Controllability of Dynamical Systems: Threat Models and Reactive Security

Carlos Barreto[1], Alvaro A. Cárdenas[1], and Nicanor Quijano[2]

[1] Department of Computer Science,
University of Texas at Dallas
[2] Department of Electrical Engineering,
Universidad de Los Andes, Colombia

Abstract. We study controllability and stability properties of dynamical systems when actuator or sensor signals are under attack. We formulate a detailed adversary model that considers different levels of privilege for the attacker such as read and write access to information flows. We then study the impact of these attacks and propose reactive countermeasures based on game theory. In one case-study we use a basic differential game, and in the other case study we introduce a heuristic game for stability.

1 Introduction

The security of cyber-physical control systems has received significant attention in the last couple of years [1,2,3,4,5,6]. In this document we focus on controllability and stability properties of dynamical systems and discuss the theoretical background to analyze how these properties behave under attacks.

The first part of the paper focuses on defining a threat model and risk-assessment analysis based in the theory of linear state space systems and is general enough to be applicable to a wide-range of cyber-physical systems.

The second part of the paper covers reactive security—when the control signal of the defender changes in response to attacks—as a game between a controller and an attacker.

2 System Model

Probably the most general and widely used framework in control systems is the theory of Linear Time Invariant (LTI) state space systems. In this setting we consider a forced (i.e., non-homogeneous) system of linear differential equations:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is a vector of physical quantities representing the state of the system at time t, $u(t) \in \mathbb{R}^p$ is the control input at time t, $y(t) \in \mathbb{R}^q$ is a vector

S.K. Das, C. Nita-Rotaru, and M. Kantarcioglu (Eds.): GameSec 2013, LNCS 8252, pp. 45–64, 2013.

of sensor measurements at time t, and A, B, C and D are matrices representing the dynamics of the system.

Even when the system under control exhibits non-linear dynamics, non-linear systems can usually be approximated by linear systems to study their properties near their region of operation (we will show an example of this linearization in one of the case studies in the paper).

3 Control/Security Properties

Similar to security properties such as confidentiality, integrity, and availability, there are several control properties that a system designer or plant operator would like to maintain, even under attack.

In the theory of linear state space systems, two fundamental (and dual) properties are controllability and observability. In this paper we focus on controllability properties.

Controllability means that the state of the system can be driven to any arbitrary place by using the manipulated variables (i.e., the control input). An LTI system is controllable iff

$$\text{rank}([B \quad AB \quad \cdots \quad A^{n-1}B]) = n \tag{2}$$

Another important property of a control system is **stability**. There are several notions of stability (asymptotic stability, Lyapunov stability, BIBO stability, etc.); however, they all intuitively describe the notion that the state $x(t)$ of the system will converge (or remain relatively close) to a desired set point x^* after disturbances.

Stabilizability is a weaker notion of controllability, and it is satisfied if the uncontrollable modes of the system are stable.

4 Attack Model

In this section we define an attack model for control systems containing three part: goals of the attacker, offline information, and online information.

4.1 Goals of an Attacker

While in a general setting an attacker can have many different objectives, in this paper we focus on attackers that try to manipulate the controllability or stability of the system.

Obtain Control: One goal can be to obtain controllabilty of the system with the minimal attack effort: find the smallest set of controller compromises u_a such that the system becomes fully controllable by the attacker.

Disrupt Control: A weaker objective can be to simply make the system uncontrollable to the defender (even if the system is also uncontrollable by the attacker).

Make the System Unstable: If the control strategy of the defender is fixed, a different objective available to the attacker is to make the system unstable.

4.2 Offline Information Available to the Attacker

Well-informed attackers can create more precise attacks and can determine with confidence the effect of their actions. In this paper we assume the attacker has access to the following information:

System Parameters: Matrices A, B, C, D. Without knowledge of these matrices, attacks will have random effects and the consequences will be unpredictable by the attacker.

Control Algorithm: The attacker knows the output $u(t)$ the controller will give to any sensor values $y(t)$. One simple example is when $y(t) = x(t)$ and $u(t) = Kx(t)$. In this example, if the attacker knows the control algorithm, it means the attacker has knowledge of the matrix K.

4.3 Online Information (and Access) Available to the Attacker

Table 1. Online Capabilities of the Attacker

	Impact	Explanation
Read-Only $y(t)$	The attacker can get information on state of the system. It can estimate the state if the system is observable or partially estimate some modes.	The attacker can eavesdrop on $y(t)$ but cannot send fake $y_a(t)$ values (i.e., it cannot impersonate itself as the controller to the actuator).
Write-Only $y(t)$	The attacker can launch deception (also known as false data injection) attacks to the controller, but without having knowledge of the state of the system.	The attacker can impersonate itself to the controller, but cannot eavesdrop on legitimate sensor readings $y(t)$.
Read-Write $y(t)$	The attacker can try to estimate the state of the system and use that information to launch deception attacks.	The attacker can eavesdrop on $u(t)$ and send false sensor readings $y_a(t)$ to the controller.
Read-Only $u(t)$	If the attacker knows the initial state x_0 and has access to $u(t)$ since time t_0, it can estimate the state of the system.	Attacker can eavesdrop on $u(t)$ but cannot send fake $u_a(t)$ values (i.e., it cannot impersonate itself as the controller to the actuator).
Write-Only $u(t)$	The attacker can manipulate the output of the actuator.	The attacker can impersonate itself to the actuator, but cannot eavesdrop on legitimate $u(t)$ commands.
Read-Write $u(t)$	The attacker can manipulate the output of the actuator and has information of the original intended control signal.	The attacker can eavesdrop on $u(t)$ and send false $u_a(t)$ commands to the actuator.

In this section we propose two tables describing a new attacker model that considers the information attackers have online, and the privilege access they have over the **regulatory control loop**–in this paper we leave out of the scope supervisory, hierarchical, and human-machine interface attacker models.

Table 2. Examples of Attacks. Empty blocks can be considered as combinations of attacks described in the first row and the first column. In practice, an attacker that compromises a PLC can potentially change (depending on the architecture of the PLC) the sensor readings and send them to other PLCs or Human Machine Interfaces. We do not consider this case here as our focus is regulatory control.

