IEEE Conference of Decision and Control, Las Vegas, NV, December 10-13, 2002.

A DATA DRIVEN APPROACH TO LEARNING DYNAMICAL SYSTEMS

Paul B. Brugarolas
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr, Pasadena, CA 91109

Michael G. Safonov
Dept. Electrical Engineering – Systems, University of Southern California
3740 McClintock Ave, Los Angeles, CA 90089-2563

Abstract

This paper presents a mathematical framework for learning (properties of a dynamical system) from experimental data via hypothesis falsification, and its application to control and system identification problems.

Keywords: control, system identification, behavioral systems, adaptive control, model validation, set membership, falsification, experimental data.

1. INTRODUCTION

In this paper we look at the problem of learning from data. An unknown or partially unknown dynamical system is the object of our study. Consider some experimental data collected from this system. The question is: How can we learn from this experimental data if the dynamical system has a certain property? For example, how can we learn if the dynamical system is a low pass system, or if a mathematical model derived from first principles does correctly represent the dynamics of this system, or if a controller in closed loop with this system will exhibit certain closed loop performance. The answers to these questions may be obtained by formulating hypotheses and testing them against the experimental data. As a result the data either falsifies the hypothesis or it unfalsifies it. We prefer to use the term unfalsified instead of validated because there always remains the possibility that new data may show that a previously "validated" hypothesis was false. So, the most we could learn is that a property of the dynamical system is not false.

The key idea is the concept of falsification of hypotheses, which has its origins in Popper's "criteria of falsifiability" for determining the scientific status of a theory [16]. Its application to control and system identification problems have appeared in the recent literature [1-4,9,12,17-20,26]. The objective of this paper is to provide a general formulation for learning through hypothesis falsification, and to show how it applies to control and system identification problems.

An important area of application is that of switching approaches to control or identification. In the recent years, there have been important advances in developing a theory about switching systems and applying it to solve system identification and control problems. Some examples are unfalsified control [1-4,17-20], supervisory control [11,12], multiple model adaptive control [13], and model and controller validation [1,5-10,15,26]. In fact, our approach is a generalization of the unfalsified control concept to learning about dynamical systems. The main advantages of this approach to learning are its generality, its capability to make maximum use of the data (all available information in the data is used), and its assumption free formulation.

This paper is organized as follows. Section 2 contains the basic concepts of the framework. It briefly reviews Willems' behavioral approach to system theory that is the foundation on which our framework for learning stands. In section 2, we also give a definition for experimental data, and present our hypothesis unfalsification framework for learning. Section 3 applies this framework to control and system identification problems. Section 4 provides some practical considerations and a brief control design example using this framework. Section 5 presents the conclusions of this study.

2. BASIC CONCEPTS

2.1 Background on the Behavioral Approach to Mathematical Systems Theory

This section provides an informal review of the basic concepts behind Willems' behavioral approach to mathematical systems theory [14,23-25], which will be used to develop the hypothesis learning framework.

Consider a phenomenon. Assume that this phenomenon produces outcomes in a set Z, called the universum. Consider the subset of the universum made of the possible outcomes of the phenomenon, called the behavior $B \subset Z$. Then a mathematical model of the phenomenon is defined to be the pair (Z, B). Mathematical models could be given by multiple representations. Often, behaviors are given as the solution set to behavioral equations. Hence, a behavioral equation representation of the mathematical model (Z, B) is (Z, E, f), where $f : Z \to E$ where E is the equating space ($Null(E) = 0$), and $B = \{z \in Z | f(z) = 0\} = \ker(f)$, that is the kernel of the behavioral equation. Not to be confused with the image representation [14], which we do not use in the present paper.

