Persistently Feasible Robust Safe Control by Safety Index Synthesis and Convex Semi-Infinite Programming

Tianhao Wei[1], Shucheng Kang[2], Weiye Zhao[1], and Changliu Liu[1]

Abstract—Model mismatches prevail in real-world applications. Ensuring safety for systems with uncertain dynamic models is critical. However, existing robust safe controllers may not be realizable when control limits exist. And existing methods use loose over-approximation of uncertainties, leading to conservative safe controls. To address these challenges, we propose a control-limits aware robust safe control framework for bounded state-dependent uncertainties. We propose safety index synthesis to find a robust safe controller guaranteed to be realizable under control limits. And we solve for robust safe control via Convex Semi-Infinite Programming, which is the tightest formulation for convex bounded uncertainties and leads to the least conservative control. In addition, we analyze when and how safety can be preserved under unmodeled uncertainties. Experiment results show that our robust safe controller is always realizable under control limits and is much less conservative than strong baselines.

I. INTRODUCTION

Safety is critical in robotic systems. Safe control, as the last defense of the system, ensures real-time safety by keeping the system state in a safe set (known as forward invariance) [1]. Energy-function based methods were proposed to realize safe control [2]. With the energy function (also called safety index, barrier function, and Lyapunov functions), the safe control problem is converted to online quadratic programming (QP). QP-based safe control has been widely studied for deterministic dynamic models [2].

However, dynamic models are approximations of the real-world systems [3]. There can always be uncertainties. Safe control has been extended to *uncertain dynamic models* (UDM) [4], [5]. However, existing works usually assume the robust safe control is always realizable (also called *persistently feasible*) [6], [7] or there is no control limit [8], [4], [5]. It remains unclear how to design a persistently feasible robust safe controller under control limits.

Another challenge of robust safe control is how to solve the QP-based safe control with model uncertainty and control limits. There can be infinitely many constraints in such a QP because each possible dynamic model leads to one constraint. Therefore, existing methods use different ways to reduce the constraints by over-approximating the uncertainties. Some methods upper bound the residual time derivative of the barrier function induced by uncertainty [9], [10], [11]. However,

sometimes it is unclear how to choose the bound. And the bound is a coarse over-approximation because it is often state-independent or control-independent, leading to conservative safe controls. Some methods formulate robust safe control as Second Order Cone Programming (SOCP) [4], [5]. However, as we will show in the experiment (fig. 4 and fig. 5), SOCP formulation can greatly over-approximate the uncertainty leading to infeasible control, or lead to safety violations when the uncertainty is not Gaussian. The presence of control limits requires a tight approximation of the uncertainty. Otherwise, the desired control may exceed the control limits due to over-approximation of the uncertainty. Besides, the modeled uncertainty may not be accurate. It is unclear what happens when the actual uncertainty is greater than we expected.

To address the above challenges, we propose a general robust safe control framework. First, we propose a sound but not complete method *UR-SIS* (uncertainty-robust safety index synthesis) to synthesize a control-limit aware and uncertainty-robust safe controller that is guaranteed to be persistently feasible. In case such a safety index does not exist or we cannot find one, UR-SIS tries to maximize feasible state rate. Second, we formulate robust safe control with model uncertainty as *convex semi-infinite programming* (CSIP), which is the **exact** formulation (no over-approximation) for convex bounded uncertainties and the **tightest** convex formulation for arbitrary bounded uncertainties. We propose a real-time optimal solver based on the cutting-plane method. We also show that, SOCP formulation is a special case of our CSIP formulation for hyper-ellipsoid bounded uncertainties. Compared to existing methods, our method considers exact or tight uncertainty bounds, therefore leads to larger admissible control sets and consequently less-conservative robust safe controls. Lastly, we provide a theoretical analysis of what happens when unmodeled uncertainty exists. In particular, we present the condition on the unmodeled uncertainty that the system state can be kept in a larger set (than the safe set) with the learned UR-SI. Combining all the parts, we are able to ensure robustness and non-conservativeness of the safe control, and forward invariance of the safe set for bounded dynamic model uncertainties and even unmodeled uncertainties.

The remainder of the paper is organized as follows: Section II formulates the problem and introduces notations. Section III presents the proposed robust safe control framework in detail. Section IV evaluates the proposed methods on two examples: planar robot and Segway.

[1]These authors are with Robotics Institute, Carnegie Mellon University, `twei2, weiyezha, cliu6@andrew.cmu.edu`

[2]Shucheng Kang is with the Department of Electrical Engineering, Tsinghua University. Work done during an internship at Carnegie Mellon. `ksc19@mails.tsinghua.edu.cn`

II. FORMULATION

We consider the following nonlinear control-affine system with uncertain dynamic models (UDM) [1]:

$$\dot{x} = f(x) + g(x)u, \ f(x) \in \Sigma_f(x), \ g(x) \in \Sigma_g(x), \quad (1)$$

where state $x \in X \subseteq \mathbb{R}^n$ and control $u \in U \subseteq \mathbb{R}^m$. The terms $f(x) \in \mathbb{R}^n$ and $g(x) \in \mathbb{R}^{n \times m}$ are random matrices (elements $f(x)_i, g(x)_{i,j}$ are random variables) which are bounded by $\Sigma_f(x) \subseteq \mathbb{R}^n$ and $\Sigma_g(x) \subseteq \mathbb{R}^{n \times m}$ respectively. $\Sigma_f(x)$ and $\Sigma_g(x)$ are measurable functions on the state space which are state-dependent. For simplicity, in the following discussion, we may write $\forall f(x), g(x)$ to represent the condition $\forall f(x) \in \Sigma_f(x), \ g(x) \in \Sigma_g(x)$.

We consider the safety specification as a requirement that the system state should be constrained in a closed and connected set $\mathcal{X}_S \subseteq X$, which we call the safe set. We assume $\mathcal{X}_S$ is the zero-sublevel set of a *safety index* $\phi_0 : X \mapsto \mathbb{R}$ given by the user. That is $\mathcal{X}_S = \{x \mid \phi_0(x) \leq 0, x \in X\}$. Constraining states inside $\mathcal{X}_S$ can be expressed as a *forward invariance* problem: when $\phi_0(x(t_0)) \leq 0$, ensure $\phi_0(x(t)) \leq 0, \ \forall t > t_0$. Forward invariance can be guaranteed with minimal invasion by QP-based safe set algorithms.

