Trusted Autonomy: Theory and Applications

John S. Baras
Institute for Systems Research
ECE, CS, ME, AE, Bioeng., DOIT, AMSSC
University of Maryland

February 26, 2021

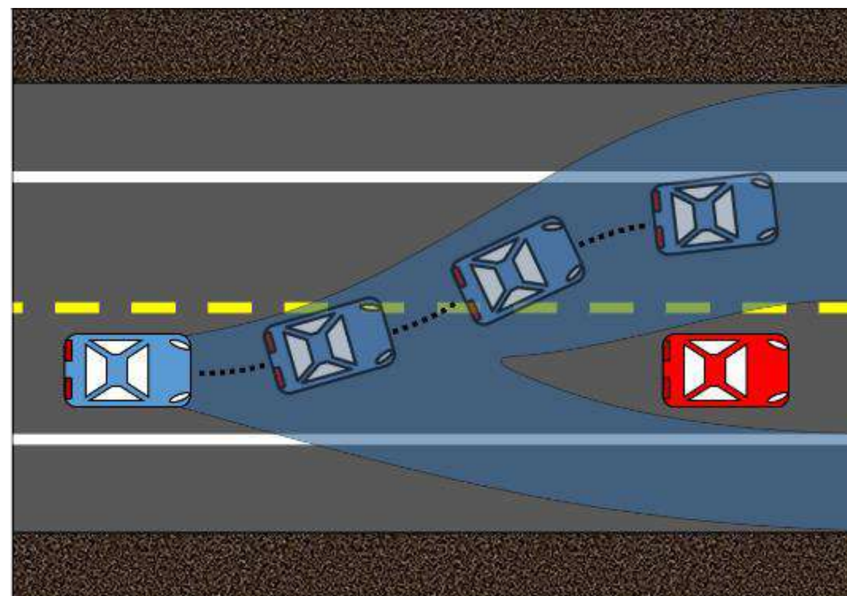
Maryland Robotics Colloquium

Acknowledgments

- **Joint work with:** Xiaobo Tan, Wei Xi, Yuchen Zhou, Dipankar Maity, Zhenyu Lin, Usman Fiaz, Christos Mavridis, Charles Meehan, Nilesh Suriyarachchi, Faizan Tariq, Aneesh Raghavan, Siddharth Bansal, Fatemeh Alimardani, Sandeep Damera, Ren Mao, Wentao Luan, Yiannis Aloimonos, Cornelia Fermuller, Rance Cleaveland
- **Sponsors:** ONR, DARPA, NSF, ARL, ARO, NRL, Lockheed Martin, Northrop Grumman, ABB, STEER, Nokia Bell-Labs, DoD USRD

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 and other specs

Motivation: Human-Robot Collaboration and Safety

Motivation: Learning Tasks, Changing Environments

- Teach through demonstrations
  - Easy training, hard to generalize to new constraints
- Program planning techniques
  - Generalize to constraints, manually design objectives

Motivation: Collaborative Autonomy and Trust








Motivation: Future Networks Automation --- 5G, SDN, NFV, MEC, IoT, NaaS,….

- **Internet and Telecom convergence → Integrated networks: Future Internet**
- **Novel services, applications and communication paradigms**
  - **Internet of Things (IoT) and Smart cities, M2M and Vehicular communications, Content/media oriented communications, Social networks,**
  - **Internet of Everything (IoE), etc.**
- **Novel, emergent technologies** are changing networks and services architectures :
  - ***Supporting technologies***
    - ***Cloud Computing***
    - ***Fog/Edge Computing /Mobile Edge Computing /Cloudlets***
    - ***Software Defined Networks (SDN)***
    - ***Network Function Virtualization (NFV)***
    - ***Advances in wireless technologies: 4G-LTE, LTE-A, WiFi, 5G***

Key Challenges Addressed

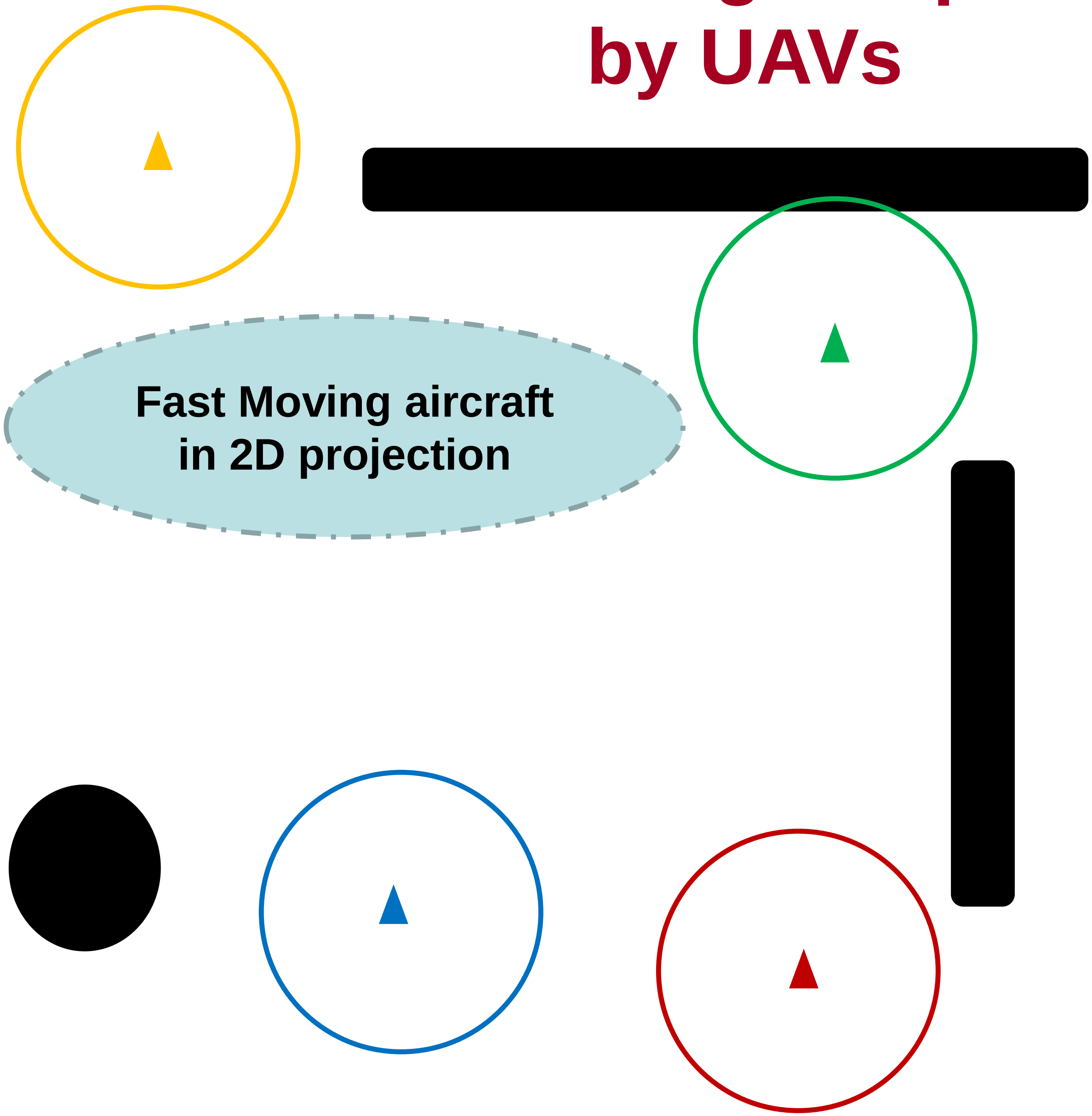
- Safe behaviors – can they be learned?
  - UAVs in commercial airspace
  - Autonomous vehicles & human-driven cars
- Human-machine collaboration – off line and on-line learning – is it safe?
  - Safety is critical and fundamental
- Physical limitations
  - To avoid states that lead to unavoidable faults, collisions, wrong behavior – Prediction? How fast? How accurate?

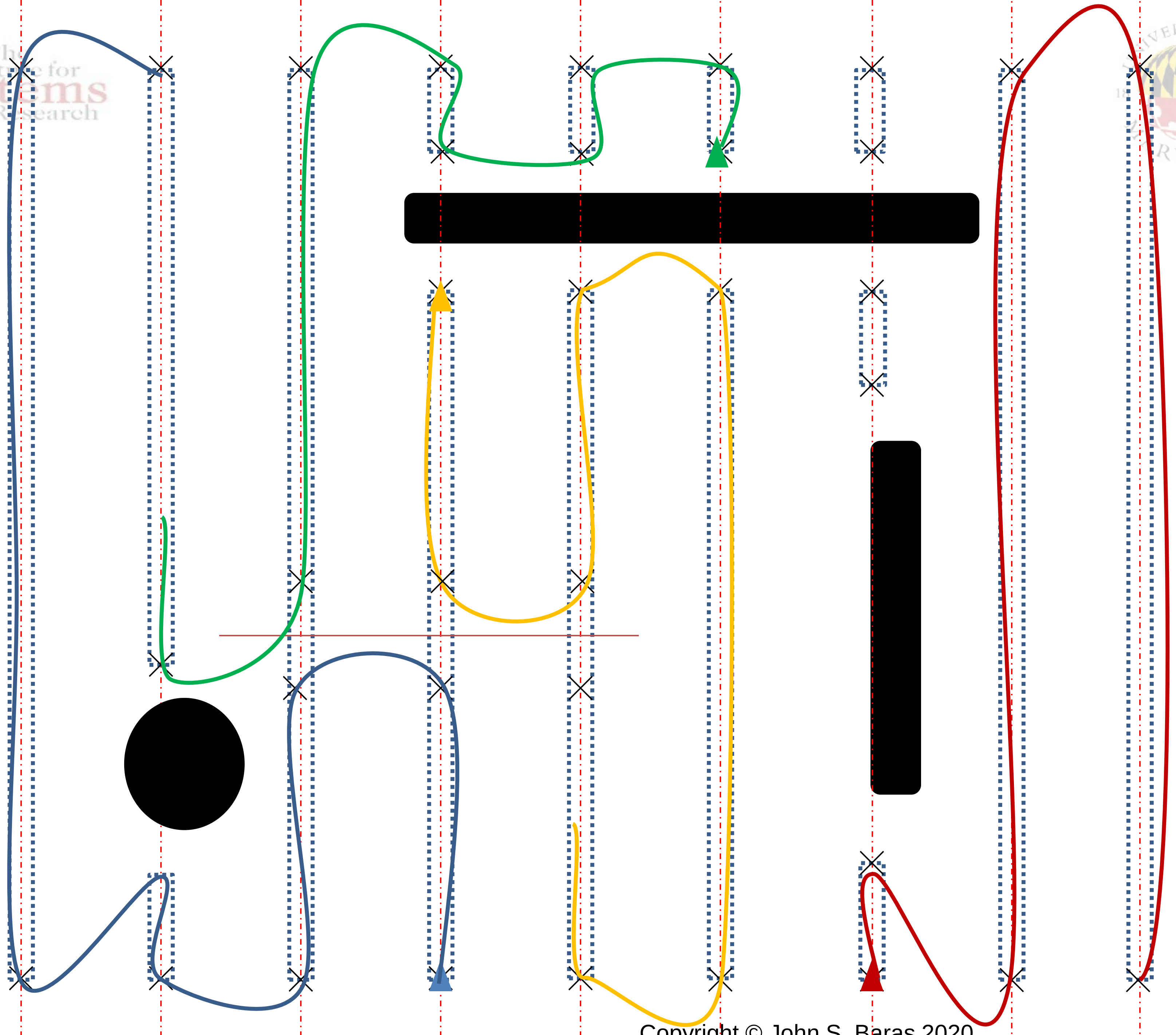
Assured Autonomy : Spatial and Temporal Tolerances

- Artificial potential based method
- Reachable set based verification
- Control synthesis using optimization
- Mixed integer optimization based method
- Timed automata based method

Tan, Xi, Zhou, Maity, Baras, several papers 2015 -2019

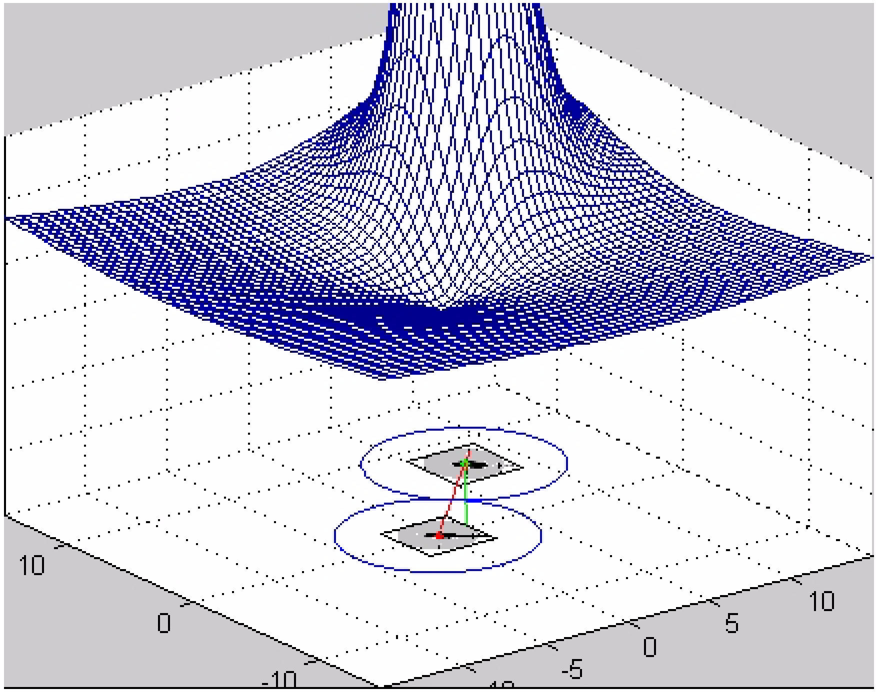
Monitoring Airspace by UAVs





Autonomy via Potential Functions

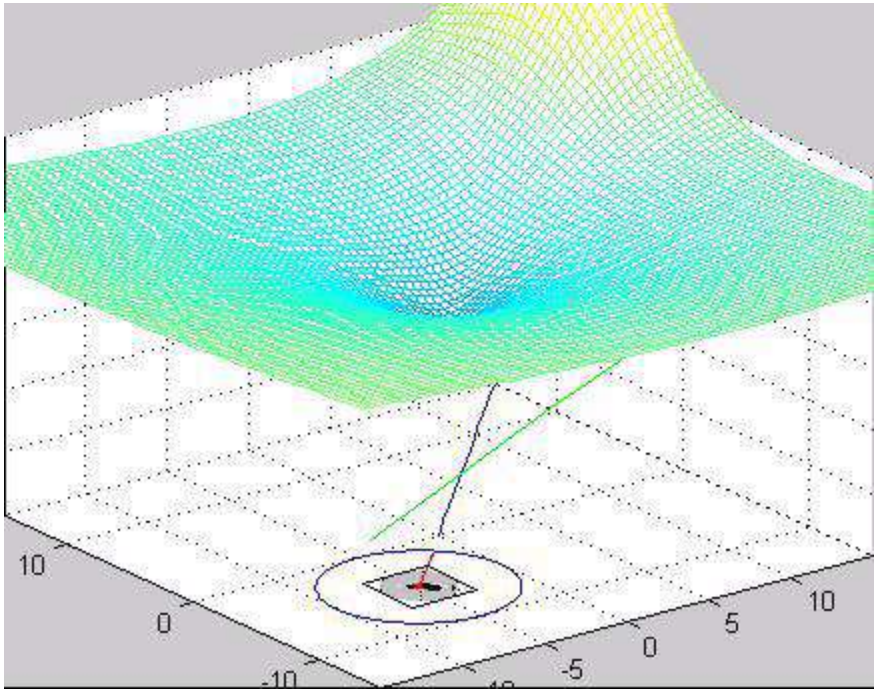
$$J_a(\mathbf{x}_a) = \sum_{i=1}^{N_a} b_i f(\mathbf{x}_a, \mathbf{x}_{ai}, \mathbf{v}_{ri})^{-1} + g(\mathbf{x}_a, \mathbf{x}_m, \mathbf{v}_m)^{-1} + \left\| \mathbf{x}_a - \mathbf{x}_{\gamma a} \right\|_2^2$$



UAS agents avoid one another.

Autonomy via Potential Functions

$$J_{\mathrm{a}}(\mathbf{x}_{\mathrm{a}}) = \sum_{i=1}^{N_{\mathrm{a}}} b_i f(\mathbf{x}_{\mathrm{a}}, \mathbf{x}_{\mathrm{a}i}, \mathbf{v}_{\mathrm{r}i})^{-1} + g(\mathbf{x}_{\mathrm{a}}, \mathbf{x}_m, \mathbf{v}_m)^{-1} + \left\| \mathbf{x}_{\mathrm{a}} - \mathbf{x}_{\gamma\mathrm{a}} \right\|_2^2$$



Gradient pushes UAS out of the way of piloted aircraft.

Multiple Collaborating Vehicles

Mission

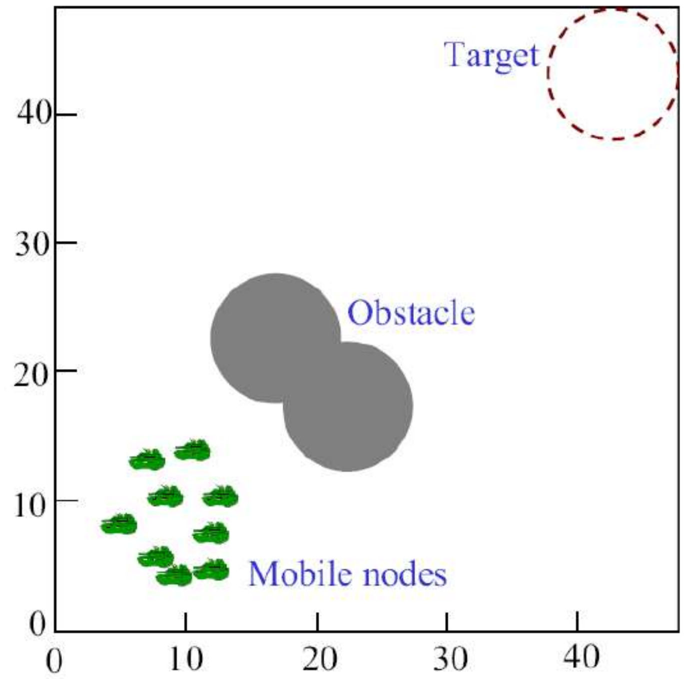
Autonomous, distributed maneuvering of a vehicle group to reach and cover a target area

Constraints

Desired inter-vehicle distance
Obstacles avoidance
Threats (stationary or moving) avoidance

Requirement

Using only local or static information



Artificial Potentials – Gradient-Flow Approach

Dilemma of the Deterministic gradient-flow approach

Potentials-based approach can accommodate multiple objectives and constraints in a distributed and computationally effective way

The system dynamics could be trapped by the local minima

Weighted sum of potential functions:

$$J_{i,t}(q_i) = \lambda_g J^g(q_i) + \lambda_n J_{i,t}{}^n(q_i) + \lambda_o J^o(q_i) + \lambda_s J^s(q_i) + \lambda_m J_t{}^m(q_i)$$

Target (attraction) potential J^g

Neighbor (avoidance) potential J^n

Obstacle potential J^o

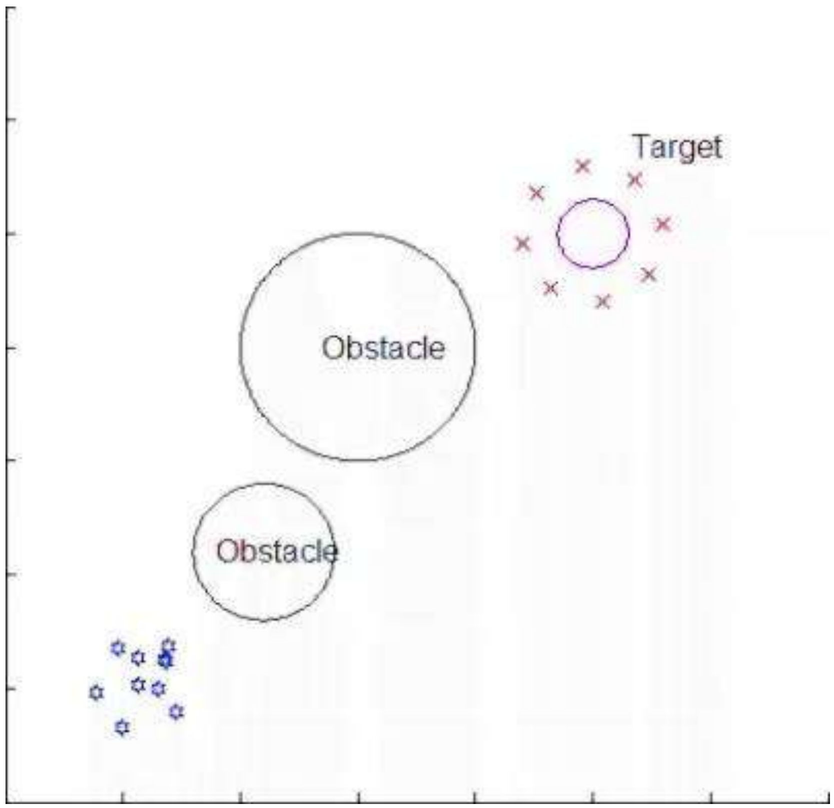
Potential J^s due to stationary threats

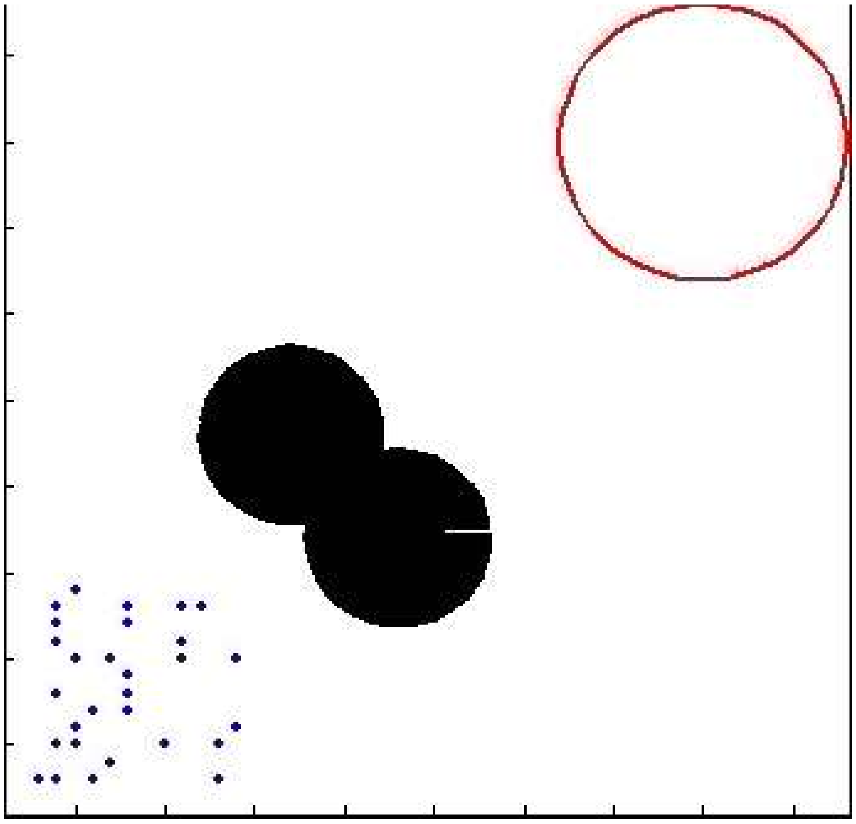
Potential J^m due to moving threats

Gradient flow:

$$\dot{q}_i(t) = -\frac{\partial J_{i,t}(q_i)}{\partial q_i}$$

Being Trapped by Local Minima





Different initial conditions may cause vehicles to be trapped by local minimum

Modeling a Swarm as a GF

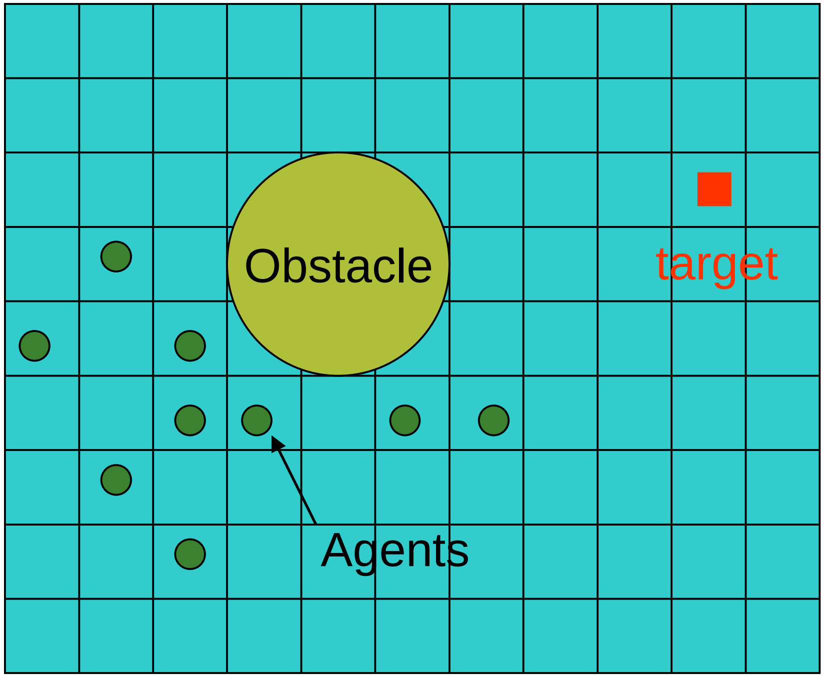
2D mission space on discrete lattice cells

