

# Lyapunov Perception Contracts for Operating Design Domains

**Yangge Li**

LI213@ILLINOIS.EDU

**Chenxi Ji**

CHENXIJ2@ILLINOIS.EDU

**Jai Anchalia**

JAIA2@ILLINOIS.EDU

*University of Illinois Urbana-Champaign*

**Yixuan Jia**

YIXUANY@MIT.EDU

*Massachusetts Institute of Technology*

**Benjamin C Yang**

BCYANG2@ILLINOIS.EDU

**Daniel Zhuang**

DZHUANG6@ILLINOIS.EDU

**Sayan Mitra**

MITRAS@ILLINOIS.EDU

*University of Illinois Urbana-Champaign*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

There are two barriers to assessing the reliability of visual control systems that use machine learning (ML) models for perception and state estimation. First, the reasoning has to include the image rendering process, which is affected by environmental factors such as lighting and weather in complex ways. Second, we lack meaningful specifications for ML models like deep neural networks (DNNs). In this paper, we introduce Lyapunov Perception Contracts (LPC) as a method to address these challenges. We show how these contracts can be used as specifications for DNN-based state estimators, which assure closed-loop stability. We propose a method for synthesizing LPC from data and the models for the controller and plant dynamics. We also show how LPCs can be used to find operating design domains for visual controllers that operate in finitely parameterized environments. We illustrate applications of this method in a visual automated landing system using image data from both rendered simulations and Google Earth.<sup>1</sup>

**Keywords:** Safety of Learning-enabled systems, Asymptotic Stability, Assured Autonomy

## 1. Introduction

Visual autonomous systems take control actions using machine learning (ML) models for recognition and interpretation of images. Camera-based driverless cars and automated landing and collision avoidance systems for aircraft are exemplars of this family. Assuring safety and reliability of such systems is a challenging problem. First, any comprehensive reliability analysis of the whole system has to reason about the perception pipeline, which not only depends on the state of the system, but also external factors like lighting, fog, etc. Secondly, modular testing of perception, which is typically implemented using deep neural networks (DNNs), is not feasible because we do not have specifications for them. In this paper, we address these challenges by introducing *Lyapunov perception contracts (LPC)* and an algorithm for computing them. LPCs can be used to identify *operating design domains (ODDs)*, i.e., the range of environmental factors in which reliability of visual autonomy can be assured.

Consider a dynamical system  $\dot{x} = f(x, o(x, e))$ , with an ML-based observer  $o(x, e)$ , where  $x$  is the state and  $e$  is a vector of environmental factors like lighting, fog, etc., and a Lyapunov

---

1. Additional results and proofs provided in supplementary material <https://arxiv.org/abs/2311.08652>.

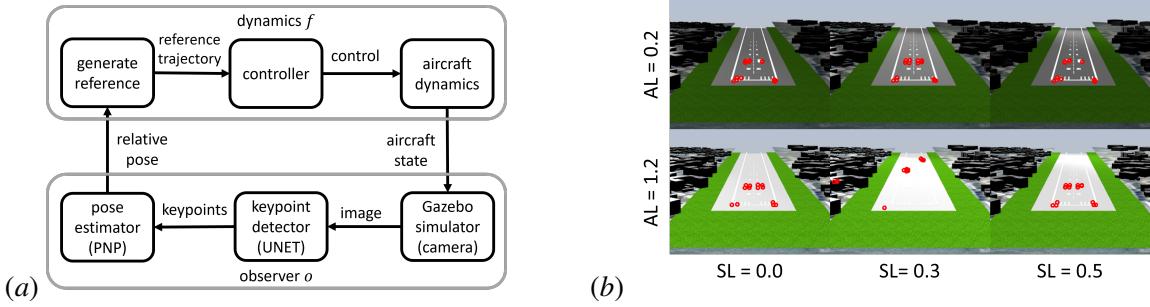


Figure 1: (a) Architecture of the vision-based automated landing system (AutoLand). (b) Camera view and keypoint detection variations under ambient and directed lighting conditions.

function  $V(x)$  for the system. Given a Lyapunov function  $V(x)$ , we aim to identify the operational design domain—the range of environmental conditions under which the vision-based system remains stable. A Lyapunov perception contract (LPC) is a set-valued observer  $M$  such that (a)  $M$  over-approximates  $o$  and (b)  $V$  is a Lyapunov function for the set-valued dynamical system  $\dot{x} \in f(x, M(x))$  that ensure stability/convergence of the vision-based system. LPCs strictly generalize the invariance-based perception contracts of Hsieh et al. (2022); Astorga et al. (2023) and we show how they can be used to find the ODDs in which the system stability can be assured.

We have developed an algorithm called the DARELPC for computing Lyapunov perception contracts. DARELPC simultaneously constructs a contract and identifies an ODD for the contract. It first constructs a candidate contract that over-approximates the observer  $o$  under an initial ODD. It then checks whether the set-valued dynamical system  $\dot{x} \in f(x, M(x))$  satisfies the Lyapunov condition using linear bound propagation tool CROWN Zhang et al. (2018). If the Lyapunov condition is not satisfied, the algorithm iteratively refines the contract and shrinks the ODD based on both the stability requirement and the sampled environment data, until a valid contract can be established. To the best of our knowledge, DARELPC is the first approach that can help automatically identify the environmental conditions of vision-based control systems, that can assure system stability.

