Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Continuous and discrete abstractions for planning, applied to ship docking ⋆

Pierre-Jean Meyer * **He Yin** ** **Astrid H. Brodtkorb** ***
Murat Arcak * **Asgeir J. Sørensen** ***

* *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA,* {pjmeyer, arcak}@berkeley.edu
** *Department of Mechanical Engineering, University of California, Berkeley, USA,* he_yin@berkeley.edu
*** *Centre for Autonomous Marine Operations (AMOS), Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Otto Nielsens vei 10, 7052 Trondheim, Norway,* {astrid.h.brodtkorb, asgeir.sorensen}@ntnu.no

Abstract: We propose a hierarchical control framework for the synthesis of correct-by-construction controllers for nonlinear control-affine systems with respect to reach-avoid-stay specifications. We first create a low-dimensional continuous abstraction of the system and use Sum-of-Squares (SOS) programming to obtain a low-level controller ensuring a bounded error between the two models. We then create a discrete abstraction of the continuous abstraction and use formal methods to synthesize a controller satisfying the specifications shrunk by the obtained error bound. Combining both controllers finally solves the main control problem on the initial system. This two-step framework allows the discrete abstraction methods to deal with higher-dimensional systems which may be computationally expensive without the prior continuous abstraction. The main novelty of the proposed SOS continuous abstraction is that it allows the error between abstract and concrete models to explicitly depend on the control input of the abstract model, which offers more freedom in the choice of the continuous abstraction model and provides lower error bounds than when only the states of both models are considered. This approach is illustrated on the docking problem of a marine vessel.

Keywords: Abstraction-based control, hierarchical control, model reduction, symbolic control, high level planning.

1. INTRODUCTION

Abstraction-based control synthesis aims to abstract a system into a simpler model, synthesize a controller on the abstraction and finally refine this controller to ensure the satisfaction of the same control objective on the initial system. Starting from a continuous initial system modeled as a differential equation, two abstraction-based control approaches can be considered. In the *hierarchical control* approach, we create a continuous abstraction with less variables or simpler dynamics than the initial model, and we create a low-level controller for the concrete model to track the abstract one (Girard and Pappas, 2009). Note that this is slightly different from *model reduction* in which the input and output variables of both models are kept identical (Antoulas et al., 2001). In the *symbolic control* approach, we create a discrete abstraction by partitioning the state space and using reachability analysis methods to over-approximate the continuous dynamics into a finite transition system (see e.g. Reissig et al., 2016). Due to the state space partitioning, discrete abstractions are limited in their scalability. One possible approach to reduce the complexity is to decompose the concrete system into smaller subsystems for which discrete abstractions are more easily created (see e.g. Pola et al., 2017). This method is applicable to weakly interconnected networked systems, but is not always practical for strongly interconnected systems with no clear structure to guide the decomposition.

In this paper, we address the scalability problem of discrete abstractions through an alternative approach, by considering a two-step process sketched in Figure 1 and described in more details in Section 2.2. In the first step, we design a continuous abstraction of the concrete model and use Sum-of-Squares (SOS) programming to find a low-level controller ensuring that the concrete model tracks trajectories of the continuous abstraction with an associated error bound. Therefore, for the concrete system to satisfy a *reach-avoid-stay* specification (reach a target set while avoiding unsafe sets, then stay there), it is sufficient to look for a controller of the continuous abstraction satisfying the same specification with sets shrunk by the error bound.

⋆ This work was supported by the Peder Sather Center for Advanced Study, a consortium of UC Berkeley and nine Norwegian academic institutions. It was also supported in part by the U.S. National Science Foundation grant ECCS-1906164, the U.S. Air Force Office of Scientific Research grant FA9550-18-1-0253, and Research Council of Norway through the Centres of Excellence funding scheme, project number 223254 AMOS, FRINATEK project 274441 UNLOCK, and MAROFF project 280655 ORCAS.

The second step aims to create a discrete abstraction of the lower-dimensional continuous abstraction and synthesize, using formal methods, a correct-by-construction controller to satisfy the shrunk specifications.

Although continuous and discrete abstractions are not novel ideas on their own, few results have attempted to combine them, and their applicability has been limited to restrictive classes of systems, such as a double integrator (Fainekos et al., 2009), piecewise affine systems (Mickelin et al., 2014), differentially flat systems (Colombo and Girard, 2013) or bipedal robots (Ames et al., 2015). In contrast, the SOS-based continuous abstraction proposed here is applicable to the large class of control-affine nonlinear systems approximated with polynomial dynamics. In addition to its broader applicability, the proposed method allows the error between abstract and concrete models to depend not only on the states of both models, but also on the control input of the abstract model. This input dependence is particularly important when abstracting a dynamical model into its kinematic version, since we want to minimize the error between the velocities which are states of the concrete model and inputs of the abstract one. More generally, this offers more freedom in the choice of the continuous abstraction model and provides lower error bounds than when only the abstract state is considered (Singh et al., 2018; Smith et al., 2019).

We apply this approach to a scenario where a marine vessel docks autonomously at a harbor. Today, this maneuver is done manually, due to high risk of collision and strict requirements for precision, even when system faults have occurred. Typically, path planning for autonomous ships will consist of an offline algorithm making the preliminary plan based on available information like time and fuel consumption constraints, weather, and pre-defined safety margins, and an online part doing contingency-handling (e.g. collision avoidance). In order for autonomous ships to be allowed to sail, the control system software must be verified so that it is at least as safe as human navigated ships (DNV GL, 2018). By using correct-by-construction methods for design of offline path planning algorithms, the burden on simulation-based testing of the autonomous control system implementation is greatly reduced.

