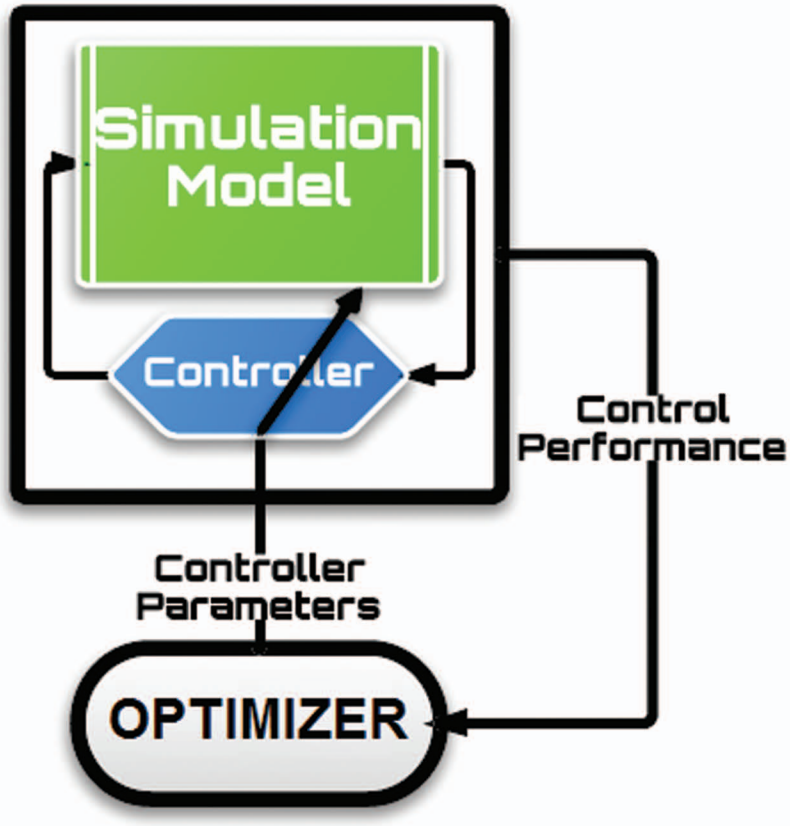

SIMONE BALDI,
IAKOVOS MICHAILIDIS,
ELIAS B. KOSMATOPOULOS,
and PETROS A. IOANNOU

As mathematical programming techniques and computer capabilities evolve, control designs are required to address control problems of ever-increasing scale and complexity. A particularly promising methodology toward such a purpose is *simulation-based control design (cosimulation)*. In cosimulation, the controller utilizes an optimizer to minimize or maximize a cost function, related to system performance, whose optimization involves an iterative process of system simulation and controller redesign.

A "Plug and Play" Computationally Efficient Approach for Control Design of Large-Scale Nonlinear Systems Using Cosimulation

A COMBINATION OF TWO "INGREDIENTS"

Digital Object Identifier 10.1109/MCS.2014.2333272
Date of publication: 16 September 2014

Two large-scale numerical examples have been used to demonstrate the effectiveness of the method.

The main applications, which gave a boost to the development of simulation-based designs, include decision making in manufacturing systems or operational and managerial decision support in other discrete-event processes (see [1]–[5] and references therein for monographs and survey papers). The advantages of such an approach are many. The controller design does not require any simplified or approximated state-space model of the actual system. Moreover, the controller is verified and evaluated using realistic conditions, including physical constraints, delays, and atypical behaviors occurring during the real-life operations of the system. Finally, the control design can be performed in a "plug and play" fashion. The notion of plug and play is that the control design is performed without a tedious and time-consuming analysis of the system properties or of the control design. Instead, the control design is directly driven by input/output data coming from the system, without requiring, for example, any knowledge of a state-space model for the system and its properties. Unfortunately, as simulation-based control design employs optimizers that are called to operate over an elaborate and complex simulation model, their efficiency may become problematic, especially when they are applied to *large-scale systems* (LSS) that involve a large number of states, control inputs, and parameters (see [6]–[10] for some recent large-scale applications). In fact, in most cases, the lack of a mathematical model of the system to be controlled makes the gradient computation of the cost function infeasible. For this reason, gradient descent or similar methods employing Jacobian and Hessian matrices [11] are not implementable, and optimization is entrusted to *derivative-free* optimization methods, which might show extremely slow convergence in the case of LSS.

This article describes a computationally efficient simulation-based control design approach that has the capability of handling optimization problems arising from large-scale nonlinear systems, with fast convergence properties and low computational requirements. This simulation-based control design is a combination of two different "ingredients." The first ingredient is the *Cognitive-based Adaptive Optimization* (CAO) algorithm, which is an adaptive optimization algorithm developed in [12] and [13] and applied to different control problems, including large-scale traffic control systems, energy efficient buildings, and multirobot teams [14]–[16]. Although CAO is a derivative-free optimization algorithm, theoretical investigations performed in [12] and [13] showed that its update step is close to the gradient descent direction so that the CAO performs essentially as the classical gradient-descent optimization method without requiring, however, analytic forms of the cost function and its gradient.

The second ingredient of the proposed algorithm is an approximate solution of the Hamilton-Jacobi-Bellman (HJB) equation and the transformation of the nonconvex optimal control problem associated with the HJB equation into a *convex* problem [17], [18].

The suitable combination of the two ingredients results in a simulation-based optimization algorithm with some key characteristics: the algorithm (abbreviated as PCAO because the adopted control policy is parameterized via a positive definite matrix P) does not require the availability of an analytical model of the system and retains the derivative-free and the "plug and play" nature of the CAO algorithm. Most importantly, the incorporation of an HJB-related optimization criterion constrains the search space to that of the controller parameters that satisfy an additional necessary condition defined by the HJB equation instead of the whole space. The result is an optimization algorithm with global convergence properties, which can efficiently handle large-scale optimization problems. Such an aforementioned constraint of the search space is made possible without requiring elaborate and computationally demanding tools. In general, the incorporation of such a constraint into standard optimization algorithms requires the use of semidefinite programming (SDP) tools, which significantly increase the computational costs and, moreover, possess severe numerical instability problems when applied to large-scale problems. The PCAO approach avoids this high computational burden.

The purpose of this article is to describe the main features of the PCAO algorithm, analyze its convergence and stability properties, and demonstrate its efficiency using simulations of two large-scale, real-life systems (a traffic network and an energy-efficient building) for which *conventional optimization techniques fail to provide an efficient simulation-based control design.*

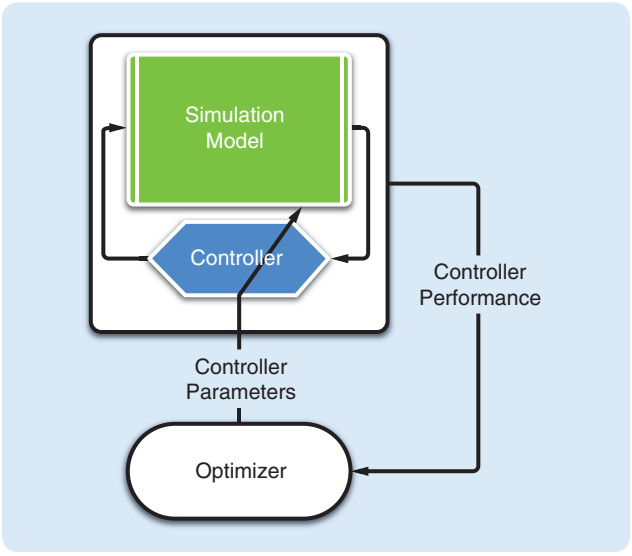

FIGURE 1 A simulation-based control design (cosimulation) setup.

Finally, PCAO is described and analyzed for the case where the HJB-based approximation results in a *nonconvex* optimization formulation. A suitable transformation converts the nonconvex problem into a convex optimization. A switching algorithm is proposed that combines the advantages of the two versions of PCAO, namely, the nonconvex and convex optimization formulations, respectively.

BACKGROUND

Despite the significant progress made in optimal nonlinear control theory [19]–[21], the construction of efficient control strategies for LSS still remains an open and challenging problem. Existing methods are, in general, not applicable to LSS because of the computational difficulties associated with the *exact* solution of the HJB partial differential equation. Similarly, the model predictive control of nonlinear systems [22], [23] also faces dimensionality issues: in most cases, predictive control computations for nonlinear systems amount to numerically solving online a nonconvex, high-dimensional mathematical programming problem.

Direct collocation methods numerically solve the optimal control problem by approximating the state and inputs using an appropriate function approximation and formulating a constrained optimization problem aiming at finding the coefficients of the function approximations [24], [25]. In this case, the collocation method yields a set of equality conditions, referred to as the *matching conditions* for the states. Although collocation methods typically work better than single-shooting methods, they can get stuck in local minima due to the nonlinear dependence on the state nodes.

Fuzzy methods have been used to solve the optimal control problem [26], [27]. Genetic algorithms are typically used to solve the corresponding optimization problem. These algorithms are known not to scale well as the dimension of the problem increases.

TABLE 1 The PCAO problem formulation.

Optimal Control Setup

To simplify the notation, mathematical analysis is provided for the case where the system dynamics are purely continuous, the entire state vector is available for measurement, and no delays or exogenous signals are present. Moreover, to avoid lengthy mathematical expressions, the simulation horizon is assumed to be infinite.