Algorithm 1 (QP-based safe set algorithms)**.** Given a reference control u_{ref} and a safety index ϕ, QP-based safe set algorithms find safe control u by:

$$\min_u \|u - u_{\text{ref}}\|^2 \ s.t. \ \dot{\phi}(x,u) \leq -\gamma(\phi(x)), \quad (2)$$

where γ is a piecewise smooth function and $\gamma(\phi(x)) > 0$ when $\phi(x) > 0$. γ can be non-continuous and designed differently. A special case of γ is an extended class $\mathcal{K}$ function on $\phi(x)$, corresponding to the control barrier function (CBF) method [2]. Safe set algorithms require ϕ to be *persistent feasible*: $\forall x, \exists u$, such that $\dot{\phi}(x,u) \leq -\gamma(\phi(x))$. However, a user-defined safety index ϕ_0 may not be naturally persistently feasible. It is often necessary to design a ϕ based on ϕ_0 such that we can constrain the state in $S : \{\phi \leq 0\} \subseteq \mathcal{X}_S$ [1].

Extending algorithm 1 to UDM requires the safety constraint holds for all possible models. We define the extended problem as *uncertainty-robust QP* (UR-QP).

Definition 1 (UR-QP)**.**

$$\min_u \|u - u_{\text{ref}}\|^2 \ s.t. \ \dot{\phi}(x,u) \leq -\gamma(\phi(x)), \ \forall f(x), g(x).$$
$$(3)$$

And we say ϕ is an *uncertainty-robust safety index* (UR-SI) if it ensures persistent feasibility for all possible models.

Definition 2 (UR-SI)**.** A safety index ϕ is a UR-SI if there exists a piecewise smooth, strictly increasing function γ, and $\gamma(0) = 0$, such that the following *feasibility* condition holds:

$$\forall x, \exists u, \ \dot{\phi}(x,u) \leq -\gamma(\phi(x)), \ \forall f(x), g(x) \quad (4)$$

[1]Although our method assumes control affine dynamics, it is applicable to non-control affine systems, since we can always have a control affine form through dynamics extension [12].

To solve the UR-QP eq. (3) or to verify eq. (4), we are interested in the robust safe control set $U_r(x)$:

$$U_r(x) := \{u \mid \dot{\phi}(x,u) \leq -\gamma(\phi(x)), \ \forall f(x), g(x)\}. \quad (5)$$

With $U_r(x)$, eq. (3) can be converted into the following form

$$\min_{u \in U \cap U_r(x)} \|u - u_{\text{ref}}\|^2. \quad (6)$$

Prior work [13] has shown that $U_r(x)$ can be defined with the Lie derivatives. For simplicity, we may omit (x) when there is no ambiguity.

Definition 3. We denote Lie derivatives by $L_f\phi := \nabla\phi^T f$, $L_g\phi := \nabla\phi^T g$. And the range of $L_f\phi$ and $L_g\phi$ are denoted by $V_f := \{L_f\phi \mid f \in \Sigma_f\}$ and $V_g := \{L_g\phi \mid g \in \Sigma_g\}$.

Then the safety constraint can be transformed as follows:

$$\forall f,g, \dot{\phi}(x,u) \leq -\gamma(\phi) \iff \max_{f,g} \dot{\phi}(x,u) \leq -\gamma(\phi) \quad (7)$$

$$\iff \max_{L_f\phi, L_g\phi} L_f\phi + L_g\phi \cdot u \leq -\gamma(\phi) \quad (8)$$

$$\iff \max_{L_g\phi \in V_g} L_g\phi \, u \leq -\gamma(\phi) - \max_{L_f\phi \in V_f} L_f\phi =: c \quad (9)$$

Then $U_r(x)$ can be defined equivalently by

$$U_r = \{u \mid \max_{v \in V_g} v^T u \leq c\} = \{u \mid v^T u \leq c, \forall v \in V_g\}. \quad (10)$$

Although U_r is defined by linear constraints, the lack of a closed form of U_r makes it difficult to solve eq. (6). Even deciding if U_r is empty is difficult. Prior work [13] considers a simplified scenario and proposes a minimax formulation. But the minimax is generally intractable. In this work, we study how to solve this problem precisely and efficiently.

Besides, it is challenging to synthesize a UR-SI. Because a UR-SI requires eq. (4) to hold for all x, which involves infinitely many states. And, how to deal with unmodeled uncertainty remains unknown.

III. METHOD

To address these challenges, we first introduce a method to synthesize a UR-SI with finite states. Then we present an efficient and tight solver of the UR-QP by CSIP, which give optimal solutions of eq. (6) for convex bounded uncertainties. We also show that when the bound is an ellipsoid, CSIP reduces to SOCP. In the end, we analyze how the forward invariance set changes in presence of unmodeled large uncertainty.

A. Uncertainty-robust safety index synthesis (UR-SIS)

The purpose of safety index synthesis is to design a UR-SI ϕ based on ϕ_0 such that the closed-loop trajectory is constrained in the safe set $\mathcal{X}_S$ even under uncertainty.

To synthesize a UR-SI, we extend our previous work on synthesizing safety index for deterministic learned dynamics [14]. We proved that the feasibility of the whole state space can be ensured by only verifying finite sampled states. In this work, we generalize the method to uncertain dynamics and adjust the sampling density based on the uncertainty. Then we will be able to use an evolutionary algorithm to

find a valid UR-SI with the sampled states. This method works for low-dimensional parameterizations of CBF, which we call the safety index [1].