This paper is organized as follows. Section 2 formulates the considered problem and provides an overview of the proposed two-step approach. Section 3 presents the first step and main theoretical contribution of this paper on continuous abstraction. Section 4 provides the discrete abstraction procedure of the second step, which is presented for self-containment of the overall approach. Finally, the proposed method is illustrated in Section 5 for the docking problem on the 6-dimensional model of a marine vessel. Due to space limitations, proofs of the theoretical results are only included in the extended version of this paper. [1]

2. PRELIMINARIES

Let $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_+$ denote the sets of non-negative integers, real numbers and non-negative real numbers, respectively. For $\xi \in \mathbb{R}^n$, $\mathbb{R}[\xi]$ represents the set of polynomials in ξ with real coefficients, and $\mathbb{R}^m[\xi]$ and $\mathbb{R}^{m\times p}[\xi]$ denote all vector and matrix valued polynomial functions. The subset $\Sigma[\xi] = \{p = p_1^2 + p_2^2 + ... + p_M^2 \mid p_1, ..., p_M \in \mathbb{R}[\xi]\}$ of $\mathbb{R}[\xi]$ is the set of SOS polynomials in ξ. A set $X \subseteq \mathbb{R}^n$ is an interval of the vector space $\mathbb{R}^n$ if there exists $\underline{x}, \overline{x} \in X$ such that for all $x \in X$ we have $\underline{x} \le x \le \overline{x}$ using componentwise inequalities. Given a positive vector $\varepsilon \in \mathbb{R}^n_+$ and a set $X \subseteq \mathbb{R}^n$, we introduce $X^{+\varepsilon} = \{x + e \in \mathbb{R}^n \mid x \in X,\ e \in [-\varepsilon, \varepsilon]\}$ and $X^{-\varepsilon} = \{x \in \mathbb{R}^n \mid x + [-\varepsilon, \varepsilon] \subseteq X\}$ as the set X expanded and shrunk by the interval $[-\varepsilon, \varepsilon]$, respectively.

2.1 Problem formulation

Consider a control-affine nonlinear system

$$\dot{x} = f(x, w) + g(x, w)u, \tag{1}$$

with state $x \in X \subseteq \mathbb{R}^{n_x}$, bounded control input $u \in U \subseteq \mathbb{R}^{n_u}$, bounded disturbance input $w \in W \subseteq \mathbb{R}^{n_w}$ and Lipschitz continuous functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x \times n_u}$. The sets X, U and W are assumed to be intervals of their respective spaces.

The control objectives are formulated as *reach-avoid-stay* games which combine several safety and reachability subgoals. In addition to the state constraints defined by the set X, we define two subsets $X_a, X_r \subseteq X$, where the safety specification aims to avoid the set X_a at all time, while the reach-stay objective is to reach the set X_r in finite time and then stay there forever.

Problem 1. Given system (1) and subsets $X_a, X_r \subseteq X$, find a set of initial states $X_0 \subseteq X$ and a control strategy $u : X \to U$ such that for any disturbance signal $w : \mathbb{R}_+ \to W$, all trajectories $x : \mathbb{R}_+ \to \mathbb{R}^{n_x}$ of the closed-loop system initialized in X_0 satisfies $x(t) \in X \backslash X_a$ for all $t \ge 0$ and there exists $t_r \ge 0$ such that $x(t) \in X_r$ for all $t \ge t_r$.

2.2 Overview of the proposed approach

In this paper, we solve Problem 1 in a two-step approach summarized below and in Figure 1, first by creating a continuous abstraction of the concrete model (1), and next by using formal methods to synthesize a correct-by-construction controller on a discrete abstraction of the lower-dimensional continuous abstraction.

Given the concrete model (1), the continuous abstraction

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \hat{w}), \tag{2}$$

and the state and input constraints on both the concrete model ($X \subseteq \mathbb{R}^{n_x}$, $U \subseteq \mathbb{R}^{n_u}$, $W \subseteq \mathbb{R}^{n_w}$) and the abstract model ($\hat{X} \subseteq \mathbb{R}^{\hat{n}_x}$, $\hat{U} \subseteq \mathbb{R}^{\hat{n}_u}$, $\hat{W} \subseteq \mathbb{R}^{\hat{n}_w}$), the first step is to design a low-level controller $\kappa : \mathbb{R} \times X \times \hat{X} \times \hat{U} \to U$ ensuring that the concrete model (1) can track trajectories of the abstract model (2) with an error upper bounded by the known vector $\varepsilon \in \mathbb{R}^{n_x}$. This is achieved by applying the Sum-of-Squares (SOS) methods detailed in Section 3.

To solve the reach-avoid-stay specifications (X, X_a, X_r) from Problem 1 on the concrete model (1), it is thus sufficient to solve an auxiliary problem on the abstract model (2) with respect to the reach-avoid-stay specification $(X^{-\varepsilon}, X_a^{+\varepsilon}, X_r^{-\varepsilon})$ with the shrunk state constraints $X^{-\varepsilon}$ and target set $X_r^{-\varepsilon}$ and the expanded set of states to be avoided $X_a^{+\varepsilon}$ (see Figure 3 in Section 5 for an illustration of these sets). The second step of the approach, detailed in Section 4, then consists in creating a discrete

[1] Available at: https://arxiv.org/abs/1911.09773

abstraction of the abstract model (2) to synthesize a symbolic controller $\mathcal{K} : \hat{X} \to \hat{U}$ solving this auxiliary problem.

Combining the symbolic controller $\mathcal{K} : \hat{X} \to \hat{U}$ with the low-level controller $\kappa : \mathbb{R} \times X \times \hat{X} \times \hat{U} \to U$ then results in a controller solving Problem 1 on the concrete system.