The control design problem can be formulated as an optimal control problem

$$\min \bar{J} = \int_0^\infty \bar{\Pi}(\chi(s), \upsilon(s))\,ds \tag{1}$$

$$\text{s.t.} \quad \dot{\chi}(t) = F(\chi(t), \upsilon(t)), \tag{2}$$

$$\bar{C}(\chi, \upsilon) \le 0,$$

where χ, υ denote the simulator's state vector and controls vector and $\bar{\Pi}, F, \bar{C}$ are nonlinear vector functions that correspond to the system (simulator) instantaneous performance, dynamics, and constraints, respectively. By backstepping the control input $\dot{\upsilon}(t) = u(t)$, and by putting the nonlinear constraints as soft constraints inside the cost function, the optimal control problem (1), (2) is transformed into

$$\min J = \int_0^\infty \Pi(x(s))\,ds \tag{3}$$

$$\text{s.t.} \; \dot{x}(t) = f(x(t)) + Bu(t), \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{4}$$

where x, u denote the transformed system state and control vectors, respectively. Below, without loss of generality, an optimal control problem in the form (3), (4) is considered.

Controller Structure

For scalability reasons, a control design where the controller switches among different linear controllers depending on the operating conditions of the system—PWL control—is considered. For more general controller structures, see Remark 1.

COSIMULATION PROBLEM SETUP

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

- **[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/realizations for the exogenous factors) so as to ensure that the optimized controller is able to

efficiently handle many different possible real-life situations. The optimizer receives the *total system performance* over the control time horizon.

» **[Step 2]** The optimizer calculates the new controller parameters in an attempt to improve 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 convergence is reached.

In most cosimulation methodologies [6]–[9], the system performance measure is the integral over the control horizon of a cost function that is to be minimized or maximized. In the PCAO framework, the system performance is based on a close-to-optimality index defined by the integral version of the HJB equation. The smaller the close-to-optimality index, the more accurately the applied control action is able to satisfy the HJB equation and come closer to achieving optimality. The consequence is a reduction of the search space where the optimizer must search for the controller parameters. Instead of searching among all the possible controller parameters, only the controller parameters that satisfy the HJB more closely are considered.

Problem Formulation

The problem formulation of the optimization algorithm is presented in Table 1, and the approach can be easily extended to include more general cases than those described. As it will be explained in the next sections, the proposed approach is based on HJB-based arguments. The proposed approach can be extended to include a wider class of systems by appropriately revising the HJB equation. For instance, when stochastic disturbances are present, the deterministic HJB equation can be replaced by its stochastic version. In the presence of bounded disturbances, the HJB equation can be replaced by the HJB-Isaacs equation. For the case where not all system states are measured, the HJB equation can be revised to include the presence of a state estimator (and the respective state-estimation error). In this respect, this approach conceptually covers many more cases even though each case has to be analyzed and evaluated separately.

P-BASED PWL CONTROLLER APPROXIMATION

Consider the smooth piecewise-linear (PWL) control

$$u = -\sum_{i=1}^{L} \beta_i(x) K_i x, \tag{5a}$$

where $\beta_i(x), i = 1, \ldots, L$, is a set of *mixing signals* [28] designed to satisfy

$$\sum_{i=1}^{L} \beta_i(x) = 1, \text{ for all } x$$
$$\beta_i(x) \approx 1 \text{ if } x \in S_i, \quad \beta_i(x) \approx 0 \text{ if } x \notin S_i, \tag{5b}$$

where $S_i, i = 1, \ldots, L$ denotes a partition of the state space into L disjoint subsets. Instead of activating only the ith linear controller whenever $x \in S_i$ as in conventional PWL control [29], [30], the controller (5) mixes, in proximity of the boundary of each partition S_i, the linear controllers for which the respective mixing signal is not negligible. In such a way, the overall controller is smooth.

Remark 1

The use of the mixing functions (5b) encompasses a general class of control policies, like PWL controller structures [31] and fuzzy controllers arising from Takagi-Sugeno descriptions [32]. In the latter case, the mixing functions have the form of fuzzy membership functions. Nonlinear approximations arising from sum-of-squares (or piecewise sum-of-squares) tools [33] can be included in this framework as well. An extension of the proposed formulation so as to deal with more general classes of hybrid systems is under study.

Instead of finding the optimum values of the controller matrices K_i, the optimum values of an *overparameterized* version of these matrices is calculated. More precisely, an overparametrized version of the controller (5),

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

is used, where the symbol "prime" is used to indicate transpose, $M_z(x)$ is the Jacobian matrix of $z(x)$ with respect to x, and

$$z(x) = \begin{bmatrix} \sqrt{\beta_1(x)}\, x \\ \sqrt{\beta_2(x)}\, x \\ \vdots \\ \sqrt{\beta_L(x)}\, x \end{bmatrix}, \tag{7}$$

$$P = \begin{bmatrix} P_1 & 0 & \ldots & 0 \\ 0 & P_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & P_L \end{bmatrix}, \tag{8}$$

where P_i are *positive definite matrices*. By defining $z_i(x) = \sqrt{\beta_i(x)}\, x$, straightforward algebraic manipulations can be used to establish that the controller (6) is equivalent to

$$u = -\sum_{i=1}^{L} \sqrt{\beta_i(x)}\, B' M_{z_i}(x) P_i x, \tag{9}$$

where $M_{z_i}(x)$ is the Jacobian matrix of $z_i(x)$ with respect to x. Thus, *the controller (6) is an overparametrized version of the smooth PWL controller (5). The reason for using such an overparameterized controller is motivated by the solution of the HJB equation as explained in the next section*. In the case of constraints on the control action or more general state/input constraints, the vector $z(x)$ in (7) can be augmented with additional terms depending on how the constraints are satisfied or violated. The interested reader is referred to [34] for more details. Close to optimality with respect to the HJB follows with a few modifications in the problem formulation.

The PCAO Algorithm

According to the HJB equation, the controller that optimizes the system performance [given by the criterion (3)] can be obtained as the solution of the partial differential equation

$$u^* = \underset{u}{\operatorname{argmin}} \left\{ \left(\frac{\partial V^*}{\partial x} \right)' (f(x) + Bu) + \Pi(x) \right\}, \tag{10}$$

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

$$u^* = -B' \left(\frac{\partial V^*}{\partial x} \right). \tag{11}$$

Given the mixing signals $\beta_i(x)$ described in the previous section, it was shown in [28] that there exist positive definite matrices $P_i = P_i' \succ 0$ such that the optimal-cost-to-go function $V^*(x)$ can be approximated as

$$V^*(x) = V(x) + O(1/L), \tag{12}$$

where $V(x) = \sum_{i=1}^{L} \beta_i(x)(x'P_i x) = z'(x)Pz(x)$, the term $O(1/L)$ stands for the error introduced due to approximation, and L is the number of mixed controller components defined in (5). Therefore the optimal controller u^* given in (11) can be approximated as

$$u^* = -B' \left(\frac{\partial V}{\partial x} \right) + O(1/L),$$

which, after some algebraic manipulations, can be rewritten as

$$u^* = -B' M_z(x) P z(x) + O(1/L). \tag{13}$$

In other words, the optimal controller u^* can be approximated by the P-based controller given in (6).

By using the approximations (12) and (13) and integrating (10) over the interval $[t, t+\delta t)$, where $\delta t > 0$ denotes the controller sampling time, it follows that the use of the optimal controller u^* implies

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

where $\Delta V(x(t)) = V(x(t+\delta t)) - V(x(t))$. Given (14) and the previous assumptions, assume the controller

$$\hat{u} = \hat{u}(x(t); \hat{P}) = -B' M_z(x) \hat{P} z(x) \tag{15}$$

is applied to the actual system, where $\hat{P}$ denotes an estimate of the unknown matrix P. Define the "error" term

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

where $\hat{V} = \hat{V}(x(t); \hat{P}) = z'(x)\hat{P}z(x)$ and $\Delta \hat{V}(t) = \hat{V}(x(t+\delta t)) - \hat{V}(x(t))$. Also define

$$\mathcal{E}(\hat{P}) = \sum_{t=0}^{T} \varepsilon^2(x(t), \hat{P}), \tag{17}$$

where T denotes the total simulation time, that is, the time required for a simulation period. Using (14) and (16), it can be seen that $\mathcal{E}(\hat{P})$ provides a measure of how far the estimate $\hat{P}$ is from its optimal value P. More precisely, by using (14) and the definition for $\varepsilon(x(t), \hat{P})$, it can be seen that

$$\mathcal{E}(\hat{P}) = \mathcal{F}(\hat{P}) + O(1/L), \tag{18}$$

where $\mathcal{F}(\hat{P})$ satisfies

$$\mathcal{F}(\hat{P}) = 0 \iff \hat{P} \equiv P. \tag{19}$$

Using this equation, a standard gradient descent method for updating $\hat{P}$,

$$\hat{P}_{\text{new}} = \hat{P}_{\text{old}} - \eta \nabla_{\hat{P}} \mathcal{E}(\hat{P}_{\text{old}}),\ \eta > 0, \tag{20}$$

can be employed, in an attempt to minimize the error term $\mathcal{E}(\hat{P})$ and, thus, to have $\hat{P}$ converge as close as possible to its optimal value P. When attempting to apply (20), three shortcomings arise.

- An analytic expression for the gradient $\nabla_{\hat{P}} \mathcal{E}(\hat{P})$ in (20) is needed. Such an analytic expression is practically impossible to obtain as it involves calculations that require an analytic expression for the overall simulation model.
- Since the matrix $\hat{P}$ must remain positive definite throughout the application of the gradient descent algorithm (20), a *constrained gradient descent* algorithm must be implemented to guarantee that $\hat{P}$ remains positive definite. The replacement of the unconstrained gradient descent by its constrained version is not trivial. SDP constraints are complex nonlinear functions with respect to the elements of the $\hat{P}$ matrix, which can be incorporated into optimization methods (by using, for example, penalty functions and generalized Lagrangian multiplier) only at the expense of increased computational complexity. Most importantly, the incorporation of SDP constraints into optimization methods typically introduces numerical instability problems, which result in inefficient and nonconvergent performance even in problems of medium scale.
- Another problem comes from the fact that the term $O(1/L)$ in (19) acts as a disturbance. Due to the presence of this term, the best the gradient algorithm can guarantee is convergence of $\hat{P}$ to $P + O(1/L)$ that may be far from P in cases where $O(1/L)$ is not negligible. Moreover, such a disturbance term may lead to divergence of the gradient descent algorithm; even in cases where this term is "small," the term may destroy the convergence properties of the gradient descent algorithm.

