



HAL
open science

Computational morphisms of I/O general linear time invariant systems based on mean field conditions

Alexandre Muzy, Bernard P Zeigler

► **To cite this version:**

Alexandre Muzy, Bernard P Zeigler. Computational morphisms of I/O general linear time invariant systems based on mean field conditions. 2020. hal-02429240v1

HAL Id: hal-02429240

<https://hal.science/hal-02429240v1>

Preprint submitted on 6 Jan 2020 (v1), last revised 17 Mar 2021 (v5)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computational morphisms of I/O general linear time invariant systems based on mean field conditions

Alexandre Muzy*, Bernard P. Zeigler†

January 6, 2020

1 Introduction

This article presents preliminary results concerning the implementation of the abstraction of linear dynamic systems based on computational morphisms and mean field conditions. Computational modeling is done using the iterative system specification formalism (Zeigler, Muzy, and Kofman 2018). Base networks of linear systems are abstracted into lumped networks. The lumping is detailed providing an analytical tool to study the dynamics abstracted from the base network. Mean field conditions ensure the preservation of the average activity in the network. Also, the coupling of networks is studied to be able to construct networks of networks while still getting a good idea of the dynamics of the whole system.

In Section 2, mathematical system and mean field theories are introduced. In Section 3, the mathematical framework of linear time invariant systems with inputs/outputs is defined. In Section 4, the mean field abstraction is clearly defined for linear systems using computational morphisms. Section 5 presents the abstraction of network dynamics based on the connections in the network. Finally, in Section 6 a conclusion and perspectives are provided.

2 Mathematical system and mean field theories

2.1 Mathematical system theory

Mathematical general systems (Arbib 1972; Klir 1985; Mesarovic and Takahara 1989; Mesarovic and Takahara 1975; Wymore 1967; Arnold 1994; Harrison 1969; Ho 1992) consist of state-based systems with inputs and outputs. These

*Université Côte d'Azur, I3S CNRS, France, Email: alexandre.muzy@cns.fr.

†Chief Scientist, RTSync Corp, 530 Bartow Drive Suite A Sierra Vista, AZ 85635, United-States of America.

systems can be linear or non-linear, with few hypotheses about their structure (as time invariance described in the mathematical framework section), making these structures very abstract. Input/Output (I/O) interactions of systems make them very realistic but require adding particular mathematical properties to derive theorems about the expected behavior of these systems. In the theory of modeling and simulation (Zeigler, Muzy, and Kofman 2018), a computational specification of general systems has been proposed. The computational systems considered here consist of linear systems.

2.2 Mean field theory

Many references using the mean field hypotheses in the context of neural networks could be cited here. In Table 1, we focus on the main usual hypotheses and breakthrough results with respect to neural network structures (Nykamp et al. 2017; El Boustani and Destexhe 2009; Ostojic 2014) (allowing neuronal networks with arbitrary random degree distributions) and behavior (Faugeras 2009) (exploring the state correlation between neurons). For mathematical convergence to a fixed point, usual hypotheses consist of all-to-all couplings between networks, an infinite number of neurons, weights inversely proportional to the number of neurons. We will show here that the all-to-all couplings between networks is sufficient but not necessary and that many-to-many coupling is necessary. The same way, we will show that the an infinite number of neurons in a network is sufficient but not necessary and that a finite number of neurons is possible.

3 Mathematical framework

3.1 I/O general state-based systems

Definition 1. A *deterministic general I/O system* is a structure (cf. behavior introduced in Figure 1)

$$SYS = (\delta, \lambda)$$

Where

$\delta : Q \times \Omega \rightarrow Q$ is the *transition function*, with Q the *set of states*, Ω the *set of (piecewise continuous) input segments* $\omega : [t_1, t_2] \rightarrow X^1$, with $[t_1, t_2]$ the *interval of the segment* and X the *set of input values*. The sets of input values X and states Q are arbitrary.

$\lambda : Q \rightarrow Y$ is the *output function*, which can be considered as a (partial) observation of the state of the system.

