Local4Global

SYSTEM-OF-SYSTEMS THAT ACT LOCALLY FOR OPTIMIZING GLOBALLY

611538, FP7-ICT-2013.3.4

Deliverable D3.1:

The L4G Self-Learning Mechanism

Deliverable Version:	D3.1, v.2.0
Document Identifier:	l4g_wp4_d3.1_Certh_v2.0.pdf
Preparation Date:	August 18, 2015
Document Status:	Final
Author(s):	Elias B. Kosmatopoulos, Iakovos T. Michailidis, Christos Korkas and Christos Ravanis
Dissemination Level:	PU – Public

Project funded by the European Community in the 7th Framework Programme

EU FP7 - SMALL/MEDIUM-SCALE FOCUSED RESEARCH PROJECT (STREP), FP7-ICT-2013.3.4: ADVANCED COMPUTING, EMBEDDED AND CONTROL SYSTEMS

D) FROM ANALYZING TO CONTROLLING BEHAVIOUR OF SYSTEM OF SYSTEMS (SOS)

Deliverable SUMMARY SHEET

Deliverable Details	
Type of Document:	Deliverable
Document Reference #:	D3.1
Title:	The L4G Self-Learning Mechanism
Version Number:	2.0
Preparation Date:	August 18, 2015
Delivery Date:	August 18, 2015
Author(s):	Elias B. Kosmatopoulos, Iakovos T. Michailidis, Christos Korkas and Christos Ravanis
Document Identifier:	l4g_wp4_d3.1_Certh_v2.0.pdf
Document Status:	Final
Dissemination Level:	PU – Public

Project Details	
Project Acronym:	Local4Global
Project Title:	SYSTEM-OF-SYSTEMS THAT ACT LOCALLY FOR OPTIMIZING GLOBALLY
Project Number:	611538
Call Identifier:	FP7-ICT-2013.3.4
Call Theme:	ADVANCED COMPUTING, EMBEDDED AND CONTROL SYSTEMS
Project Coordinator:	CERTH – Centre for Research and Technology – Hellas
Participating Partners:	CERTH – Centre for Research and Technology – Hellas; ETHZ – Eidgenössische Technische Hochschule Zürich; RWTH – RWTH Aachen University; IK4 – IK4 TEKNIKER; TRV – TRANSVER GmbH; TUC – Technical University of Crete; TUM – Technische Universtität Muench
Instrument:	STREP
Contract Start Date:	October 1, 2013
Duration:	36 Months

Deliverable D3.1: Short Description

This document summarizes and describes the so-called Local4Global Self Learning Mechanism.

Keywords: system of systems; technical system of systems; systems engineering; system of systems engineering; automatic control systems; constituent system optimization; distributed control; Local4Global optimization tool.

Deliverable D3.1: Revision History

Version:	Date:	Status:	Comments
0.1	10/07/2014	Draft	First Draft by Iakovos Michailidis and Christos Korkas and Christos Ravanis (CERTH)
0.2	5/08/2014	Draft	First Revision from Elias Kosmatopoulos (CERTH)
0.3	25/8/2014	Draft	Corrections/comments by Christos Korkas (CERTH)
0.4	10/9/2014	Draft	Corrections/comments by Iakovos Michailidis (CERTH)
1.0	28/10/2014	Submitted	Certh Final Version
1.1	15/07/2015	Draft	Corrections/comments by Christos Korkas and Iakovos Michailidis (CERTH)
1.2	5/08/2015	Draft	Second Revision from Elias Kosmatopoulos (CERTH)
2.0	18/08/2015	Final	Finalized version by Christos Korkas (CERTH)

Table Of Contents

1 Executive Summary

Over the recent past years research effort has been dedicated towards addressing a generic solution in System of Systems (SoS) control problems. Two are the main obstacles that have to be bypassed in such problem cases, especially in real life applications, rendering the optimization problem into a complicated/challenging one: (i) modelling/simulation tools are usually used in order to construct an as-close-as-possible to reality accurate model, whose construction though requires a considerable amount of time and effort and (ii) furthermore, standard control system designs when applied to SoS exhibit poor performance as they are required to handle very high-dimensional problems. In this deliverable together with deliverable D3.2, we present a first attempt towards addressing these issues. More precisely, a new adaptive optimal control methodology for SoS is presented and evaluated. The main attributes of the proposed control methodology is its local nature with minimum requirements for coordination between the constituent system of the SoS and its model-free nature.

