Symbolic Planning for Dynamic Robots

Laya Shamgah
Advisor: Dr. Karimoddini

TECHLAV Project

- **Testing, Evaluation and Control of Heterogeneous Large-scale Autonomous systems of Vehicles (TECHLAV)**
- **Thrust 1: Modeling, Analysis and Control of Large-scale Autonomous Vehicles (MACLAV)**
- **Task 1-5: Hierarchical Hybrid Cooperative Control of LSASV**

TECHLAV Project

Objective

Developing capable and scalable models for autonomous collaborative, robust and distributed decision-making, group coordination, planning, and tasking through effective interaction with human operators

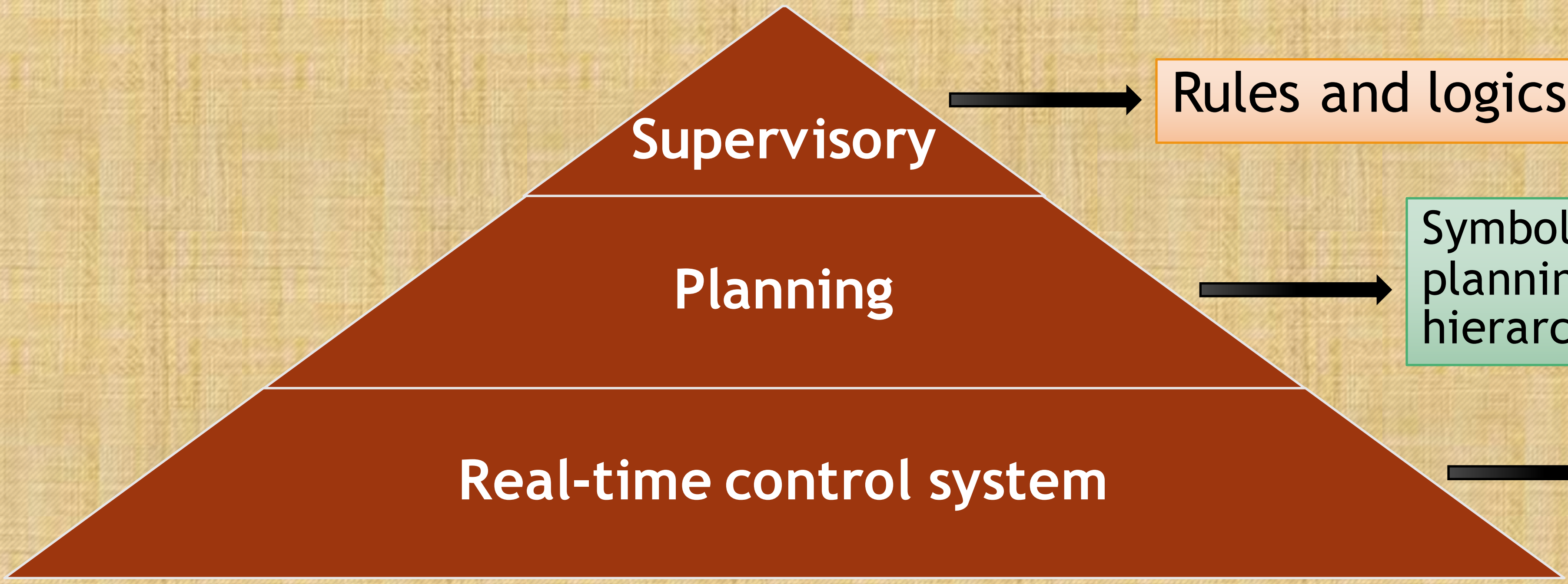
Impact

Reaching higher levels of autonomy and teaming of multi-agent systems by using agents which are able to accomplish assigned missions autonomously and have the capability of autonomous collaboration with other teammates and human operators.

Relation to TECHLAV

Task 1-5: Hierarchical Hybrid Cooperative Control of LSASV

Hierarchical Control



Rules and logics,…

Symbolic planning facilitates planning and coordination in a hierarchical control structure.

