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

FENG TAN, LIANSHENG LIU, STEFAN WINTER, QIXIN WANG, NEERAJ SURI, LEI BU, YU PENG, XUE LIU, XIYUAN PENG

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 paper 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.

Additional Key Words and Phrases: Lyapunov Stable, Hybrid Automata, Hybrid Model, Testing, Cyber-Physical Systems

ACM Reference Format:
Feng Tan, Liansheng Liu, Stefan Winter, Qixin Wang, Neeraj Suri, Lei Bu, Yu Peng, Xue Liu, and Xiyuan Peng, 2017. Cross-Domain Noise Impact Evaluation for Black Box Two-Level Control CPS. *ACM Trans. Cyber-Phys. Syst.* 000, 000, Article 000 (000), 24 pages.
DOI: 000.000

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 paper focuses on one aspect of the integration: how to evaluate the impact of *cross-domain noises* in control CPSs. Specifically, this paper assumes a classic control CPS architecture described by Fig. 1. It consists of a "physical" control subsystem (simplified as the "*physical subsystem*" [1] in

[1] 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 analogue. For example, when a ground computer (i.e. the

The research project related to this paper is supported in part by Hong Kong RGC GRF PolyU152164/14E, RGC ECS PolyU5328/12E, RGC Germany/HK Joint Research Scheme G-PolyU503/16, The Hong Kong Polytechnic Univ. fund A-PJ80, A-PK46, A-PL82, G-YN37, G-UA7L, G-YBMW. We thank Mr. Enyan Huang from Dept. of Computing, the Hong Kong Polytechnic Univ. and Prof. Yao Chen from Dept. of Computer Science, Southwestern Univ. of Finance and Economics, China for their contributions to the research related to this paper. We also thank anonymous reviewers and editors for their valuable comments to improve this paper.

Authors' affiliations: F. Tan and Q. Wang are with the Embedded Systems and CPS Lab, Dept. of Computing, The Hong Kong Polytechnic Univ., Hong Kong S.A.R.; S. Winter and N. Suri are with the Dept. of Computer Science, TU Darmstadt, Germany; L. Bu is with Dept. of Computer Science and Technology, Nanjing Univ., China; Y. Peng and X. Peng are with Dept. of Automatic Test and Control, Harbin Institute of Technology, China; X. Liu is with School of Computer Science, McGill Univ., Canada; L. Liu worked in three institutes throughout the duration of the project leading to this paper, respectively Dept. of Computing, The Hong Kong Polytechnic Univ., School of Computer Science, McGill Univ., and Dept. of Automatic Test and Control, Harbin Institute of Technology.
Corresponding Author: Qixin Wang, Email: csqwang@comp.polyu.edu.hk

DOI: 000.000

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 paper calls the control CPS architecture of Fig. 1 the *two-level control CPS* (2L-CCPS) architecture.

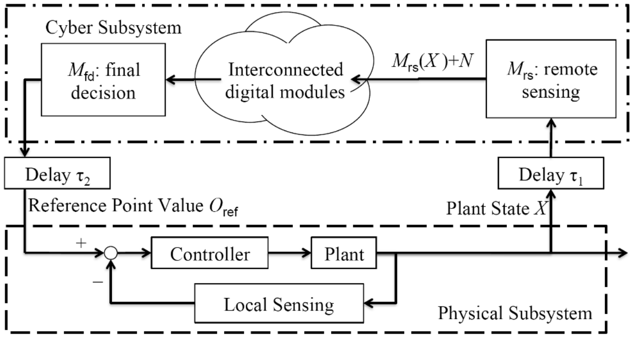

Cyber Subsystem
M_{fd}: final decision
Interconnected digital modules
$M_{\mathsf{rs}}(X)+N$
M_{rs}: remote sensing
Delay τ_2
Delay τ_1
Reference Point Value O_{ref}
Plant State X
+
−
Controller
Plant
Local Sensing
Physical Subsystem

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.

More specifically, in Fig. 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 $(x_1, x_2)^{\mathsf{T}}$ 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 $(10, 0)^{\mathsf{T}}$ commands the physical subsystem to move the cart to location 10 and stop there.

Besides the physical subsystem, the dash-dot box in Fig. 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_{rs} in Fig. 1), processes the sensed state, and decides the new reference point value. The new reference point value is the output (see M_{fd} in Fig. 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 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[2].

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; while 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*

"cyber subsystem") uses analogue wireless signals to remotely control a purely analogue (consider mechanical is a kind of analogue) drone, the "physical subsystem" (i.e. the drone) is purely analogue. However, when the ground computer uses WiFi to remotely control a WiFi+analogue drone, the "physical subsystem" (i.e. the drone) is indeed a mixture of digital and analogue parts.

[2] According to Franklin et al. [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.

paper 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 paper. Instead, this paper 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 N in 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 formulae. 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 formulae 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.

To address the challenge on how to evaluate cross-domain noise's impacts, this paper 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[3]. 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 sub-problem 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 inter-disciplinary 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 fault tolerance.

[3] Again 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 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.

Main contributions and insights of this paper are summarized as following.

(1) We propose a benchmark metric and corresponding measurement method to evaluate cross-domain noise impacts to 2L-CCPSs with black box cyber subsystems.
(2) We further propose a method to effectively shrink the benchmark, exploiting the inter-disciplinary 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.

Paper Organization

The rest of the paper 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 paper.

2. RELATED WORK

In a more general sense, this paper addresses a sub-problem of fault propagation profiling, a hot topic in system dependability research. Works of Hiller et al. [Hiller et al. 2004] propose using conditional probability to profile the permeability, exposure, and impact of faults in a network of software modules. Oliner et al. [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. [Distefano et al. 2011] propose a compositional calculus to analyse software fault propagation with closed form formulae. Jhumka et al. [Jhumka and Leeke 2011] use software fault propagation profiling results to guide the placement of fault detector assertions. Pham et al. [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. [Sierla et al. 2013] study CPS fault propagation with an explicit object-oriented and event based model. Ge et al. [Ge et al. 2009] analyse 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 paper 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 paper 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 paper's topic.

Model predictive control [Camacho and Bordons 2013] focuses on repeatedly deriving optimal control signals to control the plant. This paper, 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 paper 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 paper'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.

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 paper, 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 a n-dimensional vector $X(t) = (x_1(t), x_2(t), \ldots, x_n(t))^{\mathsf{T}}$. 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 respectively $\dot{x_i}$, $i = 1, \ldots, n$) to denote the derivative $\frac{\mathrm{d}X}{\mathrm{d}t}$ (and respectively $\frac{\mathrm{d}x_i}{\mathrm{d}t}$, $i = 1, \ldots, n$).

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

$$\begin{aligned} \frac{\mathrm{d}(X - O_{\mathsf{ref}})}{\mathrm{d}t} &= \mathbf{A}(X - O_{\mathsf{ref}}) + \mathbf{B}U, \quad (1) \\ U &= -\mathbf{K}(X - O_{\mathsf{ref}}), \quad (2) \end{aligned}$$

where $O_{\mathsf{ref}} \in \mathbb{R}_n$ 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 $X - O_{\mathsf{ref}} = 0$); $\mathbf{A} \in \mathbb{R}_{n\times n}$ and $\mathbf{B} \in \mathbb{R}_{n\times m}$ are two constant matrices dependent on the plant's physics; $U(t) = (u_1(t), u_2(t), \ldots, u_m(t))^{\mathsf{T}}$ is the controller output created as per Eq. (2); $\mathbf{K} \in \mathbb{R}_{m\times n}$ is a constant matrix defining the control strategy. Denote $\tilde{X} \overset{\mathsf{def}}{=} X - O_{\mathsf{ref}}$, the system of Eq. (1)(2) can be rewritten into the following form.

$$\dot{\tilde{X}} = \mathbf{F}\tilde{X}, \quad (3)$$

where $\mathbf{F} = \mathbf{A} - \mathbf{BK}$.