	No Acces to $u(t)$	Read-Only $u(t)$	Write-Only $u(t)$	Read-Write $u(t)$
No Access to $y(t)$	No Attacks.	The attacker has physical access to the actuator and can read the analog signals it receives.	The attacker installs its own actuators.	Man-in-the-Middle between PLC and actuator.
Read-Only $y(t)$	The attacker installs its own sensors.			The attacker compromises a PLC.
Write-Only $y(t)$	Attacker uses physical attack (e.g., move a temperature sensor to a refrigerator).		Attacker changes configuration parameters of PLC.	
Read-Write $y(t)$	The attacker obtains the secret keys of the sensors.			

We assume attackers can control a subset of sensors or actuators, but they will have different level of access depending on the model assumed. For example an attacker might be able to get read access to the sensors but not write access; or it can get write-only access to the actuators but not read-access. We think this level of granularity is very important to model precisely what the attackers can do to the system and we argue that this level of granularity has been missing in a lot of the previous work for the security of control systems. The proposed information and privilege-level of attackers during run-time can be seen in Table 1. Examples of when do these assumptions make sense are given in Table 2

5 Attacking Controllability

Using the attacker model defined in the last section, we now turn to the problem of how controllability and stability can be attacked. This analysis can be used for risk assessment by identifying the resiliency of the system to attacks or to identify the actuators and sensors that are most valuable to the system.

5.1 Attacking Controllability with $u(t)$

When an attacker has **Write-Only** or **Read-Write** access to a subset of control signals u_a (it does not matter which), because the ordering of the vector x is arbitrary, we can always partition the system in the following form:

$$\begin{aligned} \dot{x} = &\quad Ax + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_s \\ u_a \end{bmatrix} \\ \dot{x} = &\ Ax + \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} u_s + \begin{bmatrix} B_{12} \\ B_{22} \end{bmatrix} u_a \\ \dot{x} = &\quad Ax + B_s u_s + B_a u_a \end{aligned} \tag{3}$$

where $B_s = [B_{11} \quad B_{21}]^T$, $B_a = [B_{12} \quad B_{22}]$, u_s represents the first s rows of the vector u and u_a the remaining rows (which we assume are under the control of the adversary).

Table 3. Assuming the original system is controllable, we can study the impact to the system when the attacker obtains write-access to the subset of actuators $u_a(t)$. For the third and last rows, the attacker needs to change the parameter of the controller K_a or needs to get access to sensor readings to estimate $x(t)$.

Defender $u_s(t)$	**Attacker** $u_a(t)$	**Attacker Goal**	**Result**
$Kx(t)$	arbitrary	Obtains controllability iff:	$\text{rank}([B_a \quad A_s B_a \quad \cdots \quad A_s^{n-1} B_a]) = n$
arbitrary	0 (DoS)	Disrupts controllability iff:	$\text{rank}([B_s \quad AB_s \quad \cdots \quad A^{n-1} B_s]) < n$
arbitrary	$K_a x(t)$	Disrupts controllability iff:	$\text{rank}([B_s \quad A_a B_s \quad \cdots \quad A_a^{n-1} B_s]) < n$
arbitrary	arbitrary	Maximize cost for defender at minimum effort	Differential game. Section 6
$K_s x(t)$	$K_a x(t)$	Destabilize the system	Heuristic stability game. Section 8

With this model we can now ask questions regarding the vulnerability of the system under attack. We study three cases as described in Table 3. These cases are just a few examples of the type of questions we can ask about the system under attack and are not exhaustive. In the next paragraphs we explain how we obtained the results of Table 3.

Attacker Goal: Obtain Control with Integrity Attacks. A common control strategy is to use state-feedback, where the original control signal (the control signal without attack) looks like $u = Kx$.

If the system is under attack (as described in Eq. (3)), only a portion of these control signals will maintain their integrity, thus:

$$u = \begin{bmatrix} u_s \\ u_a \end{bmatrix} = \begin{bmatrix} K_s x \\ u_a \end{bmatrix} \tag{4}$$

where K_s are the first s rows of the original matrix K.

With state feedback, the dynamical system under attack becomes:

$$\begin{aligned}\dot{x} &= Ax + B_sK_sx + B_au_a\\ \dot{x} &= (A + B_sK_s)x + B_au_a\\ \dot{x} &= \quad A_sx + B_au_a\end{aligned} \tag{5}$$

where $A_s = (A + B_sK_s)$. Therefore an attacker can obtain complete state controllability of the system iff

$$\text{rank}([B_a \quad A_sB_a \quad \cdots \quad A_s^{n-1}B_a]) = n \tag{6}$$

Attacker's Goal: Disrupt Control with Denial-of-Service Attacks. Because this is a set of forced differential equations, a denial-of-service means the forcing function would not be available. Therefore, if the attacker performs a denial-of-service attack on the system defined by Eq. (3), we get $u_a(t) = 0$ and thus the resulting dynamical system is:

$$\dot{x} = Ax + B_su_s \tag{7}$$

In this setting, an attacker can disrupt complete state controllability of the system via DoS attacks iff

$$\text{rank}([B_s \quad AB_s \quad \cdots \quad A^{n-1}B_s]) < n \tag{8}$$

Attacker's Goal: Disrupt Control with a State Feedback Integrity Attack. In some cases the attacker can disrupt controllabilty of the system via simple state feedback attacks with an appropriate attack gain matrix K_a. Replacing this state feedback attack in Eq. (3), we get:

$$\begin{aligned}\dot{x} &= Ax + B_sK_ax + B_su_s\\ \dot{x} &= (A + B_aK_a)x + B_su_s\\ \dot{x} &= A_ax + B_su_s.\end{aligned} \tag{9}$$

In this setting, an attacker can disrupt complete state controllability of the system via state-feedback integrity attacks iff

$$\text{rank}([B_s \quad A_aB_s \quad \cdots \quad A_a^{n-1}B_s]) < n \tag{10}$$

5.2 Attacking Controllability with $y(t)$

In this section we show that when the defender uses state feedback control, the adversary can use sensor measurements to reduce the problem to that of Eq. (3), and therefore, we can reproduce all the results we summarized in Table 3.

In a general LTI setting, the sensor measurements $y(t)$ are a function of the state and the inputs, and to deal with attackers that compromise the integrity of sensors we would need to consider *observability* and the design of *state estimators.* In this paper we make the assumption that $y(t) = x(t)$ (a valid assumption in many practical cases) and we leave the more general problem of attacks to the observability of the system for future work.

