Active Model Discrimination with Applications to Fraud Detection in Smart Buildings [⋆]

Farshad Harirchi [∗,⋆⋆] **Sze Zheng Yong** [∗∗,⋆⋆]
Emil Jacobsen [∗∗∗,⋆⋆] **Necmiye Ozay** [∗]

[∗] *EECS Department, University of Michigan, Ann Arbor, MI 48109, USA (e-mail:* {harirchi,necmiye}@umich.edu).
[∗∗] *School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, USA (e-mail:* szyong@asu.edu).
[∗∗∗] *Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail:* emiljaco@kth.se).

Abstract: In this paper, we consider the problem of active model discrimination amongst a finite number of affine models with uncontrolled and noise inputs, each representing a different system operating mode that corresponds to a fault type or an attack strategy, or to an unobserved intent of another robot, etc. The active model discrimination problem aims to find optimal separating inputs that guarantee that the outputs of all the affine models cannot be identical over a finite horizon. This will enable a system operator to detect and uniquely identify potential faults or attacks, despite the presence of process and measurement noise. Since the resulting model discrimination problem is a nonlinear non-convex mixed-integer program, we propose to solve this in a computationally tractable manner, albeit only approximately, by proposing a sequence of restrictions that guarantee that the obtained input is separating. Finally, we apply our approach to attack detection in the area of cyber-physical systems security.

Keywords: Input design; Model discrimination; Fault and attack detection; Smart building.

1. INTRODUCTION

The public interest in the integration of smart systems into everyday lives is on the rise. These systems, also known as cyber-physical systems (CPS) and include smart homes, smart grids, intelligent transportation and smart cities, are essentially engineered systems that consist of interacting networks of physical and computational components. However, as more and more components and functionalities become integrated, there is an increased risk of unintended system faults and also intentional attacks. Recent attack incidents, e.g., the Maroochy water breach, the StuxNet computer worm and industrial security incidents [Cárdenas et al. (2008); Farwell and Rohozinski (2011)], highlight a need to address the security of CPS.

Model discrimination as a tool to detect faults, attacks or more broadly, the system mode of operation, is an important common problem in statistics, machine learning and systems theory. Hence, algorithms developed for this problem can have a significant impact on a wide array of problems in CPS security, robotics, process control, medical devices, fault detection, etc.

Literature Review. The problem of discriminating among a set of models arises in a wide variety of applications such as fault or attack detection and isolation, mode estimation in hybrid systems and intention estimation in human-robot interactions. The model discrimination approaches can be broadly categorized as passive and active methods. In passive methods, the aim is to guarantee discrimination regardless of the input applied to the system, while active methods seek an input that guarantees distinction among models. The former provides "stronger" guarantees but is limited to problems with specific system properties, while the latter is "weaker" but is more widely applicable.

The control and hybrid systems community are predominantly interested in finding system properties that lead to model discrimination. In the spirit of active methods, Grewal and Glover (1976) and Babaali and Egerstedt (2004) introduced the system properties of distinguishability and controlled-discernibility, respectively as the existence of an input that enables distinguishing between the generated trajectories for a given time horizon for all admissible initial conditions. As the passive counterpart, Lou and Si (2009); Rosa and Silvestre (2011) introduced the concept of input-distinguishability, i.e., the ability to distinguish between the behaviors of two linear models for a given time horizon regardless of the inputs applied to the models.

On the other hand, the fault detection and isolation community is more focused on the development of computationally tractable model discrimination algorithms. In Harirchi and Ozay (2015, 2016); Harirchi et al. (2016), a computational method for passively discriminating among

[⋆] This work is supported in part by an Early Career Faculty grant from NASA's Space Technology Research Grants Program and DARPA grant N66001-14-1-4045.

[⋆⋆] Equal contribution from these authors.

fault models is proposed, where a time horizon length that is sufficient for providing guarantees on discrimination of system and fault models is calculated (referred to as the T-detectability problem). Then, a model invalidation approach is proposed to detect and isolate different fault models in real-time with a receding horizon scheme without compromising detection/isolation guarantees.

A great deal of attention is also given to model-based active fault detection approaches. In Nikoukhah and Campbell (2006); Tabatabaeipour (2015); Scott et al. (2014); Raimondo et al. (2016), set-membership approaches are proposed to isolate multiple linear time-varying models subject to uncertainties and noise. However, these approaches either suffer from expensive computation or numerical instability when uncertainties are directly described by polytopic constraints, or they sacrifice optimality by approximating these polytopes with zonotopes for computational tractability. In contrast, our recent work in Jacobsen et al. (2017), in the context of intention estimation in autonomous vehicles, considers an active method for discriminating among a set of affine models with uncontrolled inputs using a Mixed-Integer Linear Program with Special Ordered Sets constraints. This approach can directly handle polyhedral uncertainties and can also consider asymmetric responsibilities for state bounds satisfaction in multi-agent interactions. However, the effect of noise was neglected, and it was also assumed that the excitation input does not affect the state constraints.

Then again, the CPS security community is concerned with both finding system properties and algorithm development. The property of strong detectability/observability of each model has been found to be a necessary condition in Yong et al. (2015); Fawzi et al. (2014) for detection and isolation of data injection attacks, and passive attack detection algorithms have been proposed using SMT solvers in Shoukry et al. (2014), state observers in Pasqualetti et al. (2013), l_1-relaxations in Fawzi et al. (2014) and mode estimation filters in Yong et al. (2015). To our best knowledge, active attack detection via input design has not been explored. The "closest" existing methods use input watermarks [Mo et al. (2015)] and moving targets [Weerakkody and Sinopoli (2015)] to aid attack detection.

Contributions. In this paper, we present an active approach for discriminating among a set of affine models with uncontrolled inputs, which extends our previous work in Jacobsen et al. (2017). We additionally consider noise in the modeling framework (to be distinguished from uncontrolled inputs) and relaxed a relatively strong assumption that the designed input cannot affect the state constraints that the uncontrolled input is responsible for. This results in a nonlinear non-convex mixed-integer program. We propose to solve this in a computationally tractable manner, albeit only approximately, by proposing a sequence of restrictions. Instead of relaxations that result in the loss of feasibility, our solution is guaranteed to be feasible in the sense that the obtained input guarantees the discrimination among models, but the resulting objective value may be suboptimal. Moreover, to the extent of our knowledge, our work is the first to apply active model discrimination to attack detection problems in CPS, which we illustrate with a fraud detection scenario in smart buildings.

2. MOTIVATING EXAMPLE: FRAUD DETECTION

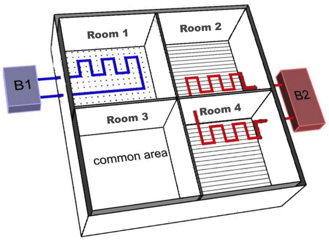


Fig. 1. Four zone radiant system with two heaters

As a motivation for the input design problem studied in this paper, we consider a potential utility bill fraud scenario in smart buildings. Specifically, we consider attacks on a building heat management system that intelligently regulates the indoor temperature by a feedback law based on weather forecast and sensor readings. An attack takes the form of sensor measurement manipulation by a tenant (via data injection attack) in order to achieve his/her own desired temperature range by taking advantage of the temperature regulation system of the building. Ideally, an attacker aims to remain undetected, hence paying less in utility bills than his/her actual energy consumption.

From our perspective as the building manager, this attack scenario can result in higher utility costs, thus there is an incentive for detecting potential sensor data injection attacks. At the same time, to make the case for intentional fraud, there is also a need to make sure that the change of temperature is not a result of a serendipitous fault.