We illustrate this method on a visual landing system: AutoLand (see Figure 1)—a challenging application with 6-dimensional nonlinear dynamics and a realistic perception pipeline. The perception pipeline estimates the *relative pose of the aircraft with respect to the touchdown area* on the runway using a combination of a DNN and a classical computer vision algorithm. The controller in AutoLand uses the estimated pose to fly the aircraft from 2300 meters—along a reference trajectory of 3° slope—down to the touchdown area. We apply DARELPC to AutoLand with image data from rendered simulations as well as Google Earth. For both data sets, DARELPC finds LPCs and corresponding ODDs where the system is shown to be stable. The identified ODDs are interesting combinations of non-convex sets of environmental conditions (Figure 2 (a)).

In summary, our contributions are: 1) We introduce Lyapunov Perception Contracts as a method assuring closed-loop stability of vision-based control systems. 2) We present an algorithm for simultaneously constructing LPCs and identifying ODDs for visual controllers operating in finitely parameterized environments. 3) We show the effectiveness of the method on a realistic automated landing system using image data from rendered simulations and Google Earth.

## 2. The Problem: Identifying Domain of Stability

### 2.1. Stability of Set-valued Dynamical Systems

We model a visual control system as a dynamical system with an observer. Given a dynamical system  $\dot{x} = f(x)$ , an initial state  $x_0$  uniquely defines a solution or an *execution* of the system, which we denote by  $x(t)$  (suppressing the dependence on  $x_0$ ). For a set-valued mapping  $F : X \rightarrow 2^X$ , consider the *set-valued dynamical system* defined by the differentiable inclusion  $\dot{x} \in F(x)$ . An execution  $x(t)$  for this system satisfies  $\dot{x}(t) \in F(x(t))$ ,  $\forall t \in \mathbb{R}_{\geq 0}$ . The distance between a point  $x \in X$  and a set  $X^* \subseteq X$  is defined as  $d_{X^*}(x) := \inf_{y \in X^*} \|x - y\|$ .

A dynamical system is *asymptotically stable with respect to a set  $X^* \subseteq X$*  if  $\forall x_0 \in X$ , the execution  $x(t)$  satisfies  $\lim_{t \rightarrow \infty} d_{X^*}(x(t)) = 0$ ; this definition extends analogously to set-valued dynamical systems. A Lyapunov function is a standard certificate for proving asymptotic stability of dynamical systems. Here we define Lyapunov functions for set-valued dynamical systems and provide a sufficient condition for asymptotical stability of such system (Theorem 6.2 from [Goebel \(2024\)](#)).

**Theorem 1** *For system  $\dot{x} \in F(x)$ , given 1) a nonempty, compact set  $X^* \subseteq X$ ; 2) a continuously differentiable function  $V : X \rightarrow \mathbb{R}_{\geq 0}$  such that  $V(x) > c$  for all  $x \in X \setminus X^*$  and  $V(x) \leq c$  for all  $x \in X^*$ ; and 3) a continuous function  $W : X \rightarrow \mathbb{R}_{\geq 0}$  such that  $W(x) = 0$  iff  $x \in X^*$ . If*

$$\nabla V(x) \cdot f_x \leq -W(x), \forall x \in X, f_x \in F(x), \quad (1)$$

*then  $X^*$  is asymptotical stable.  $V$  is called a Lyapunov function for the system.*

### 2.2. Vision-based Control System

Let  $X \subseteq \mathbb{R}^n$  be the *state space*,  $Y \subseteq \mathbb{R}^m$  be the *observation space*, and  $E$  be the *environmental space*. The evolution of a vision-based autonomous system in environment  $e \in E$  can be modeled as

$$\dot{x} = f(x, o(x, e)) \quad (2)$$

where  $f : X \times Y \rightarrow X$  is called the *dynamics*, and  $o : X \times E \rightarrow Y$  is called the *observer*. The observer  $o$  is influenced by the environment and can output different observations for the same state under different environments. Typically,  $X$  and  $f$  are available in analytical form and are amenable to formal analysis. In contrast, the environment  $E$  and the observer  $o$  are not, and we only have black-box access to  $o$ .

**Problem statement.** Given a vision-based system (2), a candidate Lyapunov function  $V$ , and a target set  $X^* \subset X$ , we are interested in finding a subset  $E_0 \subseteq E$  of the environment in which the system is asymptotically stable with respect to the set  $X^*$ .

## 3. Assurance with Lyapunov Perception Contracts

The vision-based control system is a composition of the dynamic function  $f$  and the observer  $o$ , and we are interested in developing compositional or modular solutions to the above problem. The observer  $o$  in a real system is implemented using multiple machine learning (ML) models and its

output depends on the environment in complicated ways. For example, a camera-based pose estimation system of the type used in AutoLand uses camera images which depend on lighting direction, lens flare, albedo and other factors, in addition to the actual physical position of the camera with respect to the target. Testing such ML-based observers is challenging because, in general, the observer and the ML models have no specifications (see detailed discussion on this in [Hsieh et al. \(2023\)](#); [Mitra et al. \(2025\)](#)).

