52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

A "Plug-n-Play" Computationally Efficient Approach for Control Design of Large-Scale Nonlinear Systems using co-Simulation

Simone Baldi, Iakovos Michailidis, Hossein Jula, Elias B. Kosmatopoulos and Petros A. Ioannou

Abstract— **Recently, there has been a growing interest towards *simulation-based control design (co-simulation)*, where the controller utilizes an optimizer to minimize or maximize an objective function (system performance) whose evaluation involves simulation of the system to be controlled. However, existing simulation-based approaches are not able to handle in a computationally efficient way large-scale optimization problems involving hundreds or thousands of states and parameters. In this paper, we propose and analyze a new simulation-based control design approach, employing an adaptive optimization algorithm capable of efficiently handle large-scale control problems. The convergence properties of the proposed algorithm are established. Simulation results exhibit efficiency of the proposed approach when applied to large-scale problems. The simulation results employ two large-scale real-life systems for which conventional popular optimization techniques totally fail to provide an efficient simulation-based control design.**

Index Terms— **Simulation-based Control Design, Co-Simulation, Optimization, Large Scale System.**

I. Introduction

Recently, there has been a growing interest towards *simulation-based control design (co-simulation)*, where the controller utilizes an optimizer to minimize or maximize an objective function (system performance) whose evaluation involves simulation of the system to be controlled, see e.g., [1], [2]. The advantages of such an approach are many: the controller design does not require any simplification or approximation of the actual system as in conventional state-space control design. Moreover, the controller is verified and evaluated using realistic conditions that can include all different physical constraints, delays, atypical behavior and peculiarities that are present in real-life. Finally, the control design can be performed in a "plug-n-play" fashion without the need to get involved in a tedious and time-consuming analysis of the system properties and characteristics. Unfortunately, as simulation-based control design employs optimizers that are called to operate over a complex simulation model, their efficiency may become quite problematic especially when they are applied to *large scale systems* that may involve thousands of states and parameters.

In this paper, we propose and analyze a new simulation-based control design approach which overcomes the computational problem. The basic "ingredient" of the proposed approach is the *Cognitive-based Adaptive Optimization (CAO)* [3]-[4]. By combining the CAO algorithm with a recently introduced approximate solution to the Hamilton-Jacobi-Bellman equation [5], a new optimization algorithm is developed that is specifically tailored for the needs and objectives of simulation-based control design, with the capability of efficiently handling large-scale problems.

The convergence properties of the proposed algorithm are established. Simulation results exhibit the rapid and efficient convergent of the proposed approach. These simulation results employ two large-scale real-life systems (a traffic network and an energy efficient building) for which *conventional popular optimization techniques totally fail to provide an efficient simulation-based control design.*

S. Baldi, I. Michailidis, and E. B. Kosmatopoulos are with Informatics & Telematics Institute, Center for Research and Technology Hellas (ITI-CERTH), Thessaloniki, Greece `sbaldi@dsi.unifi.it`, {`michaild`, `kosmatop`}`iti.gr`.

H. Jula is with Dept. of Electrical Engineering, Penn State Harrisburg, PA USA `huj2@psu.edu`.

P. A. Ioannou is with Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA `ioannou@usc.edu`.

II. Problem Formulation

A. Problem Set-up and Exposition Assumptions

The general set-up of a simulation-based control design is depicted in Fig. 1. a simulation model of the system is connected to a parameterized controller whose parameters are updated by some kind of optimizer. For each choice of the controller parameters, the closed-loop system is simulated (hence the term co-simulation approach) and a quantitative performance measure is provided to the optimizer. A simulation-based control design iterative procedure can be summarized as:

[Step 0] The parametrized controller to be optimized is initialized with some initial controller parameters.

[Step 1] The controller parameters are used to simulate the closed-loop system performance over the whole *simulation period*. Each simulation period may involve testing/simulating the same controller under different scenarios (different initial conditions and different characteristics for the exogenous factors) so as to make sure that the optimized controller is be able to efficiently handle many different possible real-life situations. The optimizer receives the *total system performance* over the control horizon.