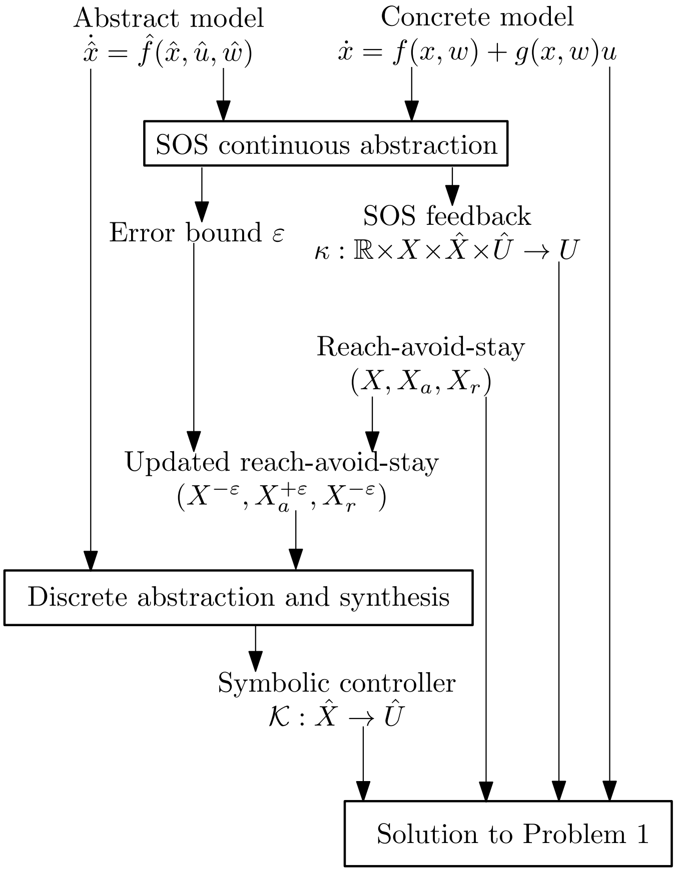

Fig. 1. Overview of the design steps to solve Problem 1.

3. CONTINUOUS ABSTRACTION

The first step of the proposed approach is to create a simplified version of the concrete model (1), referred to as *continuous abstraction* or *abstract model* and defined with hatted notations as in (2), with state $\hat{x} \in \hat{X} \subseteq \mathbb{R}^{\hat{n}_x}$, control input $\hat{u} \in \hat{U} \subseteq \mathbb{R}^{\hat{n}_u}$ and disturbance $\hat{w} \in \hat{W} \subseteq \mathbb{R}^{\hat{n}_w}$. Since the main goal of this first step is to reduce the complexity of the second step in Section 4 (exponential in the state-control dimension), we want to choose a continuous abstraction whose state and control dimensions satisfy $\hat{n}_x + \hat{n}_u < n_x + n_u$.

We introduce a map $\pi : \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_x}$ providing a reference trajectory to be followed by the concrete model, based on both the state and the control signals of the abstract model, while all other methods in the literature (see references in Section 1) only rely on the abstract state. We can then define the error $e \in \mathbb{R}^{n_x}$ between trajectories of the concrete and abstract models:

$$e = x - \pi(\hat{x}, \hat{u}). \tag{3}$$

In this paper, we use affine maps $\pi(\hat{x}, \hat{u}) = P[\hat{x}; \hat{u}] + \Omega$, where matrix $P \in \mathbb{R}^{n_x \times (\hat{n}_x + \hat{n}_u)}$ has at most one non-zero element per row, and $\Omega \in \mathbb{R}^{n_x}$.

The error dynamics resulting from (3) are given as

$$\dot{e} = f_e(e, \hat{x}, \hat{u}, w, \hat{w}) + g_e(e, \hat{x}, \hat{u}, w)u - \frac{\partial \pi(\hat{x}, \hat{u})}{\partial \hat{u}}\dot{\hat{u}}, \tag{4}$$

with $f_e(e, \hat{x}, \hat{u}, w, \hat{w}) = f(e+\pi(\hat{x}, \hat{u}), w) - \frac{\partial \pi(\hat{x},\hat{u})}{\partial \hat{x}} \hat{f}(\hat{x}, \hat{u}, \hat{w})$ and $g_e(e, \hat{x}, \hat{u}, w) = g(e+\pi(\hat{x}, \hat{u}), w)$. Let $E_0 \subseteq \mathbb{R}^{n_x}$ denote a compact set of initial conditions for the error system (4).

In Section 4, $\hat{u}$ is first designed as a discrete-time signal, then implemented in the abstract model (2) with a zero-order hold. This means

$$\begin{aligned} &\hat{u}(t) = \hat{u}(\tau_i),\ \forall t \in [\tau_i, \tau_{i+1}),\ \text{with } \tau_i = iT_s, \\ &\hat{u}(\tau_{i+1}) = \hat{u}(\tau_i) + \Delta\hat{u}(\tau_{i+1}), \end{aligned} \tag{5}$$

where T_s is the sampling period, $\Delta\hat{u}(t)$ is the periodic change in the abstract control, restricted to a set $\Delta\hat{U} \subseteq \mathbb{R}^{\hat{n}_u}$. Since the signal $\hat{u}$ is piecewise constant, we thus have

$$\dot{\hat{u}}(t) = 0,\ \forall t \in [\tau_i, \tau_{i+1}).$$

We initially focus our analysis of the error system (4) on the first sampling period $[0, T_s)$, before the input jump $\Delta\hat{u}$ at time T_s. Given the bounded set of initial conditions E_0, we want to enforce the boundedness of the error state during $[0, T_s)$ by introducing a low-level controller

$$u(t) = \kappa(t, e(t), \hat{x}(t), \hat{u}(t)), \tag{6}$$

defined by a time-varying, error-state feedback control law $\kappa : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_u}$. Below, we provide the design requirements on κ to obtain such an error bound.