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* $\bar{\mathcal{A}} \overset{\mathsf{def}}{=} \mathbb{R}_n - \mathcal{A}$, 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 meters limit means a plant fault happens.

3.2. Cyber Subsystem Model

We assume the following about the cyber subsystem (see Fig. 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

[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.

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. 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 τ_1 (see Fig. 1); and the time cost to run the cyber subsystem and to deliver the outputted reference point value to the physical subsystem is τ_2 (see Fig. 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 Fig. 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 $M_{\mathsf{rs}}(X)+N$ to the rest of the cyber subsystem, where $M_{\mathsf{rs}}(X)$ 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 Fig. 2, denoted as H, models the combined "cyber" and "physical" aspects of 2L-CCPS.

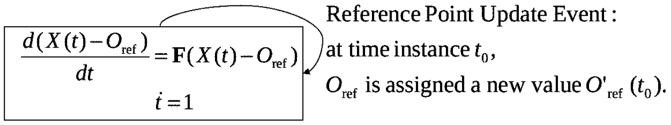

$\frac{d(X(t)-O_{\mathrm{ref}})}{dt}=\mathbf{F}(X(t)-O_{\mathrm{ref}})$
$\dot{t}=1$
Reference Point Update Event:
at time instance t_0,
O_{ref} is assigned a new value $O'_{\mathrm{ref}}(t_0)$.

Fig. 2. Hybrid automaton H that models 2L-CCPS

H's node describes the continuous behavior of the combined model. It includes Eq. (3) and the continuous increase of time: $\dot{t} = 1$. 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 $O'_{\mathsf{ref}}(t_0)$ in Fig. 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 paper. Instead, we focus on cross-domain noises (i.e., the noise denoted by RV N in Fig. 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 Fig. 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 Eq. (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 $O_{\mathsf{ref}}(t)$ $(t \in [t_0, +\infty))$. In practice, the derivative on the left hand side of Eq. (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 $O_{\mathsf{ref}}(t)$ $(t \in (t_0, +\infty))$.

Suppose the current time is $t_0 - \tau_1 - \tau_2$ (where τ_1 and τ_2 are the two delay time costs, see Fig. 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 Fig. 2), changing O_{ref} to $O'_{\mathsf{ref}}(t_0)$. 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 Appendix 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 Fig. 2), changing O_{ref} to $O'_{\mathsf{ref}}(t_0)$. 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 Fig. 1). Therefore, whether a plant fault will happen (i.e., $X(t)$ reaches the forbidden region $\bar{\mathcal{A}}$ 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* $\mathsf{Pr}(R(N, X^0) = 1)$ as $p(N, X^0)$; and consequently $\mathsf{Pr}(R(N, X^0) = 0) = 1 - p(N, X^0)$. 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 η 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 jth ($j = 1, \ldots, \eta$) elementary trial as R_j. Denote $\bar{R} \overset{\mathsf{def}}{=} \frac{1}{\eta} \sum_{j=1}^{\eta} R_j$. According to the well-known central limit theorem, when η is big enough, we can use $\bar{R}$ 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 $\delta_p \in (0, +\infty)$,

$$\text{if } \eta \geqslant \left(\frac{\Phi^{-1}(1-\frac{\alpha}{2})}{2\delta_p}\right)^2, \tag{4}$$

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

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

$$\text{Also Ineq. (4)} \Rightarrow \sqrt{\eta} \geqslant \frac{\Phi^{-1}(1-\frac{\alpha}{2})}{2\delta_p} \Rightarrow \delta_p \geqslant \frac{\Phi^{-1}(1-\frac{\alpha}{2})}{2\sqrt{\eta}}. \tag{5}$$

$$\text{Therefore, } \bar{R} \sim \text{Normal}(\mu, \sigma^2/\eta) \Rightarrow \mathsf{Pr}(|\bar{R} - \mu| \leqslant \frac{\sigma}{\sqrt{\eta}}\Phi^{-1}(1-\frac{\alpha}{2})) \geqslant 1 - \alpha$$

$$\Rightarrow \mathsf{Pr}(|\bar{R} - p| \leqslant \frac{1}{2\sqrt{\eta}}\Phi^{-1}(1-\frac{\alpha}{2})) \geqslant 1 - \alpha \qquad (\text{as } \mu = p \text{ and } \sigma \leqslant \tfrac{1}{2})$$

$$\Rightarrow \mathsf{Pr}(|\bar{R} - p| \leqslant \delta_p) \geqslant 1 - \alpha \text{ (due to Ineq. (5)).} \qquad \blacksquare$$

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

As $\bar{R}$'s realization, we have $\bar{r} = \frac{1}{\eta}\sum_{j=1}^{\eta} r_j$, where r_j is RV R_j's realization in the corresponding elementary trail. To get r_j, the simple way is to *emulate* the jth 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 Eq. (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 $\bar{\mathcal{A}}$, then $r_j = 1$; otherwise $r_j = 0$.

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

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

$$\mathbf{F}^{\mathsf{T}}\mathbf{P} + \mathbf{P}\mathbf{F} = -\mathbf{I}, \tag{6}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

Correspondingly, given control systems of Eq. (3) that are stable in the sense of Lyapunov, there are mature tools [Brogan 1991] to derive the aforementioned $\mathbf{P}$.

With $\mathbf{P}$, we can define a *Lyapunov function* $V(X(t), O_{\mathsf{ref}}(t))$ as follows.

$$V(X(t), O_{\mathsf{ref}}(t)) \overset{\mathsf{def}}{=} (X(t) - O_{\mathsf{ref}}(t))^{\mathsf{T}}\mathbf{P}(X(t) - O_{\mathsf{ref}}(t)). \tag{7}$$

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 $X(t_0) = X^0 \in \mathbb{R}_n$ and $O'_{\mathsf{ref}}(t_0) \in \mathbb{R}_n$, let $X(t)$ $(t \in [t_0, +\infty))$ be the trajectory of plant state evolved according to Eq. (3) when $O_{\mathsf{ref}}(t) \equiv O'_{\mathsf{ref}}(t_0)$, then $\forall t \in [t_0, +\infty)$,

$$\frac{\mathrm{d}\, V(X(t), O_{\mathsf{ref}}(t))}{\mathrm{d}\, t} \leqslant 0. \tag{8}$$