For instance, consider a building floor with two apartments and a common area (Room 3), where the left apartment consists of Room 1 and the right apartment consists of Rooms 2 and 4 (cf. Fig. 1). Suppose further that the intelligent building heating management regulates the floor temperature to approximately $21°C$ when the ambient temperature is $(10 \pm 2)°C$. A possible attack scenario is that the left/right tenant chooses to intentionally decrease the temperature reading of the apartment in order to trick the heating management system into heating his/her apartment to the desired temperature range between $24°C$ and $26°C$. On the other hand, the higher temperature may also be a result of an accidental persistent fault in the temperature regulation system.

Thus, our objective as the building manager is to minimally perturb the temperature regulation system in order to conclusively distinguish among a nominal, a fault and an attack scenario, hence detecting an intentional fraud.

3. PRELIMINARIES

3.1 Notation and Definitions

Let $\mathbf{x} \in \mathbb{R}^n$ denote a vector and $\mathbf{M} \in \mathbb{R}^{n\times m}$ represent a matrix. The vector norm of $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_i$ with $i \in \{1, 2, \infty\}$, $\mathbf{M}^\intercal$ denotes the transpose of $\mathbf{M}$, while $\mathbf{1}$ and I represent the vector of ones and the identity matrix of appropriate dimensions, respectively. The diag and vec operators are defined for a collection of matrices $\mathbf{M}_i, i = 1, \ldots, n$ and matrix $\mathbf{M}$ as follows:

$$\underset{i=1}{\overset{n}{\operatorname{diag}}}\{\mathbf{M}_i\} = \begin{bmatrix} \mathbf{M}_1 & & \\ & \ddots & \\ & & \mathbf{M}_n \end{bmatrix}, \quad \underset{i=1}{\overset{n}{\operatorname{vec}}}\{\mathbf{M}_i\} = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_n \end{bmatrix},$$

$$\underset{N}{\operatorname{diag}}\{\mathbf{M}\} = I_N \otimes \mathbf{M}, \qquad \underset{N}{\operatorname{vec}}\{\mathbf{M}\} = \mathbf{1}_N \otimes \mathbf{M},$$

where $\otimes$ is the Kronecker product. The set of positive integers up to n is denoted by $\mathbb{Z}_n^+$, and the set of non-negative integers up to n is denoted by $\mathbb{Z}_n^0$. We will also make use of Special Ordered Set of degree 1 (SOS-1) constraints in our optimization solution, defined as follows:

Definition 1. (SOS-1 Constraints [Gurobi (2015)]). A special ordered set of degree 1 (SOS-1) constraint is a set of integer, continuous or mixed-integer scalar variables for which at most one variable in the set may take a value other than zero, denoted as SOS-1: $\{v_1, \ldots, v_N\}$. For instance, if $v_i \neq 0$, then this constraint imposes that $v_1 = \ldots = v_{i-1} = v_{i+1} = \ldots = v_N = 0$.

3.2 Modeling Framework and Problem Formulation

Consider N discrete-time affine time-invariant models $\mathcal{G}_i = (A_i, B_i, C_i, D_i, f_i, g_i)$, each with states $\boldsymbol{x}_i \in \mathbb{R}^n$, outputs $z_i \in \mathbb{R}^p$, and inputs $\boldsymbol{u}_i \in \mathbb{R}^m$. The models evolve according to the state equations

$$\boldsymbol{x}_i(k+1) = A_i \boldsymbol{x}_i(k) + B_i \boldsymbol{u}_i(k) + f_i, \tag{1}$$

and their output equations are

$$z_i(k) = C_i \boldsymbol{x}_i(k) + D_i \boldsymbol{u}_i(k) + g_i. \tag{2}$$

The initial condition for model i, denoted by $\boldsymbol{x}_i^0 = \boldsymbol{x}_i(0)$, is constrained to a polyhedral set:

$$\boldsymbol{x}_i^0 \in \mathcal{X}_0 = \{\boldsymbol{x} \in \mathbb{R}^n : P_0 \boldsymbol{x} \leq p_0\}, \ \forall i \in \mathbb{Z}_N^+. \tag{3}$$

The states $\boldsymbol{x}_i$ are partitioned into $x_i \in \mathbb{R}^{n_x}$ and $y_i \in \mathbb{R}^{n_y}$, where $n_y = n - n_x$, and the inputs $\boldsymbol{u}_i$ are partitioned into $u \in \mathbb{R}^{m_u}$, $v_i \in \mathbb{R}^{m_v}$ and $w_i \in \mathbb{R}^{m_w}$, where $m_u + m_v + m_w = m$, as follows:

$$\boldsymbol{x}_i(k) = \begin{bmatrix} x_i(k) \\ y_i(k) \end{bmatrix}, \boldsymbol{u}_i(k) = \begin{bmatrix} u(k) \\ v_i(k) \\ w_i(k) \end{bmatrix}. \tag{4}$$

These partitions facilitate the modeling of 'responsibilities' for the different components of the inputs $\boldsymbol{u}_i$. The states x_i and y_i represent the subset of the states $\boldsymbol{x}_i$ that are the 'responsibilities' of the controlled and uncontrolled inputs, u and w_i, respectively. The term 'responsibility' in this paper is to be interpreted as u and w_i, respectively, having to independently satisfy the following polyhedral state constraints (for $k \in \mathbb{Z}_T^+$):

$$x_i(k) \in \mathcal{X}_{x,i} = \{x \in \mathbb{R}^{n_x} : P_{x,i} x \leq p_{x,i}\}, \tag{5}$$

$$y_i(k) \in \mathcal{X}_{y,i} = \{y \in \mathbb{R}^{n_y} : P_{y,i} y \leq p_{y,i}\}, \tag{6}$$

subject to constrained inputs described by polyhedral sets (for $k \in \mathbb{Z}_{T-1}^0$):

$$u(k) \in \mathcal{U} = \{u \in \mathbb{R}^{m_u} : Q_u u \leq q_u\}, \tag{7}$$

$$w_i(k) \in \mathcal{W}_i = \{w \in \mathbb{R}^{m_{w_i}} : Q_{w,i} w \leq q_{w,i}\}. \tag{8}$$

On the other hand, the noise input v_i is also polyhedrally constrained, i.e.,

$$v_i(k) \in \mathcal{V} = \{v \in \mathbb{R}^{m_v} : Q_v v \leq q_v\}, \tag{9}$$

has no responsibility to satisfy any state constraints. In the fraud detection example in the previous section, u is the input that the building manager designs, v is the unknown variation in the ambient temperature, while w_i is the attack signal that is chosen by a selfish tenant.

Remark 1. Since it is the responsibility of w_i to satisfy the constraint in (6), it is important to make sure that the models are meaningful in the sense that over the time horizon T of interest and for each $i \in \mathbb{Z}_N^+$,

$$\exists w_i(k) \in \mathcal{W}_i, \forall k \in \mathbb{Z}_{T-1}^0 : \text{(6) is satisfied} \tag{10}$$

for any given $\boldsymbol{x}_i^0 \in \mathcal{X}_0$and for any given $u(k) \in \mathcal{U}$ and $v_i(k) \in \mathcal{V}$ for all $k \in \mathbb{Z}_{T-1}^0$. We refer to affine models satisfying this assumption as *well-posed* and assume throughout the paper that the given affine models are always well-posed. Note that models that do not satisfy this assumption are unpractical, since we would essentially be delegating responsibility to the uncontrolled input that is impossible to satisfy.