As in the previous section, we assume that the defender is using a state-feedback control law, i.e., $u(t) = Kx(t)$. Therefore, if the sensors are not compromised, the evolution of the system will follow the equation:

$$\dot{x} = (A + BK)x \tag{11}$$

However, if some of the sensors are compromised with **Write-Only** or **Read-Write** access to a subset of sensor signals $y_a(t) = x_a(t)$, then the evolution of the dynamical system will change to the following equation:

$$\begin{aligned}
\dot{x} =& \quad Ax + \begin{bmatrix} B_{11}K_{11} & B_{12}K_{12} \\ B_{21}K_{21} & B_{22}K_{22} \end{bmatrix} \begin{bmatrix} x_s \\ x_a \end{bmatrix} \\
\dot{x} =& Ax + \begin{bmatrix} B_{11}K_{11} & 0 \\ B_{21}K_{21} & 0 \end{bmatrix} x + \begin{bmatrix} B_{12}K_{12} \\ B_{22}K_{22} \end{bmatrix} x_a \\
\dot{x} =& \quad Ax + (BK)_s x + (BK)_a x_a \\
\dot{x} =& \quad (A + (BK)_s)x + (BK)_a x_a \\
\dot{x} =& \qquad A_s x + B_a u_a
\end{aligned} \tag{12}$$

Note that Eq. (12) has the same form as Eq. (3), however, in this case

$$A_s = A + \begin{bmatrix} B_{11}K_{11} & 0 \\ B_{21}K_{21} & 0 \end{bmatrix}, \quad B_a = \begin{bmatrix} B_{12}K_{12} \\ B_{22}K_{22} \end{bmatrix}, \quad \text{and} \quad u_a = x_a \tag{13}$$

where the **fake** sensor measurement $x_a(t)$ becomes in practice, the control signal of the attacker $u_a(t)$.

Thus, any controllability question we can make with Eq. (3)—in particular the ones summarized in Table 3—can be reproduced in this new setting by analyzing Eq. (12) with the appropriate matrices.

6 Reactive Security: Differential Games

The primary line of defense for any system are its proactive security mechanisms. Therefore, in practice we must use the threat model to identify the most valuable targets for an adversary and invest in protecting them. In this paper, however, we focus on reactive security mechanisms; that is, we focus on algorithms for responding to attacks.

If an attack is detected, the defender can respond with different actions. Some of the possible responses include reconfiguration of the system, attack isolation, or even a system shutdown (for safety reasons). In this work we are interested in defenses that respond to attacks by changes in their control actions; thus creating a game-theory problem where the actions of the players are the control signals each of them has access to. In particular, we assume that if the system is not under attack, the system will operate with a *vanilla* control signal $u(t)$; however, when the system detects an attack, it changes to a *reactive* control signal $u_s(t)$ to maintain the system under the best possible conditions. This creates a differential game between the defender and the attacker.

The theory of noncooperative differential games considers a general dynamical system

$$\dot{x}(t) = f(t, x, u_d, u_a), \qquad x(0) = x_0 \tag{14}$$

with two (or more in some cases) control signals $u_d(t)$ and $u_a(t)$, each of them controlled by a player in the game, and where each player has a utility function it wants to minimize.

Solutions for a Nash equilibrium in differential games usually consider two types of solutions, (1) open-loop solutions, and (2) closed-loop solutions.

In open-loop solutions the control signals $u_i(t)$ $(i = d, a)$ are independent of the current state of the system $x(t)$. Open-loop solutions can be computed by using Pontryagin maximum principle, which results in a system of ordinary differential equations with two-point boundary value problems.

In closed-loop solutions the control signals $u_i(t, x)$ depend on time, and also on the state of the system $x(t)$. Closed-loop solutions are derived by using the principle of dynamic programming, which results in a system of nonlinear Hamilton-Jacobi partial differential equations. These equations can be ill-posed in general and thus closed-loop solutions are usually considered under Linear-Quadratic (LQ) differential games.

An LQ differential game has linear-dynamics and quadratic utility functions. The dynamical system considered in 2-player LQ games matches Eq. (3):

$$\dot{x}(t) = Ax(t) + B_s u_s(t) + B_a u_a(t) \tag{15}$$

while the utility function (for the finite-time case) has the form:

$$J_i(u_d, u_a) = \int_0^T \left[x^T(t) Q_i(t) x(t) + u_i^T(t) R_{ii} u_i(t) + u_j^T(t) R_{ij} u_j(t) \right] dt + x^T(T) Q_{T,i} X(T)$$

where $i = d, a$ and $j \neq i$. This utility function is a natural extension to the traditional optimal control problem.

6.1 Threat Model and Differential Games Solutions

In this section we use our threat model to analyze solutions to differential games. In particular, we note that open-loop strategies make sense only if an attacker has **Write-Only** $u(t)$ and **Write-Only** $y(t)$. The write-only access to the sensors (and actuators) will prevent an attacker from estimating the state of the system, while allowing the attacker to use its control signal to affect the state of the system. From the defender point of view, since the attacker has write access to

$y(t)$, the defender cannot trust the sensor readings and will turn to open-loop control policies as well.

Now we turn our attention to closed-loop strategies. Recall that for closed-loop control strategies we assume that both players have access to the state of the system and use it for deciding their next control actions. Therefore these strategies make sense for an attacker that has **Read-only** $y(t)$**, Write-only** $u(t)$**:** If the system is observable, or in particular, if $y(t) = x(t)$, then **read-only** $y(t)$ allows the attacker to get access to $x(t)$ but does not allow the attacker to modify the sensor readings. This ensures the defender that the sensor readings are trustworthy and can be used to obtain $x(t)$ accurately.

7 Differential Game Example

We use a recent model for data integrity attacks in demand-response programs for the smart grid [7]. The model considers actuator attacks as an aggregate effect for multi-agent systems that all receive the same input control signal.

