Cross-Domain Noise Impact Evaluation for Black Box 1
Two-Level Control CPS

Feng Tan, Liansheng Liu, Stefan Winter, Qixin Wang, Neeraj Suri, Lei Bu, Yu Peng, Xue Liu, Xiyuan Peng

ACM Transactions on Cyber-Physical Systems 2019

Most of the papers available from this document appear in print, and the corresponding copyright is held by the publisher. While the papers can be used for personal use, redistribution or reprinting for commercial purposes is prohibited.
Cross-Domain Noise Impact Evaluation for Black Box Two-Level Control CPS

FENG TAN and LIANSHENG LIU, Department of Computing, The Hong Kong Polytechnic University
STEFAN WINTER, Department of Computer Science, TU Darmstadt
QIXIN WANG, Department of Computing, The Hong Kong Polytechnic University
NEERAJ SURI, Department of Computer Science, TU Darmstadt
LEI BU, State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
YU PENG, Department of Automatic Test and Control, Harbin Institute of Technology
XUE LIU, School of Computer Science, McGill University
XIYUAN PENG, Department of Automatic Test and Control, Harbin Institute of Technology

Control Cyber-Physical Systems (CPSs) constitute a major category of CPS. In control CPSs, in addition to the well-studied noises within the physical subsystem, we are interested in evaluating the impact of cross-domain noise: the noise that comes from the physical subsystem, propagates through the cyber subsystem, and goes back to the physical subsystem. Impact of cross-domain noise is hard to evaluate when the cyber subsystem is a black box, which cannot be explicitly modeled. To address this challenge, this article focuses on the two-level control CPS, a widely adopted control CPS architecture, and proposes an emulation-based evaluation methodology framework. The framework uses hybrid model reachability to quantify the cross-domain noise impact, and exploits Lyapunov stability theories to reduce the evaluation benchmark size. We validated the effectiveness and efficiency of our proposed framework on a representative control CPS testbed. Particularly, 24.1% of evaluation effort is saved using the proposed benchmark shrinking technology.

CCS Concepts: • Computer systems organization → Embedded and cyber-physical systems; • Software and its engineering → Software testing and debugging.
1 INTRODUCTION

Cyber-Physical Systems (CPSs) (Sha et al. 2008) converge the discrete computing and continuous physical domains. One representative category of CPSs is control CPSs, where computer systems control physical objects in real time. Naturally, control CPSs demand integration of computer science and control theories.

This article focuses on one aspect of the integration: how to evaluate the impact of cross-domain noises in control CPSs. Specifically, this article assumes a classic control CPS architecture described by Figure 1. It consists of a “physical” control subsystem (simplified as the “physical subsystem”1 in the following) and a “cyber” computing subsystem (simplified as the “cyber subsystem” in the following). The physical and cyber subsystems form a two-level control loop. The physical subsystem conducts the inner control loop, which carries out fine-time-grain sensing (the “local sensing” in the figure) and actuation of the plant (i.e., the physical object being controlled). The cyber subsystem conducts the outer control loop, which carries out coarse-time-grain reference point updates. For simplicity, in the following, this article calls the control CPS architecture of Figure 1 the two-level control CPS (2L-CCPS) architecture.

More specifically, in Figure 1, the dashed box delineates the physical subsystem, which is the same as a conventional non-CPS control system. The external input to the physical subsystem is the reference point value, a vector that specifies the target state of the plant. Given the reference point value, the physical subsystem takes charge of maneuvering the plant until the plant’s state reaches the reference point value. For example, suppose the plant is a cart, with vector $(x_1, x_2)^T$ as its state, where $x_1$ is the cart’s current location and $x_2$ is the cart’s current velocity. A reference point value of $(10, 0)^T$ commands the physical subsystem to move the cart to location 10 and stop there.

Besides the physical subsystem, the dash-dot box in Figure 1 delineates the cyber subsystem. Specifically, the cyber subsystem is a set of interconnected digital modules (can be both software and/or hardware; e.g., digital signal processors). These digital modules collaboratively carry out a workflow that remotely senses the plant state (see $M_{s}$ in Figure 1), processes the sensed state, and decides the new reference point value. The new reference point value is the output (see $M_{d}$ in Figure 1) of the cyber subsystem, and is fed back to the physical subsystem.

The reference point update events take place in coarse-time-grain: they happen discretely and are separated by long time intervals. In contrast, the local sensing and controller actuation in

\[^{1}\text{Note the term “physical subsystem” is a notational convenience. Strictly speaking, it refers to the low-level control system (aka “inner control loop”), which may or may not be purely analog. For example, when a ground computer (i.e., the “cyber subsystem”) uses analog wireless signals to remotely control a purely analog (consider mechanical is a kind of analog) drone, the “physical subsystem” (i.e., the drone) is purely analogue. However, when the ground computer uses WiFi to remotely control a WiFi+analog drone, the “physical subsystem” (i.e., the drone) is indeed a mixture of digital and analog parts.}\]
Cross-Domain Noise Impact Evaluation for Black Box Two-Level Control CPS

Fig. 1. 2L-CCPS, a classic control CPS architecture. Note that the cyber subsystem digital modules can be interconnected via local or remote function calls.

the physical subsystem (i.e., the inner control loop) take place in fine-time-grain. They run in continuous time, or periodically with a sufficiently small period.

For example, for a 2L-CCPS to remotely fly a drone, the drone (the physical subsystem) has its onboard fine-time-grain sensing and actuation for attitude control; whereas, the ground station (the cyber subsystem) uses visuals to conduct remote coarse-time-grain sensing of the drone, and to command the drone where to go. In the following, unless otherwise denoted, the “sensing” of this article refers to the latter, i.e., the coarse-time-grain remote sensing for computing new reference point values by the cyber subsystem.

In practice, sensed signals are always accompanied with noises. These noises constitute a major source of errors. Noises within conventional control systems (e.g., the physical subsystem of a 2L-CCPS) include local sensing noises, controller output disturbances, and plant modeling errors. They are well-studied and can be well contained (Hovakimyan and Cao 2010). Hence, these noises are not the focus of this article. Instead, this article focuses on the noise that crosses the boundaries between the cyber and physical subsystems, i.e., the so-called cross-domain noise. Specifically, in a 2L-CCPS, cross-domain noise (see Fig. 1) refers to the noise that arises from the remote sensing (see module $M_{rs}$ in Fig. 1) of the plant. It propagates through the cyber subsystem, and goes back to the physical subsystem as the error component of the new reference point value.

**Challenge and Overall Idea of the Proposed Solution Framework.** In a conventional control system, noises (i.e., sensing noises, controller output disturbances, and plant modeling errors) propagate through the sensing, controller, and plant module, which can all be modeled by closed-form formulas. Correspondingly, the impacts of the noises can be analytically evaluated. In contrast, the cross-domain noise in a 2L-CCPS propagates through the discrete cyber subsystem (see Fig. 1), which cannot be modeled by closed-form formulas in general. The situation is worse when the cyber subsystem is black box: e.g., when the cyber subsystem is encapsulated by a third party vendor.

---

2According to Franklin et al. (1994), when replacing an analog controller with a discrete controller, we can empirically regard the discrete controller as an analog controller, if the sampling rate is faster than 20 times the closed-loop bandwidth of the analog physical subsystem.
To address the challenge on how to evaluate cross-domain noise’s impacts, this article aims to make an initial step forward: we propose a methodology framework to evaluate the impacts of the cross-domain noise in a 2L-CCPS with a black box cyber subsystem. The overall idea of our framework is as follows.

We first prepare a benchmark, i.e., a set of sample states of the plant. For each sample state of the benchmark, we carry out Monte Carlo emulation. In each emulation trial, the benchmark sample state, plus the cross-domain noise, are entered into the cyber subsystem. The cyber subsystem then outputs the (noisy) next reference point value, which is fed across the domain boundary into a physical subsystem simulator to measure the accident risk. Via the above Monte Carlo emulation, we establish a quantitative relationship between the cross-domain noise level and the plant accident risk increase. This relationship becomes a metric to evaluate the impact of the cross-domain noise. We further propose a control theory based method to shrink the benchmark size, to make our evaluation more efficient.

Contributions and Basic insights. In a more general sense, our proposed framework addresses a subproblem of fault propagation profiling, a hot topic in system dependability research. Compared to existing fault propagation profiling works, our cyber subsystem model is a black box to the users; our physical subsystem model is at the granularity of differential equation level; we extensively exploit interdisciplinary control theory; and we focus on evaluating cross-domain noises unique to CPSs.

The framework is also related to control CPS fault diagnosis and fault tolerance. Compared to existing control CPS fault diagnosis/tolerance works, our cyber subsystem model is a black box to the users, hence the cyber subsystem does not have an accurate model. In addition, we are neither focusing on fault diagnosis (the cause of fault is known, i.e., cross-domain noise), nor on fault tolerance.

Main contributions and insights of this article are summarized as follows.

1. We propose a benchmark metric and corresponding measurement method to evaluate cross-domain noise impacts to 2L-CCPS with black box cyber subsystems.
2. We further propose a method to effectively shrink the benchmark, exploiting the interdisciplinary Lyapunov stability control theories.
3. We validated the effectiveness and efficiency of our proposed methodology framework on a representative 2L-CCPS testbed. Particularly, the benchmark shrinking technology reduces 24.1% of the evaluation effort.

Article Organization. The rest of the article is organized as follows. Section 2 discusses related work. Section 3 describes the overall systems model to set the context for discussion. Section 4 elaborates our basic cross-domain noise impact evaluation method. Section 5 proposes a method to effectively shrink the evaluation benchmark. Section 6 demonstrates and validates the proposed methodology framework. Section 7 concludes the article.