Actuators,…

Outline

1. Robot Motion Planning and Control
2. Task Specification
3. Temporal Logic
4. Symbolic Planning
5. Future Work

Mobile Robots

Components

- mechanical elements (wheels and gears)
- electromechanical devices (motors, clutches and brakes)
- digital circuits (processors and smart sensors)
- software programs (embedded controllers)

Constraints

- mechanical constraints (e.g., a car-like robot cannot move sideways)
- limited energy resources, and computation, sensing, and communication capabilities.

Complexities

- Environment is cluttered with possibly moving and shape changing obstacles.
- Their objectives can change over time, such as in the case of appearing and disappearing targets.

New Task Requirements for Robots

- Converging to a desired operating point while always staying within a safe set
- Executing sequenced tasks
- Reaching certain areas and visiting certain areas infinitely often
- Avoiding obstacles

Robot Motion Planning and Control

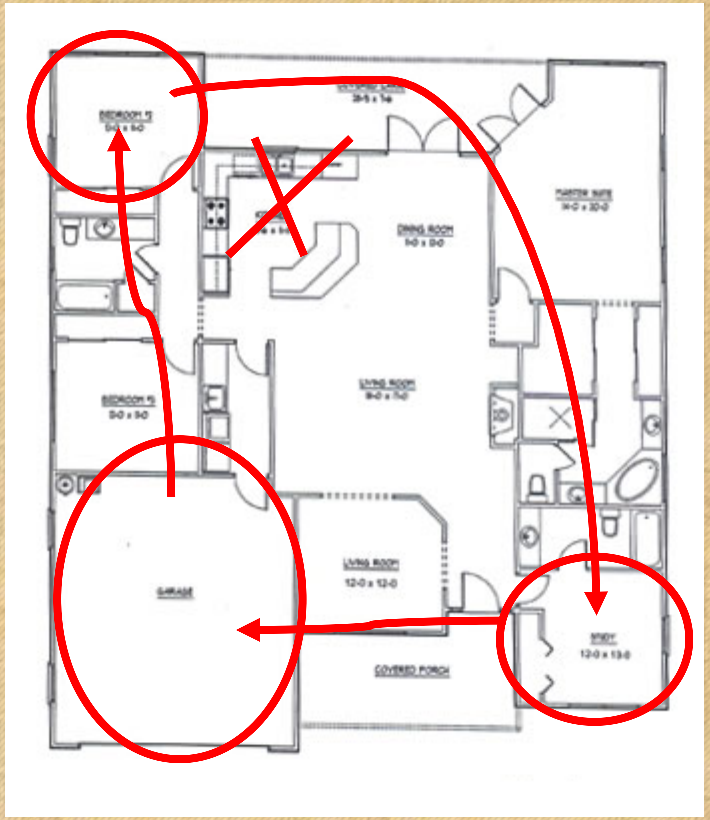
Problem

- automatic construction of robot control strategies from task specifications given in high-level, human-like language

Challenge

- the development of frameworks for control design which is:
  - computationally efficient
  - allowing for systematic, provably correct, control design
  - accommodating both the robot constraints and the complexity of the environment
  - allowing for expressive human-like task specifications