In this approach, dynamical systems are viewed as mathematical models. These mathematical models have behaviors that evolve with time and such their outcomes are signals. These signals live in the signal space Z, which plays the role of the universum. Hence $\Sigma = (Z, B)$. Behaviors for dynamic systems could be represented in different ways. Often they are represented by differential or difference equations, in which case the behavioral equation is of the form $f(z) = R\left(\frac{d}{dt}\right)z = 0$. This type of representation is called a kernel representation. For example, most traditional input-output and state space representations can be cast in this form [14]. This behavioral equation representation will be the preferred representation for models and controllers studying the present paper.

Interconnections between dynamical [14] systems that live in the same signal space Z are described in this approach as follows. Given two dynamical systems $\Sigma_1 = (Z, B_1)$ and $\Sigma_2 = (Z, B_2)$, then the interconnection of Σ_1 and Σ_2 is the following dynamical system $\Sigma_1 \wedge \Sigma_2 = (Z, B_1 \cap B_2)$. This means that the interconnected system consists of the trajectories consistent with both behaviors.

2.2 Experimental data

This section explains what is understood by experimental data from a dynamical system and provides a formal definition in the behavioral approach. Experimental data from a system usually consists of command signals to the actuators, and measurement signals from the sensors as shown in Figure 1. For this reason the concept of an extended plant is given as follow.

Definition 2.1 *The **extended plant** is defined to contain the plant, sensors and actuators.*

The experimental data is gathered by means of an observation process. This process will be defined using an observation operator, P_τ, which definition follows.

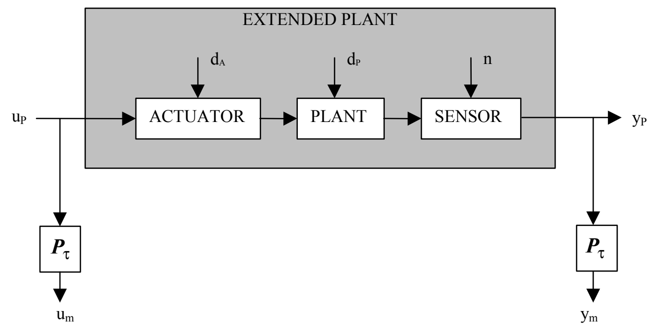

Figure 1: Extended plant

Consider the signal space Z, which contains the extended plant input-output space $U_P \times Y_P \subset Z$.

Definition 2.2 *An **observations operator** P_τ is a mapping $P_\tau : Z \to Z$.*

Examples of observation operators are the time-truncation operator $P_\tau(x) = \begin{cases} x(t) & if \quad t \le \tau \\ 0 & if \quad t > \tau \end{cases}$, and the time-sampling operators $\boldsymbol{P_\tau(x) = \{x_1, x_2, ..., x_\tau\}}$.

Then, the measurements are $(u_m, y_m) = P_\tau(u_P, y_P)$, the observed inputs and outputs of the extended plant.

Definition 2.3 *A **data interpolant set** of an extended plant is a subset $D(\tau) \subset Z$ where $D(\tau) = \{z \in Z | P_\tau(u, y) = (u_m, y_m)\}$.*

For example, if the observations operator were the time-sampling operator, then the data interpolant set $D(\tau)$ will consist of all possible signals $z \in Z$ that interpolate the observed samples.

2.3 Learning via Hypotheses Falsification

This section presents the mathematical framework for learning from experimental data via the hypothesis testing principle. It is developed in the behavioral framework presented in the previous section. Experimental data from a system is used to test a hypothesis about the system. If the hypothesis passes the test then we will say that it has been validated or, more precisely, unfalsified. The term unfalsified stresses the tentative nature of experimental validation: future data may later falsify a hypothesis that had been validated by past data.

Consider a signal space Z. Consider a data interpolant set from the extended plant $D \subset Z$. Consider a hypothesis, a mathematical model that defines a relation between signals in Z , which is formally defined as follows.

Definition 2.4 *A **hypothesis** is a dynamical system* (Z, E, h) *with behavior* $H = \{z \in Z | h(z) = 0\} = \ker(h)$ *where* $h : Z \to E$.

This hypothesis will be tested against the experimental databy the following condition.