The proposed methodology comprises two different ingredients: the first of these ingredients is the so-called *self-learning mechanism* and the second one is the *situation awareness mechanism*. The first of these mechanisms – which is described and analysed in detail in the present Deliverable – is responsible for overcoming the need for a model for the SoS dynamics. Instead of using a model for the SoS dynamics, the L4G control approach embeds in each of the SOS constituent systems, a self-learning mechanism which is responsible for providing to the constituent system a "just-enough-accurate" knowledge of the overall SoS dynamics. It is emphasized that as the overall SoS dynamics may be of extremely large scale, the self-learning mechanism must be able to (i) provide the "just-enough-accurate" knowledge of the SoS dynamics by making use of the minimum possible amount of information and (ii) be as computationally simple as possible. These two requirements are met by the proposed L4G-CAO algorithm as shown in this Deliverable.

2 INTRODUCTION

The optimization and control of the operations in Systems-of-Systems (SoS) has recently attracted the interest of many researchers. The vast majority of the existing approaches assume perfect or sufficient knowledge of the dynamics of the overall SoS, i.e., the dynamics of each and every constituent system along with their interactions with the other constituent systems and the external environment, see e.g., [1, 2]. However, the requirement for perfect or sufficient knowledge of the SoS dynamics renders the overall SoS control design practically infeasible in many SoS applications, as they typically involve a large number of constituent systems with highly complex and uncertain dynamics [3, 4]. Moreover, even if the dynamics of the overall SoS were known, the current state-of-the-art in control systems is not able to provide computationally efficient, practically implementable solutions to problems of the scale, complexity, heterogeneity and constantly changing structure/topology/hierarchy of SoS. To circumvent this problem, localized and oversimplified controllers can be employed [5, 6, 7] which, however, put optimality or even efficiency at stake.

In this deliverable together with deliverable D3.2, we present a first attempt towards addressing the shortcomings of the existing control methodologies for SoS. More precisely, a new adaptive optimal control methodology for SoS is presented and evaluated. The main attributes of the proposed control methodology is its local nature with minimum requirements for coordination between the constituent system of the SoS and its model-free nature. The proposed methodology comprises two different ingredients: the first of these ingredients is the so-called *self-learning mechanism* and the second one is the *situation awareness mechanism*. The first of these mechanisms – which is described and analysed in detail in the present Deliverable – is responsible for overcoming the need for a model for the SoS dynamics. Instead of using a model for the SoS dynamics, the L4G control approach embeds in each of the SOS constituent systems, a self-learning mechanism which is responsible for providing to the constituent system a "just-enough-accurate" knowledge of the overall SoS dynamics. It is emphasized that as the overall SoS dynamics may be of extremely large scale, the self-learning mechanism must be able to (i) provide the "just-enough-accurate" knowledge of the SoS dynamics by making use of the minimum possible amount of information and (ii) be as computationally simple as possible. These two requirements are met by the proposed "just-enough-learning mechanism" as shown in this Deliverable. More precisely:

- As the ultimate goal of the L4G methodology is not to model the SoS dynamics but rather to control and optimize their performance, instead of adopting an adaptive modelling approach (e.g., adaptive system identification), we adopt an adaptive optimization approach where the goal is to optimize the SoS performance subject to the "constraint" that the SoS dynamics are not known. The benefit of adopting an adaptive optimization approach is that a *local* model of the overall performance as a function of the SoS control parameters is needed and not a global one as it is required by adaptive modelling approaches. In the adaptive optimization approach there is no need to know the SoS dynamics for the whole range of its control parameters but, rather, to know its dynamics around the current location of the SoS control parameters so as to be able to "move them" to a better location around the current one.
- Moreover, as in an SoS setup we may have to face with a huge number of information points (sensors), the proposed approach must make sure that it uses the minimum

possible information: as it is shown in the next sections, the proposed approach makes use of only local – to the constituent system – measurements plus a global measurement on the time-history of the global performance of the SoS, reducing thus to the minimum its communication requirements.