¹A piecewise continuous input segment is a map from each time point $t \in [t_1, t_2]$ (with t_1 and t_2 not fixed) to a corresponding input value $x \in X$.

publication	non linear state correlation between neurons	all-to-all coupling btwn networks	many-to-many coupling btwn networks	infinite nb of neurons	non infinite nb of neurons	weights inv. prop. to nb of neurons	arbitrary degree distribution in a network
(Faugeras 2009)	x	x		x		x	
(Nykamp et al. 2017)	x			x			x
(El Boustani and Destexhe 2009)				x			x
(Ostojic 2014)	?			x			x
us		x	x	x	x		x

Table 1: Mean field conditions in neural networks.

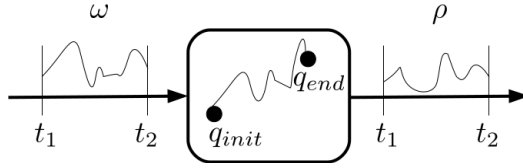


Figure 1: General I/O system dynamics: When receiving an input segment $\omega \in \Omega$, the system achieves a transition from initial state $q_{init} \in Q$ to final state $q_{end} \in Q$ and returns an output segment $\rho \in P$.

For one input segment $\omega \in \Omega$ defined over an interval $[t_1, t_2]$, with t_1 and t_2 not fixed², the system goes continuously from one initial state $q_{init} \in Q$ to one final state $q_{end} \in Q$ by its transition function: $q_{end} = \delta(q_{init}, \omega)$. To do so, intermediate states are computed for particular (allowed) time breakpoints $t \in [t_1, t_2]$ based on the *composition property of the transition function* (cf. Figure 2): $\delta(q, \omega) = \delta(\delta(q, \omega_{[t]}), \omega_{[t]})$, with $\omega_{[t]} = \omega|_{[t_1, t]}$ and $\omega_{[t]} = \omega|_{[t, t_2]}$ being respectively the *left sub-segment* and the *right sub-segment* of ω . Finally, the system generates an *output segment* $\rho \in P$ such that $\rho : [t_1, t_2] \rightarrow Y$ and $\rho_{[t]} = \lambda(\delta(q, \omega_{[t]}))$. The *set of input segments*, Ω , is the union of all input segments $\omega_{[t]}$ and $\omega_{[t]}$ and the *set of output segments*, P , is the union of all output segments $\rho_{[t]}$ and $\rho_{[t]}$.

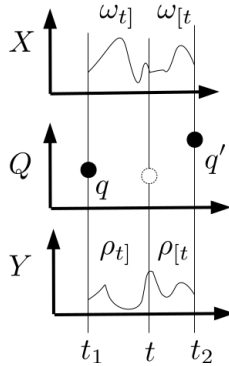


Figure 2: Composition of segments.

The *current state* is the minimal information to deterministically compute the *next state* in a very large state space. The system is markovian. However, notice that a current state can be seen as the result of previous input-state transitions (Zadeh and Desoer 1963). Then, the state of the system can be considered at a higher dependence order, a state being the result of several previous state transitions. Notice also that the system holds inputs and outputs, which is a more general and convenient principle for modeling complex systems,

²Segments can be also defined as starting from time 0 showing then that they can be translated, this is the *time invariance property of systems* (Zeigler, Muzy, and Kofman 2018).

although it makes these systems more unpredictable. In (Ivanov 2013), it is proven also that previous inputs can be stored in states showing the equivalence of both closed and open system structures.

Systems are very abstract and general structures that proved to map all usual modeling formalisms (Zeigler, Muzy, and Kofman 2018). They allow integrating and comparing these formalisms. However, abstract does not mean trivial in the sense that the properties shown for arbitrary inputs, states and outputs can be shown to hold at a lower specification level, i.e., for specific inputs, states and outputs.

Systems are *time invariant*, i.e., any input segment $\omega : [t_1, t_2] \rightarrow X$, applied at time t_1 can be applied at a time t_3 , leading to the same state and output transitions. Defining a translation operator for each time $t \in T$, as $TRANS_\tau : \Omega \rightarrow \Omega$, for an input segment ω , $\omega' = TRANS(\omega)$, with $\omega'(t + \tau) = \omega(t)$ for all $t \in [t_1, t_2]$. Then, a system $SYS = (\delta, \lambda)$ is time invariant for all input segments $\omega \in \Omega$ and all times $\tau \in T$, if:

1. Ω is *closed under translation*: for $\omega \in \Omega \Rightarrow TRANS_\tau(\omega) \in \Omega$.
2. δ is time invariant: for all states $q \in Q$, $\delta(q, \omega) = \delta(q, TRANS_\tau(\omega))$.