---

3Again, use the aforementioned remotely flying drone example. In each Monte Carlo trial, the benchmark sample can be a video frame (i.e., a photo) of the remote drone and its nearby obstacles. The video frame plus additive white Gaussian noise (i.e., the cross-domain noise) is inputted into the ground station (i.e., the cyber subsystem). This mimics the fact that the ground station’s video camera is noisy. Then the ground station conducts computer vision recognition and decision making as a black box. The decision, i.e., the outputted new reference point on where to fly the remote drone, is fed back to a drone simulator, which simulates the next step physical trajectory of the drone. Expectedly, with higher additive white Gaussian noise, the ground station would more likely make wrong decisions, and the simulated drone trajectory will have a higher probability of hitting the obstacles. By carrying out many randomized trials of such, we will establish the quantitative relationship between the additive white Gaussian noise level and the obstacle-hitting probability.
2 RELATED WORK

In a more general sense, this article addresses a sub-problem of fault propagation profiling, a hot topic in system dependability research. Works of Hiller et al. (2004) propose using conditional probability to profile the permeability, exposure, and impact of faults in a network of software modules. Oliner and Aiken (2011) propose using principal component analysis and temporal correlations to discover influence relationships between software modules, to profile anomaly propagation. Distefano et al. (2011) propose a compositional calculus to analyze software fault propagation with closed-form formulas. Jhumka and Leeke (2011) use software fault propagation profiling results to guide the placement of fault detector assertions. Pham et al. (2015) propose a UML based annotation and inference framework to analyze concurrent fault propagations in component based software systems. However, all the above works focus on pure software system, rather than CPS.

There are works on profiling CPS fault propagation. Sierla et al. (2013) study CPS fault propagation with an explicit object-oriented and event based model. Ge et al. (2009) analyze CPS failure probability using the PRISM (Kwiatkowska et al. 2002) probabilistic model checker. There are also works on using various artificial intelligence and/or statistics tools to quantify CPS fault propagation (Augustine et al. 2012). However, the above works all assume a white box cyber subsystem, or at least a cyber subsystem where the interconnection details of digital modules are known to the user.

As cross-domain noise impact evaluation is a subtask of holistic system analysis, the solution proposed by this article can be plugged into holistic system analysis frameworks, such as FMEA or FMECA (US Dept. of the Army 2015). For example, for FMEA, our impact evaluation results can serve as a system failure rate input related to cross-domain noise.

This article is also related to fault-tolerant control CPS. Conventional fault-tolerant control CPS works deal with sensing errors, actuation errors, system parameter errors, or even system model changes. They typically require white box models of the cyber subsystem (Gao et al. 2015a). Research on fault-tolerant control CPS with black box cyber subsystems is relatively young. There are works on using redundancy to deal with faults in such control CPS (Wang et al. 2013). Such topic is apparently orthogonal to this article’s topic.

Model predictive control (Camacho and Bordons 2013) focuses on repeatedly deriving optimal control signals to control the plant. This article, however, is not focusing on how to control the plant.

There are also works on using data mining, machine learning, and/or inference to diagnose the cause of faults (Gao et al. 2015b). In contrast, our article is not about diagnosis. The cause of fault is given: the cross-domain noise. We want to evaluate its impact on the physical subsystem given different noise levels and various initial plant states. On the other hand, our evaluation results can serve as a training set for data mining, machine learning, or as the prerequisite conditional probability distribution needed by Bayesian inference. In this sense, this article’s work complements the diagnosis works.

The work in Tan et al. (2014) proposes using a Bayesian network for cross-domain noise profiling in control CPS. However, it is a one-page work-in-progress abstract and its proposed methodology may not be valid when noise is non-Gaussian.

The work of this paper is included in the first author’s PhD thesis.

3 OVERALL SYSTEMS MODELS

We shall first set the context for our discussion by introducing the overall systems model. This includes the physical and cyber aspects of the 2L-CCPS architecture, and the combined systems model.
3.1 Physical Subsystem Model

In this article, we assume the physical subsystem of a 2L-CCPS is a Linear Time Invariant (LTI) control system, which is arguably the most widely used control system.

For an LTI control system, the state of the plant at time \( t \) is described by an \( n \)-dimensional vector \( X(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \). The vector is also called the plant’s state vector (in the following, we use the term “plant’s state” and “plant’s state vector” interchangeably), and each element of the state vector is also called a state variable. For simplicity, we often omit the parameter \( t \) when writing state vector and/or variables, and use \( \dot{X} \) (and \( \dot{x}_i, i = 1, \ldots, n \), respectively) to denote the derivative \( \frac{dX}{dt} \) (and \( \frac{dx_i}{dt}, i = 1, \ldots, n \), respectively).

The dynamics of the plant is governed by the following systems of differential equations.

\[
\frac{d(X - O_{ref})}{dt} = AX - O_{ref} + BU,
\]

\[
U = -K(X - O_{ref}),
\]

where \( O_{ref} \in \mathbb{R}^n \) is the reference point value from the cyber subsystem: the objective of control is to maneuver the plant state vector \( X \) to \( O_{ref} \) (so that \( X - O_{ref} = 0 \)); \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) are two constant matrices dependent on the plant’s physics; \( U(t) = (u_1(t), u_2(t), \ldots, u_m(t))^T \) is the controller output created as per Equation (2); \( K \in \mathbb{R}^{m \times n} \) is a constant matrix defining the control strategy. Denote \( \bar{X} \overset{\text{def}}{=} X - O_{ref} \); the system of Equations (1), (2) can be rewritten into the following form.

\[
\dot{\bar{X}} = F\bar{X},
\]

where \( F = A - BK \).

Besides the above systems of differential equations, the dynamics of the plant are also governed by allowed region \( \mathcal{A} \subseteq \mathbb{R}^n \) (or equivalently, forbidden region \( \mathcal{A}^\perp = \mathbb{R}^n - \mathcal{A} \), i.e., the complement of the allowed region) in the state space \( \mathbb{R}^n \). Every time \( X \) exceeds the allowed region (i.e., reaches the forbidden region), a plant fault happens. For example, for a drone swarm control CPS, any two drones must maintain a distance of over 500 meters. Dropping below this 500 meter limit means a plant fault happens.

3.2 Cyber Subsystem Model

We assume the following about the cyber subsystem (see Figure 1).

**Assumption 1.** Except for \( M_{rs} \) and \( M_{fd} \) and their interfaces to the rest of the cyber subsystem, the cyber subsystem is a black box to the 2L-CCPS user. The user knows nothing about the existence,\(^4\) interconnection details, and implementation details of all other cyber subsystem digital modules. This is common in practice. For example, in computer operating systems (OSs), except for some application layer modules (analogous to \( M_{rs} \) and \( M_{fd} \)), the rest of the OS modules are black boxes to OS users.

**Assumption 2.** The cyber subsystem, however, is a white box to the 2L-CCPS vendor. The vendor can suggest to the user alternatives to upgrade (or patch, or reconfigure) the 2L-CCPS without revealing cyber subsystem modular details, i.e., the interconnection and internal implementation details of digital modules. This is again a common practice, e.g.,

---

\(^4\)After deployment, if the 2L-CCPS vendor requests to upgrade (or patch, or reconfigure) some of the digital modules, existence of these modules may be revealed to the user, but not the interconnection and internal implementation details of these modules.
OS vendors often suggest different ways to patch OSs to users without revealing the modular details.

**Assumption 3.** The time cost to deliver a plant’s state sample to the cyber subsystem is $\tau_1$ (see Figure 1); and the time cost to run the cyber subsystem and to deliver the outputted reference point value to the physical subsystem is $\tau_2$ (see Figure 1). Every time the cyber subsystem delivers a new reference point value to the physical subsystem, we say a reference point update event happens.

**Assumption 4.** The cyber subsystem decides the new reference point value purely based on the most recent remote sensing of the plant’s state. In other words, the cyber subsystem is memoryless.

According to **Assumption 1**, to users, the cyber subsystem is a black box except the known existence of the “remote sensing” module $M_{rs}$ and the “final decision” module $M_{fd}$ (see Figure 1). The single cyber subsystem input port sends the current state of the plant $X$ into $M_{rs}$; and the single cyber subsystem output port sends the decision from $M_{fd}$ as the new reference point value $O_{ref}'$ to the physical subsystem. $M_{rs}$ senses the state of the physical plant, and outputs $M_{rs}(X) + N$ to the rest of the cyber subsystem, where $M_{rs}(X)$ is the sensing result without noise, and $N$ is the cross-domain noise random variable (RV). The cross-domain noise RV $N$ hence will propagate throughout the black box cyber subsystem to interfere the final decision making.

### 3.3 Combined Model

The hybrid automaton (Tabuada 2009) of Figure 2, denoted as $H$, models the combined “cyber” and “physical” aspects of 2L-CCPS.

$H$’s node describes the continuous behavior of the combined model. It includes Equation (3) and the continuous increase of time: $i = 1$. $H$’s edge describes the discrete behavior of the combined model. It represents a reference point update event: at time $t_0$, the cyber subsystem can change the value of reference point $O_{ref}$ by delivering the cyber subsystem’s output to the physical subsystem. After a reference point update event, $O_{ref}$ takes a new value (denoted as $O_{ref}'(t_0)$ in Figure 2) and remains constant until the next reference point update event. Note, to comply with reality, we assume the triggering of reference point update events is non-zeno.

### 4 CROSS-DOMAIN NOISE IMPACT EVALUATION FRAMEWORK

As mentioned in Section 1, noises in the physical subsystem, such as local sensing noises, controller output disturbances, and plant modeling errors, are well studied and can be well contained by the physical subsystem. Therefore, these noises are not the focus of this article. Instead, we focus on cross-domain noises (i.e., the noise denoted by RV $N$ in Figure 1), which are not contained within the physical subsystem. Correspondingly, in the following, unless explicitly denoted, we use the term “noise” and “cross-domain noise” interchangeably. Our goal is to propose a framework of methods to evaluate the cross-domain noise’s impact on a 2L-CCPS (see Figure 1). In this section, we propose a hybrid automata reachability based metric to quantify the impact, and propose a corresponding basic measurement method.
4.1 Elementary Trial and Reachability Probability

The physical subsystem of a 2L-CCPS is modeled by Equation (3); hence, is memoryless. That is, the future trajectory of the plant $X(t)$ ($t \in (t_0, +\infty)$, where $t_0$ is the current time) is only dependent on the current state $X(t_0)$ and the current and future reference point values $O_{\text{ref}}(t)$ ($t \in [t_0, +\infty)$).