[Step 2] The optimizer calculates the new controller parameters in an attempt to decrease the system performance at the next time step and Step 1 is executed again. This iterative procedure (Steps 1-2) is continued over many iterations until an optimum in performance is reached.

Before we continue on further analyzing the problem and presenting the proposed solution, we will impose some "exposition assumptions" to the problem, allowing to keep the notation as simple as possible.

Assumption 1: Optimal Control Set-up. Analysis is provided for the case where the system dynamics are purely con-

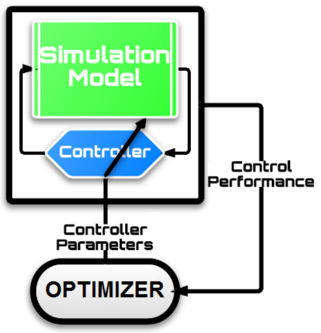


Fig. 1. Simulation-based Control Design (Co-Simulation) Set-up.

tinuous, no delays or exogenous signals are present and the overall state vector is available for measurement. The ideas presented in this paper can be extended, by appropriately revising the HJB equation, to more general problems (*e.g.*, the HJB equation can be replaced by its stochastic version in case of stochastic exogenous signals or by the HJB-Isaacs equation in case of bounded deterministic exogenous signals, a state-observer can be included in case where not all of the system states are available for measurement).

Having the above exposition assumption in mind, it is quite general to assume that the simulator (system) dynamics have the following form:

$$\begin{aligned} \dot{\chi}(t) &= F(\chi(t), \upsilon(t)) \\ & \bar{C}(\chi, \upsilon) \leq 0 \end{aligned} \tag{1}$$

where χ, υ denote the system state and control vectors, respectively, and F and $\bar{C}$ are nonlinear vector functions corresponding to the system dynamics and system constraints, respectively. Moreover, the system performance over a simulation period can be described as follows:

$$\bar{J} = \int_0^\infty \bar{\Pi}(\chi(s), \upsilon(s)) ds \tag{2}$$

where $\bar{\Pi}$ is an appropriate nonlinear function that corresponds to the *instantaneous cost*.

Using similar arguments as in [5], we can transform (1) and (2) into a more suitable for our purposes form. More precisely, by employing pre-integrator using fictitious control and penalty functions (see [5] for more details), the system dynamics (1) can be transformed as follows:

$$\dot{x}(t) = f(x(t)) + Bu(t), \;\; B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{3}$$

while the system performance can be transformed as follows:

$$J = \int_0^\infty \Pi\left(x(s)\right) ds \tag{4}$$

where x, u are the transformed system state and control vectors and f, Π are nonlinear functions that depend on $F, \bar{C}$ and $\bar{\Pi}$. The forms (3) and (4) possess certain advantages over (1) and (2): the system dynamics depend linearly on the control vector u and, moreover, the presence of nonlinear constraints is avoided.

Assumption 2: Controller structure. The approach is exposed for the case where the controller switches among different linear controllers depending on the operating conditions of the system (PieceWise Linear–PWL control) [6], [7], although extensions to nonlinear controllers (*e.g.*, polynomial controllers) can be considered. Let us consider a PWL controller which takes the form:

$$u = K_i x, \forall x \in \mathcal{S}_i, \; i \in \{1, \dots, L\} \tag{5}$$

where u, x are the system control and state vectors, respectively, K_i are constant control matrices of appropriate dimensions and $\mathcal{S}_i, i = 1, \dots, L$ denotes a partition of the state space into L disjoint subsets.

B. P-based PWL Controller Approximation

The PWL controller (5) is not smooth. Since the proposed approach requires the use of a smooth controller, the so-called *mixing signals* $\beta_i(x)$ are introduced [5], [8]. The mixing signals are designed so as to be smooth and to satisfy $\beta_i(x) \approx 1$ if $x \in \mathcal{S}_i$ and $\beta_i(x) \approx 0$ if $x \notin \mathcal{S}_i$. Using such a design we can then use a controller which instead of activating only the i-th linear controller whenever $x \in \mathcal{S}_i$, it activates all linear controllers for which the respective mixing signal is not negligible. In such a way, the overall controller is smooth. Many different choices for selecting the mixing signals $\beta_i(x)$ have been proposed in the literature; see, *e.g.* [9], [10].