We first prove that ϕ is a UR-SI if it has feasible solutions on a sampled state set B based on the following assumption:

Assumption 1. γ, ϕ and $\nabla\phi$ are Lipschitz continuous functions with Lipschitz constants k_γ, k_ϕ and $k_{\nabla\phi}$ respectively. The set change $\Delta(\Sigma_f, x, x')$ and $\Delta(\Sigma_g, x, x')$ are bounded by $k_{\Sigma_f}\|x-x'\|$ and $k_{\Sigma_g}\|x-x'\|$ respectively, where $\Delta(A, x, x') := \max_{a\in A(x)}\min_{a'\in A(x')}\|a - a'\|$. And $\|u\|$ and $\|\dot{x}\|$ are bounded by M_u and $M_{\dot{x}}$ respectively.

Lemma 1. Suppose 1) we sample a state subset $B \subset X$ such that $\forall x \in X$, $\min_{x'\in B}\|x - x'\| \leq \delta$, where δ is a constant representing the sampling density; 2) $\forall x' \in B$, there exists a safe control u, s.t. $\dot{\phi}(x', u) \leq -\gamma(\phi(x')) - \epsilon$, $\forall f(x'), g(x')$, where $\epsilon = k_\phi(k_{\Sigma_f} + k_{\Sigma_g}M_u)\delta + k_{\nabla\phi}\delta M_{\dot{x}} + k_\gamma k_\phi\delta$. Then ϕ satisfies eq. (4) and therefore is a UR-SI.

Proof. According to condition 1), $\forall x \in X$, $\exists x' \in B$ such that $\|x - x'\| \leq \delta$. And based on assumption 1, we have $\|f(x) - f(x')\| \leq k_{\Sigma_f}\delta$, and $\|g(x) - g(x')\| \leq k_{\Sigma_g}\delta$. According to condition 2), for this x', we can find u s.t. $\dot{\phi}(x', u) \leq -\gamma(\phi(x')) - \epsilon$. Next we show x and u satisfy eq. (4) by <u>triangle inequality</u> and <u>Lipschitz condition</u>:

$$\dot{\phi}(x, u) = \dot{\phi}(x, u) - \dot{\phi}(x', u) + \dot{\phi}(x', u) \tag{11}$$

$$=\nabla\phi(x)\dot{x} - \nabla\phi(x')\dot{x}' + \dot{\phi}(x', u) \tag{12}$$

$$=\nabla\phi(x)\dot{x} - \nabla\phi(x')\dot{x}' - \nabla\phi(x)\dot{x}' + \nabla\phi(x)\dot{x}' + \dot{\phi}(x', u) \tag{13}$$

$$=\nabla\phi(x)(\dot{x} - \dot{x}') + [\nabla\phi(x) - \nabla\phi(x')]\dot{x}'$$
$$- \gamma(\phi(x)) + \gamma(\phi(x)) + \dot{\phi}(x', u) \tag{14}$$

$$\leq\|\nabla\phi(x)\|\|f(x) + g(x)u - f(x') - g(x')u\|$$
$$+ \|\nabla\phi(x) - \nabla\phi(x')\|\|\dot{x}'\|$$
$$- \gamma(\phi(x)) + \gamma(\phi(x)) - \gamma(\phi(x')) - \epsilon \tag{15}$$

$$\leq k_\phi(k_{\Sigma_f}\delta + k_{\Sigma_g}\delta M_u) + k_{\nabla\phi}\delta M_{\dot{x}}$$
$$- \gamma(\phi(x)) + k_\gamma k_\phi\delta - \epsilon \tag{16}$$

$$\leq -\gamma(\phi(x)) \tag{17}$$

Therefore, ϕ ensures feasibility for all states. □

With this lemma, we can validate if a given ϕ is a UR-SI with a finite sampled state set B. Next, we show how to find a UR-SI ϕ with B. Previous work [1] has proposed a general parameterized form ϕ_θ that guarantees $S := \{\phi_\theta \leq 0\} \subseteq \mathcal{X}_S$ without considering the feasibility of ϕ_θ. The parameterization also applies to UDM because S only depends on the state, which is certain. Therefore, all we need to do is to optimize θ to make ϕ_θ a UR-SI. The synthesis problem can be formulated as

$$\max_\theta \left|\{x \mid \min_u \dot{\phi}_\theta(x, u) < -\gamma(\phi_\theta(x))\}\right|. \tag{18}$$

The problem is non-differentiable. Therefore we apply a derivative-free evolutionary algorithm CMA-ES [14], which

iteratively optimizes θ by evaluating current θ candidates on B and proposing new θ candidates from the best performers. The objective is to maximize the feasible rate $r := \frac{|B^*|}{|B|}$, where $B^* := \{x' \mid \min_u \dot{\phi}(x', u) \leq -\gamma(\phi(x')) - \epsilon, \forall f(x'), g(x'). x' \in B\}$ is the feasible sampled state set. To decide if a state x' belongs to B^*, we rely on the robust safe control solvers to be proposed next. The algorithm stops when the feasible rate r converges. If $r = 1$, then we find a UR-SI ϕ. This method is sound but not complete. That is, we cannot assert a UR-SI does not exist if we cannot find one by this method. We will investigate complete methods in the future.

B. Robust safe control for convex bounded uncertainties

The UR-QP eq. (6) is difficult to solve because the lack of the closed form of $U_r(x)$ makes it difficult to characterize the boundary of $U_r(x)$. To address this issue, our key observation is that when $\Sigma_g(x)$ is a polytope, $U_r(x)$ can be well over-approximated by a polytope set $\hat{U}_r(x)$ which has a closed form of boundary. And the over-approximation can be iteratively reduced to zero. If $\Sigma_g(x)$ is not a polytope, we can construct its convex hull and then apply the method.

Next, we first introduce the polytope assumption and then give a formal description of the method. Note that the method do not rely on assumption 1.

Definition 4 ($g_{\text{flat}}(x)$). Represent $g(x)$ by $[\vec{g_1}, \vec{g_2}, \cdots \vec{g_n}] \in \mathbb{R}^{n\times m}$, where $\vec{g_i} \in \mathbb{R}^m$. Then we define $g_{\text{flat}}(x)$ as $[\vec{g_1}^T, \vec{g_2}^T, \cdots, \vec{g_n}^T]^T \in \mathbb{R}^{nm}$.