The system can be modeled as a scalar differential equation where p denotes the percentage of agents receiving the real pricing signal, and $1 - p$ denotes the percentage of compromised agents that receive a fake u_a the attack signal.

$$\dot{x} = ax + pbu_d + (1-p)bu_a, \quad x(0) = x_0 \tag{16}$$

As discussed before, we consider a game between a defender that wants to minimize a utility penalizing the deviation of x from the steady state 0 and the amount of control (the additional price of electricity u_d).

$$J_d(x, u_d) = \frac{1}{2}\int_0^T [\alpha x^2 + \beta (u_d)^2] dt \tag{17}$$

And an attacker that wants to maximize the state trajectory deviation from the target subject with the minimum amount of effort:

$$J_a(x, u_a) = \frac{1}{2}\int_0^T [-\alpha x^2 + \beta (u_a)^2] dt \tag{18}$$

In general, parameters α and β for the objectives of the attacker and the defender can be different, but we assume they are the same to simplify notation.

We consider open-loop solutions. To find the necessary conditions for optimality of the game (a Nash equilibrium between the two players) we need to use Pontryagin's minimum principle.

First we start by defining the Hamiltonian for the defender:

$$H_d(x, u_d, u_a, \lambda_d) = \frac{1}{2}(\alpha x^2 + \beta (u_d)^2) + \lambda_d (ax + pbu_d + (1-p)bu_a) \tag{19}$$

and the Hamiltonian for the attacker

$$H_a(x, u_d, u_a, \lambda_a) = \frac{1}{2}(-\alpha x^2 + \beta (u_a)^2) + \lambda_a (ax + pbu_d + (1-p)bu_a) \tag{20}$$

The necessary conditions for an optimal solution need to satisfy several constraints. First we find the partial derivative of the Hamiltonian with respect to the control inputs:

$$\beta u_d^* + \lambda_d^* pb = 0$$
$$\beta u_a^* + \lambda_a^*(1-p)b = 0$$

Therefore, the optimal control action by the defender is:

$$u_d^*(t) = -\frac{\lambda_d^*(t)pb}{\beta} \tag{21}$$

$$\tag{22}$$

and the optimal control action by the attacker is

$$u_a^*(t) = -\frac{\lambda_a^*(t)(1-p)b}{\beta} \tag{23}$$

To find the evolution of $\lambda_a^*(t)$ and $\lambda_d^*(t)$ we find the costate equations:

$$\begin{aligned} -\dot{\lambda}_d^* &= \alpha x^* + a\lambda_d^*, \quad \lambda_d^*(T) = 0 \\ \implies -\frac{\dot{\lambda}_d^*}{\alpha} &= x^* + a\frac{\lambda_d^*}{\alpha}, \quad \lambda_d^*(T) = 0 \end{aligned} \tag{24}$$

and

$$\begin{aligned} -\dot{\lambda}_a^* &= -\alpha x^* + a\lambda_a^*, \quad \lambda_a^*(T) = 0 \\ \implies -\left(-\frac{\dot{\lambda}_a^*}{\alpha}\right) &= x^* + a\left(-\frac{\lambda_a^*}{\alpha}\right), \quad \lambda_a^*(T) = 0 \end{aligned} \tag{25}$$

We can simplify the analysis by noting that Eq. (24) and Eq. (25) can be modeled by the following differential equation:

$$-\dot{q} = ax^* + aq, \quad q(T) = 0. \tag{26}$$

Once we solve for $q(t)$ w eknow that $\lambda_d^*(t) = \alpha q(t)$ and $\lambda_a^*(t) = -\alpha q(t)$.

As shown by Bensoussan [8], Eq. (16) and Eq. (26) can be solved explicitly by a decoupling argument, resulting in:

$$x^*(t) = x_0 \frac{s(e^{s(T-t)} + e^{-s(T-t)}) - a(e^{s(T-t)} - e^{-s(T-t)})}{s(e^{sT} + e^{-sT}) - a(e^{sT} - e^{-sT})} \tag{27}$$

where

$$s = \sqrt{a^2 + \frac{\alpha(pb)^2 - \alpha((1-p)b)^2}{\beta}} \tag{28}$$

(as long as s is not a complex number)

$$q(t) = H(t)x^*(t) \tag{29}$$

and

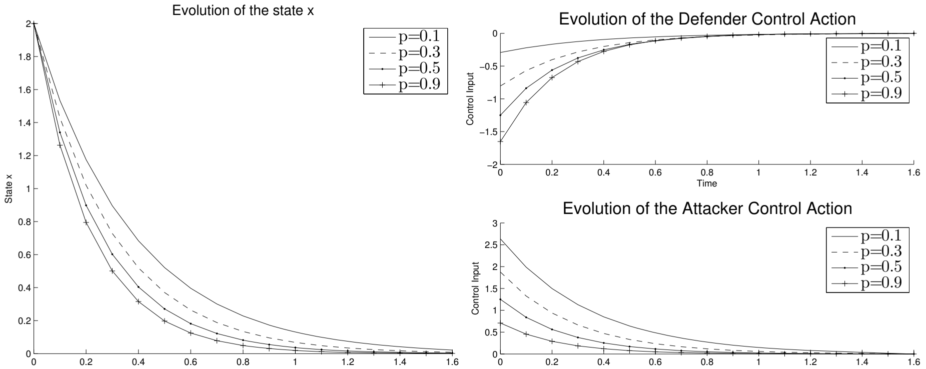
$$\frac{1}{H(t)} = -a + s\frac{e^{2s(T-t)}+1}{e^{2s(T-t)}-1} \tag{30}$$

7.1 Simulation Results

In this section we illustrate the behavior of the open loop differential game described by Eq. (16). Let us consider the system parameters $a = -4$, $b = 1$, $\alpha = 10$, $beta = 1$, $x(0) = 2$. Note that the solution $x^*(t)$ of the system is real for any time $t \in [0, T]$ if $x(0) \in \Re$ and the parameter s (see Eq. (28)) is real for any $p \in [0, 1]$.

The behavior of the system state x, as well as the control actions of each player for different values of p are depicted in Fig. 1. Although the system converges to zero for any value of p, the time of convergence is affected by the value of p. Specifically, when the attacker has control over the majority of the system inputs, i.e., when $1-p > 0.5$, it is able to delay the convergence of the system to the equilibrium. However, the attacker tends to make more effort in its control signal as $1-p$ is increased, i.e., $\int_0^T |u_a(t)|dt$ increases as $1-p$ increases.