Moreover, in Section 5 of deliverable 3.2, a tutorial/example of the proposed Local4Global strategy, and it's attributes (which are presented in Deliverables 3.1 and 3.2) is presented and explained. A full tutorial about system connections, information exchange, parameters calibration and troubleshooting can be found in Section 4 of Deliverable 4.2.1

The structure of this Deliverable is as follows: In Section 2, we present the L4G optimization for the case of a static problem. The main attributes of the case and and how the L4G logic interacts with the SoS are described. Section 3, presents the L4G-CAO algorithm which is adaptively Solving the L4G static optimization problem. In Section 4, we describe how static problem is "translated" into a problem of Adaptive Fine Tuning of the controllers of an SoS, and how the L4G-CAO algorithm can be applied to such a problem.

3 Local4Global optimization for SOS: The Static Case

As already mentioned in the Introduction, the proposed L4G self-learning mechanism is being developed by adopting an adaptive optimization set-up, appropriately defined so as to meet the SoS control design requirements. For this reason, in this section we introduce the non-adaptive optimization problem suitably defined under an SoS framework and then, in the next section, we proceed with its adaptive version.

We now proceed with the definition of the non-adaptive SoS-related static optimization problem. Let us consider an SoS which consists of N constituent systems interacting with each other towards minimizing the following global performance measure (or, global cost):

$$J = J(x_1, x_2, \ldots, x_N) \tag{1}$$

where $J(\cdot)$ is a non-negative scalar nonlinear function and $x_i, i = 1, 2, \ldots, N$ denote the vectors of decision variables for the ith constituent system (note that $x_1, x_2, \ldots, x_N$ are not necessarily of the same dimension). The *Local4Global-Static Optimization Problem — L4G-SOP* is defined as follows:

Definition 1 *[Local4Global Static Optimization Problem – L4G-SOP]* We say that an iterative algorithm solves – locally – the L4G-SOP for the SoS associated with equation (1), if the following are satisfied:

- The vectors x_i of decision variables of the constituent systems are updated as follows:

$$x_i(k+1) = P_i(k, x_i(k), J(k)) \tag{2}$$

  where k denotes the iteration number, $P_i(\cdot)$ is a nonlinear function [that depends only on k, $x_i(k)$ and $J(k)$] and $J(k)$ denotes the value of the global cost function at the k-th iteration, i.e.,

$$J(k) = J\Big(x_1(k), x_2(k), \ldots, x_N(k)\Big)$$

- The global cost $J(k)$ satisfies

$$J(k) < J(k-1) \text{ if } \nabla J(\mathbf{x}(k)) \neq 0$$

  where $\mathbf{x}(k) = [x_1^\tau(k), x_2^\tau(k), \ldots, x_N^\tau(k)]^\tau$, i.e., $\mathbf{x}(k)$ denotes the augmented vector consisting of the decisions variables of all constituent systems. △

In simple words, *the above definition states that an iterative algorithm solves the L4G-SOP, if it achieves to locally optimize the cost criterion (1) by having each of the constituent systems updating its decision variables using only local information plus information about the global cost (and without requiring any information about the state of the other constituent systems).*

Apparently, there are two questions raised from the above definition: first of all, whether there exists an iterative scheme that satisfies the conditions of the above definition (i.e., the question "is the L4G-SOP solvable?") and, if yes, how to construct an algorithm that solves the L4G-SOP. The following Lemma will be proven crucial in providing a definite answer to both of these questions.

Lemma 1 Suppose that the vectors $x_i(k)$ are updated using an equation of the form (2), i.e., they are updated using only local information plus information about the global cost function. Then, under some very mild assumptions on $J(\cdot)$ we have that there exists a finite integer d and a nonlinear function $\bar{J}_i(\cdot)$ such that

$$J(k+1) \quad = \quad \bar{J}_i\Big(k, J(k), J(k-1), \dots, J(k-d), x_i(k)\Big)$$

for some nonlinear function $\bar{J}_i(\cdot)$.