### 3.1. Lyapunov Perception Contracts

A natural way of tackling this problem is to create an over-approximation  $M : X \rightarrow 2^Y$  of the observer  $o$  that turns system (2) to a set-valued dynamical system

$$\dot{x} \in f(x, M(x)) \quad (3)$$

The output of the observer  $o(x, e)$  can be made to be contained in this over-approximation  $M$ , by simply making  $M$  large or conservative. However, such an over-approximation may not be useful for establishing system stability. Thus,  $M$  has to balance observer conformance and system stability, which leads to the following notion of *Lyapunov perception contract* for the system (2).

**Definition 2** For system (2), given a target set  $X^* \subset X$  and a candidate Lyapunov function  $V$ , a mapping  $M : X \rightarrow 2^Y$  is called  $\langle E, X^* \rangle$ -Lyapunov perception contract if it satisfies:

1. Conformance:  $\forall x \in X, \forall e \in E, o(x, e) \in M(x)$ ;
2. Correctness:  $V$  is a Lyapunov function with respect to set  $X^*$  for system (3).

**Proposition 3** For system (2), if  $M$  is a  $\langle E, X^* \rangle$ -Lyapunov perception contract, then (2) is asymptotically stable with respect to  $X^*$ .

Proposition 3 follows from the fact that  $f(x, o(x, e)) \in f(x, M(x))$  over  $X$  and  $E$ . This implies that if  $V$  is a Lyapunov function for system (3), then it is also Lyapunov function for (2). Then asymptotic stability of (2) follows from Theorem 1. In this paper, we represent perception contracts  $M : X \rightarrow 2^Y$  as a ball  $B_{M_r(x)}(M_c(x)) \subseteq Y$  of radius  $M_r(x)$  centered at  $M_c(x)$ . We will next discuss the computation of  $M$ .

Regarding the construction of the Lyapunov function, we note that the controller for an visual autonomous system is usually designed first with an ideal observer that has access to ground truth values. This ideal observer is then approximately implemented using the perception pipeline. The Lyapunov function for that ideal controller can serve as an initial guess for  $V$ .

### 3.2. Learning Conformant Perception Contracts from Data

The LEARNCTR function (Algorithm 1) takes as input a set of states  $\tilde{X} \subseteq X$ , an environment set  $\tilde{E} \subseteq E$  over which a perception contract is constructed. It also takes the *user-specified target conformance measure pr*, the tolerable gap  $\epsilon$  (between the empirical and actual conformance measures), and the confidence parameter  $\delta$ . The algorithm aims to construct contracts with conformance measure higher than target. The algorithm computes the required number of samples  $n = -\frac{\ln \delta}{2\epsilon^2}$ , and then draws the samples of  $\tilde{X} \times \tilde{E}$  according to a user-specified distribution  $\mathcal{D}$ . These samples are fed to the observer to create the training set  $L_c$ . This data set  $L_c$

**Algorithm 1**


---

<b>Function</b> LEARNCTR $\tilde{T}$ $(\tilde{X}, \tilde{E}, o, pr, \epsilon, \delta)$ : <div style="margin-left: 20px;"> <math>n \leftarrow -\frac{\ln \delta}{2\epsilon^2}</math> <math>\{\langle x_i, e_i \rangle\}_{i=1}^n \leftarrow \text{SAMPLE}_{\mathcal{D}}(\tilde{X}, \tilde{E}, n)</math> <math>\hat{y}_1, \dots, \hat{y}_n \leftarrow o(x_1, e_1), \dots, o(x_n, e_n)</math> <math>L_c \leftarrow \{\langle x_1, \hat{y}_1 \rangle, \dots, \langle x_n, \hat{y}_n \rangle\}</math> <math>M_c \leftarrow \text{REGRESSION}(L_c)</math> <math>L_r \leftarrow \{\langle x_i,  \hat{y}_1 - M_c(x_i)  \rangle\}_{i=1}^n</math> <math>M_r \leftarrow \text{QUANTILEREGRESSION}(L_r, pr + \epsilon)</math> <b>return</b> <math>\langle M_c, M_r \rangle</math> </div>	<b>Function</b> DARELPC $(\tilde{X}, E, e_d, X^*, o)$ : <div style="margin-left: 20px;"> <b>Parameters:</b> <math>pr, \epsilon, \delta, V</math> <math>\tilde{E} \leftarrow E</math> <math>M_{\tilde{E}} \leftarrow \text{LEARNCTR}\tilde{T}(\tilde{X}, \tilde{E}, o, pr, \epsilon, \delta)</math> <b>while</b> <math>\text{True}</math> <b>do</b> <div style="margin-left: 20px;"> <b>if</b> CHECKLYAP<math>(\tilde{X}, M_{\tilde{E}}, V, X^*)</math> <b>then</b> <div style="margin-left: 20px;"> <b>break</b> </div> <math>\tilde{E} \leftarrow \text{SHRINKENV}(\tilde{E}, e_d)</math> <math>M_{\tilde{E}} \leftarrow \text{LEARNCTR}\tilde{T}(\tilde{X}, \tilde{E}, o, pr, \epsilon, \delta)</math> </div> <b>return</b> <math>\langle M_{\tilde{E}}, \tilde{E} \rangle</math> </div>
---	--

---

is used to generate a center function  $y = M_c(x)$  using regression. The radius function  $M_r$  is obtained using QUANTILEREGRESSION Dekkers (2014) such that  $pr + \epsilon$  fraction of data points in  $L_r$  satisfy that  $|y_i - M_c(x_i)| \leq M_r(x_i)$ , which gives the empirical conformance measure  $\hat{p} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\hat{y}_i \in M(x_i))$  for the obtained  $M$  where  $\mathbb{I}$  is the indicator function. Using Hoeffding's inequality Hoeffding (1963), it follows that the real conformance measure of the constructed perception contract  $M$  is at least  $pr$  with probability at least  $1 - \delta$ .