Proposition 2. Given the error dynamics (4) and $\gamma \in \mathbb{R}$, $T_s > 0$, $\hat{X} \subseteq \mathbb{R}^{\hat{n}_x}$, $\hat{U} \subseteq \mathbb{R}^{\hat{n}_u}$, $\hat{W} \subseteq \mathbb{R}^{\hat{n}_w}$, $W \subseteq \mathbb{R}^{n_w}$, if there exists a $\mathcal{C}^1$ function $V : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}$, and $\kappa : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_u}$, such that

$$E_0 \subseteq \{e \mid V(0, e) \leq \gamma\}, \tag{7}$$

$$\begin{aligned} &\frac{\partial V(t, e)}{\partial e}(f_e(e, \hat{x}, \hat{u}, w, \hat{w}) + g_e(e, \hat{x}, \hat{u}, w)\kappa(t, e, \hat{x}, \hat{u})) \\ &\quad + \frac{\partial V(t, e)}{\partial t} \leq 0,\ \forall t, e, \hat{x}, \hat{u}, w, \hat{w},\ \text{s.t. } t \in [0, T_s), \\ &\quad V(t, e) = \gamma,\ \hat{x} \in \hat{X},\ \hat{u} \in \hat{U},\ w \in W,\ \hat{w} \in \hat{W}, \end{aligned} \tag{8}$$

then for all $e(0) \in E_0$, we have $e(t) \in \{e \mid V(t, e) \leq \gamma\}$, for all $t \in [0, T_s)$.

Although Proposition 2 is stated for the first sampling period $[0, T_s)$, it can be used for any other sampling period $[\tau_i, \tau_{i+1})$ with $\tau_i = iT_s$.

Corollary 3. Define the funnel $F(t) = \{e \mid V(t, e) \leq \gamma\} \subseteq \mathbb{R}^{n_x}$. For all $e(\tau_i) \in F(0)$, we have $e(\tau_i + t) \in F(t)$, for all $t \in [0, T_s)$, under the control signal $u(\tau_i + t) = \kappa(t, e(\tau_i + t), \hat{x}(\tau_i + t), \hat{u}(\tau_i + t)))$.

Next, we focus on the effect of the input jump $\Delta\hat{u}$ at the end of each sampling period as in (5). From (3), $\Delta\hat{u}$ induces a jump on the error described as follows, where τ_{i+1}^- and τ_{i+1}^+ denote sampling instant τ_{i+1} before and after the discrete jump, respectively:

$$\begin{aligned} e(\tau_{i+1}^+) &= x(\tau_{i+1}^+) - P[\hat{x}(\tau_{i+1}^+); \hat{u}(\tau_{i+1}^+)] - \Omega \\ &= x(\tau_{i+1}^-) - P[\hat{x}(\tau_{i+1}^-); \hat{u}(\tau_{i+1}^-) + \Delta\hat{u}(\tau_{i+1}^+)] - \Omega \\ &= e(\tau_{i+1}^-) - P\left[0; \Delta\hat{u}(\tau_{i+1}^+)\right]. \end{aligned}$$

We introduce the additional condition below to characterize the error jump induced by the control jump $\Delta\hat{u}$ in terms of the funnel from Corollary 3.

Proposition 4. Given $\gamma \in \mathbb{R}$, $\Delta\hat{U} \in \mathbb{R}^{\hat{n}_u}$, if there exists a function $V : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}$ satisfying

$$V(0, e - P[0; \Delta\hat{u}]) \leq \gamma,\ \forall e, \Delta\hat{u}, \qquad \text{s.t. } V(T_s, e) \leq \gamma,\ \Delta\hat{u} \in \Delta\hat{U}, \quad (9)$$

then for all $e(\tau_{i+1}^-) \in F(T_s)$, we have $e(\tau_{i+1}^+) \in F(0)$.

We next combine the conditions for the error boundedness for each sampling period and discrete jump from Propositions 2 and 4, respectively, to obtain the main result on the boundedness of the error at all time, formulated below and illustrated in Figure 2.

Theorem 5. If there exists V and κ satisfying (7)–(9), define $\varepsilon \in \mathbb{R}_+^{n_x}$ such that $\cup_{t\in[0,T_s)} F(t) \subseteq [-\varepsilon, \varepsilon]$. Then for all $t \geq 0$, $\hat{x}(t) \in \hat{X}$, $\hat{u}(t) \in \hat{U}$, $\Delta\hat{u}(t) \in \Delta\hat{U}$, $w(t) \in W$ and $\hat{w}(t) \in \hat{W}$, the error system (4) under control law $u(t) = \kappa(\tilde{t}, e(t), \hat{x}(t), \hat{u}(t)))$ with $\tilde{t} = (t \mod T_s) \in [0, T_s)$ satisfies: $e(0) \in E_0 \Rightarrow \forall t \geq 0,\ e(t) \in [-\varepsilon, \varepsilon]$.

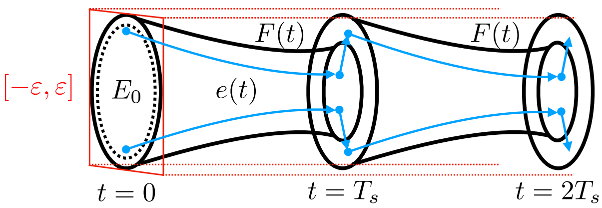

Fig. 2. Illustration of Theorem 5, with initial error set E_0, funnels F on each sampling period, bounded error jumps at sampling times. The red interval hull $[-\varepsilon, \varepsilon]$ of $\cup_{t\in[0,T_s)} F(t)$ bounds the error $e(t)$ for all times.