In practice, the derivative on the left-hand side of Equation (3) is finite, therefore, we can also say the future trajectory of $X(t)$ ($t \in (t_0, +\infty)$) is only dependent on the current state $X(t_0)$ and future reference point values $O_{\text{ref}}(t)$ ($t \in (t_0, +\infty)$).

Suppose the current time is $t_0 - \tau_1 - \tau_2$ (where $\tau_1$ and $\tau_2$ are the two delay time costs; see Figure 1), and the current plant state $X(t_0 - \tau_1 - \tau_2)$ is given: $X(t_0 - \tau_1 - \tau_2) = X^0$. We carry out the following elementary trial. At $t_0 - \tau_1 - \tau_2$, the cyber subsystem samples the current plant state and triggers the corresponding reference point update event at $t_0 - \tau_1 - \tau_2 + \tau_1 + \tau_2 = t_0$ (see Figure 2), changing $O_{\text{ref}}$ to $O'_{\text{ref}}(t_0)$. After that, the cyber subsystem triggers no more reference point update event.

With the concept of elementary trial, we shall propose a methodology framework to evaluate the cross-domain noise impact on a 2L-CCPS. Meanwhile, to simplify our theoretical modeling and analysis, we assume the following.

**Assumption 5.** Unless otherwise denoted, in the following theoretical modeling and analysis sections (i.e., from here to the end of Section 5, including Appendices A, B, and C), we assume $\tau_1 = \tau_2 = 0$.

Later, in Section 5.4, we will discuss the implications of Assumption 5 to real-world systems with non-zero delays. But for now, under Assumption 5, suppose the current time is $t_0$, and the current plant state is $X(t_0) = X^0$, then an elementary trial shall run as follows. At $t_0$, the cyber subsystem samples the current plant state and triggers the corresponding reference point update event at $t_0$ (see Figure 2), changing $O_{\text{ref}}$ to $O'_{\text{ref}}(t_0)$. After that, the cyber subsystem triggers no more reference point update event.

In the elementary trial, the sampling, and hence the cyber subsystem’s decision making, are interfered by the cross-domain noise RV $N$ (see Figure 1). Therefore, whether a plant fault will happen (i.e., $X(t)$ reaches the forbidden region $\mathcal{A}$ during $(t_0, +\infty)$) becomes random, and can be represented by a Bernoulli RV of $R(N, X^0)$: $R(N, X^0) = 1$ represents that a plant fault will happen; and $R(N, X^0) = 0$ otherwise. We call $R(N, X^0)$ the reachability RV under cross-domain noise RV $N$ and given $X^0$, and denote the reachability probability $Pr(R(N, X^0) = 1)$ as $p(N, X^0)$; and consequently, $Pr(R(N, X^0) = 0) = 1 - p(N, X^0)$. Intuitively, $p(N, X^0)$ reflects the risk of the 2L-CCPS under cross-domain noise RV $N$ and given $X^0$ (interested readers can refer to Appendix A to further understand this intuition). In the following, unless otherwise denoted, we simplify $R(N, X^0)$ as $R$ and $p(N, X^0)$ as $p$.

4.2 Measuring Reachability Probability

Next, we describe how to measure the value of $p(N, X^0)$. Under cross-domain noise RV $N$ and given $X(t_0) = X^0$, we run a campaign of $\eta$ elementary trials. The value of $p(N, X^0)$ can be estimated by averaging the results of these elementary trials.

Specifically, denote the reachability RV for the $j$th ($j = 1, \ldots, \eta$) elementary trial as $R_j$. Denote $\hat{R} \overset{\text{def}}{=} \frac{1}{\eta} \sum_{j=1}^{\eta} R_j$. According to the well-known central limit theorem, when $\eta$ is big enough, we can use $\hat{R}$ to estimate $p(N, X^0)$. This is quantitatively elaborated by the following proposition.
Proposition 4.1 (Campaign Scale). Under cross-domain noise RV $N$, given $X(t_0) = X^0$, $\alpha \in [0, 1]$, and $\delta_p \in (0, +\infty)$,

$$if \eta \geq \left(\frac{\Phi^{-1}(1 - \frac{\alpha}{2})}{2\delta_p}\right)^2,$$

(4)

where $\Phi$ is the cumulative distribution function of the standard normal distribution and $\Phi^{-1}$ is $\Phi$'s inverse; then $\hat{R}$ falls within range $p \pm \delta_p$ with confidence level of $(1 - \alpha)$. That is, $Pr(|\hat{R} - p| \leq \delta_p) \geq 1 - \alpha$.

Proof. Due to the memoryless assumption of the cyber and physical subsystems, $R_j$'s are identical independent distribution RVs, and $R_j \sim Bernoulli(p)$. According to the central limit theorem, RV $\hat{R}$ therefore conforms to the normal distribution $\text{Normal}(\mu, \sigma^2/\eta)$, where $\mu$ and $\sigma^2$ are respectively the expectation and variance of $R_j$. As $R_j \sim Bernoulli(p)$, $\mu = p$ and $\sigma^2 = p(1-p) \leq \frac{1}{4}$ (because $p \in [0, 1]$), i.e., $\sigma \leq \frac{1}{2}$.

Also Inequality (4) $\Rightarrow \sqrt{\eta} \geq \frac{\Phi^{-1}(1 - \frac{\alpha}{2})}{2\delta_p} \Rightarrow \delta_p \geq \frac{\Phi^{-1}(1 - \frac{\alpha}{2})}{2\sqrt{\eta}}.$

Therefore, $\hat{R} \sim \text{Normal}(\mu, \sigma^2/\eta) \Rightarrow Pr(|\hat{R} - \mu| \leq \frac{\sigma}{\sqrt{\eta}} \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)) \geq 1 - \alpha$

$\Rightarrow Pr\left(|\hat{R} - p| \leq \frac{1}{2\sqrt{\eta}} \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\right) \geq 1 - \alpha$ (as $\mu = p$ and $\sigma \leq \frac{1}{2}$)

$\Rightarrow Pr(|\hat{R} - p| \leq \delta_p) \geq 1 - \alpha$ (due to Inequality (5)).

Proposition 4.1 implies that under cross-domain noise RV $N$, given $X(t_0) = X^0$, $\alpha$, and $\delta_p$, after a measurement campaign of $\eta$ ($\eta$ satisfies Inequality (4)) elementary trials, we derive a realization $\hat{r}$ of RV $\hat{R}$, which can be used as an estimation of $p$, i.e., $\hat{p} = \hat{r}$, with confidence level of at least $(1 - \alpha)$.

As $\hat{R}$'s realization, we have $\hat{r} = \frac{1}{\eta} \sum_{j=1}^{\eta} r_j$, where $r_j$ is RV $R_j$'s realization in the corresponding elementary trial. To get $r_j$, the simple way is to emulate the $j$th elementary trial as follows:

**Step 1.** Feed the initial plant state $X^0$ into the real cyber subsystem and derive $O_{ref}'$

**Step 2.** Simulate the physical subsystem of Equation (3), from simulator time $t_0$ to simulator time $+\infty$, with initial plant state $X^0$, and updated reference point value $O_{ref}'$. If the resulted trajectory $X(t)$ ($t \in [t_0, +\infty)$) reaches the forbidden region $\mathcal{A}$, then $r_j = 1$, otherwise $r_j = 0$.

In practice, infinite time simulation is impossible. Therefore, **Step 2** has to be accelerated. This when the physical subsystem (described by Equation (3)) is an LTI control system.

In control engineering, it is a well established practice that LTI control systems in the form of Equation (3) are designed to be stable in the sense of Lyapunov (Brogan 1991). Specifically, $K$ of Equation (2) is designed such that a positive definite symmetric matrix $P \in \mathbb{R}^{n\times n}$ exists to satisfy

$$F^TP + PF = -I,$$

(6)

where $I$ is the $n \times n$ identity matrix.

Correspondingly, given control systems of Equation (3) that are stable in the sense of Lyapunov, there are mature tools (Brogan 1991) to derive the aforementioned $P$. 

ACM Transactions on Cyber-Physical Systems, Vol. 3, No. 1, Article 2. Publication date: August 2018.
1. ElementaryTrialEmulation(input: N, X⁰; output: r_j){
2.   Input X(t₀) = X⁰ into the cyber subsystem to generate O_refl(t₀); // or equivalently, let Mₙ output Mₙ(X(t₀)) + N to the rest // of the cyber subsystem to generate O_refl(t₀), where X(t₀) = X⁰. 
3.   Current simulator time t ← t₀; 
4.   O_refl ← O_refl(t₀); 
5.   while (true){
6.     Derive X(t) according to Eq. (3):
7.       if (X(t) ∈ ˘A) { r_j ← 1; break; } 
8.       if (V(X(t), O_refl) < inf²∈˘A {V(Y, O_refl)}) { r_j ← 0; break; } 
9.       t ← t + δ; // δ: per iteration simulator time increment 
10.      if (t ≥ T_sim) { // T_sim: maximum simulation time 
11.         r_j ← 1; break; 
12.       } 
13.   
14. }

Fig. 3. Pseudo C code to emulate an elementary trial, to calculate r_j. It is an emulation because Line 2 uses the real cyber subsystem.

With P, we can define a Lyapunov function V(X(t), O_refl(t)) as follows.

\[ V(X(t), O_refl(t)) \overset{\text{def}}{=} (X(t) - O_refl(t))^T P (X(t) - O_refl(t)). \] (7)

Intuitively, Lyapunov function represents a virtual “potential energy” of the physical plant. If the physical subsystem is stable, this potential energy should monotonically decrease. This is quantified by the following proposition.