Using the above partitions of states and inputs, the corresponding partitioning of the state and output equations in (1) and (2) are:

$$\begin{aligned} \boldsymbol{x}_i(k+1) = {} & \begin{bmatrix} A_{xx,i} & A_{xy,i} \\ A_{yx,i} & A_{yy,i} \end{bmatrix} \boldsymbol{x}_i(k) \\ & + \begin{bmatrix} B_{xu,i} & B_{xv,i} & B_{xw,i} \\ B_{yu,i} & B_{yv,i} & B_{yw,i} \end{bmatrix} \boldsymbol{u}_i(k) + \begin{bmatrix} f_{x,i} \\ f_{y,i} \end{bmatrix}, \\ z_i(k) = {} & C_i \boldsymbol{x}_i(k) + [D_{u,i} \ \ D_{v,i} \ \ D_{w,i}] \, \boldsymbol{u}_i(k) + g_i. \end{aligned}$$

$$\overline{A}_{i,T} = \begin{bmatrix} A_i \\ A_i^2 \\ \vdots \\ A_i^T \end{bmatrix}, \quad \Theta_{i,T} = \begin{bmatrix} I & 0 & \cdots & 0 \\ A_i & I & \cdots & 0 \\ \vdots & & \ddots & \\ A_i^{T-1} & A_i^{T-2} & \cdots & I \end{bmatrix}, \quad \begin{matrix} \overline{A} = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{\overline{A}_{i,T}\}, & \overline{C} = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{E_i\}, & \tilde{f}_{i,T} = \Theta_{i,T} \overline{f}_{i,T}, & \tilde{g}_{i,T} = \underset{T}{\operatorname{vec}}\{g_i\}, \\ E_i = \underset{T}{\operatorname{diag}}\{C_i\}, & \overline{f}_{i,T} = \underset{T}{\operatorname{vec}}\{f_i\}, & \tilde{f} = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{\tilde{f}_{i,T}\}, & \tilde{g} = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{\tilde{g}_{i,T}\}, \end{matrix}$$

$$\overline{D}_u = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{F_{u,i}\}, \quad \overline{D}_v = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{F_{v,i}\}, \quad \overline{D}_w = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{F_{w,i}\}, \quad \Gamma_u = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{\Gamma_{u,i,T}\}, \quad \Gamma_v = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{\Gamma_{v,i,T}\}, \quad \Gamma_w = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{\Gamma_{w,i,T}\};$$

for $\dagger = \{x, y\}$ and $* = \{u, v, w\}$:

$$B_{*,i} = \begin{bmatrix} B_{x*,i} \\ B_{y*,i} \end{bmatrix}, \quad \Gamma_{*,i,T} = \begin{bmatrix} B_{*,i} & 0 & \cdots & 0 \\ A_i B_{*,i} & B_{*,i} & \cdots & 0 \\ \vdots & & \ddots & \\ A_i^{T-1} B_{*,i} & A_i^{T-2} B_{*,i} & \cdots & B_{*,i} \end{bmatrix}, \quad \begin{matrix} F_{*,i} = \underset{T}{\operatorname{diag}}\{D_{*,i}\}, & \\ B_{\dagger *,\mathrm{d},i,T} = \underset{T}{\operatorname{diag}}\{B_{\dagger *,i}\}, & M_{\dagger,i,T} = A_{\dagger,\mathrm{d},i,T} \begin{bmatrix} I \\ \overline{A}_{i,T-1} \end{bmatrix}, \\ A_{\dagger,\mathrm{d},i,T} = \underset{T}{\operatorname{diag}}\{\begin{bmatrix} A_{\dagger x,i} & A_{\dagger y,i} \end{bmatrix}\}, & M_\dagger = \operatorname{diag}_{i=1}^N\{M_{\dagger,i,T}\}, \end{matrix}$$

$$\tilde{f}_{\dagger,i,T} = A_{\dagger,\mathrm{d},i,T} \begin{bmatrix} 0 \\ \Theta_{i,T-1} \end{bmatrix} \overline{f}_{i,T-1} + \overline{f}_{\dagger,i,T}, \quad \tilde{f}_\dagger = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{\tilde{f}_{\dagger,i,T}\}, \quad \overline{f}_{\dagger,i,T} = \underset{T}{\operatorname{vec}}\{f_{\dagger,i}\},$$

$$\Gamma_{\dagger *,i,T} = A_{\dagger,\mathrm{d},i,T} \begin{bmatrix} 0 & 0 \\ \Gamma_{*,i,T-1} & 0 \end{bmatrix} + B_{\dagger *,\mathrm{d},i,T}, \quad \Gamma_{\dagger u} = \underset{i=1}{\overset{N}{\operatorname{vec}}}\{\Gamma_{\dagger u,i,T}\}, \quad \Gamma_{\dagger v} = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{\Gamma_{\dagger v,i,T}\}, \quad \Gamma_{\dagger w} = \underset{i=1}{\overset{N}{\operatorname{diag}}}\{\Gamma_{\dagger w,i,T}\}, \tag{$\star$}$$

Further, we will consider a time horizon of length T and introduce some time-concatenated notation. The time-concatenated states and outputs are defined as

$$\boldsymbol{x}_{i,T} = \underset{j=1}{\overset{T}{\text{vec}}}\{\boldsymbol{x}_i(j)\}, \quad x_{i,T} = \underset{j=1}{\overset{T}{\text{vec}}}\{x_i(j)\},$$
$$y_{i,T} = \underset{j=1}{\overset{T}{\text{vec}}}\{y_i(j)\}, \quad z_{i,T} = \underset{j=1}{\overset{T}{\text{vec}}}\{z_i(j)\},$$

while the time-concatenated inputs are defined as

$$\boldsymbol{u}_{i,T} = \underset{j=0}{\overset{T-1}{\text{vec}}}\{\boldsymbol{u}_i(j)\}, \; u_T = \underset{j=0}{\overset{T-1}{\text{vec}}}\{u(j)\},$$
$$v_{i,T} = \underset{j=0}{\overset{T-1}{\text{vec}}}\{v(j)\}, \; w_{i,T} = \underset{j=0}{\overset{T-1}{\text{vec}}}\{w_i(j)\}.$$

Then, concatenating $\boldsymbol{x}_i^0$, $\boldsymbol{x}_{i,T}$, $x_{i,T}$, $v_{i,T}$, $w_{i,T}$, $y_{i,T}$ and $z_{i,T}$ across all modes as

$$\boldsymbol{x}_0 = \underset{i=1}{\overset{N}{\text{vec}}}\{\boldsymbol{x}_i^0\}, \; \boldsymbol{x}_T = \underset{i=1}{\overset{N}{\text{vec}}}\{\boldsymbol{x}_{i,T}\}, \; x_T = \underset{i=1}{\overset{N}{\text{vec}}}\{x_{i,T}\}, \tag{11}$$
$$v_T = \underset{i=1}{\overset{N}{\text{vec}}}\{v_{i,T}\}, w_T = \underset{i=1}{\overset{N}{\text{vec}}}\{w_{i,T}\}, \; y_T = \underset{i=1}{\overset{N}{\text{vec}}}\{y_{i,T}\}, z_T = \underset{i=1}{\overset{N}{\text{vec}}}\{z_{i,T}\},$$

the states and outputs over the entire time horizon can be written as simple functions of the initial state $\boldsymbol{x}_0$ and input vectors u_T and w_T, as well as noise v_T:

$$x_T = M_x\boldsymbol{x}_0 + \Gamma_{xu}u_T + \Gamma_{xv}v_T + \Gamma_{xw}w_T + \tilde{f}_x, \tag{12}$$
$$y_T = M_y\boldsymbol{x}_0 + \Gamma_{yu}u_T + \Gamma_{yv}v_T + \Gamma_{yw}w_T + \tilde{f}_y, \tag{13}$$
$$\boldsymbol{x}_T = \overline{A}\boldsymbol{x}_0 + \Gamma_u u_T + \Gamma_v v_T + \Gamma_w w_T + \tilde{f}, \tag{14}$$
$$z_T = \overline{C}\boldsymbol{x}_T + \overline{D}_u u_T + \overline{D}_v v_T + \overline{D}_w w_T + \tilde{g}, \tag{15}$$

where the matrices and vectors M_*, Γ_{*u}, Γ_{*v}, Γ_{*w} and $\tilde{f}_*$ for $* \in \{x, y\}$, as well as $\overline{A}$, Γ_u, Γ_v, Γ_w, $\overline{C}$, $\overline{D}_u$, $\overline{D}_v$, $\overline{D}_w$, $\tilde{f}$ and $\tilde{g}$ are summarized in ($\star$).

3.3 Active Model Discrimination Problem

The problem of input design for model discrimination can be defined formally as follows:

Problem 1. (Active Model Discrimination). Find an optimal input sequence u_T^* subject to the input constraint in (7) and its corresponding 'responsibility' in (5) that minimizes a desired or given cost function $c(u_T)$ such that for any plausible output sequence z_T that satisfies (3),(6),(8),(9), only one model is valid. In other words, find a feasible and optimal input sequence u_T^* such that the output trajectories of each pair of models have to differ in at least one time instance for all possible initial states $\boldsymbol{x}_0$ and uncontrolled input and noise sequences w_T and v_T. The optimization problem can be formally stated as follows:

$$\min_{u_T, x_T} c(u_T)$$
$$\text{s.t.} \quad \forall k \in \mathbb{Z}_T^+ : (2),(4),(5) \text{ hold}, \tag{16a}$$
$$\forall k \in \mathbb{Z}_{T-1}^0 : (1),(7) \text{ hold}, \tag{16b}$$
$$\forall i, j \in \mathbb{Z}_N^+ : i < j, \exists k \in \mathbb{Z}_T^0 : z_i(k) \neq z_j(k) \tag{16c}$$
$$\forall v_T, w_T, y_T, \boldsymbol{x}_0 : (3),(6),(8),(9) \text{ hold}. \tag{16d}$$

From the above problem statement, we note that the input u we design has to satisfy the state constraints in (5) for all models $i \in \mathbb{Z}_N^+$ and for any initial state $\boldsymbol{x}_i^0 \in \mathcal{X}_0$ and noise input $v_i(k) \in \mathcal{V}$, similar in spirit to robust control problems. On the other hand, the uncontrolled input w_i has to only satisfy its corresponding state constraints in (6), with the "aid" of the initial state and noise input.

4. ACTIVE MODEL DISCRIMINATION APPROACH

In this section, we present our approach to solve Problem 1. Section 4.1 follows a similar approach to our previous work in Jacobsen et al. (2017) to formulate the active model discrimination problem as a mixed-integer linear program (MILP), but further includes direct feedthrough terms and also process and measurement noise. On the other hand, Section 4.2 proposes a sequence of optimization problems that can overcome the difficulty associated with relaxing a relatively restrictive assumption in Jacobsen et al. (2017).

4.1 Model Discrimination via an MILP

In this section, our objective is to convert Problem 1 to a tractable optimization problem. Hence, we first replace the non-convex *separability condition* in (16c) with $|z_i(k) - z_j(k)| \geq \epsilon$, where ϵ is the amount of desired separation or simply the machine precision or optimization tolerance, and then by introducing slack variables and SOS-1 constraints, we transform (16c) into:

$$\begin{array}{ll} \forall i, j \in \mathbb{Z}_N^+, i < j, & z_{i,l}(k) - z_{j,l}(k) - \varepsilon + s_{i,j,k,l,1} \geq 0, \\ \forall l \in \mathbb{Z}_p^+, k \in \mathbb{Z}_T^0, : & z_{j,l}(k) - z_{i,l}(k) - \varepsilon + s_{i,j,k,l,2} \geq 0, \\ \forall \alpha \in \{1, 2\} & a_{i,j,k,l,\alpha} \in \{0, 1\}, \\ & \text{SOS-1: } \{s_{i,j,k,l,\alpha}, a_{i,j,k,l,\alpha}\}, \end{array} \tag{17}$$

$$\sum_{k \in \mathbb{Z}_T^0} \sum_{l \in \mathbb{Z}_p^1} \sum_{\alpha \in \{1,2\}} a_{i,j,k,l,\alpha} \geq 1, \tag{18}$$

where a is the vector of binary variables $a_{i,j,k,l,\alpha}$ concatenated over the indices in the order i, j, k, l, α, and s is similarly a vector of slack variables $s_{i,j,k,l,\alpha}$, defined as $s = [s_1^\intercal \; s_2^\intercal]^\intercal$, where s_α for $\alpha \in \{1, 2\}$ is defined in ($\star\star$).

The above problem formulation is still nontrivial because of the semi-infinite constraints in the form of (16d). Hence, we will convert this problem into the following mixed-integer linear program (MILP) using some tools from robust optimization Ben-Tal et al. (2009); Bertsimas et al. (2011). However, before we proceed, note that the constraint (6) that represents the responsibility of the uncontrolled (attack) input can be obtained in terms of the uncertain variables $\overline{x} = [\boldsymbol{x}_0^\intercal \; v_T^\intercal \; w_T^\intercal]^\intercal$ as

$$H_y\overline{x} \leq \overline{p}_y - \overline{P}_y\Gamma_{yu}u_T - \overline{P}_y\tilde{f}_y,$$

where H_y, $\overline{p}_y$, $\overline{P}_y$, Γ_{yu} and $\tilde{f}_y$ are defined in ($\star$) and ($\star\star$). Then, we can obtain the following result by assuming that $\overline{P}_y\Gamma_{yu} = 0$ (to avoid bilinear terms as will be discussed below), proof of which follows directly from our previous work in Jacobsen et al. (2017).

Theorem 1. (Discriminating Input Design (DID)). Let $\overline{P}_y\Gamma_{yu} = 0$ where $\overline{P}_y$ and Γ_{yu} are defined in ($\star$) and ($\star\star$). Then, given well-posed affine models and the separability index ϵ, the following problem:

$$\min_{u_T, s, a, \Pi} c(u_T) \tag*{(P_{DID})}$$
$$\text{s.t.} \; \overline{Q}_u u_T \leq \overline{q}_u, \Phi^\intercal\Pi = R^\intercal, \Pi \geq 0, \tag{19a}$$
$$\Pi^\intercal\phi \leq r(u_T, s), \tag{19b}$$
$$a \in \{0, 1\}^{pTN(N-1)}, \tag{19c}$$
$$\sum_{k \in \mathbb{Z}_T^0} \sum_{l \in \mathbb{Z}_p^1} \sum_{\alpha \in \{1,2\}} a_{i,j,k,l,\alpha} \geq 1, \tag{19d}$$
$$\text{SOS-1: } \{s_{i,j,k,l,\alpha}, a_{i,j,k,l,\alpha}\}, \tag{19e}$$

$\forall i, j \in \mathbb{Z}_N^+ : i < j$, $\forall l \in \mathbb{Z}_p^1$, $\forall k \in \mathbb{Z}_T^0$ and $\alpha \in \{1, 2\}$, where Π is a matrix of dual variables, while $\overline{Q}_u$, Φ, R, $\overline{q}_u$, ϕ and $r(u_T, s)$ are problem-dependent matrices and vectors that are defined in ($\star\star$), is equivalent up to the separability index ϵ to Problem 1 and its solution is optimal.