**Proposition 4** *With probability at least  $1 - \delta$ , the actual conformance measure  $p = \mathbb{E}_{x, e \sim \mathcal{D}}[\mathbb{I}(o(x, e) \in M(x))]$  of the computed perception contract  $M$  is at least  $pr$ .*

For certain environments  $e \in \tilde{E}$ , the system (2) may not be stable, and therefore, even a conformant contract will not establish correctness. In Section 3.3, we will develop techniques for shrinking the environment over which contracts are correct.

### 3.3. Requirement Guided Refinement of Perception Contracts

We now present the Data and Requirement Guided Lyapunov Perception Contract (DARELPC) algorithm (Algorithm 1), which uses LEARNCTR $\tilde{T}$  to find a correct Lyapunov perception contract and applies CHECKLYAP to verify asymptotic stability.

**CHECKLYAP** $(\tilde{X}, M_{\tilde{E}}, V, X^*)$  The function CHECKLYAP takes as input a set of states  $\tilde{X}$ , a candidate perception contract  $M_{\tilde{E}}$ , a candidate Lyapunov function  $V$ , and a target set  $X^*$ . The function first partitions  $\tilde{X}$  into fine grids and computes the upper bound of time derivative of  $V$  for each partition via linear bound propagation tool CROWN Zhang et al. (2018). The function then checks whether  $\tilde{X} \setminus X^*$  intersects with the region where  $\dot{V}$ 's upper bound is none negative. If no such intersection exists, it implies that all states of  $\tilde{X} \setminus X^*$  have a negative Lyapunov time derivative for the system  $\dot{x} \in f(x, M(x))$ . According to Theorem 1, this means  $V$  is a valid Lyapunov function with respect to  $X^*$ . Otherwise, asymptotic stability is not guaranteed.

Verifying the Lyapunov condition over the entire state space is challenging, especially since the perception error—and thus the radius of the perception contract—can become too large for the condition to hold. Therefore, DARELPC takes as input the region of interest  $\tilde{X} \subseteq X$ , and environment  $E_0$  sets, the target set  $X^*$ , the parameters of LEARNCTR $\tilde{T}$  and the candidate Lyapunov function. DARELPC confirms the correctness requirement of the computed Lyapunov perception contract by checking if  $V$  is a Lyapunov function of the system (3) with respect to set  $X^*$  using CHECKLYAP.

If  $V$  is indeed a Lyapunov function with respect to  $X^*$  for system (3) then a Lyapunov perception contract  $M_{\tilde{E}}$  is found and the corresponding range of environment is returned. Otherwise, the algorithm shrinks the range of allowed environment  $\tilde{E}$  via the SHRINKENV. SHRINKENV takes a set of environmental parameters  $\tilde{E}$  and generates a new set of environmental parameters. The sequence of sets generated by repeatedly apply SHRINKENV will monotonically converge to a singleton set containing only a nominal environment  $e_d$ . DARELPC is sound, in the sense that:

**Theorem 5** *The output  $M_{\tilde{E}}$  of Algorithm DARELPC is a  $\langle \tilde{E}, X^* \rangle$ -LPC for system  $\dot{x} = f(x, o(x, e))$ .*

Under the assumption that vision-based system is asymptotically stable under nominal environment  $e_d$ , we are also able to show that DARELPC is complete.

## 4. Case Study: Visual Auto Landing

AutoLand is a vision-based auto-landing system for fixed-wing aircraft. The system consists of a DNN-based pose estimator and a controller that uses the estimated pose to guide the aircraft along a reference trajectory with constant slope (See Figure 1). The system dynamics (2) is defined in terms of error dynamics with respect to this reference trajectory. AutoLand-like systems are being developed for various types of aircraft to land in GPS-denied scenarios and airports without expensive Instrument Landing Systems (ILS). The target set  $X^*$  is a bounded region where the error between aircraft state and reference state should be less than 3.6m (FAA). For our analysis, we developed a detailed simulation of the aircraft dynamics, the controller, and the perception pipeline implemented with visual pose estimation.

### 4.1. AutoLand control system with DNN-based pose estimator

The state space  $X$  of the AutoLand system is defined by the state variables  $(\mathbf{x}, \boldsymbol{\theta})$ , where  $\mathbf{x} = (x, y, z)$ ,  $\boldsymbol{\theta} = (\phi, \theta, \psi)$  (roll,pitch,yaw) are position and orientation of aircraft. The perception pipeline functions as an observer, denoted by  $o$ , which estimates the aircraft's positional states relative to the touchdown point. Specifically, the observation space  $Y$  is defined by the estimated position  $(\hat{x}, \hat{y}, \hat{z})$ . The orientation  $\boldsymbol{\theta}$  are excluded from the observation space, as they are directly measured by onboard sensors, such as gyroscopes, rather than inferred from vision.

