Trusted Autonomy

John S. Baras
Institute for Systems Research
University of Maryland

Control Systems Quest for Autonomy

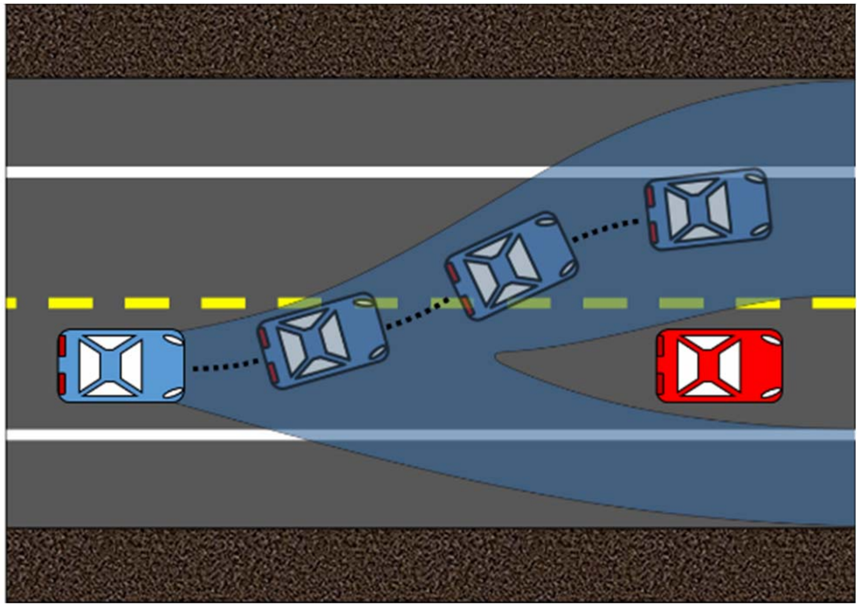
A Symposium in Honor of Professor Panos J. Antsaklis

October 27-28, 2018

University of Notre Dame University

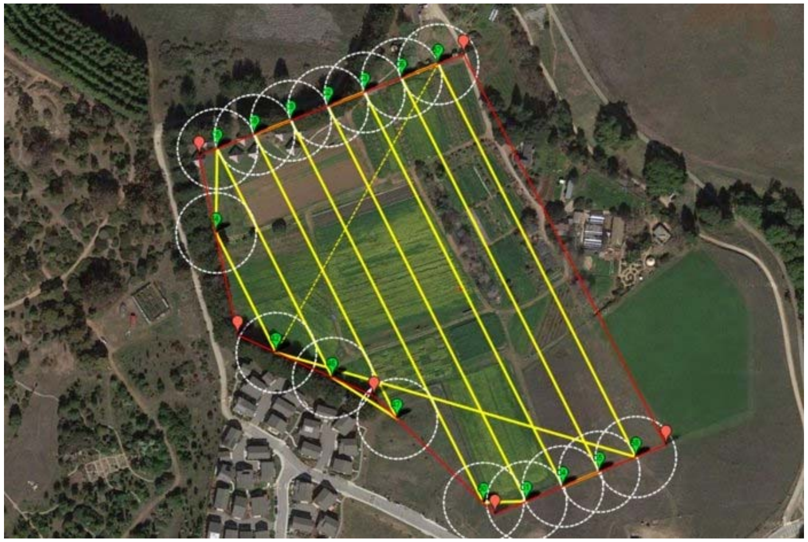
Motivation

- Safety
  - – UAVs in commercial airspace
  - – Autonomous vehicles & human-driven cars
- Human involvement
  - – Safety is critical and fundamental
- Physical limitation
  - – To avoid states that lead to unavoidable collision



Motivation

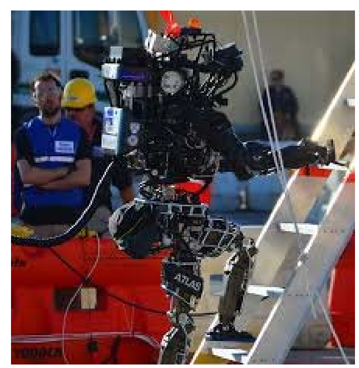
- Synthesize plan from task specifications
  - Agriculture monitor
  - Security and surveillance
  - Search and rescue
  - Disaster relief / Emergency communications
- Perform task in an optimal manner with given time constraints



The Institute for Systems Research

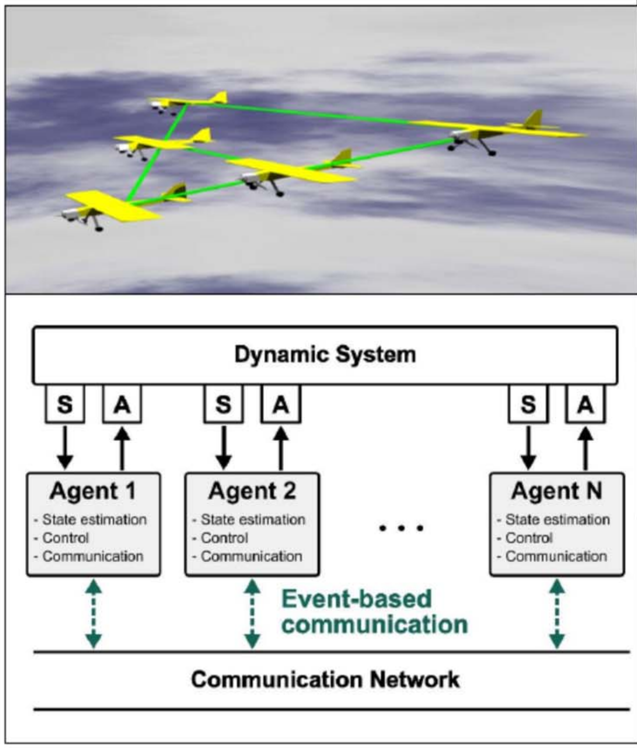
Collaborative Autonomy and Trust

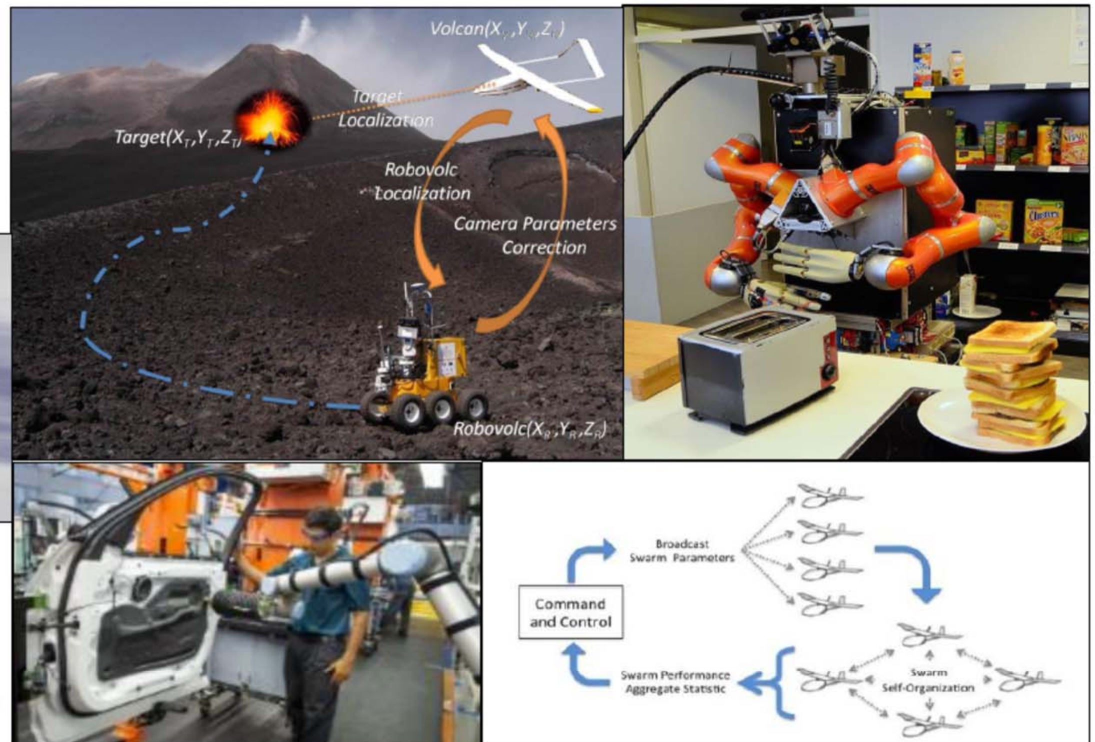
University of Maryland 1856



Motivation

Motivation





The finite time logical constraints may arise due to the complex task description or decision making process, while the information constraints emerge as a consequence of the limitations on communication and computation capabilities.