- Agent s can communicate with neighboring agents in N_s which stay within the interaction range R_s
- An agent can go at most R_m within one move, which defines the phase space Λ_s
- Gibbs potential is designed to reflect global objective

$$U(x) = \sum_{c \in C} \Psi_c(x),$$

$$\Phi_s(x) = \sum_{c \in C_s} \Psi_c(x_s, x_{N(s)})$$

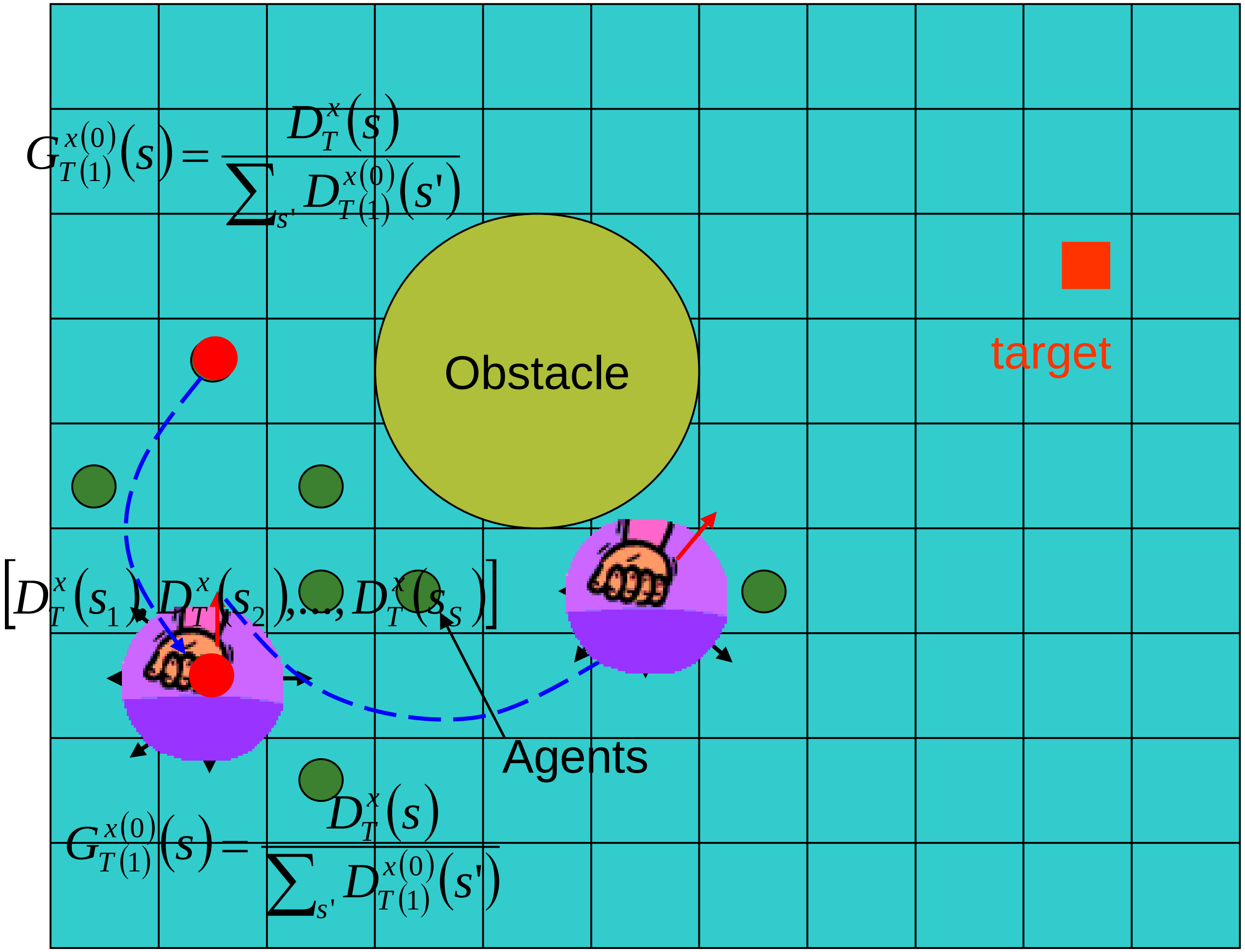
$$= \lambda_g J_s^g + \lambda_o J_s^o + \lambda_n J_s^n$$



Difficulties in applying classical results

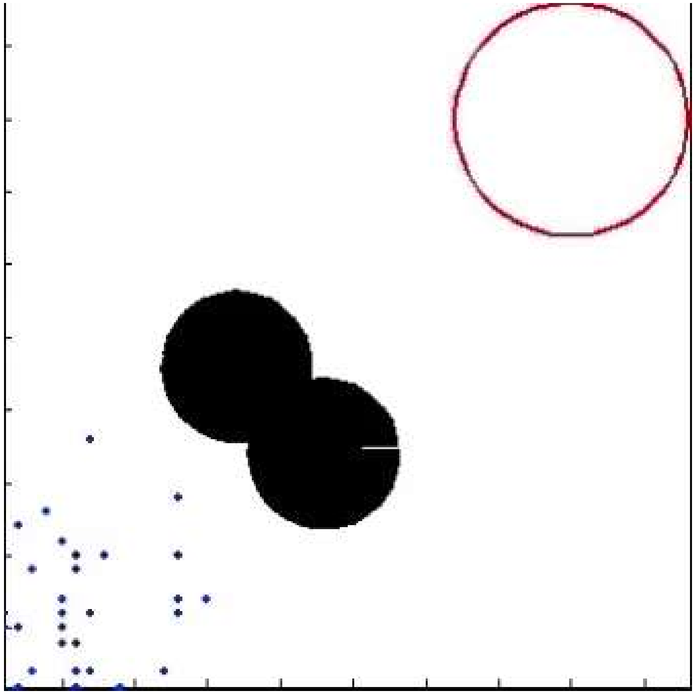
- **Non-stationary neighborhood system**
- **Time-varying and state-dependent phase space**

Animation of Sequential Gibbs Sampling Algorithm



Stochastic Path Planning Simulation

Stochastic path exploration based on MRF can lead multiple vehicles getting around the obstacles

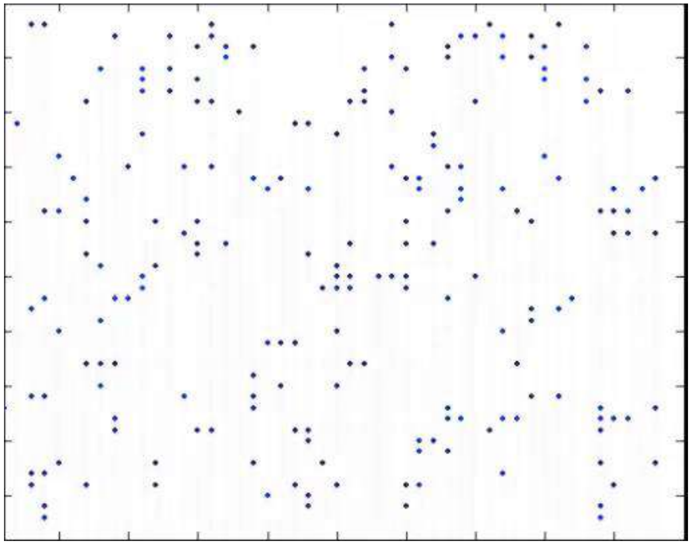


Potential function

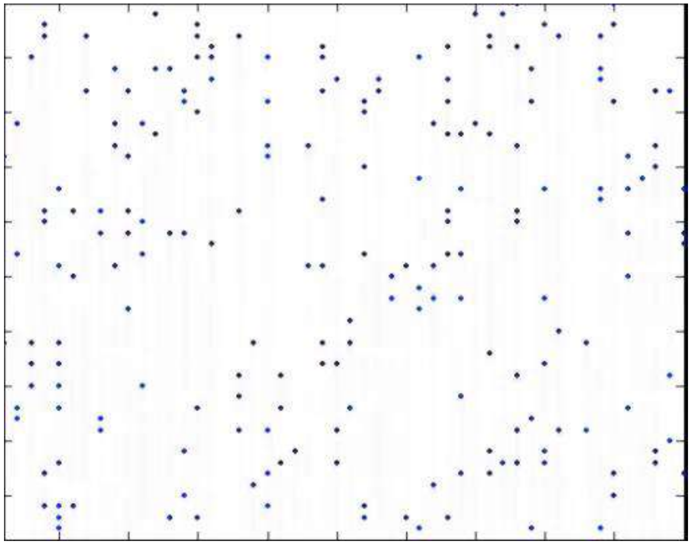
$$\Phi_s(x) = \lambda_g J_s^g + \lambda_o J_s^o + \lambda_n J_s^n$$

- Target (attraction) potential J^g
- Neighbor (avoidance) potential J^n
- Obstacle potential J^o

Simulation: Gathering



specified center $Z_0=(25,25)$

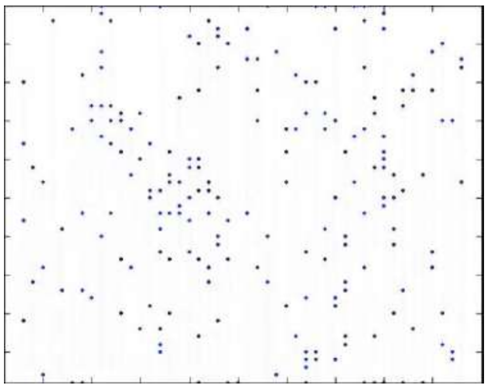


unspecified center

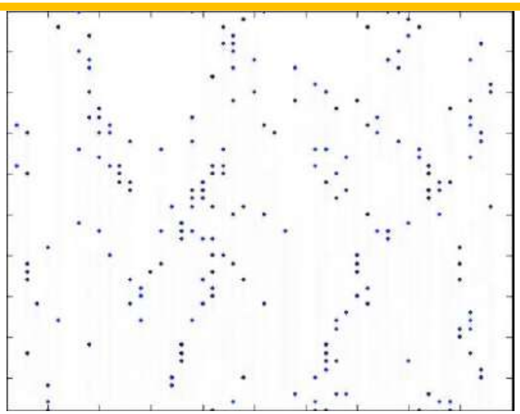
200 nodes on 50 by 50 grid ; $\lambda_1 = 0.05$, $\lambda_2 = 1$, $\Delta = 10^3$

$R_m = 2\sqrt{2}, R_s = 6\sqrt{2}$; $T(n) = 1/(4\log(400+n))$

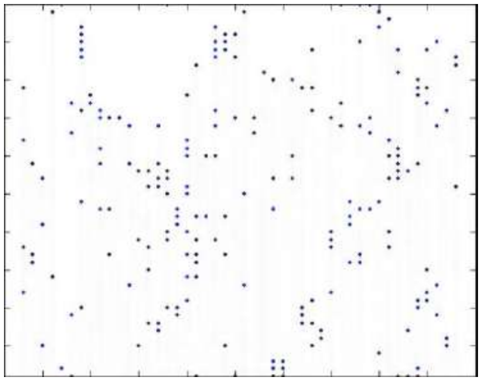
Simulation: Line Formation



One line



Three lines



Two lines

200 nodes on 50 by 50 grid

λ=10 , Δ=5

$R_m = 2\sqrt{2}$

$R_s = 10\sqrt{2},\ 6\sqrt{2},\ 4\sqrt{2}$

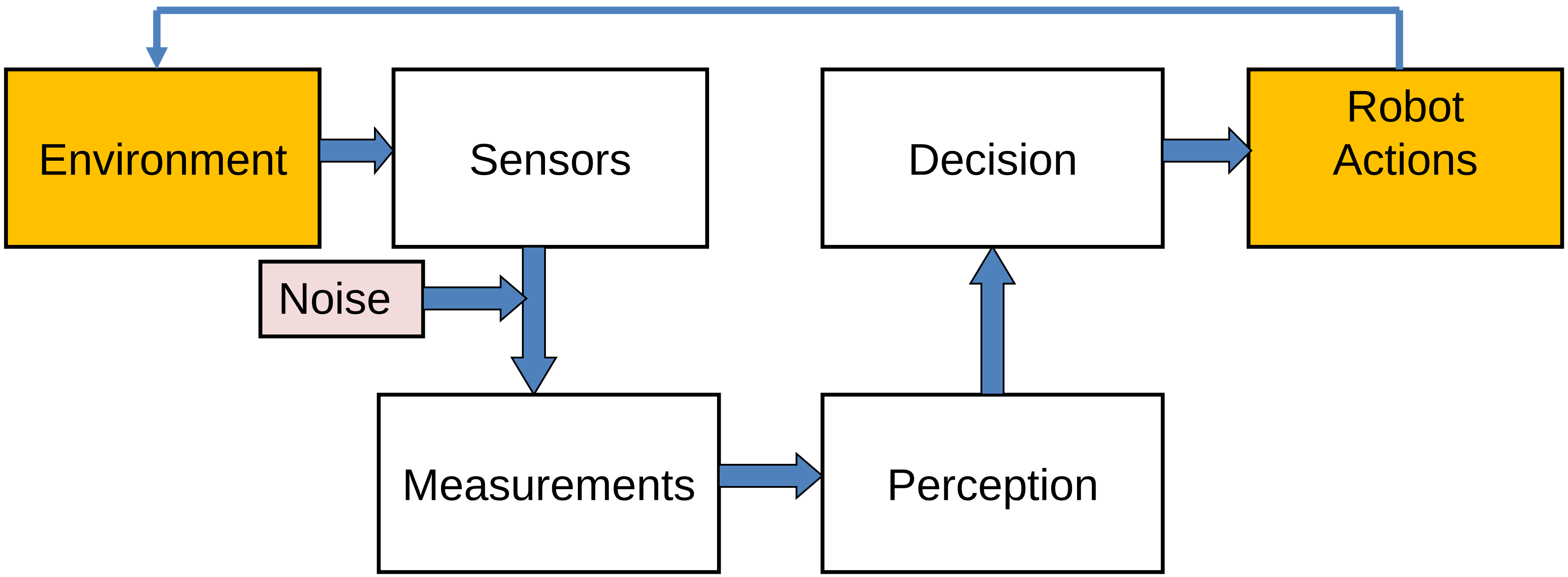
$T(n) = 1/(4\log(400+n))$

Sensor Errors and Noise – Learning and Robustness

UAVs **learn** environments from sensors

Issues in practice

- Cost-effective sensors are preferred
- Noises introduced by sensors may affect decision process



Reachability Analysis for the Nonlinear System

- Linearization

$$\dot{\mathbf{x}}(t) = f(\mathbf{x},\mathbf{u},\mathbf{x}_d)$$
$$= f(\mathbf{x}^*,\mathbf{u}^*,\mathbf{x}_d) + \nabla_{\mathbf{x}} f\big|_{\mathbf{x}=\mathbf{x}^*}(\mathbf{x}-\mathbf{x}^*)(t) + \nabla_{\mathbf{u}} f\big|_{\mathbf{u}=\mathbf{u}^*}(\mathbf{u}-\mathbf{u}^*)(t)$$
$$+ \text{higher order term},$$
$$\mathbf{x}(0) \in \mathcal{X}_0,\ \mathbf{u}(t) \in \mathcal{U}$$

- Separation
  - Enlarge by inputs
  - Enlarge by linearization error

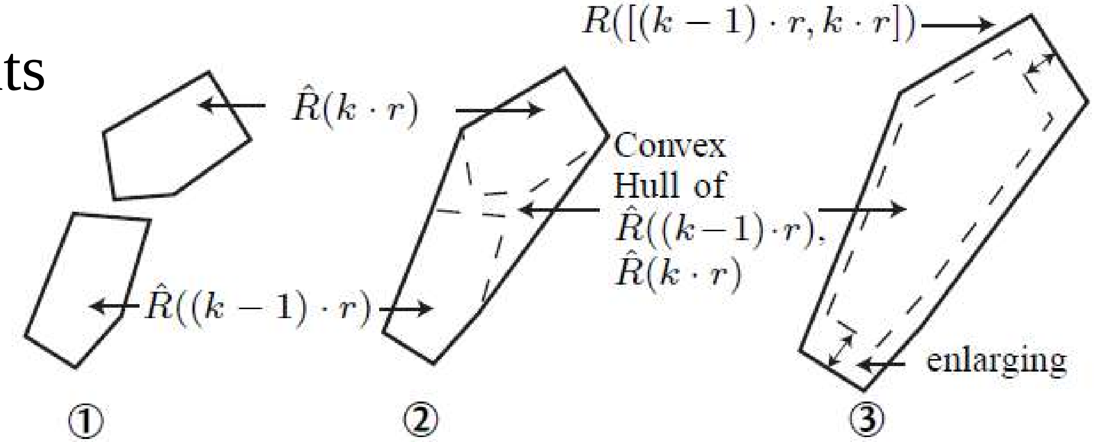
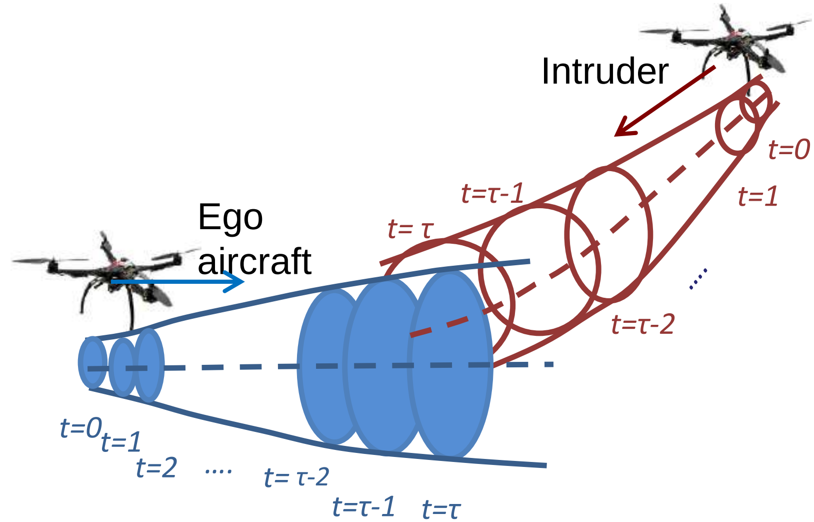
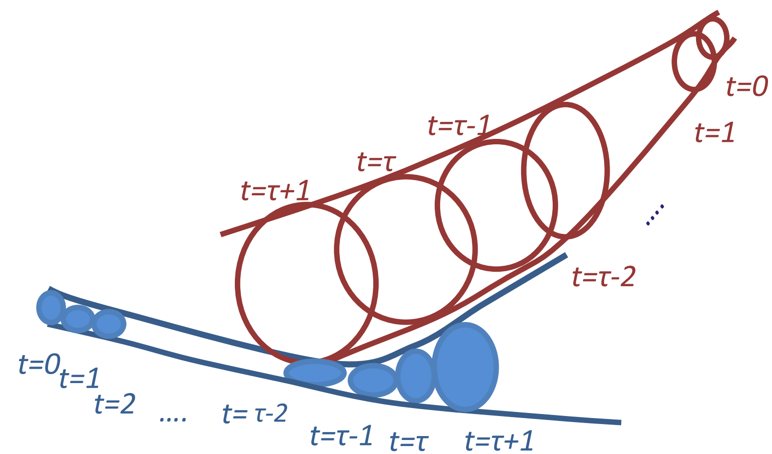


Image courtesy of [Althoff 2010]: Safety Verification of Autonomous Vehicles for Coordinated Evasive Maneuvers
Matthias Althoff, Daniel Althoff, Dirk Wollherr and Martin Buss

Collision Avoidance of Two UAVs with Time Varying Control Tubes

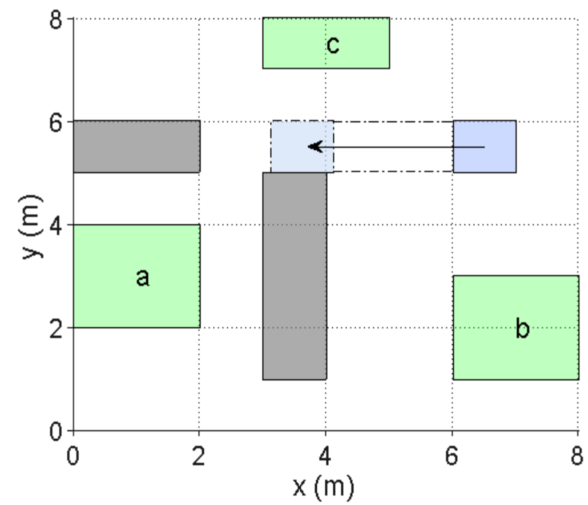
- We seek a control set update rule design for ego aircraft in a non-collaborative setting
  - Guarantee collision avoidance with reachable tube of the intruder aircraft
  - The control constraint set should be time varying
  - Collision avoidance at every time instance
- Seek a tighter control constraint 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





Path Planning with Space and Temporal Logic 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
  - A timed-automata based method



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

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 $\lozenge_{I_1} \square_{I_2}$ etc.

Optimization Based Method

$$\min_{u} \quad J(x(t,u),u(t))$$

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

$$\mathbf{x}_{t_0} \vDash \varphi$$

Remark:

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

From MTL Constraints to Linear Constraints

A polygon can be represented as intersections of several half-planes.

The constraint $z_i^t = 1$ iff $h_i^T x(t) \leq k_i$ is enforced by the linear constraints:

$$\begin{aligned} h_i^T x(t) &\leq k_i + M(1 - z_i^t) \\ h_i^T x(t) &\geq k_i - M z_i^t + \epsilon \end{aligned} \tag{1}$$

where M is a very large positive number and ϵ is a very small positive number, and $z_i^t \in \{0,1\}$.

Let $\mathcal{P} = \cap_{i=1}^{n} H_i$ be a polygon with $H_i = \{x |\, h_i^T x \leq k_i\}$.
Define $P_t^{\mathcal{P}} = \wedge_{i=1}^{n} z_i^t$, then $P_t^{\mathcal{P}} = 1$ iff $x(t) \in \mathcal{P}$.

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 $\mathbf{x}_{\mathbf{t_0}} \vDash \boldsymbol{\varphi}$ can been converted into the linear and integer constraints.

Remark:

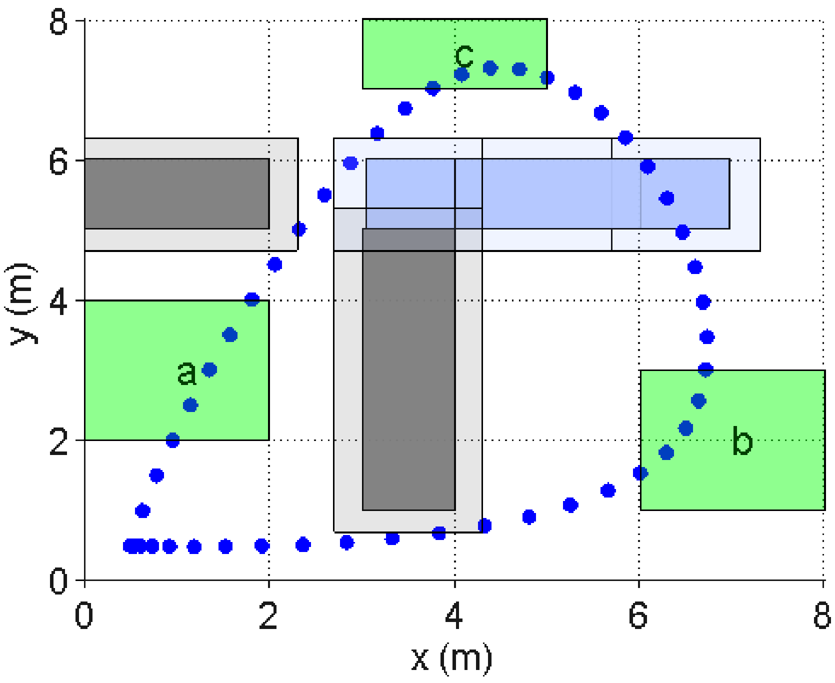
If $J(\cdot,\cdot)$ $f(.,\ldots,.)$ are linear functions of $x(t)$ and $u(t)$, then entire problem will be a Mixed-Integer Linear Optimization Problem.

Results and Discussion

- Specification in MTL

$$\phi_1 = \Diamond\Box_{[0,2]}A \wedge \Diamond\Box_{[0,2]}B \wedge \Diamond\Box_{[0,2]}C \wedge \Box\neg O$$

- The resulting trajectory for the linearized quadrotor dynamics, projected in 2D.



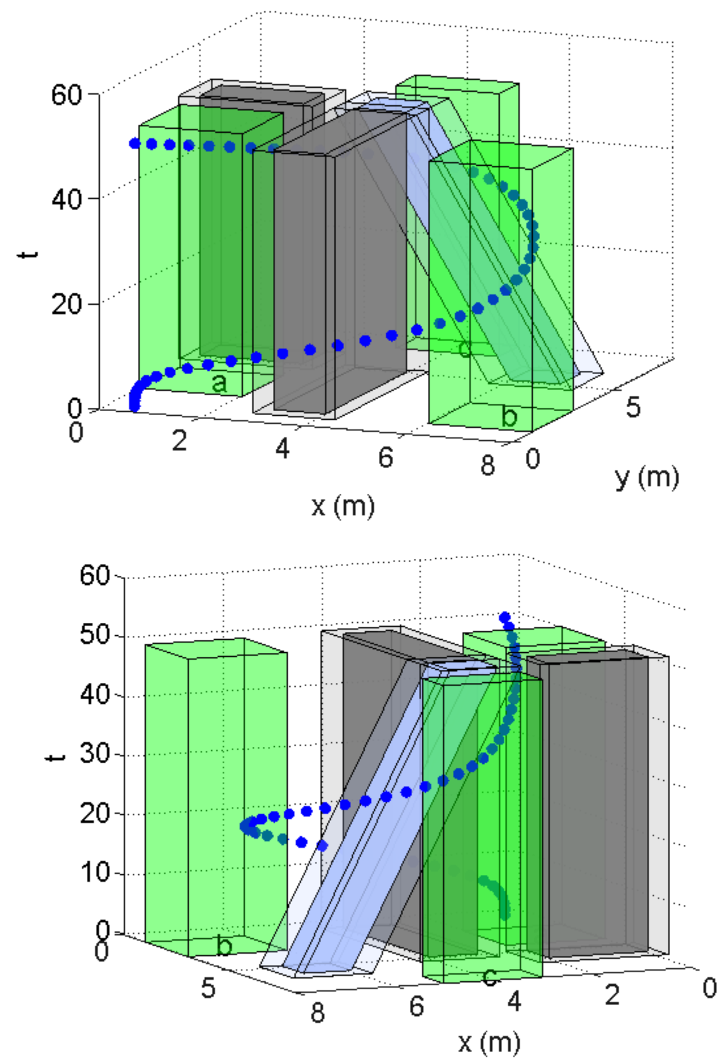
2D projection of the trajectory of the quadrotor satisfying the task.