**Dynamics.** In this work, we use the simplified aircraft dynamics from Nakamura and Savant (1992)

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\theta}} \end{pmatrix} = \begin{pmatrix} b_1 & 0 \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (4)$$

where  $u_1(t) = v(t)$  is the velocity of the aircraft and  $u_2(t) = \mathbf{w}(t) = (w_x, w_y, w_z)$ , represents the angular velocity of the aircraft and

$$b_1(\boldsymbol{\theta}) = (\cos \psi \cos \theta, \sin \psi \cos \theta, -\sin \theta)^T \quad (5)$$

$$B_2(\boldsymbol{\theta}) = \begin{pmatrix} 1 & \sin \phi \cos \theta & \cos \phi \sin \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix} \quad (6)$$

A reference trajectory defined by reference position  $\mathbf{x}_r$ , reference orientation  $\boldsymbol{\theta}_r$ , reference velocity  $v_r$  and reference angular velocity  $\mathbf{w}_r$  is given to guide aircraft to safely land toward the runway. This reference approach trajectory begins 2.3 kilometers from the touchdown point at an altitude of 120 meters, with a descend slope of  $-3^\circ$  in the z direction, and maintains a constant speed of  $10m/s$ .

**Lyapunov function and Controller** Denote  $B_{2r} = B_2(\boldsymbol{\theta}_r)$  be  $B_2$  matrix taking reference angles;  $R$  be the rotation matrix from the local frame to inertial frame. The error state is derived as  $\mathbf{x}_e = R^T(\mathbf{x}_r - \mathbf{x})$  for positional error and  $\boldsymbol{\theta}_e = \boldsymbol{\theta}_r - \boldsymbol{\theta}$  for rotational error.

We adopt the control policy from Nakamura and Savant (1992) by replacing actual error states  $(x_e, y_e, z_e)$  with the observed error states  $(\hat{x}_e, \hat{y}_e, \hat{z}_e)$  as

$$u_1 = v_r \cos \psi_e \cos \theta_e - \gamma^2 \hat{x}_e \quad (7)$$

$$u_2 = \mathbf{w}_r + B_2^{-1}\{(B_{2r} - B_2)\mathbf{w}_r + p_e + \hat{q}\} \quad (8)$$

where  $\gamma = 2$ ,  $p_e = (k_1 \sin \phi_e, k_2 \sin \theta_e, k_3 \sin \psi_e)^T$ ,  $\mathbf{k} = (k_1, k_2, k_3)^T = (5, 5, 5)^T$  and  $\hat{q} = (0, -\frac{\hat{z}_e v_r}{k_2}, \frac{\hat{y}_e v_r \cos \theta_e}{k_3})^T$ .

We consider the following Lyapunov function:

$$V = \frac{1}{2} \mathbf{x}_e^T \mathbf{x}_e + \mathbf{k}^T f(\boldsymbol{\theta}_e) \quad (9)$$

where  $f(\boldsymbol{\theta}_e) = (1 - \cos \phi_e, 1 - \cos \theta_e, 1 - \cos \psi_e)^T$ . From the requirement that the error between aircraft state and reference state is less than 3.6m, we formally define  $X^* = \{(\mathbf{x}_e, \boldsymbol{\theta}_e) | V(\mathbf{x}_e, \boldsymbol{\theta}_e) \leq 6.48\}$  and choose  $W(\mathbf{x}_e, \boldsymbol{\theta}_e) = 0.001$  for all  $(\mathbf{x}_e, \boldsymbol{\theta}_e) \in X \setminus X^*$ .

**Vision-based observer.** The observations  $(\hat{x}, \hat{y}, \hat{z})$  are computed using a realistic perception pipeline composed of three functions (Figure 1): (1) The airplane's front *camera* generates an image. (2) A *key-point detector* detects specific key-points on the runway in that image using U-Net Ronneberger et al. (2015). In this case study, we use the U-Net implementation from Alexandre (2013). (3) *Pose estimator* uses the detected key-points and the prior knowledge of the actual positions of those key-points on the ground to estimate the pose of the aircraft. This last calculation is implemented using the perspective-n-point algorithm Fischler and Bolles (1981). Apart from the impact of the environment on the image, the observer is deterministic.

**Environment.** We consider two different environments: 1) a simulated environment rendered using Gazebo and 2) Google Earth data for Heathrow and O'Hare airports pulling using CesiumJS. For the Gazebo simulation, we study the effects of the ambient light intensity in  $[0.2, 1.2]$  and the position of the sun, represented by a spotlight with angle  $[-0.1, 0.6]$ . The effect of different environmental parameters is shown in Figure 2. These ranges are chosen to include a nominal environment  $e_d = (1.0, 0)$ . For the data from Google Earth, we choose ambient light intensity between  $[0.6, 1]$  and snow level  $[0, 0.4]$  as  $E_0$ . The effect of these environmental parameters is shown in Figure 3. The nominal environment in this case is  $e_d = (1.0, 0.0)$  corresponding to no change in lighting and no snow.