Intelligent and Learning Autonomous Systems: Composability and Correctness

- Formal models of tasks and missions combining spatial and temporal tolerances (both deterministic and stochastic)
- Contract-based design methodology for composability
- Self-monitoring and self-learning and self-adjustment for correct autonomous execution of tasks
- Integrate formal models, associated model-checking and contract-based design with the rigorous model-based systems engineering methodology and framework we have developed

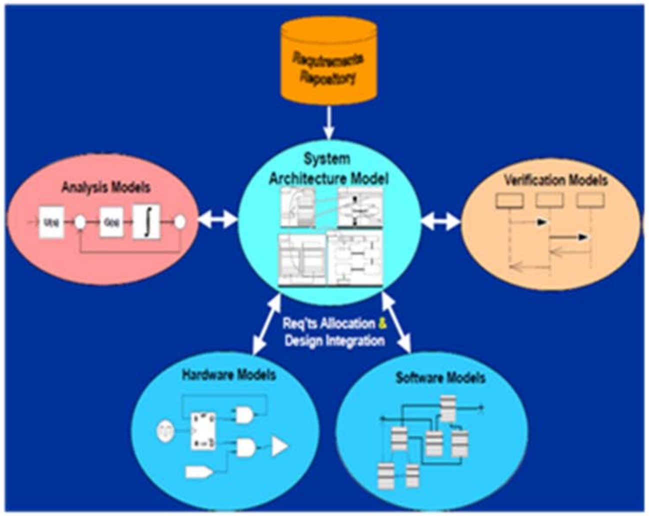
A Rigorous Framework for Model-based Systems Engineering

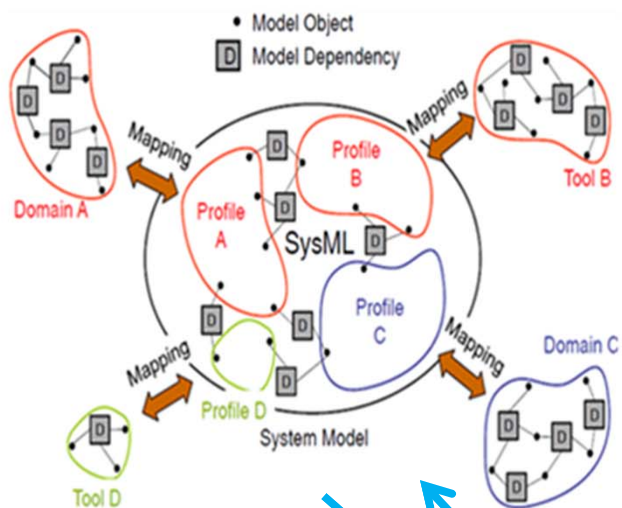
The Challenge & Need:

Develop scalable holistic methods, models and tools for enterprise level system engineering

Multi-domain Model Integration via System Architecture Model (SysML)

System Modeling Transformations





"Master System Model"

Update System Model

ILOG SOLVER, CPLEX, CONSOL-OPTCAD

Tradeoff parameters

ADD & INTEGRATE

- **Multiple domain modeling tools**
- **Tradeoff Tools (MCO & CP)**
- **Validation / Verification Tools**
- **Databases and Libraries of annotated component models from all disciplines**

DB of system components and models

BENEFITS

- **Broader Exploration of the design space**
- **Modularity, re-use**
- **Increased flexibility, adaptability, agility**
- **Engineering tools allowing conceptual design, leading to full product models and easy modifications**
- **Automated validation/verification**

APPLICATIONS

- **Avionics**
- **Automotive**
- **Robotics**
- **Smart Buildings**
- **Power Grid**
- **Health care**
- **Telecomm and WSN**
- **Smart PDAs**
- **Smart Manufacturing**

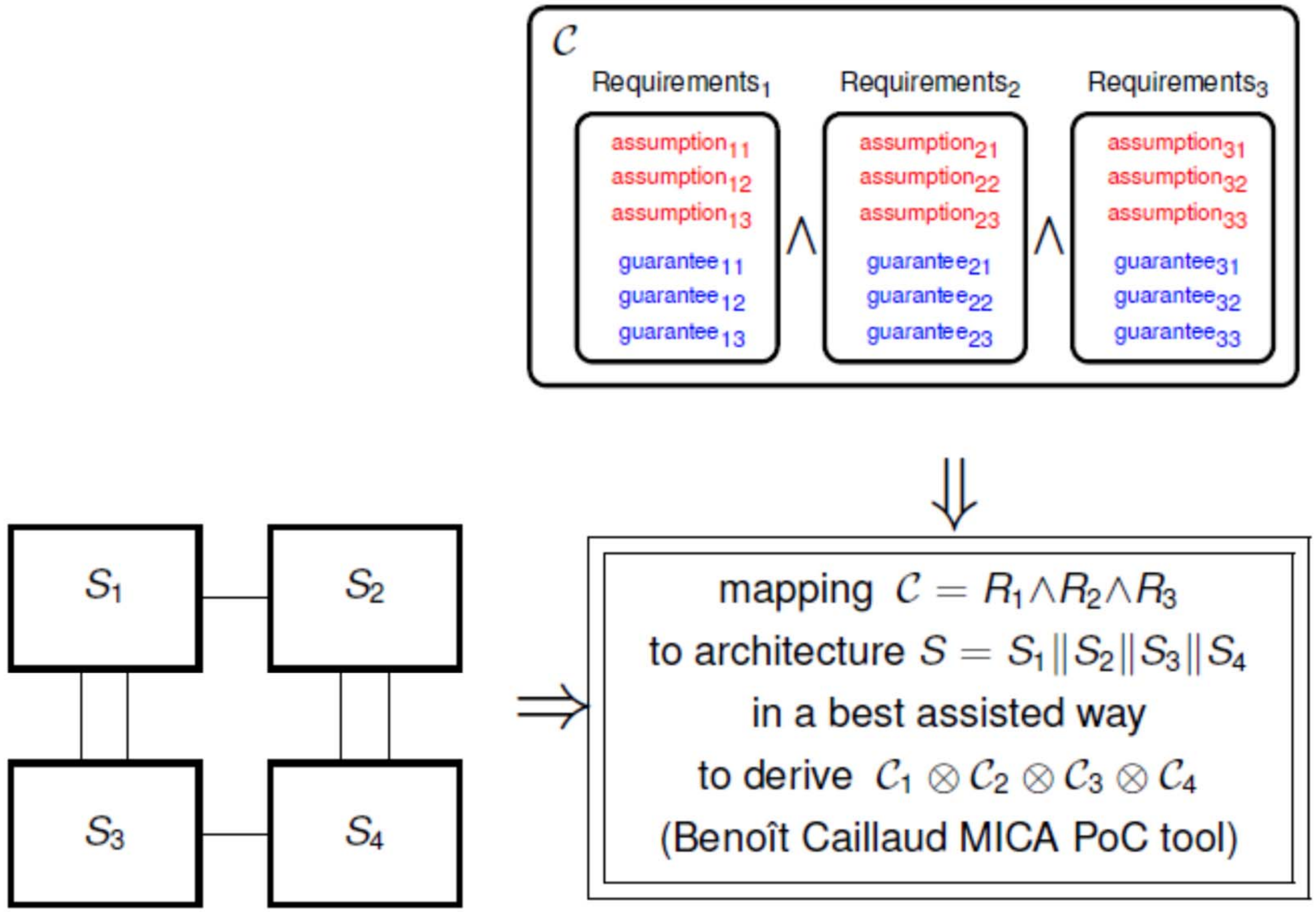
Adaptable Formal Verification

- Traditional formal methods
  - Formulate specification, system
  - Prove that system satisfies spec
    - Model checking: proof search is automatic
    - Theorem proving: proof search requires human assistance
  - Developed for discrete systems
- For **compositionality**: contract-based specifications
  - Spec includes assumption A, guarantee G
  - Idea: system sastisfies A/G if, whenever environment satisfies A, environment composed with system satistisfies G
- **Our focus**
  - Hybrid systems?
  - Evolving environments?
  - Systems that learn?