Proof: The proof can be established using results on representing state-space systems by input/output models. More precisely, we can rewrite the overall dynamics in state-space form as follows:

$$\begin{aligned} \bar{x}(k+1) &= F\left(\bar{x}(k)\right) \\ y(k) &= h\left(\bar{x}(k), x_i(k)\right) \end{aligned}$$

where $\bar{x} = [x_1^\tau, x_2^\tau, \dots, x_{i-1}^\tau, x_{i+1}^\tau, \dots x_N^\tau]^\tau$, $F = [P_1^\tau, P_2^\tau, \dots, P_{i-1}^\tau, P_{i+1}^\tau, \dots P_N^\tau]^\tau$ and $y = J$. Please note that x_i is considered as an exogenous input in the above equations. Using standard results from transforming state-space into input/output systems (see e.g., Theorem 2) we can establish the proof. ◇

The above Lemma states that the global performance index $J(k)$ can be calculated – at the i constituent system level – through a nonlinear function $\bar{J}_i(\cdot)$ by using a "portion" of the previously measured values of J $(J(k), J(k-1), \dots, J(k-d))$ and the locally available information $x_i(k)$. Please note, however, that even in the case where the analytical mathematical from of the global cost $J(\cdot)$ is available, constructing the analytical mathematical form of $\bar{J}_i(\cdot)$ is not an easy task. In the next section, we will see that our proposed methodology does not need an analytical mathematical form of neither the function $\bar{J}_i(\cdot)$ nor of the global cost function $J(\cdot)$.

4 Adaptively Solving the L4G-SOP: the L4G-CAO Algorithm

Lemma 1 indicates that if it is possible to construct the function $\bar{J}_i(\cdot)$, then solving the L4G-SOP is feasible using standard optimization tools. However, as $\bar{J}_i(\cdot)$ is, in general, not possible to be analytically constructed, we can employ the so-called Cognitive Adaptive Optimization (CAO) algorithm [8, 9] which is applicable to cases of optimizing a cost function for which an analytical form is not available. The key idea of the CAO algorithm is to concurrently estimate the cost function and optimize it based on this estimation by making sure that such a concurrent accomplishment of two – sometimes, mutually competing – tasks (estimation and optimization) is efficiently performed. Next we describe how the original CAO algorithm must be revised – by making use of Lemma 1 – towards solving the L4G-SOP. We will call this algorithm the L4G-CAO algorithm in order to distinguish it from the original CAO algorithm. *It must be emphasized that the L4G-CAO algorithm does require an analytical mathematical form of the global cost $J(\cdot)$: all it needs, is that the global cost $J(\cdot)$ is available for measurement at each of the algorithm's iterations.*

The L4G-CAO Algorithm

For each constituent system and for each iteration k perform the following:

- Construct an estimator for $J(k+1)$ at the ith constituent system level as follows:

$$J(k+1) \approx \hat{J}(k+1) = \theta_i^\tau(k)\phi_i\Big(J(k), J(k-1), \ldots, J(k-d), x_i(k)\Big) \tag{3}$$

where ϕ_i is the regression vector and θ_i is the estimator vector. Standard function approximation schemes (e.g. polynomials) can be used to construct estimator (3). The reader is referred to [8, 9] for more details on how to construct such an estimator (it must be emphasized that it suffices to use estimators of very "simple" structure and not very elaborate ones). The estimation vector θ_i is constructed using standard Least-Squares (LS) estimation principles, i.e., $\theta_i(k)$ is obtained by solving the following LS optimization problem:

$$\theta_i(k) = \mathrm{argmin}_\vartheta \sum_{\ell=k-T(k)}^{k-1} \Big(\vartheta^\tau \phi_i\Big(J(\ell), J(\ell-1), \ldots, J(\ell-d), X_i(\ell)\Big) - J(\ell+1)\Big)^2 \tag{4}$$

where $T(k)$ denotes the time-window over which the LS estimation is taking place. Please note that as the integer d of Lemma 1 is not known, it suffices to use in (3) an upper bound for d.

- Choose a positive function $\alpha(k)$ to be either a constant positive function or a time descending function satisfying $\alpha(k) > 0$, $\sum_{k=0}^{\infty} \alpha(k) = \infty$, $\sum_{k=0}^{\infty} \alpha^2(k) < \infty$. Generate – randomly or pseudo-randomly – a *set of L candidate perturbations* $\delta x_i^{(1)}(k), \delta x_i^{(2)}(k), \ldots, \delta x_i^{(L)}(k)$ where $\delta x_i^{(j)}(k)$ are vectors of the same dimension as $x_i(k)$ and L is a positive integer that is larger than $2\text{dim}(x_i)$.