To overcome all of the above problems, the CAO algorithm [12], [16] is appropriately extended to solve the

TABLE 2 The PCAO algorithm.

Initialize.

i) Set $k=0$. Let T denote the simulation time of the overall *simulation period*.

ii) Choose positive constants $\varepsilon_1 < \varepsilon_2$ and positive integers N, 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 scalar function $\alpha(k)$ satisfying

$$\alpha(k) > 0, \lim_{k\to\infty}\alpha(k)=0, \quad \sum_{k=0}^{\infty}\alpha(k)=\infty, \quad \sum_{k=0}^{\infty}\alpha^2(k)<\infty.$$

(The reader is referred to [12], [13], and [16] for guidelines on the choice of $N, T_h, \alpha(k)$ as well as of the linear-in-the-parameters (LIP) estimator in Step 2; also, in [28], guidelines are provided on the choice of the constants $\varepsilon_1, \varepsilon_2$.)

Step 1. Apply the controller (15) with $\hat{P}=\hat{P}(k)$ for the next *simulation period* and calculate $\varepsilon(x(t),\hat{P}(k))$ and $\varepsilon(\hat{P}(k))$ and [cf. (16) and (17)].

Step 2. Construct a LIP estimator of $\mathcal{E}(\hat{P})$ as

$$\mathcal{E}(\hat{P})=\theta'\phi(\hat{P}),$$
$$\theta=\operatorname*{argmin}_{\vartheta}\sum_{i=k-\tau}^{k}\left(\mathcal{E}(\hat{P}(i))-\vartheta'\phi(\hat{P}(i))\right)^2,$$

where ϕ and θ are the regressor vector and the parameters vector of the estimator, respectively, and $\tau=\min\{k,T_h\}$.

Step 3. Calculate $P_{\text{best}}(k)$ to be the *best* $\hat{P}$ *obtained so far*

$$\hat{p}_{\text{best}}(k)=\operatorname*{argmin}_{s=0,1,\ldots,k}\{\mathcal{E}(\hat{p}(s))\}. \tag{21}$$

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

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

where $\Delta\hat{P}^{(i)}$ are random symmetric positive definite matrices with the same structure as in (8) and satisfying $\varepsilon_1 I \preceq \Delta\hat{P}^{(i)} \preceq \varepsilon_2 I$.

Step 5. The controller to be applied for the next simulation period is

$$\hat{P}(k+1)=\operatorname*{argmin}_{\hat{P}_{\text{cand}}^{(i)}}\hat{\mathcal{E}}(\hat{P}_{\text{cand}}^{(i)}). \tag{22}$$

Step 6. Set $k=k+1$ and GO TO **Step 1**.

close-to-optimal problem. This leads to the revised version of the CAO algorithm (abbreviated as PCAO) presented in Table 2. The main features of this algorithm are as follows:

- *Explicit and analytic formulas of the objective criterion to be optimized or of its gradient are not required*. The PCAO algorithm employs adaptive and stochastic approximation techniques for estimating (aka learning) the objective function. As shown in [12] and [16], the employment of these techniques results in an adaptive optimization scheme that is equivalent to an *approximate gradient descent method* that does not need to calculate the gradient of the objective criterion or the use of any type of differentiation.
- The SDP constraints introduced by the HJB-related objective criterion (positive definiteness of the matrix $\hat{P}$) are incorporated into the PCAO optimization algorithm (see Step 4 in Table 2) without requiring the use of SDP tools and constraints but only by using a simple property of the positive semidefinite cone.
- Crucial in the algorithm is the use of a *simulation-based resetting policy*: the best $\hat{P}_{\text{best}}(k)$ is calculated among all matrices $\hat{P}(s)$, $s=0,1,2,\ldots,k$ tested up to that instant of time (see Step 3 of the algorithm). Then, the next estimate $\hat{P}(k+1)$ is calculated as an update of $\hat{P}_{\text{best}}(k)$ and not of $\hat{P}(k)$. In other words, if the current estimate matrix $\hat{P}(k)$ has not produced the best performance over the entire simulation period, then it is reset to the best of the past values of $\hat{P}(s)$, $s=0,1,2,\ldots,k$. In such a way, any possible instantaneous destabilizing effect of the "disturbance" term $O(1/L)$ is avoided, as can be seen by using similar arguments as in [16]. In cases where $\hat{P}(k)$ is reset to one of its past values, the information collected at the kth iteration is not "thrown away." Instead, this information is incorporated in the learning scheme of the PCAO estimator (see Step 2) so as to be suitably exploited.

Convergence properties of the algorithm are established in Theorem 1.

Theorem 1

The PCAO algorithm described in Table 2 guarantees that $\hat{P}$ converges almost surely to the set

$$\mathcal{D}=\{\hat{P}:\hat{P}\text{ is positive definite and }\nabla_{\hat{P}}\mathcal{F}(\hat{P})=0\}.$$

Proof

See "Proof of Theorem 1." ■

Remark 2

The set $\mathcal{D}$ to which the algorithm converges does not depend on the term $O(1/L)$. The stochastic approximation nature of the algorithm (the fact that the use of *random perturbations* is incorporated; see Step 4 in Table 2) has the ability to "average out" the effect of the disturbance term $O(1/L)$, with the result that the algorithm guarantees convergence to a minimum of the function $\mathcal{F}(\hat{P})$.

Proof of Theorem 1

The proof of Theorem 1 uses technical results from [12], [13], and [16], whose proofs are not repeated here to avoid unnecessary duplication of results already published elsewhere. The interested reader is referred to these references for more details on these technical results.

Consider the LIP estimator defined in Step 2 of Table 2, and define

$$\theta^* \in \operatorname*{argmin}_{\vartheta} \sup_{\varepsilon_1 I \preceq \hat{P} \preceq \varepsilon_2 I} \left| \mathcal{E}(\hat{P}) - \vartheta' \phi(\hat{P}) \right|,$$

and $\nu(\hat{P}) = \mathcal{E}(\hat{P}) - \theta^{*\prime}\phi(\hat{P})$. The calculation of the candidate perturbations in Step 4 of the algorithm guarantees that $\hat{P}(k)$ satisfies

$$\varepsilon_1 I \preceq \hat{P}(k) \preceq \varepsilon_2 I, \quad \hat{P}(k+1) - \hat{P}(k) = O(\alpha(k)). \tag{S1}$$

The application of standard Taylor expansion arguments and (S1) gives that

$$\begin{aligned} \mathcal{E}(\hat{P}(k+1)) &= \mathcal{E}(\hat{P}(k)) + \operatorname{vec}(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\mathcal{E}(\hat{P}(k)) \\ &\quad + O\left(|\operatorname{vec}(\Delta\hat{P}(k))|^2 \varepsilon_2\right) \\ &= \mathcal{E}(\hat{P}(k)) + \operatorname{vec}(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\mathcal{E}(\hat{P}(k)) \\ &\quad + \alpha^2(k) O(\varepsilon_2), \end{aligned} \tag{S2}$$

where $\Delta\hat{P}(k) = \hat{P}(k+1) - \hat{P}(k)$ and $\operatorname{vec}(\cdot)$ denotes the standard vec operator that stacks the columns of a matrix into a vector. As shown in [12], [13], and [16], the use of the candidate perturbations in Step 4 renders the regressor vector ϕ persistently exciting. Therefore, standard results in online parameter estimation (see, for example, [42]) can be used to establish that the estimate $\hat{\mathcal{E}} = \vartheta'\phi$ of $\mathcal{E}$ satisfies

$$\begin{aligned} \mathcal{E}(\hat{P}(k)) &= \theta'(k)\phi(\hat{P}(k)) + \epsilon(k) + O\left(\Delta\hat{P}(k-1)\nu(\hat{P})\right) \\ &= \hat{\mathcal{E}}(\hat{P}(k)) + \epsilon(k) + \alpha(k) O(\nu(k)), \end{aligned} \tag{S3}$$

where $\epsilon(k)$ is a scalar signal that converges to zero exponentially fast. Combining (S2) and (S3) and using (S1) gives that

$$\begin{aligned} \mathcal{E}(\hat{P}(k+1)) &= \mathcal{E}(\hat{P}(k)) + \operatorname{vec}(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) \\ &\quad + O(\epsilon(k)) + \alpha(k) O\left(vec(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\nu(\hat{P}(k))\right) \\ &\quad + \alpha^2(k) O(\varepsilon_2) \\ &= \mathcal{E}(\hat{P}(k)) + \operatorname{vec}(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) \\ &\quad + O(\epsilon(k)) + \alpha^2(k) O\left(|\nabla_{\operatorname{vec}(\hat{P})}\nu(\hat{P}(k))|\right) \\ &\quad + \alpha^2(k) O(\varepsilon_2) \\ &= \mathcal{E}(\hat{P}(k)) + \operatorname{vec}(\Delta\hat{P}(k))'\nabla_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) \\ &\quad + O(\epsilon(k)) + \alpha^2(k) O(\varepsilon_2) + \alpha^2(k) O(\varepsilon_2). \end{aligned} \tag{S4}$$

As was established [12], [13], and [16], the choice of $\Delta\hat{P}(k)$ according to Step 5 of the algorithm, guarantees that

$$\operatorname{vec}(\Delta\hat{P}(k))\nabla'_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) = -\alpha(k) O\left(|\nabla'_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k))|^2\right)$$

and (S4) becomes

$$\begin{aligned} \mathcal{E}(\hat{P}(k+1)) &= \mathcal{E}(\hat{P}(k)) - \alpha(k) O\left(|\nabla'_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k))|^2\right) \\ &\quad + O(\epsilon(k)) + \alpha^2(k) O(\varepsilon_2). \end{aligned} \tag{S5}$$

Using (S3), the facts that $\alpha(k) > 0$, $\lim_{k\to\infty}\alpha(k) = 0$, $\sum_{k=0}^{\infty}\alpha(k) = \infty$, $\sum_{k=0}^{\infty}\alpha^2(k) = \infty$, and that $\lim_{k\to\infty}\varepsilon_k = 0$, direct application of the Robbins and Siegmud theorem [S6] to the above equality establishes that

$$\begin{aligned} \lim_{k\to\infty}\Delta'_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) = 0 \Rightarrow \lim_{k\to\infty}&(\nabla'_{\operatorname{vec}(\hat{P})}\hat{\mathcal{E}}(\hat{P}(k)) + O(\epsilon) \\ &+ \alpha(k) O(\nabla'_{\operatorname{vec}(\hat{P})}\nu(\hat{P}(k)))) = 0 \\ \Rightarrow \lim_{k\to\infty}&\nabla'_{\operatorname{vec}(\hat{P})}\mathcal{E}(\hat{P}(k)) = 0, \end{aligned}$$

which establishes the proof.