Contracts for Hybrid Systems

- How to specify A, G?
- Idea: use hybrid automata
  - A: hybrid automaton describing "plant"
  - G: hybrid automaton describing "desired" composite behavior
  - Composition operator(s) derived from e.g. hybrid process algebra (parallel composition, superposition, etc.)
- Theory, algorithms, synthesis approaches need development

Contract-Based Requirements Engineering



system architecture (SysML)

Evolving Contracts

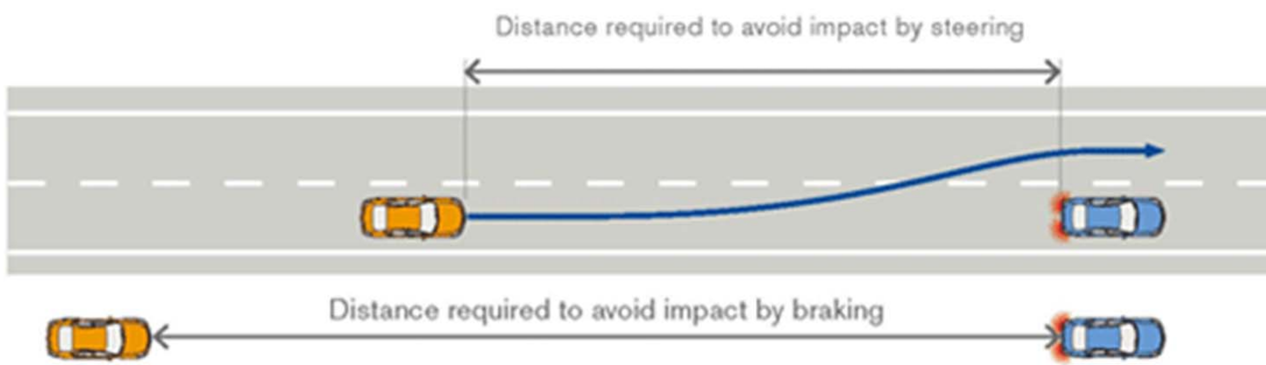
- Suppose system proven correct with respect A/G, and A is different at "run-time"?
  - Must adapt in the moment (e.g. Simplex architecture)
  - Must factor in change to contract
  - But how?
- Contract adaption
  - Theory of contract monitoring to detect deviations
  - Adaptation of A, G based on proofs of correctness
  - Use of on-the-fly model-checking techniques to compute, adapt proofs
- Contract synthesis
  - Use ideas from synthesis of temporal-logic specs from run-time data
  - Combine observations of environment, system to mine contracts from systems

Autonomy V&V: Spatial and Temporal Tolerances

- Reachable set based safety verification and control synthesis
  - Reachable set based verification
  - Control synthesis using optimization
- Motion planning for temporal logics with finite time intervals
  - Mixed integer optimization based method
  - Timed automata based method

Reachable Set Based Verification

- Verification of the safety of the motion planner and the trajectory tracking controls for UAV[1] and autonomous car.
  - Reference trajectories
  - Trajectory tracking controls
  - Q: How to prove safety of the system given sensor noise, control disturbance and dynamics of the system?



1. J. Moschler, Y. Zhou J. S. Baras and J. Joh, "A System Engineering Approach to Collaborative Coordination of UAS in the NAS with Safety Guarantees", Proceedings of 2013 Integrated Communications Navigation and Surveillance (ICNS) Conference, Herndon, Virginia, pp. U1-1~U1-12, April 8~10, 2014.

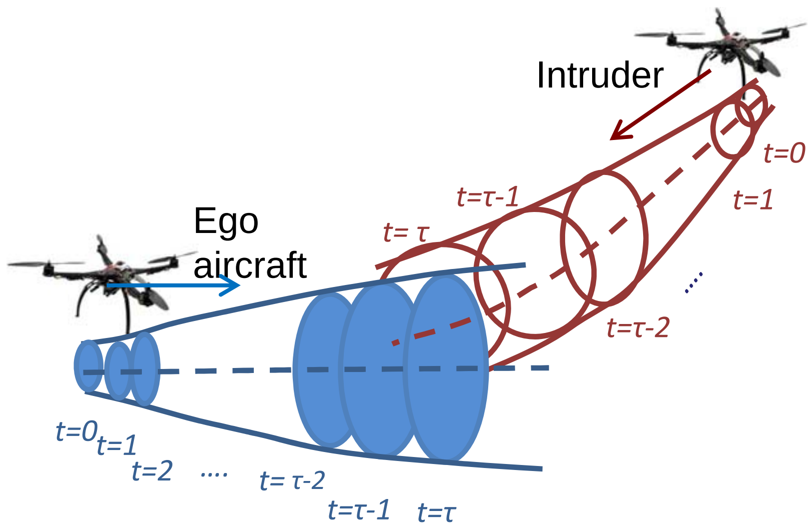
Quadrotors

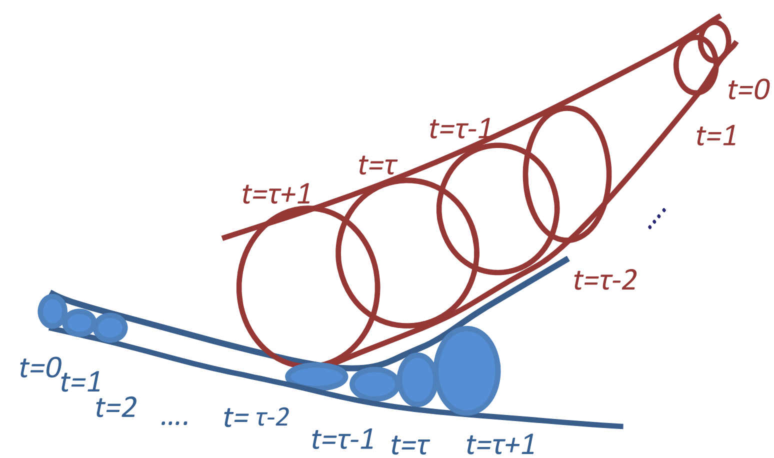
- Control synthesis of safe reachable tubes for collision avoidance using convex optimization
  - Proposed a method to convert the collision avoidance of reachable tubes to convex optimization problems
  - Analyzed for collaborative and non-collaborative settings
  - Resulting control tube can be constant over time[2] or time varying[3]
  - Demonstrated on high dimensional quadrotor and fixed-wing dynamic model

2. Y. Zhou and J. S. Baras, "Reachable Set Approach to Collision Avoidance for UAVs", Proceedings 54th IEEE Conference on Decision and Control, Osaka, Japan, pp. 5947-5952, December 15-18, 2015.
3. Y. Zhou, A. Raghavan and J. S. Baras, "Time Varying Control Set Design for UAV Collision Avoidance Using Reachable Tubes", Proceedings 55th IEEE Conference on Decision and Control, Las Vegas, USA, pp. 6857-6862, December 12-14, 2016.

Collision Avoidance of Two UAVs with Time Varying Control Tubes

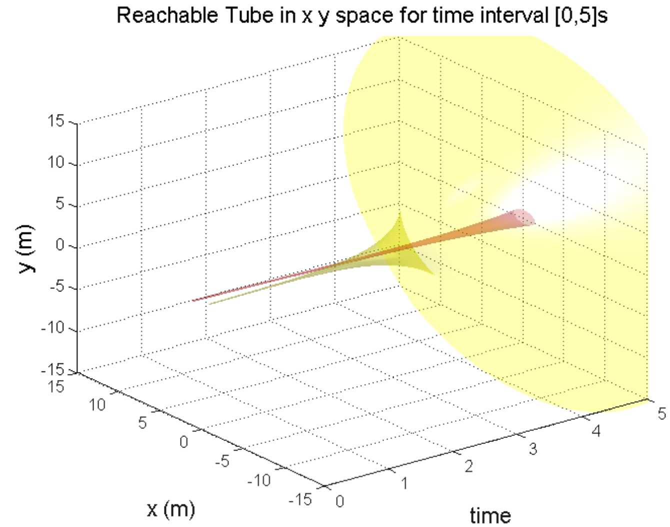
- We seek a control set update rule design for ego aircraft in a non-collaborative setting
  - Guarantee collision avoidance with reachable tube of the intruder aircraft
  - The control constraint set should be time varying
  - Collision avoidance at every time instance
- Seek a tighter control constraint set such that
  - Collision free from predicted reachable set of intruder at all times
  - The control set should be as large as possible.
  - Variation in the control set should be small