3.2 Linear time invariant systems

Definition 2. A Linear time invariant System (LSYS) is a structure

$$LSYS = (\delta, \lambda)$$

Where

X, Q, Y are finite dimensional vector spaces over \mathbb{R} ,

$\delta(q, \omega) = qe^{A(\omega)} + \int_0^{l(\omega)} e^{A(l(\omega)-\tau)} B\omega(\tau) d\tau$ is the transition function³,

$\lambda(q) = CQ$ is the output function,

$A : Q \rightarrow Q$, $B : X \rightarrow Q$ and $C : Q \rightarrow Y$ are linear operators.

Definition 3. A matrix representation $\begin{cases} q'(t) = Aq(t) + Bx(t) \\ y(t) = Cq(t) \end{cases}$ is represented by a Linear Time Invariant System as $\begin{cases} \delta(q, \omega) = qe^{At} + \int_0^t e^{A(t-\tau)} B\omega_\tau d\tau \\ \lambda(q) = CQ \end{cases}$.

Definition 4. A linear time invariant system consists of linear transition and output functions with additive and distributive properties:

1. $\delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>}) = \delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>})$
2. $\delta(aq, a\omega_{t>}) = a\delta(q, \omega_{t>})$

³Notice that all segments are translated to 0, for simplicity.

$$3. \lambda(\delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>})) = \lambda(\delta(q_1, \omega_{1,t>})) + \lambda(\delta(q_2, \omega_{2,t>}))$$

Let us prove these properties.

Proposition 1. $\delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>}) = \delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>})$

Proof. Based on matrix representation,

$$\begin{aligned} \delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>}) &= e^{At}q_1 + \int e^{A(t-\tau)}B\omega_{1,\tau>}d\tau + e^{At}q_2 + \int e^{A(t-\tau)}B\omega_{2,\tau>}d\tau \\ &= e^{At}q_1 + e^{At}q_2 + \int e^{A(t-\tau)}B\omega_{1,\tau>}d\tau + \int e^{A(t-\tau)}B\omega_{2,\tau>}d\tau \\ &= e^{At}(q_1 + q_2) + \int e^{A(t-\tau)}B(\omega_{1,\tau>} + \omega_{2,\tau>})d\tau \\ &= \delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>}) \end{aligned}$$

□

Proposition 2. $\delta(aq, a\omega_{t>}) = a\delta(q, \omega_{t>})$

Proof. Similar to Proposition 1.

□

Proposition 3. $\lambda(\delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>})) = \lambda(\delta(q_1, \omega_{1,t>})) + \lambda(\delta(q_2, \omega_{2,t>}))$

Proof. Starting from additive properties,

$$\begin{aligned} \lambda(\delta(q_1 + q_2, \omega_{1,t>} + \omega_{2,t>})) &= \lambda(\delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>})) && \text{by Proposition 1} \\ &= C(\delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>})) && \text{by Definition 2} \\ &= C(\delta(q_1, \omega_{1,t>}) + \delta(q_2, \omega_{2,t>})) && \text{by linearity of } C \\ &= \lambda(\delta(q_1, \omega_{1,t>})) + \lambda(\delta(q_2, \omega_{2,t>})) \end{aligned}$$

□

Example 1. Pool of two 1-D linear components with identical structure:

$$\begin{cases} q_1' = aq_1 + bx_1 \\ y_1 = cq_1 \end{cases} \quad \text{and} \quad \begin{cases} q_2' = aq_2 + bx_2 \\ y_2 = cq_2 \end{cases}$$

Let the feedback be the (same) average output to each component:

$$x_1 = x_2 = \frac{y_1 + y_2}{2} = \frac{cq_1 + cq_2}{2} = \frac{c(q_1 + q_2)}{2}$$

The state of the network is:

$$\begin{aligned} q_1' + q_2' &= a(q_1 + q_2) + b(x_1 + x_2) = a(q_1 + q_2) + 2bx_1 = a(q_1 + q_2) + bc(q_1 + q_2) \\ \text{with } Q &= q_1 + q_2 \text{ and } Q' = (a + bc)Q \end{aligned}$$

3.3 Computational morphisms

I/O general systems can be considered as abstract machines achieving *temporal computations (or executions of system (output) transitions functions)*. A temporal computation relies on a delay (possibly zero) between inputs and outputs. Computations take time. The number of the computations should be finite to guarantee that the simulation ends.