Assumption 2 (Polytope $\Sigma_f(x), \Sigma_g(x)$). We assume the uncertainties are bounded by polytopes $\Sigma_f(x)$ and $\Sigma_g(x)$. That is, $\exists A_f, b_f, A_g, b_g$ such that

$$\Sigma_f(x) = \{f(x) \mid A_f(x)f(x) < b_f(x)\}, \tag{19}$$
$$\Sigma_g(x) = \{g(x) \mid A_g(x)g_{\text{flat}}(x) < b_g(x)\}. \tag{20}$$

Consequently, $V_f(x)$ and $V_g(x)$ are also polytopes because $\nabla\phi(x)$ is a linear transformation,

A polytope $\Sigma_f(x)$ eases the computation of $c(x)$ in the RHS of eq. (9). In this case, $c(x)$ can be solved by linear optimization because $V_f(x)$ is a linear convex set, as done in [15]. Next, we propose a necessary condition of the existence of $U_r(x)$ and an efficient method to solve eq. (6).

Lemma 2. If $U_r(x)$ is not an empty set, then when $c(x) < 0$, $0 \notin \text{int } V_g(x)$.

Proof. We prove this by contradiction. $0 \in \text{int } V_g(x) \implies \exists r > 0$, s.t. $v \in V_g(x)$ if $\|v\| < r$. Given a v that $\|v\| < r$, suppose $U_r(x) \neq \emptyset$, then $\exists u \in U_r(x)$ such that $v^T u < c < 0$. But $-v \in V_g(x)$ (because $\|-v\| < r$) and $-v^T u > 0$, therefore $u \notin U_r(x)$, Contradicts with the assumption. □

Algorithm 2 (Polytope RSSA). Our key insight is that the boundary of $U_r(x)$ is mostly defined by $\mathcal{H}$: the vertices of $V_g(x)$. We define $\hat{U}_r(x) := \{u \mid v^T u \leq c, \forall v \in \mathcal{H} \subseteq V_g(x)\}$. It is easy to see that $U_r(x) \subseteq \hat{U}_r(x)$ as shown in fig. 1. To solve the UR-QP eq. (6), we first minimize
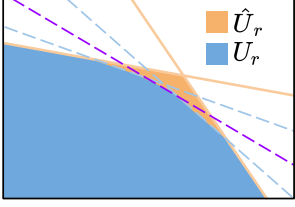
Fig. 1. $\hat{U}_r$ is defined by infinite constraints but can be over-approximated by the orange ones, which correspond to vertices of V_g. The purple constraint represents the one with the maximum violation and will be added to $\hat{U}_r$.

$\|u - u_{\text{ref}}\|$ in $U(x) \cap \hat{U}_r(x)$, which is a QP with finite linear constraints. Then, we find the constraint of max violation: $v_* = \arg\max_{v \in V_g(x)} v^T u$. If $v_*^T u > c$, we add $v_*^T u \leq c$ to $\hat{U}_r(x)$ and repeat the process.

Lemma 3. Algorithm 2 converges to the optimal solution.

Proof. Equation (6) is a special case of convex semi-infinite programming (CSIP), and Polytope RSSA is a variant of the cutting-plane method [16]. It has been proved that the method always converges to an optimal solution in [16]. □

A special case of Polytope RSSA is when $\Sigma_f(x)$ and $\Sigma_g(x)$ are hyper-ellipsoids. In this case, CSIP can be reduced to Second Order Cone Programming (SOCP), which is particularly useful when we consider chance constraints for Gaussian uncertainties. SOCP can be solved efficiently by Semi-Definite Programming (SDP) solvers.

Assumption 3 (Hyper-ellipsoid $\Sigma_f(x)$, $\Sigma_g(x)$). $\exists \mu_f(x) \in \mathbb{R}^n$, $Q_f(x) \in \mathbb{R}^{n \times n}, Q_f \succ 0$, $d_f(x) \in \mathbb{R}^+$, and $\exists \mu_g(x) \in \mathbb{R}^{mn}$, $Q_g(x) \in \mathbb{R}^{mn \times mn}$, $Q_g \succ 0$, $d_g(x) \in \mathbb{R}^+$, *s.t.*

$$\Sigma_f(x) := \{f \mid (f - \mu_f)^T Q_f^{-1}(f - \mu_f) \leqslant d_f\}, \quad (21)$$

$$\Sigma_g(x) := \{g_{\text{flat}} \mid (g_{\text{flat}} - \mu_g)^T Q_g^{-1}(g_{\text{flat}} - \mu_g) \leqslant d_g\}. \quad (22)$$

Remark. If $f \sim \mathcal{N}(\mu_f, Q_f)$, $g_{\text{flat}} \sim \mathcal{N}(\mu_g, Q_g)$, $d_f = \chi_n^2(\alpha)$ and $d_g = \chi_{mn}^2(\alpha)$, where $\alpha \in \mathbb{R}^+$ and χ^2 is the chi-square function. Then the above hyper-ellipsoids correspond to the $1 - \alpha$ confidence bounds.

Corollary. $L_g\phi$, $L_f\phi$ are also bounded by hyper-ellipsoids. Let $\mu_v(x) = \nabla\phi \cdot \mu_g$, $Q_v(x) = \nabla\phi^T Q_g \nabla\phi$. Then

$$V_g(x) := \{v \mid (v - \mu_v)^T Q_v^{-1}(v - \mu_v) \leqslant d_g\}. \quad (23)$$

Similarly, we can define $V_f(x)$ and compute c as in eq. (9).

The lemma below gives us an equivalent form of $U_r(x) := \{u \mid v^T u \leqslant c, \forall v \in V_g(x)\}$ using the hyper-ellipsoid bounds, which is defined by only one constraint.