Definition 2.5 *A hypothesis* (Z, E, h)*, is said to be **falsified** by a data interpolant set* $D \subset Z$ *if, and only if,*

$$\ker(h|D) = \varnothing \qquad (1)$$

Where $h|D : D \to E$ *denotes restriction of the domain of the behavioral operator h to* $D \subset Z$ *. Otherwise, is said to be unfalsified.*

Remarks:

- Condition (1), $\ker(h|D) = \{z \in D | h(z) = 0\} = \varnothing$, means that does not exist any behavior for which the hypothesis is consistent with the data. In some cases, this condition could be written as set intersection $H \cap D = \varnothing$. In practice, this is the case for hypotheses that describe a model or a controller, but not for hypotheses that describe a performance specification because often require a constraint to be met for any possible reference signal, unless we work in a truncated space. For example, a performance specification for a closed loop regulation control system may be $\|y - r\| < \varepsilon, \forall r$.
- Condition (1) differs from $D \subseteq H$ [23], because our definition of the data interpolant set (viz. Def. 2.3) contains all possible extensions of the measurements. Hence, for a hypothesis to be falsified we require that this hypothesis is not consistent with any of the interpolant of the measurements. However, in practice it is useful to restrict the problem to a truncated subspace (truncated by the observations operator) such there is no need to generate all possible extensions of the measurements and simpler conditions could be used. This work was reported in [3].
- Implicit in Definition 2.3, there is a decision rule $d : D \to F$. A mapping from the data to one of two possible outcomes, unfalsified or falsified, viz. $d(D) = \begin{cases} \textit{unfalsified} & \textit{iff} \quad \ker(h|D) \neq \varnothing \\ \textit{falsified} & \textit{otherwise} \end{cases}$.
- This method could be applied to learning multiple behaviors of a plant simultaneously by combining them in one hypothesis.
- Moreover, it could also be applied to a set of hypotheses. Doing so, it is possible to obtain the subset of hypothesis that are unfalsified by the data. And if in addition, a performance cost, which provides an ordering, is defined then we can learn the optimal hypothesis with respect to that cost.
- Furthermore, it could be applied to hierarchical learning. Hypotheses about hypotheses could be tested sequentially.

3. APPLICATION TO CONTROL AND SYSTEM IDENTIFICATION PROBLEMS

In this section, we apply the learning framework to control and system identification problems. As in [3], we restrict ourselves to a class of problems were the controller is causally-left-invertible (CLI class of controllers) and the performance specification is defined at the observations instances (OI class of specs). This class of problems has the advantage that limits the computational growth. In one hand the causally-left-invertibility property of the controller will guarantee that given an output from the controller, it could only have being generated if driven by an unique input signal. On the other hand, the performance specification given at the observations instances allows one to work in a truncated space, as explained in [3]. The restrictions associated with this class of problems are met in most applications.

3.1 Controller Unfalsification Problem

As we said in the introduction, this approach to learning sprang from the unfalsified control concept. For this reason, we will review the key results of the unfalsified control concept [20]. The unfalsified control concept provides a way to evaluate if a controller could meet a desired closed loop performance criterion based on data from the plant, without necessarily having to put the controller in the loop to run tests. The main idea was to create a "conceptual experiment" where a 'fictitious reference signal' was engineered for a controller such if it were used to drive the controller in closed loop with the plant, then it will have produced the measured plant input-output data. Then, the closed loop performance associated with this conceptual experiment was used to evaluate the performance of the controller. As such it is a very powerful tool to screen controllers. Furthermore, it formulated an adaptive control algorithm, where data from a plant with a controller in closed loop was used to evaluate a set of candidate controllers, and if the controller in loop is determined by be falsified then it switched for the 'best' controller from the subset of as yet unfalsified candidate controllers. Recently, related results on this topic have been reported by [12] wherein the fictitious reference signal is called the virtual reference signal. In sections 3.1 and 3.3 we recast some of these results under this new framework.