**Proposition 4.2 (Trajectory Boundary).** Given \( X(t₀) = X⁰ \in \mathbb{R}^n \) and \( O_refl(t₀) \in \mathbb{R}^n \), let \( X(t) \) \( (t \in [t₀, +\infty)) \) be the trajectory of plant state evolved according to Equation (3) when \( O_refl(t) \equiv O_refl(t₀) \), then \( \forall t \in [t₀, +\infty) \),

\[ \frac{d V(X(t), O_refl(t))}{d t} \leq 0. \] (8)

**Proof.** Proposition 4.2 is already implied in the classic proof of Lyapunov stability (Khalil 2001). The details are recompiled in Appendix B.

Due to Proposition 4.2, in an elementary trial, the plant’s Lyapunov function value monotonically drops. Particularly, if it drops below the minimum Lyapunov function value of the forbidden region ˘A, the plant state can never reach ˘A again. Based on this heuristics, we propose the algorithm of Figure 3 to emulate the \( j \)th elementary trial \( (j = 1, \ldots, \eta) \), so as to approximate \( r_j \), the realization of reachability RV \( R_j \).

In Figure 3, Line 7 corresponds to the case that trajectory \( X(t) \) is found to reach forbidden region ˘A, hence \( r_j = 1 \). In Line 8, as future trajectory \( X(t) \)’s Lyapunov function value drops below \( \inf²∈˘A \{V(Y, O_refl)\} \), a simple proof with negation can show that due to Inequality (8), \( X(t) \) will never reach any points in ˘A. Line 11 corresponds to the situation that after sufficiently long simulation, we still cannot decide if \( X(t) \) reaches ˘A; therefore, we pessimistically overapproximate with \( r_j = 1 \).

### 4.3 Quantifying Impact of Cross-Domain Noise with Reachability Probability

Now we can get the \( \eta \) realizations \( \{r_j\} \). Let \( \hat{p} \overset{\text{def}}{=} \hat{r} \overset{\text{def}}{=} \frac{1}{\eta} \sum_{j=1}^{\eta} r_j \). As per Proposition 4.1, when \( \eta \) satisfies Inequality (4), \( \hat{p} = \hat{r} \) is a \((1 - \alpha)\) confident estimation of \( p \). By definition, \( p \) is an elementary...
trial’s reachability probability (i.e., probability to reach forbidden region $\mathcal{A}$) under cross-domain noise RV $N$ and given initial plant state $X^0$. That is, $p$’s elaborative form is $p(N, X^0)$, and it measures the risk of an elementary trial.

The impact of cross-domain noise RV $N$ should be the risk increase caused by $N$. Let $I(N, X^0)$ denote the impact of $N$ on the 2L-CCPS with initial plant state $X(t_0) = X^0$. Then we propose to quantify $I(N, X^0)$ as

$$I(N, X^0) \overset{\text{def}}{=} p(N, X^0) - p(0, X^0),$$

where $p(0, X^0)$ is an elementary trial’s reachability probability under 0 cross-domain noise and given initial plant state $X^0$.

To holistically quantify the impact of $N$ to the 2L-CCPS, ideally, we should evaluate $I(N, X^0)$ for every $X^0 \in \mathbb{R}^n$. Obviously this is impractical. Instead, we propose to use a benchmark $X = \{X_i^0\}_{i=1,\ldots,b}$ of $b$ sample points in the allowed region $\mathcal{A}$ (i.e., $\forall i, X_i^0 \in \mathcal{A}$). The $b$ sample points in $X$ are fixed, or the sampling method is fixed (e.g., uniform sampling in $\mathcal{A}$). We call each sampled point $X_i^0$ a benchmark point.

With benchmark $X = \{X_i^0\}_{i=1,\ldots,b}$, we summarize our basic 2L-CCS cross-domain noise impact evaluation method as follows. Given cross-domain noise RV $N$, for each benchmark point $X_i^0 \in X$, we run the elementary trial campaign described in Sections 4.1 and 4.2 to get reachability probability $p_i(N, X_i^0)$ and $p_i(0, X_i^0)$, and follow Equation (9) to get cross-domain noise impact $I_i(N, X_i^0)$. The holistic impact of cross-domain noise RV $N$ is thus quantified by the set $\{I_i(N, X_i^0)\}_{i=1,\ldots,b}$.

5 SHRINKING BENCHMARK REGION

5.1 Refined 2L-CCPS Architecture

In Section 4, the benchmark points are sampled from the entire allowed region $\mathcal{A}$. This benchmark sampling region (simplified as “benchmark region” in the following) is too big. On the other hand, for an initial plant state $X^0 \in \mathcal{A}$ sufficiently away from the forbidden region $\mathcal{A}$, the plant trajectory may never reach $\mathcal{A}$, even perturbed by large cross-domain noises. It is therefore meaningless to include such $X^0$ in the benchmark. To make an analogy, to benchmark meteoroids’ reachability to the Earth, it is sufficient to focus on meteoroids in the solar system; meteoroids in other galaxies are practically irrelevant. Based on the above heuristics, we propose to shrink the benchmark region as follows.

We refine the classic 2L-CCPS architecture of Figure 1 by adding a bounding filter to the input port of the physical subsystem (see Figure 4). This bounding filter rejects extreme new reference point values from the cyber subsystem. Specifically, suppose at time $t_0$ a reference point update event happened, and $X(t_0) = X^0$. Then, the bounding filter will define a hyper bounding ball $\text{Ball}(X^0, \gamma)$ in the state space, centered at $X^0$ with radius $\gamma > 0$. If the new reference point value $O_{\text{ref}}'$ from the cyber subsystem is within $\text{Ball}(X^0, \gamma)$, then $O_{\text{ref}}'$ is accepted. Otherwise, $O_{\text{ref}}'$ is truncated. Formally, the filtered new reference point value $O_{\text{ref}}''$ is

$$O_{\text{ref}}'' = \begin{cases} O_{\text{ref}}' - \frac{O_{\text{ref}}' - X^0}{\|O_{\text{ref}}' - X^0\|_2} \gamma + X^0 & \text{if } \|O_{\text{ref}}' - X^0\|_2 \geq \gamma \\ O_{\text{ref}}' & \text{otherwise.} \end{cases}$$

Note, Equation (10) implies that the classic 2L-CCPS architecture (see Figure 1) is a special case of the refined 2L-CCPS architecture (see Figure 4), where $\gamma = +\infty$.

With the bounding filter, no matter what the cross-domain noise RV $N$ is, given the current plant state $X^0$, a reference point update event can only change reference point to a value within $\text{Ball}(X^0, \gamma)$. Therefore, in the refined 2L-CCPS architecture, given whatever cross-domain noise $N$, for an elementary trial starting from plant state $X^0$, the reachable state space of all possible future
trajectories is constrained. Denote this reachable state space as $\text{Traj}(N, X^0)$. Denote \[
\bar{B}^* \overset{\text{def}}{=} \{X^0 | X^0 \in \mathcal{A} \text{ and } \text{Traj}(N, X^0) \cap \mathcal{A} = \emptyset \text{ for whatever RV } N\}.
\]
Then for whatever RV $N$, $\forall X^0 \in \bar{B}^*$, $p(N, X^0) \equiv 0$ and $I(N, X^0) \equiv 0$. Therefore, if we can explicitly identify $\bar{B}^*$, then we do not need to benchmark test any point in $\bar{B}^*$. A point in $\bar{B}^*$ is thus an “irrelevant benchmark point.”

Correspondingly, the (relevant) benchmark points only need to be sampled from $B^* \overset{\text{def}}{=} \mathcal{A} - \bar{B}^*$. More specifically, we call $B^*$ the “tight shrunk benchmark region,” and call any $B \supseteq B^*$ ($B \subseteq \mathcal{A}$) a “shrunk benchmark region.” We call $\bar{B}^*$ the “tight irrelevant benchmark region,” and call any $\bar{B} \supseteq \bar{B}^*$ ($\bar{B} \subseteq \mathcal{A}$) an “irrelevant benchmark region.”

5.2 Heuristics to Shrink Benchmark Region

Now, the question is how to find $B$, or equivalently $\bar{B}$, given the bounding filter (see Figure 4).

Our solution heuristics is still based on Proposition 4.2. Basically, for a well designed LTI physical subsystem, the plant’s Lyapunov function $V(X(t), O_{\text{ref}}(t))$ exists, and is monotonically decreasing when $O_{\text{ref}}(t)$ is a constant, which is the case for elementary trials. According to Proposition 4.2, at time $t_0$, given initial plant state $X(t_0) = X^0 \in \mathcal{A}$ and bounding filtered new reference point value $O''_{\text{ref}}(t_0) \in \text{Ball}(X^0, \gamma)$, the trajectory of an elementary trial $X(t) \ (t \in [t_0, +\infty))$ is confined by the hyper-ellipsoid $E(X^0, O''_{\text{ref}}(t_0))$ of

\[
E(X^0, O''_{\text{ref}}(t_0)) \overset{\text{def}}{=} \{Y | Y \in \mathbb{R}_n \text{ and } (Y - O''_{\text{ref}}(t_0))^T \mathbf{P} (Y - O''_{\text{ref}}(t_0)) \leq V(X^0, O''_{\text{ref}})\}, \tag{11}
\]

where $\mathbf{P}$ is the positive definite symmetric matrix in the Lyapunov function of Equation (7). We call $E(X^0, O''_{\text{ref}}(t_0))$ a “Lyapunov hyper-ellipsoid.”