Beside, instead of optimizing directly the controller matrices K_i, we consider an *overparametrized* version of these matrices. More precisely, we use the following "smoothed" version of the PWL controller (5):

$$u = -B^T M_z(x) P z(x) \tag{6}$$

where $M_z(x)$ is the Jacobian matrix of $z(x)$ with respect to x and

$$z(x) = \begin{bmatrix} \sqrt{\beta_1(x)}x \\ \sqrt{\beta_2(x)}x \\ \vdots \\ \sqrt{\beta_L(x)}x \end{bmatrix}, P = \begin{bmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_L \end{bmatrix} \tag{7}$$

where P_i are *positive definite matrices*. It has to be emphasized that standard algebraic manipulations can be used to establish that the controller (6) admits the following form

$$u = \sum_{i=1}^L \bar{\beta}_i(x) K_i x \tag{8}$$

where $\bar{\beta}_i(x)$ are again mixing signals [that depend on $\beta_i(x)$] and K_i depend on the elements of P_i and B. In other words, *the controller (6) is a "smoothed" and overparametrized version of the original PWL controller (5). The reason we use such an overparameterized controller will be made in the next section.*

III. The PCAO Algorithm

Despite the significant progresses made in optimal nonlinear control theory [11], [12], [13], the existing methods are, in general, not applicable to LSS because of the computational difficulties associated to the solution of the Hamilton-Jacobi partial differential equations. In the following an approximated optimal control is formulated, where a suitable approximation of the solution of the Hamilton-Jacobi-Bellman equation is employed.

A. HJB Approximation

According to the Hamilton-Jacobi-Bellman (HJB) equation, the controller that optimizes the system performance [given by the criterion (4)] can be obtained as the solution of the following partial differential equation:

$$\dot{V}^*(x(t)) = \left(\frac{\partial V^*}{\partial x}\right)^T (f(x) + Bu^*) = -\Pi(x) \quad (9)$$

where $V^*(x)$ denotes the so-called *optimal cost-to-go function* and u^* denotes the optimal controller, which can be seen to satisfy

$$u^* = -B^T \left(\frac{\partial V^*}{\partial x}\right) \quad (10)$$

Given the mixing signals $\beta_i(x)$ described in the previous section, it was seen in [5] that there exist positive definite matrices P_i such that the optimal-cost-to-go function $V^*(x)$ can be approximated as follows

$$V^*(x) \quad = \quad V(x) + \mathcal{O}(1/L), \quad (11)$$

where $V(x) = \sum_{i=1}^{L} \beta_i(x) \left(x^T P_i x\right) = z^T(x) P z(x)$. Therefore the optimal controller u^* given in (10) can be approximated as follows:

$$\begin{aligned} u^* &= -B^T \left(\frac{\partial V}{\partial x}\right) + \mathcal{O}(1/L) & (12) \\ &= -B^T M_z(x) P z(x) + \mathcal{O}(1/L) & (13) \end{aligned}$$

In other words, the optimal controller u^* can be approximated by the P-based controller form given in equation (6) [or, equivalently, equation (8)]. This is actually the reason we choose to use the form of controller (6) within the proposed approach.

Remark 1: The typical case in approaches that employ approximation arguments is that they possess the drawback that they are efficient only if the approximation is "accurate enough", i.e., if the approximation terms $\mathcal{O}(1/L)$ are "small enough". Such a requirement, in practice, may have the implication that a large number L of mixing signals should be employed. It has to be emphasized that, although our approach employs approximation arguments, it does not suffer from the above drawback. As a matter of fact, our approach is able to provide with efficient solutions even in cases where the approximation terms $\mathcal{O}(1/L)$ are "large". This is due to the use of the stochastic approximation (adaptive optimization) [3] nature of the PCAO algorithm which achieves to compensate for the effect of the approximation terms $\mathcal{O}(1/L)$ by "averaging their effect to zero" .