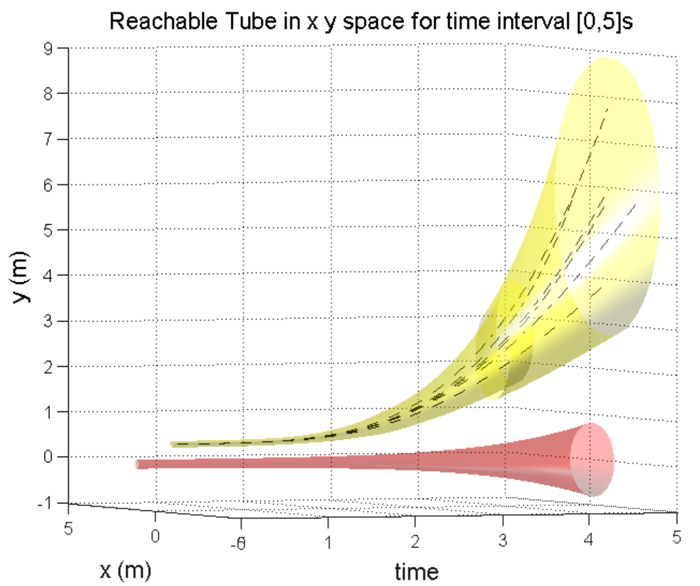




Simulation for Quadrotor

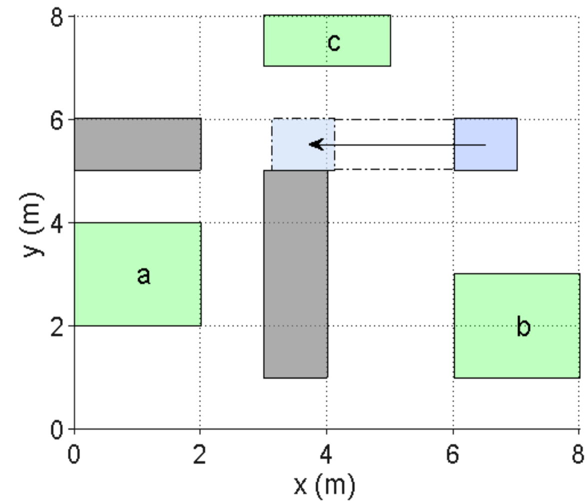
- Similar setup as the quadrotor one earlier, both UAV heading towards each other
- Top plot shows the initial reachable tubes. Red tube is for the intruder vehicle, while the yellow one is for the ego vehicle.
- Bottom plot shows the resulting reachable tube. The exemplary trajectories of full nonlinear dynamics are included as dashed lines.





Motion Planning for Temporal Logics with Finite Time Constraints

- **Problem**: How to generate trajectory/path based on temporal specifications such as ordering, repetition, safety?
- **State of the art**: motion planning with temporal constraints without duration, such as Linear Temporal Logic (LTL).
- Two methods for timed temporal logics, such as Metric Temporal Logic(MTL):
  - An optimization based method[8]
  - A timed-automata based method[9]



Task: Always visiting area a,b,c and stay there for at least 2s. Always avoiding obstacles

8. Y. Zhou, D. Maity and J. S. Baras, "Optimal Mission Planner with Timed Temporal Logic Constraints", Proceedings of 2015 European Control Conference, Linz, Austria, pp. 759-764, July 15-17,2015.
9. Y. Zhou, D. Maity, and J. S Baras. "Timed automata approach for motion planning using metric interval temporal logic.", accepted to 2016 European Control Conference, Aalborg Denmark, June 29 - July 1, 2016.

Robotic Motion Planning Problem

Given:

A dynamic workspace (environment),

A **time constrained task (φ)**,

A cost function.

Objective:

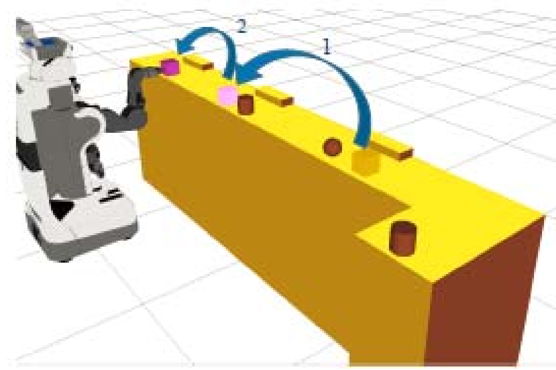
Find the suitable control input such that the robot completes the **given task** and **minimizes** the cost function.

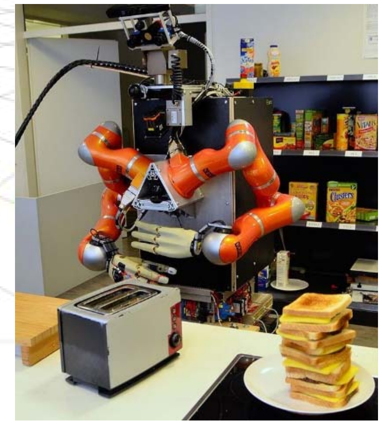
Constraints:

Avoiding collisions with all **static and moving obstacles** in the workspace.

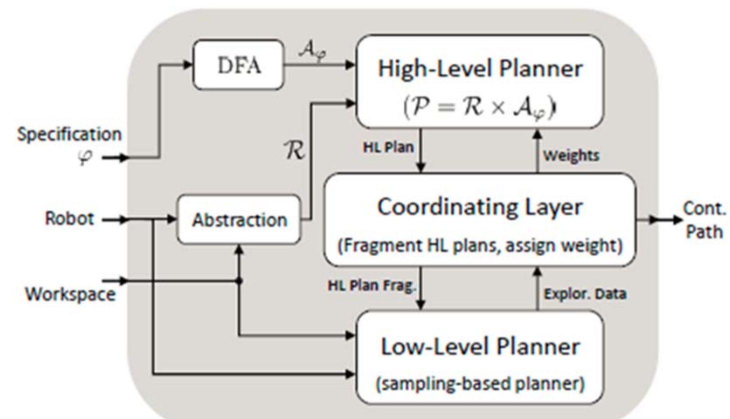
A Robotic Motion Planning Example

- Manipulation task planning[2]
  - First, take food to customers and bring the empty plates back to the preparation area. Next, show the tip jar to the ones whom have already finished eating.
- The question is how fast to take the food to the customers, or what is a good time to ask for the tips from the customers. So timing aspects are important.
- Many robotic tasks require finite time constraints.
- LTL is unable to address finite time constraints and hence we need MITL.





Towards manipulation planning with temporal logic specifications[2]



2. K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, “Towards manipulation planning with temporal logic specifications,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015,

Metric Temporal Logic (MTL) and Time Constrained Task

Definition: The syntax of MTL[12] (MITL[13]) formulas are defined according to the following grammar rules:

$$\phi ::= \top \mid \pi \mid \neg\phi \mid \phi \vee \phi \mid \phi U_I \phi$$

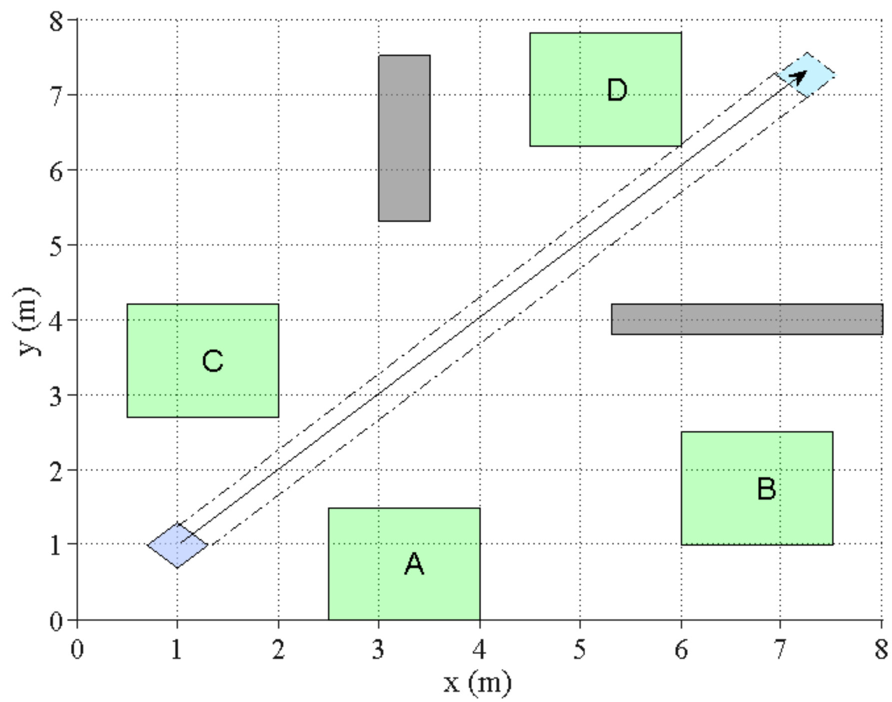
where $I \subseteq [0, \infty]$ is an interval with end points in $\mathbb{N} \cup \{\infty\}$and the end points have to be distinct . $\pi \in \Pi$ is the atomic proposition.