Remark 3

Approaches that employ approximation arguments typically possess the drawback that they are efficient only if the approximation is "accurate enough," that is, if the approximation terms $O(1/L)$ are "small enough." Such a requirement, in practice, may have the implication that a large number L of mixing signals is required. Although the PCAO employs approximation arguments, it does not suffer from the above drawback. The PCAO approach is able to provide efficient solutions even in cases where the approximation terms $O(1/L)$ are "large," due to the use of the stochastic approximation (adaptive optimization) [12] nature of the PCAO algorithm that compensates for the effect of the approximation terms $O(1/L)$ by "averaging their effect to zero."

Overcoming Local Minima

Theorem 1 guarantees convergence of the PCAO algorithm to one of the minima of the $\mathcal{F}(\hat{P})$. According to (19), the global minimum of this function is for $\hat{P} = P$ (since $\mathcal{F}(\hat{P}) = 0 \Leftrightarrow \hat{P} = P$). However, in the general case, $\mathcal{F}(x(t), \hat{P})$ is a nonconvex function with respect to $\hat{P}$, and as a result, the PCAO algorithm may remain trapped into local minima.

The use of the random perturbations provides the PCAO algorithm the potential to escape local minima. More precisely, the use of the random perturbations makes PCAO [see (42) in the proof of Theorem 1] equivalent to a gradient method affected by noise. Therefore, as in the case of simulated annealing, the noise (random perturbations) may be used for escaping local minima. However, such an ability may require large random perturbations, especially in the

TABLE 3 The convex PCAO algorithm.

Initialize.

i) Set $k=0$ Let T denote the simulation time of the overall *simulation period*.

ii) Choose positive constants $\varepsilon_1 < \varepsilon_2$ and positive integers N, T_h.

iii) Initialize vector $\hat{p}(0)$ satisfying the constraints $\sum_i \hat{p}_i = \hat{p}_i \in (-1,+1]$.

iv) Choose a positive scalar function $\alpha(k)$ satisfying $\alpha(k) > 0$, $\lim_{k\to\infty}\alpha(k)=0$,

$$\sum_{k=0}^{\infty}\alpha(k)=\infty,\ \sum_{k=0}^{\infty}\alpha^2(k)<\infty.$$

v) Set $\lambda=\sqrt{\alpha(k)}$.

Step 1. Apply the controller (27) with $\hat{p}=\hat{p}(t)$ over the next *simulation period* and calculate $Y(x(t),\hat{p}$ [cf. (29)].

Step 2. Construct a LIP estimator of $\mathcal{Y}(\hat{P})=\sum_{t=0}^{T}Y^2(x(t),\hat{p})$ as

$$\hat{\mathcal{Y}}(\hat{p})=\theta_c'\varphi(\hat{p}),$$
$$\theta_c=\arg\min_{\vartheta}\sum_{i=k-\tau}^{k}\left(\mathcal{Y}(\hat{p}(i))-\theta_c'\varphi(\hat{p}(i))\right)^2,$$

where φ,θ are the regressor and parameter vectors of the estimator, and $\tau=\min\{k,T_h\}$.

Step 3. Calculate $p_{\text{best}}(t)$ to be the best $\hat{p}$ *obtained so far*,

$$\hat{p}_{\text{best}}(k)=\arg\min_{s=0,1,\ldots,k}\{\mathcal{Y}(\hat{p}(s))\}. \tag{31}$$

Step 4. Generate N perturbed candidates (*random perturbations*) of $\hat{P}_{\text{best}}(k)$:$\hat{p}=\hat{T}(\hat{P}(k))$

$$\hat{p}^{(i)}_{\text{cand}}=(1-\alpha(k))\hat{p}_{\text{best}}(k)+\alpha(k)\Delta\hat{p}^{(i)},\ i=1,2,\ldots,N,$$

where $\Delta\hat{p}^{(i)}$ are random generated vectors satisfying the conditions $\sum_i\hat{p}_i=1,\ \hat{p}_i\in(-1,+1]$.

Step 5. The controller to be applied at the next simulation period is

$$\hat{p}(k+1)=\arg\min_{\hat{p}^{(i)}_{\text{cand}}}\{\hat{\mathcal{Y}}(\hat{p}^{(i)}_{\text{cand}})\}. \tag{32}$$

Reset $\hat{P}(k+1)=\hat{Z}(\hat{p}(k+1))$, where $\hat{Z}(\hat{p})$ is a function that provides an approximation of the transformation of the relationship

$$\hat{p}'\chi(x)\approx\Psi^*(x)\approx e^{-\frac{1}{\lambda}z'(x)\hat{P}z(x)}. \tag{33}$$

Step 6. Set $k=k+1$ and GO TO **Step 1**.

beginning of the algorithm, which may result in computational problems in simulating the closed-loop system. Furthermore, the noise may severely slow down the convergence of PCAO as it requires—as in the case of simulated annealing—that the magnitude of the random perturbations (controlled by the term $\alpha(t)$ in Table 2) decreases slowly with respect to time.

To overcome such a shortcoming, the nonconvex error function $\varepsilon(x(t),\hat{P})^2$ is replaced in the PCAO algorithm by a *transformed version that is convex with respect to* $\hat{P}$, as explained next.

TABLE 4 The switching PCAO algorithm.

Step 1. Run the algorithm of Table 2.
If the respective error function ε does not converge anymore, GO TO Step 2. Otherwise, CONTINUE.

Step 2. Switch to the algorithm of Table 3 and run it for a few iterations.
If the respective error function Y stops converging, STOP. Otherwise, GO TO Step 1.

CONVEX TRANSFORMATION

This section describes the modification of the above algorithm so as to overcome the problem of being trapped at a local minima. More precisely, define the exponential transformations

$$\Psi^*(x)=e^{-\frac{1}{\lambda}V^*(x)}\approx e^{-\frac{1}{\lambda}z'(x)\hat{P}z(x)}=\hat{\Psi}(x), \tag{23}$$

where λ is a positive constant, and a linear-in-the-parameters (LIP) approximator of $\Psi^*(x)$ as

$$\Psi^*(x)\approx\hat{\Psi}(x)=\hat{p}'\chi(x). \tag{24}$$

Since $V^*(x)$ is a positive definite function, Ψ^* and its estimate (denoted with $\hat{\Psi}(x)$) are constrained to be in the range $(0,1]$, and $\hat{\Psi}(x)=1\Rightarrow\hat{V}(x)=0$ only when $x=0$ (Lyapunov stability guarantee). This property is guaranteed by constraining the LIP approximator parameters to satisfy

$$\sum_i\chi_i(0)=1,\ \chi_i(x)\in(0,1],\quad\sum_i\hat{p}_i=1,\ \hat{p}_i\in(-1,+1]. \tag{25}$$

Equation (23) can be used to derive that

$$\hat{V}^*_x(x)=-\lambda\frac{\hat{\Psi}^*_x(x)}{\hat{\Psi}^*(x)}, \tag{26}$$

where $\hat{V}^*_x$ and $\hat{\Psi}^*_x$ denote the gradients of $\hat{V}^*$ and $\hat{\Psi}^*$, respectively, with respect to x. The estimated optimal control law

$$u=\lambda B'\frac{\hat{\Psi}^*(x)}{\hat{\Psi}^*_\chi(x)} \tag{27}$$

can be calculated from (26) and (11). Rearranging (23) gives

$$\hat{V}(x)=-\lambda\ln\hat{\Psi}(x). \tag{28}$$

The motivation for using the log transformation of the optimal cost-to-go function and its estimate is because this transformation allows the original nonconvex program to