B. The PCAO Algorithm

By using the approximations (11) and (12) and integrate (9) in the interval $[t, t+\delta t)$, where $\delta t > 0$ is a sufficiently small discretization step, one can see that in case the optimal controller u^* were applied then,

$$\Delta V(x(t)) \approx -\int_t^{t+\delta t} \Pi(x(s))\, ds + \mathcal{O}(1/L) \quad (14)$$

where $\Delta V(x(t)) = V(x(t+\delta t)) - V(x(t))$. Having the above equation in mind and the approximations provided in the previous section, let us assume that the following controller is applied to the system:

$$\hat{u} = \hat{u}(x(t); \hat{P}) = -B^T M_z(x) \hat{P} z(x) \quad (15)$$

where $\hat{P}$ denotes an estimate of the unknown matrix P. Let us also define the following "error" term

$$\varepsilon\left(x(t), \hat{P}\right) = \Delta\hat{V}(t) + \int_t^{t+\delta t} \Pi(x(s)))\, ds \quad (16)$$

where

$$\hat{V} = \hat{V}(x(t); \hat{P}) = z^T(x) \hat{P} z(x) \quad (17)$$

and $\Delta\hat{V}(t) = \hat{V}(x(t+\delta t)) - \hat{V}(x(t))$. By using equation (14) it can be seen that the 'error" term $\varepsilon\left(x(t), \hat{P}\right)$ provides us with a "measure" of how far the estimate $\hat{P}$ is from its optimal value P. More precisely, by using equation (14) and the above definition for $\varepsilon\left(x(t), \hat{P}\right)$, it can be seen after some algebraic manipulations that

$$\varepsilon\left(x(t), \hat{P}\right) = \mathcal{F}\left(x(t), \hat{P}\right) + \mathcal{O}(1/L) \quad (18)$$

where $\mathcal{F}\left(x(t), \hat{P}\right)$ is a term satisfying

$$\mathcal{F}\left(x(t), \hat{P}\right) = 0 \Longleftrightarrow \hat{P} \equiv P \quad (19)$$

i.e., it becomes zero iff $\hat{P} \equiv P$. Using the above equation, one may employ the standard gradient descent methods [14] for updating of $\hat{P}$, i.e.,

$$\hat{P}_{t+\Delta t} = \hat{P}_t - \eta \nabla_{\hat{P}} \varepsilon^2 \left(x(t), \hat{P}\right), \ \eta > 0 \quad (20)$$

in an attempt to minimize the error term $\varepsilon\left(x(t), \hat{P}\right)$ and, thus, to have $\hat{P}$ converge as close as possible to its optimal value P. However, gradient descent methods require the knowledge of $\nabla_{\hat{P}} \varepsilon^2 \left(x(t), \hat{P}\right)$ which might no be available; besides they must be constrained in order to take into account of the positive definite constraints; finally, the term $\mathcal{O}(1/L)$ in (18), which acts as a disturbance. might lead to divergence of the algorithm.

To overcome all the above problems, we employ the so-called Cognitive-based Adaptive Optimization algorithm [3], [15] appropriately extended for the needs of our problem. More precisely, a revised version of the CAO algorithm (abbreviated as PCAO) is proposed and described in Table I. The main features of this algorithm are as follows:

• Steps 1, 2 and 5 in Table I are exactly the same as in the original CAO algorithm (as applied to the problem at hand, i.e., the problem of minimizing $\varepsilon\left(x(t),\hat{P}\right)^2$). It has to be emphasized that *the CAO algorithm does not require explicit and analytic formulas of the objective criterion to be optimized or its gradient* and, as a result, it can be directly applied to the problem at hand. One of the key attributes of CAO is that it employs adaptive and stochastic approximation techniques for estimating/learning the objective function. As it was shown in [3], [15], the employment of these techniques results in an adaptive optimization scheme that is equivalent to an *approximate gradient descent method* without the need for calculating the gradient of the objective criterion or the use of any type of differentiation.

• The CAO algorithm is revised in this paper so as to keep $\hat{P}$ positive definite (see Step 4 in Table I). The particular nature of the CAO algorithm allows the straightforward incorporation of a computationally simple and efficient scheme that constraints $\hat{P}$ to be positive definite. This is done without requiring the use of elaborate SDP schemes and, most importantly, without introducing any numerical problems.

