Continuous and discrete abstractions for planning applied to ship docking

Pierre-Jean Meyer, He Yin, Murat Arcak (UC Berkeley)
Astrid Brodtkorb, Asgeir Sørensen (NTNU)

IFAC 2020

Motivation

Problem definition

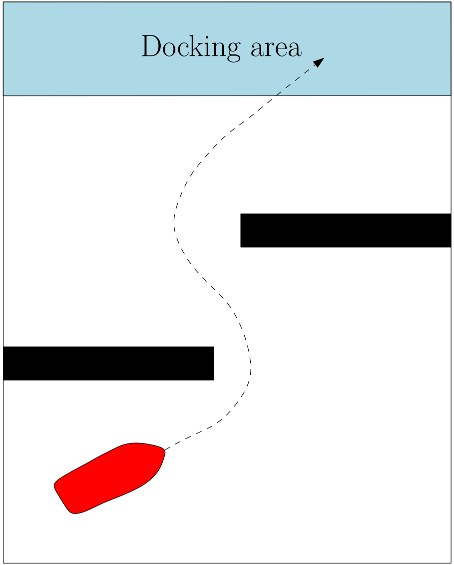
- Nonlinear system: $\dot{x} = f(x, u, w)$
- Reach-Avoid-Stay specification

Application to ship docking

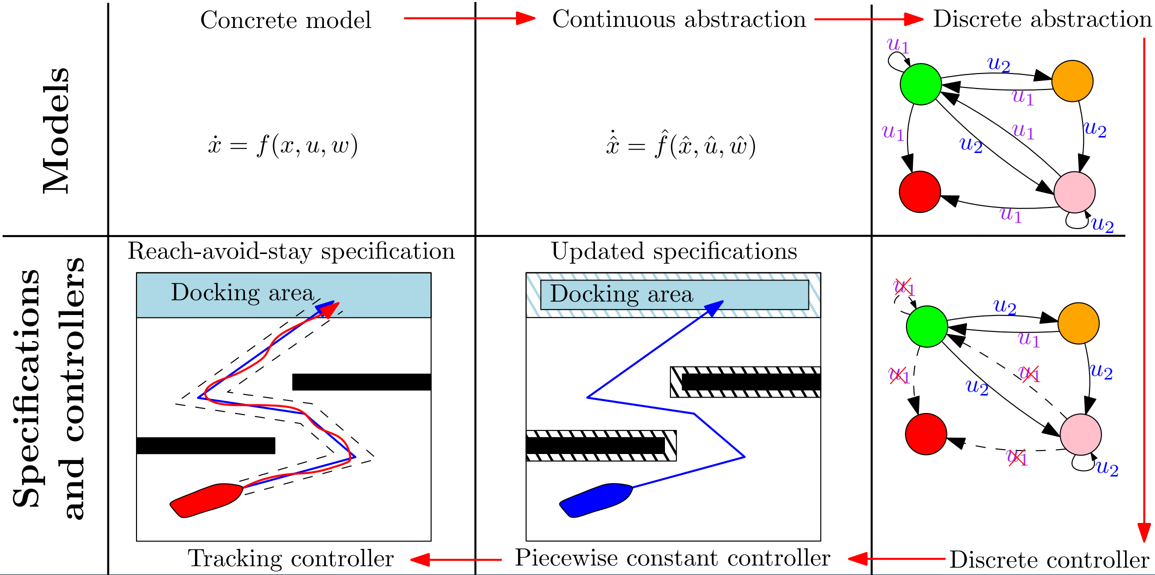
- Reach then stay in docking area
- Avoid obstacles

Correct-by-construction approach

- Offline path planning and formal methods
- Formal guarantees
- Reduced need for simulation-based testing



Overview



Concrete model
Continuous abstraction
Discrete abstraction
Models
$\dot{x} = f(x, u, w)$
$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \hat{w})$
Specifications and controllers
Reach-avoid-stay specification
Updated specifications
Docking area
Docking area
Tracking controller
Piecewise constant controller
Discrete controller