Finding storage functions V and control laws κ satisfying the constraints (7)–(9) is a difficult problem. In this paper, we use SOS programming to search for them at the cost of a restriction to polynomial candidates $V \in \mathbb{R}[(t, x)]$ and $\kappa \in \mathbb{R}^{n_u}[(t, x, \hat{x}, \hat{u})]$. For a general nonlinear system, least-squares regression, Taylor expansion and change of variables can be used to obtain a polynomial system (see e.g. Papachristodoulou and Prajna, 2002). By applying the generalized S-procedure (Parrilo, 2000) to (7)–(9), and choosing the volume of $F(t)$ as the cost function, we can define a non-convex optimization problem which is solved as in Smith et al. (2019) by solving a series of convex problems where we alternatingly fix the the decision variables V or κ. The detailed formulation of the optimization problem and the algorithm solving it are provided in the extended version of this paper. [2]

After finding the funnel $F(t)$ characterized by the resulting storage function V, computing the interval error bound $[-\varepsilon, \varepsilon] \subseteq \mathbb{R}^{n_x}$ is achieved by solving the optimization problem: $\min_\varepsilon \sum_{i=1}^{n_x} \varepsilon_i$, s.t. $F(t) \subseteq [-\varepsilon, \varepsilon]$ for all $t \in [0, T_s)$, which can be formulated as a convex SOS problem. Once ε is known, Theorem 5 implies that Problem 1 on the concrete model (1) with the reach-avoid-stay specification (X, X_a, X_r) can be solved through an auxiliary problem on the abstract model (2) with respect to the modified reach-avoid-stay specification $(X^{-\varepsilon}, X_a^{+\varepsilon}, X_r^{-\varepsilon})$ using the notations from Section 2 to shrink the state constraints $X^{-\varepsilon}$ and target set $X_r^{-\varepsilon}$ and expand the set of states to be avoided $X_a^{+\varepsilon}$ (see Figure 3 in Section 5 for an illustration of these sets). Since X, X_a, X_r are intervals of $\mathbb{R}^{n_x}$, the updated sets are also intervals. We can then define $\hat{X}^\varepsilon, \hat{X}_a^\varepsilon, \hat{X}_r^\varepsilon \subseteq \mathbb{R}^{\hat{n}_x}$ and $\hat{U}^\varepsilon, \hat{U}_a^\varepsilon, \hat{U}_r^\varepsilon \subseteq \mathbb{R}^{\hat{n}_u}$ as the projections of these sets into the abstract state-input space $\mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u}$ using the inverse image of $\pi : \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_x}$,

$$\hat{X}^\varepsilon \times \hat{U}^\varepsilon = \{(\hat{x}, \hat{u}) \in \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \mid \pi(\hat{x}, \hat{u}) \in X^{-\varepsilon}\},$$
$$\hat{X}_a^\varepsilon \times \hat{U}_a^\varepsilon = \{(\hat{x}, \hat{u}) \in \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \mid \pi(\hat{x}, \hat{u}) \in X_a^{+\varepsilon}\},$$
$$\hat{X}_r^\varepsilon \times \hat{U}_r^\varepsilon = \{(\hat{x}, \hat{u}) \in \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \mid \pi(\hat{x}, \hat{u}) \in X_r^{-\varepsilon}\}.$$

Due to the restriction of the affine map π in (3), all these hatted sets are intervals.

Problem 6. Given system (2) and subsets $\hat{X}_a^\varepsilon, \hat{X}_r^\varepsilon \subseteq \hat{X}^\varepsilon$ and $\hat{U}_a^\varepsilon, \hat{U}_r^\varepsilon \subseteq \hat{U}^\varepsilon$, find a set of initial states $\hat{X}_0 \subseteq \hat{X}^\varepsilon$ and a control strategy $\hat{\kappa} : \hat{X}^\varepsilon \to \hat{U}^\varepsilon \backslash \hat{U}_a^\varepsilon$ such that for any disturbance signal $\hat{w} : \mathbb{R}_+ \to \hat{W}$, all trajectories $\hat{x} : \mathbb{R}_+ \to \mathbb{R}^{\hat{n}_x}$ of the closed-loop system initialized in $\hat{X}_0$ satisfies $\hat{x}(t) \in \hat{X}^\varepsilon \backslash \hat{X}_a^\varepsilon$ for all $t \geq 0$ and there exists $\hat{t}_r \geq 0$ such that $\hat{x}(t) \in \hat{X}_r^\varepsilon$ and $\hat{\kappa}(\hat{x}(t)) \in \hat{U}_r^\varepsilon$ for all $t \geq \hat{t}_r$.

4. DISCRETE ABSTRACTION AND SYNTHESIS

In Section 3, we defined a continuous abstraction (2) and obtained from Theorem 5 a bound $\varepsilon \in \mathbb{R}^{n_x}$ on the error with respect to the concrete model (1). In this section, the second step of the proposed approach is to solve Problem 6 by creating a discrete abstraction of (2) and using classical model checking tools to synthesize a satisfying controller.

The discrete abstraction of (2) takes the form of a finite transition system defined as a triple $(\mathcal{X}, \mathcal{U}, \delta)$ containing a finite set of states $\mathcal{X}$, a finite set of inputs $\mathcal{U}$, and a non-deterministic transition relation $\delta : \mathcal{X} \times \mathcal{U} \to 2^{\mathcal{X}}$. We first take a finite partition of $\hat{X}^\varepsilon \backslash \hat{X}_a^\varepsilon$ into smaller intervals, where each partition element is represented by a unique discrete state (also called *symbol*) in $\mathcal{X}$. We add a last symbol $Out \in \mathcal{X}$ corresponding to the complement set $Out = \mathbb{R}^{\hat{n}_x} \backslash (\hat{X}^\varepsilon \backslash \hat{X}_a^\varepsilon)$, so that $\mathcal{X}$ is a partition of the whole state space $\mathbb{R}^{\hat{n}_x}$. Next, we define $\mathcal{U}$ as a finite discretization of the set of admissible control inputs $\hat{U}^\varepsilon \backslash \hat{U}_a^\varepsilon$.