The defender experiences a similar behavior. Particularly, if the participation of the defender p is increased, then the system approaches the stable state faster, but with a higher cost in resources for the defender, represented by $\int_0^T |u_d(t)|dt$.



(a) Evolution of the system state (b) Control applied by each player

Fig. 1. Evolution of the differential game in Eq. (16) considering different values of p

While it is clear that in most practical cases, the defender control action has a cost associated with it, in practice, the attacker action should not have these cost constraints. When an attacker compromises a control signal, the player paying for this control action is the defender, not the attacker. However, if we remove the $\beta(u_a)^2$ from the utility function of the attacker we get an ill-posed problem, where u_a does not have any constraints and cannot be find with the maximum principle.

8 Heuristic Stability Game

One of the problems with differential games is that the utility functions of the players usually need to satisfy very specific properties in order to have well-defined solutions. These properties usually limit the general applicability of these games, in particular by placing artificial limitations on what the attacker signal $u_a(t)$ can do.

In this section we discuss a stability game where the goal of the defender is to make the system stable while the goal of the attacker is to make the system unstable. This binary utility function does not allow this system to be formally analyzed for equilibrium strategies; however, our goal is to start exploring the design space to allow more realistic settings that do not impose artificial limitations on what the attack control signal $u_a(t)$ needs to satisfy.

A bioreactor is a system designed to provide some environmental conditions required to carry out a biochemical process. For example, a bioreactor might be used for processing some pharmaceuticals or food that involve the use of microorganisms or substances derived from them. Particularly, processes focused on the growth of organisms (also called biomass) should provide a batch of organisms with food in order to promote the population growth. In such cases, the process is regulated by means of the *substrate feed* (or food income) and the output mass flow (composed by both biomass and substrate), namely the *dilution rate*. In this sense, the state of the system can be described by means of the proportion of both biomass and substrate in the bioreactor, denoted by x_1 and x_2, respectively. The substrate feed and the dilution rate are control variables of the bioreactor that can be used to regulate the biomass production. In this case, the substrate feed and the dilution rate are denoted by x_{2f} and D, respectively. A dynamical model that describes the behavior of the aforementioned bioreactor is

$$\begin{aligned} \dot{x}_1(t) &= (\mu - D)x_1(t), \\ \dot{x}_2(t) &= D(x_{2f} - x_2(t)) - \tfrac{\mu}{Y}x_1(t), \end{aligned} \tag{31}$$

where μ is the average growth rate of the organisms, and Y is the substrate consumption rate. The growth rate of the population is influenced by the amount of food available in the environment; however, the population growth might not increase indefinitely with the substrate concentration. On the contrary, excess of food would induce inhibitory effects on the population growth. This behavior is modeled by means of the growth rate

$$\mu = \mu_{max}\left(\frac{x_2}{k_m + x_2 + k_1 x_2^2}\right), \tag{32}$$

where k_m and k_1 are constants. On the other hand, the substrate consumption rate is defined as ratio of the change in the population mass and the change in the substrate mass inside the bioreactor, that is $Y := -\frac{\dot{x}_1}{\dot{x}_2}$.

Table 4. Rest points of the system in Eq. (31)

x_1^*	x_2^*	
0	4	Stable
0.95103	1.512243	Unstable
1.530163	0.174593	Stable

Now, let us consider a process with the following parameters [9]: $\mu_{max} = 0.53hr^{-1}$, $k_m = 0.12$ g/liter, $k_1 = 0.4545$ liter/g, $Y = 0.4$. If we use constant inputs $D = 0.3$ and $x_{2f} = 4$, then the system in Eq. (31) is characterized by three rest points shown in Table 4. In this case, we assume that the system designer wants to stabilize the system at the unstable equilibrium point $x^* = [0.95103, 1.512243]^\top$. In particular, the control law is designed by means of state feedback. Accordingly, the design procedure involves 1) the linearization of the system around the desired equilibrium point, and 2) the design of the control law that stabilizes the system. Later, the vulnerabilities of the system are going to be analyzed.

8.1 Linearization and Control Design

The linear model of the system in Eq. (31) at the unstable equilibrium point is

$$\dot{z} = Az + Bu, \tag{33}$$

where $z = x - x^*$, $u = [u_1, u_2]^\top = [D, x_{2f}]^\top - [0.3, 4]^\top$, and

$$A = \begin{bmatrix} 0 & -0.068 \\ -0.75 & -0.13 \end{bmatrix}, \quad B = \begin{bmatrix} -0.994 & 0 \\ 2.488 & 0.3 \end{bmatrix}.$$

Now, using a state feedback control of the form $u = -Fz$, where $F = [F_1, F_2]^\top$ is a matrix in $\Re^{2\times 2}$ and F_1 and F_2 are vectors in $\Re^{2\times 1}$, we have

$$\dot{z} = (A - BF)z. \tag{34}$$

The previous expression can be rewritten in terms of each control input as

$$\dot{z} = (A - B_1 F_1^\top - B_2 F_2^\top)z, \tag{35}$$

where B_1 and B_2 are the columns of B that determine the influence that each input has in the system and $F_i^\top$ is the feedback control law at the i^{th} input, for $i \in \{1, 2\}$. In this case, the pair (A, B) is controllable, as well as $(A, B1)$ and $(A, B2)$.

In particular, we consider a situation in which the designer fixes the j^{th} input: u_j to a constant value, while it controls the i^{th} input, denoted by u_i: i.e., the designer only controls $u_i = -F_i^\top z$, resulting

$$\dot{z} = (A - B_i F_i^\top)z + B_j u_j. \tag{36}$$

The selection of F_i should guarantee that the system is stable. According to the theorem of pole shifting, when the system is controllable, it is possible to find a matrix F_i such that the system in Eq. (36) is globally asymptotically stable [10]. This is achieved if the eigenvalues λ_h of $A - B_iF_i$ have negative real part.