Results and Discussion

- Specification in MTL

$$\phi_1 = \Diamond\Box_{[0,2]}A \wedge \Diamond\Box_{[0,2]}B \wedge \Diamond\Box_{[0,2]}C \wedge \Box\neg O$$

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



Manipulation Task Planning via Model Checking

- **Manipulation Task**:
  Do not grasp until reaching the object position, grasp the object within [5,10] and avoid obstacles
- **MITL**: $\emptyset = (\neg\, grasp \ \cup pos\_object) \wedge \left(\Diamond_{[5,10]}\ grasp\right) \wedge (\Box\neg pos\_obs\ )$

UPPAAL

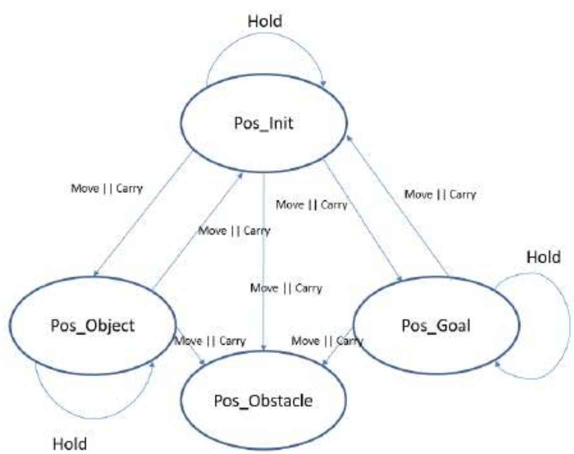
MITL specifications
(Timed automata)
[Maller, 2006]

Model-Checking Algorithm

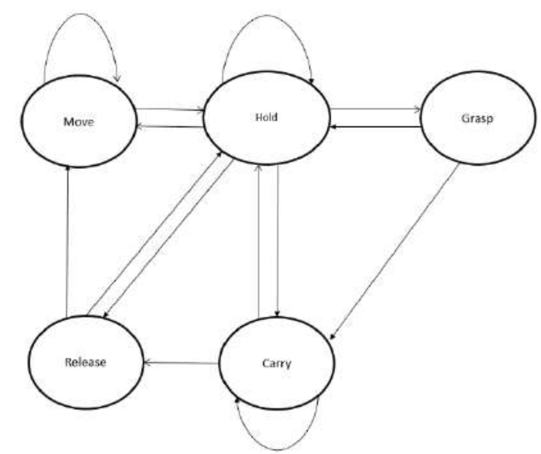
Manipulator Model
(Timed automata)

Execution sequence satisfying the MITL formula is synthesized using the UPPAAL tool

Location States



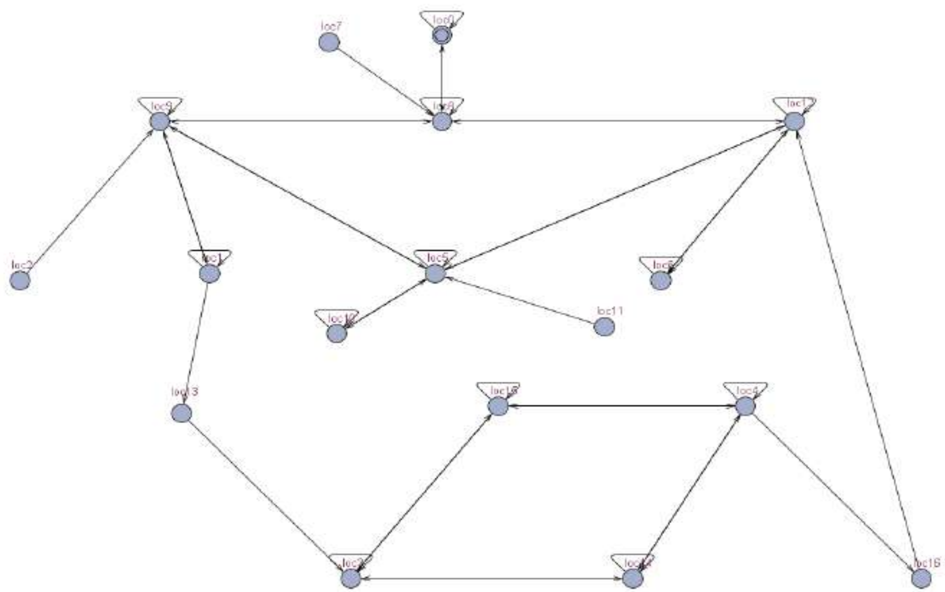
Action States

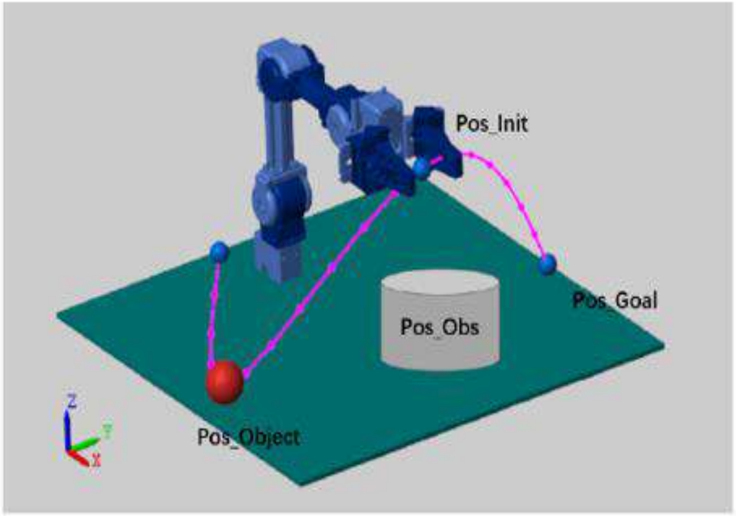


Lin and Baras, 2019 IEEE Systems Conference

Agent Model in UPPAAL

- Clock constraints not shown on the figures. Assume all transitions take 1 sec.

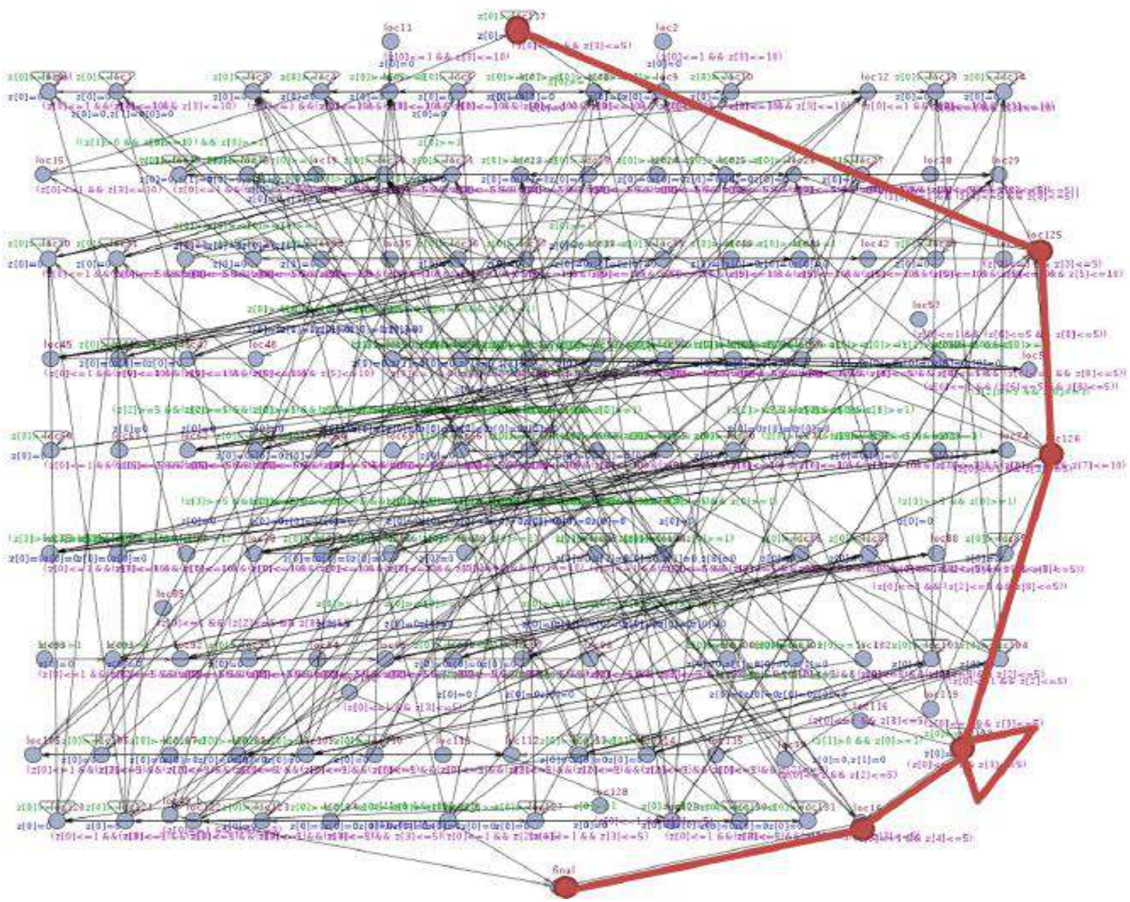




Each node is a location-action pair.
Some transitions are not possible: *(pos0, hold)* → *(pos1, hold)* or *(pos0, move)* → *(pos0, move)*

UPPAAL Solution: Case Study

$$\emptyset = (\neg\, grasp \ \cup pos\_object) \wedge (\Diamond\, [5{,}10] grasp) \wedge (\Box \neg pos\_obs\ )$$



((pos_init, hold), t=0)

((pos_init, hold), t ∈ [0,1])

((pos_init, move), t ∈ [1,2])

((pos_object, move), t ∈ [2,3])

((pos_object, hold), t ∈ [3,4])

((pos_object, hold), t ∈ [4,5])

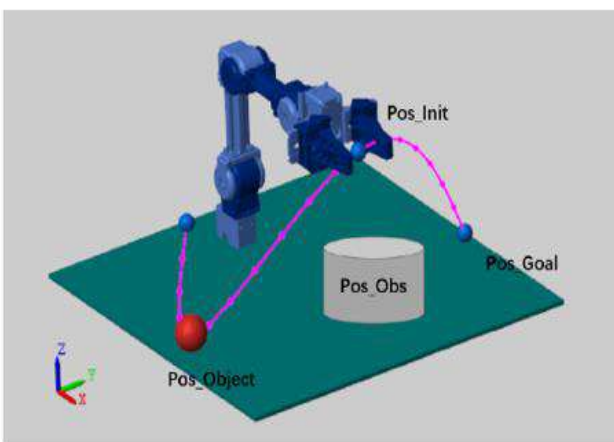
((pos_object, grasp), t ∈ [5,6])

Lin and Baras, 2019 IEEE Systems Conference

Safety Monitor Synthesis

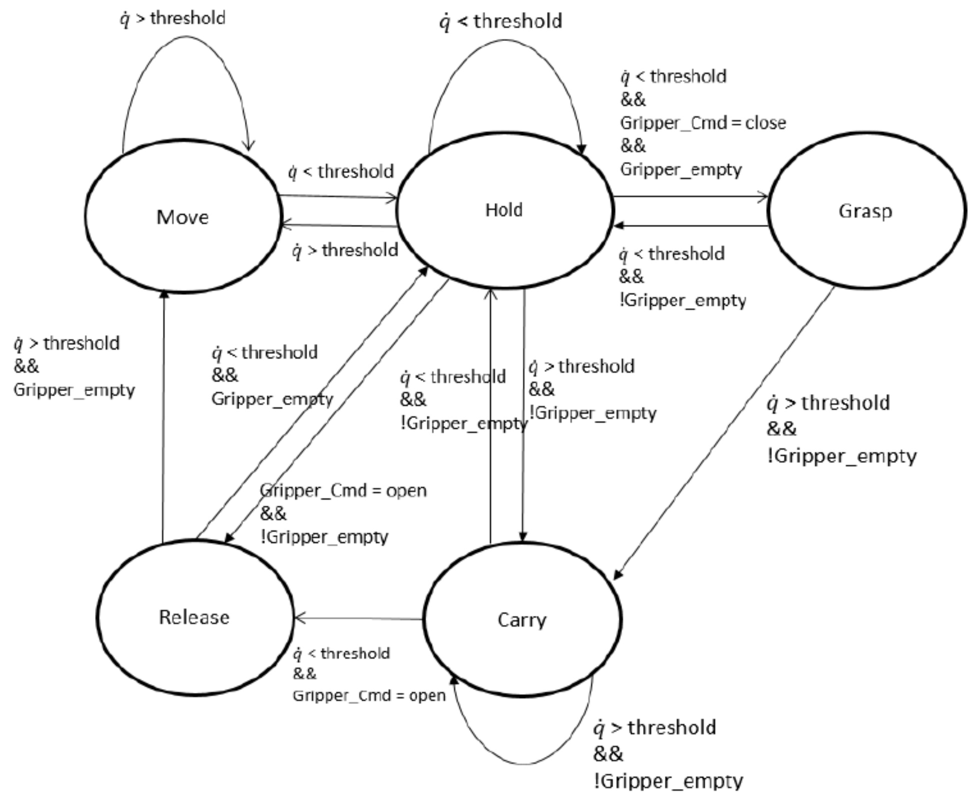
- Desired execution sequence for grasping an object:
  (pos_init, hold) → (pos_init, move) → (pos_object, move) →(pos_object, hold) → (pos_object, grasp)

- Execution sequence detected by model monitor:
  (pos_init; hold) → (pos_init, grasp) → Error detected



[LTL3 specs to Monitor, Bauer 2007]

Hybrid Automaton Manipulator Model



(1) Finite number of states
(2) Each state has continuous dynamics
(3) The transition of the hybrid automaton is determined by the sensor inputs

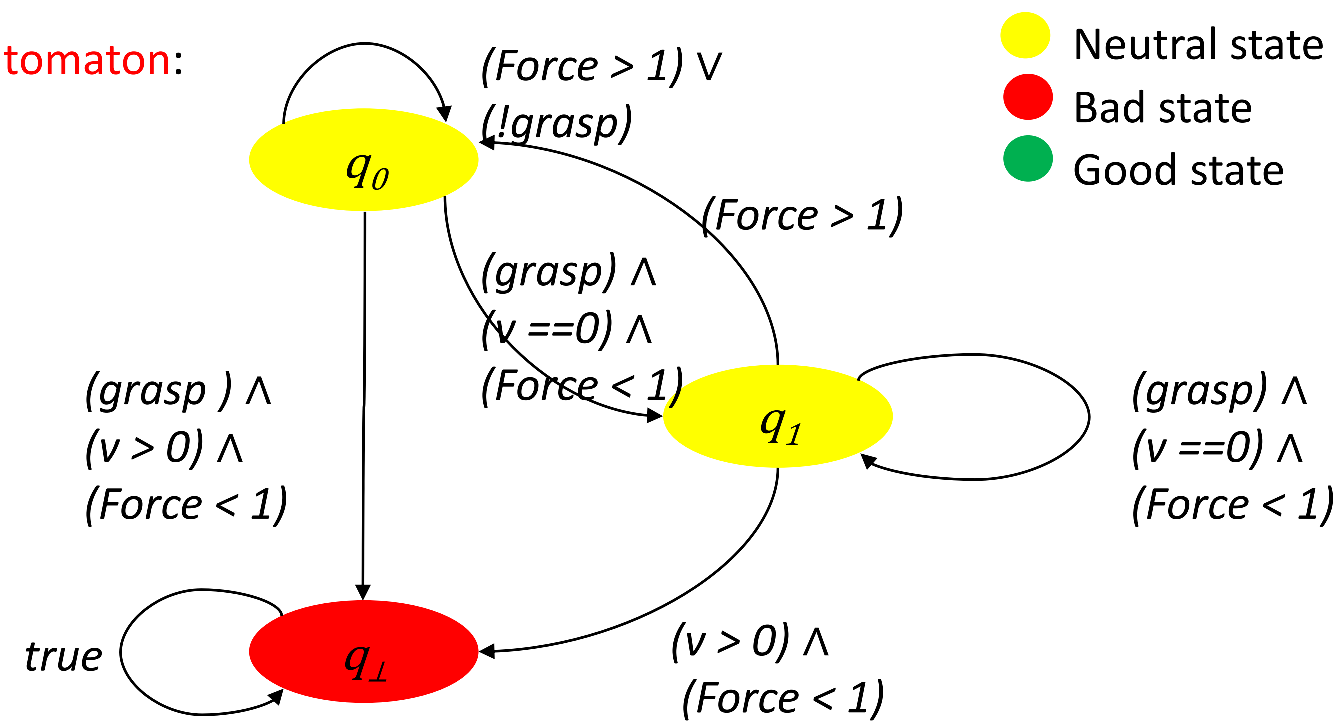
Lin and Baras, 2019 IEEE Systems Conference

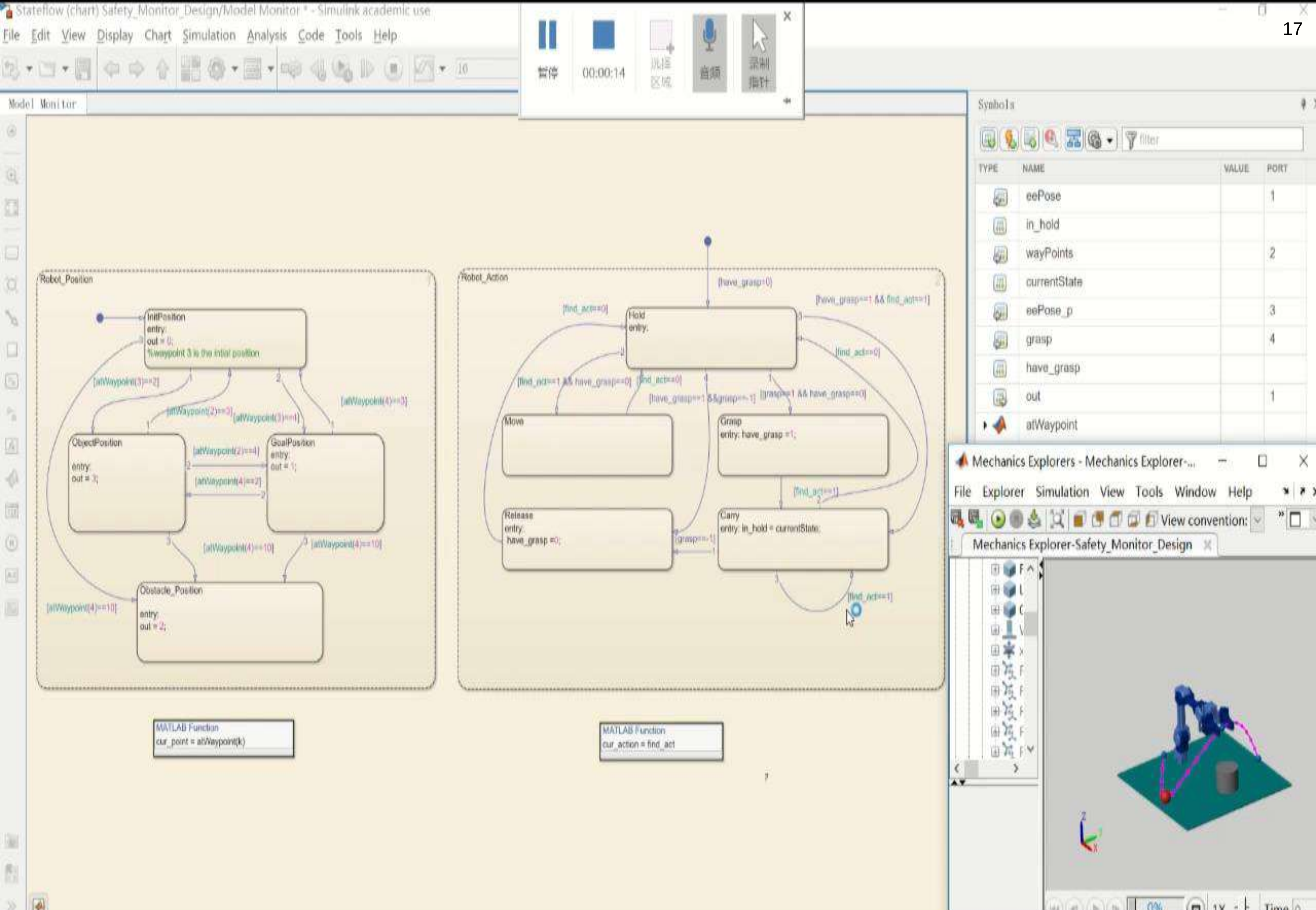
Synthesized Monitor Automaton

- Safety specification:
  The manipulator should always stay stationary while grasping, until the object is grasped firmly (Force sensor reading greater than 1N):

$$\emptyset_m = \Box\big(grasp \rightarrow (v = 0)\big) \cup (Force > 1))$$

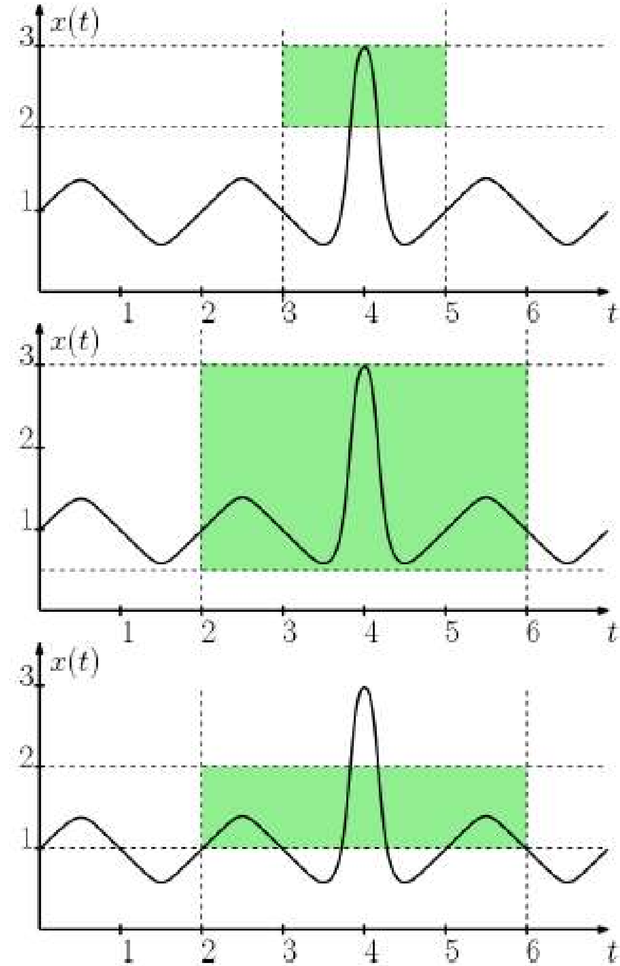
- Monitor Automaton:




Stateflow (chart) Safety_Monitor_Design/Model Monitor * - Simulink academic use
File Edit View Display Chart Simulation Analysis Code Tools Help
暂停
00:00:14
音频
Model Monitor
Symbols
TYPE NAME VALUE PORT
eePose 1
in_hold
wayPoints 2
currentState
eePose_p 3
grasp 4
have_grasp
out 1
atWaypoint
Robot_Position
InitPosition
entry:
out = 0;
%waypoint 3 is the initial position
ObjectPosition
entry:
out = 3;
GoalPosition
entry:
out = 1;
Obstacle_Position
entry:
out = 2;
Robot_Action
Hold
entry:
Move
Grasp
entry: have_grasp =1;
Release
entry:
have_grasp =0;
Carry
entry: in_hold = currentState;
[have_grasp=0]
[find_act==0]
[grasp==-1]
[find_act==1]
MATLAB Function
cur_point = atWaypoint(k)
MATLAB Function
cur_action = find_act
Mechanics Explorers - Mechanics Explorer-...
File Explorer Simulation View Tools Window Help
View convention:
Mechanics Explorer-Safety_Monitor_Design
0%
1X
Time

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 $\boxed{\rho^{\phi_3}(x,0) = -1}$



Event-triggered Feedback Control Laws for Temporal Logics

- For $\boldsymbol{w}(t) \in \mathcal{W}$, consider the nonlinear system

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

- Consider the STL fragment

$$\psi ::= \top \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2 \tag{2a}$$

$$\phi ::= G_{[a,b]}\psi \mid F_{[a,b]}\psi \mid F_{[a',b']}G_{[a'',b'']}\psi \tag{2b}$$

$$\theta ::= \bigwedge_{k=1}^{K} \phi_k \text{ with } b_k \leq a_{k+1} \mid \tilde{\phi}_1 \tag{2c}$$

where $\tilde{\phi}_k := F_{[c_k,d_k]}(\psi_k \wedge \tilde{\phi}_{k+1})$ for all $k \in \{1, \ldots K-1\}$ and $\tilde{\phi}_K := F_{[c_K,d_K]}\psi_K$.