Before defining the transition relation δ, we first introduce $\hat{x}(t; \hat{x}_0, \hat{u}, \hat{w})$ to denote the state reached at time $t \geq 0$ by the abstract model (2) starting in $\hat{x}_0$, with constant control input $\hat{u}$ and with disturbance signal $\hat{w} : [0, t] \to \hat{W}$. Then for an interval of initial states $[a, b] \subseteq \mathbb{R}^{\hat{n}_x}$, the finite time reachable set of (2) is defined as

$$\hat{R}(t, [a, b], \hat{u}) = \{\hat{x}(t; \hat{x}_0, \hat{u}, \hat{w}) \mid \hat{x}_0 \in [a, b],\ \hat{w} : [0, t] \to \hat{W}\}.$$

Then, given the time sampling T_s from Proposition 2, the set of successors associated to each pair $(s, \hat{u}) \in (\mathcal{X} \backslash \{Out\}) \times \mathcal{U}$ is defined as

$$\delta(s, \hat{u}) = \{s' \in \mathcal{X} \mid s' \cap \hat{R}(T_s, s, \hat{u}) \neq \emptyset\}. \quad (10)$$

Since the true reachable set $\hat{R}(T_s, s, \hat{u})$ can rarely be computed exactly, we can replace it in (10) by an over-approximation. Using over-approximations preserves the fact that any behavior of (2) can be reproduced by the discrete abstraction, which in turns ensures that a controller synthesized on the discrete abstraction also satisfies the desired specifications on (2). Several methods for the over-approximation of reachable sets using intervals and applicable to most nonlinear systems are described

[2] Available at: https://arxiv.org/abs/1911.09773

in Meyer et al. (2019). Problem 6 on (2) is then translated into a control problem on its discrete abstraction.

Problem 7. Find a set of initial symbols $\mathcal{X}_0 \subseteq \mathcal{X}\backslash\{Out\}$ and a control strategy $\mathcal{K} : \mathcal{X} \to \mathcal{U}$ such that any closed-loop trajectory of the discrete abstraction $s_0, s_1, \dots$ (i.e. such that $s_0 \in \mathcal{X}_0$ and $s_{i+1} \in \delta(s_i, \mathcal{K}(s_i))$ for all $i \geq 1$) satisfies $s_i \in \mathcal{X}\backslash\{Out\}$ for all $i \geq 0$, and there exists $k \in \mathbb{N}$ such that $s_i \subseteq \hat{X}_r^\varepsilon$ and $\mathcal{K}(s_i) \in \hat{U}_r^\varepsilon$ for all $i \geq k$.

The control synthesis on the discrete abstraction is achieved in two stages: first a safety game for the *stay* part of the specifications by computing a controlled invariant subset of the set of symbols fully included in the target interval $\{s \in \mathcal{X} \mid s \subseteq \hat{X}_r^\varepsilon\}$; then a reachability game for the *reach-avoid* part by iteratively finding all symbols that can be forced to reach the resulting safe set. Both these games are solved through classical fixed-point algorithms that are known to terminate in finite time and reach the maximal fixed-points (Tabuada, 2009). These algorithms are provided in the extended version of this paper. [3]

The final step of the overall approach is to refine the controllers $\mathcal{K} : \mathcal{X} \to \mathcal{U}$ obtained in this section and $\kappa : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_u}$ from Section 3 into a controller solving Problem 1 for the concrete system (1). We first introduce the function $H : \mathbb{R}^{\hat{n}_x} \to \mathcal{X}$ such that $H(\hat{x}) = s \Leftrightarrow \hat{x} \in s$, mapping each state of the continuous abstraction (2) to the unique symbol of the discrete abstraction containing it. We can then define a controller $\hat{\kappa} : \mathbb{R} \times \hat{X}^\varepsilon \to \hat{U}^\varepsilon\backslash\hat{U}_a^\varepsilon$ for (2) as the zero-order hold version of $\mathcal{K} : \mathcal{X} \to \mathcal{U}$ with sampling period T_s. For all index $k \in \mathbb{N}$ and time $t \in \mathbb{R}_+$ such that $kT_s \leq t < (k+1)T_s$, we have

$$\hat{\kappa}(t, \hat{x}(t)) = \mathcal{K}(H(\hat{x}(kT_s))). \tag{11}$$

Combining (11) with the low-level controller $\kappa : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{\hat{n}_x} \times \mathbb{R}^{\hat{n}_u} \to \mathbb{R}^{n_u}$ from (6), we obtain a controller $C : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{\hat{n}_x} \to \mathbb{R}^{n_u}$ for the concrete model defined as

$$C(t, x, \hat{x}) = \kappa(t, x - \pi(\hat{x}, \hat{\kappa}(t, \hat{x})), \hat{x}, \hat{\kappa}(t, \hat{x})). \tag{12}$$

Since this controller also depends on the abstract state $\hat{x}(t)$, its use in the concrete model (1) requires the computation of a trajectory of the closed-loop continuous abstraction, as stated in the main result of this paper below.