Algorithm 3 (Ellipsoid RSSA). Under assumption 3, $\forall x, \exists L$, *s.t.* $LL^T = d_g Q_v$. Therefore $U_r(x)$ has the second-order cone form and makes the UR-QP eq. (6) a SOCP:

$$U_r(x) = \{u \mid ||L^T u|| \leqslant -\mu_v^T u + c\} \quad (24)$$

Proof. For simplicity, let $Q(x) := d_g(x)Q_v(x)$. Then, the hyper-ellipsoid can be described as

$$\Sigma_v(x) = \{v \mid (v - \mu_v)^T Q^{-1}(v - \mu_v) \leqslant 1\} \quad (25)$$

Because $Q \succ 0$, there exists an invertible matrix L, s.t. $Q = LL^T$. Define $v' = L^{-1}(v - \mu_v)$, then V_g could be transformed to V_g': $V_g' = \{v' \mid v'^T v' \leqslant 1\}$. Substitute v with v' in U_r, we could get

$$U_r(x) = \{u \mid (Lv' + \mu_v)^T u \leqslant c, \forall v' \in V_g'\}$$

To further simplify U_r's form, we first introduce an auxiliary variable u' s.t. $u = (L^T)^{-1}u'$. Notice that $\max_{v'^T v' \leqslant 1} v'^T u' = ||u'||$. Therefore,

$$U_r(x) = \{(L^T)^{-1}u' \mid ||u'|| + (L^{-1}\mu_v)^T u' \leqslant c\} \quad (26)$$

$$= \{u \mid ||L^T u|| \leqslant -\mu_v^T u + c\}, \quad (27)$$

which is a second-order cone form. □

C. Forward invariance under unmodeled uncertainty

We have discussed how to design a robust safety index and how to find robust safe control with known uncertainty ranges. However, in practice, sometimes we cannot acquire an accurate uncertainty range in advance. Further analysis for unmodeled uncertainties is desired. We present the condition of the unmodeled uncertainties such that the forward invariance is still preserved (in a larger set) by a UR-SI.

Definition 5 (Residual dynamics). We define the unmodeled uncertainty as unknown residual dynamics $\tilde{f}(x)$ and $\tilde{g}(x)$:

$$\dot{x} = f(x) + g(x)u + \tilde{f}(x) + \tilde{g}(x)u, \quad (28)$$
$$f(x) \in \Sigma_f(x), g(x) \in \Sigma_g(x)$$

Lemma 4. A UR-SI ϕ ensures the forward invariance of the set $\hat{S} = \{x \mid \phi(x) \leqslant \gamma^{-1}(k_\phi M_{\tilde{x}})\}$ with unmodeled uncertainties shown in eq. (28), where $M_{\tilde{x}} := \max_{x,u} ||\tilde{f}(x) + \tilde{g}(x)u||$ represents the maximum norm of the residual dynamics:

Proof. It suffices to show that applying ϕ on the nominal dynamics eq. (1) ensures $\dot{\phi} \leq 0$ on $\partial\hat{S} := \{x \mid \phi(x) = \gamma^{-1}(k_\phi M_{\tilde{x}})\}$ for the true dynamics eq. (28).

$\dot{\phi}$ can be decomposed as $\dot{\phi} = \dot{\phi}_{\text{nom}} + \dot{\phi}_{\text{res}}$, where $\dot{\phi}_{\text{nom}} := \nabla\phi[f(x) + g(x)u]$, $\dot{\phi}_{\text{res}} := \nabla\phi(x)[\tilde{f}(x) + \tilde{g}(x)u]$. Because $\forall x, \exists u, \dot{\phi}_{\text{nom}}(x) \leq -\gamma(\phi(x)), \forall f(x), \forall g(x)$ and

$$\dot{\phi}_{\text{res}}(x) = \nabla\phi(x)[\tilde{f}(x) + \tilde{g}(x)u] \quad (29)$$

$$\leq k_\phi ||\tilde{f}(x) + \tilde{g}(x)u|| \leq k_\phi M_{\tilde{x}}. \quad (30)$$

We have $\forall x \in \partial\hat{S}, \exists u$ such that $\forall f(x), \forall g(x)$

$$\dot{\phi}(x) = \dot{\phi}_{\text{nom}}(x) + \dot{\phi}_{\text{res}}(x) \leq -\gamma(\phi(x)) + k_\phi M_{\tilde{x}} \quad (31)$$

$$= -\gamma(\gamma^{-1}(k_\phi M_{\tilde{x}})) + k_\phi M_{\tilde{x}} = 0 \quad (32)$$

Therefore, $\hat{S}$ is forward invariant with the UR-SI ϕ. □

IV. EXPERIMENT

Our method applies to arbitrary dynamic model uncertainties as long as they can be bounded by state-dependent polytopes or ellipsoids, including but not limited to Gaussian Process, neural network dynamic models, and parameterized models. For simplicity, we show the effectiveness of our method on two parameterized models with non-Gaussian and Gaussian uncertainties.
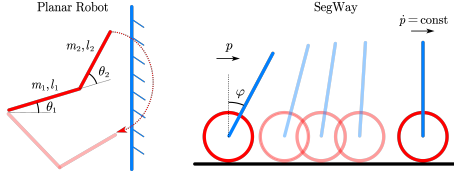
Fig. 2. Illustration of the planar robot and Segway
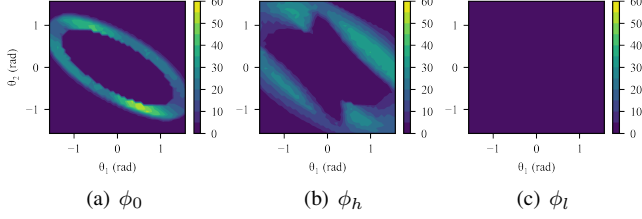

(a) ϕ_0 (b) ϕ_h (c) ϕ_l

Fig. 3. Distribution of infeasible states in the robot configuration space of three safety indexes. Each point of the graph represents a joint position. We sample 100 joint velocities at each point. The color denotes how many of them have no feasible robust safe control. (a) shows that the original safety index ϕ_0 cannot ensure feasibility. (b) shows that a hand-designed safety index ϕ_h also has many infeasible states. (c) shows that our learned UR-SI ϕ_l ensures feasibility for all states even considering model uncertainty.