## 4.2. Safety Analysis of AutoLand with Gazebo Data

We first apply DARELPC on the AutoLand system for data from Gazebo simulator. The algorithm outputs an environmental condition  $\tilde{E} \subset E$  and a corresponding  $\langle \tilde{E}, X^* \rangle$ -Lyapunov perception

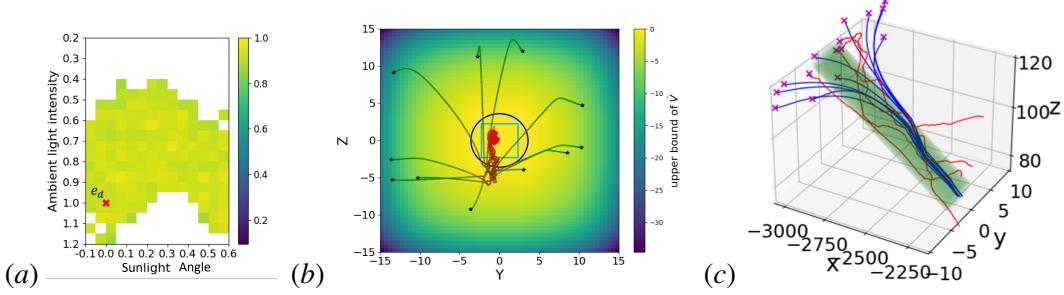


Figure 2: (a): Colored region represent operational design domain identified by DARELPC. (b): Heat map of upper bound of Lyapunov function’s time derivative. Simulated trajectories under  $\tilde{E}$  (green-red) converge to region with non-negative Lyapunov time derivative (cyan rectangle) contained in  $X^*$  (blue circle). (c): Simulation trajectories under  $\tilde{E}$  (blue) converge to  $X^*$  (green tube), trajectories under  $E \setminus \tilde{E}$  (red) is unstable.

contract  $M_{\tilde{E}}$ , after performing 12 rounds of refinements. The resulting  $\tilde{E}$  is visualized in Figure 2. In the figure, the white region represents the environmental parameters that are removed (not provably stable) during shrinking. For the colored region, different colors represent the empirical conformance of  $o$  to  $M$  over each environmental parameter partition.

*Computed perception contract achieves target conformance measure.* To compute the perception contract, we set target conformance measure  $pr = 90\%$ , the gap between actual conformance and empirical conformance  $\epsilon = 1\%$ , and the confidence parameter  $\delta = 0.1\%$ . The empirical conformance measure should be at least 91% for approximately 34K samples, which we round up to 40K. The resulting perception contract  $M_{\tilde{E}}$  achieves a 92.4% empirical conformance measure over  $\tilde{E}$ . Furthermore, 30 random simulations under  $\tilde{E}$  yield a 98.6% empirical conformance measure, demonstrating that  $M_{\tilde{E}}$  not only meets but can exceed the required conformance for certain realistic sampling distributions.

DARELPC finds nontrivial ODD. As shown in Figure 2 (a), when the ambient light intensity is below 0.6 or above 1.15, the perception performance degrades, and these parts of the environment  $E$  removed partitions under these conditions. Interestingly, around  $[0.95, 1.1] \times [0.25, 0.4]$  the ambient conditions are near nominal, yet the perception performance is poor. This can be explained as the glare from the low sun overwhelms the keypoint detector. Around  $[0.4, 0.6] \times [0.25, 0.4]$ , on the other hand, the sun’s light compensates for low ambient light, maintaining reliable perception performance for landing.

*Correctness and system-level stability.* As shown in Fig 2 (b), the blue circle with a radius of 3.6 represents the target set  $X^*$  on  $y - z$  subspace. The color bar indicates that all states in  $X \setminus X^*$  has a negative value of Lyapunov function’s time derivative, meaning that any trajectory starts from  $X \setminus X^*$  will converge into  $X^*$ . We run 10 simulations under  $\tilde{E}$ , all simulated trajectories (shown as green-red lines) converge to the target set  $X^*$  (the blue circle). In Figure 2 (c), the same trajectories (blue lines) converge into  $X^*$  (the green tube); while the sampled trajectories under invalid environment  $E \setminus \tilde{E}$  (red lines) are not guaranteed to converge into  $X^*$ .

#### 4.3. Safety Analysis of AutoLand on Google Earth Data

To further show generality of our analysis, we analyze AutoLand with data generated from Google Earth for actual airports. We use the same  $X^*$  used for Gazebo data. The result is shown in Figure 4. The identified range of environment that AutoLand is stable is shown in Figure 4 (a). The



Figure 3: Heathrow and O’Hare International runway images pulled using CesiumJS under different environmental conditions.

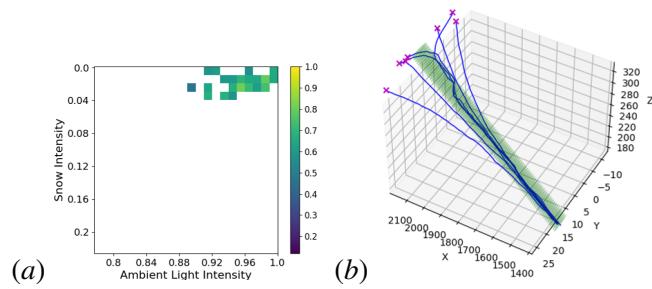


Figure 4: (a): Identified ODD for Google Earth data. (b): Simulated trajectories under  $\tilde{E}$  converge to  $X^*$  (green tube).

perception pipeline is sensitive on Google Earth data and small perturbation from the environment can drastically change the perception result. The identified environment is intuitive: as the ambient light reduces or the level of snow increases, we can no longer proof stability of the system for  $X^*$  and the system will converge to a much larger set or even becomes unstable. In this paper, we analyze the perception pipeline as is, leaving its improvement based on our analysis for future work. Figure 4 (b) shows random simulated trajectories under  $\tilde{E}$  (Blue trajectories). We can see under  $\tilde{E}$ , all simulations properly converge to  $X^*$ . However, under  $E \setminus \tilde{E}$ , the system can deviate from  $X^*$ .