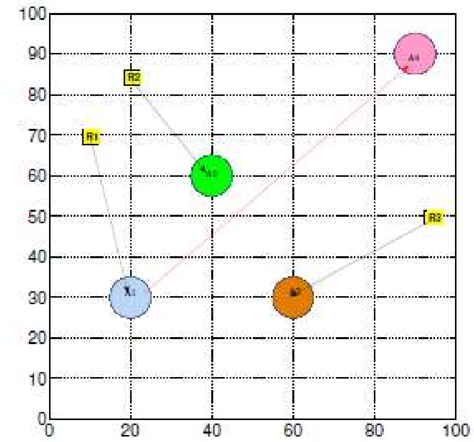
Problem

Given the system (1) and a formula ϕ as in (2b), derive an **event-triggered** control law $\hat{\boldsymbol{u}}$ which ensures $0 < r \leq \rho^{\phi}(\boldsymbol{x}, 0)$.

Simulations – Safe/Robust Collaboration

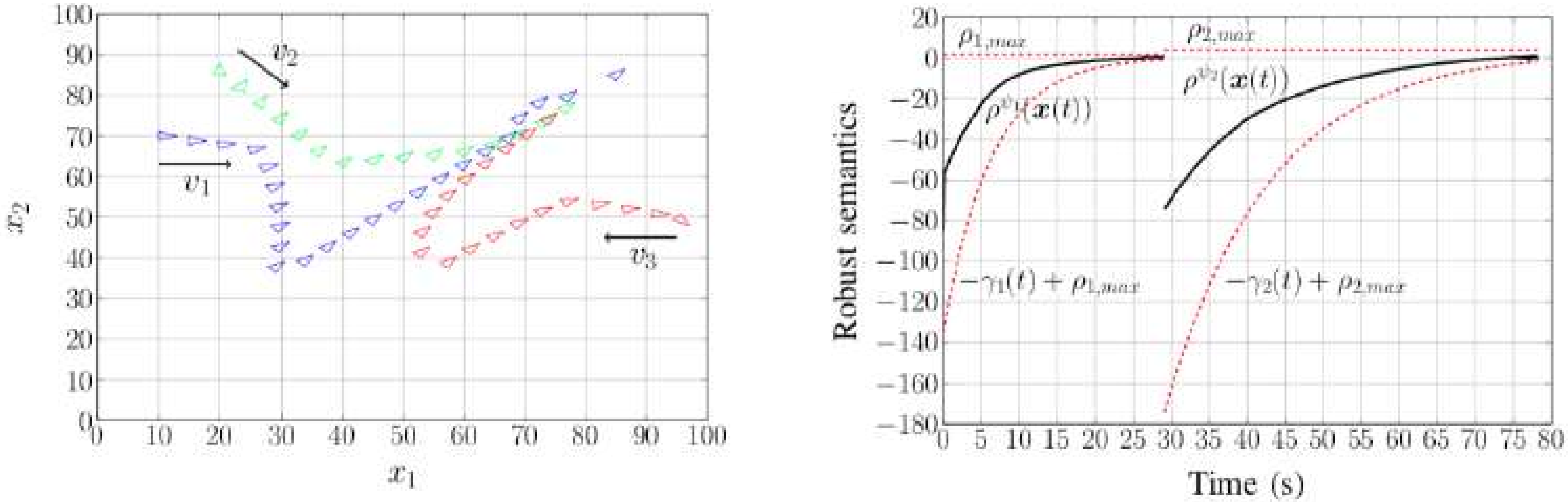
Simulations

- v_1: Eventually within $[0, 50]$ go to **A1** with $\theta_1 \approx 45^o$ and stay close to v_3
  Eventually within $[50, 100]$ go to **A4**, with $\theta_1 \approx 45^o$ and stay close to **R2**
- v_2: Eventually within $[0, 50]$ go to **A2** with $\theta_2 \approx 45^o$
  Stay close to **R1** and **R3** and keep $\theta_2 \approx 45^o$
- v_3: Eventually within $[0, 50]$ go to **A3** with $\theta_3 \approx 45^o$
  Stay close to **R3** and keep $\theta_3 \approx 45^o$



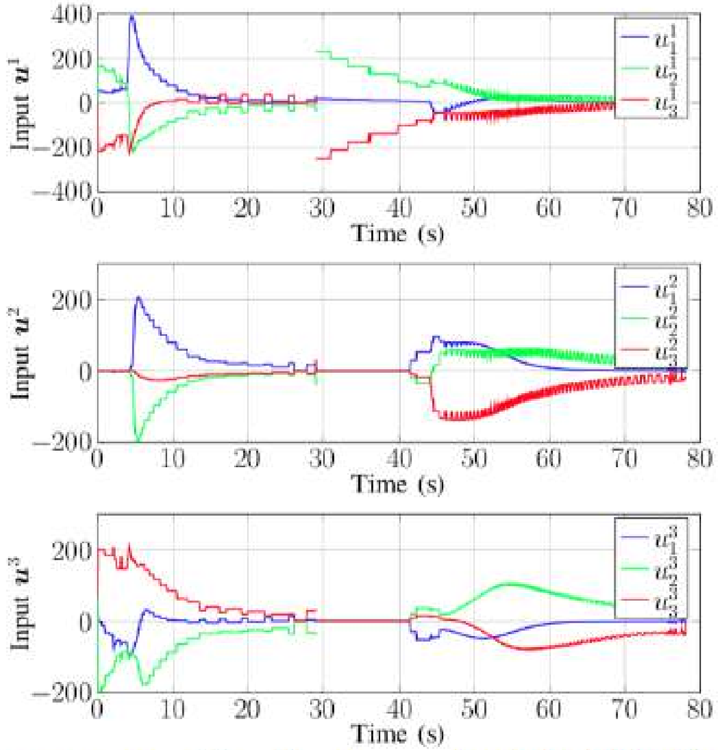
$$\phi := F_{[0,50]}((\|\boldsymbol{p}_1 - \boldsymbol{A1}\| \le \epsilon) \wedge (\|\boldsymbol{p}_2 - \boldsymbol{A2}\| \le \epsilon) \wedge (\|\boldsymbol{p}_3 - \boldsymbol{A3}\| \le \epsilon) \wedge$$
$$(|\theta_1 - 45^o| \le \epsilon) \wedge (|\theta_2 - 45^o| \le \epsilon) \wedge (|\theta_3 - 45^o| \le \epsilon)$$
$$(\|\boldsymbol{p}_1 - \boldsymbol{p}_3\| \le \epsilon))$$
$$\wedge F_{[50,100]} \cdots$$

Simulations – Safe/Robust Collaboration



- The task ϕ is robustly satisfied with $r := 0.5$

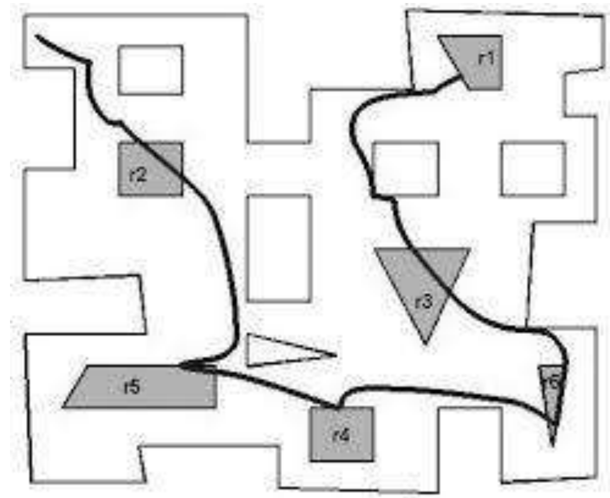
Simulations – Safe/Robust Collaboration



- The experiment was implemented with 100 Hz frequency and a total of **7725** samples.
- Only **185** triggerings, which corresponds to a reduction in communication and computation by 97.6% compared to time-triggered control.

Composable Formal Models for Safety in Autonomous Systems: Safe Robot Navigation Under Temporal Constraints

Motivation – Robot Navigation Requirements



Robot Navigation Requires

- **Temporal Constraints:**
  Navigation in Prescribed Time
- **Safety:**
  Collision Avoidance
- **Complex Planning Objectives:**
  Composition of Point-to-Point Planning Tasks
- **Real-Time Computation:**
  Avoid Complex Integer-Programming Optimization
- **High- & Low- Level Planning:**
  Simultaneously Tackle Motion Planning and Control of the Robot
- **Robustness & Adaptability**

Math. Formalism: MITL Specifications to Control Problems

Navigation Within Given Time Interval

- MITL expression

Navigating from $x(0)$ to a neighborhood $\mathcal{B}(x_p, r_p)$ of W within a given time interval $J = [0, \tau_p]$:

$$\begin{aligned} & x_0 \models \Diamond_J p \\ s.t. \quad & p = p_{(x_p, r_p)} \in \mathcal{P} \\ & J = [0, \tau_p] \\ & x_0 \triangleq x(0) \in W, \end{aligned}$$

which states that $\|x(t) - x_p\| \leq r_p$ for some $t \in [0, \tau_p]$.

- Control Problem

Assuming single integrator robot kinematics,

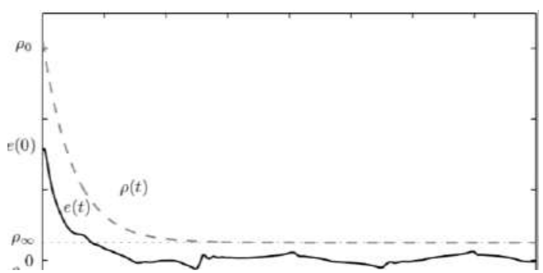
$$\dot{x} = u$$

$$u^p : \mathbb{R}_{\geq 0} \times \mathcal{W} \to \mathbb{R}^n$$

determine

such that $\mathcal{W}$ is forward invariant and

$$\|x(t) - x_p\| < r_p, \quad \forall t \geq \tau_p.$$



$$u_\gamma(t, x) = -\nabla\left(\frac{1}{2}\left[\left(k + \frac{\alpha(t)}{2}\right)\gamma(x) + k\varepsilon(t, x)(\gamma(x) - \rho(t))\right]\right)$$

Obstacle Avoidance

- MITL expression

Avoiding a neighborhood $\mathcal{B}(x_p, r_p)$ of W throughout a given time interval $J = [0, \tau_p]$:

$$\begin{aligned} & x_0 \models \Box_I \neg p, \\ s.t. \quad & p = p_{(x_p, r_p)} \in \mathcal{P} \\ & J = [0, \tau_p] \\ & x_0 \triangleq x(0) \in W \setminus \{q \in W : \|q - x_p\| \leq r_p\} \end{aligned}$$

which states that $\|x(t) - x_p\| > r_p$ for all $t \in [0, \tau_p]$.

- Control Problem

Assuming single integrator robot kinematics,

$$\dot{x} = u$$

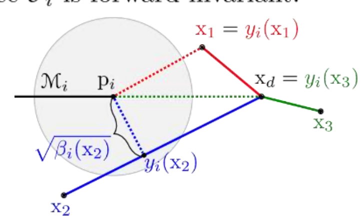
and for any obstacle

$$O_i \triangleq \{q \in W : \|q - p_i\| \leq r_i\}, \; i \in J$$

and initial configuration $x_0 \in \mathcal{F}_i = W \setminus O_i$, determine a time-varying controller

$$u : \mathbb{R}_{\geq 0} \times \mathcal{F}_i \to \mathbb{R}^n$$

such that the free space $\mathcal{F}_i$ is forward invariant.



$$u_{\beta_i}(x) = \frac{\sigma_{i,\delta}(x)}{d_i(x)}\left(\nabla\beta_i(x) + \mathbb{1}_{\mathcal{M}_i}(x)\,(x - x_d)^\perp\right)$$

$$\beta_i(x) = \inf\left\{\|q - p_i\|^2 : q \in \mathscr{S}(x)\right\}, \quad i \in \mathscr{J}$$

Mavridis, Vrohidis, Baras, Kyriakopoulos, 2019 IEEE CDC

Key Idea – Adaptive Controllers for Composable MITL Tasks

Navigation within time interval J

$$
\begin{aligned}
&\mathrm{x}_0 \models \Diamond_J p \\
s.t.\quad & p = p_{(\mathrm{x}_p, r_p)} \in \mathcal{P} \\
& J = [0, \tau_p] \\
& \mathrm{x}_0 \triangleq \mathrm{x}(0) \in \mathcal{W}
\end{aligned}
$$

Collision-free navigation

$$
\begin{aligned}
&\mathrm{x}_0 \models \Box_I \neg p, \\
s.t.\quad & p = p_{(\mathrm{x}_p, r_p)} \in \mathcal{P} \\
& J = [0, \tau_p] \\
& \mathrm{x}_0 \triangleq \mathrm{x}(0) \in \mathcal{W} \setminus \{q \in \mathcal{W} : \|q - \mathrm{x}_p\| \leq r_p\}
\end{aligned}
$$

if $p \in \mathcal{P}$ then

$$p(x) = \top \Leftrightarrow \left\|\mathrm{x} - \mathrm{x}_p\right\| \leq r_p$$

$$\dot{\mathrm{x}} = u_\gamma(t, \mathrm{x}, \rho)$$
$$\rho(t) = (\rho_0 - \rho_\infty)e^{-l(J)t} + \rho_\infty$$

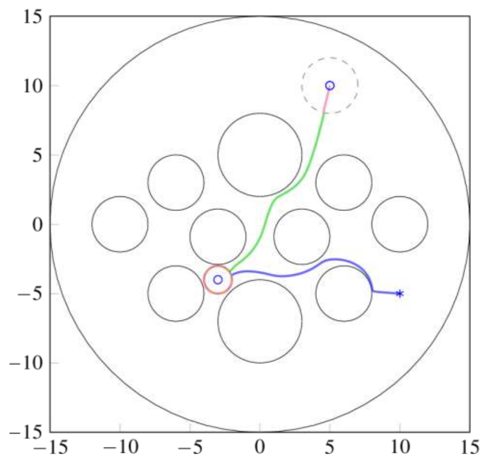
$$\dot{\mathrm{x}} = u_\beta(t, \mathrm{x})$$

Safe Navigation in Prescribed Time

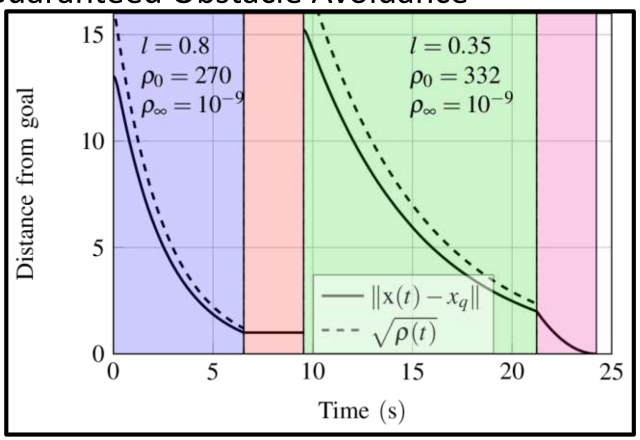
$$\dot{\mathrm{x}} = u_\gamma(t, \mathrm{x}, \rho) + u_\beta(t, \mathrm{x})$$

Robot Navigation Under Spatio-Temporal (MITL) Constraints using Time-Dependent Vector Field Control

Robot Navigation



- Safe Robot Navigation in Prescribed Time Interval
- Guaranteed Obstacle Avoidance



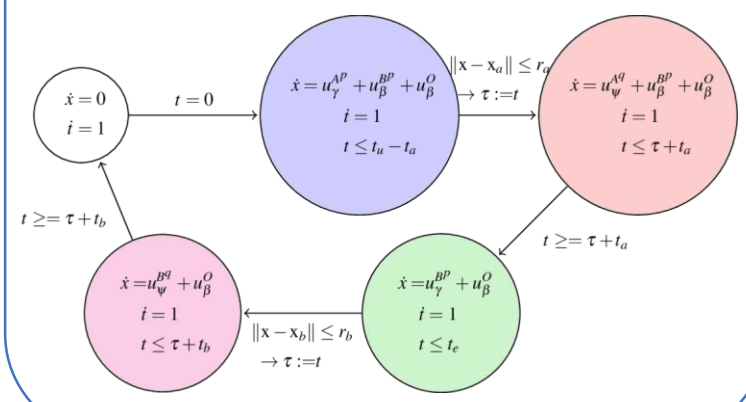
Under Spatio-Temporal (MITL) Constraints

$$\phi = \left(\Diamond_{[0,t_e+t_b]}\Box_{[0,t_b]}B^q\right) \wedge \left(\neg B^q \mathbf{U}_{[0,t_u]}\Box_{[0,t_a]}A^q\right) \wedge \left(\bigwedge_{i\in J} \Box\neg O_i\right)$$

"Until time t_u, enter the sphere $\{x \in W : \|x - x_a\| \leq r_a\}$, and, for time duration of $t_a < t_u$, do the task $f_a(t, x, x_a, r_a) = 0$. After this is finished $(t \leq t_u)$, enter the sphere $\{x \in W : \|x - x_b\| \leq r_b\}$ before time $t_e > t_u$, and, for time duration of t_b, do the task $f_b(t, x, x_b, r_b) = 0$. Meanwhile, at all times, avoid the obstacles $\{x \in W : \|x - p_i\| \leq r_i\}$, $i \in J$."

- Task Execution with Precedence Constraints

Using Time-Dependent Vector-Field-Based Control (Hybrid Automaton)



Online Control with Automatically Created Hybrid Automata – No Optimization Solver

Mavridis, Vrohidis, Baras, Kyriakopoulos, 2019 IEEE CDC

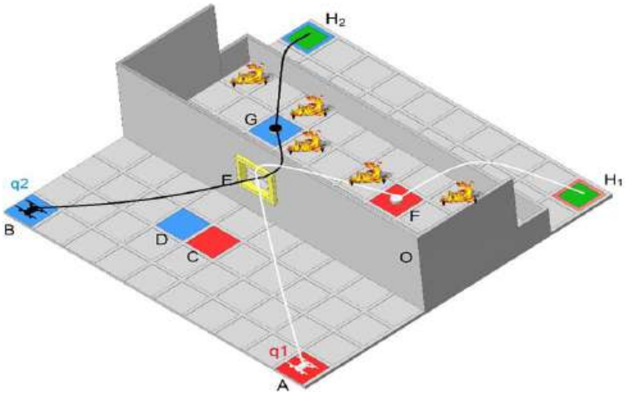
Composable and Safe Autonomy in Multiagent Systems

Hybrid, Compositional, Suboptimal, Real-time Mission Planning for UAVs

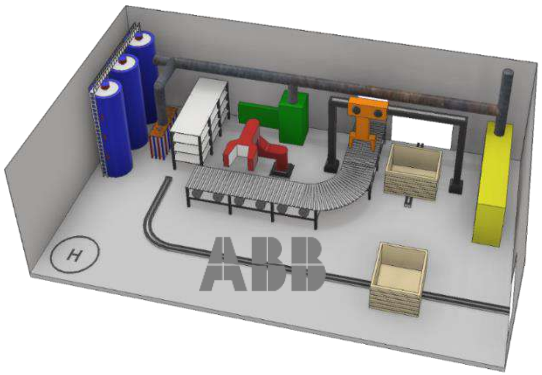
Fiaz and Baras, 2018-2020

Composable, Safe, Scalable Planning for Autonomous UAV Missions: Problem description

- **Mission**: Any high-level assignment for UAVs
  - Autonomous search and rescue and disaster relief
  - Inspection tasks in complex workspaces
- **Safety** represented as:
  - Finite time constraints
  - Spatial constraints
- **Objective**: Safe, autonomous completion of the task
  - Meet finite time constraints
  - Avoid obstacles and collision with other UAVs
- **Complexity** of solution is important!
  - Real-time solvable
  - Onboard computable



Search and rescue with multiple UAVs: constrained environment, limited time to evacuate



A smart manufacturing factory: constrained and dynamic indoor environment for safety

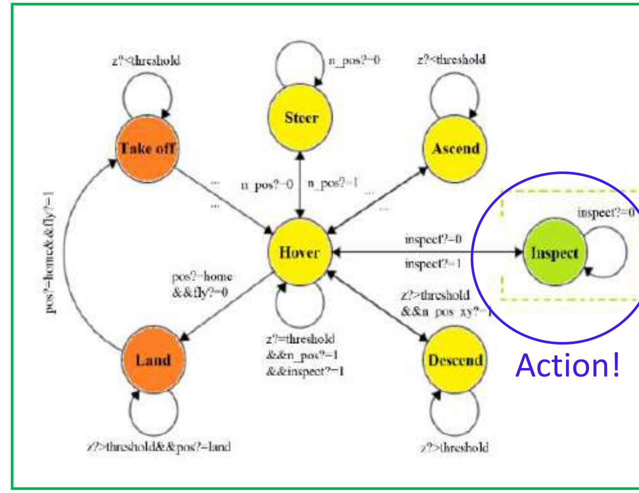
Fiaz and Baras, 2020 IFAC World Congress

Method: Divide and conquer (1)

- Represent mission as Metric(/Signal) Temporal Logic Specification

$$\phi_i = \Diamond_{[0,T_1]}(Object\ Location) \wedge \Box_{[0,T_2]}(Object\ Location)' \\ \wedge \Diamond_{[0,T_3]}\Box(Safe\ Location) \wedge \Box\neg(Obstacles) \wedge \Box\neg q_j$$

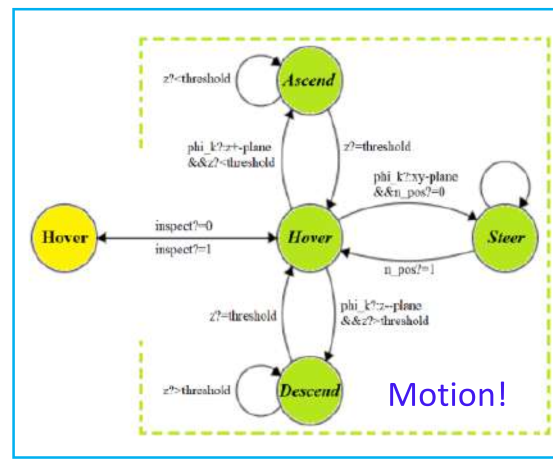
- Represent system dynamics as a Hybrid system model



- Systematically decompose mission specification into sub-tasks

Theorem 1. Given an MITL specification ϕ_i, there exists some finite M-length decomposition ϕ_i^k, $k \in \{1, 2, ..., M\}$, s.t. $\wedge_{k=1}^{M}(\phi_i^k) \implies \phi_i$, if $\sum_{k=1}^{M} T_k \leq T_i$, where T_i is the finite timing interval for ϕ_i, and T_ks are the corresponding finite timing intervals for ϕ_i^k, $\forall k \in \{1, 2, ..., M\}$.

- Represent *action* specifications as *motion* specifications



- Formulate optimal control problems for each sub-task

$$\begin{aligned} \min_{x_i, u_i} \quad & \sum_{t=0}^{T} |u_i(t)| \\ \text{s.t.} \quad & x_i(t+1) = A_l(t)x_i(t) + B_l(t)u_i(t) \\ & \mathbf{x}_{i t_0} \models \phi_i^k \end{aligned}$$

Method: Divide and conquer (2)

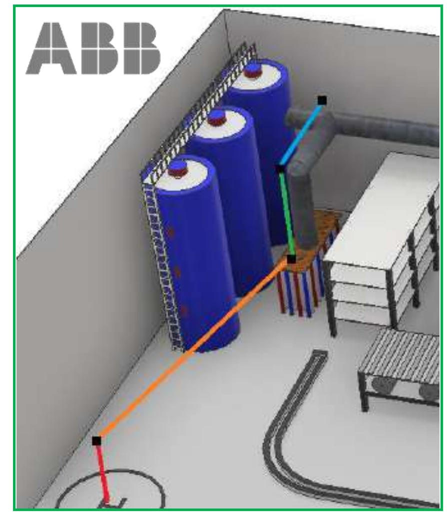
- Translate each sub-task specification to "convex" constraints

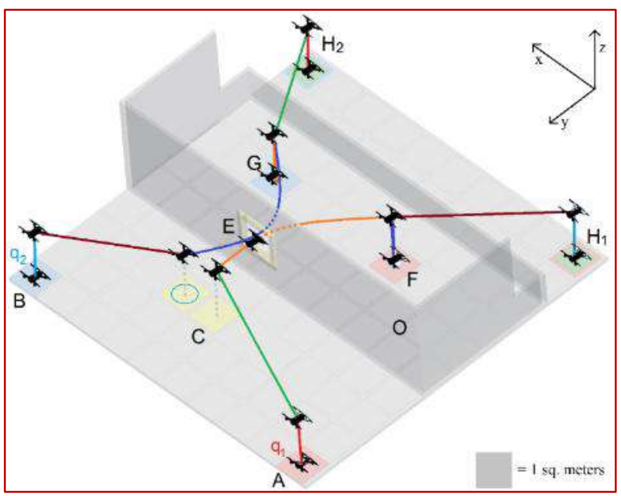
$$\boxed{\mathbf{x}_{i t_0} \models \phi_i} \qquad \boxed{x(t) \in \mathcal{K}}$$

$$\boxed{x(t) \in \cap_{i=1}^{n} \mathcal{H}_i = \cap_{i=1}^{n} \{x : h_i^T x \leq a_i\}}$$

$$\boxed{\begin{array}{c} h_i^T x(t) \leq a_i + M(1 - b_i^t) \\ h_i^T x(t) \geq a_i - M b_i^t + \epsilon \end{array}}$$

- Solve a Mixed Integer Linear Program (MILP) for each sub-task
- Generate sub-optimal final trajectory by composing optimal sub-paths





Each colored segment represents the optimal trajectory resulting from solving a MILP for its respective sub-task.

The complexity is still exponential, but because of reduction in "size" of the parent problem, this approach is shown to be **fast** and **scalable**!