We consider the case in which the universum Z , in addition to containing the extended plant input-output space, also contains the controllers and performance specification input-output spaces. That is: $Z = U_P \times Y_P \times U_K \times Y_K \times U_G \times Y_G$.

Definition 3.3 *A **controller** is a mathematical model* (Z, E, k)*, with behavior* $K = \{z \in Z | k(u_K, y_K) = 0\}$.

For example in a typical unity feedback control system the controller input is $u_K = r - y_P$ where r is the reference signal, and the controller output is the input to the plant $y_K = u_P$. Hence $U_K = R \times Y_P$ and $Y_K = U_P$.

Definition 3.4 *A **performance specification** is a mathematical model* (Z,E,g)*, with behavior* $G = \{z \in Z | g(u_G, y_G) \le \gamma, \forall u_{Gi} \in U_{Gi}\}$.

Typically g is a cost function, a mapping $g : Z \to E$. A signal $z \in G$ is said to satisfy the performance specification if the cost $g(u_G, y_G)$ is smaller or equal than a threshold $\gamma \in \Re$ for all possible values of u_{Gi}.

We note that performance specifications like mixed sensitivity or model tracking can be cast in this formulation. Furthermore, in general, inputs to the cost function are $u_G = (u_P, y_P, r)$, and the result of evaluating the cost function is $y_G = \gamma$. Note that r plays the role of the u_{Gi}, the performance specification should be met for all reference signals. Hence in this example $U_G = U_P \times Y_P \times R$ and $Y_G = \Re$.

We now formalize the controller unfalsification problem defined in [20].

Problem 3.1 ***Controller unfalsification** [20]: Given experimental data* $D(\tau)$*, a candidate controller* (Z,E,k) *in the CLI class, and a closed loop performance specification* (Z,E,g) *in the OI class, determine whether the controller is unfalsified by* $D(\tau)$*, that is determine if the performance will be violated if the controller were in the loop.*

Note that unfalsification in the definition of the problem is with respect to the control problem. The hypothesis to be tested is if the controller could satisfy the performance specification if it were in closed loop with the plant that generated the data. This hypothesis contains the controller and performance specification behaviors as follows $(Z,E,h) \equiv \left(Z,E,\begin{bmatrix} k \\ g \end{bmatrix}\right) = (Z,E,k) \wedge (Z,E,g) = (Z, K \cap G)$ where $h \equiv \begin{bmatrix} k \\ g \end{bmatrix} : Z \to E$ is the behavioral equation and the corresponding behavior is:

$H = \{z \in Z | k(u_K, y_K) = 0, g(u_G, y_G) \le \gamma, \forall u_{Gi} \in U_{Gi}\}$.

The following proposition casts the solution to this problem by using definition 2.3.

Proposition 3.1 *Given experimental data* $D(\tau)$*, a candidate controller* (Z,E,k) *in the CLI class, and a closed loop performance specification* (Z,E,g) *in the OI class, then the composite hypothesis* (Z,E,h) *is said to be falsified if, and only if,*

$$\ker(h | D(\tau)) = \varnothing \qquad (2)$$

where $(Z,E,h) \equiv \left(Z,E,\begin{bmatrix} k \\ g \end{bmatrix}\right)$ *is the composite mathematical model that contains the hypothesis and performance specification behaviors. Otherwise, is said to be unfalsified.*

Remark: Note that when evaluating composite hypothesis, it is useful to evaluate if the problem is well-posed in the sense that the controller and performance specification as individual hypothesis are unfalsified by the data, and that the composite hypothesis is feasible (i.e., $\ker(h | D(\tau)) = \varnothing$ *where* $h = \begin{bmatrix} k \\ g \end{bmatrix}$). Some of these concepts were also presented in [3].

3.2 Optimal Unfalsified Control Problem

Now we consider the problem of given a set of candidate controllers, a closed loop performance criteria, and experimental data, find the candidate controller which produces the best performance.

Consider a parameterized set of candidate controllers.