Particularly, the design of the feedback control law can be designed by applying a coordinate transformation of the form $\tilde{z} = P_i z$. This linear transformation let us express the system in Eq. (36) in terms of the matrices in canonical form

$$A^\dagger = \begin{bmatrix} 0 & -\alpha_0 \\ 1 & -\alpha_1 \end{bmatrix}, \quad B_i^\dagger = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, F_i^\dagger = [f_1, f_0],$$

where $A^\dagger = P_i^{-1}AP_i$, $B_i^\dagger = P_i^{-1}B_i$, $F_i^\dagger = F_iP_i$. On the other hand, α_0 and α_1 are the coefficients of the characteristic polynomial of A, i.e., $det(sI - A) = s^2 + \alpha_1 s + \alpha_0$ [10]. Particularly, in linear systems with one input, P_i is the controllability matrix of (A, B_i). Without loss of generality, we assume that the feedback F_i is designed to obtain poles λ_1 and λ_2 with negative real part , i.e., $\mathbf{Re}(\lambda_h) < 0$, for $h \in \{1, 2\}$. Therefore, a valid feedback law F_i must satisfy

$$\begin{aligned} det\Big(sI - (A - B_iF_i)\Big) &= s^2 + (\alpha_1 + f_1)s + (\alpha_0 + f_0) \\ &= (s - \lambda_1)(s - \lambda_2). \end{aligned} \tag{37}$$

Note that the coefficients α_0 and α_1 are positive, since λ_1 and λ_2 have negative real part.

Under these conditions, the designer is able to stabilize the system through the feedback law F_i. The design of the feedback vector F_i can be made minimizing a cost criteria, by means of the Linear Quadratic Regulator (LQR) problem (which has a cost-function similar to the one considered in LQ differential games). Since the feedback control is only applied in one of the inputs, we have that $F = [F_i, \mathbf{0}]^\top$, where $\mathbf{0}$ is a vector of zeros.

8.2 Attacker Perspective

Now, let us consider a situation in which an attacker obtains control over the j^{th} input u_j. In this case, we assume that the attacker is able to implement a control law $u_j = -F_j z$ that seeks to destabilize the system. Note that the attacker might succeed on its purpose, since (A, B_j) is controllable, for $j \in \{1, 2\}$.

It is important to note that if the attacker is able to observe the state of the system, then it would design the feedback vector F_j, such that the system in Eq. (35) is unstable. Since the defender trusts $x(t)$, we assume the attacker has **read-only** access to the sensors $y(t) = x(t)$.

In this sense, applying the similarity matrix P_j, we obain that the matrices in canonical form $A^\ddagger = P_j^{-1}(A - B_iF_i)P_j$, $B^\ddagger = P_j^{-1}B_j$, and $F = F^\ddagger P_j^{-1}$ that satisfy

$$A^\ddagger = \begin{bmatrix} 0 & -\alpha_0 - f_0 \\ 1 & -\alpha_1 - f1 \end{bmatrix}, B_j^\ddagger = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where P_j is the controllability matrix of $(A - B_iF_i, B_j)$. In this case, the characteristic polynomial of $(A - B_iF_i)$ is given by Eq. (37). Now, the attacker can break the system by modifying the coefficients of the characteristic polynomial through F_j. In particular, if we consider $F_j = F^{\ddagger}P_j^{-1}$, where $F^{\ddagger} = [g_1, g_0]$, then the characteristic polynomial is transformed into

$$det\Big(sI - (A - B_iF_i - B_jF_j)\Big) = s^2 + (\alpha_1 + f_1 + g_1)s + (\alpha_0 + f_0 + g_0). \quad (38)$$

For example, let us fix $g_1 = 0$ and set $g_0 = -(\alpha_0 + f_1 + \delta)$, with $\delta > 0$. Consequently, the characteristic polynomial of Eq. (38) becomes

$$det\Big(sI - (A - B_iF_i - B_jF_j)\Big) = s^2 + (\alpha_1 + f_1)s + -\delta.$$

Since there is a change of sign in the coefficients of the characteristic polynomial, the system in unstable under the attacker feedback law

$$F_j = [0, -(\alpha_0 + \delta)]P_j^{-1}. \quad (39)$$

In particular, the system has one unstable equilibrium point. Note that if the attacker knows the eigenvalues of the system, then it can calculate the value of $\alpha_0 + f_0$ by meas of

$$\alpha_0 + f_0 = \prod_{i=1}^{n} \lambda_h. \quad (40)$$

This relation can be extracted from the expanded characteristic polynomial in terms of the eigenvalues λ_h, for $h \in \{1, \dots, n\}$.

8.3 System Defense

We are interested in analyzing the actions that the system designer can take in order to stabilize the system when an attacker influences u_j. Intuitively, the designer would want to implement a feedback $\hat{F}_i$ that cancels the feedback law of the attacker F_j. Therefore, $\hat{F}_i$ must satisfy

$$(A - B[\hat{F}_i, F_j]^{\top})x = (A - B[F_i, \mathbf{0}]^{\top})x.$$

This can be rewritten as

$$B([\hat{F}_i, \mathbf{0}]^{\top} + [\mathbf{0}, F_j]^{\top}) = B[F_i, \mathbf{0}]^{\top}.$$

However, if $F_j \neq \mathbf{0}$, then it is not possible to find a feedback law $\hat{F}_i$ that satisfies the previous equality. This happens because the columns of $[\hat{F}_i, \mathbf{0}]^{\top}$ are linear independent from the columns of $[\mathbf{0}, F_j]^{\top}$.

Therefore, the designer must take actions that compensate, rather cancel, the attacks. In this sense, the designer would implement the feedback law $\hat{F}_1$ that shifts the poles of the attacked system to the negative imaginary semi plane. This can be done by repeating the feedback design procedure exposed above, having into account a system of the form

$$\dot{z} = (\hat{A} - B_i \hat{F}_i)z,$$

where $\hat{A} = A - B_j F_j$ is the observed system by the defender. $\hat{A}$ can be considered by the designer as a given of the control design problem. Note that the response of the designer is subject to 1) the knowledge that the system is attacked and 2) the knowledge of the effects of the attack on the system.

8.4 Simulations

In this section we analyze the defense and attack actions of two agents that seek to stabilize and destabilize a system.

We consider the case in which both defender and attacker update their feedback gains according to the actions the opponent. This can be seen as a repeated game, in which the players are the defender and the attacker. We assume that each player requires some constant time to update its action as a response to the move by the other player. Therefore, the game is repeated after a period T. Furthermore, the speed of response of the defender to attacks is measured in terms a time fraction of T, namely $DC \in [0, 1]$ (for duty-cycle). In this sense, DC indicates that in each period T, the defender makes the system stable during the interval $[0, DCT]$ and the attacker disrupt the system during (DCT, T).