The equivalence up to the separability index ϵ can be interpreted as a restriction, which enforces a 'stricter' separation of the output trajectories by a minimum 'distance' ϵ in at least one time instance. As $\epsilon \to 0$, this restriction becomes equivalent to (16c). In this paper, we consider ϵ to be the numerical precision of the solver.

Through simple matrix manipulations, it can be seen that if $\overline{P}_y \Gamma_{yu} \neq 0$, then the constraint in (19b) will become

$$\Pi^{\mathsf{T}} \psi(u_T) \leq r(u_T, s), \tag{20}$$

where $\psi(u_T)$, defined in ($\star\star$), is a linear function of u_T that is a decision variable. Hence, a bilinear term will appear, which makes the problem a non-linear mixed-integer program. In the next section, we provide an algorithm to solve Problem 1 for this case in a tractable manner.

4.2 Sequence of Restrictions

As discussed before, (P_{DID}) becomes a mixed-integer nonlinear problem because of the bilinear term that appears in (19b) as a result of the product between Π^{T} and $\psi(u_T)$. In order to solve this nonlinear program, we propose to solve a sequence of problems with restricted feasibility sets. This ensures the feasibility of the obtained solution, if the approach finds one; that is, the solution will indeed be separating. We show that this approach is computationally tractable, but comes at a cost that the obtained solution is no longer guaranteed to be optimal. First, recall from (20) that the bilinear terms come from the dependence of the uncertainty set, $\Phi\overline{x} \leq \psi(u_T)$, on our control actions, u_T. By relaxing this uncertainty set, we restrict the feasibility set of u_T. That is, by giving more freedom to the uncontrolled (attack) inputs, we restrict the set of our available actions. Let us define:

$$\tilde{\psi}_0^i = \max_{u_T \in \mathcal{U}^T} \psi^i(u_T), \tag{21}$$

where $\tilde{\psi}_0^i$ and $\psi^i(u_T)$ are the i^{th} rows of $\tilde{\psi}_0$ and $\psi(u_T)$, respectively. Clearly, $\tilde{\psi}_0$ denotes the component-wise maximum of $\psi(u_T)$ over all of our available actions. Now, replacing $\psi(u_T)$ with $\tilde{\psi}_0$ will be a relaxation of

$$\Phi\overline{x} \leq \psi(u_T) \to \Phi\overline{x} \leq \tilde{\psi}_0,$$

which in turn results in a restriction on our inputs. With this, we will now define the restricted model discrimination problem as:

Problem 2. The restricted active model discrimination problem with parameter $\tilde{\psi}$ is defined as in (P_{DID}), except with $\psi(u_T)$ being replaced with constant vector $\tilde{\psi}$. We refer to this problem as *Restricted*($\tilde{\psi}$).

Note that the above problem can now be tractably solved using off-the-shelf mixed-integer linear program (MILP) softwares. However, the obtained solution can be far from optimal. Hence, in order to obtain a better solution that remains feasible, we propose the following algorithm to implement a sequence of restrictions (cf. Fig. 2).

Algorithm 1 Sequence of Restrictions

```
Input: $\tilde{\psi}_0$ (cf. (21)), $\nu$ (convergence criterion).
Initialize: $k = 0$.
Solve $u_{T,0} = Restricted(\tilde{\psi}_0)$.
Let $\rho_0 = c(u_{T,0})$ and $\rho_{-1} = \rho_0 + \nu$.
if ($\rho_{k-1} - \rho_k \geq \nu$) then, repeat the following:
    $k \leftarrow k + 1$.
    Let $\tilde{\psi}_k = \operatorname{vec}_{i=1}^{n_\psi} \tilde{\psi}_k^i$ ($n_\psi$ = cardinality of $\psi(u_T)$) with
        $\tilde{\psi}_k^i = \max_{u_T \in \mathcal{U}^T \cap \{u_T | c(u_T) \leq \rho_{k-1}\}} \psi^i(u_T)$.
    Solve $u_{T,k} = Restricted(\tilde{\psi}_k)$.
    Let $\rho_k = c(u_{T,k})$.
else
    return $u_{T,k}, \rho_k$.
end if
```

Besides being computationally tractable, the above algorithm possesses nice properties as is shown in the following.

Theorem 2. Every solution $u_{T,k}$ produced by Algorithm 1 is a (suboptimal) feasible solution to (P_{DID}) and the cor-

$$\text{for } \dagger = \{x, y\}: \quad \overline{P}_\dagger = \operatorname*{diag}_{i=1}^{N} \operatorname*{diag}_{T}\{P_{\dagger,i}\}, \quad \overline{p}_\dagger = \operatorname*{diag}_{i=1}^{N} \operatorname*{diag}_{T}\{p_{\dagger,i}\}, \quad H_\dagger = \overline{P}_\dagger \begin{bmatrix} M_\dagger & \Gamma_{\dagger v} & \Gamma_{\dagger w} \end{bmatrix}, \quad h_\dagger(u_T) = \overline{p}_\dagger - \overline{P}_\dagger \Gamma_{\dagger u} u_T - \overline{P}_\dagger \tilde{f}_\dagger;$$

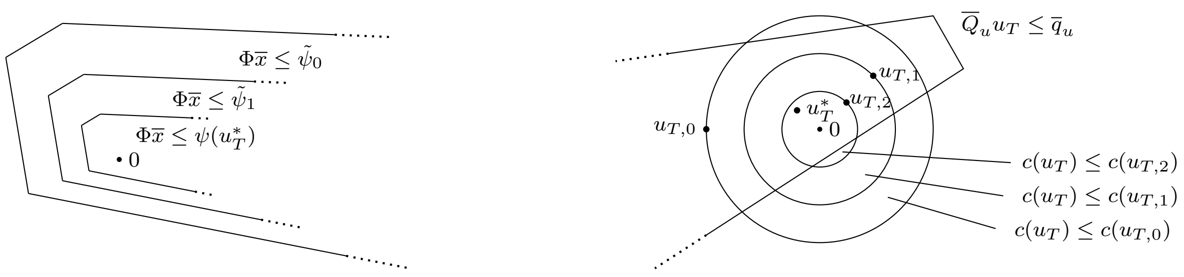
$$\overline{Q}_u = \operatorname*{diag}_{T}\{Q_u\}, \ \overline{Q}_v = \operatorname*{diag}_{NT}\{Q_v\}, \ \overline{Q}_w = \operatorname*{diag}_{i=1}^{N} \operatorname*{diag}_{T}\{Q_{w,i}\}, \ \ \overline{q}_u = \operatorname*{diag}_{T}\{q_u\}, \ \overline{q}_v = \operatorname*{diag}_{NT}\{q_v\}, \ \overline{q}_w = \operatorname*{diag}_{i=1}^{N} \operatorname*{diag}_{T}\{q_{w,i}\}, \ \ \overline{h}_y(u_T) = \overline{p}_y - \overline{P}_y \tilde{f}_y,$$