Task Planning



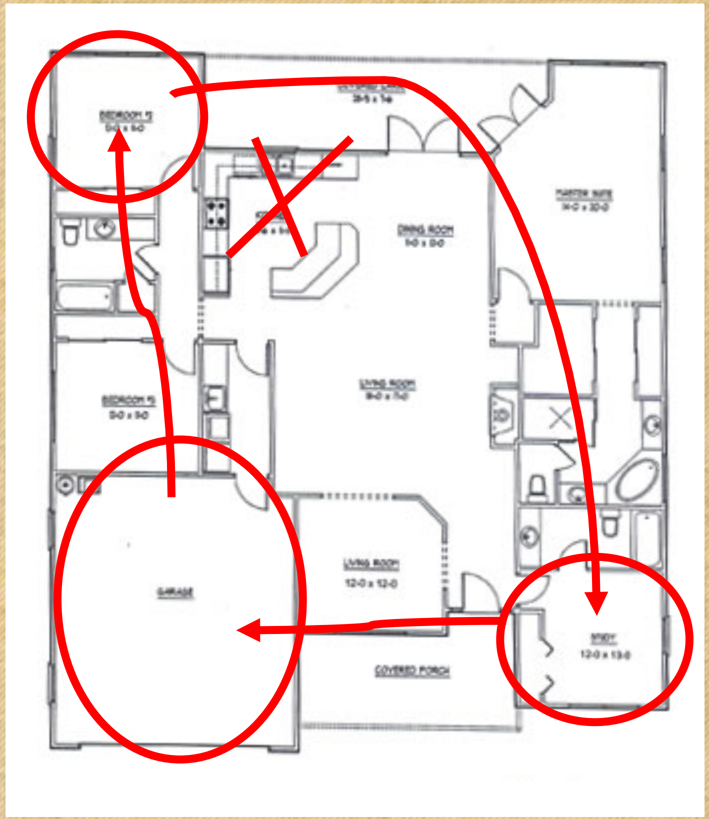
Mission requirements:

- converging to a desired operating point while always staying within a safe set
- executing sequenced tasks
- reaching certain areas and visiting certain areas infinitely often
- avoiding obstacles

How to formally describe the task?

Temporal logic

Why Temporal Logic?



- A formal high level language to describe a complex mission
- A wider class of properties than safety and stability
- Having well defined syntax and semantics, which can be easily used to specify complex behavior

Introduction to Temporal Logic

Temporal logic:

Any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time.

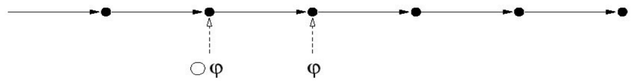
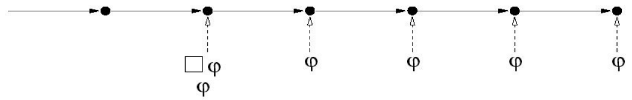
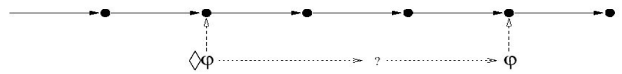
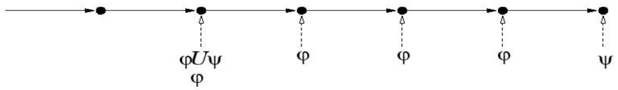
Classical logic:

- “It is Monday”

Temporal logic

- "I am *always* hungry“
- "I will *eventually* be hungry“
- "I will be hungry *until* I eat something"

Temporal Operators

Operators	Definition	Diagram
$\circ\varphi$	φ in true in the next moment of time	
$\square\varphi$	φ is true in all future moments	
$\Diamond\varphi$	φ is true in some future moment	
$\varphi u \psi$	φ is true until $\psi$is true	

Temporal Syntax

$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi$	
P	an arbitrary atomic proposition
$\neg$	Negation (complement)
$\wedge$	conjunction
$\vee$	disjunction
$\rightarrow$	implication

Example:
$(p \wedge \neg q) \rightarrow r$

Temporal Semantics

- A tool for specification, formal analysis, and verification of the executions of computer programs and systems

$\mathcal{M} = \langle \mathbb{N}, I \rangle$	a discrete, linear model of time
$I: \mathbb{N} \rightarrow 2^{\Sigma}$	maps each natural number to a set of propositions
$\vDash: (\mathcal{M} \times \mathbb{N} \times FORM) \rightarrow \{true, false\}$	Satisfaction relation