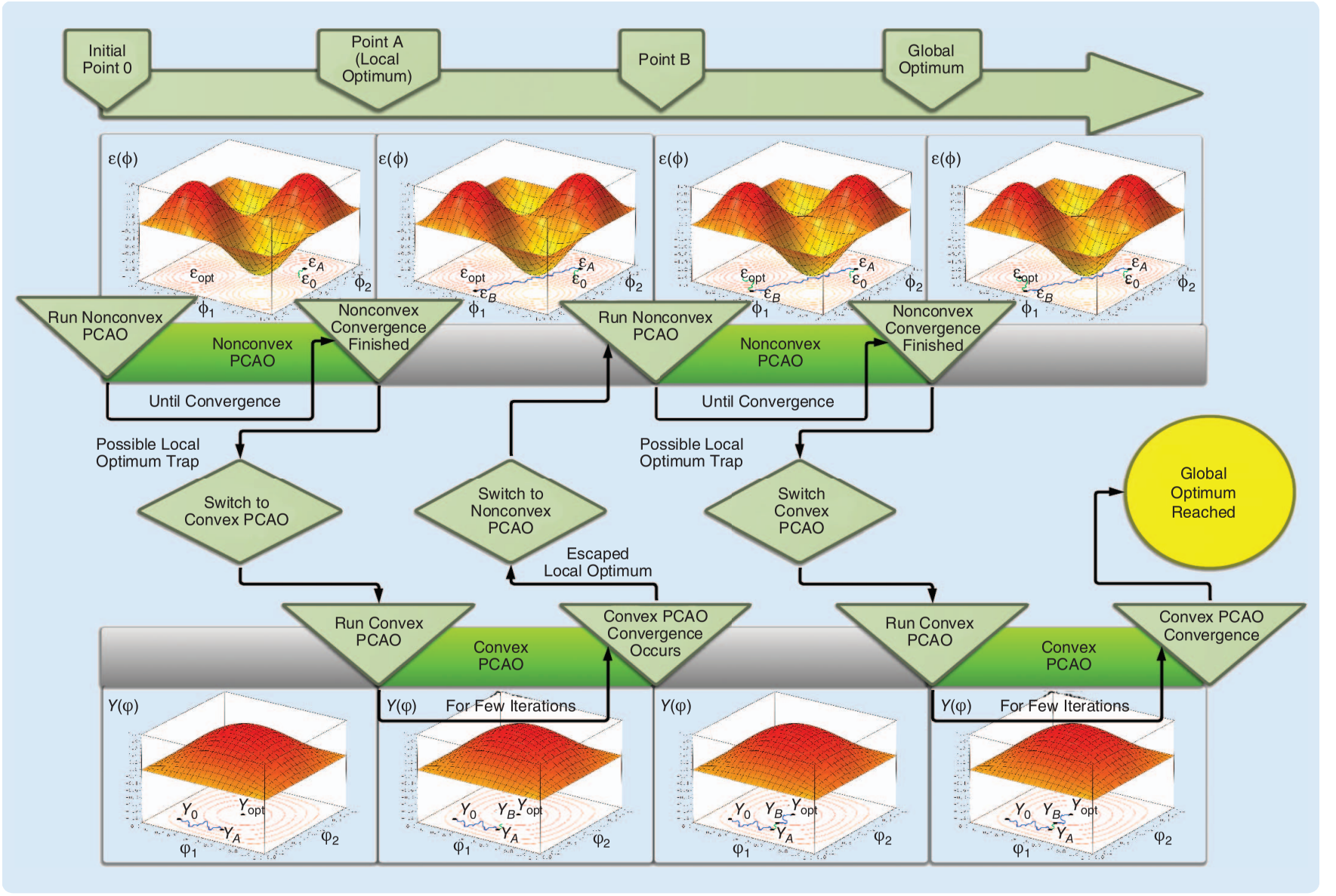

FIGURE 2 A toy example. The PCAO is called to maximize a two-dimensional performance function with a local and a global optimum. The initial point is close to the local optimum. The nonconvex PCAO runs to convergence (Point A). Then the convex PCAO is activated to check if the optimum is local and eventually escapes from the local optimum (Point B). Then the nonconvex CAO is activated again. When the nonconvex CAO convergences again, a new local optimum check is performed. If the optimum is global is optimal this time, then the convex PCAO converges and the algorithm stops. In general, in the presence of many local optima, many switches might be required.

be transformed into a convex program. More precisely, the log transformation of the optimal cost-to-go function linearizes the original nonlinear HJB equation [17] (see also [18]) for nonlinear systems of the form (4) and that are affected by zero-mean noise. The use of the random perturbations in the PCAO algorithm satisfies the aforementioned noise requirements and so linearization of the HJB is possible. More precisely, the work of [17] can be used to establish Theorem 2.

TABLE 5 Optimization algorithm tests.

Building Test Case		
Optimization Algorithm	**Convergence Iterations**	**Performance Improvement (%)**
PCAO	≈100	30–40
CAO	≈400	30–40
patternsearch	>6000	0
fmincon (derivative free)	>6000	0
Traffic Test Case		
Optimization Algorithm	**Convergence Iterations**	**Performance Improvement (%)**
PCAO	≈70	10–30
CAO	≈350	10–30
ga (genetic algorithm)	>4000	0
fmincon (derivative free)	>6000	0

Theorem 2

The *transformed error criterion*

$$Y(\hat{p}, x(t+\delta t), x(t)) = \left(\int_t^{t+\delta t} \left(\frac{\mathrm{d}\hat{V}(x)}{\mathrm{dt}} + \Pi(x) \right) \hat{\Psi}(x)\, \mathrm{ds} \right)^2 \quad (29)$$

is convex with respect to $\hat{p}$ vector.

Proof

See "Proof of Theorem 2." ■

Using a similar CAO-based modified algorithm as in the nonconvex case, the algorithm in Table 3 is introduced for minimizing (29).

One practical difficulty in the error function Y calculation process is the requirement of differentiating $\hat{V}$. To bypass this drawback, the error function is approximated as

$$Y(\hat{p}, x(t+\delta t), x(t)) \approx \left(\int_t^{t+\delta t} (a(s)\hat{V}(x(s)) + b(s)\Pi(x(s)))\hat{\Psi}(x(s))\, \mathrm{ds} \right)^2, \quad (30)$$

where $a(s) = 1/(s+c)$, $b(s) = s/(s+c)$, and c is a positive design parameter with value close to one.

SWITCHING ALGORITHM

The algorithm of Table 3, while guaranteeing convergence to the global optimum, possesses the disadvantage—due to transforming the original nonconvex problem into a convex problem—that the nonconvex error function takes unacceptably large values. Practically speaking, the large values lead to

Proof of Theorem 2

A brief outline of the proof that the error function Y is convex with respect to $\hat{p}$ is provided. Due to the use of random candidate perturbations, the overall system dynamics can be written as

$$\dot{x} = f(x) - Bu + \lambda B\xi,$$

where ξ is a standard Gaussian white noise vector. This equation and Ito calculus implies that

$$\frac{\mathrm{d}\hat{V}}{\mathrm{dt}} \approx \hat{V}_x'(f - BB'\hat{V}_x) + \frac{1}{2}\mathrm{tr}(\hat{V}_{xx}\Sigma), \quad (S6)$$

$$-\Pi \approx V_x'(f - BB'V_x) + \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma), \quad (S7)$$

where $\Sigma = BB'$, V_x, and V_{xx} denote the gradient and Hessian of V with respect to x, and tr refers to the trace of a matrix.

Using (S6), (S7), and the definitions of the functions Ψ and $\hat{\Psi}$,

$$0 \approx -\frac{1}{\lambda}\Pi\Psi + \Psi_x' f + \frac{1}{2}\mathrm{tr}(\Psi_{xx}\Sigma), \quad (S8)$$

$$0 \approx \frac{1}{\lambda}\frac{\mathrm{d}\hat{V}}{\mathrm{dt}}\hat{\Psi} + \hat{\Psi}_x' \; f + \frac{1}{2}\mathrm{tr}(\hat{\Psi}_{xx}\Sigma). \quad (S9)$$

By subtracting (S9) from (S8),

$$0 \approx -\frac{1}{\lambda}\Pi\Psi - \frac{1}{\lambda}\frac{\mathrm{d}\hat{V}}{\mathrm{dt}}\hat{\Psi} + \tilde{\Psi}_x' f + \frac{1}{2}\mathrm{tr}(\tilde{\Psi}_{xx}\Sigma),$$

$$0 \approx -\frac{1}{\lambda}\Pi\Psi + \left(\frac{1}{\lambda}\Pi\hat{\Psi} - \frac{1}{\lambda}\Pi\hat{\Psi}\right) - \frac{1}{\lambda}\frac{\mathrm{d}\hat{V}}{\mathrm{dt}}\hat{\Psi} + \tilde{\Psi}_x' f + \frac{1}{2}\mathrm{tr}(\tilde{\Psi}_{xx}\Sigma),$$

$$0 \approx -\frac{1}{\lambda}\Pi\tilde{\Psi} - \frac{1}{\lambda}\left(\Pi + \frac{\mathrm{d}\hat{V}}{\mathrm{dt}}\right)\hat{\Psi} + \tilde{\Psi}_x' f + \frac{1}{2}\mathrm{tr}(\tilde{\Psi}_{xx}\Sigma),$$

$$\frac{1}{\lambda}\left(\Pi + \frac{\mathrm{d}\hat{V}}{\mathrm{dt}}\right)\hat{\Psi} \approx -\frac{1}{\lambda}\Pi\tilde{\Psi} + \tilde{\Psi}_x' f + \frac{1}{2}\mathrm{tr}(\tilde{\Psi}_{xx}\Sigma),$$

where $\tilde{\Psi} = \Psi - \hat{\Psi}$, $\tilde{\Psi}_x = \Psi_x - \hat{\Psi}_x$, and $\tilde{\Psi}_x = \Psi_{xx} - \hat{\Psi}_{xx}$. Integrating the above equation with respect to time and using the fact that $\hat{\Psi}(x) = \hat{p}'\chi(x)$ shows that Y is quadratic (and thus convex) with respect to $\hat{p}$.

FIGURE 3 A view of the Chania building.

TABLE 6 Optimization problem dimension.

Building Test Case			
Exogenous Variables	**States**	**Inputs**	**Optimization Variables**
Outdoor temperature Outdoor humidity Outdoor solar radiation	10 office temperatures 10 office humidities	10 AC setpoints	$\frac{46\times(46+1)}{2}L = 1081\times L$
Traffic Test Case			
Exogenous Variables	**States**	**Inputs**	**Optimization Variables**
Demands at the 22 network origins	61 occupancies 61 flows	42 green times	$26106\times L$

slow convergence of the convex algorithm or, sometimes, lead to unbounded signal behavior. This situation can be avoided by using a combined algorithm that switches between the nonconvex and the convex problems. The idea behind this switching scheme is simple. The nonconvex algorithm, which does not face slow convergence problems, is applied. When the nonconvex algorithm reaches a minimum, the convex algorithm is activated to check whether the minimum is the global minimum and, if not, to help escape from the local minimum. Table 4 provides the details of such an algorithm. Additionally, Figure 2 shows a graphical representation of the method and describes the switching between the nonconvex and convex implementation so as to escape from local optima and reach the global optimal solution.