$$\text{for } * = \{v, w\}: F_* = \begin{bmatrix} F_{*,1} & -F_{*,2} & 0 & \cdots & \cdots & 0 \\ F_{*,1} & 0 & -F_{*,3} & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & \cdots & \cdots & 0 & F_{*,N-1} & -F_{*,N} \end{bmatrix}, \ E = \begin{bmatrix} E_1 & -E_2 & 0 & \cdots & \cdots & 0 \\ E_1 & 0 & -E_3 & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & \cdots & \cdots & 0 & E_{N-1} & -E_N \end{bmatrix}, \ F_u = \begin{bmatrix} F_{u,1} - F_{u,2} \\ F_{u,1} - F_{u,3} \\ \vdots \\ F_{u,N-1} - F_{u,N} \end{bmatrix},$$

$$H_{\overline{x}} = \begin{bmatrix} \operatorname*{diag}_{N}\{P_0\} & 0 & 0 \\ 0 & \overline{Q}_v & 0 \\ 0 & 0 & \overline{Q}_w \end{bmatrix}, \ h_{\overline{x}} = \begin{bmatrix} \operatorname*{vec}_{N}\{p_0\} \\ \overline{q}_v \\ \overline{q}_w \end{bmatrix}, \ g = \begin{bmatrix} \tilde{g}_1 - \tilde{g}_2 \\ \tilde{g}_1 - \tilde{g}_3 \\ \vdots \\ \tilde{g}_{N-1} - \tilde{g}_N \end{bmatrix}, \ s_{i,j,\alpha} = \begin{bmatrix} \operatorname{vec}_{l=1}^{p}\{s_{i,j,1,l,\alpha}\} \\ \vdots \\ \operatorname{vec}_{l=1}^{p}\{s_{i,j,T,l,\alpha}\} \end{bmatrix}, \quad s_\alpha = \begin{bmatrix} \operatorname{vec}_{j=2}^{N}\{s_{1,j,\alpha}\} \\ \operatorname{vec}_{j=3}^{N}\{s_{2,j,\alpha}\} \\ \vdots \\ s_{N-1,N,\alpha} \end{bmatrix},$$

$$\overline{E} = \begin{bmatrix} E \\ -E \end{bmatrix}, \quad \overline{F}_u = \begin{bmatrix} F_u \\ -F_u \end{bmatrix}, \quad \overline{F}_v = \begin{bmatrix} F_v \\ -F_v \end{bmatrix}, \quad \overline{F}_w = \begin{bmatrix} F_w \\ -F_w \end{bmatrix}, \quad \overline{g} = \begin{bmatrix} g \\ -g \end{bmatrix}, \quad \Phi = \begin{bmatrix} H_y \\ H_{\overline{x}} \end{bmatrix}, \quad \psi(u_T) = \begin{bmatrix} h_y(u_T) \\ h_{\overline{x}} \end{bmatrix}, \quad \phi = \begin{bmatrix} \overline{h}_y \\ h_{\overline{x}} \end{bmatrix},$$

$$\Lambda = \overline{E} \begin{bmatrix} \overline{A} & \Gamma_v & \Gamma_w \end{bmatrix} + \begin{bmatrix} 0_{2p \times nN} & \overline{F}_v & \overline{F}_w \end{bmatrix}, \quad \lambda(u_T, s) = \epsilon\mathbf{1} - s - \overline{g} - (\overline{E}\Gamma_u + \overline{F}_u)u_T - \overline{E}\tilde{f}, \quad R = \begin{bmatrix} -\Lambda \\ H_x \end{bmatrix}, \ r(u_T, s) = \begin{bmatrix} -\lambda(u_T, s) \\ h_x(u_T) \end{bmatrix}. \quad (\star\star)$$


$\Phi\overline{x} \leq \tilde{\psi}_0$
$\Phi\overline{x} \leq \tilde{\psi}_1$
$\Phi\overline{x} \leq \psi(u_T^*)$
0
$\overline{Q}_u u_T \leq \overline{q}_u$
$u_{T,1}$
$u_{T,2}$
u_T^*
0
$u_{T,0}$
$c(u_T) \leq c(u_{T,2})$
$c(u_T) \leq c(u_{T,1})$
$c(u_T) \leq c(u_{T,0})$

(a) Sequence of restricted polytopes $\{\overline{x} : \Phi\overline{x} \leq \tilde{\psi}\}$ in the $\overline{x}$-space; $\overline{x} = [\boldsymbol{x}_0^\top \; v_T^\top \; w_T^\top]^\top$.

(b) Sequence of restricted $c(u_T)$-balls, illustrated as 2-norm balls, and the polytope $\{u_T : \overline{Q}_u u_T \leq \overline{q}_u\}$ in the u_T-space; $\rho_k = c(u_{T,k})$.

Fig. 2. Illustration of the sequence of restrictions algorithm.

responding objective value ρ_k is monotonically decreasing. Furthermore, Algorithm 1 terminates.

Proof. Let $u_{T,k}$ be a solution to *Restricted*$(\tilde{\psi}_k)$. Then, by definition of the uncertainty set in Line 7 of Algorithm 1, $u_{T,k}$ is also a solution to *Restricted*$(\tilde{\psi}_{k+1})$. Since $\{\Phi\overline{x} \leq \tilde{\psi}_{k+1}\} \subseteq \{\Phi\overline{x} \leq \tilde{\psi}_k\}$ by construction with the componentwise maximum in (21) (cf. Fig. 2(a)), we obtain $\rho_{k+1} \leq \rho_k$ (cf. Fig. 2(a)) since the solutions are optimal for their respective problems. Thus, $\rho_k = c(u_{T,k})$ form a decreasing sequence of real numbers. Since $\rho_k \geq 0$ is bounded from below by assumption, ρ_k converge by the monotone convergence theorem; thus, the algorithm terminates.

5. APPLICATION TO FRAUD DETECTION IN SMART BUILDINGS

5.1 System Model

Nominal Model. We consider a building with four rooms, which has a radiant system with two heaters (boilers+pumps), adapted from Nghiem et al. (2013) and illustrated in Fig. 1. We assume that the building manager has direct control over the core temperatures of both boilers. Furthermore, he/she has access to temperature measurements of all four rooms. In addition, we assume constant flow for both pumps.

This system can be represented by an affine state-space model with four states and two inputs, as follows:

$$c_1\dot{T}_1(t) = k_{r,1}(T_{c,1} - T_1) + k_1(T_a - T_1) + \textstyle\sum_{j\in\{2,3\}} k_{1j}(T_j - T_1)$$
$$c_2\dot{T}_2(t) = k_{r,2}(T_{c,2} - T_2) + k_2(T_a - T_2) + \textstyle\sum_{j\in\{1,4\}} k_{2j}(T_j - T_2)$$
$$c_3\dot{T}_3(t) = k_3(T_a - T_3) + \textstyle\sum_{j\in\{1,4\}} k_{3j}(T_j - T_3)$$
$$c_4\dot{T}_4(t) = k_{r,4}(T_{c,2} - T_4) + k_4(T_a - T_4) + \textstyle\sum_{j\in\{2,3\}} k_{4j}(T_j - T_4),$$

where the list of parameters are given in Table 1 and their values [1] are chosen according to the ranges provided in Nghiem et al. (2013) and with initial conditions $20 \leq T_i(0) \leq 26$ for $i \in \{1, 2, 3, 4\}$. Additionally, we model the ambient temperature as $T_a = \overline{T}_a + \delta T_a$, with a known average $\overline{T}_a$ and a bounded uncertainty $\delta T_a \in \mathcal{T}_a$.

