\[
\begin{align*}
 v'' &:= v' \land t_2, \\
v'' &:= (v'', d''_2, d''_3),
\end{align*}
\] (51), (52)
we have that
\[
\begin{align*}
 w'' &\in \mathcal{B}' , \\
x_2(v'', d''_2, t_2) &= \xi' , \\
x_3(v'', d''_3, t_2) &= \zeta' , \\
d''_1(\tau) &= d'_2(\tau), \forall \tau \leq t_2 , \\
d''_3(\tau) &= d'_3(\tau), \forall \tau \leq t_2 ,
\end{align*}
\] (53), (54), (55), (56), (57)
and for all \( \tau > t_2 \),
\[
(x(w, \tau - t_2 + t_1), x'(w'', \tau)) \in R'.
\] (58)
By the construction of $R'$, we know that $(\xi, \xi') \in R$ and 
$\zeta = \zeta'$. Since $(\xi, \xi') \in R$, we know that there exists a 
$d''_2 \in \pi_d B_2$ such that $(v'', d''_2) \in B_2$, (54) and (56) hold, and 
for all $\tau > t_2$,

$$(x_1(v, d_1, \tau - t_2 + t_1), x_2(v'', d''_2, \tau)) \in R. \quad (59)$$

Since $\zeta = \zeta'$, if we construct $d''_3$ and $d''_4 := d''_2 \Lambda_{d_1}^2 d_4$, 
then by the state property we shall have that $(v'', d''_3) \in B_3$, (55) 
and (57) hold, and for all $\tau > t_2$,

$$x_1(v, d_1, \tau - t_2 + t_1) = x_3(v'', d''_3, \tau). \quad (60)$$

Notice that (60) holds because of the Markovian property, 
and that we have established (53). Furthermore, (58) is

implied by the construction of $R'$, (59) and (60).

Formally, we have only proven that $R'$ is a simulation of

$(\Sigma, x)$ by $(\Sigma', x')$. The proof of the converse is completely

analogous to the proof above.

This result concurs with the following results.

Continuous time LTI systems. In [10] it is proven that when 
two continuous time LTI systems of a particular form are 
bisimilar if and only if their external behaviors are equal. 
By external behavior we mean the projection of the behavior to 
the external signal space. Theorem 10 implies that external 
behavior equality is a congruence, which is known, and 
proven, for example in [10].

Discrete event systems. Consider again the automata in 
Figure 2. It is obvious that the automata on the top row 
are bisimilar. Now, consider the interconnection between the 
automaton on the top right with itself. It is not difficult 
to see that the result of this interconnection is bisimilar to 
the automaton at the bottom of the figure. This fact is also 

implied by Theorem 10.

VI. CONCLUDING REMARKS

In this paper we show that bisimulation in the behavioral 
setting has the congruence property. The congruence property 
makes bisimulation a suitable notion of systems equivalence 
to use in the context of systems interconnection.

The behavioral framework is general, it is not restricted 
to any particular class of systems. To show it, we present 
examples, in which the theory can be applied. These examples 
are in the form of linear systems and discrete automata. 
However, the application of the theory is not restricted on 
these systems only. A more general class of systems, namely 
hybrid systems, can also benefit from it. In [10] it is shown 
how hybrid systems can be cast in the behavioral framework.

Therefore, the results that we obtain here can be applied for 
hybrid systems, as well as other classes of systems that can 
be cast in the behavioral framework.

A possible future research direction, following up the 
results presented here, is as follows. In the typical setup of 
control problems in the behavioral setting, systems equivalence 
is interpreted as behavioral equality [16], [17], [18], 
[10]. Thus, the problem can be summarized as follows. Given 
$P$ and $S$, find $C$ such that $P \parallel C \approx S$.

The symbols $P$, $C$, and $S$ denote the plant system, the 
controller, and the specification respectively. The equivalence $\approx$

typically means behavioral equality. Some general solutions 
of this kind of problems are given in the above mentioned 
references. It is interesting to see how we can formulate the 
problem with $\approx$ being interpreted as bisimulation. Generally, 
the solutions for the case where $\approx$ means behavioral equality 
cannot be applied, since bisimilarity is generally more strict 
than behavioral equality.

VII. ACKNOWLEDGEMENT

The research presented in this paper is was conducted at 
the Dept. Applied Mathematics, University of Twente, The 
Netherlands. It was funded by the Netherlands Organization 
for Scientific Research (NWO) grant through project number 
617.023.002 (Compositional Analysis and Specification of 
Hybrid Systems).

REFERENCES

[1] R. Milner, Communication and concurrency. Prentice Hall Interna-
tional, 1989.

GI Conference on Theoretical Computer Science. Berlin: Springer-


theory for switching linear systems,” in Proc. 43rd IEEE Conference 

Proceedings of the Mathematical Theory of Networks and Systems, 


[7] A. I. van der Schaft, “Equivalence of dynamical systems by bisimula-
December 2004.

Workshop on Hybrid Systems : Computation and Control. Phila-

their lattice structure and bisimulation,” in Proc. Mathematical Theory 

and discrete systems: a behavioral perspective,” Ph.D. dissertation, 
University of Twente, The Netherlands, February 2005.


[15] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to automa-

[16] A. A. van der Schaft and A. A. Julius, “Achievable behavior by com-
position,” in Proceedings 41st IEEE Conf. Decision and Control. Las 


University of Groningen, June 2003.