Theorem 3 illustrates the convergence and stability properties of the switching PCAO algorithm of Table 4. The proof is straightforward and is omitted.

Theorem 3

The switching PCAO algorithm described in Table 4 guarantees that $\hat{P}$ converges almost surely (that is, with probability one) to the set

$$\mathcal{D}_{\text{global optimal}} = \left\{\hat{P}: \hat{P} \text{ is positive definite and is the global minimum of } \mathcal{F}(\hat{P})\right\}. \quad (34)$$

Some remarks regarding implementation issues conclude this section.

Remark 4

For the sake of simplicity, the exposition of the PCAO approach assumes the availability of full state measurements. From a practical point of view, it is interesting to see how the method can be implemented in the case of partial knowledge of the state, as is the case in the majority of simulation-based control designs. In such a case, the HJB equations (10)–(11) can be reformulated as

$$\left(\frac{\partial V^*}{\partial y}\frac{\partial y}{\partial x}\right)'\left(f(x) - BB'\frac{\partial V^*}{\partial y}\right) = -\Pi(y), \quad (35)$$

where V is a function of the measurements y. Extensions can be also made in case of dynamic output feedback, in which case V is a function of y plus the state of a dynamics regulator, for example $\xi = \Lambda(s)y$, where ξ is the state vector of the regulator and $\Lambda(s)$ a Hurwitz filter.

Remark 5

Due to the heterogeneous nature of the simulation models, in general, it is very difficult, if at all possible, to find a priori lower bounds on the number of mixing functions that are necessary to achieve a desired performance. Lower bounds can be found in the case that the model to be controlled can be expressed in a state-space form (see [35]); however, assuming a state-space formulation limits the purpose of the simulation-based approach, whose more general applicability has been verified in different environments with different simulation models, as shown in the section "Applications and Results." Two observations follow.

This article describes a computationally efficient simulation-based control design approach that has the capability of handling optimization problems arising from large-scale nonlinear systems.

1) Given a particular number of switching elements, the designer can verify during the optimization process and finally a posteriori if the performance is satisfactorily close to the desired performance, by testing the controller, at every time step, for every possible operating point of interest. Eventually the complexity of the control policy can be increased if any of the performance tests will fail.
2) In many practical applications, a small number of linear controllers is known to cover all the system operating points, while providing satisfactory performance. This case occurs, for example, for the energy-efficient building control and the urban traffic control systems considered in the "Applications and Results" section but also for many other real-life applications from aircraft applications to smart grids and industrial plants (see [14], [15], [36], and [37]).

Remark 6

In simulation-based optimization, a common difficulty comes from the time-consuming performance evaluation. In fact, whenever a stochastic simulation model is considered, estimating the derivative and even the performance itself becomes an issue since many replications of an experiment might be required. When simulation becomes time consuming, the described approach can offer some advantages by avoiding the numerical estimation of the derivative of the cost function.

Applications and Results

The PCAO algorithm has been evaluated using two LSS, in the areas of energy-efficient control of large buildings and urban traffic control. Table 5 shows the improvement of the CAO (as described in [16]) and of the PCAO strategy with respect to control strategies currently implemented in the two systems. A comparison with respect to different derivative-free optimization algorithms implemented in the Matlab Optimization Toolbox has been performed. Due to the large-scale nature of the systems, while these derivative-free optimization algorithms do not show any improvement and do not manage to converge to any minima even after thousands of iterations, the CAO approach achieves relevant improvements. Besides, with respect to CAO, PCAO achieves similar improvements, while converging from four to five times faster. The next sections present the control design and implementation details for the two systems.

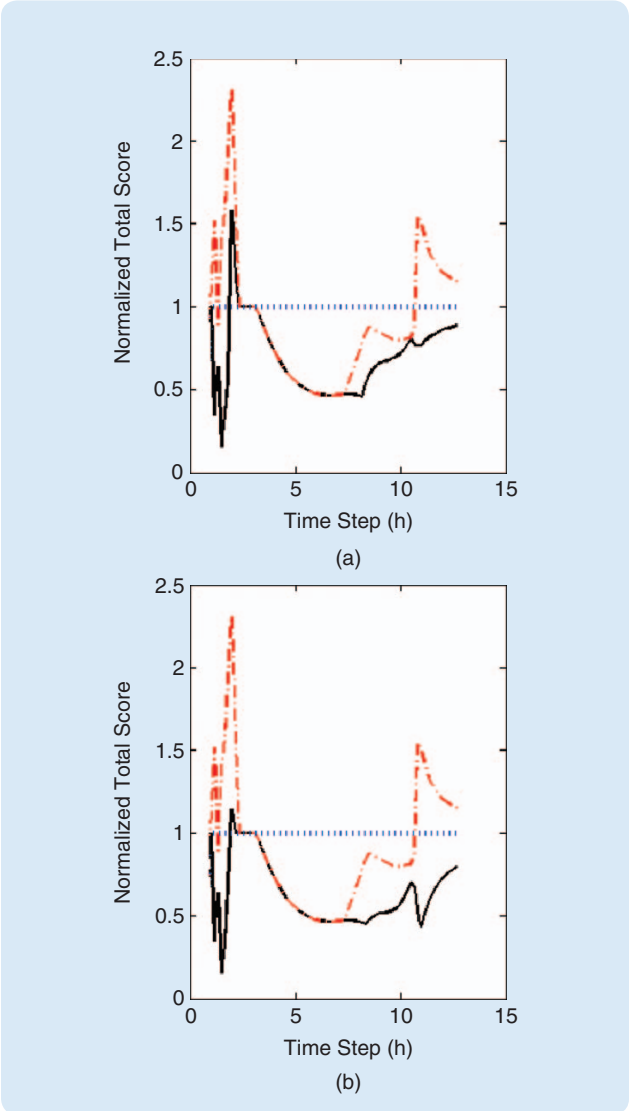

FIGURE 4 Total scores for RBC1 (red dash-dotted), RBC2 (blue dotted), and PCAO (black solid) for one day (during office hours). The total scores are normalized with respect to RBC2. Apart from some intervals of time during the day, the PCAO total score is smaller than the total scores of both rule-based controllers (RBCs). Adding more linear controllers improves the total score early in the morning and in the last hours of the afternoon. (a) Control scenario $L = 1$ and (b) control scenario $L = 4$.

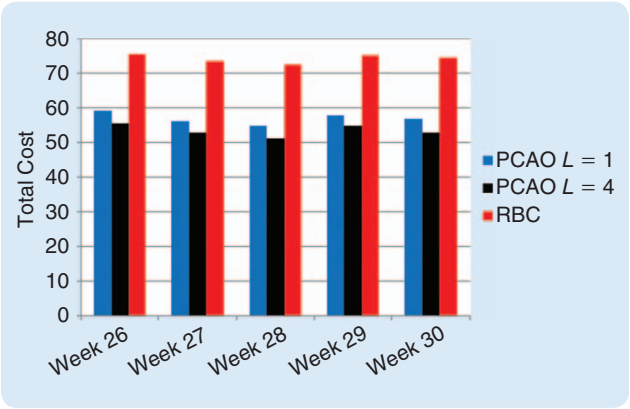

FIGURE 5 The total cost achieved during a five-week experiment (using historical data from July to August 2011), showing PCAO ($L = 1$, blue), PCAO ($L = 4$, black), and RBC2 (red).

CONTROL OF A POSITIVE ENERGY BUILDING

As a first example, a building with ten offices located at the Technical University of Crete, Chania, Greece, was considered (see Figure 3). An EnergyPlus model [38] of the building simulates the behavior of the heating/cooling system, the natural ventilation process, and the air flow dynamics. The building, equipped with a photovoltaic array, uses the photovoltaic energy (when available) to fulfill its needs. If the photovoltaic energy is less than is needed, then the building uses (nonrenewable) energy from the electrical grid. The challenge of the control design is to minimize the energy from the electrical grid by exploiting renewable energy to the maximum possible extent. The presented simulations consider the problem of operating air conditioners during summer to cool the rooms in an energy-efficient manner while keeping the users' thermal comfort at a satisfactory level.

The optimization problem is formulated in terms of the minimization of a combined performance index that takes into account the energy consumption score (E_{score}) and the user comfort score (C_{score}),

$$\min J = \int_0^T \text{Total}_{\text{score}}(t)\,\text{dt} = \int_0^T \left(E_{\text{score}}(t) + 0.1\,C_{\text{score}}(t)\right)\text{dt}, \tag{36}$$

where T is the simulation horizon and the weight of 0.1 was chosen as a scaling factor between the two cost terms. The term E_{score} is the energy consumed from the electrical grid, while C_{score} is evaluated through the Fanger index, giving the predicted percent of dissatisfied users under certain thermal conditions [39]. For sufficiently long simulation horizons ($T = 5$ weeks in these simulations), the cost (36), normalized by T, acts as an expectation operator, thus optimizing a mean performance with respect to the random external weather conditions, taking into account different realizations of the random variables.