Simulation and Results

- For the specified task:
  - Decomposed sub-tasks for quadrotor (q1)

$$\phi_{q1(AA')} = \Box A \wedge \Diamond_{[0,5]} A' \quad [mode : Take\ off]$$
$$\phi_{q1(AC)} = \Diamond_{[0,5]} C \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q1(CF)} = \Diamond_{[0,10]} F \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q1(FF')} = \Box F \wedge \Diamond_{[0,10]} F' \quad [mode : Grasp]$$
$$\phi_{q1(FH_1)} = \Diamond_{[0,10]} H_1 \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q1(H_1H_1')} = \Box H_1 \quad [mode : Land]$$

  - Decomposed sub-tasks for quadrotor (q2)

$$\phi_{q2(BB')} = \Box B \wedge \Diamond_{[0,5]} B' \quad [mode : Take\ off]$$
$$\phi_{q2(BD)} = \Diamond_{[0,5]} D \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q2(D)} = \Box D\ \mathcal{U}(\neg(pos(q1)? = C) \quad [mode : Hover]$$
$$\phi_{q2(DG)} = \Diamond_{[0,10]} G \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q2(GG')} = \Box G \wedge \Diamond_{[0,10]} G' \quad [mode : Grasp]$$
$$\phi_{q2(GH_2)} = \Diamond_{[0,10]} H_2 \wedge \Box\neg O \quad [mode : Steer]$$
$$\phi_{q2(H_2H_2')} = \Box H_2 \quad [mode : Land]$$

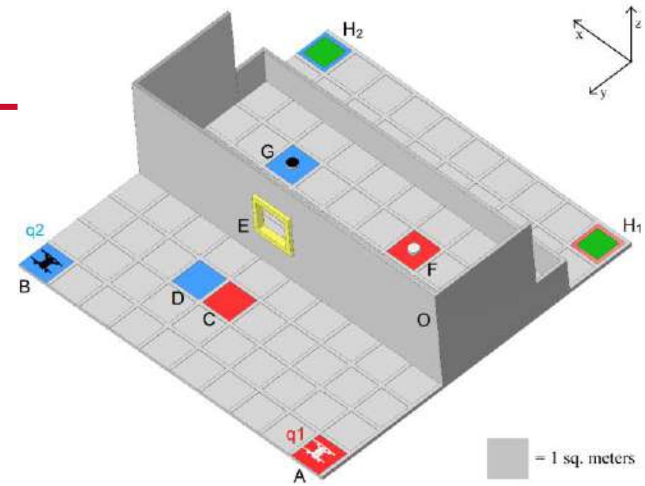
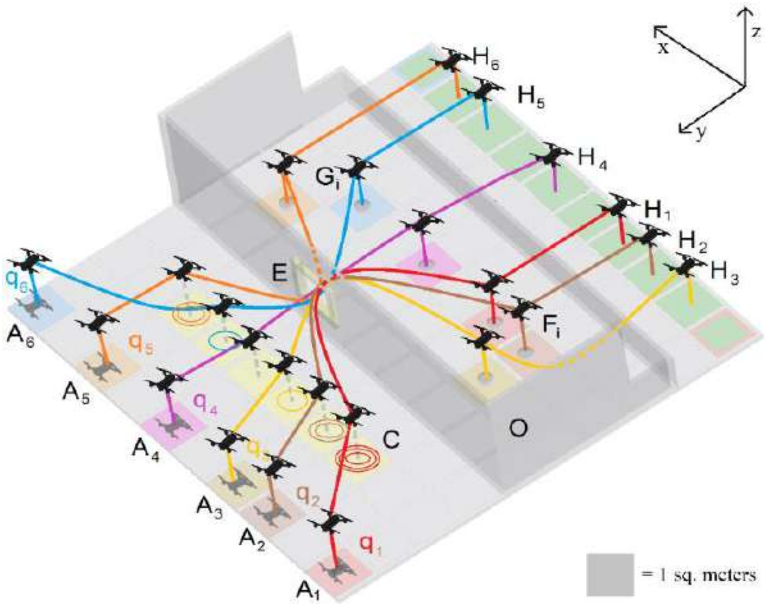


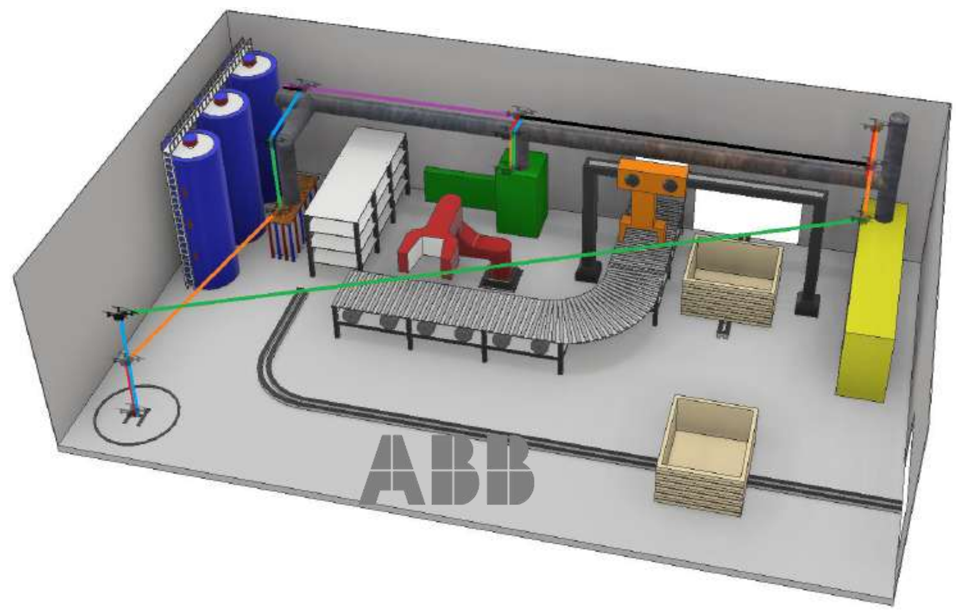
TABLE I

COMPUTATION TIMES FOR SUB-TASKS (ϕ_{sub_i})

Task(q1)	Time (sec)	Task(q2)	Time (sec)
$\phi_{q1(AA')}$	2.7	$\phi_{q2(BB')}$	2.7
$\phi_{q1(AC)}$	6.3	$\phi_{q2(BD)}$	5.8
$\phi_{q1(CF)}$	10.3	$\phi_{q2(D)}$	2.0
$\phi_{q1(FF')}$	3.0	$\phi_{q2(DG)}$	11.1
$\phi_{q1(FH_1)}$	5.7	$\phi_{q2(GG')}$	3.0
$\phi_{q1(H_1H_1')}$	2.5	$\phi_{q2(GH_2)}$	5.7
—	—	$\phi_{q2(H_2H_2')}$	2.5

Application of the method so far, and prospects





[1]: Fast, composable rescue mission planning for UAVs using metric temporal logic (Proc. IFAC World Congress 2020)

[2], [3]: Safe, composable mission planning for UAV-based inspection tasks* (ICRA/RA-L & ACC/CS-L, two papers to be submitted)

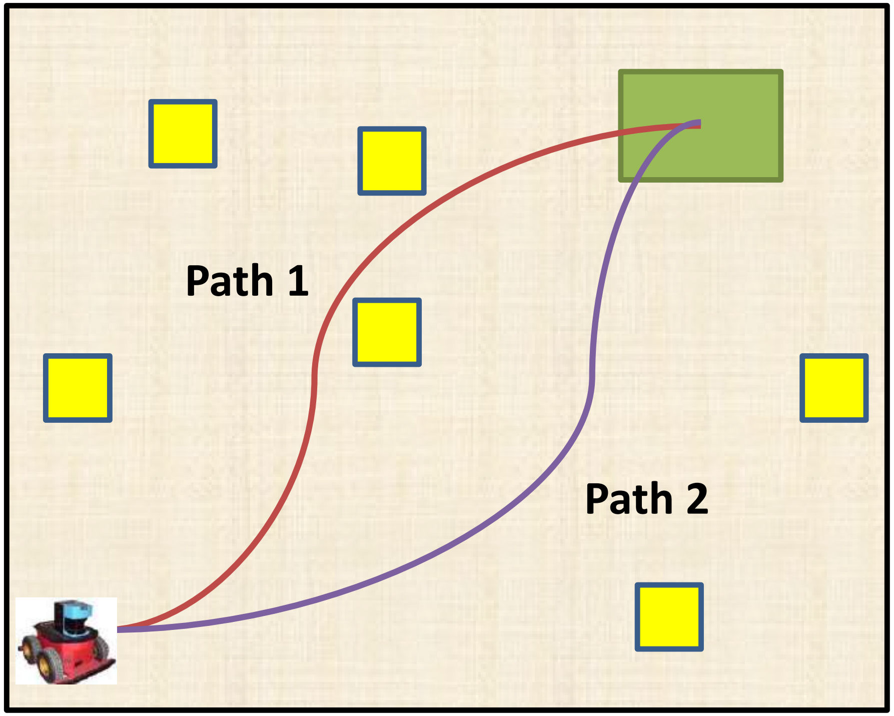
Current focus: (1) Incorporate safe learning for collaborative tasks between agents,
(2) robustness analysis to changes in environment, and
(3) enable self-monitoring and self-correction during execution.

Fiaz and Baras, 2020 IFAC World Congress

*Joint work with ABB Corporate Research, US

Optimization Based task planning with space and time tolerances

Space and Time Tolerances in Task Planning

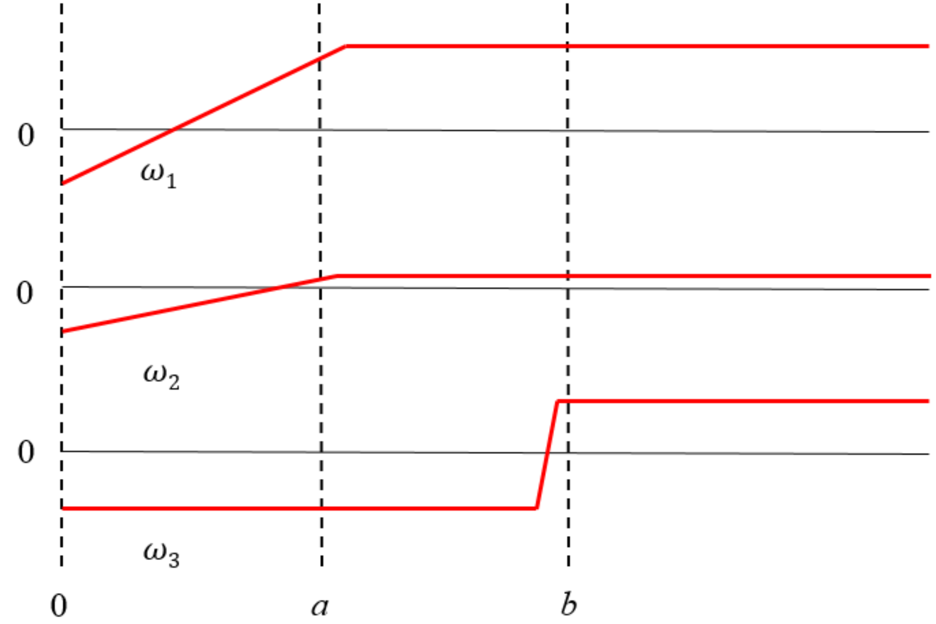


- Tolerances are import factors of the plan, runtime execution may differ from planning.

- Planned Path 2 is referred over planned Path 1, since it has a better space tolerance.

Space and Time Tolerances in Task Planning

- all three signals are considered as satisfying $\Diamond_{[a,b]}(x > 0)$ from t = 0 at the same degree.
- For ω_2, the specification will be violated if we disturb x a little
- For ω_3, the specification will be violated if we shift the signal a little to the right



Space and Time Tolerances in Task Planning

Definition (Space Robustness)

Space robustness quantifies how well a given signal s satisfies a given formula. The robustness degree is calculated recursively according to the quantitative semantic:

- $r(s, (f(s) < d), t) = d - f(s_t)$,
- $r(s, \neg(f(s) < d), t) = -r(s, (f(s) < d), t)$,
- $r(s, \varphi_1 \wedge \varphi_2, t) = \min(r(s, \varphi_1, t), r(s, \varphi_2, t))$,
- $r(s, \varphi_1 \vee \varphi_2, t) = \max(r(s, \varphi_1, t), r(s, \varphi_2, t))$,
- $r(s, \Diamond_{[a,b]}\varphi, t) = \max\limits_{t' \in [t+a, t+b]} r(s, \varphi, t')$,
- $r(s, \Box_{[a,b]}\varphi, t) = \min\limits_{t' \in [t+a, t+b]} r(s, \varphi, t')$,

Space and Time Tolerances in Task Planning

Definition (Time Robustness)

The left and right time robustness of an STL formula with respect to a trace s at time t are defined as follows

$$\theta^{-}(s, f(s), t) = \max(d \geq 0, s.t. \forall t' \in [t-d, t], (s,t) \vDash \varphi \;\Leftrightarrow, (s,t') \vDash \varphi)$$

$$\theta^{+}(s, f(s), t) = \max(d \geq 0, s.t. \forall t' \in [t, t+d], (s,t) \vDash \varphi \;\Leftrightarrow, (s,t') \vDash \varphi)$$

How much can we shift a signal to the left (or right), such that the specification is still satisfied?

Space and Time Tolerances in Task Planning

$$\max_{u(t)} \quad \alpha\, r_{time} + \beta\, r_{space}$$

Subject to

$$x(t+1) = f(t, x(t), u(t))$$

$$u_{min} \leq u(t) \leq u_{max}$$

$$\boldsymbol{X}_{t_0} \vDash \varphi$$

- α and β are the weighting coefficient and we have $\alpha + \beta = 1$
- We consider linear robot dynamics (for nonlinear dynamics use linearization)
- The timed temporal constraint $\mathbf{X}_{\mathbf{t_0}} \vDash \boldsymbol{\varphi}$ can be converted into convex and integer constraints.

Space and Time Tolerances in Task Planning

Space and time tolerances induce space and time robustness (robustness degrees)

Space tolerance

- $\tau_{space}(s,(f(s)<d),t) = d - f(s_t)$

Time tolerance

- $\theta^-(s,f(s),t) = \max(d \geq 0 \text{ s.t.} \forall t' \in [t-d,t], (s,t) \models \varphi \Leftrightarrow (s,t') \models \varphi)$
- $\theta^+(s,f(s),t) = \max(d \geq 0 \text{ s.t.} \forall t' \in [t,t+d], (s,t) \models \varphi \Leftrightarrow (s,t') \models \varphi)$

Optimization problem

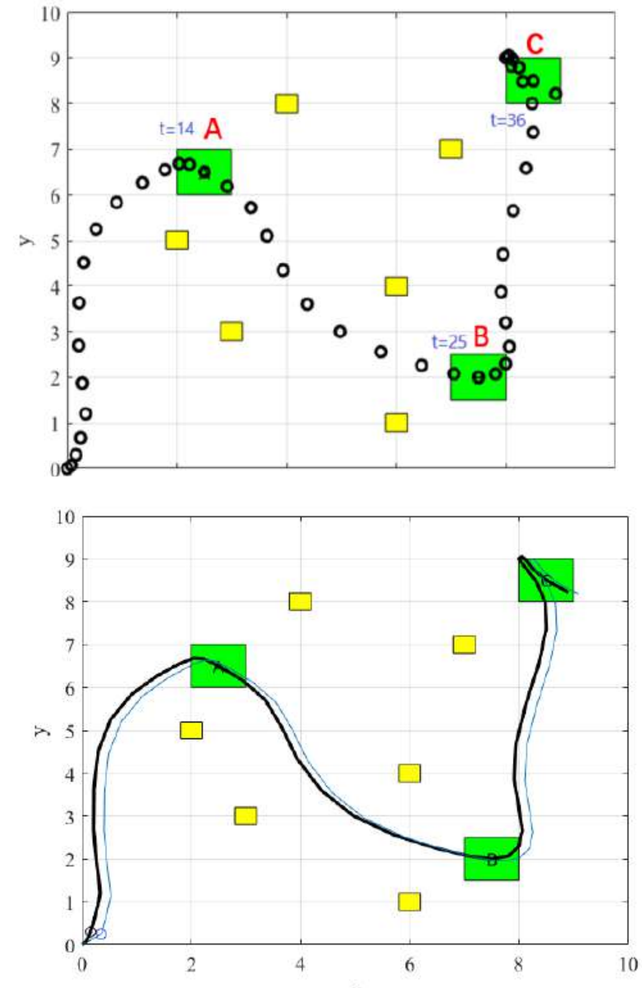
$$\begin{aligned} \max_{\boldsymbol{X}(t),u(t)} \quad & \lambda_1 r^{\varphi}_{time}(\boldsymbol{X}_{t_0}) + \lambda_2 r^{\varphi}_{space}(\boldsymbol{X}_{t_0}) \\ \textit{subject to} \quad & \boldsymbol{X}(t+1) = f(\boldsymbol{X}(t),u(t)) \\ & u_{min} \leq u(t) \leq u_{max} \\ & \boldsymbol{X}_{t_0} \models \varphi \end{aligned}$$

$$\varphi_1 = \Diamond_{[10,20]}A \wedge \Diamond_{[21,31]}B \wedge \Diamond_{[32,42]}C \wedge (\bigwedge_{i=1,\cdots,k} \Box\neg O_i)$$

=> Transformed into MIL constraints

Signal Temporal Logic (STL) specs, transformed to "convex" constraints or unions of such

Robot path that maximizes space-time tolerances under noise free environment



Lin and Baras, 2020 IFAC World Congress

Self-Monitoring and Self-Corrections

- Given reference inputs obtained in planning phase: U_r

- We are constantly evaluating whether the predicted trajectory $\mathbf{X}_p$ still satisfies the given specification and maintains a specific tolerance degree

$$X_p(\tau+1) = f(X_p(\tau), U_r(\tau)), \quad \tau = t', \cdots N-1$$
$$X_p(\tau) = X_o(\tau) \quad \text{for} \quad \tau = 1, \cdots, t'$$

- $\mathbf{X}_p$ is made of the trajectory we have observed so far $\mathbf{X}_o$ and the resulting future trajectory with reference input.

Event-triggered MPC

- An event-triggered MPC is designed for runtime self-correction

- MPC will be triggered if

$$r_{space}(\mathbf{X}_t^p) < \theta_{space} \quad \text{or} \quad r_{time}(\mathbf{X}_t^p) < \theta_{time}$$

- MPC: solve the optimization problem with a horizon T, and only apply the first control input

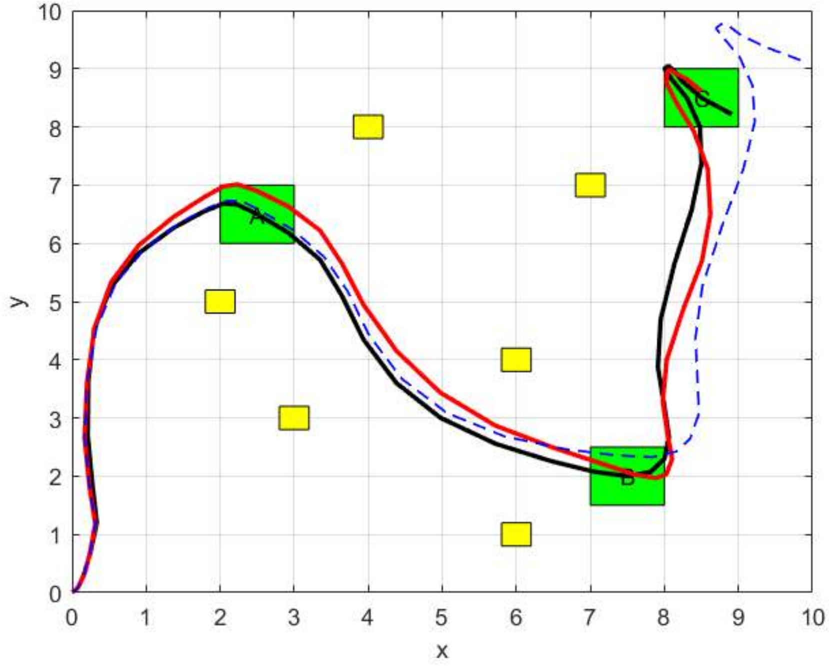
$$\min_{\mathbf{X}(t),u(t)} \sum_{\tau=t}^{\tau=t+T} (X_r(\tau) - \mathbf{X}(\tau))^T Q (X_r(\tau) - \mathbf{X}(\tau))$$

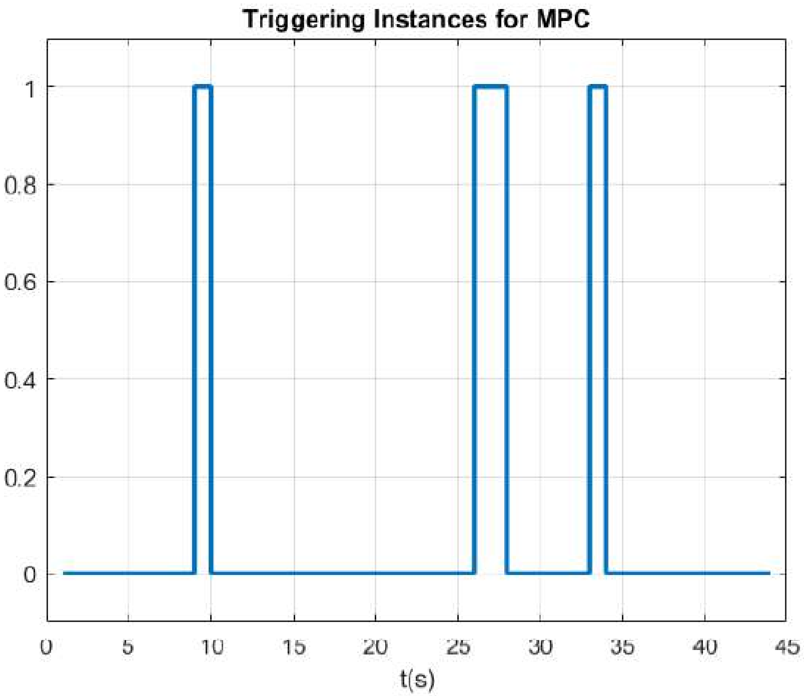
$$\text{subject to} \quad \mathbf{X}(\tau+1) = f(\mathbf{X}(\tau), u(\tau)), \tau \in [t, t+T-1]$$

$$\mathbf{X}(t+T) = X_r(t+T) \text{ , } u_{min} \le u(t) \le u_{max}$$

Case Studies

$$\varphi_1 = \Diamond_{[10,20]} A \wedge \Diamond_{[21,31]} B \wedge \Diamond_{[32,42]} C \wedge \left(\bigwedge_{i=1,\cdots,k} \Box \neg O_i \right)$$





Space and Time Tolerances in Task Planning: Self Monitoring and Self-Corrections

- Actual path may deviate from planning due to noise.
- An **event-triggered MPC** is designed for runtime self-correction.
- Given reference inputs obtained in planning phase: U_r
- We are constantly evaluating whether the predicted trajectory $\mathbf{X}_p$ still satisfies the given specification and maintains a specific tolerance degree
- MPC will be triggered if $r_{space}(\mathbf{X}_t^p) < \theta_{space}$ or $r_{time}(\mathbf{X}_t^p) < \theta_{time}$

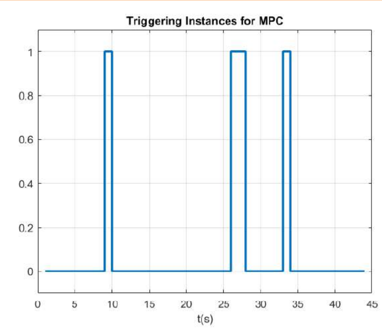
$$\varphi_1 = \Diamond_{[10,20]} A \wedge \Diamond_{[21,31]} B \wedge \Diamond_{[32,42]} C \wedge \left(\bigwedge_{i=1,\cdots,k} \Box \neg O_i \right)$$