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 $\bar{\mathcal{A}}$, the plant state can never reach $\bar{\mathcal{A}}$ again. Based on this heuristics, we propose the algorithm of Fig. 3 to emulate the jth elementary trial ($j = 1, \ldots, \eta$), so as to approximate r_j, the realization of reachability RV R_j.

```
1.  ElementaryTrialEmulation(input: $N$, $X^0$; output: $r_j$){
2.      Input $X(t_0) = X^0$ into the cyber subsystem to generate $O'_{\mathsf{ref}}(t_0)$;
        // or equivalently, let $M_{\mathsf{rs}}$ output $M_{\mathsf{rs}}(X(t_0)) + N$ to the rest
        // of the cyber subsystem to generate $O'_{\mathsf{ref}}(t_0)$, where $X(t_0) = X^0$.
3.      Current simulator time $t \leftarrow t_0$;
4.      $O_{\mathsf{ref}} \leftarrow O'_{\mathsf{ref}}(t_0)$;
5.      while (true){
6.          Derive $X(t)$ according to Eq. (3);
7.          if ($X(t) \in \bar{\mathcal{A}}$) { $r_j \leftarrow 1$; break; }
8.          if ($V(X(t), O_{\mathsf{ref}}) < \inf_{Y \in \bar{\mathcal{A}}}\{V(Y, O_{\mathsf{ref}})\}$) { $r_j \leftarrow 0$; break; }
9.          $t \leftarrow t + \delta_t$; // $\delta_t$: per iteration simulator time increment
10.         if ($t \geqslant T_{\mathsf{sim}}$){ // $T_{\mathsf{sim}}$: maximum simulation time
11.             $r_j \leftarrow 1$; 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.

In Fig. 3, Line 7 corresponds to the case that trajectory $X(t)$ is found to reach forbidden region $\bar{\mathcal{A}}$, hence $r_j = 1$. In Line 8, as future trajectory $X(t)$'s Lyapunov function value drops below $\inf_{Y \in \bar{\mathcal{A}}}\{V(Y, O_{\mathsf{ref}})\}$, a simple proof with negation can show that due to Ineq. (8), $X(t)$ will never reach any points in $\bar{\mathcal{A}}$. Line 11 corresponds to the situation that after sufficiently long simulation, we still cannot decide if $X(t)$ reaches $\bar{\mathcal{A}}$; therefore, we pessimistically over approximate with $r_j = 1$.

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

Now we can get the η realizations $\{r_j\}$. Let $\hat{p} \overset{\mathsf{def}}{=} \bar{r} \overset{\mathsf{def}}{=} \frac{1}{\eta}\sum_{j=1}^{\eta} r_j$. As per Proposition 4.1, when η satisfies Ineq. (4), $\hat{p} = \bar{r}$ 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 $\bar{\mathcal{A}}$) 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

$$I(N, X^0) \overset{\text{def}}{=} p(N, X^0) - p(0, X^0), \tag{9}$$

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 $\mathcal{X} = \{X_i^0\}_{i=1,\ldots,b}$ 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 $\mathcal{X}$ 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 $\mathcal{X} = \{X_i^0\}_{i=1,\ldots,b}$, we summarize our basic 2L-CCPS cross-domain noise impact evaluation method as follows. Given cross-domain noise RV N, for each benchmark point $X_i^0 \in \mathcal{X}$, we run the elementary trial campaign described in Section 4.1 and 4.2 to get reachability probability $p_i(N, X_i^0)$ and $p_i(0, X_i^0)$, and follow Eq. (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 $\bar{\mathcal{A}}$, the plant trajectory may never reach $\bar{\mathcal{A}}$, 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.

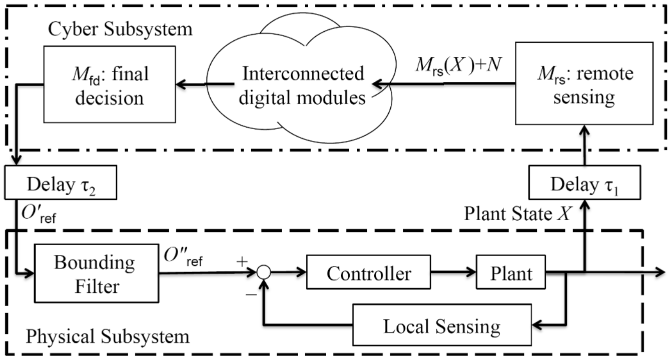


Fig. 4. Refined 2L-CCPS architecture. Note under **Assumption 5**, $\tau_1 = \tau_2 = 0$.

We refine the classic 2L-CCPS architecture of Fig. 1 by adding a *bounding filter* to the input port of the physical subsystem (see Fig. 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* $\mathsf{Ball}(X^0, \gamma)$ in the state space, centered at X^0 with radius $\gamma > 0$. If the new reference point value O'_{ref} from the cyber subsystem is within $\mathsf{Ball}(X^0, \gamma)$, then O'_{ref} is accepted. Otherwise, O'_{ref} is truncated. Formally, the filtered new reference point value O''_{ref} is

$$O''_{\text{ref}} = \begin{cases} \frac{O'_{\text{ref}} - X^0}{||O'_{\text{ref}} - X^0||_2}\gamma + X^0 & (\text{if } ||O'_{\text{ref}} - X^0||_2 \geqslant \gamma) \\ O'_{\text{ref}} & (\text{otherwise}) \end{cases} \tag{10}$$

Note Eq. (10) implies that the classic 2L-CCPS architecture (see Fig. 1) is a special case of the refined 2L-CCPS architecture (see Fig. 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 $\mathsf{Ball}(X^0, \gamma)$. 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 $\mathsf{Traj}(N, X^0)$. Denote

$$\bar{\mathcal{B}}^* \overset{\text{def}}{=} \{X^0 | X^0 \in \mathcal{A}, \text{and } \mathsf{Traj}(N, X^0) \cap \bar{\mathcal{A}} \equiv \varnothing \quad \text{for whatever RV } N\ \}.$$

Then for whatever RV N, $\forall X^0 \in \bar{\mathcal{B}}^*$, $p(N, X^0) \equiv 0$ and $I(N, X^0) \equiv 0$. Therefore, if we can explicitly identify $\bar{\mathcal{B}}^*$, then we do not need to benchmark test any point in $\bar{\mathcal{B}}^*$. A point in $\bar{\mathcal{B}}^*$ is thus an "*irrelevant benchmark point*."

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

5.2. Heuristics to Shrink Benchmark Region

Now, the question is how to find $\mathcal{B}$, or equivalently $\bar{\mathcal{B}}$, given the bounding filter (see Fig. 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 $O''_{\text{ref}}(t_0) \in \mathsf{Ball}(X^0, \gamma)$, 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

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