#### 4.4. Convergence and conformance of Perception Contracts

For termination of Algorithm 1, we assume that the perception contract we computed is convergent, i.e. as candidate contracts are computed for a sequence of environments converging to  $e_d$ , the contracts also converge to  $o$  (pointwise). To test this, we pre-sampled 80k data points and set  $pr = 85\%$ ,  $\delta = 0.1\%$ . We reduce  $\tilde{E}$  for  $M_{\tilde{E}}$  by invoking SHRINKENV function 0, 3, and 6 times on the set of environments with pre-sampled data points. SHRINKENV reduces the range of environments by removing environmental parameters where the observations is far from the actual states and removing corresponding data points sampled under these environments. The result is shown in Figure 5. We observe that after 0,3,6 round of shrinking  $o(x, e_d) \in M_{\tilde{E}_6}(x) \subseteq M_{\tilde{E}_3}(x) \subseteq M_{\tilde{E}_0}(x)$ , which provides evidence that the perception contract is convergent.

As we reduce the range of environmental parameters with 0, 3 and 6 shrinking, the training data set reduced from 80000 to 71215 and 62638, respectively. Therefore, according to Proposition 4, the empirical conformance measure should at least be 85.8% for all these three cases to achieve the 85% actual conformance measure. In practice, we are able to get 85.9%, 87.5% and 89.1% empirical conformance measure. From this, we can see that refinement preserves the conformance measure of the perception contract.

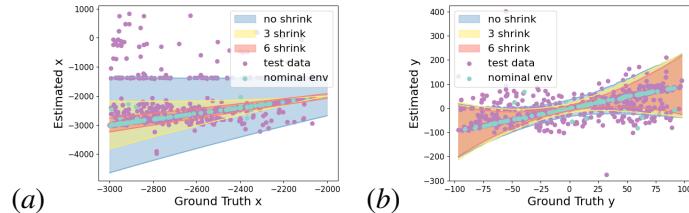


Figure 5: Visualizing perception contract with different number of shrinking for different dimensions.

## 5. Related Works

Assuring safety of autonomous systems that use ML models for perception has attracted the attention of several researchers. A recent survey [Mitra et al. \(2025\)](#) provide an extended overview between different techniques.

In [Păsăreanu et al. \(2023a\)](#), the authors characterize the performance of finite-valued observers (classifiers) using *confusion matrices*. The behavior of the observer is approximated using probability associated with each estimated output, which then coupled with a probabilistic system model to check system-level properties using probabilistic model checking. In [Păsăreanu et al. \(2023b\)](#), the same author tackles the problem by deriving weakest assumptions for the observer using system specifications. The weakest assumption describe the minimal conditions required for a component of a system to interact correctly and safely with other components and is derived solely based on the system's functional specifications using backward reachability analysis from error states.

In [Yang et al. \(2023\)](#), the author uses idea from conformal prediction [Shafer and Vovk \(2008\)](#); [Tibshirani et al. \(2019\)](#) to perform safe perception-based control under stochastic sensor uncertainty. Conformal predictions construct a prediction region such that, with user specified probability, the difference between ground truth and estimation will be bounded by the constructed prediction region. As in [Dixit et al. \(2024\)](#), the authors utilize conformal prediction to develop a calibrated perception system, which ensures that the misdetection rate caused by environmental changes remains below a user-chosen threshold for object detection tasks. This line of research is different from perception contracts as the prediction guarantees do not account for system-level specifications, such as stability.

In [Santa Cruz and Shoukry \(2022\)](#), the authors adopt a unique approach by encoding the entire system's modules, including the camera, into neural networks. Similarly, in [Habeeb et al. \(2024\)](#), the authors modeled uncertainty in image generation process from camera pose using interval images. This together with the down stream components are then analyzed using neural network model checkers inlcuding [Khedr et al. \(2021\)](#); [Zhang et al. \(2018\)](#).

Finally, we discuss *perception contracts* introduced in [Hsieh et al. \(2022\)](#); [Astorga et al. \(2023\)](#) and developed further in this paper. The system modeling language uses continuous states and discrete time, with relational (set-valued) uncertainty. The perception contract is created from ground truth labeled data and system safety properties. It serves a specification for modular testing of the ML perception components. The idea was recently extended to distributed systems to assure safety of vision-based swarm formation control [Hsieh et al. \(2023\)](#).

## 6. Discussions

The paper advances method of perception contract by introducing Lyapunov Perception Contract as a method that ensures closed loop stability of system under uncertainty from perception. We further develop an algorithm called DARELPC that finds the operational design domains for vision based control system and LPC simultaneously. We apply the method to analyze a vision-based autolanding system that uses multi-stage ML-enabled perception and high-dimensional nonlinear dynamics.