More sophisticated MTL (MITL) operators can be derived using the grammar defined above; such as: always in $I_1 \equiv \perp U_{I_1}$, eventually always $\Diamond_{I_1} \Box_{I_2}$ etc.

12. R. Koymans, "Specifying real-time properties with metric temporal logic," *Real-time systems*, vol. 2, no. 4, pp. 255–299, 1990.
13. R. Alur, T. Feder, and T. A. Henzinger, "The benefits of relaxing punctuality," J. of the ACM (JACM), vol. 43, no. 1, pp. 116–146, 1996.

Examples

- **Task 2**:
  - The specification requires the autonomous vehicle to eventually visit area A, B, C and D, and stay there for at least 2 time units, while avoiding obstacles.



A much crowded workspace. The grey boxes represent fixed obstacles, while the blue one is a moving obstacle with fixed speed.

Optimization Based Method

$$
\begin{aligned}
\min_{u} \quad & J(x(t,u),u(t)) \\
\text{Subject to} \quad & x(t+1) = f(t,x(t),u(t)) \\
& \mathbf{x}_{t_0} \models \varphi
\end{aligned}
$$

Remarks:

The task φ may be a finite duration task within an infinite time horizon task such as surveillance, periodic tasks etc.

Modification of Original Problem into MILP

$$\min_{u, z_0, \ldots, zN \in \{0,1\}^p} \quad J(x(t,u), u(t))$$

$$\text{Subject to} \quad x(t+1) = f(t, x(t), u(t))$$

$$L(x(t), z_t, t) \leq 0 \quad \forall t \in [0, N]$$

The timed temporal constraint $\mathbf{x_{t_0}} \vDash \boldsymbol{\varphi}$ can been converted into the linear and integer constraints.

Remark:

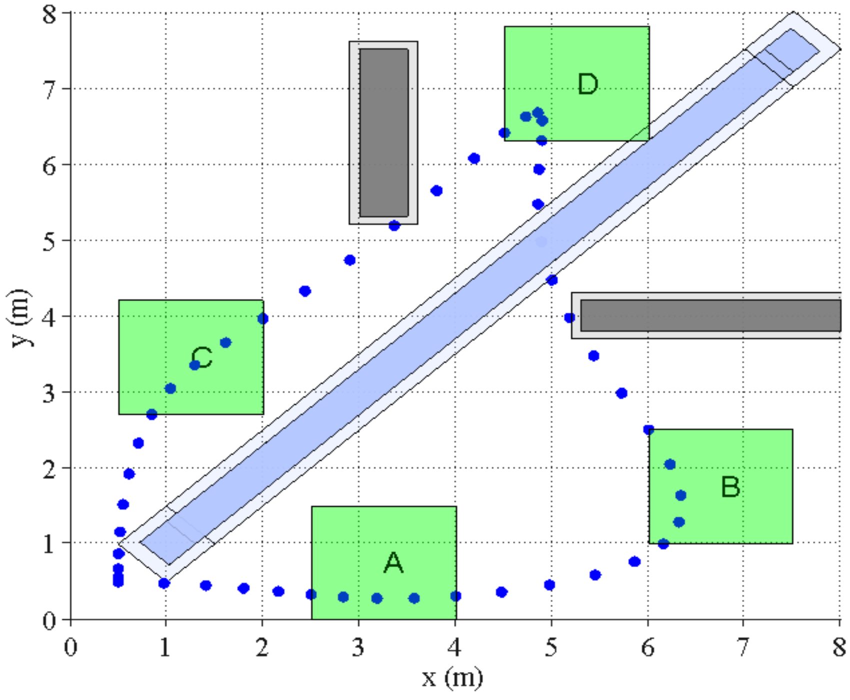
If $J(\cdot,\cdot)$ $f(.,..,.)$ are linear functions of $x(t)$ and $u(t)$, then entire problem will be a Mixed-Integer Linear Optimization Problem.

Results

- Specification in MTL

$$\phi_3 = \Diamond\Box_{[0,2]}A \wedge \Diamond\Box_{[0,2]}B \wedge \Diamond\Box_{[0,2]}C \wedge \Diamond\Box_{[0,2]}D \wedge \Box\neg O$$

- The result for linearized quadrotor dynamic projected in 2D is shown in right as blue dots



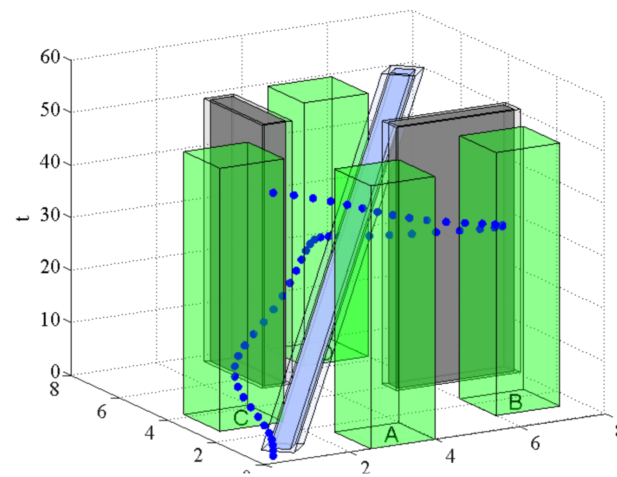
2D projection of the trajectory of the quadrotor satisfying the task.

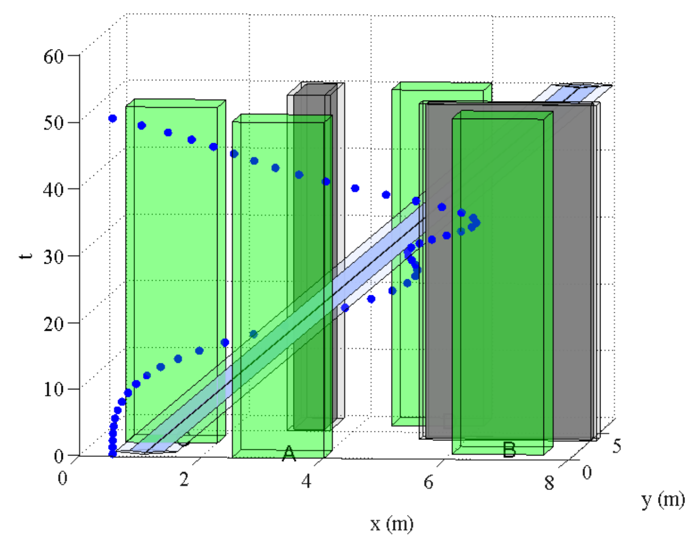
Results

- Specification in MTL

$$\phi_3 = \Diamond\Box_{[0,2]}A \wedge \Diamond\Box_{[0,2]}B \\ \wedge \Diamond\Box_{[0,2]}C \wedge \Diamond\Box_{[0,2]}D \wedge \Box\neg O$$

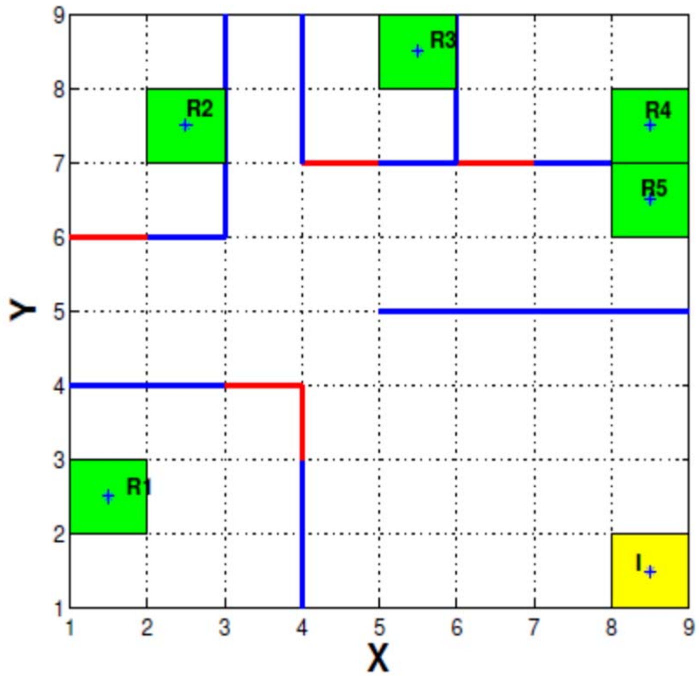
- 3D Trajectory
  - The trajectory avoids the obstacle region in time and space.