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

As shown by Fig. 5, *if none of such confining Lyapunov hyper-ellipsoids intersects with $\bar{A}$, then* $X^0 \in \bar{\mathcal{B}}^*$. Consequently, the set of such X^0s constitute a $\bar{\mathcal{B}} \subseteq \bar{\mathcal{B}}^*$.

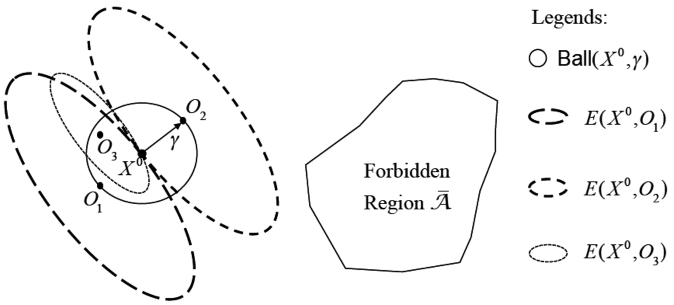


Fig. 5. Confining Lyapunov hyper-ellipsoids and forbidden region

Formally, let us define

$$V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} \overset{\mathsf{def}}{=} \sup_{\forall O''_{\mathsf{ref}} \in \mathsf{Ball}(X^0,\gamma)} \{V(X^0, O''_{\mathsf{ref}})\}, \tag{12}$$

and for arbitrary $\mathcal{Y} \subseteq \mathbb{R}_n$, define

$$V^{\mathsf{inf}}_{\mathcal{Y},\mathsf{Ball}(X^0,\gamma)} \overset{\mathsf{def}}{=} \inf_{\forall O''_{\mathsf{ref}} \in \mathsf{Ball}(X^0,\gamma)} \{V(Y, O''_{\mathsf{ref}}) | \forall Y \in \mathcal{Y}\}.$$

Then the intuition of Fig. 5 is formalized by Lemma 5.1.

LEMMA 5.1 (IRRELEVANT BENCHMARK POINT). For any state $X^0 \in \mathcal{A}$, if $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} < V^{\mathsf{inf}}_{\bar{\mathcal{A}},\mathsf{Ball}(X^0,\gamma)}$, then $X^0 \in \bar{\mathcal{B}}^*$.

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 $O''_{\mathsf{ref}}(t_0)$, is within $\mathsf{Ball}(X^0, \gamma)$. If $V^{\mathsf{inf}}_{\bar{\mathcal{A}},\mathsf{Ball}(X^0,\gamma)} > V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)}$, then the elementary trial plant state trajectory's initial Lyapunov function value $V(X(t_0), O''_{\mathsf{ref}}(t_0))$ is less than that of any state in $\bar{\mathcal{A}}$. As per Proposition 4.2, the elementary trial plant state trajectory can never reach $\bar{\mathcal{A}}$. This is true for any elementary trial starting with $X(t_0) = X^0$ under whatever RV N. Therefore $\mathsf{Traj}(N, X^0) \cap \bar{\mathcal{A}} \equiv \varnothing$ for whatever RV N. ■

5.3. Closed-Form Definition of Shrunk Benchmark Region

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

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

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

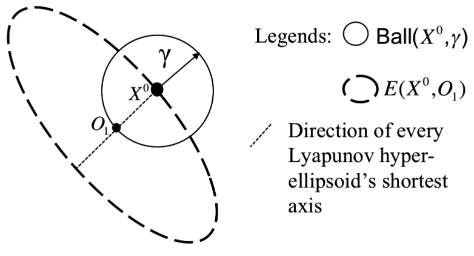


Fig. 6. Intuition of $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)}$

Fig. 6's intuition to find the closed-form formula of $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)}$ is formalized by Lemma 5.2.

LEMMA 5.2 (CLOSED-FORM VALUE OF $V^{\mathsf{SUP}}_{X^0,\mathsf{BALL}(X^0,\gamma)}$). We have $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} = \lambda^{\mathsf{max}}(\mathbf{P})\gamma^2$, where $\lambda^{\mathsf{max}}(\mathbf{P})$ is the maximal eigenvalue of $\mathbf{P}$ in Lyapunov function of Eq. (7).

Proof: According to Eq. (12), $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)}$ is the optimal objective function value for the following optimization problem:

$$\begin{aligned} \max_{O''_{\mathsf{ref}}} \quad & f_{X^0}(O''_{\mathsf{ref}}) = V(X^0, O''_{\mathsf{ref}}) = (X^0 - O''_{\mathsf{ref}})^{\mathsf{T}}\mathbf{P}(X^0 - O''_{\mathsf{ref}}) \\ \text{s.t.} \quad & (X^0 - O''_{\mathsf{ref}})^{\mathsf{T}}(X^0 - O''_{\mathsf{ref}}) \leqslant \gamma^2, \end{aligned} \tag{13}$$

where O''_{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 hyper ball, a special form of quadratic function, we can solve it as follows.

First, denote $\tilde{O}_{\mathsf{ref}} \overset{\mathsf{def}}{=} X^0 - O''_{\mathsf{ref}}$, and $f'_{X^0}(\tilde{O}_{\mathsf{ref}}) \overset{\mathsf{def}}{=} -f_{X^0}(O''_{\mathsf{ref}}) = -\tilde{O}^{\mathsf{T}}_{\mathsf{ref}}\mathbf{P}\tilde{O}_{\mathsf{ref}}$. Then problem (13) is equivalent to problem

$$\begin{aligned} \min_{\tilde{O}_{\mathsf{ref}}} \quad & f'_{X^0}(\tilde{O}_{\mathsf{ref}}) \\ \text{s.t.} \quad & \tilde{O}^{\mathsf{T}}_{\mathsf{ref}}\tilde{O}_{\mathsf{ref}} \leqslant \gamma^2. \end{aligned} \tag{14}$$

The Lagrangian of optimization problem (14) is

$$L(\tilde{O}_{\mathsf{ref}}, \nu) = \tilde{O}^{\mathsf{T}}_{\mathsf{ref}}(\nu\mathbf{I} - \mathbf{P})\tilde{O}_{\mathsf{ref}} - \nu\gamma^2,$$

and the dual function is

$$g(\nu) \;=\; \inf_{\tilde{O}_{\mathsf{ref}}}\{L(\tilde{O}_{\mathsf{ref}}, \nu)\} = \begin{cases} -\nu\gamma^2 & (\text{if } \nu\mathbf{I} - \mathbf{P} \succeq 0) \\ -\infty & (\text{otherwise}) \end{cases}$$

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

$$\begin{aligned} \max_{\nu} \quad & h \\ \text{s.t.} \quad & \nu \geqslant 0 \\ & \begin{bmatrix} \nu\mathbf{I} - \mathbf{P} & 0 \\ 0 & -\nu\gamma^2 - h \end{bmatrix} \succeq 0 \end{aligned} \tag{15}$$

As problem (14) is strictly feasible, i.e., there exists some $\tilde{O}_{\mathsf{ref}}$ (e.g. $\tilde{O}_{\mathsf{ref}} = 0$) s.t. $\tilde{O}^{\mathsf{T}}_{\mathsf{ref}}\tilde{O}_{\mathsf{ref}} < \gamma^2$, 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

$$h^* = -\lambda^{\mathsf{max}}(\mathbf{P})\gamma^2,$$

where $\lambda^{\mathsf{max}}(\mathbf{P})$ is the maximal eigenvalue of matrix $\mathbf{P}$. Then we have

$$f_{X^0}(O''_{\mathsf{ref}})^* = -f'_{X^0}(\tilde{O}_{\mathsf{ref}})^* = -h^* = \lambda^{\mathsf{max}}(\mathbf{P})\gamma^2. \qquad \blacksquare$$

Now we know that given $X^0 \in \mathcal{A}$, $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} = \lambda^{\mathsf{max}}(\mathbf{P})\gamma^2$. Then, it is possible to find a sufficient condition to make $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} < V^{\mathsf{inf}}_{\bar{\mathcal{A}},\mathsf{Ball}(X^0,\gamma)}$. 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