The conformance and completeness provided by DARELPC relies on assumption that underlying sampling distribution  $\mathcal{D}$  for constructing the contract remains the same under deployed conditions. Future work will need to address detection and adaptation of distributional shifts in the perception data. Another important research direction is synthesis of controller and perception pipeline, aiming to guarantee the asymptotic stability of systems with perception uncertainty, using Lyapunov perception contracts.

## Acknowledgments

This material is based upon work supported by The Boeing Company. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of The Boeing Company.

## References

- Milesi Alexandre. Pytorch-unet. <https://github.com/milesial/Pytorch-UNet>, 2013.
- Angello Astorga, Chiao Hsieh, P. Madhusudan, and Sayan Mitra. Perception contracts for safety of ml-enabled systems. *Proc. ACM Program. Lang.*, 7(OOPSLA2), October 2023. doi: 10.1145/3622875. URL <https://doi.org/10.1145/3622875>.
- Gijs Dekkers. Quantile regression: Theory and applications, 2014.
- Anushri Dixit, Zhiting Mei, Meghan Booker, Mariko Storey-Matsutani, Allen Z. Ren, and Anirudha Majumdar. Perceive with confidence: Statistical safety assurances for navigation with learning-based perception, 2024.
- Federal Aviation Administration (FAA), U.S. Department of Transportation, and Aviation Supplies & Academics (ASA). *Instrument Rating Practical Test Standards for Airplane, Helicopter and Powered Lift* (2024): Faa-S-8081-4e. Practical Test Standards Series. Aviation Supplies & Academics, Incorporated, 2010. ISBN 9781560277798. URL <https://books.google.com/books?id=jpuoSQAACAAJ>.
- Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL <https://doi.org/10.1145/358669.358692>.
- R. Goebel. *Set-valued, Convex, and Nonsmooth Analysis in Dynamics and Control: An Introduction*. Other titles in applied mathematics. Society for Industrial and Applied Mathematics, 2024. ISBN 9781611977974. URL <https://books.google.com/books?id=To6z0AEACAAJ>.
- P. Habeeb, Deepak D’Souza, Kamal Lodaya, and Pavithra Prabhakar. Interval image abstraction for verification of camera-based autonomous systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(11):4310–4321, 2024. doi: 10.1109/TCAD.2024.3448306.
- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459. URL <http://www.jstor.org/stable/2282952>.
- Chiao Hsieh, Yangge Li, Dawei Sun, Keyur Joshi, Sasa Misailovic, and Sayan Mitra. Verifying controllers with vision-based perception using safe approximate abstractions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4205–4216, 2022.

Chiao Hsieh, Yubin Koh, Yangge Li, and Sayan Mitra. Assuring safety of vision-based swarm formation control, 2023.

Haitham Khedr, James Ferlez, and Yasser Shoukry. Peregrinn: Penalized-relaxation greedy neural network verifier. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification*, pages 287–300, Cham, 2021. Springer International Publishing. ISBN 978-3-030-81685-8.

Sayan Mitra, Corina Păsăreanu, Pavithra Prabhakar, Sanjit A. Seshia, Ravi Mangal, Yangge Li, Christopher Watson, Divya Gopinath, and Huafeng Yu. *Formal Verification Techniques for Vision-Based Autonomous Systems – A Survey*, pages 89–108. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-75778-5. doi: 10.1007/978-3-031-75778-5\_5. URL [https://doi.org/10.1007/978-3-031-75778-5\\_5](https://doi.org/10.1007/978-3-031-75778-5_5).

Y. Nakamura and S. Savant. Nonlinear tracking control of autonomous underwater vehicles. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages A4–A9 vol.3, 1992. doi: 10.1109/ROBOT.1992.220005.

Corina S. Păsăreanu, Ravi Mangal, Divya Gopinath, Sinem Getir Yaman, Calum Imrie, Radu Calinescu, and Huafeng Yu. Closed-loop analysis of vision-based autonomous systems: A case study. In Constantin Enea and Akash Lal, editors, *Computer Aided Verification*, pages 289–303, Cham, 2023a. Springer Nature Switzerland. ISBN 978-3-031-37706-8.

Corina S. Păsăreanu, Ravi Mangal, Divya Gopinath, and Huafeng Yu. Assumption generation for learning-enabled autonomous systems. In Panagiotis Katsaros and Laura Nenzi, editors, *Runtime Verification*, pages 3–22, Cham, 2023b. Springer Nature Switzerland. ISBN 978-3-031-44267-4.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Ulices Santa Cruz and Yasser Shoukry. Nnlander-verif: A neural network formal verification framework for vision-based autonomous aircraft landing. In Jyotirmoy V. Deshmukh, Klaus Havelund, and Ivan Perez, editors, *NASA Formal Methods*, pages 213–230, Cham, 2022. Springer International Publishing. ISBN 978-3-031-06773-0.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9: 371–421, jun 2008. ISSN 1532-4435.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. *Conformal prediction under covariate shift*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Shuo Yang, George J. Pappas, Rahul Mangharam, and Lars Lindemann. Safe perception-based control under stochastic sensor uncertainty using conformal prediction. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6072–6078, 2023. doi: 10.1109/CDC49753.2023.10384075.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems*, 31:4939–4948, 2018. URL <https://arxiv.org/pdf/1811.00866.pdf>.