Simulations are made assuming that agents have knowledge about the actions of each other. In particular, the defender designs the feedback law F_i using a Linear-quadratic (LQ) state-feedback regulator that minimizes the cost function

$$J = \int_0^\infty \left(x^\top Q x + u^\top R u\right) dt,$$

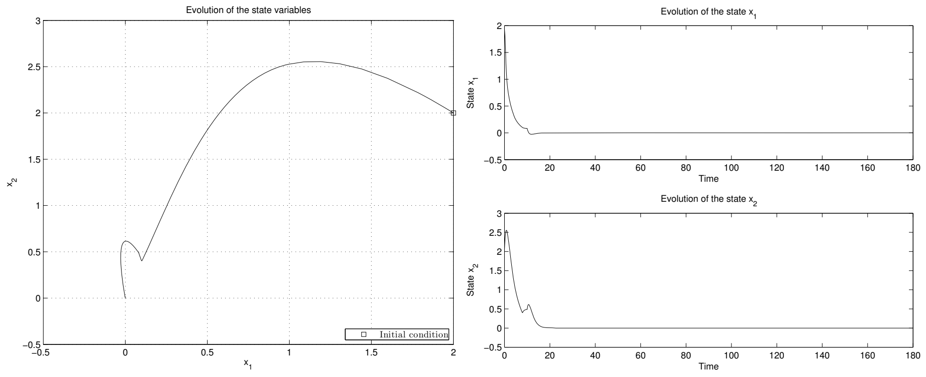
where

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

On the other hand, the attacker actions are calculated according to Eq. (39) and (40).

Control of the Dilution Rate. First, consider the situation in which the defender manipulates the dilution rate D and the attacker controls the substrate feed x_{2f}. In this case, the natural behavior of the system under the influence of the defender's control signal is to reach the origin and remain there. However, the system becomes unstable if the attacker manipulates the substrate feed input and there is no response by the system designer. It can be seen that the system become unstable after the attack, that takes place at $t = 10s$.

Fig. 2 and 3 show the evolution of the game for $DC = 0.8$ and $DC = 0.1$, respectively. Note that although the attacker actions tend to affect the system, the defender is able to stabilize the system, even with a slow reaction to the attacker actions, i.e., the case with $DC = 0.1$.



(a) State space evolution

(b) Evolution of the system states

Fig. 2. Evolution of the system when both attacker and defender compete by turns. $DC = 0.8$.

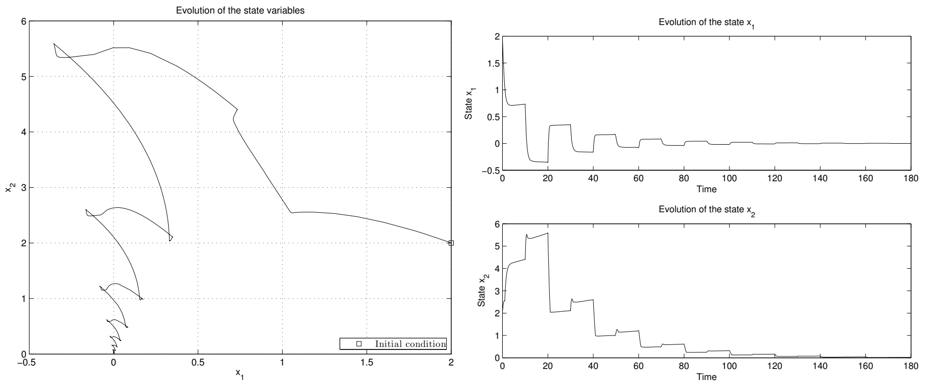
Control of the Substrate Feed. Here we consider the evolution of the system states when both defender and attacker play a game of control by manipulating the substrate feed and the dilution rate, respectively. It can be seen that an attack on the dilution rate is able to make unstable the system, with a notable effect on the biomass level x_1, with respect to the previous scenario.

Fig. 4 and 5 show the evolution of the game for $DC = 0.8$ and $DC = 0.5$, respectively. It can be seen that the defender is able to stabilize the system for $DC = 0.8$. However, the defender requires a lower reaction time to control the system, in contrast to the case when it controls the dilution rate. Specifically, when the defender controls the dilution rate (scenario 1), it is able to stabilize the system with a DC of 0.1 (see Fig. 3). Now, when the defender controls the substrate feed, it is not able to stabilize the system with a DC of 0.5.

Stability Experiments. Now we analyze the system behavior as a function of the parameter DC for a particular period T. We present numerical experiments to observe properties of the system.

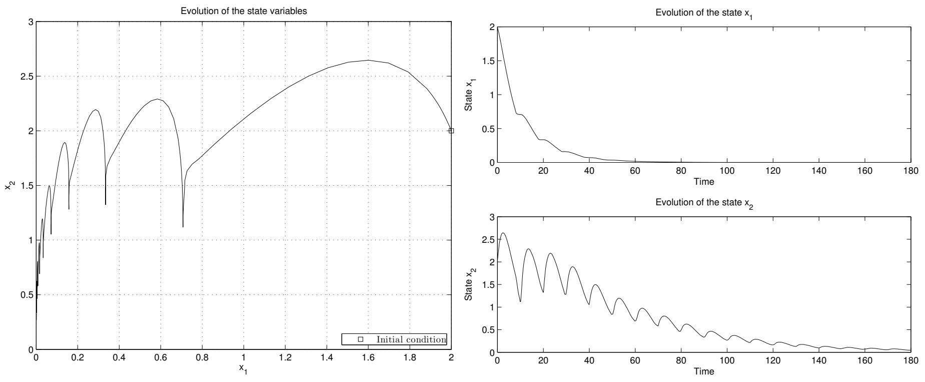
Since the control law has jumps each time a player updates its strategy, obtaining explicit stability solutions is difficult (although in future work we plan to use the theory of hybrid systems to better characterize the stability of these systems). We are interested in the stability of the output variable defined as $y = x_1$, i.e., the stability of the biomass concentration. Specifically, our analysis is made by approximating an exponential function of the form $h(t) = e^{\sigma t}$ to the output $y(t)$. If the parameter σ is positive, then we conclude the system is unstable for a particular DC. On the other hand, if the system is stable, the variable $y(t)$ would be approximated by means of a decreasing exponential function.