$$\mathsf{Dis}(X, \mathcal{Y}) \overset{\mathsf{def}}{=} \mathsf{inf}\{||X - Y||_2 | \forall Y \in \mathcal{Y}\}.$$

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

$$\mathsf{Dis}(X^0, \mathcal{Y}) > \sqrt{\frac{\lambda^{\mathsf{max}}(\mathbf{P})}{\lambda^{\mathsf{min}}(\mathbf{P})}}\gamma + \gamma + \varepsilon \overset{\mathsf{def}}{=} \Gamma, \tag{16}$$

where $\lambda^{\mathsf{max}}(\mathbf{P})$ and $\lambda^{\mathsf{min}}(\mathbf{P})$ are respectively the maximum and minimum eigenvalues of the positive definite symmetric matrix $\mathbf{P}$ of Eq. (7), then $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} < V^{\mathsf{inf}}_{\mathcal{Y},\mathsf{Ball}(X^0,\gamma)}$.

Proof: $\forall O''_{\mathsf{ref}} \in \mathsf{Ball}(X^0, \gamma), \forall Y \in \mathcal{Y}$,

$$V(Y, O''_{\mathsf{ref}}) = (Y - O''_{\mathsf{ref}})^{\mathsf{T}}\mathbf{P}(Y - O''_{\mathsf{ref}}). \tag{17}$$

Due to the bounding filter, we know that

$$(O''_{\mathsf{ref}} - X^0)^{\mathsf{T}}(O''_{\mathsf{ref}} - X^0) \leqslant \gamma^2.$$

Also, as $\mathsf{Dis}(X^0, \mathcal{Y}) > \Gamma$, we have

$$(Y - X^0)^{\mathsf{T}}(Y - X^0) > \Gamma^2.$$

From Eq. (17), we get

$$\begin{aligned} & V(Y, O''_{\mathsf{ref}}) \geqslant \lambda^{\mathsf{min}}(\mathbf{P})(Y - O''_{\mathsf{ref}})^{\mathsf{T}}(Y - O''_{\mathsf{ref}}) \\ = \ & \lambda^{\mathsf{min}}(\mathbf{P})[(Y - X^0) - (O''_{\mathsf{ref}} - X^0)]^{\mathsf{T}}[(Y - X^0) - (O''_{\mathsf{ref}} - X^0)] \\ > \ & \lambda^{\mathsf{min}}(\mathbf{P})(\Gamma - \gamma)^2 \qquad \text{(see Lemma C.1 in Appendix C)} \\ > \ & \lambda^{\mathsf{max}}(\mathbf{P})\gamma^2 + \lambda^{\mathsf{min}}(\mathbf{P})\varepsilon^2 = V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} + \lambda^{\mathsf{min}}(\mathbf{P})\varepsilon^2. \end{aligned}$$

That is, $\forall O''_{\mathsf{ref}} \in \mathsf{Ball}(X^0, \gamma)$, $\forall Y \in \mathcal{Y}$, we have $V(Y, O''_{\mathsf{ref}}) > V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} + \lambda^{\mathsf{min}}(\mathbf{P})\varepsilon^2$. Therefore, $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} < V^{\mathsf{inf}}_{\mathcal{Y},\mathsf{Ball}(X^0,\gamma)}$. ■

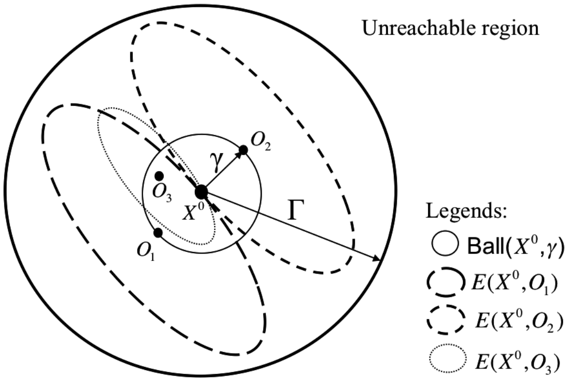


Fig. 7. Visual intuition of irrelevance distance Γ

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

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,

$$\mathcal{B} \overset{\mathsf{def}}{=} \{X^0 | X^0 \in \mathcal{A}, \text{ and } \mathsf{Dis}(X^0, \bar{\mathcal{A}}) \leqslant \Gamma\} \tag{18}$$

is a shrunk benchmark region.

Proof: $\forall X^0 \in \bar{\mathcal{B}} = \mathcal{A} - \mathcal{B}$, $\mathsf{Dis}(X^0, \bar{\mathcal{A}}) > \Gamma$. Due to Lemma 5.3, we know that $V^{\mathsf{sup}}_{X^0,\mathsf{Ball}(X^0,\gamma)} < V^{\mathsf{inf}}_{\bar{\mathcal{A}},\mathsf{Ball}(X^0,\gamma)}$. Due to Lemma 5.1, we know $X^0 \in \bar{\mathcal{B}}^*$. Therefore, $\bar{\mathcal{B}} \subseteq \bar{\mathcal{B}}^*$. That is $\mathcal{B} \supseteq \mathcal{B}^*$. ■

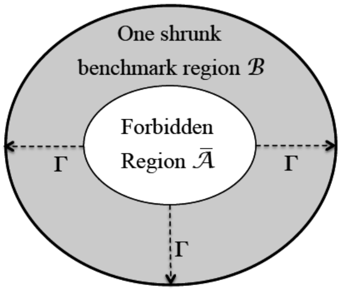

One shrunk benchmark region $\mathcal{B}$
Forbidden Region $\bar{\mathcal{A}}$
Γ
Γ
Γ

Fig. 8. A shrunk benchmark region derived via Theorem 5.4

Fig. 8 illustrates an example shrunk benchmark region derived via Theorem 5.4. Now, to build benchmark $\mathcal{X}$, instead of sampling the entire allowed region $\mathcal{A}$, we only need to sample the shrunk benchmark region $\mathcal{B}$.

5.4. Discussions on Assumption 5

So far, unless otherwise denoted, all contents of Section 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 Section 4 and 5 provides only an idealized theoretical approximation of the reality. But this does not render the theoretical evaluation results useless, becuase 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 τ_1 and τ_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 Section 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 automata based model checking useless: we still need model checking to know the real computer system better, and to trust the real computer systems more.

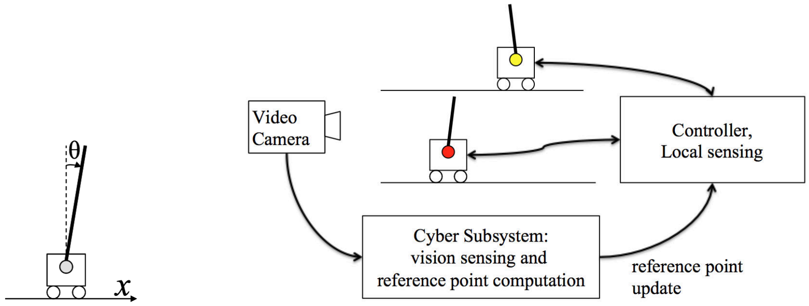
6. EVALUATION

In this section, we evaluate our proposed methodology framework in Section 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 (IP) Testbed

Our testbed is a 2L-CCPS that runs computer vision assisted parallel inverted pendulums [Brogan 1991] (see Fig. 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 Fig. 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.



(a) An inverted pendulum (b) Holistic testbed view

Fig. 9. Parallel-inverted-pendulum testbed

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

As an LTI control system[5], the physical dynamics of IP$_i$ is governed by the following systems of differential equations [Googol 2016].

$$\frac{\mathrm{d}(X_{\mathsf{ip}i} - O_{\mathsf{ipref}i})}{\mathrm{d}\,t} = \mathbf{A}_{\mathsf{ip}i}(X_{\mathsf{ip}i} - O_{\mathsf{ipref}i}) + \mathbf{B}_{\mathsf{ip}i} U_{\mathsf{ip}i},$$

$$U_{\mathsf{ip}i} = -\mathbf{K}_{\mathsf{ip}i}(X_{\mathsf{ip}i} - O_{\mathsf{ipref}i}),$$

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