Simulation and computers consist more and more of a huge number of components interacting together. A fundamental modeling challenge remains the development of a guiding mathematical framework to constructively set and analyze the behavior of networks of components at both local and global levels.

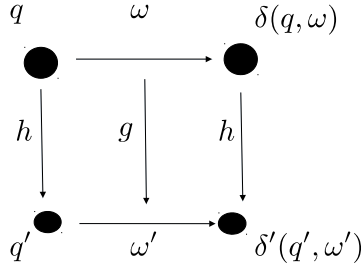


Figure 3: Morphism mapping.

The difficulty of developing such modeling structures is due to the number of local state computations and to the interactions between the components (the temporal coordination of the distributed computations). To abstract local system behaviors into network ones, relying on local (temporal) state computations, *computational morphisms* can be used.

Definition 5. A *system morphism* or generalized homomorphism, between a detailed system SYS (or base model) and another abstract system SYS' (or lumped model), is a pair (g, h) such that (cf. Figure 3):

1. $g : \Omega \rightarrow \Omega'$
2. $h : \bar{Q} \rightarrow^{onto} Q'$, where $\bar{Q} \subseteq Q'$,
3. for all $q \in \bar{Q}$, $\omega' \in \Omega'$, $h(\delta(q, g(\omega'))) = \delta'(h(q), \omega')$ (transition function preservation)

4 Mean field abstraction

Figure 4 shows the morphism between a base network and a lumped network.

Definition 6. We summarize all usual mean field assumptions into only *two sufficient and necessary assumptions* for abstracting linear networks:

1. **Stability:** The transition function of the network resultant system admits a fixed point, the system having input to output feedback,
2. **Homogeneity:** All the components of the network have the same dynamics (or state transition function) and the same structure (same transition/output functions and receive the same inputs).

Both assumptions lead to the network structure of Figure 5.

Mean field morphism between a base network model and a lumped network model is presented in Figure 6.

In the computational context, the number n of components needs not to be infinite.

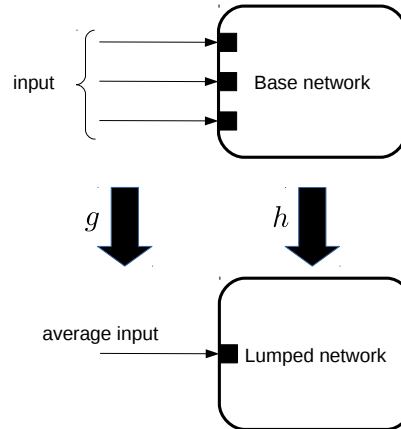


Figure 4: Morphism between base and lumped networks.

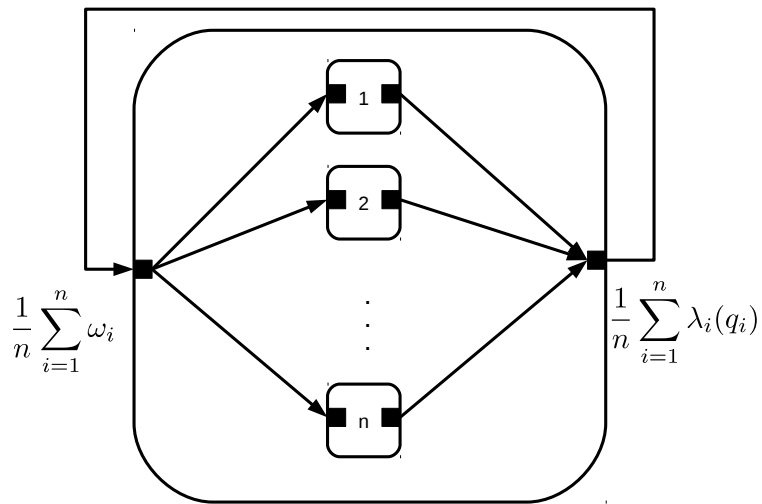


Figure 5: Coupling.