Definition 3.5 *A **set of candidate controllers** is a set of hypothesis* $(Z,E,k(\theta))$ *where* $\theta \in \Theta$ *is a parameter vector, such that* $K(\theta) = \{z \in Z | k(u_K, y_K, \theta) = 0\}$.

Problem 3.2 ***Optimal Unfalsified Control:*** *Given experimental data* $D(\tau)$*, a set of candidate controllers* $(Z,E,k(\theta))$ *in the CLI class where* $\theta \in \Theta$*, a closed loop performance specification* (Z,E,g) *in the OI class. Find the best unfalsified controller.*

The solution to this problem is given in the following proposition.

Proposition 3.2 *Given experimental data* $D(\tau)$*, a set of candidate controller* $(Z,E,k(\theta))$ *in the CLI class, and a closed loop performance specification* $(Z,E,g(\gamma))$ *in the OI class, then the optimal unfalsified controller is*

$$\theta^* = \arg\min \gamma$$

$$\text{subject to } \ker(h(\theta)) \ne \varnothing$$

where $h(\theta) = \begin{bmatrix} k(\theta) \\ g \end{bmatrix}$.

Remark: This optimization problem could have no solution if all the candidate controllers happen to be falsified, or it could have multiple solutions if several unfalsified candidate controllers happen to give the same performance.

3.3 Model Unfalsification Problem

The model validation problem, more precisely unfalsification, consists on verifying that a model fits the data given a certain criteria. As in the controller unfalsification problem, the model and the fitting criteria are mathematical models. To solve the problem, the combined hypothesis is constructed and tested using definition 2.3. In this case, the criteria could be of the prediction error type and in a sense it represents the knowledge on the noise about the system. Hence, we test the hypothesis about the model being able to predict the output of the plant modulo de noise given in the data.

3.4 Unfalsified Parametric System Identification Problem

The parametric system identification problem consists on given a parametric model identifying the parameter set that fits the data for a given fitting criteria. Given a set of parametric models, identification of the best parameters could be achieved by applying the model unfalsification approach to every element of the set as it was shown for the control case. Then, picking the one that gives the smallest cost in the performance specification.

4. PRACTICAL CONSIDERATIONS

In this section we will give some practical consideration that will be valid for all the problems. An important first remark is that the unfalsification process could be run just once, or it can be run iteratively on evolving past data as each new datum is acquired. If we run it iteratively on evolving past data then it can be used for real time learning, as in the unfalsified switching control. Adaptive updates of the parameters can be done either periodically or aperiodically, or by a mixture of both. If we run it periodically, we could run it at any rate. For quickly varying systems we may want to run it at the highest possible rate (sensor sampling rate) but for slowly varying systems we could run it at a slower rate. If we decide to run it aperiodically, it may be convenient to define a function to decide when to run it based on observed performance. A more sophisticated function could be used for a mixed approach. Every time we call the learning processor, the data available defines the observations operator.

Now we present some ideas on how to choose the performance specification and hypotheses.

- *Performance Specification*: As expressed earlier without loss of generality the performance specification, could be defined using a cost function. This cost function should measure in some sense the error between the actual performance and desired one. In this case unfalsification becomes relative to cost level. The cost function could be of many different types. For example, [20] used a weighted l_{2e} norm specification of the mixed sensitivity type, [2] used a model reference specification in terms of l_1 norms, [9,26] used an analogous one with l_2 norms. For real-time application with a time varying system we suggest to introduce a forgetting factor to rest importance to the old measurements.

- *Hypotheses*. The controller (or model) hypotheses could be of many different types. However from the computational point of view, in the control problem it is quite useful if the controllers are causally left invertible since then we could use the fictitious reference signal approach, and has been widely used [1,2,4,12,18-20]. In addition, it will be also of great advantage to have it parameterized in such a way that the fictitious reference signal could be represented as a product of two vectors, one that depends on the parameters (typically through a nonlinear function) and another that contains the dynamics as in [2]. Examples of parameterizations used that meet the causally left invertible condition are the traditional parameterization used in MRAC [4,20], or parameterizations of the PID type as used in [2]. Other parameterizations that may not satisfy this condition are also possible. For example [9,26] use ARX parameterizations.