• What is also crucial in the proposed algorithm is the use of an appropriate *resetting scheme* as follows: the estimate matrix $\hat{P}(t)$ generated at the t-th simulation step is evaluated by also simulating its effect to the closed-loop system over the whole simulation period and the best $\hat{P}_{best}(t)$ is calculated among all matrices $\hat{P}(s),\ s=0,\Delta t,2\Delta t,\dots,t$ generated so far (see step 3 of the algorithm). Then, the next estimate $\hat{P}(t+\Delta t)$ is calculated as an update of $\hat{P}_{best}(t)$ and not of $\hat{P}(t)$. In other words, if the current estimate matrix $\hat{P}(t)$ has not produced the best performance over the whole simulation period then it is being reset to $\hat{P}_{best}(t)$ (i.e., it is being reset to one of the past values $\hat{P}(s),\ s=0,\Delta t,2\Delta t,\dots,t-\Delta t$. In such a way, any possible instantaneous destabilizing effect of the "disturbance" term $\mathcal{O}(1/L)$ is avoided, as it can be seen by using similar arguments as in [15]. It has to be emphasized that in cases where $\hat{P}(t)$ is being reset to one of its past values, this does not mean that the information collected at the t-th iteration is "thrown away": this information is incorporated within the learning scheme of PCAO (see step 2) so as to be suitably exploited.

The proposed algorithm guarantees convergence to a minimum of the function $\mathcal{F}^2\left(x(t),\hat{P}\right)$ as summarized in the following theorem:

Theorem 1: The PCAO algorithm described in Table I, guarantees that $\hat{P}_t$ converges almost surely to the following set:

$$\mathcal{E}=\left\{\hat{P}:\hat{P}\text{ is positive definite and }\nabla_{\hat{P}}\mathcal{F}^2\left(x(t),\hat{P}\right)=0\right\}$$

Please note that the set $\mathcal{E}$ does not depend on the term $\mathcal{O}(1/L)$.

Proof: See [15]. ■

Remark 2: The stochastic approximation nature of the proposed algorithm (i.e., the fact that it incorporates the use of random perturbations, see Step 4 in Table I) has the ability to "average out" the effect of the disturbance term $\mathcal{O}(1/L)$. As a matter of fact and by using similar arguments as in the proofs of the main results of [3], [15], [4], it can be seen that the proposed algorithm satisfies

$$\begin{aligned}\hat{P}_{t+\Delta t} &= \Big[\hat{P}_t-\alpha(t)\nabla_{\hat{P}}\mathcal{F}^2\left(x(t),\hat{P}\right)\\ &\quad +e(t)+\alpha(t)\xi(t)\mathcal{O}(1/L)\Big]_{\mathcal{P}} \qquad (21)\end{aligned}$$

where the symbol $[(\cdot)]_{\mathcal{P}}$ is used to denote the convex operation of projecting $(\cdot)$ into the set of positive definite matrices, $e(t)$ is a term that exponentially decays to zero and $\xi(t)$ is a zero-mean random term (introduced due to the use of random perturbations in Step 4 of the algorithm). By using the fact that $\xi(t)$ is zero-mean, the properties of $\alpha(t)$ (see Table I) and the Robbins and Siegmud theorem [16] on nonnegative almost-super-martingales, it can be seen in a similar way as in the proof of Theorem 2 in [3] that the effect of the term $\alpha(t)\xi(t)\mathcal{O}(1/L)$ in (21) is "averaged to zero".

IV. Applications And Results

The proposed algorithm has been tested in two Large Scale Systems, one in the area of control of large buildings and the second one in the area of urban traffic control.

Table II shows the improvement of PCAO strategy respect to Rule Base Controllers (shortened as RBCs), which are the control strategies actually implemented in the two systems: in both cases, PCAO strategy achieves better performance than rule-based. For comparison purposes, PCAO strategy is compared with well established constrained global optimization algorithms, incorporated in Matlab Optimization Toobox. The compared algorithms did not accomplish to converge to any minima even after thousands of iterations (see Table III).