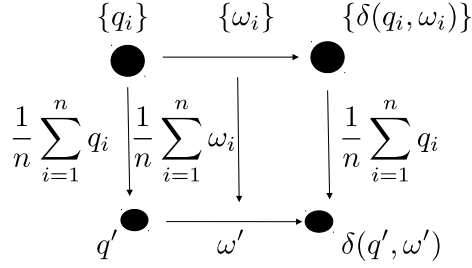


Figure 6: Commutative diagram of mean field network morphism.

Definition 7. A base model network of linear time invariant systems consists of

$$\eta_{base} = (\delta, \lambda)$$

Where

$$\begin{aligned} \delta(q, \omega) &= \times_i \delta_i(q_i, \omega_i) \text{ with } q = (\dots, q_i, \dots) \text{ and } \omega = (\dots, \omega_i, \dots), \\ \lambda(q) &= \times_i \lambda_i(q_i) \end{aligned}$$

Definition 8. A lumped model network of linear time invariant systems consist of

$$\eta_{lumped} = (\delta', \lambda')$$

Where $\omega' = \frac{1}{n} \sum_{i=1}^n \omega_i$, $\delta' \stackrel{def}{=} \delta_i$, with $q' = \frac{1}{n} \sum_{i=1}^n q_i$ and $\lambda' \stackrel{def}{=} \lambda_i$.

Following previous assumptions, the relationship between the base and lumped model networks consists of:

Theorem 1. *If there exists a fixed point in the base model network of linear time invariant systems for a particular input segment $\omega \in \Omega$, there exists a fixed point in the lumped network of linear time invariant systems for the same input segment $\omega \in \Omega$.*

Proof. The fixed point in the base model network consists of $\omega = \lambda(\delta(q, \omega))$, the latter equation can be developed as follows:

$$\begin{aligned} \omega &= \lambda(\delta(q, \omega)) \\ &= \lambda(\dots, \delta_i(q_i, \omega_i), \dots) && \text{based on base network structure (cf. Definition 7)} \\ &= \lambda'(\frac{1}{n} \sum_{i=1}^n \delta_i(q_i, \omega_i)) && \text{based on lumped network structure (cf. Definition 8)} \\ &= \lambda'(\delta_i(\frac{1}{n} \sum_{i=1}^n q_i, \frac{1}{n} \sum_{i=1}^n \omega_i)) && \text{based on linear system properties (cf. Definition 4)} \\ &= \lambda'(\delta'(q', \omega')) \end{aligned}$$

□

Remark 1. As shown by this theorem, the usual mean field assumption requiring the set of components to be infinite is sufficient but not necessary.

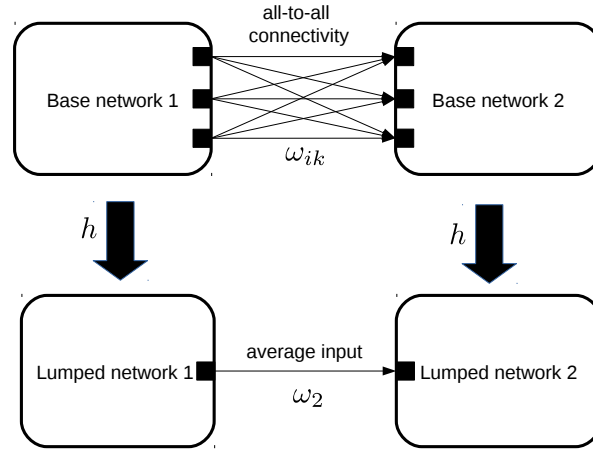


Figure 7: Morphism between two base networks and two lumped networks.

5 Connection of networks

Figure 7 shows the morphism between two base networks and two lumped networks.

The set of coupled base networks is closed under morphism, meaning that coupling base networks, a morphism holds when coupling corresponding lumped networks.