where $X_{\text{ip}i}$, $O_{\text{ipref}i}$, $U_{\text{ip}i}$, $\mathbf{A}_{\text{ip}i}$, $\mathbf{B}_{\text{ip}i}$, $\mathbf{K}_{\text{ip}i}$ respectively correspond to X, O_{ref}, U, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{K}$ in Eq. (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).

$$\begin{aligned}\mathbf{A}_{\text{ip}i} &= \begin{pmatrix} 0.000 & 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 \\ 0.000 & 0.000 & 29.400 & 0.000 \end{pmatrix}, \\ \mathbf{B}_{\text{ip}i} &= (0.000, 1.000, 0.000, 3.000)^{\mathsf{T}}, \\ \mathbf{K}_{\text{ip}i} &= (-5.0505, -5.8249, 35.2502, 6.2750).\end{aligned}$$

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

$$\frac{\mathrm{d}(X_{\text{tb}} - O_{\text{tbref}})}{\mathrm{d}\,t} = \mathbf{A}_{\text{tb}}(X_{\text{tb}} - O_{\text{tbref}}) + \mathbf{B}_{\text{tb}}U_{\text{tb}}, \quad (19)$$

$$U_{\text{tb}} = -\mathbf{K}_{\text{tb}}(X_{\text{tb}} - O_{\text{tbref}}), \quad (20)$$

where $X_{\text{tb}} = \begin{pmatrix} X_{\text{ip1}} \\ X_{\text{ip2}} \end{pmatrix}$, $O_{\text{tb}} = \begin{pmatrix} O_{\text{ipref1}} \\ O_{\text{ipref2}} \end{pmatrix}$, $\mathbf{A}_{\text{tb}} = \begin{pmatrix} \mathbf{A}_{\text{ip1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\text{ip2}} \end{pmatrix}$, $\mathbf{B}_{\text{tb}} = \begin{pmatrix} \mathbf{B}_{\text{ip1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{\text{ip2}} \end{pmatrix}$, and $\mathbf{K}_{\text{tb}} = \begin{pmatrix} \mathbf{K}_{\text{ip1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\text{ip2}} \end{pmatrix}$.

Both IPs move along the x-axis. The given allowed region $\mathcal{A}$ for our testbed is[6]

$$\mathcal{A} = \{X_{\text{tb}} | X_{\text{tb}} \in \mathbb{R}_8, \text{and } 0.15 \le x_{\text{ip2}} - x_{\text{ip1}} \le 0.2\}. \quad (21)$$

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 subsyste is a white box to the vendor. Fig. 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 $O_{\text{ipref}i} = (x_{\text{ipref}i}, 0, 0, 0)^{\mathsf{T}}$. That is, at equilibrium, the inverted pendulum cart should stop at $x_{\text{ipref}i}$, 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 $x_{\text{ipref}i}$: 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 $x_{\text{ipref}i}$.

As shown in Fig. 10, the cyber subsystem's computation data flow starts from M_0, the "remote sensing" module, where a USB 2Mega 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 module M_1 and M_2 respectively for red and 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_{ip1} and x_{ip2}, the current locations of the two carts. This is realized by feeding D_1,

[6] Here we are assuming the rail of the IPs 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 Ineq. (21), regarding IP_1 and IP_2 as if moving along a same rail.

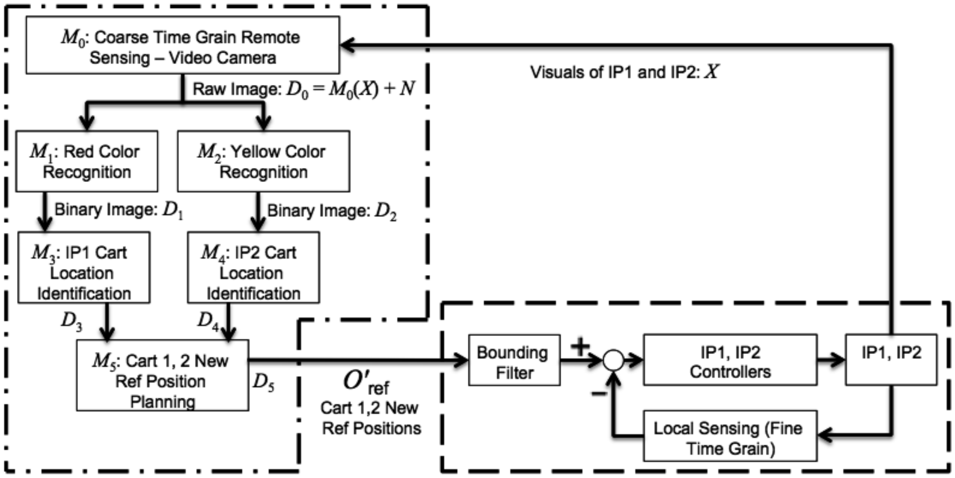

M_0: Coarse Time Grain Remote Sensing – Video Camera
Visuals of IP1 and IP2: X
Raw Image: $D_0 = M_0(X) + N$
M_1: Red Color Recognition
M_2: Yellow Color Recognition
Binary Image: D_1
Binary Image: D_2
M_3: IP1 Cart Location Identification
M_4: IP2 Cart Location Identification
D_3
D_4
M_5: Cart 1, 2 New Ref Position Planning
D_5
O'_{ref}
Cart 1,2 New Ref Positions
Bounding Filter
+
−
IP1, IP2 Controllers
IP1, IP2
Local Sensing (Fine Time Grain)

Fig. 10. Testbed cyber subsystem white box details in the vendor's view (note according to **Assumption 1** in Section 1, to the user, the cyber subsystem is a black box except M_0, M_5 and their interfaces to the rest of the cyber subsystem)

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_{ip1}; while the output of M_4 (i.e., D_4) is the estimation of x_{ip2}. D_3 and D_4 are fed to M_5, the "final decision" module, to compute the new reference point values, i.e., x_{ipref1} and x_{ipref2}.

6.2. Offline Cross-Domain Noise Impact Evaluation

In our testbed of Fig. 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 Section 4 and 5.

As summarized by the last paragraph of Section 4, the first step of the evaluation framework is to prepare a benchmark $\mathcal{X} = \{X_i^0\}_{i=1,\ldots,b}$. Without loss of generality, the user chooses $b = 1000$. For the time being, the user first tries the framework without benchmark region shrinking. That is, the user sample $b = 1000$ benchmark points from the entire allowed region $\mathcal{A}$ (see Eq. (21)).

For each benchmark point X_i^0 ($i = 1, \ldots, b$), the framework asks the user to emulate η elementary trials following the algorithm of Fig. 3. Particularly, the user implements Line 2 according to the alternative way described in the comment. That is, M_0 outputs $M_0(X_i^0) + N$ to the rest of the cyber subsystem to generate $O'_{\text{ref}}(t_0)$ (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 $X_i^0 \in \mathcal{X}$, the user prepares a high quality 640×480 pixels picture P_i as M_0's noiseless output. That is, $P_i = M_0(X_i^0)$. Let N denote the cross-domain noise RV, and $D_{0,i}$ denote the noisy output of M_0 corresponding to X_i^0. Then $D_{0,i} = M_0(X_i^0) + N = P_i + N$.

Indeed $D_{0,i}$ is also a 640×480 pixels 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, …, 640; k = 1, 2, …, 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., $N(j, k) \sim$ 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.

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