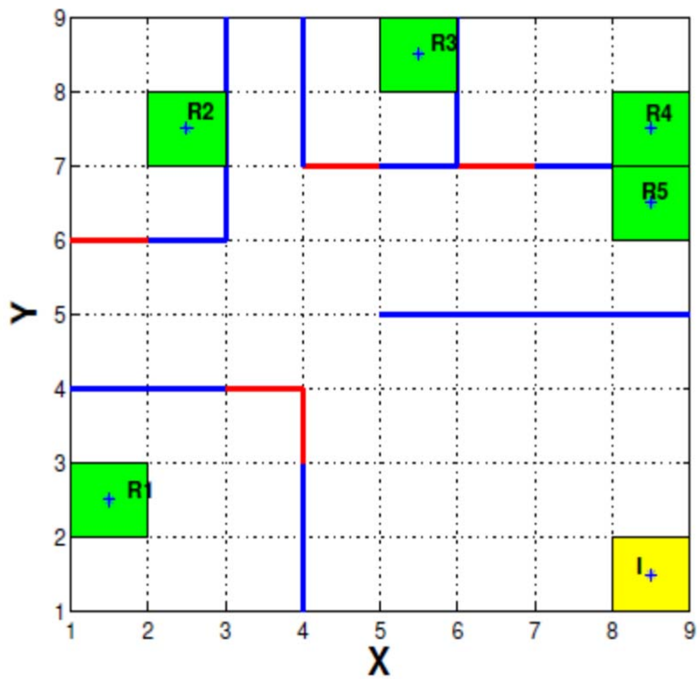
Robot Motion Planning Problem

Example: Starting from I, visit R_3 within the time interval T_1, visit R_4 within time interval T_2; before visiting R_3 or R_4, robot must visit R_2. Eventually visit R_1 and R_5, and complete the whole task in the **least time**.

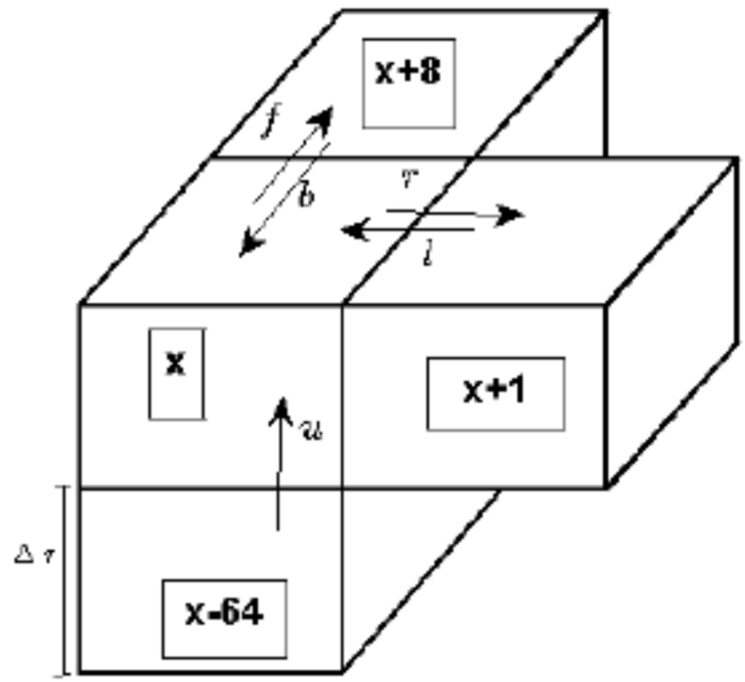


The workspace of the robot.

Workspace as State Transition System



The workspace of the robot.



Transitional relationship among the blocks in discretized workspace-time.

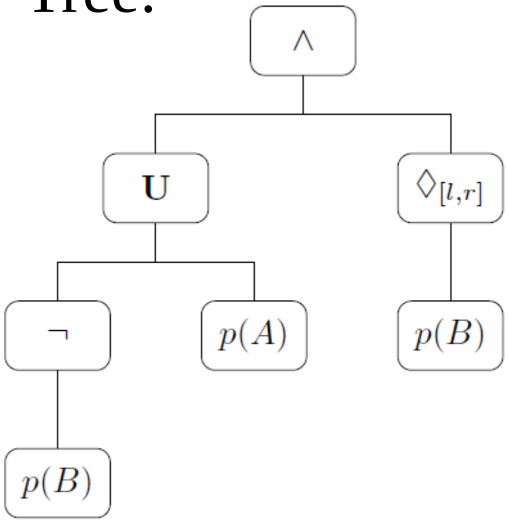
Timed Automata Based Planning Example

- Convert temporal logic formula to a timed automaton
  - Represent temporal logics as a tree structure
  - Every operator in the tree can be represented as a timed automaton with input and output
  - The product of them results into a timed automaton again

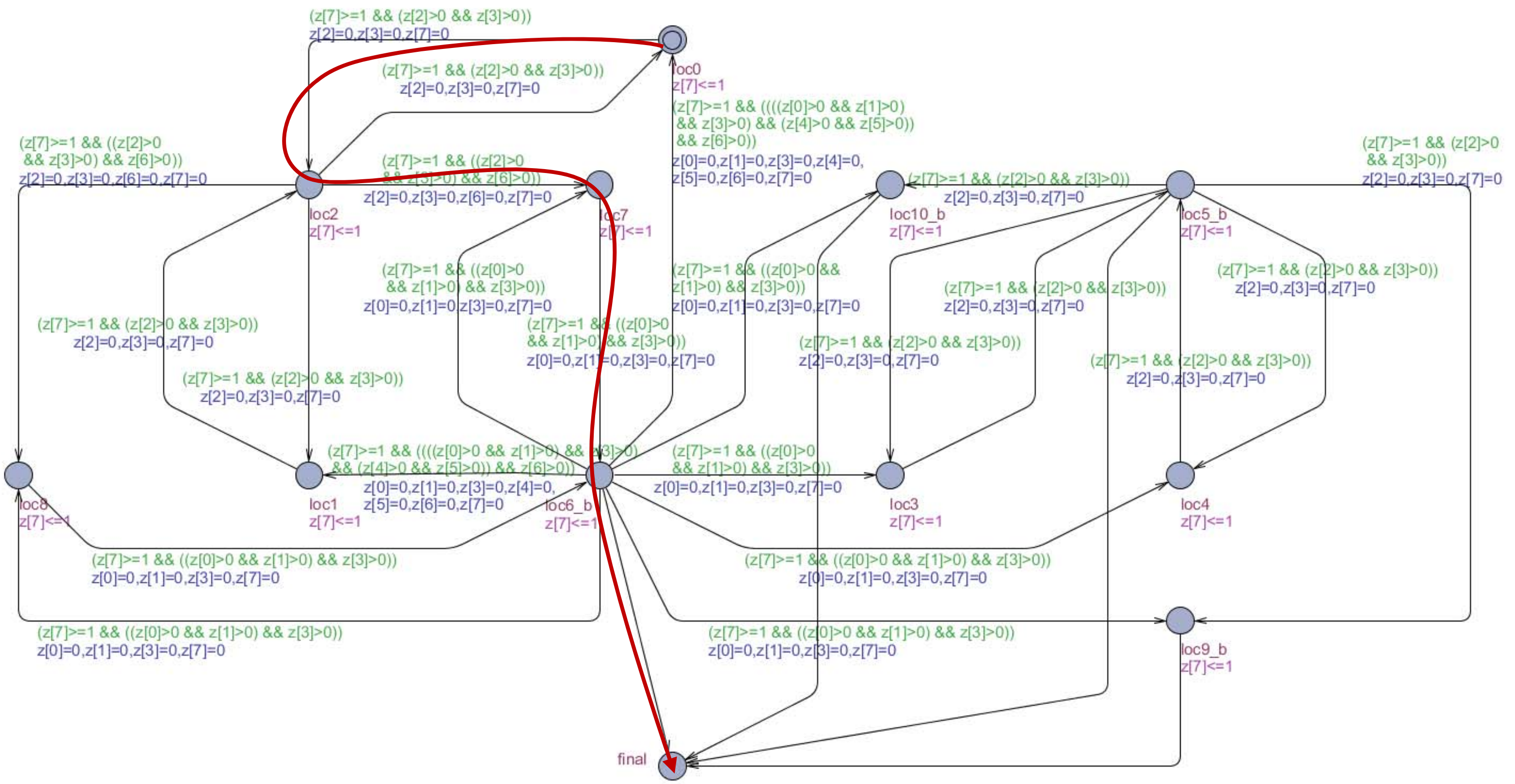
- Specs:
  Visit A before B and visit B within [l, r]
- MTL:

$$\phi = (\neg B \mathbf{U} A) \wedge (\Diamond_{[l,r]} B)$$

- Tree:



Shortest Path – Resulting Path



- Generated timed automata and the fastest path using UPPAAL

Robot Dynamics and Control

We consider a non-holonomic unicycle robot dynamics as given below:

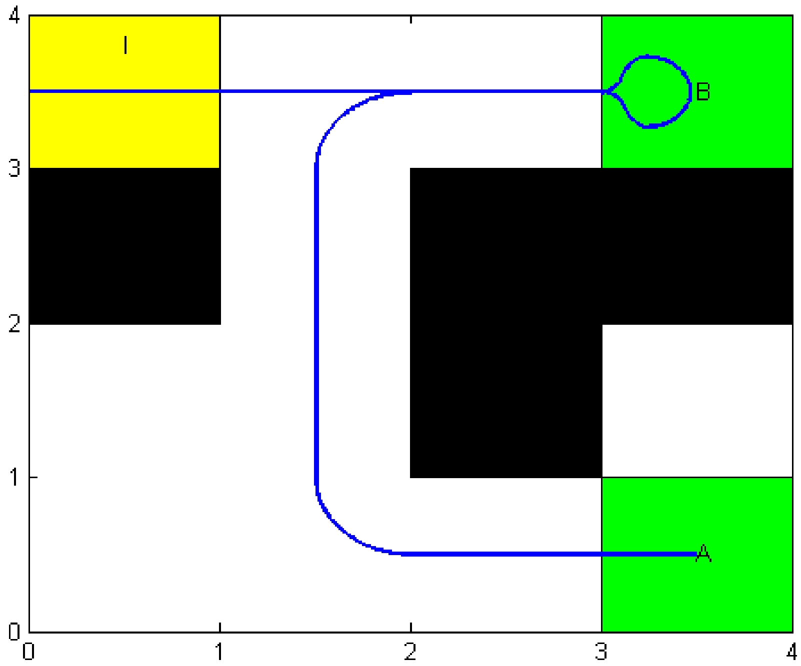
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = u \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} + w \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

u and w are the control inputs.

Since we are dealing with time-bounded motion planning, we represent state and time pair by (q, t) where $q = [x, y, w]$.

Simulations

- Task: $\phi = (\neg A \mathbf{U} B) \wedge (\lozenge B)$



The robot starts from the initial position I (yellow block) and according to the task, it visits B before visiting A.

Complexity Analysis

Table: Computation Time for Typical MITL Formula

MITL Formula	Map Grid size	Transformation Time (in s)	Number of Transitions	Synthesis Time (in s)
ϕ_1	2x2	< 0.001	22	0.016
ϕ_2	2x2	0.004	69	0.018
ϕ_3	2x2	0.40	532	0.10
ϕ_4	2x2	0.46	681	0.12
ϕ_1	4x4	0.004	181	0.062
ϕ_1	8x8	0.015	886	0.21
ϕ_2	8x8	0.015	1795	0.32

$\phi_1 = (\neg A \mathbf{U} B) \wedge (\Diamond A)$

$\phi_2 = \Box \Diamond_{[0,2]} A$

$\phi_3 = \Diamond_{[0,4]} A \wedge \Diamond_{[0,4]} B$

$\phi_4 = \Diamond_{[2,4]} A \wedge \Diamond_{[0,2]} B$

New Approach

- Existing controllers are feedback in nature { generally requires expensive communication & sensing resources.
- Abstraction based approaches generally suffer from state explosion -- computationally expensive.
- Inability to incorporate time constraints in many existing approaches.

We use **Signal Temporal Logic** and derive event-triggered control strategies.
Key features: state- and time-constrained tasks, robust, computationally-ecient (abstraction-free), inexpensive implementation (event-trigger communication & control)

Signal Temporal Logic (STL)

- **Predicates**[8] μ are obtained after evaluation of a predicate function $h : \mathbb{R}^n \to \mathbb{R}$ as

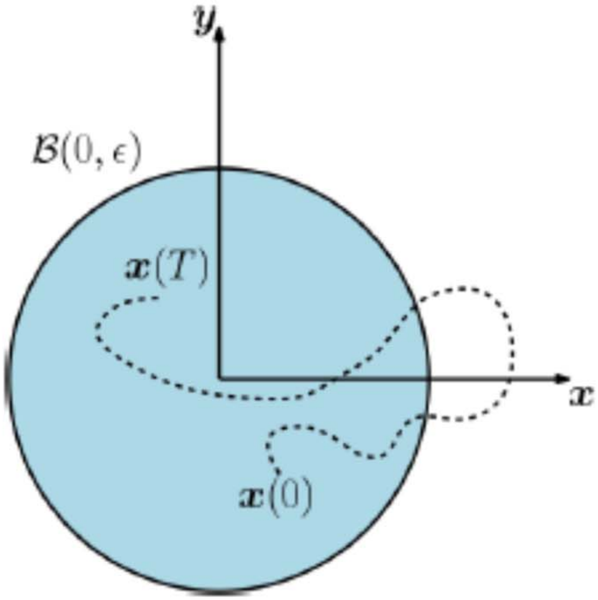
$$\mu := \begin{cases} \top \text{ if } h(\mathsf{x}) \geq 0 \\ \perp \text{ if } h(\mathsf{x}) < 0 \end{cases}$$

Example:

- Assume the predicate $\|\mathsf{x}(\tau)\| \leq \epsilon$.
- The predicate function

$$h(\mathsf{x}(\tau)) := \epsilon - \|\mathsf{x}(\tau)\|$$

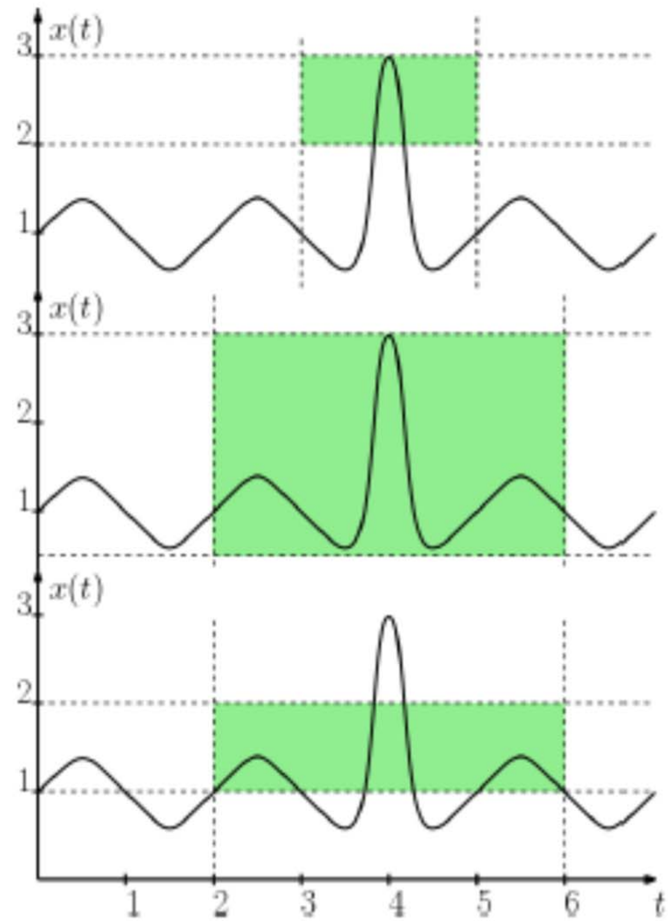
indicates $\|\mathsf{x}(\tau)\| \leq \epsilon$ iff $h(\mathsf{x}(\tau)) \geq 0$.



[8] O. Maler and D. Nickovic. *Int. Conference on Formal Modeling and Analysis of Timed Systems*. 2004.