- Estimate the effect of each of the candidate perturbations to the current vector $x_i(k)$ by employing the estimator (3) and pick the candidate perturbation with the "best" effect, i.e., choose the vector $\delta x_i^{(j^*)}(k)$ that satisfies

$$\begin{aligned} \delta x_i^{(j^*)}(k) = \text{argmin}_{j=1,\ldots,L} \theta_i^\tau(k) \phi_i \Big(& J(k), J(k-1), \ldots, \\ & J(k-d), x_i(k) + \alpha(k)\delta x_i^{(j)}(k) \Big) \end{aligned} \tag{5}$$

- Set

$$x_i(k+1) = x_i(k) + \alpha(k)\delta x_i^{(j^*)}(k)$$

Note 1: The above algorithm does not need information about what is happening to the rest of the agents. All it needs is information about the "local control vector" $x_i(k)$ as well as information about the global cost $J(k)$.

Note 2: The choice of the parameters $T(k), \alpha(k)$ and the regressor vector $\phi(\cdot)$ is the same as in the original CAO algorithm [8, 9]. The reader is referred to [8, 9] for details and guidelines on the choice of the parameters.

Note 3–The Self-Learning Mechanism: What is important to notice is that by using the L4G-CAO algorithm, each of the constituent systems has its own learning mechanism [given by equation (4)] that is used for optimizing its own control variables. It must be also emphasized that there is no need for an elaborate and "large-in-dimension" estimator or approximator to be used for learning: all the constituent system needs to know is a local approximation of $\bar{J}_i(\cdot)$ around the current value of x_i so as to be able to move x_i towards a location around its current one that improves $\bar{J}_i(\cdot)$.

The next Theorem establishes the properties of the L4G-CAO algorithm. In essence, the next Theorem establishes that the L4G-CAO algorithm approximately solves the L4G-SOP problem.

Theorem 1 *The L4G-CAO algorithm guarantees that*

$$\mathbf{x}(k+1) = \mathbf{x}(k) - A(k)\nabla J(\mathbf{x}(k)) + \epsilon(k)$$

where $A(k)$ is a positive definite matrix and $\epsilon(k)$ is a term that converges exponentially fast to the subset $\mathcal{D} = \{\epsilon(k) : |\epsilon(k)| \leq \epsilon\}$ where ϵ is a positive constant that can be made arbitrarily small [by increasing the "size" of the estimator (3)].

Proof: By using Lemma 1, it can be seen that the proof is exactly the same as the proofs of the main results of [8, 9]. ◇

Note: In simple words, the above Theorem states that the L4G-CAO algorithm approximately solves the L4G-SOP in the sense that the global cost function satisfies

$$J(k) < J(k-1) + \mathcal{O}(\epsilon(k))$$

In other words, the L4G-CAO algorithm guarantees convergence of $\mathbf{x}$ arbitrarily close to a local minimum of $J(\cdot)$.

5 L4G-AFT: Application to Adaptive Fine-Tuning of model-based controllers for SoS

The L4G-SOP can be seen to be seen to be equivalent to the problem of L4G Adaptive Fine Tuning of the controllers of an SoS. As a matter of fact, this is the approach we will follow within the Local4Global project for integrating the distributed model based controllers to be developed by the partner ETHZ with the model-free advances of the partner CERTH. Next, we describe how the L4G-CAO algorithm can be applied to such a problem.

Let us assume an SoS whose ith constituent system dynamics evolve according to the following difference equation:

$$z_i(t+1) = g_i(z_1(t), \ldots, z_i(t), \ldots, z_N(t), u_i(t), d_i(t)) \tag{6}$$

where z_i, u_i, d_i denote the vectors of the ith constituent system states, control inputs, and exogenous signals, respectively, t denotes the time-index, and $g_i(\cdot)$ is a – possibly unknown – non-linear vector function. Moreover, assume that the following control law is applied to the ith constituent system

$$u_i(t) = \varpi(\theta_i, z_i(t)) \tag{7}$$

where $\varpi(\cdot)$ is a known vector function and θ_i is the vector of control parameters of the controller of the ith constituent system. The performance of the overall SoS, is evaluated through the following objective function (performance index)

$$J\left(\theta; z(0), D_T\right) = \pi_T(z(T)) + \sum_{t=0}^{T-1} \pi_t\left(z(t), u(t)\right) \tag{8}$$

where π_t are known nonnegative functions, T denotes the time-horizon over which the control law (7) is applied z, u, d denote the vectors of *global* states, controls and disturbances, i.e.,

$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}, u = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}, d = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix},$$

and $D_T \triangleq [d(0), d(1), \ldots, d(T-1)]$ denotes the time-history of the exogenous signals. By defining $x = vec\left(z_0, D_T\right)$, (8) may be rewritten as $J\left(\theta, x\right) = J\left(\theta; z_0, D_T\right)$.