Theorem 2. *The set of base coupled networks is closed under morphism.*

Proof. Figure 8 shows the commutative diagram of the morphism between two base networks and two lumped networks. We prove here that the state transition of the target lumped network 2 depends on the output transition of the source network 1:

$$\begin{aligned}
 \{\delta(q_i^1)\}, \{\delta(q_k^2, \frac{1}{n} \sum_{k=1}^n \omega_k)\} &= \{\delta(q_i^1)\}, \{\delta(q_k^2, \frac{1}{n} \sum_{k=1}^n \sum_{k \in S_i} \omega_{ik})\} \\
 &= \{\delta(q_i^1)\}, \{\delta(q_k^2, \frac{1}{n^2} \sum_{k=1}^n \sum_{k \in S_i} \lambda(q_i^1))\} \\
 &= \{\delta(q_k^1)\}, \{\delta(q_i^2, \frac{1}{n} \sum_{k=1}^n \lambda(q_i^1))\} && \text{if all } |S_i| = n \\
 &= \{\delta(q_1), \lambda(q_1)\}
 \end{aligned}$$

□

Remark 2. All-to-all coupling condition is sufficient but not necessary. Many-to-many coupling is possible. For example one-to-one correspondence does not require all-to-all coupling but still works. Using for example identity mapping instead of average (between the lumped networks in Figure 8); so does any partition on the first set (of influencing components in the source network) with the second set (of influenced components in the target network) being smaller or equal to the first set.

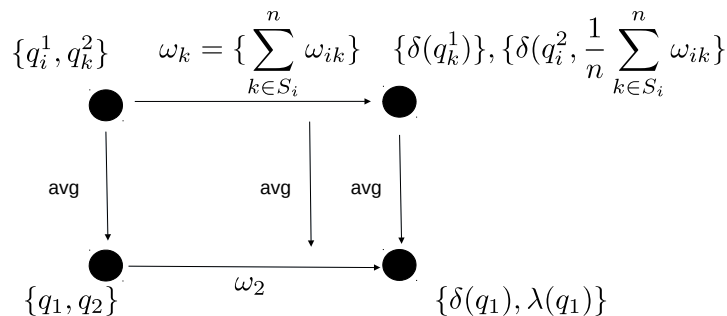


Figure 8: Commutative diagram of the morphism between two base networks and two lumped networks.

6 Conclusion

Sufficient and necessary conditions have been proposed in Definition 6 for I/O general linear invariant systems. These conditions proved to be implemented by computational morphisms leading to fixed points in networks (cf. Theorem 1) and allowing lumping coupled networks (cf. Theorem 5). Compared to usual mean field conditions shown in Table 1, computational assumptions allowed a non infinite number of components in networks as well as many-to-many couplings (cf. Remarks 1 and 2).

References

- Arbib, Michael A (1972). *Theories of abstract automata*.
- Arnold, André (1994). *Finite Transition Systems. International Series in Computer Science*.
- El Boustani, Sami and Alain Destexhe (2009). “A master equation formalism for macroscopic modeling of asynchronous irregular activity states”. In: *Neural computation* 21.1, pp. 46–100.
- Faugeras, Olivier (2009). “A constructive mean-field analysis of multi population neural networks with random synaptic weights and stochastic inputs”. In: *Frontiers in Computational Neuroscience* 3. ISSN: 1662-5188. DOI: 10.3389/neuro.10.001.2009. URL: <http://dx.doi.org/10.3389/neuro.10.001.2009>.
- Harrison, Michael A (1969). *Lectures on linear sequential machines*. Tech. rep.
- Ho, Yu-Chi (1992). *Discrete event dynamic systems: analyzing complexity and performance in the modern world*. IEEE.
- Ivanov, E. (2013). *Investigation of abstract systems with inputs and outputs as partial functions of time*. Phd dissertation. Taras Shevchenko National University of Kyiv, Ukraine.
- Klir, George J (1985). *Architecture of systems complexity*. Saunders, New York.

- Mesarovic, M.D. and Y. Takahara (1975). *General Systems Theory: Mathematical Foundations*. LNCIS 116. Academic Press.
- (1989). *Abstract Systems Theory*. LNCIS 116. Springer.
- Nykamp, Duane Q et al. (2017). “Mean-field equations for neuronal networks with arbitrary degree distributions”. In: *Physical Review E* 95.4, p. 042323.
- Ostojic, Srdjan (2014). “Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons”. In: *Nature neuroscience* 17.4, p. 594.
- Wymore, Wayne (1967). *A mathematical theory of systems engineering*. Wiley.
- Zadeh, L.A. and C.A. Desoer (1963). *Linear system theory: the state space approach*. McGraw-Hill series in system science. McGraw-Hill.
- Zeigler, Bernard P, Alexandre Muzy, and Ernesto Kofman (2018). *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*. Academic press.