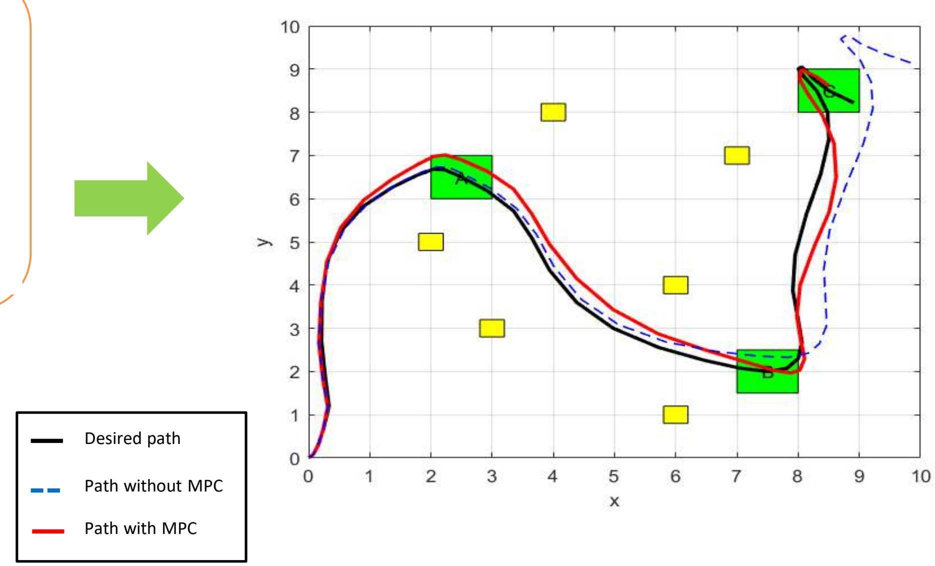
Optimization problem with a horizon T

$$\min_{\mathbf{X}(t),u(t)} \sum_{\tau=t}^{\tau=t+T} (X_r(\tau) - \mathbf{X}(\tau))^T Q (X_r(\tau) - \mathbf{X}(\tau))$$

$$\text{subject to} \quad \mathbf{X}(\tau+1) = f(\mathbf{X}(\tau), u(\tau)), \tau \in [t, t+T-1]$$

$$\mathbf{X}(t+T) = X_r(t+T) \quad , u_{min} \leq u(t) \leq u_{max}$$

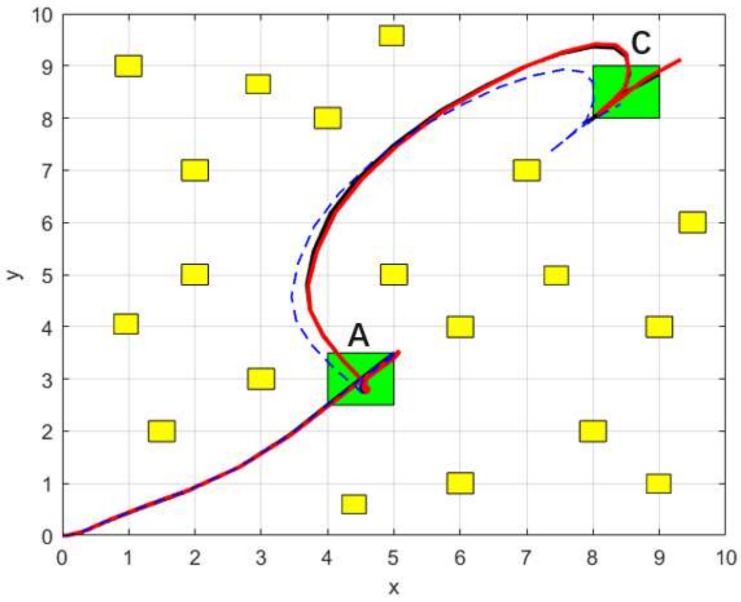


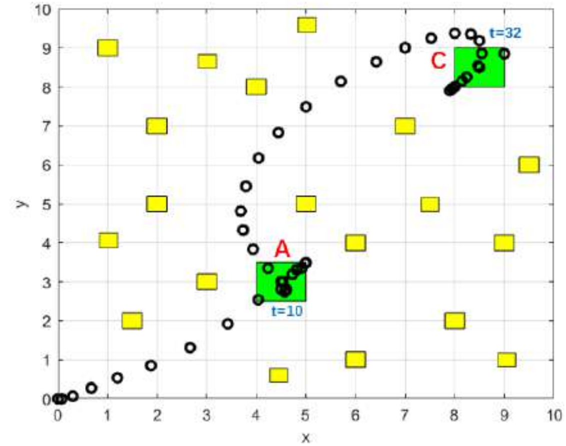


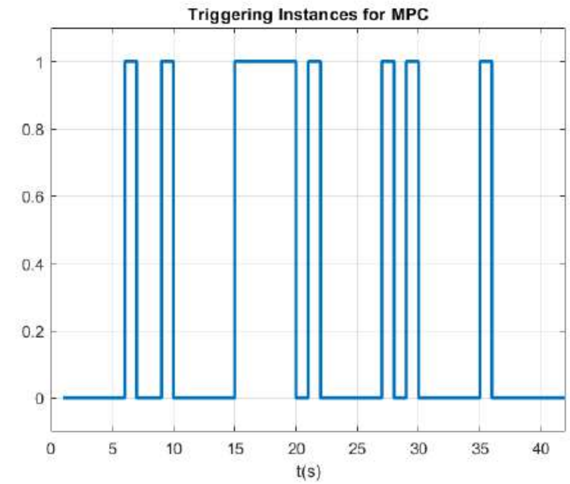
Lin and Baras, 2020 IFAC World Congress

Space and Time Tolerances in Task Planning: Self Monitoring and Self-Corrections

$$\varphi_2 = \Diamond_{[10,20]} A \wedge \Diamond_{[32,42]} C \wedge \left(\bigwedge_{i=1,\cdots,k_2} \Box \neg O_i \right)$$



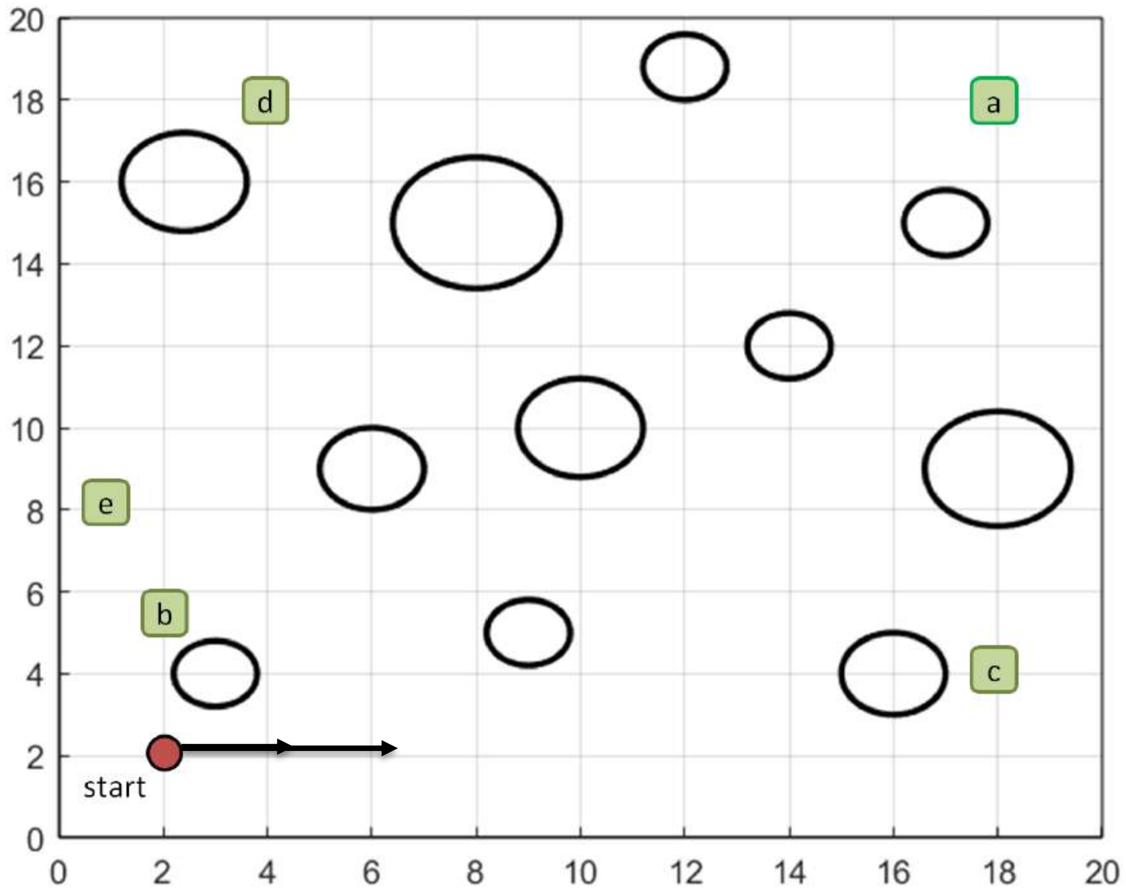




Lin and Baras, 2020 IFAC World Congress

Reinforcement learning with complicated tasks under finite time constraints

Task Workspace



States: 20 by 20 grid

Actions: move **up, down, left**, or **right** at speed of **1** or **2**

Temporal logic to automaton

$$\phi = \underbrace{((\neg d\,\mathcal{U}\,e) \rightarrow \Diamond_{[8,15]}d)}_{\text{Task 1}} \vee \underbrace{(\neg a\,\mathcal{U}\,(b \rightarrow \Diamond_{[5,10]}c) \wedge \Diamond a)}_{\text{Task 2}} \wedge (\Box\neg obs)$$

The robot can accomplish the task by achieving **either one** of the objectives:

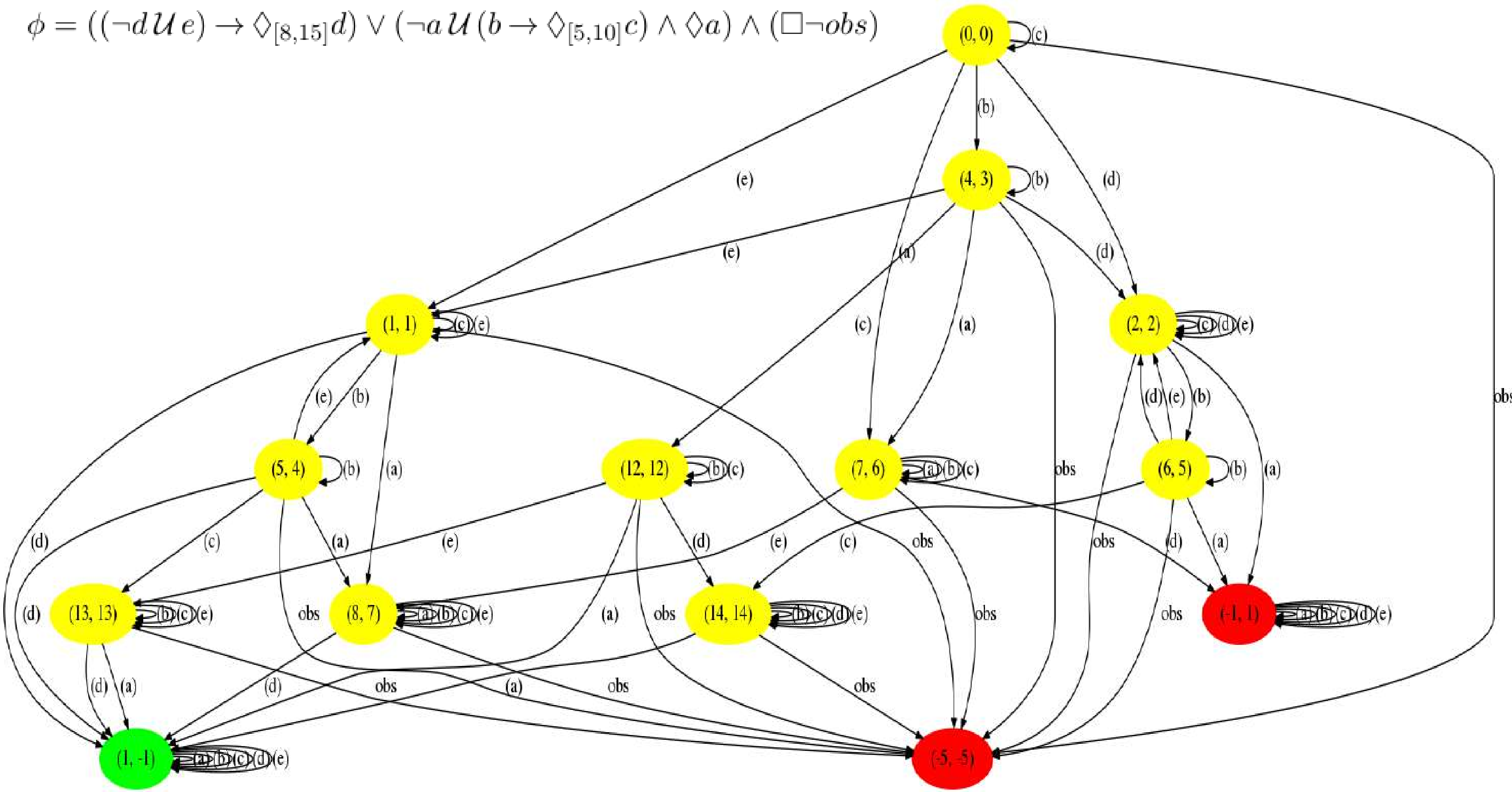
(1) do not visit position ***d*** until ***e*** has been visited, then once position ***e*** has been visited, eventually return to position ***d*** between **8 and 15 time units**.

(2) do not visit position ***a*** until ***b*** has been visited, and after visiting ***b***, the robot has to immediately visit position ***c*** between **5 and 10 time units**, and eventually return to position ***a***

Translate specification into LTL3 monitor automaton without considering time constraints first!

Temporal logic to automaton

$$\phi = ((\neg d\,\mathcal{U}\,e) \rightarrow \Diamond_{[8,15]} d) \vee (\neg a\,\mathcal{U}\,(b \rightarrow \Diamond_{[5,10]} c) \wedge \Diamond a) \wedge (\Box \neg obs)$$



[1] LTL3 tool: http://ltl3tools.sourceforge.net/

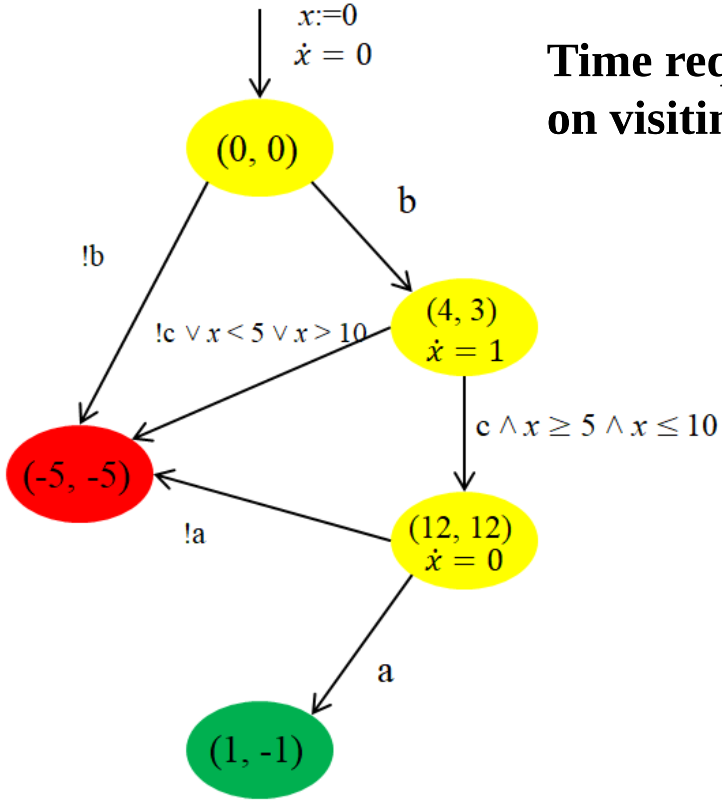
Temporal logic to automaton

1. Hard to design reward functions.

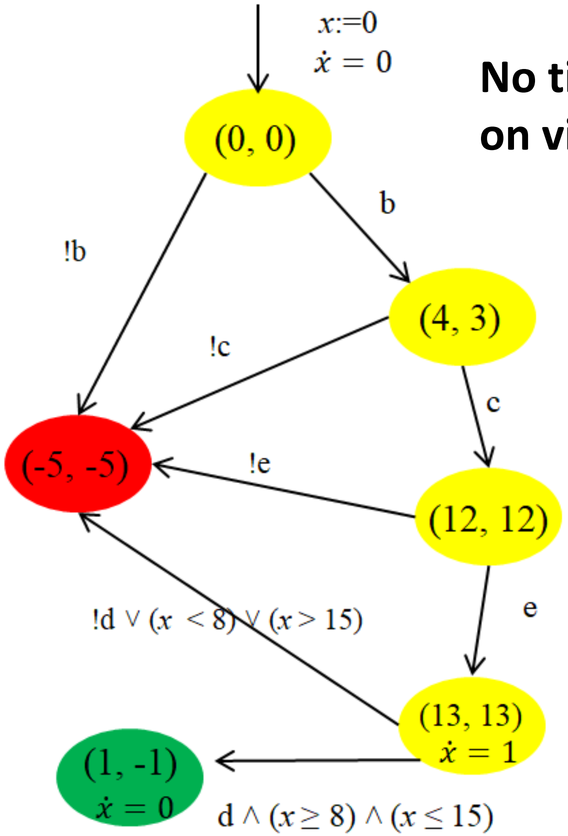
2. Hard to factor time into consideration.

3. Low Training efficiency.

Sub-task automaton – Transition Systems

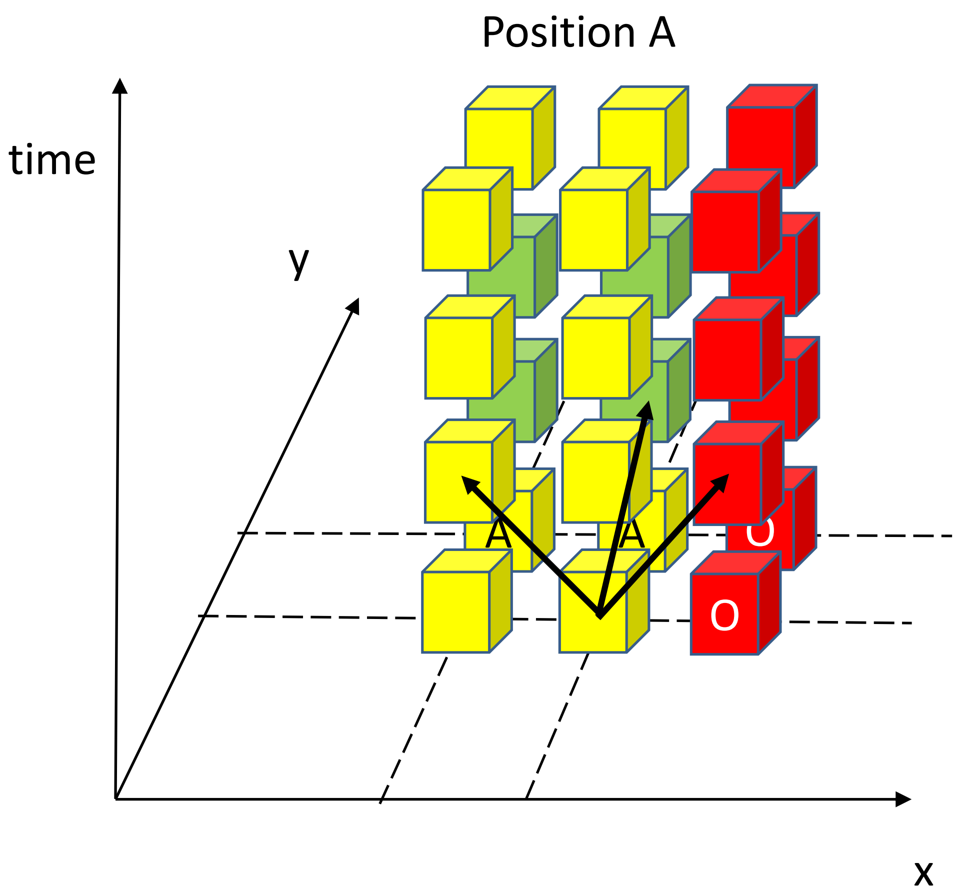


Time requirement on visiting c



No time requirement on visiting c

Reinforcement Learning in extended State Space



$$\phi = (\Diamond_{[2,3]} A) \wedge (\Box \neg O)$$

Becomes green states only between 2 seconds and 3 seconds.

State O is red for all time due to the always requirement.

Extended State!

$$S^{ext} = S \times Q_i^{\varphi} \times V^{\varphi}$$

Reinforcement Learning in extended State Space

- $Q_i^n(S_i^{ext,n}, a^n) = (1-\alpha)\, Q_i^{n-1}(S_i^{ext,n}, a^n) + \alpha \cdot [R(S_i^{ext,n}, a^n) + \gamma \max_a Q_i^{n-1}(S_i^{ext,n+1}, a)]$

Learning rate

Reward function

Estimate of optimal future value

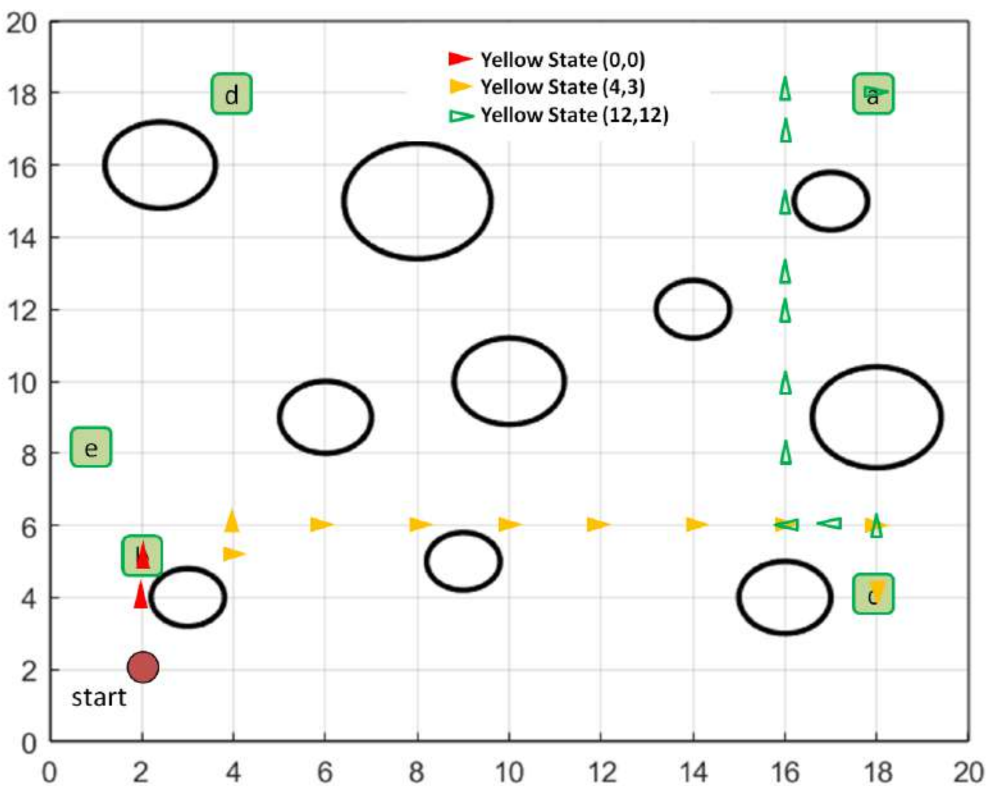
- Reward function based on sub-task automaton progression:

$$R_{s^\otimes, a} = \begin{cases} r_p & if \quad [\alpha \models \phi] \neq \perp \quad and \quad d(s) > d(s') \\ r_n & if \quad [\alpha \models \phi] = \perp \quad and \quad d(s') = \infty \\ r_s = 0 & if \quad [\alpha \models \phi] \neq \perp \quad and \quad d(s) = d(s') \end{cases}$$

- r_p: Positive reward if the next state s' has a better progression (smaller *d* value)
- r_n: Negative reward if the next state s' is a bad state
- r_s: Neutral reward if the next state s' has the same progression as current state s

Case Study

$$\phi = ((\neg d\,\mathcal{U}\,e) \rightarrow \Diamond_{[8,15]}d) \vee (\neg a\,\mathcal{U}\,(b \rightarrow \Diamond_{[5,10]}c) \wedge \Diamond a) \wedge (\Box\neg obs)$$

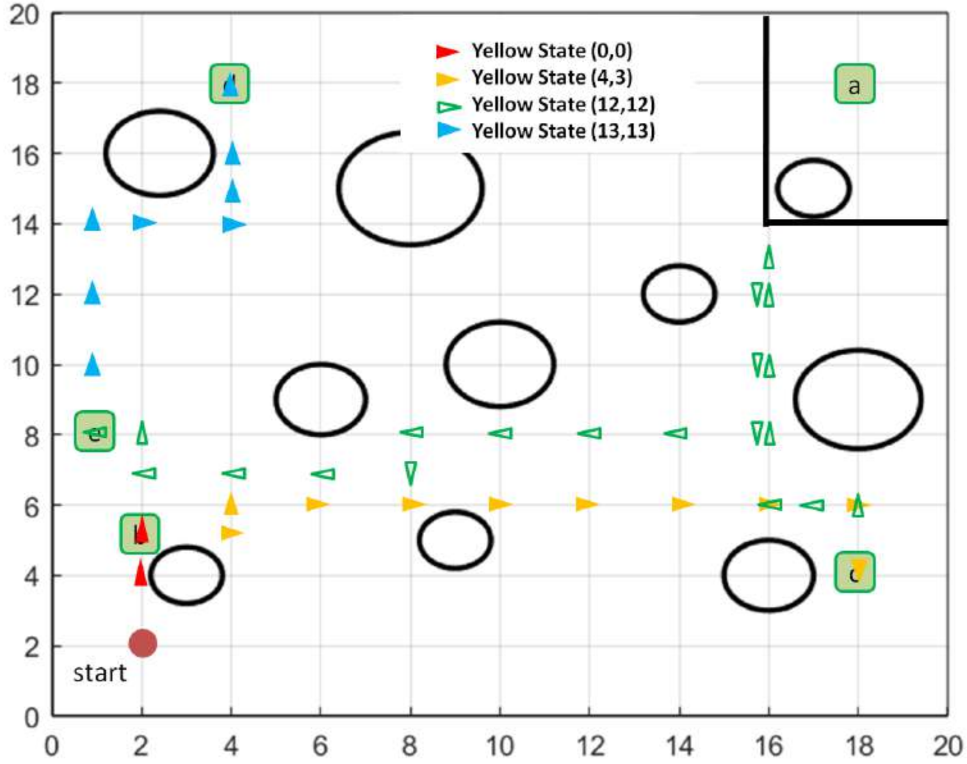


1. Path followed by b->c->a

2. Task accomplished

Case Study

$$\phi = ((\neg d\,\mathcal{U}\,e) \rightarrow \Diamond_{[8,15]} d) \vee (\neg a\,\mathcal{U}\,(b \rightarrow \Diamond_{[5,10]} c) \wedge \Diamond a) \wedge (\Box \neg obs)$$