A. Planar Robot

We first test our method on a planar robot platform. We define the safety specification as not colliding with a vertical wall. Define the position and velocity of planar robot's end-effector as $p_{ee} = [x_{ee}, y_{ee}]^T$ and $\dot{p}_{ee} = [\dot{x}_{ee}, \dot{y}_{ee}]^T$. Then the safety constraint $\phi_0 = -\Delta x = x_{ee} - x_{wall} < 0$ should always be satisfied. The detailed dynamic model can be found online [2]. We assume the payload m_2 (the mass of the second arm) is uncertain. Specifically, we assume $m_2 \sim \mathcal{N}(1.0, 0.3^2) \cdot I_{[0.1, 1.9]}$. But after transformation, the dynamic model uncertainty is not Gaussian (the transformation involves multiplicative inverse and matrix inverse).

1) Feasibility: We first show that the safety index learning ensures feasibility. To learn a UR-SI, we use the parameterized form proposed by [1]: $\phi = \max\{\phi_0, -x_{wall}^\alpha + x_{ee}^\alpha + k_v \dot{x}_{ee} + \beta\}$, where α, k_v, β are learnable parameters. The search ranges for the parameters are: $\alpha \in (0.1, 5.0), k_v \in (0.1, 5.0), \beta \in (0.001, 1.0)$. We set $x_{wall} = 1.5$, and uniformly sample 250000 states in the whole state space. Figure 3 compares ϕ_0, a manually tuned safety index ϕ_h ($\alpha = 1.0, k_v = 0.2, \beta = 0.0$), and a synthesized UR-SI ϕ_l with the Polytope RSSA solver. ($\alpha = 0.57, k_v = 2.15, \beta = 0.072$). UR-SI achieves a 0 infeasible rate.

2) Robust safe control solver: We compare the performance of our solvers Polytope RSSA and Ellipsoid RSSA, with a popular method: constant bounded robust safe control (Constant RSSA) [9], [10], [11], which considers the uncertainty as residual dynamics, and uses a constant to bound the residual time derivative of ϕ: $\dot{\phi}_{res}$ caused by uncertainties.

We construct the uncertainty bounds for both Ellipsoid RSSA and Polytope RSSA by sampling methods. We first get 50 dynamic model samples by sampling m_2 from the
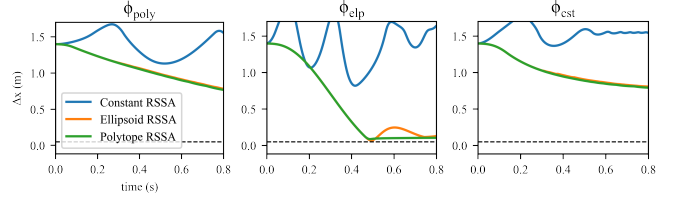
(a) Case study 1. $\phi_0 < 0$, $\phi < 0$
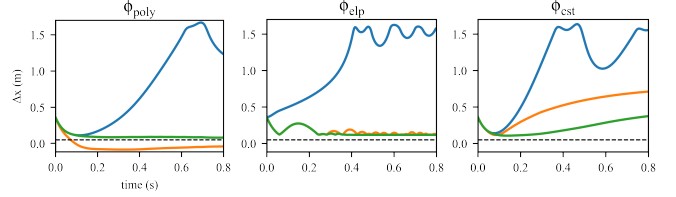
(b) Case study 2. $\phi_0 < 0$, $\phi > 0$

Fig. 4. Comparison of RSSA solvers with three UR-SI and under two conditions. From left to right in each row, we learn a UR-SI based on different RSSA strategies: Polytope RSSA, Ellipsoid RSSA with 95% confidence level, and Constant RSSA. In all cases, Polytope RSSA maintains safety and is less conservative than Constant RSSA. In the bottom-left graph, Ellipsoid RSSA fails to keep safe because the Gaussian assumption does not capture the uncertainty correctly. And if we use a large confidence level in Ellipsoid RSSA, e.g., 99%, CMA-ES cannot find a UR-SI. Feasibility plot can be found online [2] in the appendix.

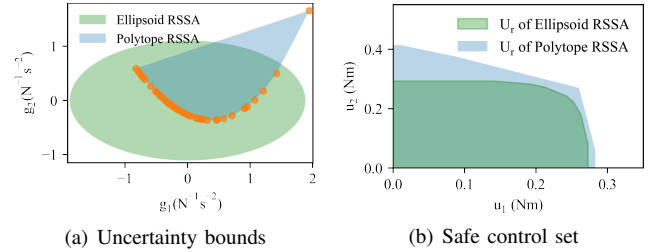(a) Uncertainty bounds   (b) Safe control set

Fig. 5. (a) shows polytope (CSIP) and ellipsoid (SOCP) uncertainty bounds for a 2D slice of $g(x)$. Orange points represent dynamic model samples. The ellipsoid corresponds to the 3σ confidence region (99.7%) of the fitted Gaussian distribution. It still does not cover the true dynamic model distribution, while the polytope bound does. (b) shows the corresponding robust safe control sets U_r. Polytope RSSA has a larger U_r because it is less conservative.

distribution and execute the nonlinear dynamics transformation. Then we either fit a Gaussian distribution for Ellipsoid RSSA or construct a polytope for Polytope RSSA.

For a fair comparison, we learned a UR-SI for each solver, denoted by ϕ_{poly}, ϕ_{elp} and ϕ_{cst}. With a true yet unknown $m_2 = 0.5$, we plot trajectories of Δx of all the solvers with the three UR-SIs under two different initial conditions: 1. when $\phi_0 < 0$ and $\phi < 0$, fig. 4 (a) shows that all the methods ensures safety, but as for the conservativeness, Polytope RSSA < Ellipsoid RSSA < Constant RSSA. 2. when $\phi_0 < 0$ but $\phi > 0$, fig. 4 (b) shows that Ellipsoid RSSA violates the safety constraint; To explain it in detail, we visualize the uncertainty bounds and corresponding safe control sets in fig. 5. It shows that even the 99.7% Gaussian confidence bound does not cover the uncertainty correctly, leading to potential safety violations. It demonstrates that existing methods assuming Gaussian uncertainties may be

unreliable when the uncertainty is not Gaussian. Besides, we can see that the SOCP formulation is not as tight as the CSIP formulation in general. Note that Ellipsoid RSSA also can guarantee safety if the bound correctly covers the uncertainty, and the bound does not have to be constructed from Gaussian confidence intervals.