Theorem 8. Given $E_0 \subseteq \mathbb{R}^{n_x}$ bounding the initial error state (which is a design parameter in Proposition 2), the set of winning initial states for Problem 1 is $X_0 = \{\pi(\hat{x}, \mathcal{K}(H(\hat{x}))) \in \mathbb{R}^{n_x} \mid H(\hat{x}) \in R\} + E_0$. Given an initial state $x_0 \in X_0$, let $\hat{x} : \mathbb{R} \to \mathbb{R}^{\hat{n}_x}$ be any trajectory of the continuous abstraction (2) with controller $\hat{\kappa}$ in (11) initialized in $\{\hat{x}_0 \in \bigcup_{s \in R} s \mid x_0 - \pi(\hat{x}_0, \mathcal{K}(H(\hat{x}_0))) \in E_0\}$. Then, the closed-loop system (1) with controller (12) satisfies the reach-avoid-stay specification from Problem 1.

5. CASE STUDY: MARINE VESSEL

The autonomous docking maneuver consists of four phases: transit, transition from high speed to low speed maneuvering, docking, and dockside keeping a steady contact force with the dock. In this work we focus on the transition phase, which is challenging due to large changes in the ship dynamics when the speed is reduced. This means that unlike the general Problem 1, we only consider a *reach-avoid* specification to reach the area near the dock (light blue in Figure 3) while avoiding the piers (gray areas). The *stay* part of the specification is omitted as it is handled in the later docking and dockside phases.

The ship motion at moderate speed can be modeled as in Fossen (2011):

$$\dot{\eta} = R(\psi)\nu + v_c, \tag{13a}$$

$$M\dot{\nu} + C(\nu)\nu + D\nu = \tau + R(\psi)^\top \tau_{wind}, \tag{13b}$$

where $\eta = [N; E; \psi]$ are the South-North and West-East positions and heading of the ship ($\psi = 0$ points North, $\psi = \pi/2$ points East), $\nu = [u; v; r]$ are the surge and sway velocities, and yaw rate of the ship. $R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is a rotation matrix. $\tau \in \mathbb{R}^3$ is the control input affecting the three acceleration states of the ship. $v_c \in \mathbb{R}^3$ and $\tau_{wind} \in \mathbb{R}^3$ are disturbances corresponding to current velocities and wind forces. The inertia matrix including hydrodynamic added mass $M = \begin{bmatrix} 87.4 & 0 & 0 \\ 0 & 98.3 & 2.48 \\ 0 & 2.48 & 22.2 \end{bmatrix}$, damping matrix $D = \begin{bmatrix} 6.58 & 0 & 0 \\ 0 & 37.7 & 2.66 \\ 0 & 2.66 & 19.3 \end{bmatrix}$ and Coriolis matrix $C(\nu) = \nu(1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 98.3 \\ 0 & 0 & 2.48 \end{bmatrix}$ are chosen for a $1:30$ scale model of a platform supply vessel.

Using the notations from (1), we have state $x = [\eta; \nu] \in \mathbb{R}^6$, control input $u = \tau \in \mathbb{R}^3$ and disturbance input $w = [v_c; \tau_{wind}] \in \mathbb{R}^6$. The controls are unconstrained ($U = \mathbb{R}^3$) and the disturbances signals are assumed to be evolve in $W = [-0.01, 0.01]^5 \times [-0.05, 0.05]$. The chosen reach-avoid specification focuses on the first three states with the safety constraints $X = [0, 10] \times [0, 6.5] \times [-\pi, \pi] \times \mathbb{R}^3$, the obstacles $X_a = X_{a1} \cup X_{a2}$ with $X_{a1} = [2, 2.5] \times [0, 3] \times [-\pi, \pi] \times \mathbb{R}^3$ and $X_{a2} = [5, 5.5] \times [3.5, 6.5] \times [-\pi, \pi] \times \mathbb{R}^3$ (in grey in Figure 3), and the target set $X_r = [7, 10] \times [0, 6.5] \times [\pi/3, 2\pi/3] \times \mathbb{R}^3$ (light blue).

The continuous abstraction is chosen as the kinematics part of the concrete model (13):

$$\dot{\hat{\eta}} = R(\hat{\psi})\hat{\nu} + \hat{v}_c \tag{14}$$

where the abstract states, inputs and disturbances are $\hat{x} = \hat{\eta}$, $\hat{u} = \hat{\nu}$ and $\hat{w} = \hat{v}_c$. The map π is chosen as $\pi(\hat{x}, \hat{u}) = [\hat{x}; \hat{u}]$. However, instead of defining error as in (3), we redefine the error state as $e = \phi \cdot (x - \pi(\hat{x}, \hat{u}))$, where $\phi = \left[R^{-1}(\hat{\psi}), \mathbf{0}_{3\times3}; \mathbf{0}_{3\times3}, \boldsymbol{I}_3\right]$. The matrix ϕ allows to replace the trigonometric functions in $\hat{\psi}$ in the error dynamics (4) by trigonometric functions in $e(3) = (\psi - \hat{\psi})$, which can easily be approximated by polynomials in certain range of $e(3)$. The input, input jump, and disturbances spaces for the abstract model are $\hat{U} = [0, 0.18] \times [-0.05, 0.05] \times [-0.1, 0.1]$, $\Delta\hat{U} = [-0.18, 0.18] \times [-0.1, 0.1] \times [-0.2, 0.2]$, and $\hat{W} = [-0.01, 0.01]^3$. The SOS optimization problem is run with degree-2 polynomials to characterize the storage function V, control law κ, and multipliers s, l, and terminates in 6 minutes on a computer with 3.6GHz processor and 62GB of RAM. The resulting error bounds ε on (N, E, ψ) are $[-0.427, 0.427] \times [-0.432, 0.432] \times [-0.235, 0.235]$ and the expanded obstacles $X_a^{+\varepsilon}$ and shrunk target set $X_r^{-\varepsilon}$ are outlined in green in Figure 3. Due to the consideration of the abstract control $\hat{u} = \hat{\nu}$ in the error definition (3), the obtained error bounds are less conservative than those com-

[3] Available at: https://arxiv.org/abs/1911.09773

puted using Singh et al. (2018), that is $[-0.462, 0.462] \times [-0.493, 0.493] \times [-0.339, 0.339]$.