$\langle \mathcal{M}, i \rangle \vDash \mathrm{p}$ **iff** $p \in I(i)$ **(for** $p \in \Sigma$**)**

$$\langle \mathcal{M}, i \rangle \models \neg\varphi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \not\models \varphi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \wedge \psi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \models \varphi \text{ and } \langle \mathcal{M}, i \rangle \models \psi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \vee \psi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \models \varphi \text{ or } \langle \mathcal{M}, i \rangle \models \psi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \Rightarrow \psi \quad \textbf{iff} \quad \text{if } \langle \mathcal{M}, i \rangle \models \varphi \text{ then } \langle \mathcal{M}, i \rangle \models \psi$$

$$\mathcal{M}, i \models \top$$

$$\mathcal{M}, i \not\models \bot$$

LTL for Robot Task Specifications

- **Coverage:** eventually visit all regions
- **Sequencing:** visit P2 before you go to P3
- **Avoidance:** until you go to P2 avoid P1 and P3

Examples:

1. "visit rooms 1,2,3 while avoiding corridor 1!"

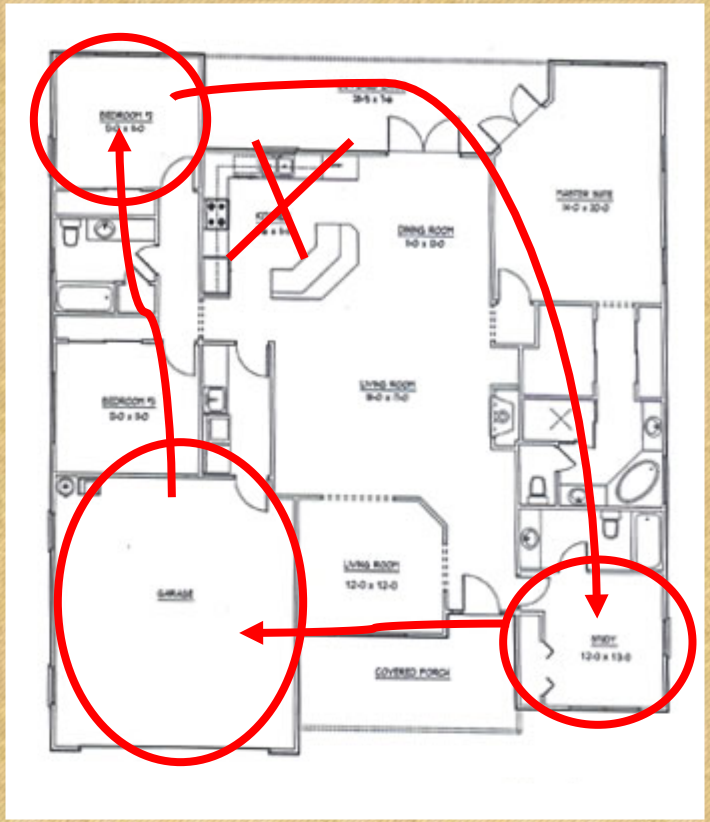
$$\varphi = \Box\neg(\text{corridor1}) \wedge \Box(\text{room1}) \wedge \Box(\text{room2}) \wedge \Box(\text{room3})$$

2. "if the light is on, visit rooms 1 and 2 infinitely often!"

$$\varphi = \Box(\text{lightOn}) \rightarrow (\Box\Diamond(\text{room1}) \wedge \Box\Diamond(\text{room2}))$$

3. "if you are in room 3 and Mike is there, beep!"

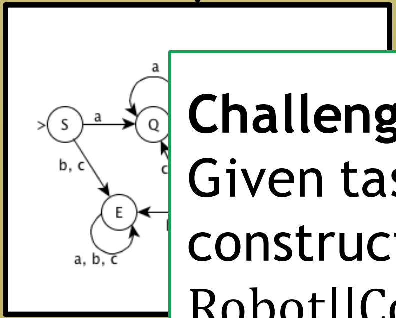
$$\varphi = \Box((\text{room3}) \wedge (\text{SeeMike}) \rightarrow (\text{beep}))$$