3) Unmodeled uncertainties: We study the forward invariance set change under unmodeled uncertainties. We test several m_2 that are out of the modeled uncertainty bound $[0.1, 1.9]$ in safety index learning and record $\phi_{\max}$: the largest ϕ of many sampled trajectories, as an indicator of the forward invariant set boundary. Table I shows that the forward invariant set grows incrementally with the uncertainty level, and $\phi_{\max}$ is always below the theoretical bound predicted in lemma 4.

m_2	0.005	0.08	1.0	2.0	3.0
$\phi_{\max}$	32.09	0.41	-0.0013	0.92	1.51
Theoretical bound	256.4	3.51	0.0	1.07	8.75

TABLE I
FORWARD INVARIANCE WITH UNMODELED UNCERTAINTIES.

B. Segway

To highlight the optimality and efficiency of Ellipsoid RSSA in case of Gaussian uncertainty. We also test our method on a realistic Segway model. We consider a tracking task with a safety specification on the tilt angle: $\phi_0 = |\varphi| - 0.1$. We consider a safety index $\phi = \max\{\phi_0, -0.1^\alpha + |\varphi|^\alpha + k_v \text{sign}(\varphi)\dot{\varphi} + \beta\}$, where α, k_v, β are learnable parameters. The nominal controller is designed to maintain $\dot{p}$ at $1m/s$. The detailed dynamic model can be found online [3] We assume the motor torque constant K_m is a truncated Gaussian distribution: $K_m \sim \mathcal{N}(2.524, 0.3^2) \cdot I_{[1.624,3.424]}$. After transformation, the dynamic model uncertainty is a state-dependent Gaussian distribution.

1) Robust safe control solvers: We can efficiently compute the correct ellipsoid bounds for Ellipsoid RSSA because the dynamic model follows Gaussian distributions. But for Polytope RSSA, we still rely on the sampling method to construct the bound. As shown in fig. 6 (a), they both ensure safety and are non-conservative. But the computation time of Ellipsoid RSSA does not grow with the number of samples as shown in fig. 6 (b), which is a significant advantage when the dimensions are high and require more samples. The constant RSSA is overly conservative because $\dot{\phi}_{res}$ is too large.

V. DISCUSSION

In this work, we proposed a general framework of robust safe control for uncertain dynamic models. The future work includes studying complete methods for safety index synthesis, generalizing the method to the case that f and g have correlations, and generalizing the method to high-dimensional applications that require a large number of samples to learn the safety index and construct bounds.
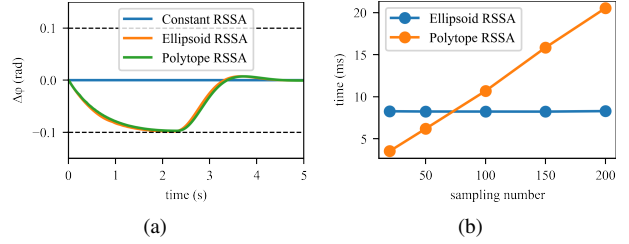

Fig. 6. (a) Comparing three different RSSA strategies in Segway. Both Polytope RSSA and Ellipsoid RSSA maintain safety. However, Constant RSSA always forces u to zero (not moving) because of conservativeness. (b) Comparison of the average computation time of Polytope RSSA and Ellipsoid RSSA.

REFERENCES

[1] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2014.

[2] T. Wei and C. Liu, "Safe control algorithms using energy functions: A uni ed framework, benchmark, and new directions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 238–243.

[3] R. Wajid, A. U. Awan, and M. Zamani, "Formal Synthesis of Safety Controllers for Unknown Stochastic Control Systems using Gaussian Process Learning," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*. PMLR, May 2022, pp. 624–636.

[4] A. J. Taylor, V. D. Dorobantu, S. Dean, B. Recht, Y. Yue, and A. D. Ames, "Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6469–6476.

[5] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6762–6769.

[6] J. Buch, S.-C. Liao, and P. Seiler, "Robust control barrier functions with sector-bounded uncertainties," *IEEE Control Systems Letters*, vol. 6, pp. 1994–1999, 2021.

[7] K. Garg and D. Panagou, "Robust control barrier and control lyapunov functions with fixed-time convergence guarantees," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2292–2297.

[8] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, Oct. 2018.

[9] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 6286–6291.

[10] Q. Nguyen and K. Sreenath, "Robust Safety-Critical Control for Dynamic Robotics," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1073–1088, Mar. 2022.

[11] L. Brunke, S. Zhou, and A. P. Schoellig, "Barrier Bayesian Linear Regression: Online Learning of Control Barrier Conditions for Safety-Critical Control of Uncertain Systems," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*. PMLR, May 2022, pp. 881–892.

[12] C. Liu and M. Tomizuka, "Algorithmic safety measures for intelligent industrial co-robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3095–3102.

[13] C. Noren, W. Zhao, and C. Liu, "Safe Adaptation with Multiplicative Uncertainties Using Robust Safe Set Algorithm," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 360–365, Jan. 2021.

[14] T. Wei and C. Liu, "Safe control with neural network dynamic models," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 739–750.

[15] C. Liu and M. Tomizuka, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 465–470.

[16] S. Gustafson and K. Kortanek, "Numerical treatment of a class of semi-infinite programming problems," *Naval Research Logistics Quarterly*, vol. 20, no. 3, pp. 477–504, 1973.

[3] https://arxiv.org/abs/2209.06896

APPENDIX

A. SCARA

Define joint positions of the robot arm as $\theta = [\theta_1, \theta_2]^T$ and joint velocities as $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2]^T$. Here we assume all states $x = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T$ and control inputs u are bounded: $X : [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-2, 2] \times [-2, 2]$ and $U : [-20, 20] \times [-20, 20]$.