A. First Test Case: TUC building

As a first numerical example, a 10-office building located in Chania, Greece was considered. A model of the building has been developed in EnergyPlus, a simulation environment for energy buildings [17].

The presented simulations consider the problem of operating air-conditioners during summer, to cool-climate the rooms in an energy efficient manner, while satisfying a user comfort level. The simulations use historical weather data collected in July-August of 2010.

The cost function to be minimized is a combined criterion which takes into account the energy consumption and the user comfort:

$$Total_{score}=s{*}Energy_{score}+(1-s){*}Comfort_{score} \qquad (24)$$

where $0\le s\le 1$ regulates the importance between the two cost terms. The Fanger index, evaluating the Predicted Percent of Dissatisfied people, has been used as a measure of comfort.

The two tested RBCs employ the simple control strategies:

- RBC1: HVAC set points to $24°C$ (during office hours)
- RBC2: HVAC set points to $25°C$ (during office hours)

Table I: The PCAO Algorithm

Initialize.

i. Set $t = 0$. Let also T denote the simulation time of the overall *simulation period*.
ii. Choose positive constants $\varepsilon_1 < \varepsilon_2$ and a positive integer T_h.
iii. Initialize $\hat{P}(0)$ to be a positive definite matrix satisfying the constraints $\varepsilon_1 I \preceq \hat{P}(0) \preceq \varepsilon_2 I$.
iv. Choose a positive function $\alpha(t)$ to be either a constant positive function or a time descending function satisfying $\alpha(t) > 0, \sum_{t=0}^{\infty} \alpha(t) = \infty, \sum_{t=0}^{\infty} \alpha^2(t) < \infty$

[The reader is referred to [3], [15], [4] on guidelines for the choice of $T_h, \alpha(t)$ as well as of the LIP estimator in Step 2; also, in [5], guidelines are provided on the choice of the constants $\varepsilon_1, \varepsilon_2$.]

Step 1. Apply the controller (15) with $\hat{P} = \hat{P}(t)$ for the next *simulation step* and calculate $\varepsilon\left(x(t), \hat{P}\right)$ [cf. equation (16)].

Step 2. Construct a linear-in-the-parameters (LIP) estimator of $\varepsilon\left(x(t), \hat{P}\right)$ as follows:

$$\hat{\varepsilon}\left(x(t); \hat{P}(t)\right) = \theta^T \phi\left(x(t); \hat{P}(t)\right)$$

$$\theta = \arg\min_{\vartheta} \sum_{i=t-\delta t T}^{t} \left(\varepsilon\left(x(i); \hat{P}(i)\right) - \vartheta^T \phi\left(x(i); \hat{P}(i)\right)\right)^2$$

where ϕ, θ are the regressor vector and the parameters vector of the estimator, respectively and $T = \min\left(\frac{t}{\delta t}, T_h\right)$

Step 3. Apply the controller (15) with $\hat{P} = \hat{P}(t)$ over the whole *simulation period* (i.e., for $s = 0, \Delta t, \ldots, T$) and calculate $P_{best}(t)$ to be the *best* $\hat{P}$ *obtained so far*, i.e.,

$$\hat{P}_{best}(t) = \arg \min_{\hat{P}(s), s=0,\Delta t, \ldots t} \left\{\sum_{\tau=0}^{T} \varepsilon(x(\tau); \hat{P}(s))\right\}^2 \tag{22}$$

Step 4. Generate N perturbed candidates (*random perturbations*) of $\hat{P}_{best}(t)$:

$$\hat{P}_{cand}^{(i)} = (1 - \alpha(t))\, \hat{P}_{best}(t) + \alpha(t)\Delta\hat{P}^{(i)}, \quad i = 1, 2, \ldots, N$$

where $\Delta\hat{P}^{(i)}$ are random symmetric positive definite matrices with the same structure as in (7), and satisfying $\varepsilon_1 I \preceq \Delta\hat{P}^{(i)} \preceq \varepsilon_2 I$.