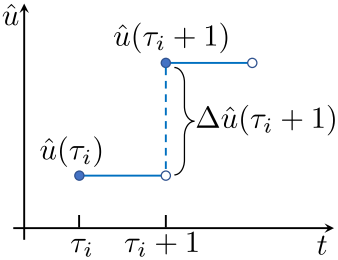
Continuous abstraction - Error definition

Error $e = x - \left(P \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix} + \Omega\right)$ between systems $\dot{x} = f(x, u, w)$ and $\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \hat{w})$

- Novelty: **depends explicitly on abstract input** $\hat{u}$
- Error dynamics: $\dot{e} = f_e(e, \hat{x}, u, \hat{u}, w, \hat{w})$
- **Goal**: tracking controller $u(t) = \kappa(t, e(t), \hat{x}(t), \hat{u}(t))$ to bound the error around 0

Challenge: discrete abstraction creates **piecewise constant control input** $\hat{u}$

- bound error for constant $\hat{u}$
- bound error for jumps $\Delta\hat{u}$

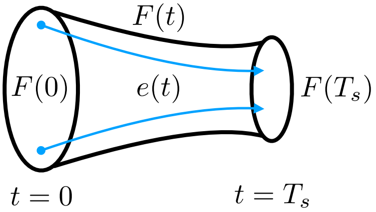


Key idea: bound a Lyapunov function $V(t,e)$ by some constant $\gamma \in \mathbb{R}$
$\rightarrow$ define funnel $F(t) := \{e : V(t,e) \leq \gamma\}$ of allowed errors trajectories

Bounding error flow with constant $\hat{u}$

- Constraints on $\kappa(t, e(t), \hat{x}(t), \hat{u}(t))$ and $V(t,e)$ such that

$$e(0) \in F(0) \implies \forall t \in [0, T_s), e(t) \in F(t)$$



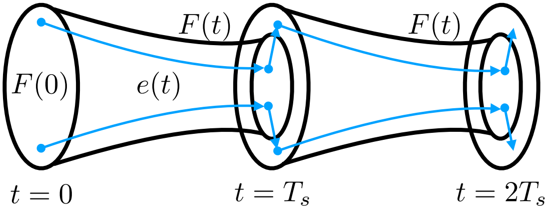
Continuous abstraction - Bounding the error

Key idea: bound a Lyapunov function $V(t, e)$ by some constant $\gamma \in \mathbb{R}$
$\rightarrow$ define funnel $F(t) := \{e : V(t, e) \leq \gamma\}$ of allowed errors trajectories

Bounding error jumps

- Additional constraint on $V(T_s, e(T_s))$ such that

$$e(T_s^-) \in F(T_s) \implies e(T_s^+) \in F(0)$$

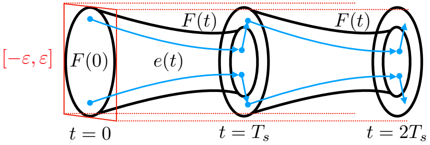


Main result: such choices of κ and V ensure $e(0) \in F(0) \implies \forall t \geq 0, e(t) \in F(t)$

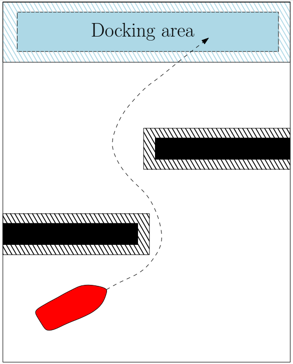
Assumption

- *Concrete system $\dot{x} = f(x, u, w)$ is control affine*
- *Error dynamics $\dot{e} = f_e(e, \hat{x}, u, \hat{u}, w, \hat{w})$ are polynomial*
- *State and input constraints $\hat{X}$, $\hat{U}$, $\Delta\hat{U}$, W, $\hat{W}$ are semi-algebraic sets*