The control-affine dynamic model for SCARA is as follows:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ -M(\theta)^{-1} H(\theta, \dot{\theta}) \end{bmatrix} + \begin{bmatrix} 0 \\ M(\theta)^{-1} \end{bmatrix} u \tag{33}$$

where $M(\theta)$ is the mass matrix and $H(\theta, \dot{\theta})$ is the Coriolis matrix:

$$M(\theta_1, \theta_2) = \begin{bmatrix} 2A + 2B + 2C\cos(\theta_2) & 2B + C\cos(\theta_2) \\ 2B + C\cos(\theta_2) & 2B \end{bmatrix} \tag{34}$$

$$H(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \begin{bmatrix} -C\sin\theta_2 \cdot (2\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2 \\ C\sin\theta_2 \cdot \dot{\theta}_1^2 \end{bmatrix} \tag{35}$$

And A, B, C only depend on the robot arm's masses and lengths:

$$\begin{cases} A = \frac{1}{6}m_1 l_1^2 + \frac{1}{2}m_2 l_1^2 \\ B = \frac{1}{6}m_2 l_2^2 \\ C = \frac{1}{2}m_2 l_1 l_2 \end{cases} \tag{36}$$

B. Segway

Given wheel's position p and frame's tilt angle φ, we define $q = [p, \varphi]^T$ and $\dot{q} = [\dot{p}, \dot{\varphi}]^T$. SegWay's dynamic model can be written as

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -M(q)^{-1} H(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ M(q)^{-1} B \end{bmatrix} u \tag{37}$$

where $M(q)$, $H(q, \dot{q})$, B and b_t are defined as follows:

$$M(q) = \begin{bmatrix} m_0 & mL\cos(\varphi) \\ mL\cos(\varphi) & J_0 \end{bmatrix} \tag{38}$$
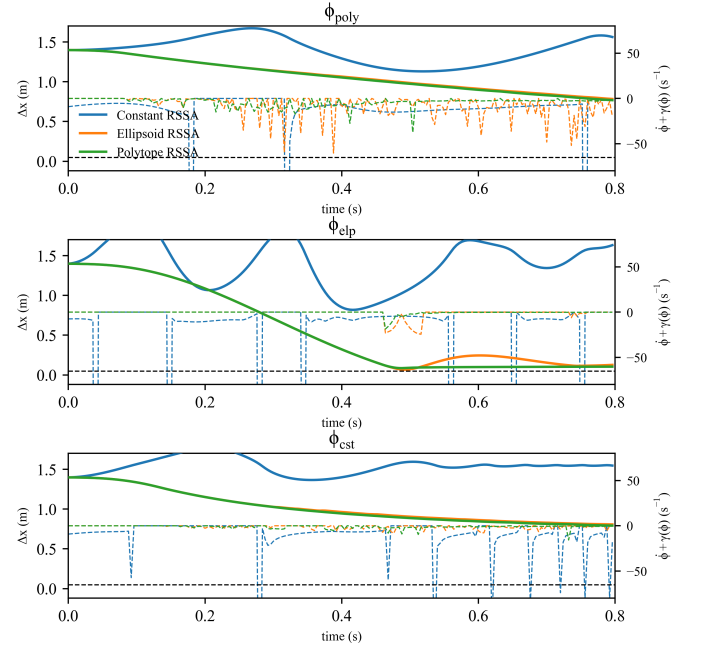
$$H(q, \dot{q}) = \begin{bmatrix} -mL\sin(\varphi)\dot{\varphi}^2 + \frac{b_t}{R}(\dot{p} - R\dot{\varphi}) \\ -mgL\sin(\varphi) - b_t(\dot{p} - R\dot{\varphi}) \end{bmatrix} \tag{39}$$

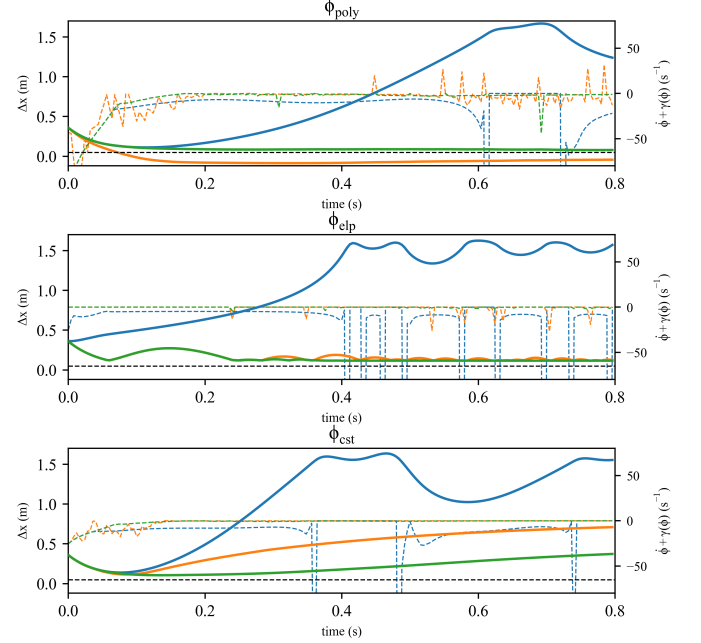$$B = \begin{bmatrix} \frac{K_m}{R} \\ -K_m \end{bmatrix} \tag{40}$$

$$b_t = K_m \frac{K_b}{R} \tag{41}$$

C. Feasibility Plot

Figure 7 shows the trajectory of ϕ_0 and $\dot{\phi} + \gamma(\phi)$. Solid lines represent the value of $-\phi_0$, and the dashed lines represent $\dot{\phi} + \gamma(\phi)$. We can see from the figure that the Polytope RSSA safety index is persistently feasible ($\dot{\phi} + \gamma(\phi)$ always < 0) while the Ellipsoid RSSA one is not.

(a) Case study 1. $\phi_0 < 0$, $\phi < 0$

(b) Case study 2. $\phi_0 < 0$, $\phi > 0$

Fig. 7. $\Delta x \sim \dot{\phi} + \gamma(\phi)$