Symbolic Planning

Task

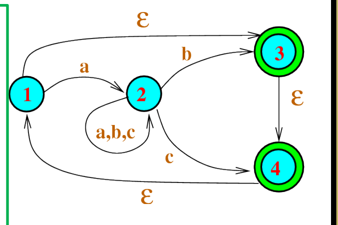
$\varphi = \square(\ (\text{room3}) \wedge (\text{SeeMike}) \rightarrow (\text{beep}))$



Hybrid Controller

Challenge:
Given task φ and a dynamical model of the robot construct controllers such that:
$\text{Robot}||\text{Controller} \ \vDash \varphi$

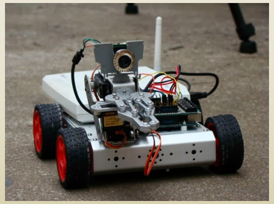
Abstraction



Interface

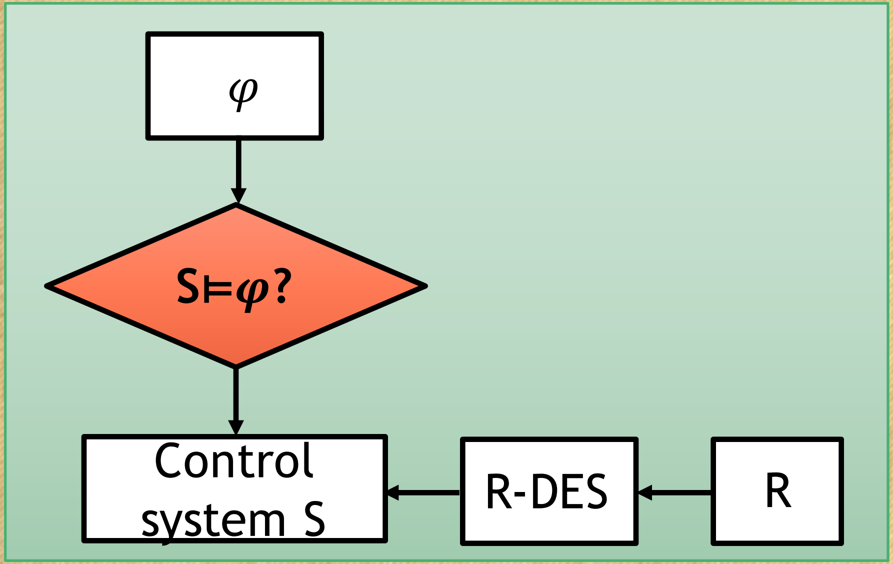
Robot

$\dot{x}(t) = u(t)$
$x(t) \in P \subset \mathbb{R}^2$
$u(t) \in U \subset \mathbb{R}^2$