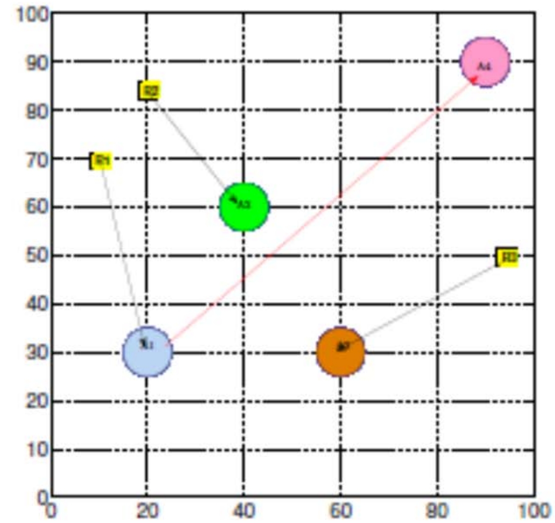
Signal Temporal Logic (STL)

- $\phi_1 = F_{[3,5]}(2 \leq x \leq 3)$ **holds** and $\boxed{\rho^{\phi_1}(x,0) = 0.5}$
- $\phi_2 = G_{[2,6]}(0 \leq x \leq 6)$ **holds** and $\boxed{\rho^{\phi_2}(x,0) = 0.01}$
- $\phi_3 = G_{[2,6]}(1 \leq x \leq 2)$ **does not hold** and $\boxed{\rho^{\phi_3}(x,0) = -1}$



Experimental Result

- **R1**: 1) Eventually Go to **A1** and $\theta_1 = 45^o$
  2) go to **A4** and stay close to **R2**
- **R2**: 1) Eventually Go to **A2** and $\theta_1 = 45^o$
  2) Stay close to **R1** and **R3**
- **R3**: 1) Eventually Go to **A3** and $\theta_1 = 45^o$
  2) Stay close to **R3**



- **Tasks**:

$$\begin{aligned}
\phi_1 &:= F_{[0,50]}((\|\mathbf{p}_1 - \mathbf{A1}\| \leq \epsilon) \wedge \|\theta_1 - 45^o\| \leq \epsilon)) \\
&\quad \wedge F_{[50,100]}((\|\mathbf{p}_1 - \mathbf{A4}\| \leq \epsilon) \wedge \|\mathbf{p}_1 - \mathbf{p}_2\| \leq \epsilon) \\
\phi_2 &:= F_{[0,50]}((\|\mathbf{p}_2 - \mathbf{A2}\| \leq \epsilon) \wedge (\|\theta_2 - 45^o\| \leq \epsilon)) \\
&\quad \wedge F_{[50,100]}(\|\mathbf{p}_2 - \mathbf{p}_3\| \leq \epsilon) \wedge \|\mathbf{p}_2 - \mathbf{p}_3\| \leq \epsilon \\
\phi_3 &:= F_{[0,50]}(\|\mathbf{p}_3 - \mathbf{A3}\| \leq \epsilon) \wedge (\|\theta_3 - 45^o\| \leq \epsilon)) \\
&\quad \wedge F_{[50,100]}(\|\mathbf{p}_3 - \mathbf{p}_2\| \leq \epsilon)
\end{aligned}$$

Agent Trajectories

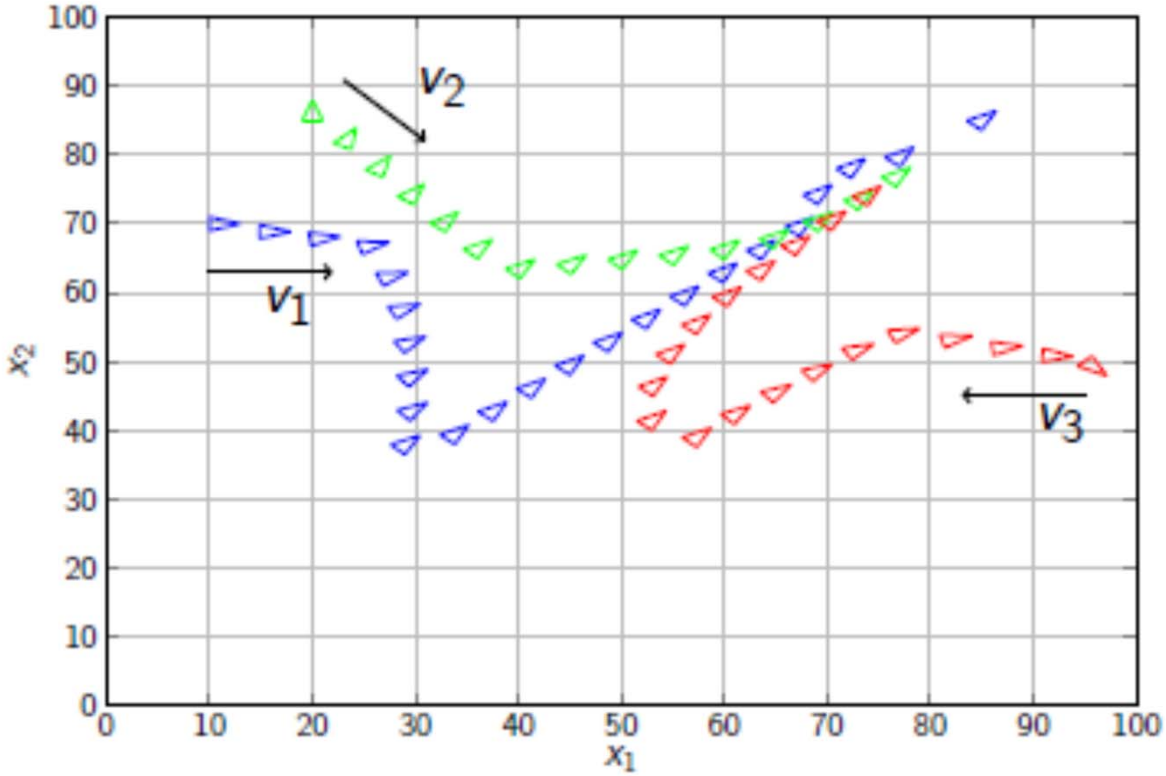


Figure : Agent Trajectories

Robustness

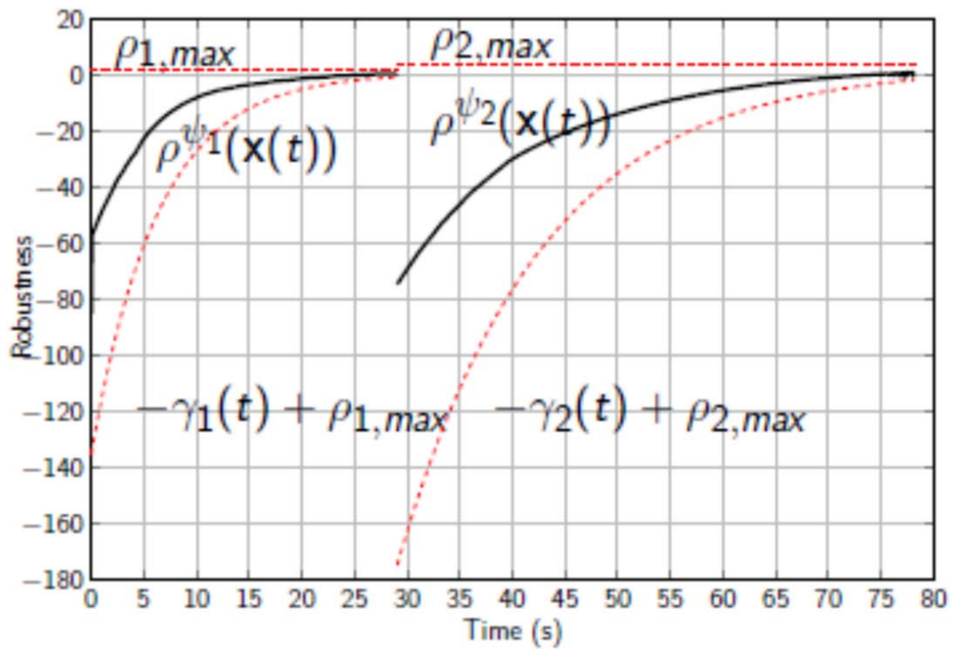


Figure : Robustness

- The experiment was implemented in 100 Hz frequency with a total of **7725** samples.
- Using our event-triggered feedback policy the control was updated only **185** times.

Summary and Future Work

Summary:

- Abstraction-free, computationally-efficient, and robust method for bottom-up multi-agent systems
  - Robustness considered on two levels: robustness with respect to the task and with respect to disturbances
- Event-triggered control reduced the amount of communication significantly.

Thank you!

baras@isr.umd.edu

301-405-6606

http://dev-baras.pantheonsite.io/

Questions?