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

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

With the above methodology to generate $D_{0,i} = M_0(X_i^0) + N$ for each benchmark point X_i^0, the user implements the elementary trial emulation described by Fig. 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, upgrade M_3 only.

For each setting, for each benchmark point $X_i^0 \in \mathcal{X}$ ($i = 1, \ldots, 1000$) and each noise level $||N|| = l, l \in \{1, 2, \ldots, 5\}$, the user runs a campaign of $\eta = 1000$ elementary trial emulations, and derive the cross-domain noise impact value as per Eq. (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 $\gamma = 0.001$m (see Fig. 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 Fig.11.

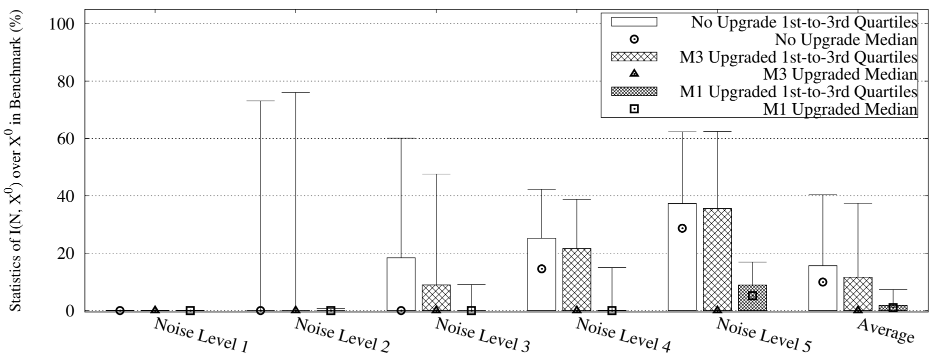


Fig. 11. Statistics of cross-domain noise impact values $\{I(N, X^0)\}_{\forall X^0 \in \mathcal{X}}$, without shrinking benchmark region

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, Fig.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 Eq. (19)(20),

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{pmatrix},$$

where

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

The user chooses $\varepsilon = 0.0002$, so the irrelevance distance $\Gamma = \sqrt{\frac{\lambda^{\max}(\mathbf{P})}{\lambda^{\min}(\mathbf{P})}}\gamma + \gamma + \varepsilon = 0.016$ (see Eq. (16)), which defines the shrunk benchmark region $\mathcal{B}$ via Eq. (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 1000 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 Fig. 12. The results also apparently favor upgrading M_1.

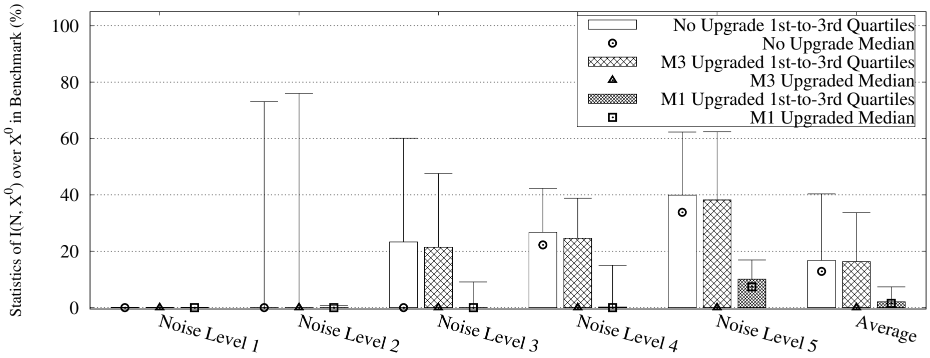


Fig. 12. Statistics of cross-domain noise impact values $\{I(N, X^0)\}_{\forall X^0 \in \mathcal{X}}$, with shrunk benchmark region

6.4. Runtime Experiment Validation

Through our proposed evaluation framework, Section 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 Eq. (22) for the definition of these values; in our experiment implementation, 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 I 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 Section 6.2 and

6.3, hence validates the usefulness of our proposed cross-domain noise impact evaluation methodology framework.

Table I. Percentage of Trials that Encounter Plant Fault(s)

Scenario	Total Number of Faults	Faulty Trial Percentage
No Upgrade	49	49%
Upgrade M_1	20	20%
Upgrade M_3	39	39%

Note as discussed in Section 5.4, the evaluation in Section 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 paper, 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:

(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 inter-disciplinary 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 ($k \geqslant 1$) reference point update events, respectively happened at $t_0 < t_1 < \ldots < t_{k-1}$. Let X_i ($i = 0, \ldots, k-1$) denote the plant state right before the ith reference point update event. Let R_i denote the reachability RV for X_i under N, and $p_i = \mathsf{Pr}(R_i = 1)$. Let ϖ denote the probability that the trajectory of $X(t)$ ($t \in [t_0, +\infty)$) never reaches $\bar{\mathcal{A}}$ (i.e., the 2L-CCPS never encounters plant fault). Then $\varpi \geqslant \Pi_{i=0}^{k-1}(1-p_i)$.

Proof: Starting from X_i, what happens during $[t_i, t_{i+1})$ ($i = 0, \ldots, k-1$, where $t_k \overset{\mathsf{def}}{=} +\infty$) is exactly what happens to an elementary trial starting from X_i during $[0, t_{i+1} - t_i)$ (suppose the elementary trial starts from time 0). Therefore, the probability of not reaching $\bar{\mathcal{A}}$ during $[t_i, t_{i+1})$ is no less than $(1-p_i)$. As per Eq. (3), $X(t)$ is continuous on $[t_0, +\infty)$, therefore, $\varpi \geqslant \Pi_{i=0}^{k-1}(1-p_i)$. ■

Particularly, if p_is are upper bounded by p^{max}, then $\varpi \geqslant (1-p^{\mathsf{max}})^k$. In the extreme case, if $p^{\mathsf{max}} = 0$, then $\varpi = 1$. That is, the control CPS has 0 probability of encountering a plant fault.

B. PROOF OF PROPOSITION 4.2

$$\begin{aligned}
& \frac{\mathrm{d}\,V(X(t), O_{\mathsf{ref}}(t))}{\mathrm{d}\,t} \quad (\text{where } O_{\mathsf{ref}}(t) \equiv O'_{\mathsf{ref}}(t_0)) \\
= \;& \dot{X}^{\mathsf{T}}\mathbf{P}(X(t) - O'_{\mathsf{ref}}(t_0)) + (X(t) - O'_{\mathsf{ref}}(t_0))^{\mathsf{T}}\mathbf{P}\dot{X} \qquad (\text{see Eq. (7)}) \\
= \;& (\mathbf{F}(X(t) - O'_{\mathsf{ref}}(t_0)))^{\mathsf{T}}\mathbf{P}(X(t) - O'_{\mathsf{ref}}(t_0)) \\
& + (X(t) - O'_{\mathsf{ref}}(t_0))^{\mathsf{T}}\mathbf{P}\mathbf{F}(X(t) - O'_{\mathsf{ref}}(t_0)) \; (\text{see Eq. (3)}) \\
= \;& (X(t) - O'_{\mathsf{ref}}(t_0))^{\mathsf{T}}(\mathbf{F}^{\mathsf{T}}\mathbf{P} + \mathbf{P}\mathbf{F})(X(t) - O'_{\mathsf{ref}}(t_0)) \\
= \;& -(X(t) - O'_{\mathsf{ref}}(t_0))^{\mathsf{T}}\mathbf{I}(X(t) - O'_{\mathsf{ref}}(t_0)) \quad (\text{see Eq. (6)}) \\
= \;& -(X(t) - O'_{\mathsf{ref}}(t_0))^{\mathsf{T}}(X(t) - O'_{\mathsf{ref}}(t_0)) \leqslant 0. \qquad \blacksquare
\end{aligned}$$

C. SHORTEST DISTANCE FROM A BALL TO A CONCENTRIC BALL COMPLEMENT

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

LEMMA C.1. Given $\Gamma \geqslant \gamma > 0$, then $\forall X, Y \in \mathbb{R}_n$ s.t. $X^{\mathsf{T}}X \leqslant \gamma^2$ and $Y^{\mathsf{T}}Y > \Gamma^2$, we have $\mathsf{dis}(X, Y) > \Gamma - \gamma$.