As shown by Figure 5, if none of such confining Lyapunov hyper-ellipsoids intersects with $\mathcal{A}$, then $X^0 \in \bar{B}^*$. Consequently, the set of such $X^0$’s constitute a $\bar{B} \subseteq B^*$. Formally, let us define \[
V^\text{sup}_{X^0, \text{Ball}(X^0, \gamma)} \overset{\text{def}}{=} \sup_{O''_{\text{ref}} \in \text{Ball}(X^0, \gamma)} \{V(X^0, O''_{\text{ref}})\}, \tag{12}
\]
and for arbitrary $\mathcal{Y} \subseteq \mathbb{R}_n$, define \[
V^\text{inf}_{\mathcal{Y}, \text{Ball}(X^0, \gamma)} \overset{\text{def}}{=} \inf_{O''_{\text{ref}} \in \text{Ball}(X^0, \gamma)} \{V(Y, O''_{\text{ref}}) | YY \in \mathcal{Y}\}.
\]

Then the intuition of Figure 5 is formalized by Lemma 5.1.
**Lemma 5.1 (Irrelevant Benchmark Point).** For any state $X^0 \in \mathcal{A}$, if $V_{X^0, \text{Ball}(X^0, \gamma)}^\sup < V_{\mathcal{A}, \text{Ball}(X^0, \gamma)}^\inf$, then $X^0 \in \overline{B}^\ast$.

**Proof.** For any elementary trial starting with $X(t_0) = X^0$, no matter what RV $N$ is, the resulted new reference point after bounding filtering, denoted as $O''_{\text{ref}}(t_0)$, is within $\text{Ball}(X^0, \gamma)$. If $V_{\mathcal{A}, \text{Ball}(X^0, \gamma)}^\inf > V_{X^0, \text{Ball}(X^0, \gamma)}^\sup$, then the elementary trial plant state trajectory’s initial Lyapunov function value $V(X(t_0), O''_{\text{ref}}(t_0))$ is less than that of any state in $\overline{\mathcal{A}}$. As per Proposition 4.2, the elementary trial plant state trajectory can never reach $\overline{\mathcal{A}}$. This is true for any elementary trial starting with $X(t_0) = X^0$ under whatever RV $N$. Therefore, $\text{Traj}(N, X^0) \cap \overline{\mathcal{A}} = \emptyset$ for whatever RV $N$. $\square$

### 5.3 Closed-Form Definition of Shrunken Benchmark Region

This subsection shall extend Lemma 5.1 to find a closed-form $\overline{B}$, hence $B$.

Our heuristics is to first find the closed-form formula for $V_{X^0, \text{Ball}(X^0, \gamma)}^\sup$. Using this formula, we then find a sufficient condition for $V_{X^0, \text{Ball}(X^0, \gamma)}^\sup < V_{\mathcal{A}, \text{Ball}(X^0, \gamma)}^\inf$. Then any $X^0$ satisfying the sufficient condition should belong to $\overline{B}^\ast$. Consequently, the set of such $X^0$’s constitute a $\overline{B} \subseteq \overline{B}^\ast$.

Figure 6 gives the intuition to find the closed-form formula to calculate $V_{X^0, \text{Ball}(X^0, \gamma)}^\sup$. Given $X^0$ and $O''_{\text{ref}} \in \text{Ball}(X^0, \gamma)$, the maximum Lyapunov function value $V(X^0, O''_{\text{ref}})$ is achieved when we choose $O''_{\text{ref}} = O_1$, so that the radius of $\text{Ball}(X^0, \gamma)$ exactly overlaps with the semi-minor axis of Lyapunov hyper-ellipsoid $E(X^0, O''_{\text{ref}})$ (see Equation (11)). Note the directions and lengths ratio of the major and minor axes of all Lyapunov hyper-ellipsoids are fixed once $P$ is given; and $E(X^0, O''_{\text{ref}})$ is centered on $O''_{\text{ref}}$ and has $X^0$ on the surface.
Figure 6’s intuition to find the closed-form formula of $V_{X^0, \text{Ball}(X^0, y)}^{\sup}$ is formalized by Lemma 5.2.

**Lemma 5.2 (Closed-Form Value of $V_{X^0, \text{Ball}(X^0, y)}^{\sup}$).** We have $V_{X^0, \text{Ball}(X^0, y)}^{\sup} = \lambda_{\max}(P)y^2$, where \(\lambda_{\max}(P)\) is the maximal eigenvalue of $P$ in Lyapunov function of Equation (7).

**Proof.** According to Equation (12), $V_{X^0, \text{Ball}(X^0, y)}^{\sup}$ is the optimal objective function value for the following optimization problem:

$$
\begin{align*}
\max_{O_{\text{ref}}} & \ f_{X^0}(O_{\text{ref}}') = V(X^0, O_{\text{ref}}') = (X^0 - O_{\text{ref}}')^T P (X^0 - O_{\text{ref}}') \\
\text{s.t.} & \ (X^0 - O_{\text{ref}}')^T (X^0 - O_{\text{ref}}') \leq y^2,
\end{align*}
$$

(13)

where $O_{\text{ref}}'$ is the only optimization variable.

Problem (13) is a typical Quadratic Constrained Quadratic Optimization (QCQP) problem (Boyd and Vandenberghe 2004). As this problem has a single constraint and the constraint itself is a hyperball, a special form of quadratic function, we can solve it as follows.

First, denote $\tilde{O}_{\text{ref}} \overset{\text{def}}{=} X^0 - O_{\text{ref}}'$, and $f_{X^0}(\tilde{O}_{\text{ref}}) = -f_{X^0}(O_{\text{ref}}') = -\tilde{O}_{\text{ref}}'P\tilde{O}_{\text{ref}}$. Then, problem (13) is equivalent to problem

$$
\begin{align*}
\min_{\tilde{O}_{\text{ref}}} & \ f_{X^0}(\tilde{O}_{\text{ref}}) \\
\text{s.t.} & \ \tilde{O}_{\text{ref}}'\tilde{O}_{\text{ref}} \leq y^2.
\end{align*}
$$

(14)

The Lagrangian of optimization problem (14) is

$$
L(\tilde{O}_{\text{ref}}, v) = \tilde{O}_{\text{ref}}'(vI - P)\tilde{O}_{\text{ref}} - vy^2,
$$

and the dual function is

$$
g(v) = \inf_{\tilde{O}_{\text{ref}}} L(\tilde{O}_{\text{ref}}, v) = \begin{cases} 
-vy^2 & \text{(if } vI - P \succeq 0) \\
-\infty & \text{(otherwise)}
\end{cases}
$$

where “$\succeq 0$” means the matrix on the left-hand side is positive semidefinite. Using a Schur complement (Boyd and Vandenberghe 2004), the Lagrange dual problem to problem (14) is

$$
\begin{align*}
\max_{v} & \ h \\
\text{s.t.} & \ v \succeq 0
\end{align*}
$$

(15)

$$
\begin{bmatrix} vI - P & 0 \\
0 & -vy^2 - h \end{bmatrix} \succeq 0.
$$

As problem (14) is strictly feasible, i.e., there exists some $\tilde{O}_{\text{ref}}$ (e.g., $\tilde{O}_{\text{ref}} = 0$) s.t. $\tilde{O}_{\text{ref}}'\tilde{O}_{\text{ref}} \leq y^2$, problem (15) holds strong duality to problem (14) (Boyd and Vandenberghe 2004). Hence, the two problems’ optimal values are equal. By solving problem (15), we have the optimal value

$$
h^* = -\lambda_{\max}(P)y^2,
$$

where $\lambda_{\max}(P)$ is the maximal eigenvalue of matrix $P$. Then we have

$$
f_{X^0}(O_{\text{ref}}'^*) = -f_{X^0}'(\tilde{O}_{\text{ref}}) = -h^* = \lambda_{\max}(P)y^2.\quad \Box
$$

Now we know that given $X^0 \in \mathcal{A}$, $V_{X^0, \text{Ball}(X^0, y)}^{\sup} \leq \lambda_{\max}(P)y^2$. Then, it is possible to find a sufficient condition to make $V_{X^0, \text{Ball}(X^0, y)}^{\sup} < V_{\mathcal{A}, \text{Ball}(X^0, y)}^{\inf}$. To find such sufficient condition, let us first
define the distance between a point \( X^0 \in \mathbb{R}_n \) and a region \( \mathcal{Y} \subseteq \mathbb{R}_n \) as
\[
\text{Dis}(X, \mathcal{Y}) \overset{\text{def}}{=} \inf\{|X - Y|_2|\forall Y \in \mathcal{Y}\}.
\]

Then a sufficient condition is described by Lemma 5.3.

**Lemma 5.3 (Irrelevance Distance).** Given \( \mathcal{Y} \subseteq \mathbb{R}_n \), state \( X^0 \in \mathcal{A} \), and an arbitrarily small positive constant \( \varepsilon > 0 \), if
\[
\text{Dis}(X^0, \mathcal{Y}) > \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}} \gamma + \gamma + \varepsilon \overset{\text{def}}{=} \Gamma,
\]
where \( \lambda_{\max}(P) \) and \( \lambda_{\min}(P) \) are, respectively, the maximum and minimum eigenvalues of the positive definite symmetric matrix \( P \) of Equation (7), then \( V_{X^0, \mathcal{Y}}^{\sup} < V_{\mathcal{Y}, \mathcal{Y}}^{\inf} \).