Two rule-based controllers (RBCs) in the real building are used for comparison purposes. The RBCs employ the simple control strategy of keeping the heating, ventilation, and air conditioning (HVAC) setpoints of each room constant to 24 °C (RBC1) or to 25 °C (RBC2) during office hours. The states, input variables, exogenous weather variables, and the dimension of the optimization problem are shown in Table 6. The PCAO algorithm was run both with $L = 1$ and $L = 4$. The results, showing improvements

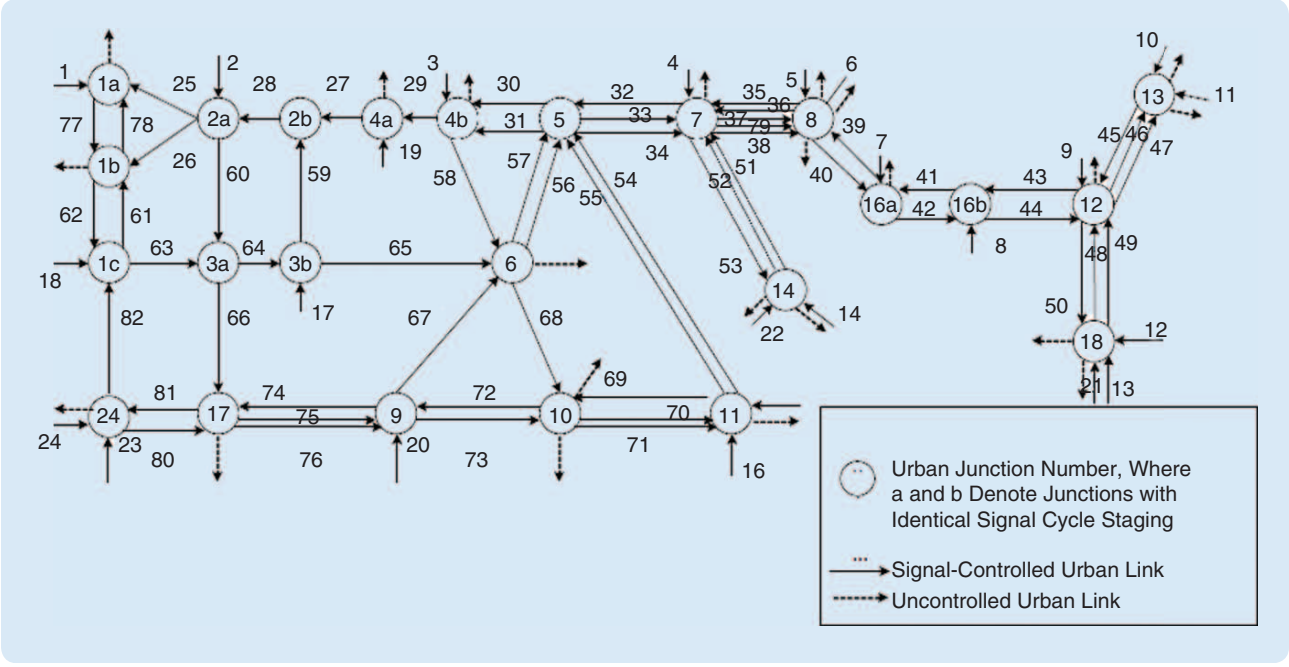

FIGURE 6 A schematic map of the Chania traffic network showing junction and link numbers.

in the total cost with respect to both RBCs, are shown in Figure 4 for a single day and in Figure 5 for a five-week experiment (using historical weather data from week #26 to week #30 of 2011).

URBAN TRAFFIC CONTROL

Figure 6 displays the Chania traffic city network. The network includes the city's main shopping district and faces serious congestion, especially in summer, during the morning (8–9 a.m.), afternoon (2–3 p.m.), and evening (9–10 p.m.) peak hours, with link queues that may spill back into upstream links. An AIMSUN-based simulation model [40] was developed for the network. The overall network comprises 16 controlled junctions with a total of 42 control inputs (green times of the junction phases) and 122 sensor measurements (61 loop detectors in the network that provide occupancy and flow measurements). The states, input variables, exogenous variables, and the dimension of the optimization problem are shown in Table 6.

The traffic-responsive urban control (TUC) strategy [41], a linear-quadratic-based control strategy actually implemented in the Chania network, was considered for comparison purposes. The cost criterion is to maximize the mean speed (MS) over the whole traffic network,

$$\max J = \int_0^T MS(t)\,\mathrm{d}t, \tag{37}$$

where $T = 60$ days is a sufficiently long control horizon to account for the different realizations of the random demand scenarios at the 22 network origins. Figure 7 depicts the improved behavior of the PCAO control system with respect to the TUC strategy. Figure 8 shows the evolution of the MS for a 30-day simulation using historical traffic demand data from June 2008.

CONCLUSIONS

An algorithm for simulation-based control design for LSS, referred to as PCAO, has been established, analyzed, and evaluated using two real applications. The algorithm minimizes a performance index related to how close the applied control action satisfies the HJB equation associated with 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 HJB equation, make the methodology capable of controlling LSS. Two large-scale numerical examples have been used to demonstrate the effectiveness of the method and its capability of handling efficiently large-scale control design optimization problems.

ACKNOWLEDGMENTS

The research leading to these results has been partially funded by the European Commission FP7-ICT-5-3.5, Engineering of

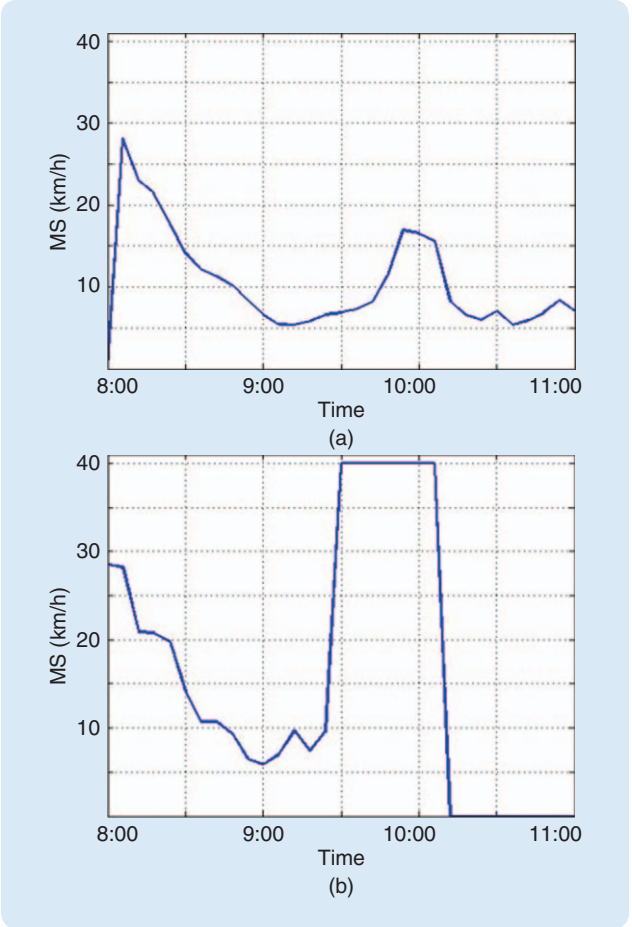

FIGURE 7 Traffic network mean speeds (MSs) under different strategies. The horizontal axis spans a 3-h period corresponding to the morning network peak. The PCAO controller clears the congestion (at around 9:30, the MS increases to 40 km/h, the legal urban speed limit constraint, and then drops to zero when the queue has disappeared). On the other hand, the traffic-responsive urban control (TUC) strategy is unable to clear the congestion at around 9:30, which results in a second congestion after 10:00 (low MS). (a) TUC controller and (b) PCAO controller.

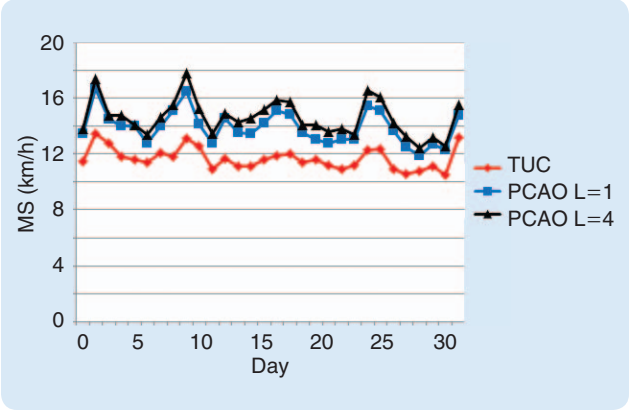

FIGURE 8 The mean speed achieved during a 30-day experiment using historical data from July 2008: traffic-responsive urban control (TUC) (red), PCAO ($L = 1$, blue) and PCAO ($L = 4$, black).

The traffic-responsive urban control strategy, a linear-quadratic-based control strategy actually implemented in the Chania network, was considered for comparison purposes.

Networked Monitoring and Control Systems, under the contract 257806 (AGILE).

AUTHOR INFORMATION

Simone Baldi (s.baldi@tudelft.nl) received the B.Sc. in electrical engineering, the M.Sc. in automatic systems control engineering, and the Ph.D. in automatic systems control and computer science engineering from the Università degli Studi di Firenze (University of Florence), in 2005, 2007, and 2011, respectively. He is currently an assistant professor with the Delft Center for Systems and Control at the Delft University of Technology, The Netherlands. Previously, he held postdoc research positions at the Information Technologies Institute, Centre for Research and Technology, Hellas, Greece, and at the Computer Science Department, University of Cyprus. His research interests are in the area of control theory and include adaptive control, switching supervisory control, and approximately optimal control of nonlinear large-scale systems, with applications in energy-efficient building control and intelligent transportation systems. He can be contacted at the Delft University of Technology, Mekelweg 2 (3mE building), 2628 CD, Delft, The Netherlands, Tel.: +31 015 2781823.

Iakovos Michailidis graduated from the Department of Electrical and Computer Engineering of the Polytechnic School of Aristotle University of Thessaloniki in 2010. He is a Ph.D. candidate at the Department of Electrical and Computer Engineering of the Polytechnic School at Democritus University of Thrace, Xanthi, Greece. He has worked on selectivity of protective devices and short-circuit analysis on energy distribution networks. He is currently conducting research in the area of model-based and model-free optimal control techniques for large-scale systems, with applications in efficient building control and traffic networks.