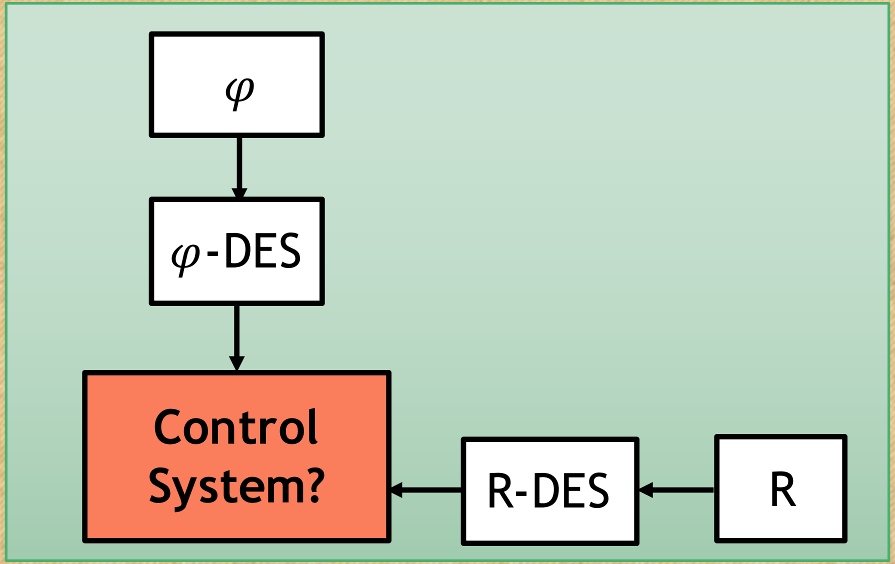


Control Synthesis

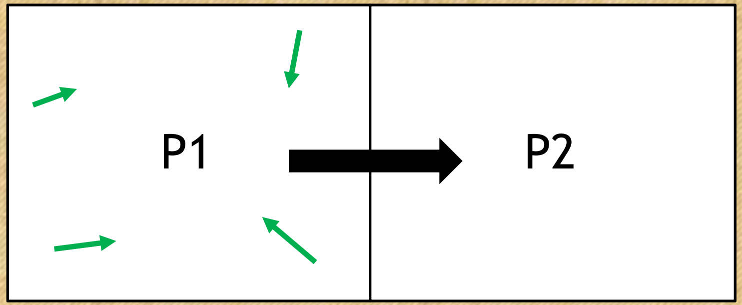
Verification



Planning



Applying the DES Supervisor to the Robot



Challenge:

- DES supervisor tells the robot to go from P1 to P2.
- The planner has no idea about the status of robot.

Question:

- How to map the high-level task into the continuous controller?

Bisimulation Equivalency

- We have to couple equivalency to specification.

Language equivalency →

- Preserves LTL properties
- Continuous system and DES generate the same trajectories

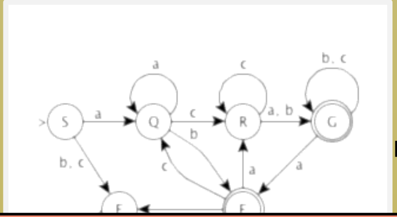
Bisimulation equivalency →

- Preserves LTL properties
- Continuous system and DES are equivalent in all states (stronger)

Symbolic Planning

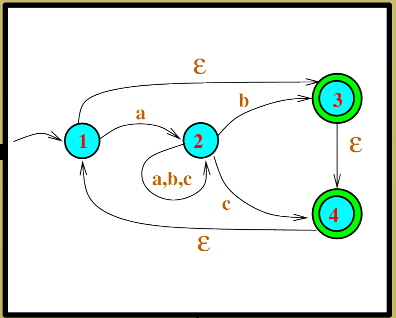
Task

$\varphi = \square(\text{(room3)} \wedge \text{(SeeMike)} \rightarrow \text{(beep)})$