The system is discretized with a sampling time of 5 minutes. The system matrices of the discrete state space are obtained as follows:

Table 1. Radiant system parameters

T_a	ambient air temperature (°C)
T_i	air temperature of zone i (°C)
$T_{c,i}$	core temperature of zone i (°C)
k_i	thermal conductance between T_i and T_a $(W/(Km^2))$
$k_{r,i}$	thermal conductance between $T_{c,i}$ and T_i $(W/(Km^2))$
k_{ij}	thermal conductance between zones i and j $(W/(Km^2))$
c_i	thermal capacitance of zone i (kJ/K)
$c_{r,i}$	thermal capacitance of the slab of zone i (kJ/K)

$$A = \begin{bmatrix} 0.0907 & 0.0659 & 0.1232 & 0.0672 \\ 0.0659 & 0.0817 & 0.0758 & 0.0725 \\ 0.1109 & 0.0682 & 0.2558 & 0.1344 \\ 0.0576 & 0.0621 & 0.1280 & 0.1233 \end{bmatrix}, \; C = I_4, \; f = \begin{bmatrix} 0.4583 \\ 0.4357 \\ 0.5928 \\ 0.4648 \end{bmatrix},$$

$$B_u = \begin{bmatrix} 0.4303 & 0.1768 \\ 0.1232 & 0.5373 \\ 0.1571 & 0.2143 \\ 0.0536 & 0.5289 \end{bmatrix}, \; B_v = \begin{bmatrix} 0.04583 \\ 0.0436 \\ 0.0593 \\ 0.0465 \end{bmatrix}, \; D = \mathbf{0},$$

where the inputs are the boiler core temperatures $T_{c,1}$ and $T_{c,2}$, while the noise is the uncertainty in the ambient temperature δT_a. An output feedback controller $\tilde{u} = Kz + u_{ff} + u$ (with $K = \begin{bmatrix} -0.087 & 0 & 0 & 0 \\ 0 & 0.3228 & 0 & -0.0974 \end{bmatrix}$ and $u_{ff} = \begin{bmatrix} 23.7704 \\ 17.0980 \end{bmatrix}$) is designed to regulate the temperature of the first room and the average temperature of the second and fourth rooms to a common desired temperature (21°C for this example). In addition, the manager adds a signal u as an active input that will be designed to detect attacks and faults. In the nominal case (no faults nor attacks), the output equation corresponds to the temperature measurements of all rooms, i.e., $z(t) = x(t) = [T_1(t) \; T_2(t) \; T_3(t) \; T_4(t)]^\top$.

Attack Model. The attack model assumes that the tenant of Room 1 (attacker) manipulates the temperature reading in his/her room by adding a signal $a_1(t)$ while preventing the active input u from being added by the building manager. More specifically, the attacker exploits the output feedback mechanism with the goal of regulating the temperature of Room 1 to his/her desired range. Mathematically, the output equation for the attack model is given by:

$$z(t) = Cx(t) + Du(t) + \begin{bmatrix} a_1(t) \\ \mathbf{0} \end{bmatrix}, -200 \leq a_1(t) \leq 200. \quad (22)$$

We further assume that the desired temperature of the attacker is $24 \leq T_1(t) \leq 26$. In our modeling framework, this means that it is the responsibility of the attack signal $a_1(t)$ to keep $T_1(t)$ in the desired range at all times.

[1] $\overline{T}_a = 10$, $k_1 = \frac{1}{2.1}$, $k_2 = \frac{1}{2.1}$, $k_3 = \frac{1}{2.1}$, $k_4 = \frac{1}{1.9}$, $k_{r,1} = \frac{1}{0.125}$, $k_{r,2} = \frac{1}{0.125}$, $k_{12} = k_{21} = \frac{1}{0.16}$, $k_{13} = \frac{1}{0.16}$, $k_{24} = k_{42} = \frac{1}{0.2}$, $k_{34} = k_{43} = \frac{1}{0.16}$, $c_1 = 1800$, $c_2 = 1800$, $c_3 = 2000$, $c_4 = 2100$, $c_{r,1} = 3500$, $c_{r,2} = 3500$.

Fault Model. To make the problem more interesting, we also consider a fault model that can cause a behavior similar to the one of the attack scenario. We assume that the valve of the pump of the first heater is stuck somewhere in middle, and therefore, the flow is slower. The effect of such a fault is modeled by: (i) changing the heat conductance between the first room and its core water to half of its nominal value, and (ii) modeling the uncertainty in the position of valve as a fault input (a noise with no responsibilities), $w_f(t)$.

Using the above models, Fig. 3 shows a sample trajectory of nominal, attack and fault models for two different active inputs u. As shown in the figure, the fault and attack models can produce similar outputs when no separating input is applied, whereas a separating input forces the trajectory of the fault model out of the attacker's desired temperature, thus enables us to distinguish between them.

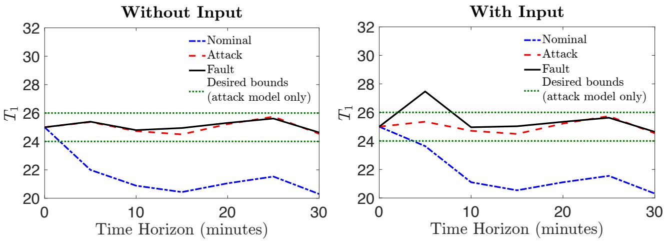


Fig. 3. Representative trajectories of three models when no input is applied (left), and when a separating input is applied (right).

5.2 Simulation Results

In this section, we present some results that are obtained when applying the proposed active model discrimination approach to the motivating example of fraud detection in smart buildings, described in Section 2. Our proposed method is general enough to capture a wide variety of attack and fault models in large residential and commercial buildings, but for illustration purposes, we restrict ourselves to a small residential building where the nominal, attack and fault models are obtained from first principles, as previously described with the following uncertainty and input sets: $\mathcal{T}_a = [-2,\ 2]$, $\mathcal{U} = \{u(t) \mid u(t) \in [-30,\ 30]\}$ and $\mathcal{W}_f = \{w_f(t) \mid w_f(t) \in [-85,\ -150]\}$ (set of fault inputs), unless otherwise specified. All the examples are implemented on a 3.4 GHz machine with 16 GB of memory that runs MacOS. For the implementation of the active model discrimination algorithms, we utilized Yalmip [Löfberg (2004)] and Gurobi [Gurobi (2015)] in MATLAB.

Convergence of Sequence of Restrictions. In Section 4.2, we proposed an iterative approach for tractably finding a feasible but potentially suboptimal solution to the active model discrimination problem. In Fig. 4, we plot the objective value versus the iteration number of the algorithm for three different objective functions. As illustrated by figure, the objective value is monotonically decreasing and converges after a few iterations, as desired. The criterion to stop the iterative process in this numerical example is chosen to be $\|u_T^{k-1}\|_\star - \|u_T^k\|_\star < \nu$, where $\star \in \{1, 2, \infty\}$, ν is the numerical machine tolerance and u_T^k denotes the solution at iteration k.

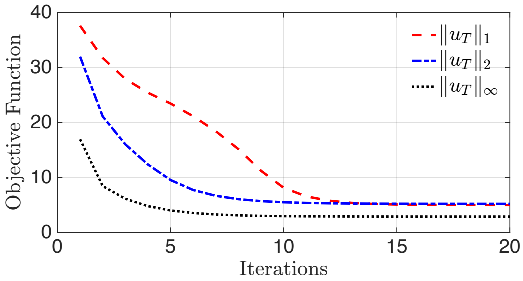


Fig. 4. Objective value vs. iterations.