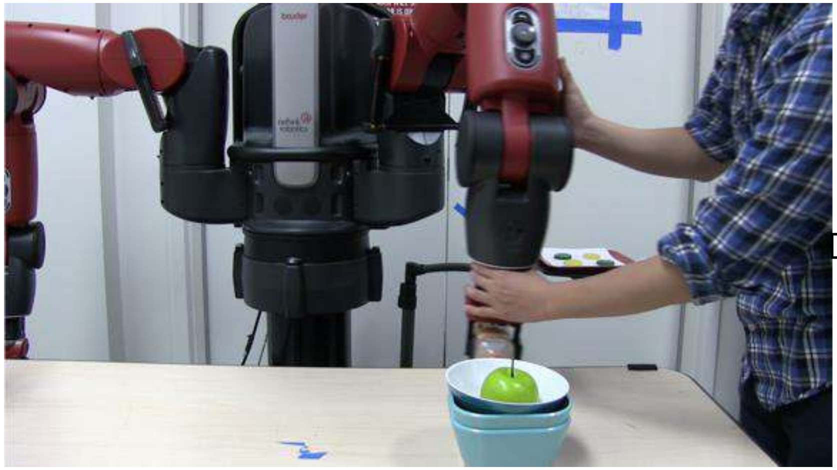


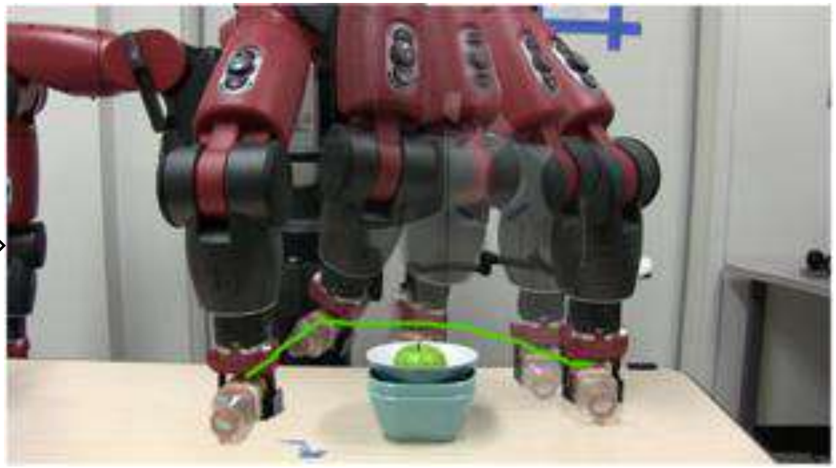
1. Path followed by b→c[5,10]→a initially
2. a is not reachable by following the plan
3. Now follow b→c→e→d[8,15]
4. Task accomplished

Teaching Robots Manipulation Tasks

Example Task: Transferring

- New preferences/objectives
  - Avoid bowl *above* it OR *around* it? bottle or knife?
  - Adjust the objective of adapted trajectory
  - Learn preferences to adapt movement in new situation

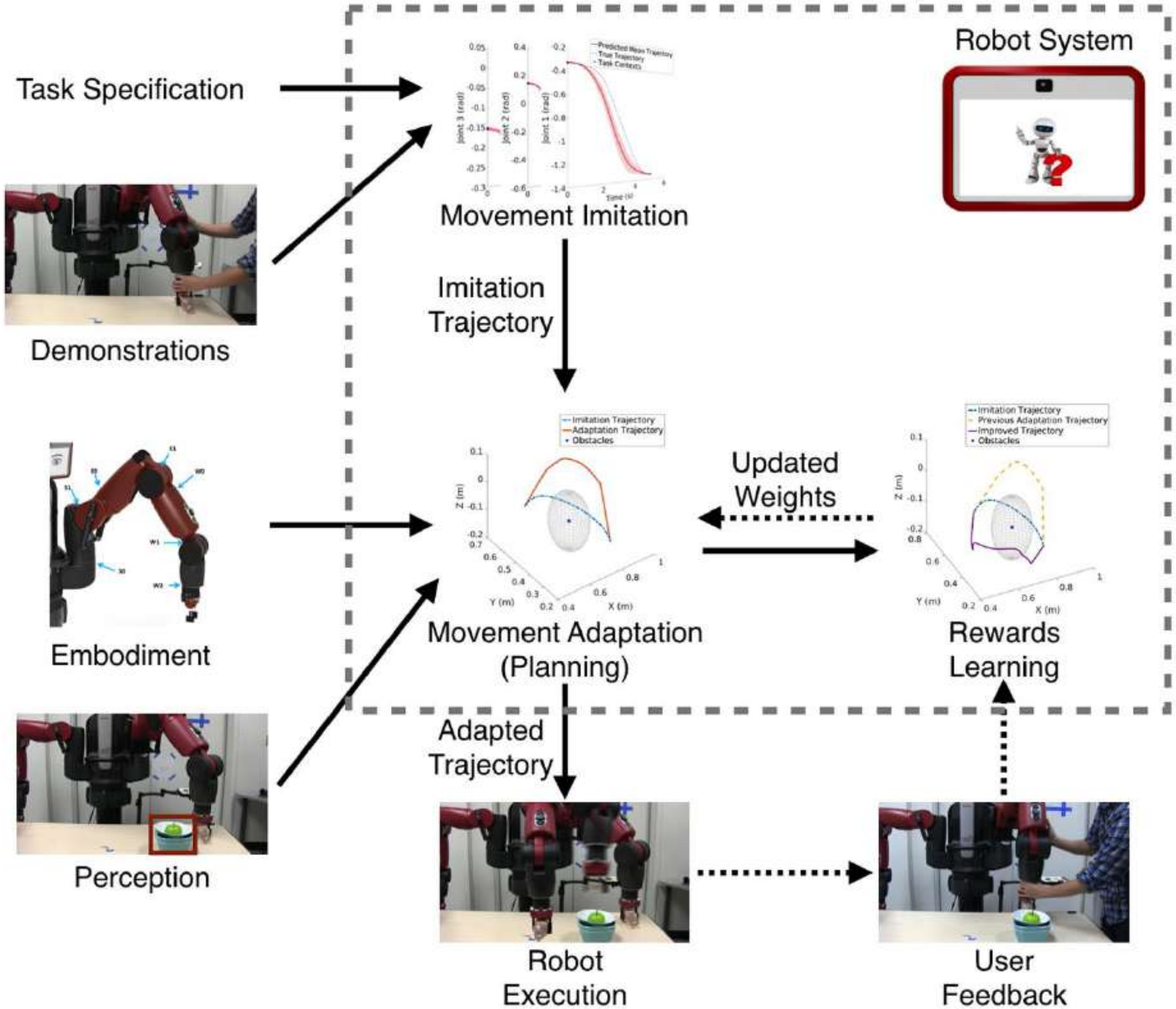




Feedback Demonstrations

Movement Adaptation after learning

Teaching Manipulation Tasks: Our System



Example: Opening Microwave

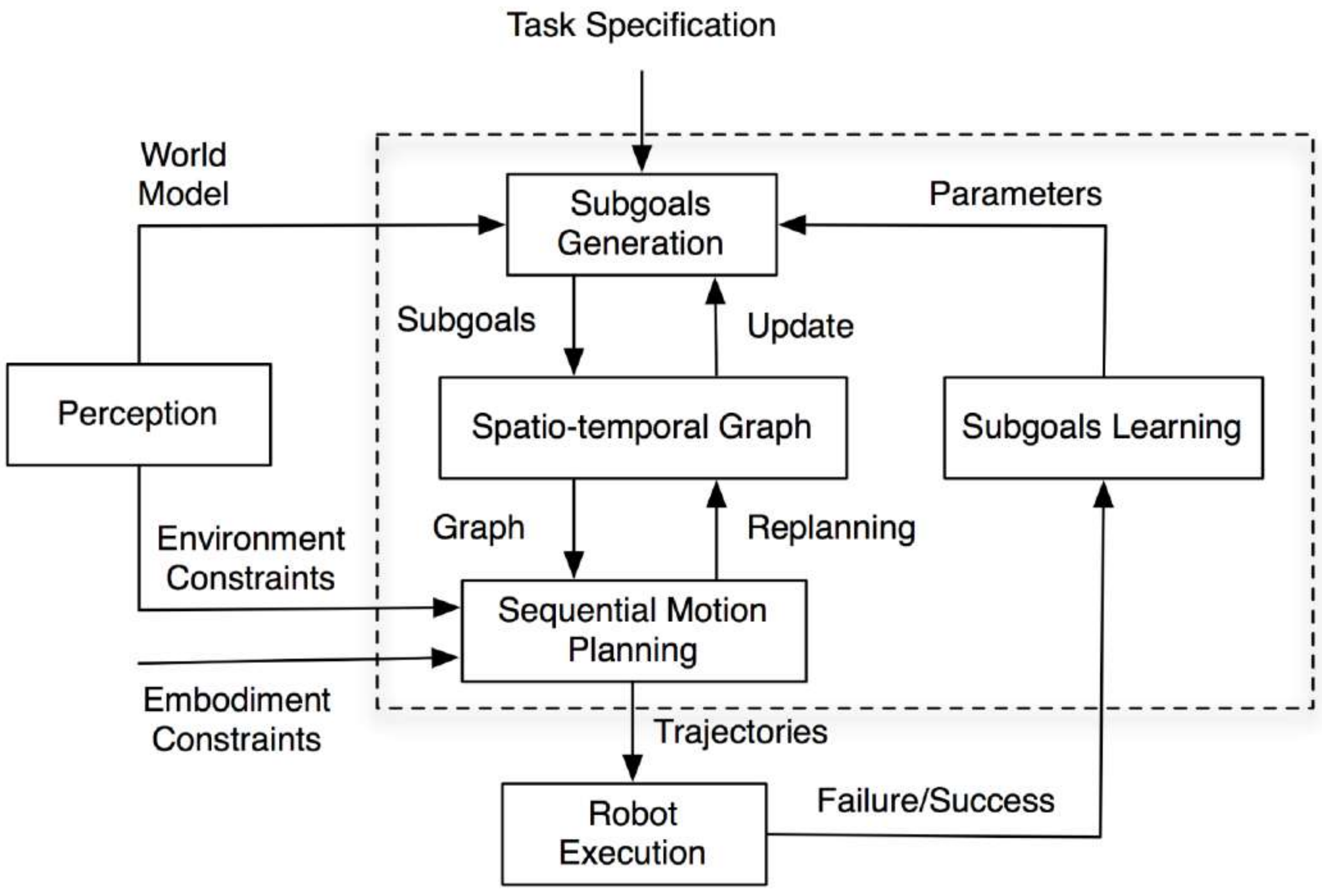
- Given sequential task with three primitives

Reaching Grasping   Pulling Opening   Inserting Opening

Subgoal:grasping location   Subgoal:pulling angle   Subgoal:inserting angle

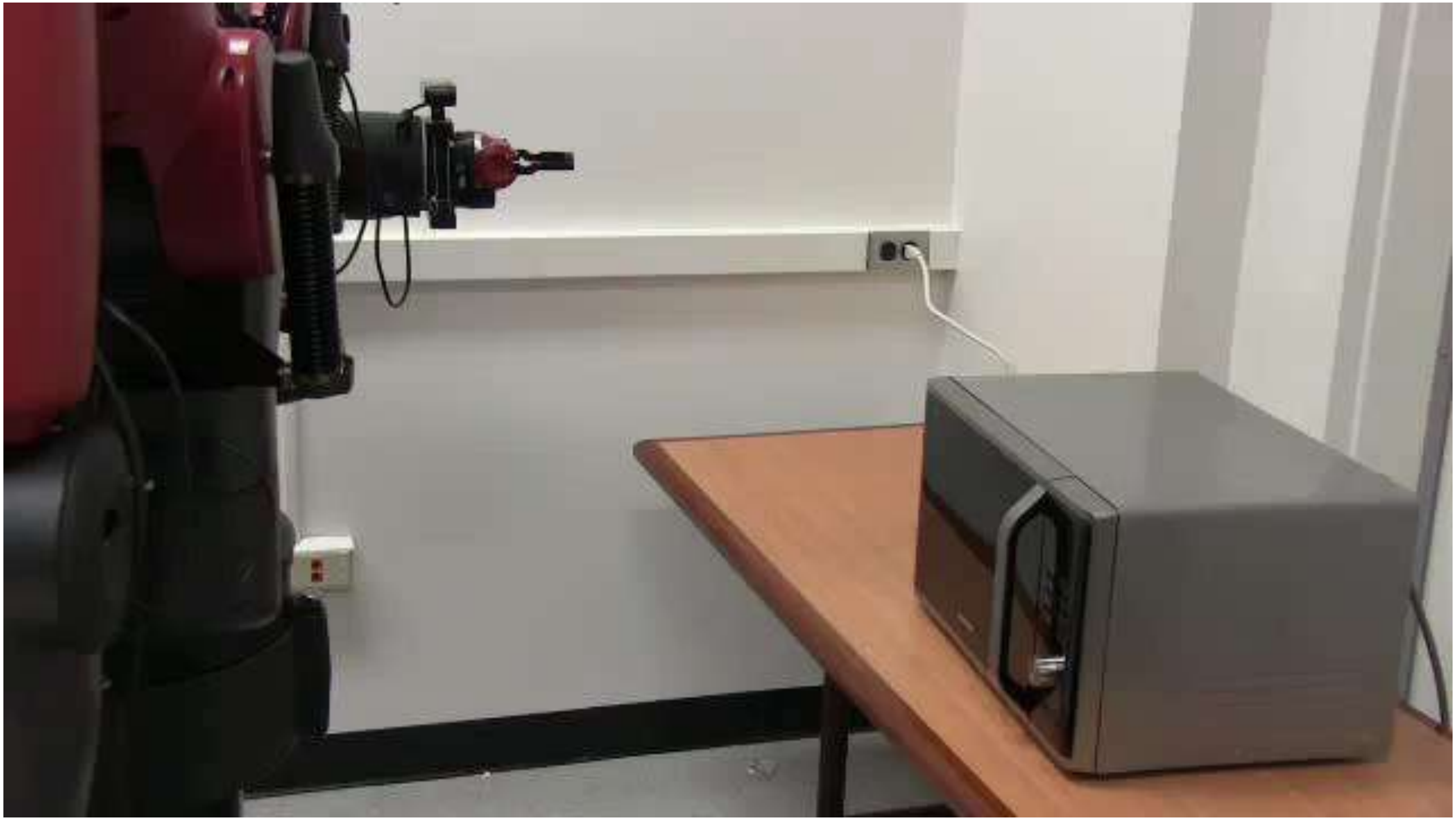
- Motion planning
  – Subgoals generation and selection
  – Motion planning for each primitive action

Approach



Block diagram of dynamic motion planning for sequential task with subgoals learning

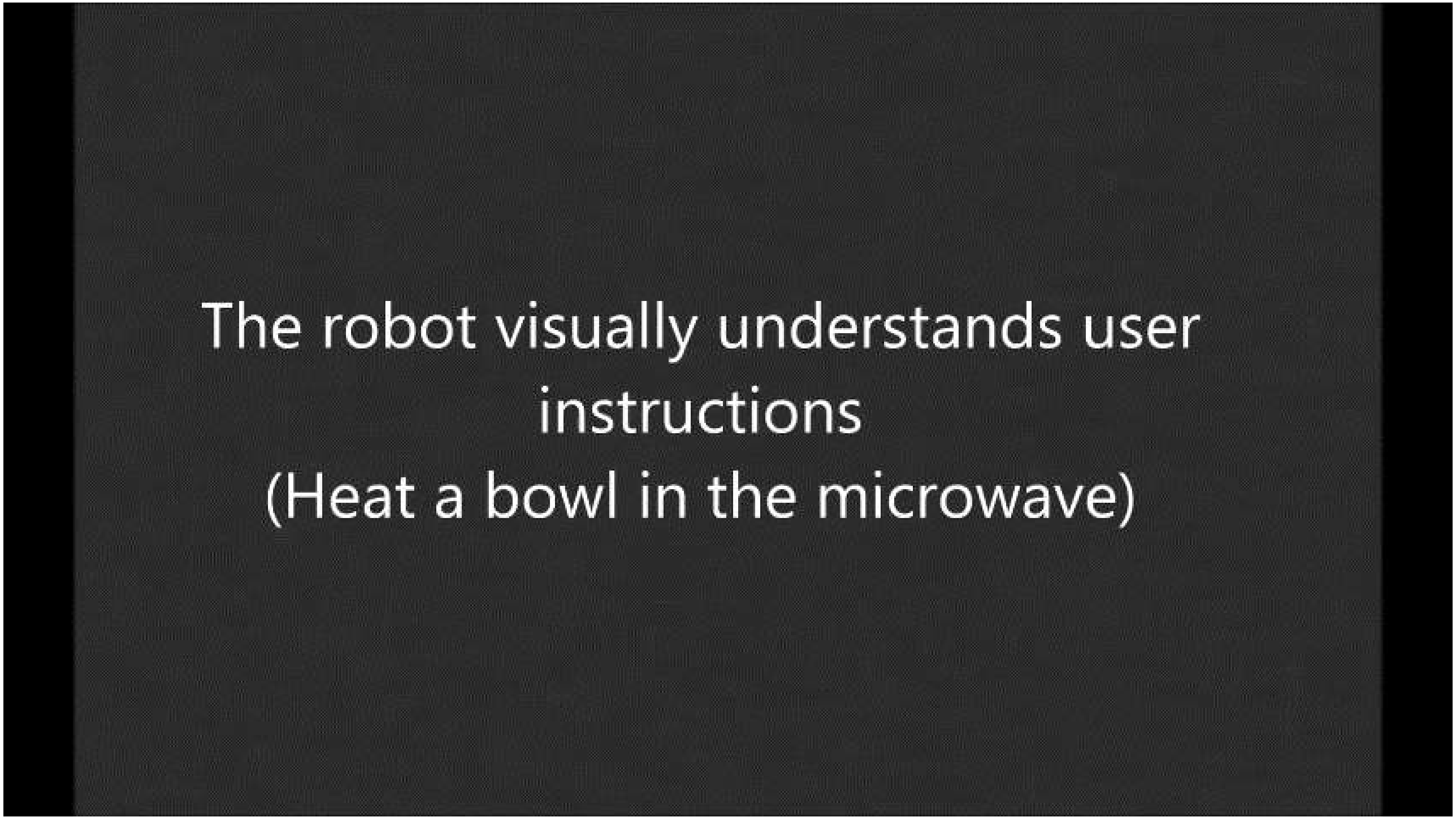
Opening Microwave



Attention, Object Detection, Recognition

- Large demands of robots automation
- Fundamental task: object detection & recognition

Microwave Operation



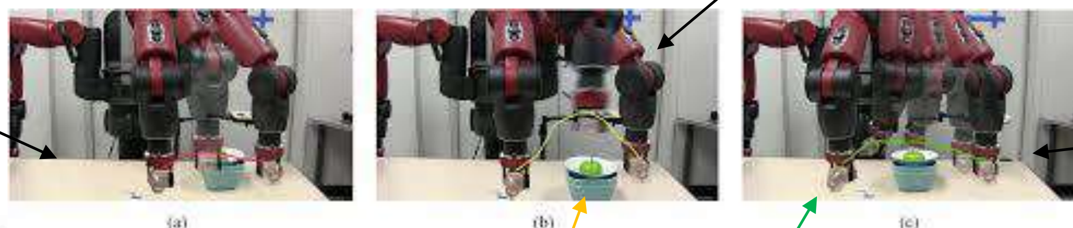
AI and ML – our Approach

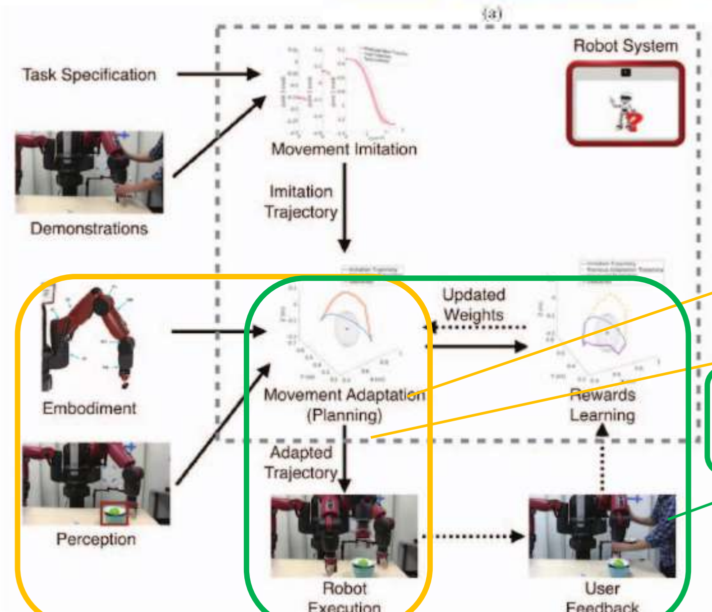
Applications in Human-Robot Collaboration
Learning by demonstration with spatio-temporal constraints

Cost-oriented obstacle avoidance

Learned Task

Human-Oriented Obstacle avoidance

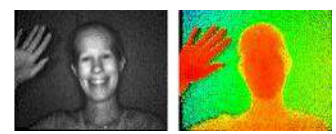


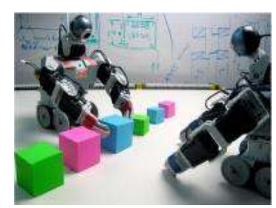


Perception: Object Detection

Self-Assessment: Ask Feedback from User

Task Generalization: Invariant Features





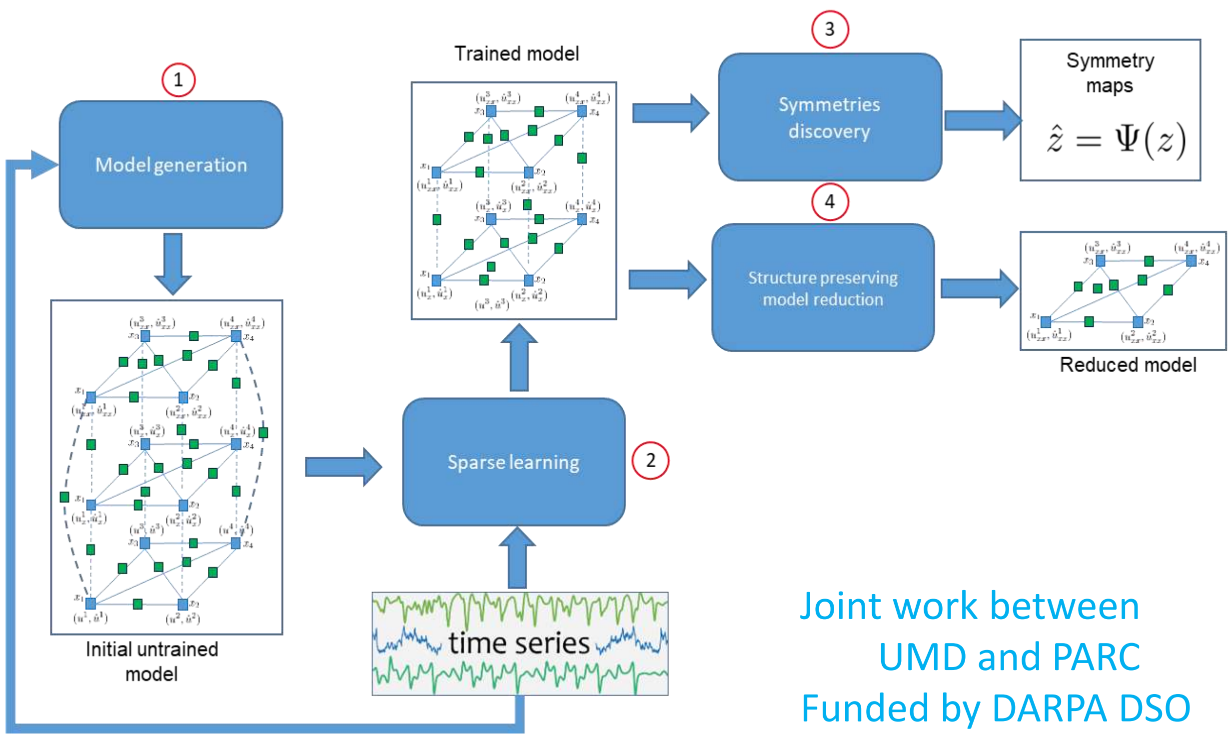
Baras et. al., Co-active learning to adapt Humanoid Movement for manipulation

Swarms and Collectives

- Learning **coordination laws** of biological and man-made swarms from observed trajectories
- Employ port-Hamiltonian formalism because it links concretely with mathematical physics methods used (symmetries, invariants, Noether's theorems) and because it can be used to model any multi-physics system

AI and ML can help in all steps of the work flow:

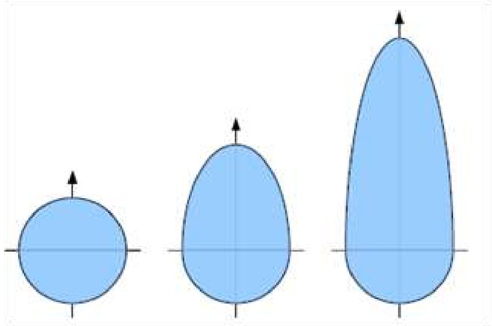
- Model generation
- Sparse learning
- Symmetry discovery
- Structure preserving model reduction
- Discovering and simplifying collaboration and communication laws in swarms



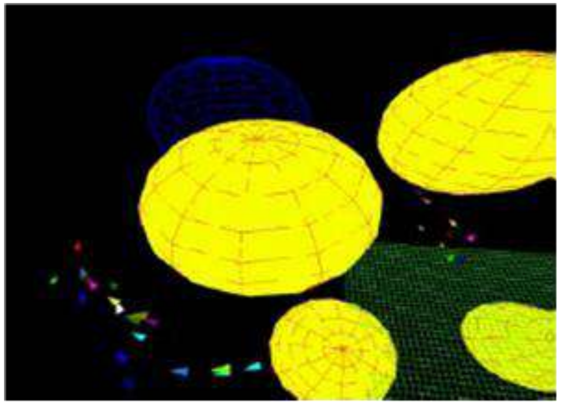
Joint work between UMD and PARC
Funded by DARPA DSO

Boids (Reynolds, http://www.red3d.com/cwr/boids/)

(a)



(b)

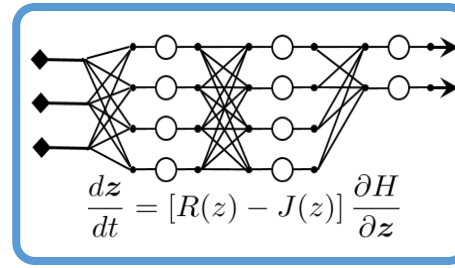


(c)

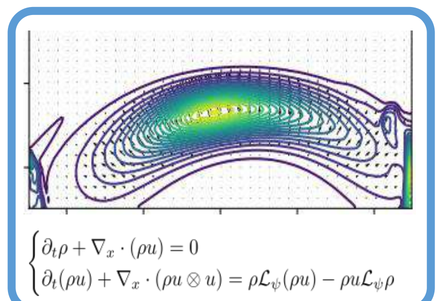
(a) Boid animation of birds in complex environments (http://nervo.tv/index.html?sect=5&proj=foxmovies); (b) ‘bubles’ of different shapes from slow to higher velocities; (c) diverse bubbles navigating obstacles to a goal

Learning Swarm Coordination Laws: Status

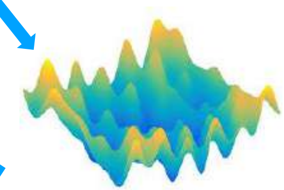
- Build a novel mathematical and software framework to learn coordination laws, of biological and man-made autonomous ensembles (swarms) from observed trajectories (robust to noise and missing data).
- Learning coordination laws of dynamics, symmetry maps, invariants, conservation laws and reduced models from real or simulated flock data
- Port-Hamiltonian modeling of particle (agent) interaction for robust representation and learning
- Methodology for homeogeneous and heterogeneous potentials, Boids models profiles, number of leaders, clusters of followers
- Learning Dynamic Leadership profiles, leaders, clusters of followers
- Developed fast. Accurate algorithms scalable to thousands of agents
- Macroscopic hydrodynamic PDE model (mean field) for learning coordination laws from density evolution



Microscale: Port-Hamiltonian Behavior Representation



Macroscale: Wave-like PDEs



Large-Scale Learning Framework

Learning Complex Flocking Coordination Laws

Mavridis, Tirumalai, Baras, 2020 IEEE CDC; DARPA Reports 2019-2020

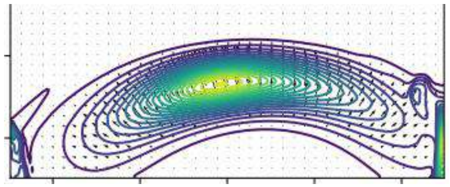
Learning Interaction Dynamics from Density Evolution

- **Learning Interaction Laws from Particle Trajectories**

$$\begin{cases} \dfrac{dx_i}{dt} &= v_i \\ \dfrac{dv_i}{dt} &= \frac{1}{N}\sum_{j\neq i}\left[\psi(x_i,x_j)(v_j - v_i) - \nabla_{x_i}U(x_i,x_j)\right] \end{cases}$$

*real-life data?