The problem of Adaptive Fine Tuning (AFT) of the control parameters is – by evaluating the performance of the controller (7) for different values of the controller vector θ – to "tune" θ so as it converges close to a local minimum of J while keeping, during the fine-tuning experiments, J as small as possible. The fine-tuning process involves many different experiments (of duration T); during each experiment the vector θ remains constant and after the experiment ends, the performance of the controller (for the particular choice for θ) is evaluated using the performance index J. Note that the performance index J is a function of the closed-loop system (6), (7) dynamics; as a result J depends on the – possibly unknown – function $g(\cdot)$ and thus it is not possible, in general, to obtain an analytic form for J and its gradient.

The AFT problem has been addressed in the papers [8, 9], where efficient algorithms have been proposed for its solution. The solutions, however, proposed in [8, 9], assume a *centralized* form, with all control parameters being tuned using information about the overall system states. In an SoS-setup, such a centralized form is not practically implementable: instead, each constituent system control parameters θ_i must be updated using only local information (plus information about the global criterion time-history). Thus, in an SoS-setup it makes sense to address the L4G-AFT problem which is similar to the original AFT by additionally requiring that θ_i are updated using only local information plus information about the global cost time history.

6 Conclusions

The deliverable present and evaluate a new adaptive optimal control methodology for Systems of Systems (SoS). The main attributes of the proposed control methodology is its local nature with minimum requirements for coordination between the constituent system of the SoS and its model-free nature. In this deliverable, we focus in the second attribute of the proposed methodology, the self-learning mechanism. This attribute is responsible for overcoming the need for a model for the SoS dynamics. Instead of using a model for the SoS dynamics, the L4G control approach embeds in each of the SOS constituent systems, a self-learning mechanism which is responsible for providing to the constituent system a "just-enough-accurate" knowledge of the overall SoS dynamics. The proposed adaptive algorithm L4G-CAO, manages to offer this "just-enough" information for each sub-system with (i) the minimum data transfer and (ii) the minimum computational cost.

References

[1] Mehdi Maasoumy Haghighi. Controlling energy-efficient buildings in the context of smart grid: A cyber physical system approach. (UCB/EECS-2013-244), Dec 2013.

[2] Dongfei Fu, El-Houssaine Aghezzaf, and Robin De Keyser. A model predictive control framework for centralised management of a supply chain dynamical system. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):250–260, 2014.

[3] Aditya Kumar and Prodromos Daoutidis. Nonlinear dynamics and control of process systems with recycle. *Journal of Process Control*, 12(4):475 – 484, 2002.

[4] Joseph Z. Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice*, 11(8):847 – 858, 2003. Process Dynamics and Control.

[5] Ralph M. Hermans, Andrej Jokić, Mircea Lazar, Alessandro Alessio, Paul P.J. van den Bosch, Ian A. Hiskens, and Alberto Bemporad. Assessment of non-centralised model predictive control techniques for electrical power networks. *International Journal of Control*, 85(8):1162–1177, 2012.

[6] Dongfei Fu, Clara M Ionescu, El-Houssaine Aghezzaf, and Robin De Keyser. Decentralized and centralized model predictive control to reduce the bullwhip effect in supply chain management. *Computers & Industrial Engineering*, 73:21–31, 2014.

[7] Marina Murillo Guido Sanchez, Leonardo Giovanini and Alejandro Limache. *Distributed Model Predictive Control Based on Dynamic Games Ch.4*. 2011.

[8] E. B. Kosmatopoulos. An adaptive optimization scheme with satisfactory transient performance. *Automatica*, 45(3):716–723, 2009.

[9] E. B. Kosmatopoulos and A. Kouvelas. Large scale nonlinear control system fine-tuning through learning. *Neural Networks, IEEE Transactions on*, 20(6):1009–1023, 2009.