Effect of Uncertainty in Ambient Temperature. The effect of noise or uncertainty is demonstrated here through simulation. We solve sequences of restricted active model discrimination problems (cf. Algorithm 1) for a range of uncertainty bounds in the ambient temperature. As expected, the value of objective value after the convergence of the sequence of restrictions increases when the uncertainty bound grows. The results are illustrated in Fig. 5.

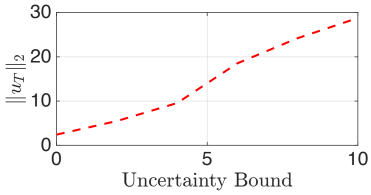


Fig. 5. Objective value vs. noise bound.

Effect of Objective Function on Separating Inputs. The intermediate separating inputs (before convergence) obtained from Algorithm 1 are illustrated in Fig. 6 for three different objective functions. Additionally, on the bottom right plot, the resulting "converged" inputs from Algorithm 1 are shown for all three objective functions, when only cooling (negative) inputs are allowed. Note that the color bars indicate the direction of convergence with increasing iterations. This also corresponds to a non-increasing sequence of objective values, as desired. Moreover, we observe that heating solutions have smaller objective values in this particular example.

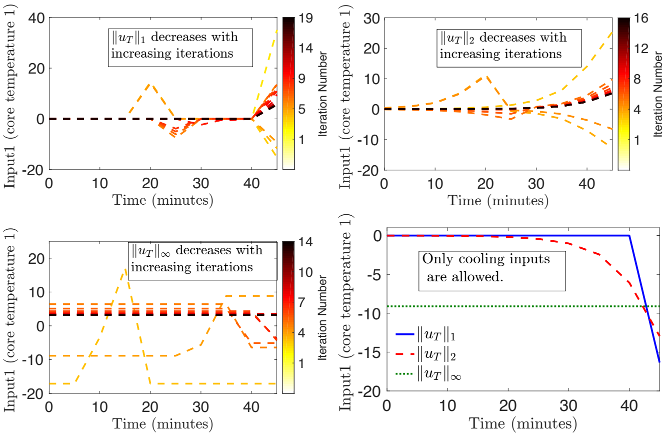


Fig. 6. Separating inputs for all iterations of Algorithm 1 for different objective functions (except bottom right). Separating inputs obtained from Algorithm 1 when only cooling is allowed (bottom right).

6. CONCLUSION

In this paper, we proposed an optimization-based approach for finding a separating input that guarantees the distinction amongst multiple noisy affine models with uncontrolled inputs. In our modeling framework, these uncontrolled inputs, which can include attack signals, are inputs of rational agents that have asymmetric responsibilities to satisfy state and input constraints that are represented by polyhedral constraints. This new framework extends our previous work in Jacobsen et al. (2017) by adding direct feedthrough and also noise (similar to uncontrolled inputs but without responsibilities). In addition, we removed the assumption that the states constraints that the uncontrolled/attack input are responsible for cannot be affected by the controlled inputs. The elimination of this assumption leads to bilinear terms that make the model discrimination problem nonlinear and non-convex.

Thus, we proposed an algorithm, which leverages a sequence of restrictions to find a feasible suboptimal solution in a computationally tractable manner. The sequence of mixed-integer linear programs (MILP) can be solved using off-the-shelf optimization softwares. To illustrate the efficacy of this approach, we applied it to the problem of fraud detection in smart buildings, where we considered both fault and attack models. The proposed approach delivered a suboptimal input that discriminates among nominal, fault and attack models, even when the separating control inputs are restricted to have significantly smaller bounds than that for the attacker.

As a future direction, we will employ nonlinear optimization approaches to solve a relaxation of Problem 1, which will provide us with a lower bound on the optimal value. Then, by comparing the result of Algorithm 1 with this obtained lower bound, we can draw conclusions about the suboptimality of the solution.

REFERENCES

Babaali, M. and Egerstedt, M. (2004). Observability of switched linear systems. In *International Workshop on Hybrid Systems: Computation and Control*, 48–63. Springer.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Bertsimas, D., Brown, D., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3), 464–501.

Cárdenas, A., Amin, S., and Sastry, S. (2008). Research challenges for the security of control systems. In *Proceedings of the 3rd Conference on Hot Topics in Security*, HOTSEC'08, 6:1–6:6.

Farwell, J. and Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Survival*, 53(1), 23–40.

Fawzi, H., Tabuada, P., and Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6), 1454–1467.

Grewal, M. and Glover, K. (1976). Identifiability of linear and nonlinear dynamical systems. *IEEE Trans. Autom. Control*, 21(6), 833–837.

Gurobi (2015). Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Harirchi, F., Luo, Z., and Ozay, N. (2016). Model (in)validation and fault detection for systems with polynomial state-space models. In *American Control Conference (ACC)*, 1017–1023.

Harirchi, F. and Ozay, N. (2015). Model invalidation for switched affine systems with applications to fault and anomaly detection. *IFAC-PapersOnLine*, 48(27), 260–266.

Harirchi, F. and Ozay, N. (2016). Guaranteed model-based fault detection in cyber-physical systems: A model invalidation approach. `arXiv:1609.05921 [math.OC]`.

Jacobsen, E., Harirchi, F., Yong, S.Z., and Ozay, N. (2017). Optimal input design for affine model discrimination with applications in intention-aware vehicles. *arXiv preprint arXiv:1702.01112*.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*. Taipei, Taiwan. URL `http://users.isy.liu.se/johanl/yalmip`.

Lou, H. and Si, P. (2009). The distinguishability of linear control systems. *Nonlinear Analysis: Hybrid Systems*, 3(1), 21–38.

Mo, Y., Weerakkody, S., and Sinopoli, S. (2015). Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems*, 35(1), 93–109.

Nghiem, T., Pappas, G., and Mangharam, R. (2013). Event-based green scheduling of radiant systems in buildings. In *ACC*, 455–460.

Nikoukhah, R. and Campbell, S. (2006). Auxiliary signal design for active failure detection in uncertain linear systems with a priori information. *Automatica*, 42(2), 219–228.

Pasqualetti, F., Dörfler, F., and Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11), 2715–2729.

Raimondo, D.M., Marseglia, G.R., Braatz, R.D., and Scott, J.K. (2016). Closed-loop input design for guaranteed fault diagnosis using set-valued observers. *Automatica*, 74, 107–117.

Rosa, P. and Silvestre, C. (2011). On the distinguishability of discrete linear time-invariant dynamic systems. In *IEEE CDC-ECC*, 3356–3361.

Scott, J., Findeisen, R., Braatz, R.D., and Raimondo, D. (2014). Input design for guaranteed fault diagnosis using zonotopes. *Automatica*, 50(6), 1580–1589.

Shoukry, Y., Nuzzo, P., Puggelli, A., Sangiovanni-Vincentelli, A.L., Seshia, S.A., and Tabuada, P. (2014). Secure state estimation for cyber physical systems under sensor attacks: a satisfiability modulo theory approach. *arXiv preprint arXiv:1412.4324*.

Tabatabaeipour, S.M. (2015). Active fault detection and isolation of discrete-time linear time-varying systems: a set-membership approach. *International Journal of Systems Science*, 46(11), 1917–1933.

Weerakkody, S. and Sinopoli, B. (2015). Detecting integrity attacks on control systems using a moving target approach. In *IEEE Conference on Decision and Control (CDC)*, 5820–5826.

Yong, S., Zhu, M., and Frazzoli, E. (2015). Resilient state estimation against switching attacks on stochastic cyber-physical systems. In *IEEE Conference on Decision and Control (CDC)*, 5162–5169.