Elias B. Kosmatopoulos received the diploma, M.Sc., and Ph.D. from the Technical University of Crete, Greece, in 1990, 1992, and 1995, respectively. He is currently an associate professor with the Department of Electrical and Computer Engineering at the Democritus University of Thrace, Xanthi, Greece. Previously, he was a faculty member with the Department of Production Engineering and Management at the Technical University of Crete, a research assistant professor with the Department of Electrical Engineering Systems at the University of Southern California, and a postdoctoral fellow with the Department of Electrical and Computer Engineering at the University of Victoria, British Columbia, Canada. His research interests are in the areas of neural networks, adaptive optimization and control, energy-efficient buildings, smart grids, and intelligent transportation systems. He is the author of over 40 journal papers. He is currently leading many research projects funded by the European Union, with a total budget of about €10 million.

Petros A. Ioannou received the B.Sc. with first-class honors from University College London, England, in 1978 and the M.S. and Ph.D. from the University of Illinois, Urbana-Champaign, in 1980 and 1982, respectively. In 1982, he joined the Department of Electrical Engineering Systems at the University of Southern California, Los Angeles. He is currently a professor in the same department and the director of the Center of Advanced Transportation Technologies. He also holds courtesy appointments with the Departments of Aerospace and Mechanical Engineering and Industrial and Systems Engineering. He is the associate director for research for the University Transportation Center METRANS at the University of Southern California. He was visiting professor at the University of Newcastle, Australia, and the Australian National University in Canberra, the Technical University of Crete, and served as the dean of the School of Pure and Applied Science at the University of Cyprus in 1995. In 2009, he was with the Department of Electrical Engineering and Information Technologies, Cyprus University of Technology, while on sabbatical leave from the University of Southern California. He received the Outstanding Transactions Paper Award from the IEEE Control Systems Society in 1984 and a Presidential Young Investigator Award in 1985 for his research in adaptive control. In 2009 he received the IEEE ITSS Outstanding ITS Application Award and the 2009 IET Heaviside Medal for Achievement in Control by the Institution of Engineering and Technology (formerly IEE). In 2012, he received the IEEE ITSS Outstanding ITS Research Award. He has been an associate editor for *IEEE Transactions on Automatic Control*, *International Journal of Control*, *Automatica*, and *IEEE Transactions on Intelligent Transportation Systems*. He is a member of the Board of Governors of the IEEE Intelligent Transportation Society. He is a Fellow of IEEE, the International Federation of Automatic Control, and the Institution of Engineering and Technology and the author

or coauthor of eight books and over 200 research papers in the area of control, vehicle dynamics, neural networks, nonlinear dynamical systems, and intelligent transportation systems.

REFERENCES

[1] S. Andradóttir, "Simulation optimization," in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, J. Banks, Ed. New York: Wiley, 1998, pp. 307–333.

[2] M. C. Fu, "Optimization for simulation: Theory vs. practice," *INFORMS J. Comput.*, vol. 14, no. 3, pp. 192–215, 2002.

[3] J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben, "A survey of recent advances in discrete input parameter discrete-event simulation optimization," *IIE Trans.*, vol. 36, no. 6, pp. 591–600, 2004.

[4] E. Tekin and I. Sabuncuoglu, "Simulation optimization: A comprehensive review on theory and applications," *IIE Trans.*, vol. 36, no. 11, pp. 1067–1081, 2004.

[5] M. C. Fu, C.-H. Chen, and L. Shi, "Some topics for simulation optimization," in *Proc. Winter Simulation Conf.*, 2008, pp. 27–38.

[6] A. T. Al-Hammouri, M. S. Branicky, and V. Liberatore, "Co-simulation tools for networked control systems," in *Hybrid Systems: Computation and Control*, M. Egerstedt and B. Mishra, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 16–29.

[7] M. Trcka, J. L. M. Hensen, and M. Wetter, "Co-simulation of innovative integrated HVAC systems in buildings," *J. Building Perform. Simulation*, vol. 2, no. 3, pp. 209–230, 2009.

[8] Z. Yu and A. Dexter, "Simulation based predictive control of low energy building systems using two-stage optimization," in *Proc. IBPSA*, 2009, pp. 1562–1568.

[9] T. Nghiem and G. H. Pappas, "Receding-horizon supervisory control of green buildings," in *Proc. American Control Conf.*, San Francisco, CA, 2011, pp. 4416–4421.

[10] D. Quaglia, R. Muradore, R. Bragantini, and P. Fiorini, "A systemC/matlab co-simulation tool for networked control systems," *Simulation Modelling Pract. Theory*, vol. 23, pp. 71–86, Apr. 2012.

[11] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[12] E. B. Kosmatopoulos, "CLF-based control design for unknown multi-input nonlinear systems with good transient performance," *IEEE Trans. Autom. Contr.*, vol. 55, no. 11, pp. 2635–2640, 2010.

[13] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos, "Multi-robot 3D coverage of unknown areas," in *Proc. Int. J. Robotics Research*, pp. 738–752, vol. 31, 2012.

[14] L. Doitsidis, S. Weiss, A. Renzaglia, M. W. Achtelik, E. B. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboad vision," *Auton. Robots*, vol. 33, no. 12, pp. 173–188, 2012.

[15] M. Pichler, "Simulation-assisted building energy performance improvement using sensible control decisions," in *Proc. 3rd ACM Workshop Embedded Sensing Systems Energy-Efficiency Buildings (BuildSys '11)*, Seattle, WA, 2011, pp. 61–66.

[16] E. B. Kosmatopoulos and A. Kouvelas, "Large-scale nonlinear control system fine-tuning through learning," *IEEE Trans. Neural Networks*, vol. 20, no. 6, pp. 1009–1023, 2009.

[17] W. Fleming, "Exit probabilities and optimal stochastic control," *Appl. Math. Optim.*, vol. 4, no. 4, pp. 329–346, 1978.

[18] H. J. Kappen, "Linear theory for control of nonlinear stochastic systems," *Phys. Rev. Lett.*, vol. 95, pp. 200201–200203, Nov. 2005.

[19] T. Basar and P. Bernhard, *H∞-Optimal Control and Related Minimax Design Problems*. Boston, MA: Birkhauser, 1995.

[20] W. M. Lu and J. C. Doyle, "H∞ control of nonlinear systems: A convex characterization," *IEEE Trans. Automat. Contr.*, vol. 40, no. 9, pp. 1668–1675, 1995.

[21] M. R. James and J. S. Baras, "Partial observed differential games, infinite-dimensional Hamilton-Jacobi-Isaac equations, and nonlinear H∞ control," *SIAM J. Control Optim.*, vol. 34, no. 4, pp. 1342–1364, 1996.

[22] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[23] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini, "A stabilizing model-based predictive control for nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

[24] M. L. Kaplan and J. H. Heegaard, "Second-order optimal control algorithm for complex systems," *Int. J. Numer. Methods Eng.*, vol. 53, no. 9, pp. 2043–2056, 2002.

[25] O. Strik, "Numerical solution of optimal control problems by direct collocation," in *Optimal Control—Calculus of Variations, Optimal Control Theory and Numerical Methods*, R. Burlirsch, A. Miele, J. Stoer, and K. H. Well, Eds. Basel, Switzerland: Birkhauser, 1993, pp. 129–143.

[26] Q. Sun, R. Li, and P. Zhang, "Stable and optimal adaptive fuzzy control of complex systems using fuzzy dynamic model," *Fuzzy Sets Syst.*, vol. 133, no. 1, pp. 1–17, 2003.

[27] C. Masmoudi, N. K. Rekik, M. Djemel, and N. Derbel, "Optimal control for discrete large scale nonlinear systems using hierarchical fuzzy systems," in *Proc. Second Int. Conf. Machine Learning Computing*, Bangalore, Karnataka, India, 2010, pp. 91–95.

[28] 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," in *Proc. 50th IEEE Conf. Decision Control European Control*, Orlando, FL, 2011, pp. 5407–5412.

[29] A. Rantzer and M. Johansson, "Piecewise linear quadratic optimal control," *IEEE Trans. Automat. Contr.*, vol. 45, no. 4, pp. 629–637, 2000.

[30] L. Rodrigues, "Stability analysis of piecewise-affine systems using controlled invariant sets," *Syst. Control Lett.*, vol. 53, no. 2, pp. 157–169, 2004.

[31] M. Johansson and A. Rantzer, "Computation of piecewise quadratic lyapunov functions for hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 555–559, 1998.

[32] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Menlo Park, CA: Addison Wesley Longman, 1998.

[33] S. Prajna, A. Papachristodoulou, and F. Wu, "Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach," in *Proc. Asian Control Conf.*, 2004, pp. 157–165.

[34] S. Baldi, I. Michailidis, E. B. Kosmatopoulos, A. Papachristodoulou, and P. A. Ioannou, "Convex design control for practical nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 59, no. 7, pp. 1692–1705, 2014.

[35] E. B. Kosmatopoulos, "An adaptive optimization scheme with satisfactory transient performance," *Automatica*, vol. 45, no. 3, pp. 716–723, 2009.

[36] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos, "Cognitive-based adaptive control for cooperative multi-robot coverage," in *Proc. IEEE Int. Conf. Robotics Intelligent System*, Taipei, Taiwan, 2010, pp. 3314–3320.

[37] D. Kolokotsa, D. Rovas, E. B. Kosmatopoulos, and K. Kalaitzakis, "A roadmap towards intelligent net zero- and positive-energy buildings," *Sol. Energy*, vol. 85, no. 12, pp. 3067–3084, 2012.

[38] 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 Buildings*, vol. 33, no. 4, pp. 319–331, 2001.

[39] *Thermal Environmental Conditions for Human Occupancy*, ANSI/ASHRAE Standard 55–2004.

[40] J. Barcelo, J. Casas, and J. L. Ferrer, "Analysis of dynamic guidance systems by microsimulation with aimsun2," in *Proc. 6th World Conf. ITS*, Toronto, ON, Canada, 1999.

[41] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proc. IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[42] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[43] H. Robbins and D. Siegmund, "A convergence theorem for nonnegative almost supermartingales and some applications," in *Optimizing Methods in Statistics*, J. S. Rustari, Ed. New York: Academic Press, 1971, pp. 233–257.