- *Data*: The experimental data available defines the observations operator. Typically, experimental data will be discrete signals of finite length. However, [9] presented a study where signals may asymptotically approach infinite duration.

Example:

Typical implementation of these algorithms is in the form of a switching system. An example of such is an unfalsified switching control design for a missile autopilot [2]. In this example, a parameterized PID-type switching controller was implemented, where the parameters were allowed to vary in a discretized interval around a nominal value. In this specific example, there were 5 parameters that lead to 3125 controllers. The switching occurs when the actual controller in the loop performance was below an l_1 criteria, which monitor the tracking error. In Figures 2 and 3 we show two plots from this study.

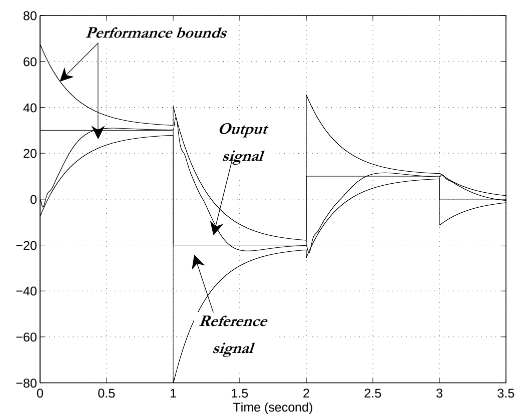

Figure 2: Reference signal, output signal, and error bounds.

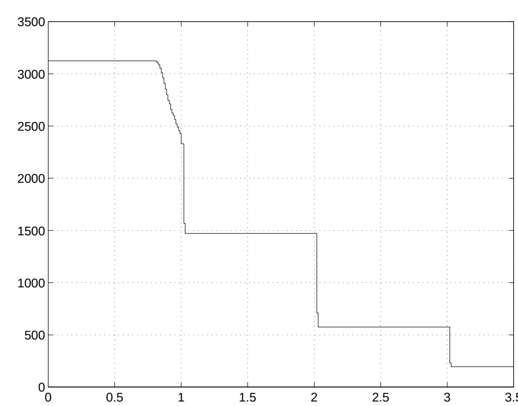

Figure 3: Evolution of the number of unfalsified controllers.

Figure 2 shows the command signal, a train of steps, the errors bounds to monitor performance, and the actual output signal of the switching system. Figure 3 shows the evolution of the number of controllers. For more information about this study please go to reference [2].

5. CONCLUSIONS

This paper provides a concise, but general mathematical framework for learning about dynamical systems through hypothesis falsification. This framework is developed in the behavioral setting. It offers a clear and distinct description of what can be learned from experimental data alone. Furthermore, this paper shows how to formulate and solve control and system identification problems within this framework.

6. ACKNOWLEDGEMENTS

Paul B. Brugarolas's research supported in part by JPL.
Michael Safonov's research supported in part by AFOSR Grant F49620-01-1-0302.

7. REFERENCES

[1] T.F. Brozenec and M.G. Safonov, "Controller invalidation," in *Proc. IFAC World Congress*, San Francisco, CA , July 19-23,1996. Oxford, England: Elsevier, pp. 224-233.

[2] P. B. Brugarolas, V. Fromion, and M. G. Safonov "Robust switching missile autopilot," in *Proc. American Control Conference*, Philadelphia, PA, June 22-26, 1998.

[3] P.B. Brugarolas and M.G. Safonov, "A canonical representation for unfalsified control in truncated spaces," in *Proc. IEEE CACSD*, Hawaii, August 22-27, 1999.

[4] F.B. Cabral and M.G. Safonov "A falsification perspective on Model Reference Adaptive Control" in *Proc. IEEE Conf. on Decision and Control*, Kobe, Japan, December 11-13, 1996.