In particular, scenario 1, in which the defender and the attacker control the dilution rate and the substrate feed, with $T = 100$ is stable for any DC in the interval $[0.0120, 1]$. On the other hand, in scenario 2 the attacker can destabilize



(a) State space evolution (b) Evolution of the system states

Fig. 3. Evolution of the system when both attacker and defender compete by turns. $DC = 0.1$.

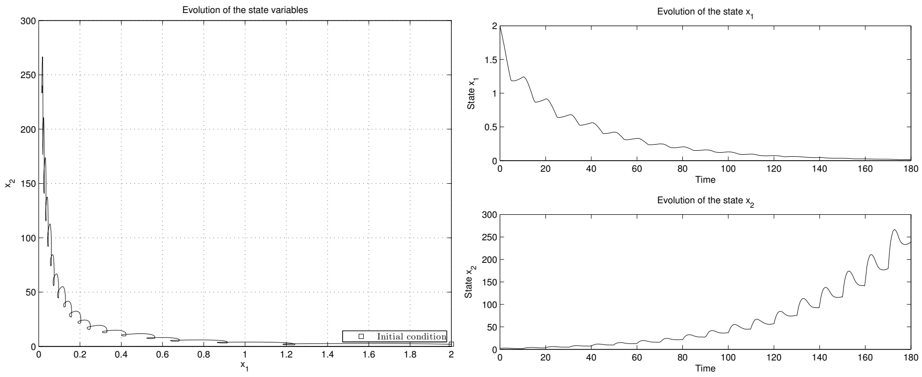


(a) State space evolution (b) Evolution of the system states

Fig. 4. Evolution of the system when both attacker and defender compete by turns. $DC = 0.8$.

the system with a DC in the interval $[0.1, 0.32]$ and $T = 10$. The dependence of the stability of $y(t)$ with respect to DC for the scenario 2 is shown in the Fig. 6.

Note that in scenario 1, the defender is able to stabilize the system for very low reaction times, with respect to the scenario 2. This implies that in scenario 1 the attacks have to be effective for a larger period of time to destabilize the system. Hence, the control using the dilution rate is more robust to attacks than the control using the substrate feed. In this case, the measurement of the reaction time is relative to the period T.



(a) State space evolution

(b) Evolution of the system states

Fig. 5. Evolution of the system when both attacker and defender compete by turns. $DC = 0.5$.

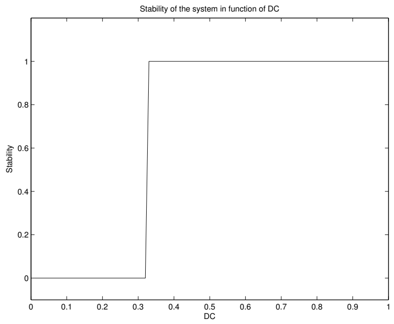


Fig. 6. Stability of the repeated game as a function of DC for scenario 2

9 Future Work

In this paper we have formulated new threat models for controllability and stability of dynamical systems and discussed some ideas on how to model reactive security games between a defender and an attacker. There are many open problems and several directions for future research.

As mentioned in the last section, one particular improvement that can be done to our analysis of the heuristic game is to leverage the theory of hybrid systems to analyze the stability of the game. Hybrid systems can also be used for the study of reachability, which is analogous to the problem of controllability in LTI systems. Computational reachability analysis of systems might be a good tool for analyzing more realistic control problems with bounded states or control actions.

Similarly the concept of proactive security needs further study. An intuitive idea for selecting the most valuable actuators is to consider the following problem: let $B = [\mathbf{b}_1 \ \mathbf{b}_2 \ \cdots \ \mathbf{b}_m]$, then for every input u_i there is a column vector $\mathbf{b}_i \in \mathbb{R}^n$ that uniquely defines how the actuator will affect the physical state of the system. To find the actuator that has the ability to control more states (and thus the most valuable target for the attacker) we can perform the following analysis: $\arg\max_i rank \left[\mathbf{b}_i \ A\mathbf{b}_i \ ... \ A^{n-1}\mathbf{b}_i\right]$. The defender can then invest more in protecting the information flow of the actuators depending on the ranking of each actuator.

Acknowledgments. This research was supported by The MITRE Corporation as part of an independent research project. The authors wish to thank Roshan K. Thomas and Adam L. Hahn of MITRE for their insights.

References

1. Amin, S., Schwartz, G.A., Shankar Sastry, S.: Security of interdependent and identical networked control systems. Automatica (2012)
2. Mo, Y., Sinopoli, B.: False data injection attacks in control systems. In: Preprints of the 1st Workshop on Secure Control Systems (2010)
3. Pasqualetti, F.: Secure control systems: A control-theoretic approach to cyber-physical security. Ph.D. dissertation, University of California (2012)
4. LeBlanc, H.J., Zhang, H., Sundaram, S., Koutsoukos, X.: Consensus of multi-agent networks in the presence of adversaries using only local information. In: Proceedings of the 1st International Conference on High Confidence Networked Systems, pp. 1–10. ACM (2012)
5. Shoukry, Y., Araujo, J., Tabuada, P., Srivastava, M., Johansson, K.H.: Minimax control for cyber-physical systems under network packet scheduling attacks. In: Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems, pp. 93–100. ACM (2013)
6. Teixeira, A., Shames, I., Sandberg, H., Johansson, K.H.: Revealing stealthy attacks in control systems. In: 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1806–1813. IEEE (2012)
7. Tan, R., Krishna, V.B., Yau, D.K., Kalbarczyk, Z.: Impact of integrity attacks on real-time pricing in smart grids. In: ACM Computer and Communications Security (CCS) (2013)
8. Bensoussan, A.: Explicit solutions of linear quadratic differential games. In: Stochastic Processes, Optimization, and Control Theory: Applications in Financial Engineering, Queueing Networks, and Manufacturing Systems, pp. 19–34. Springer (2006)
9. Bequette, B.W.: Process Control: Modeling, Design and Simulation, vol. 6. Prentice Hall, Upper Saddle River (2003)
10. Sontag, E.: Mathematical Control Theory. In: Deterministic Finite-Dimensional Systems, 2nd edn. Texts in Applied Mathematics, vol. 6, Springer, New York (1998)