**Proof.** \( \forall O_{ref}'' \in \text{Ball}(X^0, \gamma), \forall Y \in \mathcal{Y}, \)
\[
V(Y, O_{ref}'') = (Y - O_{ref}'')^T P (Y - O_{ref}'').
\] (17)

Due to the bounding filter, we know that
\[
(O_{ref}' - X^0)'(O_{ref}' - X^0) \leq \gamma^2.
\]

Also, as \( \text{Dis}(X^0, \mathcal{Y}) > \Gamma \), we have
\[
(Y - X^0)'(Y - X^0) > \Gamma^2.
\]

From Equation (17), we get
\[
V(Y, O_{ref}'') \geq \lambda_{\min}(P)(Y - O_{ref}'')'(Y - O_{ref}'') = \lambda_{\min}(P)[(Y - X^0) - (O_{ref}' - X^0)][(Y - X^0) - (O_{ref}' - X^0)]
\]
\[
> \lambda_{\min}(P)(\Gamma - \gamma)^2 \quad \text{(see Lemma C.1 in Appendix C)}
\]
\[
> \lambda_{\max}(P)\gamma^2 + \lambda_{\min}(P)\varepsilon^2 = V_{X^0, \mathcal{Y}}^{\sup} + \lambda_{\min}(P)\varepsilon^2.
\]

That is, \( \forall O_{ref}'' \in \text{Ball}(X^0, \gamma), \forall Y \in \mathcal{Y}, \) we have \( V(Y, O_{ref}'') > V_{X^0, \mathcal{Y}}^{\sup} \). Therefore,
\[
V_{X^0, \mathcal{Y}}^{\sup} < V_{\mathcal{Y}, \mathcal{Y}}^{\inf}.
\]

We call \( \Gamma \) the irrelevance distance. Figure 7 visualizes the intuition of \( \Gamma \). Basically, if \( \text{Dis}(X^0, \mathcal{Y}) > \Gamma \), then no Lyapunov hyper-ellipsoid \( E(X^0, O_{ref}'') \) (\( \forall O_{ref}'' \in \text{Ball}(X^0, \gamma) \)) can intersect with \( \mathcal{Y} \). Hence, elementary trial trajectories starting from \( X^0 \) can never reach \( \mathcal{Y} \). In case \( \mathcal{Y} = \mathcal{A} \) and \( X^0 \in \mathcal{A} \), \( X^0 \) thus is an irrelevant benchmark point: \( X^0 \in \mathcal{B}^* \).

Lemma 5.3 thus helps us to find a closed-form shrunk benchmark region \( \mathcal{B} \), as described by Theorem 5.4.

**Theorem 5.4 (Shrunk Benchmark Region).** For the refined 2L-CCPS architecture,
\[
\mathcal{B} \overset{\text{def}}{=} \{X^0|X^0 \in \mathcal{A}, \text{Dis}(X^0, \mathcal{A}) \leq \Gamma\}
\]
is a shrunk benchmark region.

**Proof.** \( \forall X^0 \in \mathcal{B} = \mathcal{A} - \mathcal{B} \), \( \text{Dis}(X^0, \mathcal{A}) > \Gamma \). Due to Lemma 5.3, we know that \( V_{X^0, \mathcal{A}}^{\sup} < V_{\mathcal{A}, \mathcal{A}}^{\inf} \). Due to Lemma 5.1, we know \( X^0 \in \mathcal{B}^* \). Therefore, \( \mathcal{B} \subseteq \mathcal{B}^* \). That is, \( \mathcal{B} \supseteq \mathcal{B}^* \).
Figure 8 illustrates an example shrunk benchmark region derived via Theorem 5.4. Now, to build benchmark $X$, instead of sampling the entire allowed region $A$, we only need to sample the shrunk benchmark region $B$.

5.4 Discussions on Assumption 5

So far, unless otherwise denoted, all contents of Sections 4 and 5 are based on Assumption 5, which idealizes delay time costs as $\tau_1 = \tau_2 = 0$.

In reality, the delay time costs cannot be zero. Therefore, the evaluation methodology framework proposed in Sections 4 and 5 provides only an idealized theoretical approximation of the reality. But this does not render the theoretical evaluation results useless, because they increase our knowledge and confidence on the real system.

That said, the knowledge and confidence derived from the idealized theoretical approximation are particularly relevant when $\tau_1$ and $\tau_2$ are sufficiently small: e.g., several orders of magnitude smaller than the interval between consecutive reference point update events. This is corroborated by our evaluations in Section 6, where real 2L-CCPS experiment results (see Section 6.4) match idealized theoretical evaluation results (see Sections 6.2 and 6.3).

From a more generic perspective, using idealized theoretical approximation results to increase knowledge and confidence of computer systems is a well adopted engineering practice. For example, when using automata based model checking to verify complex computer systems (those involving thousands of lines of source code), the formal model can rarely exactly match all the source code (that is why we still have to test and debug the source code after model checking).
But this does not render automatic-based model checking useless: we still need model checking to know the real computer system better, and to trust the real computer systems more.

6 EVALUATION

In this section, we evaluate our proposed methodology framework in Sections 4 and 5. Specifically, we evaluate the cross-domain noise impacts of two cyber subsystem upgrade alternatives for an inverted pendulum (Brogan 1991) testbed. By comparing the two evaluation results, a better alternative is chosen. Runtime experiments are then carried out to verify the choice. We also show that Section 5’s benchmark region shrinking method can save 24.1% of the offline evaluation effort, meanwhile achieving the same evaluation goal.

6.1 Inverted Pendulum Testbed

Our testbed is a 2L-CCPS that runs computer vision assisted parallel inverted pendulums (Brogan 1991) (see Figure 9). In the testbed, two unmanned carts respectively maintain the standing of their inverted pendulums (IPs), and maintain a certain cart-convoy formation. The physical subsystem controls the unmanned IP carts’ fine-grain movements, while the cyber subsystem coordinates the cart-convoy formation using computer vision. This is a representative 2L-CCPS testbed, which can be generalized to many real-world applications: e.g., computer vision guided driving or convoy-formation of unmanned automobiles (Beyeler et al. 2014), unmanned aerial vehicles (Kong et al. 2014), and computer vision assisted industrial robot coordination (Kim et al. 2012). All of such systems involve a physical subsystem of mission-critical plants (the unmanned automobiles, the unmanned aerial vehicles, the industrial robots), just like the unmanned carts with IPs, and a computer vision assisted cyber subsystem that runs complex computations to decide coarse-grain coordination.

Specifically, the physical subsystem of the testbed consists of two inverted pendulums: IP$_1$ and IP$_2$. An inverted pendulum is a metal rod with one end hinged on a cart, and the other end free to rotate around the hinge (see Figure 9(a)). The cart can move along a piece of metal rail. The controller of the inverted pendulum takes charge of moving the cart back and forth along the rail to keep the hinged metal rod (the inverted pendulum) standing upright.

For IP$_i$ ($i = 1, 2$), let $X_{ipl}(t)$ denote its plant state. $X_{ipl}$ then includes four state variables (see Figure 9(a)): respectively, the current location $x_{ipl}(t)$ (m) and velocity $\dot{x}_{ipl}(t)$ (m/sec) of the cart, and the current angular displacement $\theta_{ipl}(t)$ (rad) and velocity $\dot{\theta}_{ipl}(t)$ (rad/sec) of the rod from the upright position. That is, $X_{ipl}(t) = (x_{ipl}(t), \theta_{ipl}(t), \dot{x}_{ipl}(t), \dot{\theta}_{ipl}(t))$. 

Fig. 9. Parallel-inverted-pendulum testbed.
As an LTI control system, the physical dynamics of IP \( i \) is governed by the following systems of differential equations (Googol 2016).

\[
\frac{d(X_{ip} - O_{ipref})}{dt} = A_{ipi}(X_{ip} - O_{ipref}) + B_{ipi}U_{ipi},
\]

\[
U_{ipi} = -K_{ipi}(X_{ip} - O_{ipref}),
\]

where \( X_{ip}, O_{ipref}, U_{ipi}, A_{ipi}, B_{ipi}, \) and \( K_{ipi} \) respectively correspond to \( X, O_{ref}, U, A, B, \) and \( K \) in Equations (1) and (2). The specific inverted pendulums we use are made by Googol (Googol 2016), and have the following configurations (for both \( i = 1 \) and 2).

\[
A_{ipi} = \begin{bmatrix}
0.000 & 1.000 & 0.000 & 0.000 \\
0.000 & 0.000 & 0.000 & 0.000 \\
0.000 & 0.000 & 0.000 & 1.000 \\
0.000 & 0.000 & 29.400 & 0.000
\end{bmatrix},
\]

\[
B_{ipi} = \begin{bmatrix}
0.000, 1.000, 0.000, 3.000\end{bmatrix}^T,
\]

\[
K_{ipi} = (-5.0505, -5.8249, 35.2502, 6.2750).
\]

As we have two inverted pendulums, the holistic plant of our testbed can be described by the following differential equation systems.

\[
\frac{d(X_{tb} - O_{tbref})}{dt} = A_{tb}(X_{tb} - O_{tbref}) + B_{tb}U_{tb},
\]

\[
U_{tb} = -K_{tb}(X_{tb} - O_{tbref}),
\]

where \( X_{tb} = \begin{bmatrix} X_{ip1} \\ X_{ip2} \end{bmatrix}, O_{tb} = \begin{bmatrix} O_{ipref1} \\ O_{ipref2} \end{bmatrix}, A_{tb} = \begin{bmatrix} A_{ip1} & 0 \\ 0 & A_{ip2} \end{bmatrix}, B_{tb} = \begin{bmatrix} B_{ip1} \\ B_{ip2} \end{bmatrix}, \) and \( K_{tb} = \begin{bmatrix} K_{ipi} & 0 \\ 0 & K_{ipi} \end{bmatrix}. \)

Both IPs move along the x-axis. The given allowed region \( \mathcal{A} \) for our testbed is

\[
\mathcal{A} = \{X_{tb}|X_{tb} \in \mathbb{R}^8, \text{and} 0.15 \leq x_{ip1} - x_{ip2} \leq 0.2\}. \tag{21}
\]

That is, IP \( 1 \) and IP \( 2 \)'s carts cannot go too close nor too apart.\(^7\)

The cyber subsystem of our testbed takes charge of computing new reference points for the plant (i.e., IP \( 1 \) and IP \( 2 \)) using computer vision sensing inputs. Due to Assumption 2 in Section 1, the cyber subsystem is a white box to the vendor. Figure 10 depicts the white box details.