Step 5. The controller to be applied at the next time-step, *i.e.*, $\hat{P}(t + \delta t)$ is:

$$\hat{P}(t + \delta t) = \arg \min_{\hat{P}_{cand}^{(i)}} \left\{\hat{\varepsilon}\left(x(t); \hat{P}_{cand}^{(i)}\right)\right\}^2 \tag{23}$$

Step 6. Set $t = t + \delta t$ and GO TO **Step 1**.

Table II: PCAO Application Improvements

Test Case	No. States	No. Controls	No. Mixing Signals	$\varepsilon_1/\varepsilon_2$	Convergence Iterations	Improvement Percentage(%)
Building	46	10	1	0.0001/0.03	100-200	30-40
Building	46	10	4	0.0001/0.03	200-500	35-40
Traffic Network	122	46	1	0.001/0.01	30-70	12-23
Traffic Network	122	46	4	0.001/0.01	50-60	10-31

Table III: Optimization Algorithms Tests

CHANIA BUILDING			CHANIA TRAFFIC		
Optimization Algorithm	Convergence Iterations	Improvement (%)	Optimization Algorithm	Convergence Iterations	Improvement (%)
PCAO	$\sim$ 100	30-40	PCAO	$\sim$ 70	10-30
patternsearch	>4000	0	ga	>4000	0
fmincon	>6000	0	fmincon	>6000	0

The PCAO algorithm employs a centralized control strategy, performed by using a feedback vector of 46 components including the 10 zone temperatures and humidities, the 10 HVAC setpoints, external weather conditions and forecasts for the following 6 hours. The resulting number of controller parameters is $1081 \times L$. Due to lack of space, only the results for $L = 1$ are shown: apart from some small intervals of time, the PCAO total score is typically significantly smaller than both RBCs total scores (Fig. 2).

What is also important to note, is the convergence rate of PCAO to accomplish the control design. According to many different simulation experiments (for different weather conditions and different choices of L), PCAO needs 100-500 iterations to accomplish an efficient control result.

B. Second test case: Chania traffic network

The overall Chania traffic network comprises of 16 controlled junctions with a total of 42 control inputs (green times of the junctions phases) and 122 sensor measurements (total of 61 loop-detectors in the network providing occupancy and flow measurements). An AIMSUN-based simulation model was developed. By using real-life data 80 different traffic demand scenarios were simulated.

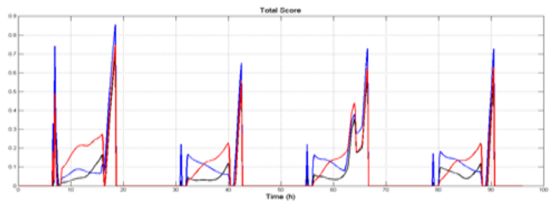
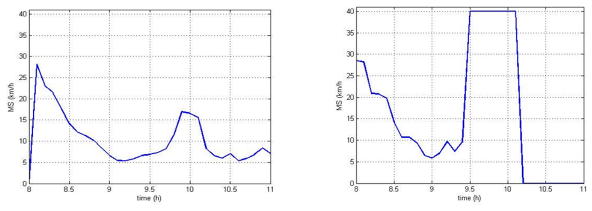


Fig. 2. Control Scenario $L = 1$: Total Score for RBC1 (blue), RBC2 (red) and PCAO (black).

The urban traffic control strategy TUC [18], actually implemented in the Chania network was chosen as the RBC. In this application, the PCAO algorithm is responsible for controlling the control parameters related to the Cycle Control Module (for the cycle time) and the Split Control module (for the green time), for a total of $26106 \times L$ optimization parameters. The cost criterion, the PCAO Control System Design Tool is called to optimize (maximize) is the Mean Speed over the whole traffic network.

Table IV shows the convergence characteristics of the PCAO Control System Design. It must be underlined that PCAO Control Design converges in few steps (40-70) and achieves $10 - 30\%$ cost improvements over another well-established and very efficient strategy(TUC).

Fig. 3(a) and 3(b) depicts the different behavior between PCAO control system and the TUC system (Base Case). The PCAO controller achieves to clear the congestion very fast (at around 9:30). After the congestion is cleared, the network operates in 'free-flow' conditions (mean speed $40\ km/h$) till all cars have been served (mean speed drops to 0). On the other hand, the Base Case controller is unable to clear the congestion at around 9:30 which results a second congestion after 10:00 (small mean speed).