- Restrict V and κ to polynomial functions
- SOS optimization problem to **minimize the funnel volume**
- Bound the funnel by an interval $[-\varepsilon, \varepsilon]$

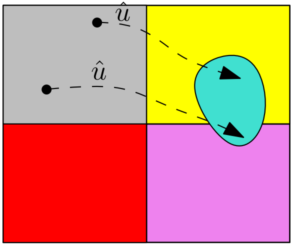


- Update reach-avoid-stay specifications



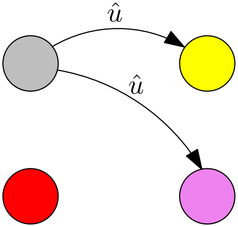
Discretization of $\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \hat{w})$

- Time: fixed sampling time T_s
- Input: finite discretization of $\hat{U}$
- State: uniform partition of $\hat{X}$



Discrete transitions

- Reachability analysis on $\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}, \hat{w})$
- Reachable set over-approximated by an interval [1]

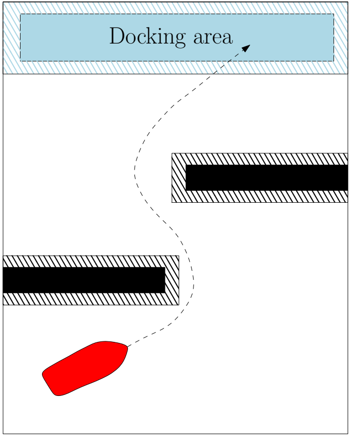


[1] Meyer, Devonport, Arcak. *TIRA: Toolbox for Interval Reachability Analysis*, HSCC 2019

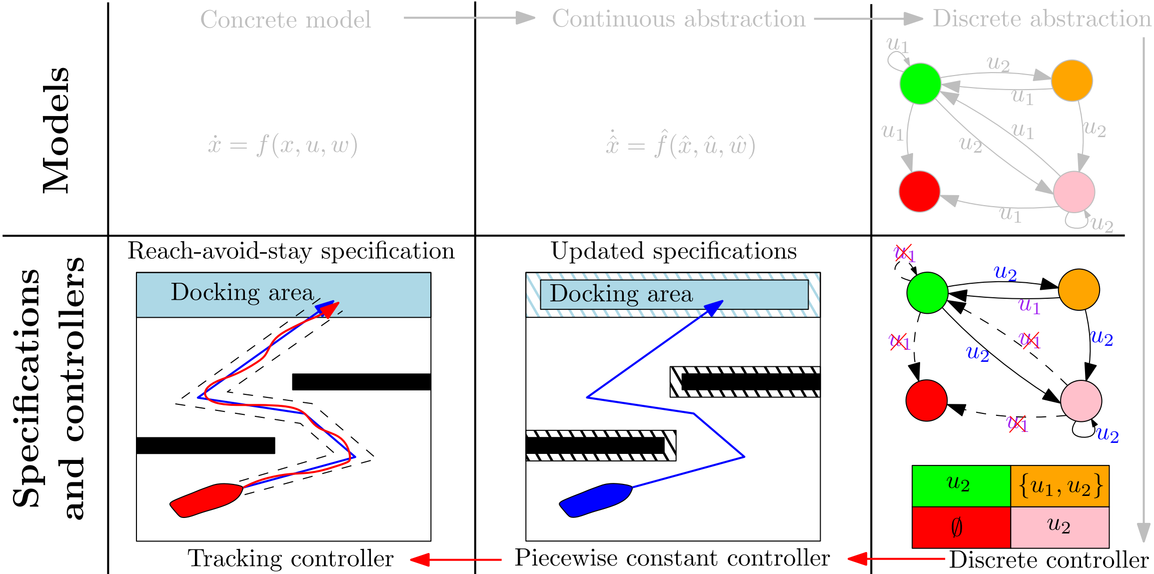
Discrete control synthesis

2-step synthesis

- Safety game: STAY in target set
  - initialize safe set as target set
  - iteratively remove states
    that cannot be kept inside safe set
- Reachability game: REACH safe subset of target set
  - initialize reach set as safe set
  - iteratively add states
    that can be brought to reach set in a single step
- (AVOID specification handled during abstraction creation)