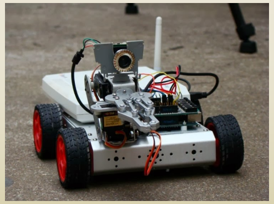
DES supervisor

Bisimulation equivalent



Robot

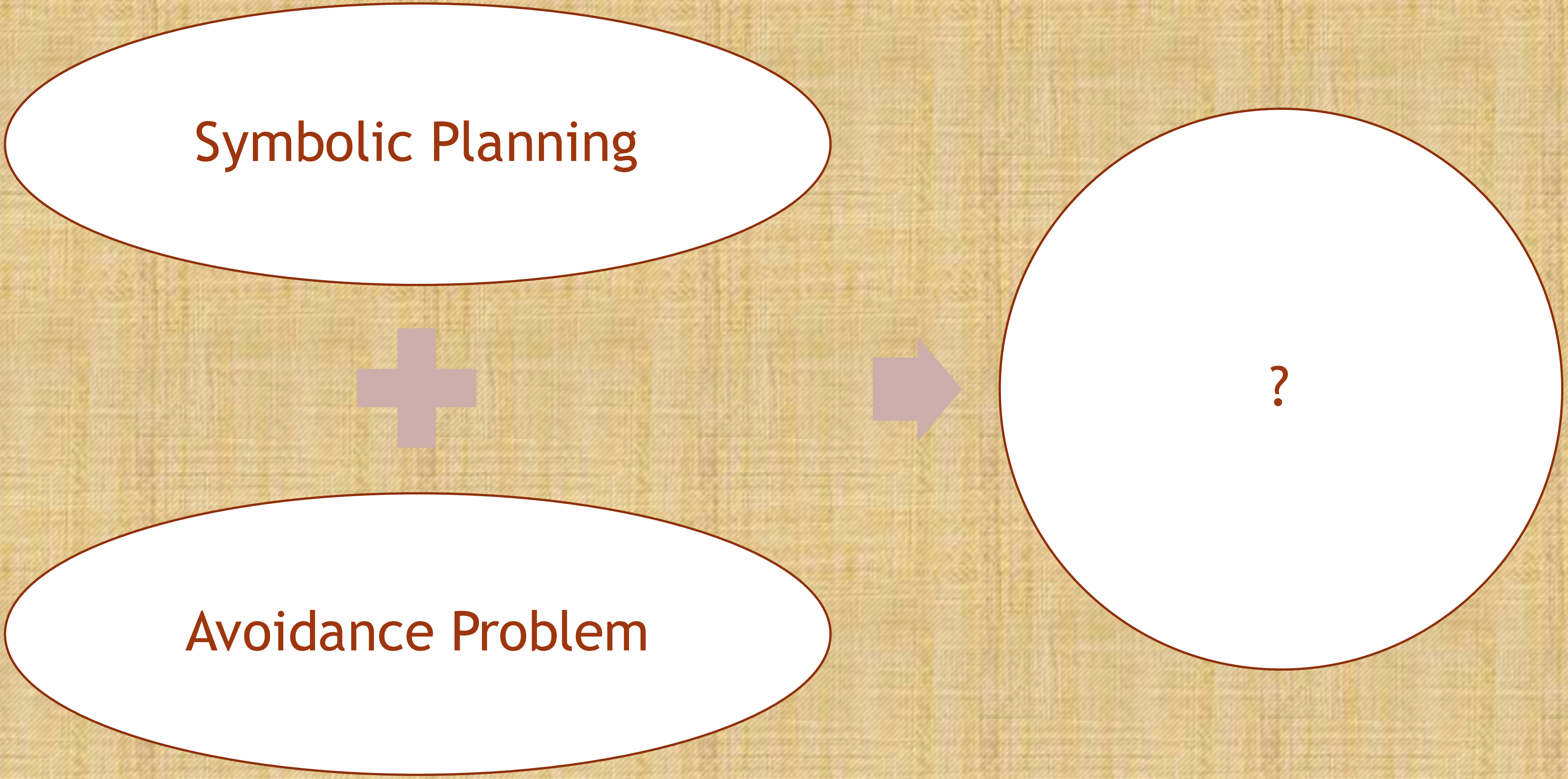
$\dot{x}(t) = u(t)$
$x(t) \in P \subset \mathbb{R}^2$
$u(t) \in U \subset \mathbb{R}^2$



Interface

if (DES supervisor ⊨φ)
then
(hybrid controller||robot ⊨ φ)