Note that a reference point represents the equilibrium state that the user aims to achieve. For inverted pendulum IP \( i \) (\( i = 1, 2 \)), the user always wants the equilibrium taking the form \( O_{ipref} = (x_{ipref}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \). That is, at equilibrium, the inverted pendulum cart should stop at \( x_{ipref} \), and the rod should stand still at upright angle. Therefore, the only update the cyber subsystem should make to a reference point is the cart’s equilibrium location \( x_{ipref} \): at different time, the cyber subsystem may want to move the cart to different locations. That is, the cyber subsystem is focusing on computing the new \( x_{ipref} \).

As shown in Figure 10, the cyber subsystem’s computation dataflow starts from \( M_0 \), the “remote sensing” module, where a USB 2 Mega pixel camera captures a 640 × 480 pixel raw image of IP \( 1 \) and IP \( 2 \). Denote the raw image captured as \( D_0 = M_0(X) + N \), where \( X \) is the current plant state, and \( N \) is the cross-domain noise. \( D_0 \) is then fed to modules \( M_1 \) and \( M_2 \), respectively, for red and

\(^{5}\)Strictly speaking, an inverted pendulum control system is not linear, but when \( \theta_{ip} \) is reasonably small (e.g., \( \leq \frac{\pi}{6} \) (rad)), the system can be regarded as linear.

\(^{6}\)Here we are assuming the rails of the IP’s are long enough. Otherwise, a more strict definition of \( \mathcal{A} \) should also include the rail length constraints.

\(^{7}\)In the actual implementation, IP \( 1 \) and IP \( 2 \) are moving along two parallel rails. Therefore, the two inverted pendulums will not really crash. However, for evaluation purposes, we still enforce the allowed region of Inequality (21), regarding IP \( 1 \) and IP \( 2 \) as if moving along a same rail.
yellow color recognition. $M_1$’s output $D_1$ is a binary image: a pixel of 1 means the corresponding pixel in $D_0$ is recognized as red; and 0 otherwise. The same applies to $M_2$ and $D_2$, except that the color to recognize is yellow.

The reason why to carry out red and yellow color recognition is because IP$^1$ and IP$^2$’s carts respectively bear a red and a yellow label. By recognizing the red and yellow label, the cyber subsystem identifies $x_{ip^1}$ and $x_{ip^2}$, the current locations of the two carts. This is realized by feeding $D_1$, $D_2$ respectively to $M_3$ and $M_4$ for IP$^1$ and IP$^2$ cart localization. The output of $M_3$ (i.e., $D_3$) is the estimation of $x_{ip^1}$; while the output of $M_4$ (i.e., $D_4$) is the estimation of $x_{ip^2}$. $D_3$ and $D_4$ are fed to $M_5$, the “final decision” module, to compute the new reference point values, i.e., $x_{ipref^1}$ and $x_{ipref^2}$.

6.2 Offline Cross-Domain Noise Impact Evaluation

In our testbed of Figure 10, raw image data (i.e., $D_0$) captured by $M_0$ are noisy. This cross-domain noise propagates through the network of digital modules, and finally affects the plant. In order to enhance robustness against the cross-domain noise, the testbed vendor proposes two upgrading alternatives: either upgrade $M_1$ to a commercial-off-the-shelf (COTS) module of $M_1’$ or to upgrade $M_3$ to a COTS module of $M_3’$, but not both, because of budget limit. Meanwhile, as both $M_1’$ and $M_3’$ are COTS, their interconnection and internal implementation details are hidden to the user. To independently decide which alternative to take, the testbed user carries out the cross-domain noise impact evaluation framework of Sections 4 and 5.

As summarized by the last paragraph of Section 4, the first step of the evaluation framework is to prepare a benchmark $X = \{X^0_i\}_{i=1,...,b}$. Without loss of generality, the user chooses $b = 1,000$. For the time being, the user first tries the framework without benchmark region shrinking. That is, the user sample $b = 1,000$ benchmark points from the entire allowed region $A$ (see Equation (21)).

For each benchmark point $X^0_i$ ($i = 1,\ldots,b$), the framework asks the user to emulate $\eta$ elementary trials following the algorithm of Figure 3. Particularly, the user implements Line 2 according to the alternative way described in the comment. That is, $M_0$ outputs $M_0(X^0_i) + N$ to the rest of the cyber subsystem to generate $O’_{ref}(t_0)$ (note according to Assumption 1 of Section 3.2, $M_0$ and its interface to the rest of the cyber subsystem is not a black box to the user).
The implementation detail is as follows. For each $X_0^i \in X$, the user prepares a high-quality $640 \times 480$ pixels picture $P_i$ as $M_0$’s noiseless output. That is, $P_i = M_0(X_0^i)$. Let $N$ denote the cross-domain noise RV, and $D_{0,i}$ denote the noisy output of $M_0$ corresponding to $X_0^i$. Then $D_{0,i} = M_0(X_0^i) + N = P_i + N$.

Indeed, $D_{0,i}$ is also a $640 \times 480$ pixel picture, with each pixel inflicted by RV $N$. The user generates $D_{0,i}$ pixel by pixel. Let $P_i(j, k) \in [0, 255]$ $(j = 1, 2, \ldots, 640; k = 1, 2, \ldots, 480)$ denote $P_i$’s red (or yellow) color value of the pixel at coordinate $(j, k)$. Let $N(j, k) \in \mathbb{R}$ denote the component of cross-domain noise $N$ at pixel coordinate $(j, k)$. Let $D_{0,i}(j, k)$ denote the noisy raw image red (or yellow) color value at pixel $(j, k)$. Then, $D_{0,i}(j, k) = P_i(j, k) + N(j, k)$ (in practice, $D_{0,i}(j, k)$’s value is rounded to the closest integer in $[0, 255]$).

Without loss of generality, the user generates the cross-domain noise RV $N$ as per Gaussian distribution, i.e., $N(j, k) \sim \text{Normal}(0, \sigma^2)$. The user defines the level of $N$, denoted as $\|N\|$, with mean square error (MSE), a well-known concept in image processing.

$$\text{MSE} \overset{\text{def}}{=} \frac{1}{J \cdot K} \sum_{j=1}^{J} \sum_{k=1}^{K} N^2(j, k), \quad (22)$$

where $J$ and $K$ are, respectively, the width and length of an image in pixels. It can be proven that $\mathbb{E}(\text{MSE}) = \sigma^2$.

The user then discretizes $10 \log_{10}$ MSE’s value range into five intervals; respectively, $(-\infty, -10]$, $[-10, 0]$, $[0, 10]$, $[10, 20]$, and $[20, 30]$. Suppose the $10 \log_{10}$ MSE derived from the current $N$ falls in the $l$th ($l \in \{1, 2, \ldots, 5\}$) interval; then the user says $\|N\| = l$.

With the above methodology to generate $D_{0,i} = M_0(X_0^i) + N$ for each benchmark point $X_0^i$, the user implements the elementary trial emulation described by Figure 3.

Now the user is ready to evaluate the impact of cross-domain noise to our testbed. The user examines three cyber subsystem settings: no upgrade, upgrade $M_1$ only, and upgrade $M_3$ only.

For each setting, for each benchmark point $X_0^i \in X$ ($i = 1, \ldots, 1,000$) and each noise level $\|N\| = I, I \in \{1, 2, \ldots, 5\}$, the user runs a campaign of $\eta = 1,000$ elementary trial emulations, and derives the cross-domain noise impact value as per Equation (9). According to Proposition 4.1, this guarantees a confidence level of 95% that the derived impact value error is within ±0.032. For the bounding filter in the physical subsystem, the user sets its radius $\gamma = 0.001m$ (see Figure 10). All the emulations are carried out on a HP workstation with Intel Core i7-3610QM and 8G RAM.

The statistics of impact values over all benchmark points are shown and compared in Figure 11.

Fig. 11. Statistics of cross-domain noise impact values $\{I(N, X_0^i)\}_{X_0^i \in X}$, without shrinking benchmark region.
As the impact value indicates the increase of plant fault probability due to cross-domain noise \( N \), the smaller the impact value, the more robust the system. Therefore, Figure 11 clearly favors upgrading \( M_1 \).

6.3 Offline Evaluation with Shrunk Benchmark Region

In Section 6.2’s evaluation, the benchmark points are sampled from the entire allowed region \( \mathcal{A} \). By applying the benchmark region shrinking methodology proposed in Section 5, the user can sample less. Specifically, using the existing LTI control Lyapunov analysis methodology (Brogan 1991), the user finds for our testbed of Equations (19), (20),

\[
P = \begin{pmatrix} Q & 0 \\ 0 & Q \end{pmatrix},
\]

where

\[
Q = \begin{pmatrix} 190.2853 & -50.0013 & 29.3842 & 10.9965 \\ -50.0013 & 436.0298 & -10.9938 & 442.5856 \\ 29.3842 & -10.9938 & 23.9030 & -50.0135 \\ 10.9965 & 442.5856 & -50.0135 & 639.884 \end{pmatrix}
\]

The user chooses \( \varepsilon = 0.0002 \), so the irrelevance distance \( \Gamma = \sqrt{\lambda_{\text{max}}(P) + \lambda_{\text{min}}(P) + \varepsilon} = 0.016 \) (see Equation (16)), which defines the shrunk benchmark region \( \mathcal{B} \) via Equation (18).

The user reuses the benchmark points used in Section 6.2, but excluding all those outside of \( \mathcal{B} \). In this way, the shrunk benchmark region \( \mathcal{B} \) removes 241 of the original 1,000 benchmark points (i.e., 24.1% of the evaluation computation effort is saved). The statistics of cross-domain noise impact values over the reduced benchmark are shown and compared in Figure 12. The results also apparently favor upgrading \( M_1 \).

6.4 Runtime Experiment Validation

Through our proposed evaluation framework, Sections 6.2 and 6.3 both come to the conclusion that the user should upgrade \( M_1 \) to \( M'_1 \). To validate the user’s decision, we carry out runtime experiments to compare the actual results of the upgrading alternatives.