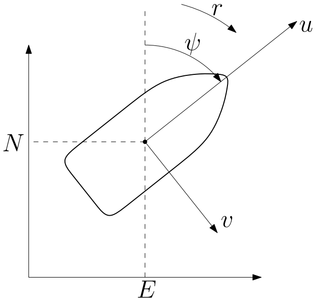
Controller refinement



Models of a marine vessel

6D concrete model

- States: 3 positions (N, E, ψ), 3 velocities (u, v, r)
- Inputs: thrusts
- Disturbances: currents + wind



3D continuous abstraction

- States: 3 positions
- Inputs: 3 velocities
- Disturbances: currents

Model ship (length 1.97 m)

Ship abstractions and synthesis

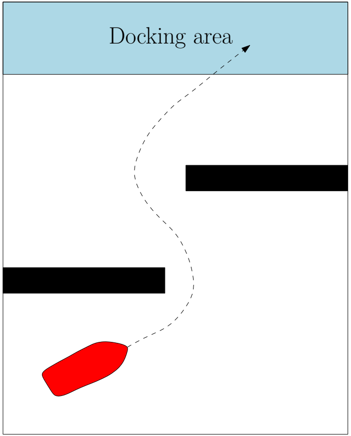
Continuous abstraction procedure

- Error on positions and velocities: $e = x - \begin{pmatrix} \hat{x} \\ \hat{u} \end{pmatrix}$
- Computation of tracking controller and error bounds: 6 min

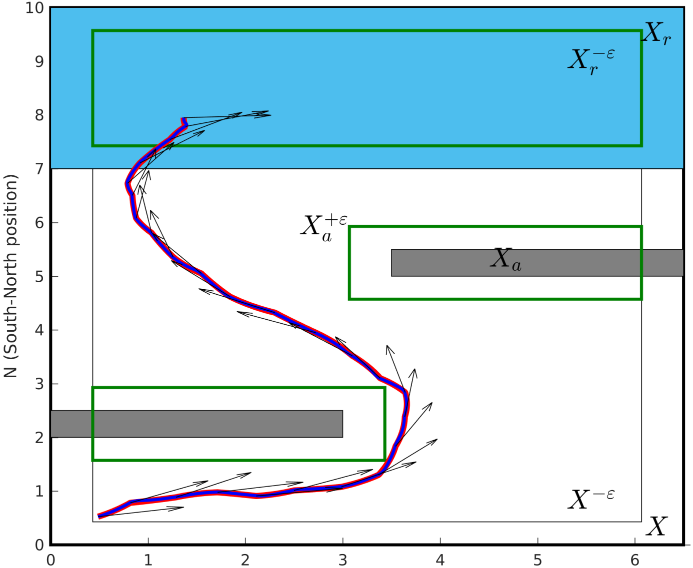
Discrete abstraction procedure

- Sampling period: 3 s
- Control discretization into 729 values (9^3)
- State partition into 125000 cells (50^3)
- Computation times: abstraction 10 s, synthesis 15 h

Results: 93% of state space is controllable



Simulation results



Concrete model

Continuous abstraction

Black arrows: ship heading

Good tracking despite conservative error bound

Conclusions

- **Formal control synthesis** for **reach-avoid-stay** problems
- Combining **continuous and discrete abstractions**
- Makes discrete abstraction **applicable to larger systems**

Current efforts: **experimental validation** on model ship at NTNU

Marine Cybernetics Laboratory, NTNU