[5] J.D. Glover and F.C. Schweppe, "Control of linear dynamic systems with set constrained disturbances," *IEEE Trans. on Automatic Control*, vol. 16, pp. 411-423, 1971.

[6] J. Helmicki, C. Jacobson, and C. Nett, "Control oriented system identification: a worst-case/deterministic approach in H-infinity," *IEEE Trans. on Automatic Control*, vol. 36, no. 10, 1991.

[7] J. Helmicki, C. Jacobson, and C. Nett, "Worst-case/deterministic identification in H-infinity: the continuous-time case, " *IEEE Trans. on Automatic Control*, vol. 37, no. 5, 1992.

[8] C. Jacobson and C. Nett, "Worst-case system identification in l1: optimal algorithms and error bounds," *System and Control Letters*, no. 19, pp. 419-424, 1992.

[9] R.L. Kosut. "Uncertainty Model Unfalsification for Robust Adaptive Control" in *Proc. IFAC Workshop on Adaptive Systems in Control & Signal P*rocessing, University of Strathclyde, Glasgow, Scotland, UK August 26-28, 1998.

[10] R.L. Kosut, M.K. Lau, and S.P. Boyd, "Set-membership identification of systems with parametric and nonparametric uncertainty," *IEEE Trans. on Automatic Control,* vol. 37, no.7, pp. 929-941, July 1992.

[11] S. Morse, "Supervisory control of families of linear set-point controllers-Part II: Robustness," in *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 13-15, 1995. New York, pp. 1750-1755.

[12] E. Mosca and T. Agnoloni, "Inference of candidate loop performance and data filtering for switching supervisory control" in *Automatica* 37 (2001) 527-534.

[13] Narendra, K.S.; Balakrishnan, J, "Adaptive control using multiple models," *IEEE Trans. on Automatic Control*, vol 42, no.2, pp171 -187

[14] J.W. Polderman and J.C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag New York. 1998.

[15] K. Poolla, P. Khargonekar, A. Tikku, J. Krause and K. Nagpal, "A time-domain approach to model validation," *IEEE Trans. on Automatic Control*, vol. 39, no. 5, pp. 951-959, May 1994.

[16] K.R. Popper. *Logic of Scientific Discovery*. 1934. (English edition: London: Routledge, 1959)

[17] M.G. Safonov. Focusing on the Knowable: Controller Invalidation and Learning, in *Control Using Logic-Based Switching*, A.S. Morse, Ed. Berlin: Springer-Verlag, 1996, pp. 224-233.

[18] M.G. Safonov and F.B. Cabral "Fitting Controllers to Data", *Systems and Control Letters* 43(2001) 299–308.

[19] M.G. Safonov and F.B. Cabral, "Robustness-oriented controller identification", *Proc. IFAC Symp. on Robust Control Design ROCOND'97*, Budapest, Hungary, June 25-27, 1997.

[20] M.G. Safonov and T.-C. Tsao, "The unfalsified control concept and learning," *IEEE Trans. on Automatic Control*, 42(6):843-847, June 1997.

[21] R. S. Smith and M. Dahleh, Eds., *The modeling of uncertainty in control system, Proc. of the 1992 Santa Barbara Workshop*, Springer-Verlag, New York, 1994.

[22] J. C. Willems, "*The analysis of feedback systems*", MIT Press, Cambridge, MA 1969.

[23] J.C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. on Automatic Control*, 36(3): 259-294, March 1991.

[24] J.C. Willems, "On Interconnections, Control, and Feedback," *IEEE Trans. on Automatic Control*, 42(2): 326 - 339, 1997.

[25] J.C. Willems and H.L. Trentelman "The Behavioral Approach to Systems and Control" *CDC 2000* Short Course. Sydney, Australia, Dec. 2000.

[26] B.R. Woodley, R.L. Kosut and J.P. How, "Uncertainty model unfalsification with simulation," in *Proc. American Control Conference*, Philadelphia, PA, June 22-26, 1998.