Future Work



North Carolina Agricultural and Technical State University

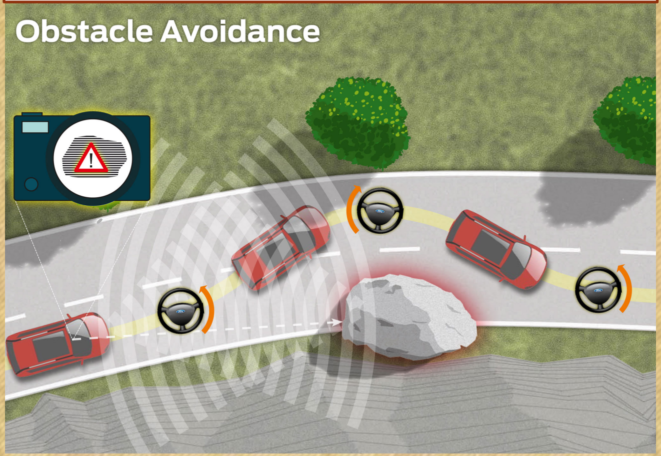
TECHLAV

ACCESS Laboratory

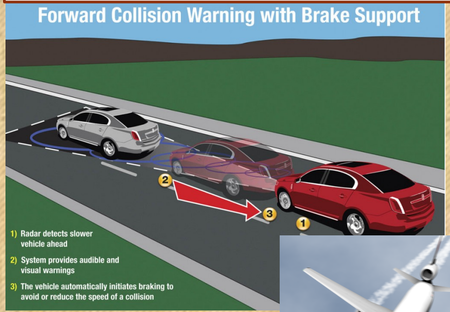
Future Work

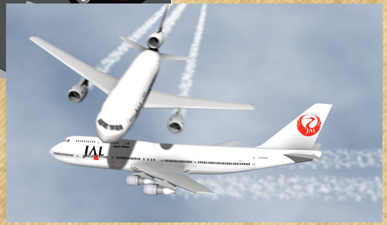
Avoidance problem

Obstacle avoidance

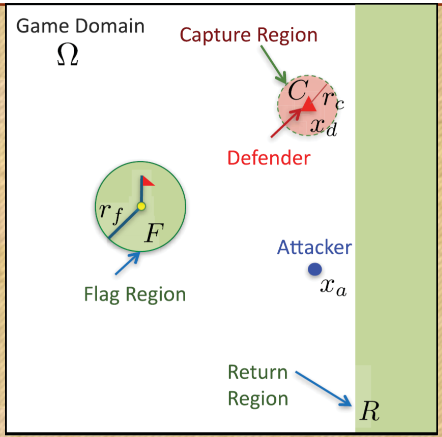


Collision avoidance





Reach-avoidance



Future Work

Reach-avoidance

Graph theory	Discrete games	Path planning with obstacle	model-predictive control	Differential game
Partitions the graph into regions that are defended and cleared by a team of pursuers to solve simplified path-planning	Visibility-based pursuit-evasion games: a group of searchers attempt to bring an evader into their field of view, through cooperative coverage of regions	Use the possible future positions of the obstacles as static obstacles, allowing using static planning methods(Certain applications with simple configurations of moving adversarial obstacles,).	Predicts opponent actions so that optimization can be performed for the controlled agents with respect to the assumed opponent behavior	Winning strategies for the opposing agents can be viewed as the solution to a zero-sum differential game. Simple tasks.

1. Formal description of tasks
2. Efficient computation
3. Combining reaching problem with avoiding problem
4. Targeting more complex tasks (sequencing and …)

References

- “Hierarchical Hybrid Symbolic Robot Motion Planning And Control”, Ali Karimoddini and Hai Lin, Asian Journal of Control, Vol. 17, No. 1, pp. 23–33, January 2015.
- “Temporal Logic Planning and Control of Robotic Swarms by Hierarchical Abstractions”, M. Kloetzer and C. Belta, IEEE TRANSACTIONS ON ROBOTICS, VOL. 23, NO. 2, APRIL 2007.
- “Optimal Path Planning under Temporal Logic Constraints”, Stephen L. Smith, Jana Tumova, Calin Belta, Daniela Rus, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, Taipei, Taiwan.
- “MDP Optimal Control under Temporal Logic Constraints”, Xu Chu Ding Stephen L. Smith Calin Belta Daniela Rus,CDC 2011.
- “Automatic Synthesis of Robust Embedded Control Software”, Tichakorn Wongpiromsarn, Ufuk Topcu and Richard M. Murray.
- “Flight Demonstrations of Cooperative Control for UAV Teams”, Jonathan How, Ellis King, and Yoshiaki Kuwata, AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, 20 - 23 September 2004, Chicago, Illinois.

Thank you ☺

Question?