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

$$\begin{aligned}
\min_{X} \quad & f_Y(X) \\
\text{s.t.} \quad & X^{\mathsf{T}}X \leqslant \gamma^2.
\end{aligned}$$

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

$$\begin{aligned}
||X^*||_2 - \gamma \leqslant 0 & \qquad (23) \\
\nu^* \geqslant 0 & \\
\nu^*(||X^*||_2 - \gamma) = 0 & \qquad (24) \\
(1 + \nu^*)X^* - Y = 0 & \qquad (25)
\end{aligned}$$

Substituting X^* from Eq. (25) into Eq. (24), we have

$$\nu^*(||X^*||_2 - \gamma) = \frac{\nu^*}{1 + \nu^*}(||Y||_2 - (1 + \nu^*)\gamma) = 0 \qquad (26)$$

As we know $Y^{\mathsf{T}}Y > \Gamma^2$ and $\Gamma \geqslant \gamma > 0$, then we have $||Y||_2 > \Gamma \geqslant \gamma > 0$. From Eq. (26), we know either $\nu^* = 0$ or $(||Y||_2 - (1 + \nu^*)\gamma) = 0$. If $\nu^* = 0$, we have $X^* = Y$ from Eq. (25), and $||Y||_2 = ||X^*||_2 \leqslant \gamma$ from Eq. (23), which contradicts the fact that $||Y||_2 > \gamma$. Thus, we have

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

Substituting $(1 + \nu^*) = ||Y||_2/\gamma$ into Eq. (25), we derive

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

Then, we have

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

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

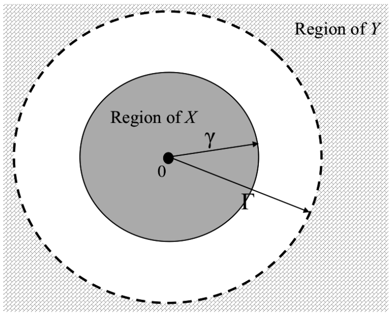


Fig. 13. Minimal distance from a ball to a concentric ball complement

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

REFERENCES

Manu Augustine, Om Prakash Yadav, Rakesh Jain, and Ajay Rathore. 2012. Cognitive map-based system modeling for identifying interaction failure modes. *Res. Eng. Design* 23:105-124 (2012).

Michael Beyeler, Florian Mirus, and Alexander Verl. 2014. Vision-Based Robust Road Lane Detection in Urban Environments. *Proc. of IEEE Intl. Conf. on Robotics and Automation (ICRA)* (May 31 - June 7 2014).

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge Univ. Press.

William L. Brogan. 1991. *Modern Control Theory (3rd Ed.)*. Prentice Hall.

Eduardo F. Camacho and Carlos Bordons. 2013. *Model Predictive Control in the Process Industry (Advances in Industrial Control)*. Springer.

Salvatore Distefano, Antonio Filieri, Carlo Ghezzi, and Raffaela Mirandola. 2011. A Compositional Method for Reliability Analysis of Workflows Affected by Multiple Failure Modes. *Proc. of CBSE* (June 2011).

Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. 1994. *Feedback Control of Dynamic Systems (3rd Ed.)*. Addison-Wesley Publishing Company.

Zhiwei Gao, Carlo Cecati, and Steven X. Ding. 2015a. A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches. *IEEE Trans. on Ind. Electronics* 62(6):3757-3767 (2015).

Zhiwei Gao, Carlo Cecati, and Steven X. Ding. 2015b. A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches. *IEEE Trans. on Ind. Electronics* 62(6):3768-3774 (2015).

Xiaocheng Ge, Richard F. Paige, and John A. McDermid. 2009. Probabilistic Failure Propagation and Transformation Analysis. *Proc. of the 28th Intl. Conf. on Computer Safety, Reliability, and Security* (2009), 215–228.

Tech. Ltd. Googol. 2016. *Linear Inverted Pendulum*. http://www.googoltech.com.

Martin Hiller, Arshad Jhumka, and Neeraj Suri. 2004. EPIC: Profiling the Propagation and Effect of Data Errors in Software. *IEEE Trans. on Computers* 53(5):1-19 (2004).

Naira Hovakimyan and Chengyu Cao. 2010. *L1 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* (2011).

Hassan K. Khalil. 2001. *Nonlinear Systems (3rd Ed.)*. Prentice Hall.

Kyekyung Kim, Joongbae Kim, Sangseung Kang, Jaehong Kim, and Jaeyeon Lee. 2012. Vision-Based Bin Picking System for Industrial Robotics Applications. *Proc. of the 9th Intl. Conf. on Ubiquitous Robots and Ambient Intelligence (URAI)* (Nov 26-29 2012), 515–516.

Weiwei Kong, Dianle Zhou, Daibing Zhang, and Jianwei Zhang. 2014. Vision-Based Autonomous Landing System for Unmanned Aerial Vehicle: A Survey. *Proc. of Intl. Conf. on Multisensor Fusion and Info. Integration for Intelligent Systems (MFI)* (Sep 28-29 2014).

Marta Kwiatkowska, Gethin Norman, and David Parker. 2002. PRISM: Probabilistic Symbolic Model Checker. *TOOLS 2002* 2324 (2002), 200–204.

Adam J. Oliner and Alex Aiken. 2011. Online Detection of Multi-Component Interactions in Production Systems. *Proc. of Dependable Systems and Networks (DSN)* (2011), 49–60.

Thanh-Trung Pham, Xavier Defago, and Quyet-Thang Huynh. 2015. Reliability prediction for component-based software systems: Dealing with concurrent and propagating errors. *Science of Computer Programming* 97 (2015), 426–457.

Lui Sha, Sathish Gopalakrishnan, Xue Liu, and Qixin Wang. 2008. Cyber-Physical Systems: A New Frontier. *IEEE SUTC* (2008), 1–9.

Seppo Sierla, Bryan M. O'Halloran, Tommi Karhela, Nikolaos Papakonstantinou, and Irem Y. Tumer. 2013. Common cause failure analysis of cyber-physical systems situated in constructed environments. *Research in Engineering Design* 24(4):375-394 (2013).

Paulo Tabuada. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer.

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)* (2014), 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*.

Xiaofeng Wang, Naira Hovakimyan, and Lui Sha. 2013. L1Simplex: fault-tolerant control of cyber-physical systems. *Proc. of ICCPS* (2013), 41–50.

Received March 2017; revised ??? ????; accepted ??? ????