For the discrete abstraction as in Section 4, we take a uniform partition of $\hat{X}$ into 50 intervals per dimension (resulting in $|\mathcal{X}| = 125000$) and a uniform discretization of $\hat{U}$ into 9 values per dimension (i.e. $|\mathcal{U}| = 729$). To define the transition relation δ as in (10), we compute interval over-approximations of the reachable set of the continuous abstraction (14) using the continuous-time mixed-monotonicity approach implemented in the tool TIRA (Meyer et al., 2019). The choice of the partition granularity with respect to the sampling period $T_s = 3$ was done so that the reachable set would jump on average two to three partition cells away from its initial cell. This ensures that the transitions do not jump too far, while also avoiding self-loops which hinder the synthesis. On a server with 24 cores at 2.5GHz and 128GB of RAM, the abstraction is created in 10 seconds and the control synthesis is achieved after 15 hours, resulting in a winning set $R \subseteq \mathcal{X}$ covering 93% of the set of symbols $\mathcal{X}$.

The synthesized controller is then converted into the controllers (11) and (12) for the abstract (14) and concrete ship models (13), respectively. The initial state is chosen as a random point in the bottom left corner of the (N, E)-plane, and both closed-loop trajectories with random disturbance signals are plotted in red for (14) and blue for (13) in Figure 3. The black arrows represent the orientation ψ of the ship at each discrete time step. We can first note that the low-level controller (6) provides a very efficient tracking of the abstract model's trajectory (red) by the concrete model (blue). Both models satisfy their reach-avoid specifications by reaching the (shrunk) target set in blue while avoiding the (expanded) obstacles in grey. Once the ship has reached the desired $[N; E]$ position (blue set) but not the correct orientation ψ, we can see it slowly drift sideways while it turns to face East.

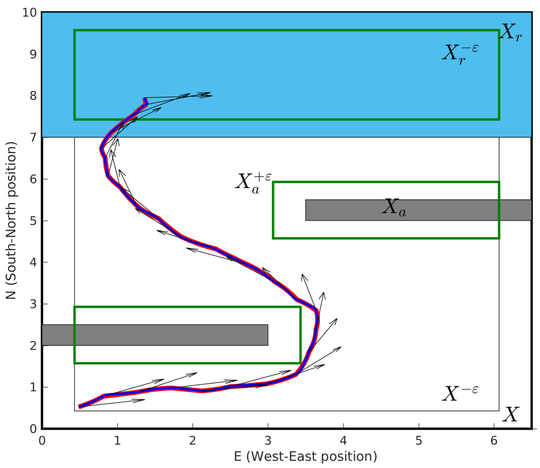

Fig. 3. Closed-loop trajectories of the abstract (red) and concrete (blue) models in the (N, E)-plane with the ship heading ψ (black arrows), the initial and shrunk state constraints X and $X^{-\varepsilon}$ (thick and thin black lines), the target set X_r (light blue), the obstacles X_a (grey) and the shrunk target set $X_r^{-\varepsilon}$ and expanded obstacles $X_a^{+\varepsilon}$ (green).

REFERENCES

Ames, A.D., Tabuada, P., Schürmann, B., Ma, W.L., Kolathaya, S., Rungger, M., and Grizzle, J.W. (2015). First steps toward formal controller synthesis for bipedal robots. In *18th International Conference on Hybrid Systems: Computation and Control*, 209–218.

Antoulas, A.C., Sorensen, D.C., and Gugercin, S. (2001). A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280, 193–219.

Colombo, A. and Girard, A. (2013). An approximate abstraction approach to safety control of differentially flat systems. In *European Control Conference*, 4226–4231.

DNV GL (2018). *Class guidelines for Autonomuos and remotely operated ships.* DNVGL-CG-0264, Edition September 2018, https://www.dnvgl.com/maritime/autonomous-remotely-operated-ships.

Fainekos, G.E., Girard, A., Kress-Gazit, H., and Pappas, G.J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352.

Fossen, T.I. (2011). *Handbook of Marine Craft Hydrodyamics and Motion Control.* Wiley.

Girard, A. and Pappas, G.J. (2009). Hierarchical control system design using approximate simulation. *Automatica*, 45(2), 566–571.

Meyer, P.J., Devonport, A., and Arcak, M. (2019). TIRA: Toolbox for interval reachability analysis. In 22^{nd} *ACM International Conference on Hybrid Systems: Computation and Control*, 224–229.

Mickelin, O., Ozay, N., and Murray, R.M. (2014). Synthesis of correct-by-construction control protocols for hybrid systems using partial state information. In *American Control Conference*, 2305–2311.

Papachristodoulou, A. and Prajna, S. (2002). On the construction of lyapunov functions using the sum of squares decomposition. In *IEEE Conference on Decision and Control*, 3482–3487.

Parrilo, P. (2000). Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. *PhD thesis, California Institute of Technology.*

Pola, G., Pepe, P., and Di Benedetto, M.D. (2017). Decentralized supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, 63(9), 2803–2817.

Reissig, G., Weber, A., and Rungger, M. (2016). Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4), 1781–1796.

Singh, S., Chen, M., Herbert, S.L., Tomlin, C.J., and Pavone, M. (2018). Robust tracking with model mismatch for fast and safe planning: an SOS optimization approach. In *Workshop on Algorithmic Foundations of Robotics.*

Smith, S., Yin, H., and Arcak, M. (2019). Continuous abstraction of nonlinear systems using sum-of-squares programming. *arXiv preprint arXiv:1909.06468.*

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach.* Springer Science & Business Media.