Specifically, we evaluate three scenarios of the testbed. In the first scenario, no digital module is upgraded. In the second scenario, only \( M_1 \) is upgraded to \( M'_1 \). In the third scenario, only \( M_3 \) is upgraded to \( M'_3 \). For each scenario, we set the cross-domain noise level \( \|N\| \) to 1, 2, 3, 4, and 5 (see Section 6.2 and Equation (22) for the definition of these values; in our experiment implementation,
Table 1. Percentage of Trials that Encounter Plant Fault(s)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total Number of Faults</th>
<th>Faulty Trial Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Upgrade</td>
<td>49</td>
<td>49%</td>
</tr>
<tr>
<td>Upgrade $M_1$</td>
<td>20</td>
<td>20%</td>
</tr>
<tr>
<td>Upgrade $M_3$</td>
<td>39</td>
<td>39%</td>
</tr>
</tbody>
</table>

module $M_0$, a noisy camera, is realized by appending a noise generator to a high-quality camera’s output). For each noise level, 20 elementary trial experiments are carried out. In each experiment, IP$_1$ and IP$_2$ start from a random initial state uniformly picked from the allowed region $\mathcal{A}$, and run for 1 minute. We record whether during this 1 minute, IP$_1$ and IP$_2$’s state ever exceeds $\mathcal{A}$. If so, a plant fault occurs.

Table 1 lists the experiment result: the total number of plant faults and the percentage of trials that involves faults. According to the table, upgrading $M_1$ apparently performs better than upgrading $M_3$ in terms of fault reduction. This matches the prediction made by offline evaluation of Sections 6.2 and 6.3, and hence validates the usefulness of our proposed cross-domain noise impact evaluation methodology framework.

Note, as discussed in Section 5.4, the evaluation in Sections 6.2 and 6.3 is a theoretical approximation of the reality. It assumes zero delay to deliver the plant state to the cyber subsystem, and to calculate and deliver the new reference point value from the cyber subsystem to the physical subsystem. In our real-world runtime experiment, the aforementioned delay is non-zero, and is in the order of magnitude of 10ms. The fact that the runtime experiment results still match the theoretical evaluation results corroborates the following: when the delay is sufficiently small, the theoretical evaluation is good enough to increase our knowledge and confidence on the real-world 2L-CCPS.

7 CONCLUSION

In this article, we propose a framework of methodology to evaluate the impact of cross-domain noise in a generic 2L-CCPS architecture, whose cyber subsystem is a black box to the user. Our contributions are as follows:

1. We proposed a benchmark metric and corresponding measurement method to quantify the cross-domain noise impact to the black box 2L-CCPS.
2. We further proposed a method to effectively shrink the benchmark, exploiting interdisciplinary Lyapunov stability control theories.
3. We validated the effectiveness and efficiency of our proposed methodology framework with a representative 2L-CCPS testbed. Particularly, the proposed benchmark shrinking technology saves us 24.1% of the evaluation effort.

APPENDIX

A MEANING OF REACHABILITY PROBABILITY

Proposition A.1 (Risk of Trajectory). Given cross-domain noise RV $N$, suppose during $[t_0, +\infty)$, a 2L-CCPS undergoes $k$ ($k \geq 1$) reference point update events, which respectively happened at $t_0 < t_1 < \cdots < t_{k-1}$. Let $X_i$ ($i = 0, \ldots, k - 1$) denote the plant state right before the $i$th reference point update event. Let $R_i$ denote the reachability RV for $X_i$ under $N$, and $p_i = \Pr(R_i = 1)$. Let $\omega$
denote the probability that the trajectory of \( X(t) \) \((t \in [t_0, +\infty)) \) never reaches \( \tilde{A} \) (i.e., the 2L-CCPS never encounters plant fault). Then \( \omega \geq \prod_{i=0}^{k-1} (1 - p_i) \).

Proof. Starting from \( X_i \), what happens during \([t_i, t_{i+1}]\) \((i = 0, \ldots, k - 1)\), where \( t_k \overset{\text{def}}{=} +\infty \) is exactly what happens to an elementary trial starting from \( X_j \) during \([0, t_{i+1} - t_i]\) (suppose the elementary trial starts from time 0). Therefore, the probability of not reaching \( \tilde{A} \) during \([t_i, t_{i+1}]\) is no less than \((1 - p_i)\). As per Equation (3), \( \forall \gamma \geq X(t) \) is continuous on \([t_0, +\infty)\), therefore, \( \omega \geq \prod_{i=0}^{k-1} (1 - p_i) \). □

Particularly, if \( p_i \)'s are upper bounded by \( p^{\max} \), then \( \omega \geq (1 - p^{\max})^k \). In the extreme case, if \( p^{\max} = 0 \), then \( \omega = 1 \). That is, the control CPS has 0 probability of encountering a plant fault.

B PROOF OF PROPOSITION 4.2

\[
\frac{dV(X(t), O_{\text{ref}}(t))}{dt} = \begin{cases} \dot{X}^T P(X(t) - O_{\text{ref}}'(t)) + (X(t) - O_{\text{ref}}'(t))^T P X & \text{(see Equation (7))} \\ (F(X(t) - O_{\text{ref}}'(t)))^T P (X(t) - O_{\text{ref}}'(t)) & \text{(see Equation (3))} \\ (X(t) - O_{\text{ref}}'(t))^T (F^T P + P F) (X(t) - O_{\text{ref}}'(t)) & \text{(see Equation (6))} \\ -\gamma \leq 0. & \end{cases}
\]

C SHORTEST DISTANCE FROM A BALL TO A CONCENTRIC BALL COMPLEMENT

\( \forall X, Y \in \mathbb{R}^n \), denote \( \text{dis}(X, Y) \overset{\text{def}}{=} \|X - Y\|_2 = \sqrt{(X - Y)^T (X - Y)} \). We have the following:

**Lemma C.1.** Given \( \Gamma \geq \gamma > 0 \), then \( \forall X, Y \in \mathbb{R}^n \) s.t. \( X^T X \leq \gamma^2 \) and \( Y^T Y \geq \Gamma^2 \), we have \( \text{dis}(X, Y) > \Gamma - \gamma \).

Proof. Define \( f_Y(X) \overset{\text{def}}{=} (X - Y)^T (X - Y) \), let us first solve the following optimization problem:

\[
\min_X f_Y(X) \quad \text{s.t.} \quad X^T X \leq \gamma^2.
\]

For this problem, we have its Lagrangian \( L(X, \nu) = \|X - Y\|_2^2 + \nu(\|X\|_2^2 - \gamma^2) \). Using the Karush-Kuhn-Tucker (KKT) conditions, we have

\[
\|X^*\|_2 - \gamma \leq 0, \quad \nu^* \geq 0, \quad \nu^*(\|X^*\|_2 - \gamma) = 0, \quad (1 + \nu^*)X^* - Y = 0. \quad (23)
\]

Substituting \( X^* \) from Equation (25) into Equation (24), we have

\[
\nu^*(\|X^*\|_2 - \gamma) = \frac{\nu^*}{1 + \nu^*} (\|Y\|_2 - (1 + \nu^*)\gamma) = 0.
\]
As we know $Y^TY > \Gamma^2$ and $\Gamma \geq \gamma > 0$, then we have $\|Y\|_2^2 \geq \gamma > 0$. From Equation (26), we know either $v^* = 0$ or $(\|Y\|_2^2 - (1 + v^*)\gamma) = 0$. If $v^* = 0$, we have $X^* = Y$ from Equation (25), and $\|Y\|_2 = \|X^*\|_2 \leq \gamma$ from Equation (23), which contradicts the fact that $\|Y\|_2 > \gamma$. Thus, we have

$$\|Y\|_2^2 - (1 + v^*)\gamma = 0 \Rightarrow 1 + v^* = \frac{\|Y\|_2^2}{\gamma}.$$  

Substituting $(1 + v^*) = \|Y\|_2^2/\gamma$ into Equation (25), we derive

$$X^* = \frac{Y}{\|Y\|_2}.$$  

Then, we have

$$f_Y(X)^* = \left\| \frac{Y}{\|Y\|_2} Y - Y \right\|_2^2 = (\|Y\|_2^2 - \gamma^2).$$  

Here, $Y$ is a given parameter to the optimization problem. As $\|Y\|_2^2 > \Gamma \geq \gamma > 0$, we have $f_Y(X)^* = (\|Y\|_2^2 - \gamma^2) > (\Gamma - \gamma)^2$. That is, $\forall X, Y \in \mathbb{R}_n$, if $X^TXT \leq \gamma^2$, $Y^TY \geq \Gamma^2$, and $\Gamma \geq \gamma > 0$, $\text{dis}(X, Y) = \sqrt{f_Y(X)} > \sqrt{f_Y(X)^*} > \Gamma - \gamma$.  

The idea of Lemma C.1 is illustrated by Figure 13.

REFERENCES


Cross-Domain Noise Impact Evaluation for Black Box Two-Level Control CPS


Naira Hovakimyan and Chengyu Cao. 2010. Li Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation. SIAM.

Arshad Jhumka and Matthew Leeke. 2011. The early identification of detector locations in dependable software. Proc. of 
IEEE Intl. Symp. on Software Reliability Engineering.


Adam J. Oliner and Alex Aiken. 2011. Online detection of multi-component interactions in production systems. Proc. of 
Dependable Systems and Networks (DSN’11), 49–60.


Feng Tan, Liansheng Liu, Stefan Winter, Qixin Wang, Neeraj Suri, Lei Bu, Yu Peng, Xue Liu, and Xiyuan Peng. 2014. WiP abstract: A framework on profiling cross-domain noise propagation in control CPS. ACM/IEEE Intl. Conf. on Cyber-
Physical Systems (ICCPS’14), 224.

US Dept. of the Army. 2015. TM 5-698-4: Failure Modes, Effects and Criticality Analyses (FMECA) for Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Facilities.


Received February 2017; revised February 2018; accepted April 2018