(a) Base Case controller (b) PCAO Controller

Fig. 3. Traffic network mean speeds during the day

V. Conclusion

An algorithm for simulation-based control design, namely PCAO, has been exposed. The proposed algorithm minimizes a performance index, related to how close the applied control action satisfies the Hamilton-Jacobi-Bellman equation associated to the optimization problem. A model of the system is used to assess the future performance of the control action. The scalability of the controller, together with the algorithm employed to solve the approximated Hamilton-Jacobi-Bellman equation, make the proposed methodology capable of handling very large control problems, thus being particularly suitable for the control of Large Scale Systems. Two large scale numerical examples have been used in order to show the effectiveness of the method, and its capability of handling efficiently large-scale control design optimization problems.

Acknowledgements

The research leading to these results has been partially funded by the European Commission FP7-ICT-5-3.5, Engineering of Networked Monitoring and Control Systems, under the contract #257806 (AGILE).

References

[1] T. Nghiem and G. H. Pappas. Receding-horizon supervisory control of green buildings. *American Control Conference. San Francisco, CA*, 2011.
[2] M. Trcka, J. L. M. Hensen, and M. Wetter. Co-simulation of innovative integrated hvac systems in buildings. *Journal of Building Performance Simulation*, 2:209–230, 2009.
[3] E. B. Kosmatopoulos. Clf-based control design for unknown multi-input nonlinear systems with good transient performance. *IEEE Transactions on Automatic Control*, 55:2635–2640, 2010.
[4] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos. Multi-robot 3d coverage of unknown areas. *International Journal of Robotics Research*, in press, 2012.
[5] S. Baldi, I. Michailidis, E. B. Kosmatopoulos, A. Papachristodoulou, and P. A. Ioannou. Convex design control for practical nonlinear systems. *IEEE Trans. on Automatic Control*, submitted.
[6] A. Rantzer and M. Johansson. Piecewise linear quadratic optimal control. *IEEE Trans. on Automatic Control*, 45:629–637, 2000.
[7] L. Rodrigues. Stability analysis of piecewise-affine systems using controlled invariant sets. *Systems & Control Letters*, 53:157–169, 2004.
[8] S. Baldi, E. B. Kosmatopoulos, K. Aboudolas, D. Rovas, A. Papachristodoulou, and P. A. Ioannou. Nonlinear control of large-scale complex systems using convex optimization tools and self-adaptation. *50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA*, 2011.
[9] M. Johansson and A. Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE Trans. on Automatic Control*, 43:555–559, 1998.
[10] K. M. Passino and S. Yurkovich. *Fuzzy Control*. Addison Wesley Longman, Menlo Park, CA, 1998.
[11] T. Basar and P. Bernhard. *$\mathcal{H}_\infty$-Optimal Control and Related Minimax Design Problems*. Birkhauser, Boston, MA, 1995.
[12] W. M. Lu and J. C. Doyle. $\mathcal{H}_\infty$ control of nonlinear systems: A convex characterization. *IEEE Trans. Automat. Contr.*, 40:1668–1675, 1995.
[13] M. R. James and J. S. Baras. Partial observed differential games, infinite-dimensional hamilton-jacobi-isaac equations, and nonlinear $\mathcal{H}_\infty$ control. *SIAM J. Contr. Optimiz.*, 34:1342–1364, 1996.
[14] D. Bertsekas. *Nonlinear Programming, 2nd Edition*. Athena Scientific, 1999.
[15] E. B. Kosmatopoulos and A. Kouvelas. Large-scale nonlinear control system fine-tuning through learning. *IEEE Transactions Neural Networks*, 20:1009–1023, 2009.
[16] H. Robbins and D. Siegmund. A convergence theorem for nonnegative almost supermartingales and some applications. In J. S. Rustari, editor, *Optimizing methods in statistics*, pages 233–257. New York: Academic Press, 1971.
[17] D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisher, and M. Witte. Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33:319–331, 2001.
[18] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91:2043–2067, 2003.