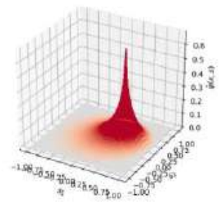
Density Evolution



$$\begin{cases} \partial_t\rho + \nabla_x\cdot(\rho u) = 0 \\ \partial_t(\rho u) + \nabla_x\cdot(\rho u\otimes u) = \rho\mathcal{L}_\psi(\rho u) - \rho u\mathcal{L}_\psi\rho - \rho\mathcal{L}_{(\nabla_x U)}\rho \end{cases}$$

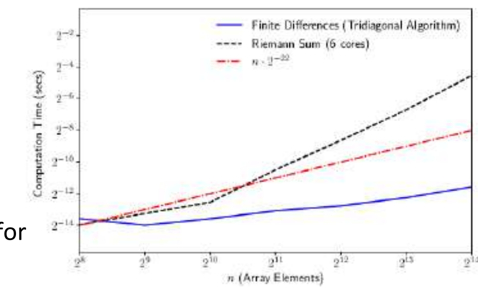
Problem: Requires Solution of Partial Integro-Differential Equations

Solution:



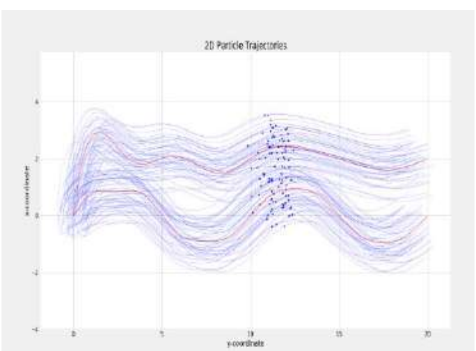
Interaction function = Green's function for

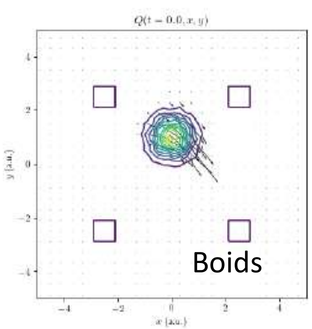
$$\mathcal{L}_x = -\frac{1}{2k}\left(\frac{\partial^2}{\partial x^2} - \lambda^2\right)$$

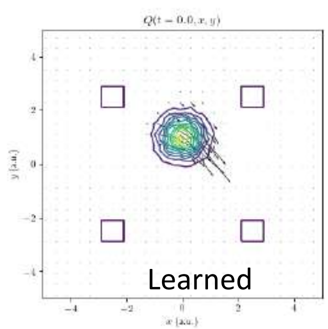


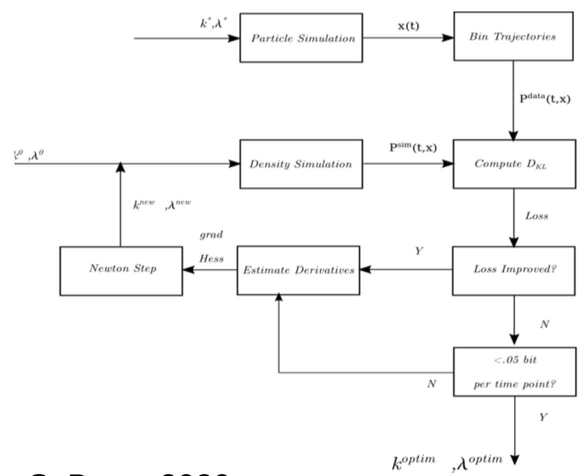
Computation Times for Nonlocal Terms

- **Simulation and Reconstruction of Complex Swarm Maneuvers** in 2D and 3D in both microscopic and macroscopic domains, capturing
- **Velocity Alignment** and **Spatial Cohesion**
- **Obstacle** and **Collision Avoidance**
- **Dynamic Leadership** and **Strong Wind effects**
- **Particle Dynamics** and the **Chorus-Line effect**





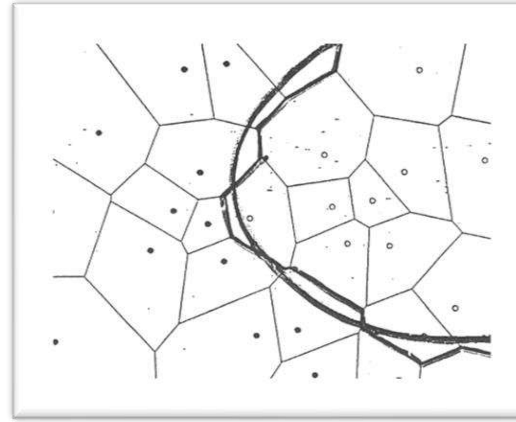




Mavridis, Tirumalai, Baras, 2020 IEEE CDC

Machine Learning and Artificial Intelligence: New Foundations

Learning Vector Quantization – A Competitive-Learning Neural Network



- Supervised Counterpart of VQ (Kohonen)
- Unsupervised Self-Organizing Map (SOM)
- **Online/Stochastic Algorithm;** Convergence ([Baras& LaVigna1991]) under convex metric
- **Interpretable, Robust, Data-Driven and Topology-Preserving**
- Sparse in the sense of memory complexity, fast to train and evaluate
- Consistent Classifier
- Widely used in time series, speech analysis and biomedical applications
- **Impressive robustness against adversarial attacks** [SHRV2019]

LVQ Algorithm

$$\begin{cases} \mu_i^{t+1} & = \mu_i^t - \alpha_i(t)\nabla_{\mu_i} d_\phi\left(X_{t+1}, \mu_i^t\right), \text{ if } c_{t+1} = c_{\mu_i}^t \\ \mu_i^{t+1} & = \mu_i^t + \alpha_i(t)\nabla_{\mu_i} d_\phi\left(X_{t+1}, \mu_i^t\right), \text{ if } c_{t+1} \neq c_{\mu_i}^t \end{cases}$$

→

Stochastic Approx. Algorithm

$$\mu_i^{t+1} = \mu_i^t + \alpha_i(t)\left[\theta_i^t(\mu) + M_i^{t+1}\right]$$

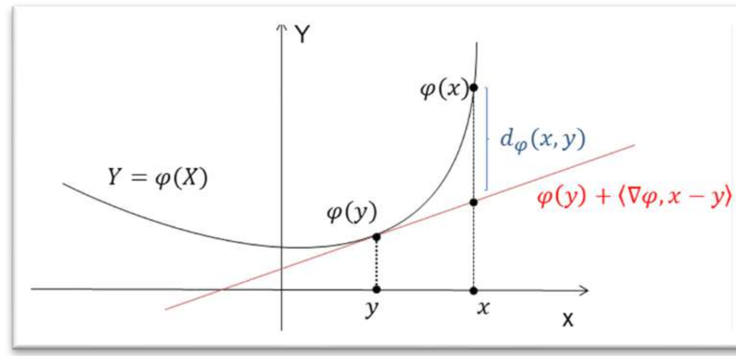
→

$$\dot{\mu} = \theta(\mu) = -\nabla_\mu J(\mu)$$

MB2020

[SHRV2019] Saralajew, Holdijk, Rees, Villmann, ArXiv, 2019; [MB2020] Mavridis and Baras. 2020 IFAC World Congress

Learning Vector Quantization with Bregman Divergences



- A novel dissimilarity measure in Bregman Divergences

$$d_\phi(x,\mu) = \phi(x) - \phi(\mu) - \frac{\partial\phi}{\partial\mu}(\mu)(x-\mu)$$

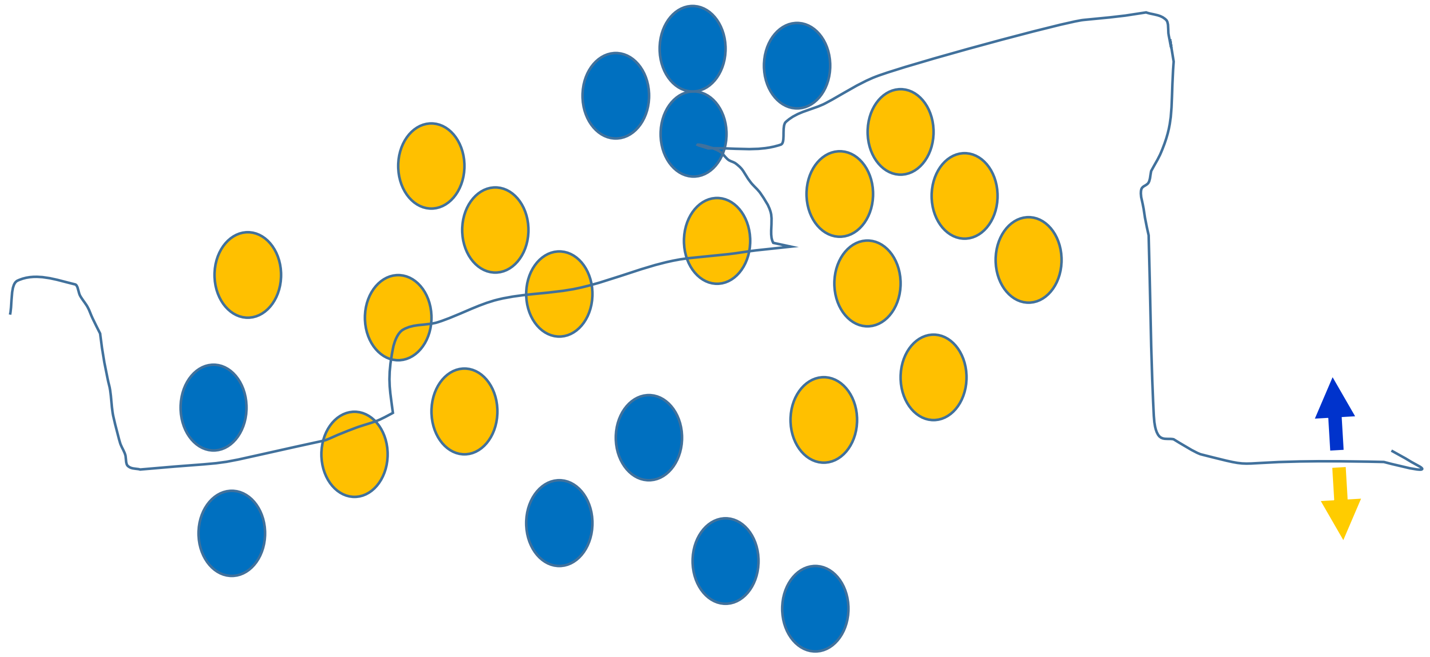
ϕ: strictly convex

- Euclidean Distance is a Bregman divergence
- **Kullback-Leibler divergence is a Bregman divergence; Unnormalized KL works as well**
- Can handle multidimensional numerical and Boolean inputs
- Correspondence with misclassification error probabilities (Hoeffding's inequality)
- Simplifies optimization steps in EM algorithms and improves efficiency of EM algorithms (LVQ, Soft-Clustering)
- **Convergence of LVQ with Bregman Divergences [MB2020]**
- Legendre duality with exponential family of densities

[MB2020] Mavridis and Baras. 2020 IFAC World Congress

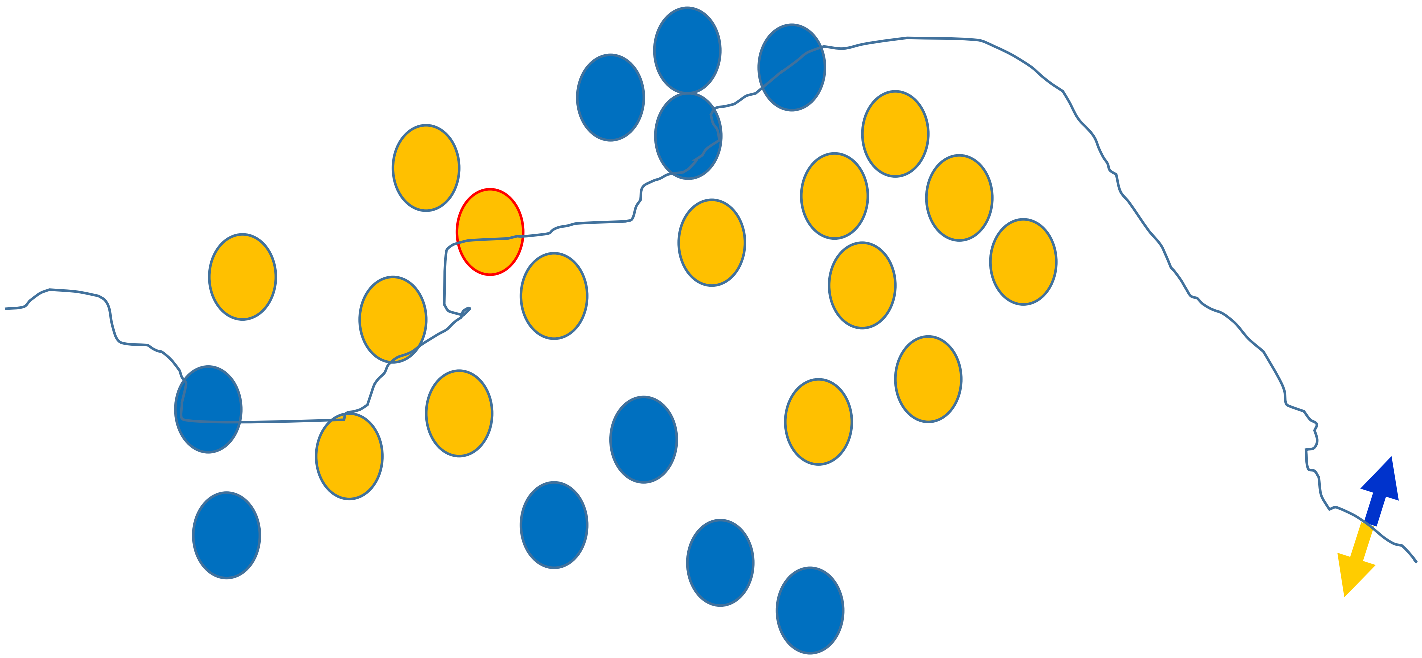
Decision Boundary

Initial random weights



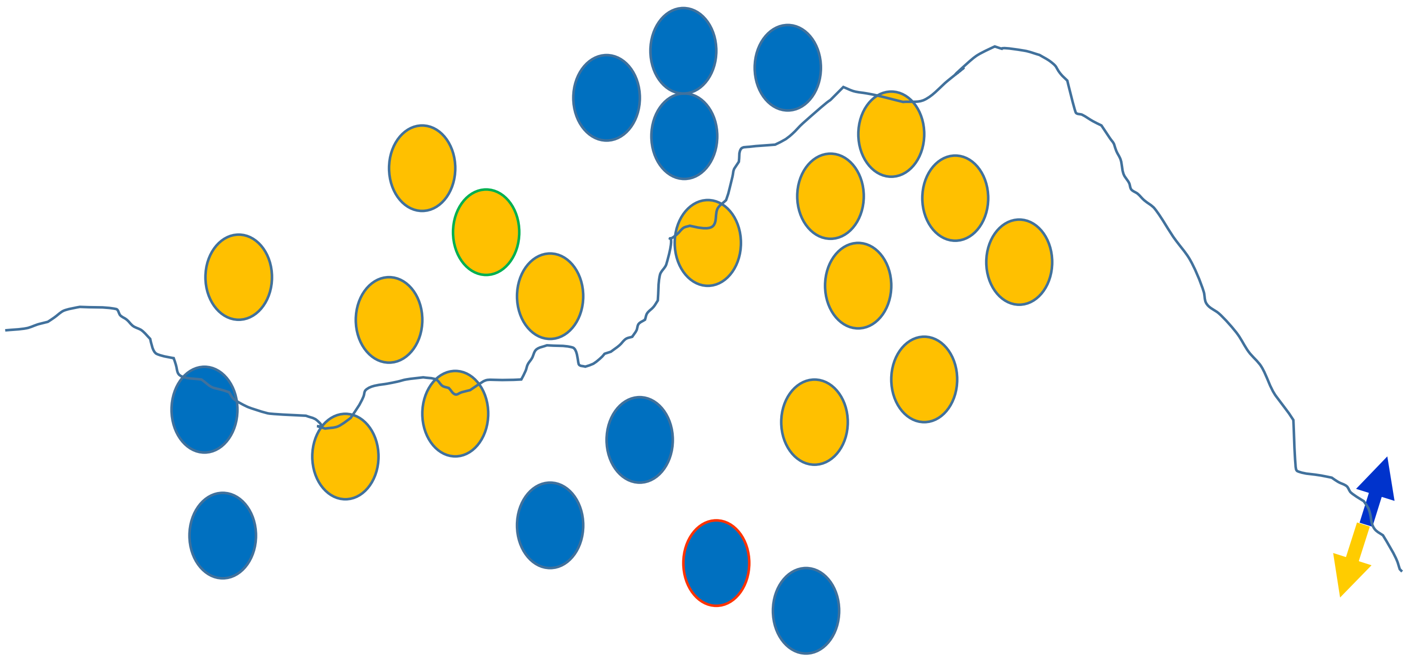
Decision Boundary (cont.)

Present a training instance / adjust the weights



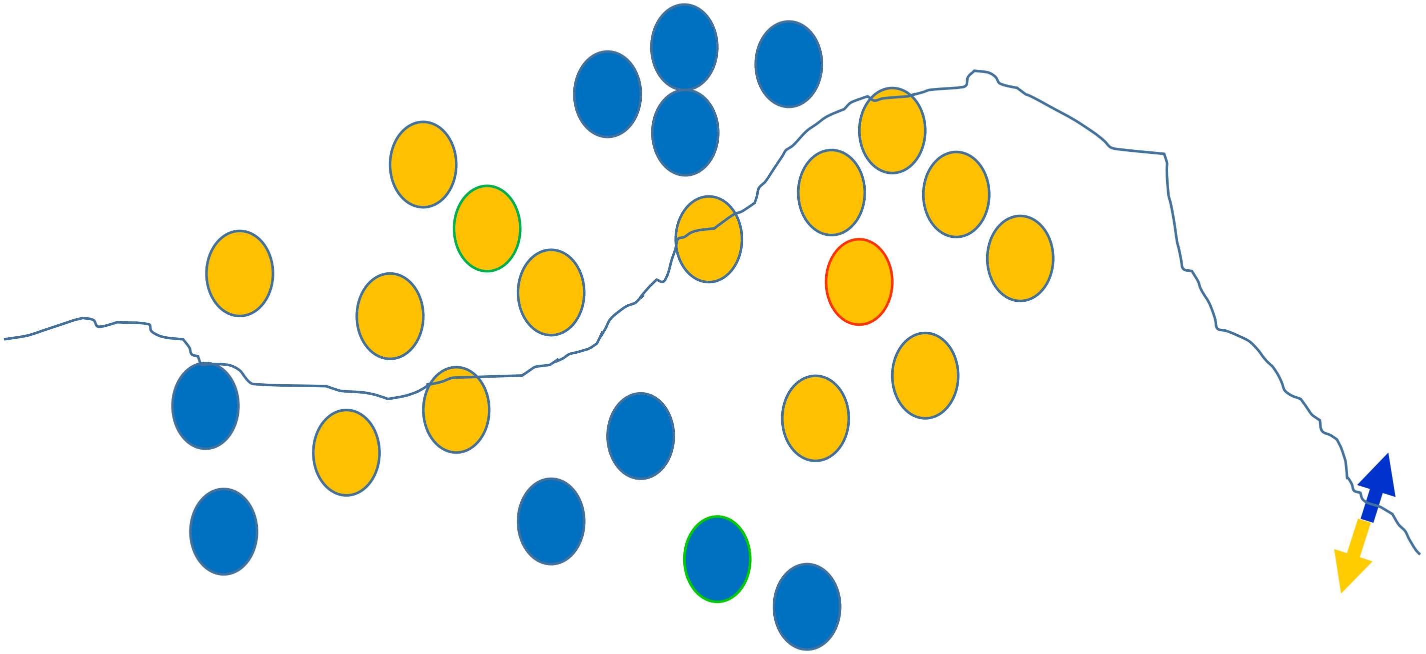
Decision Boundary (cont.)

Present a training instance / adjust the weights



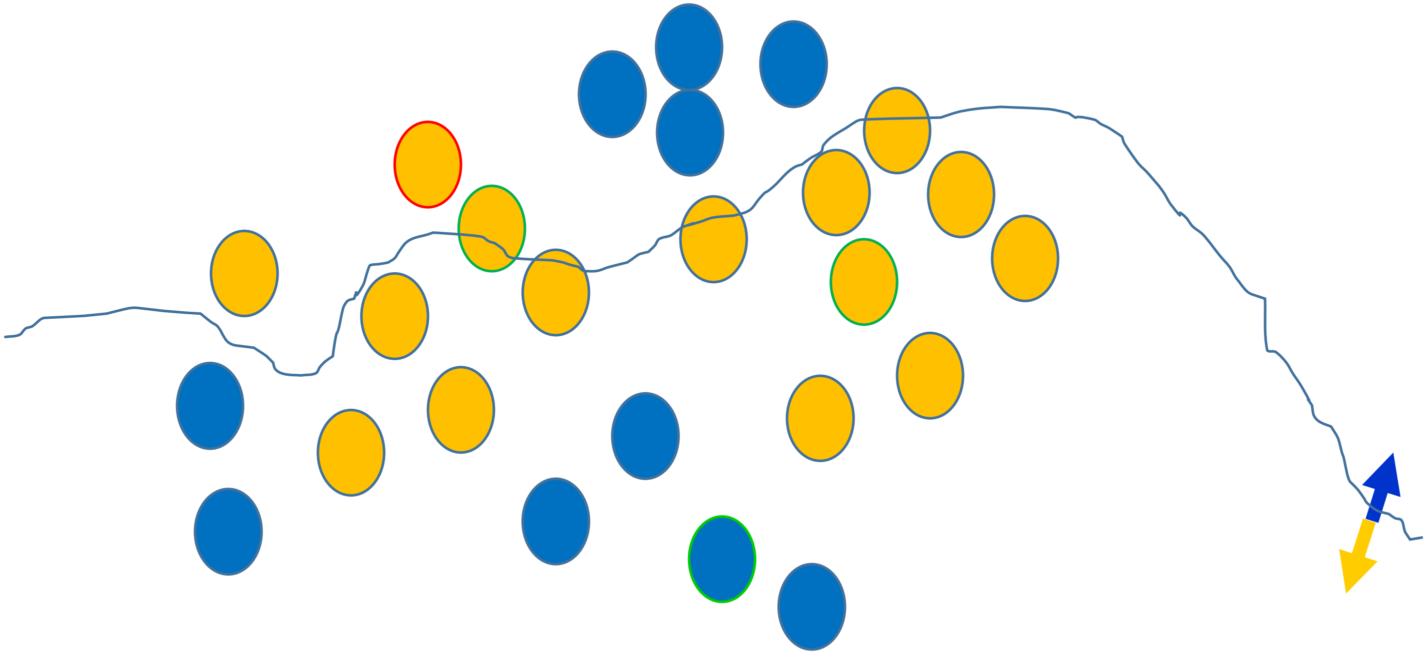
Decision Boundary (cont.)

Present a training instance / adjust the weights



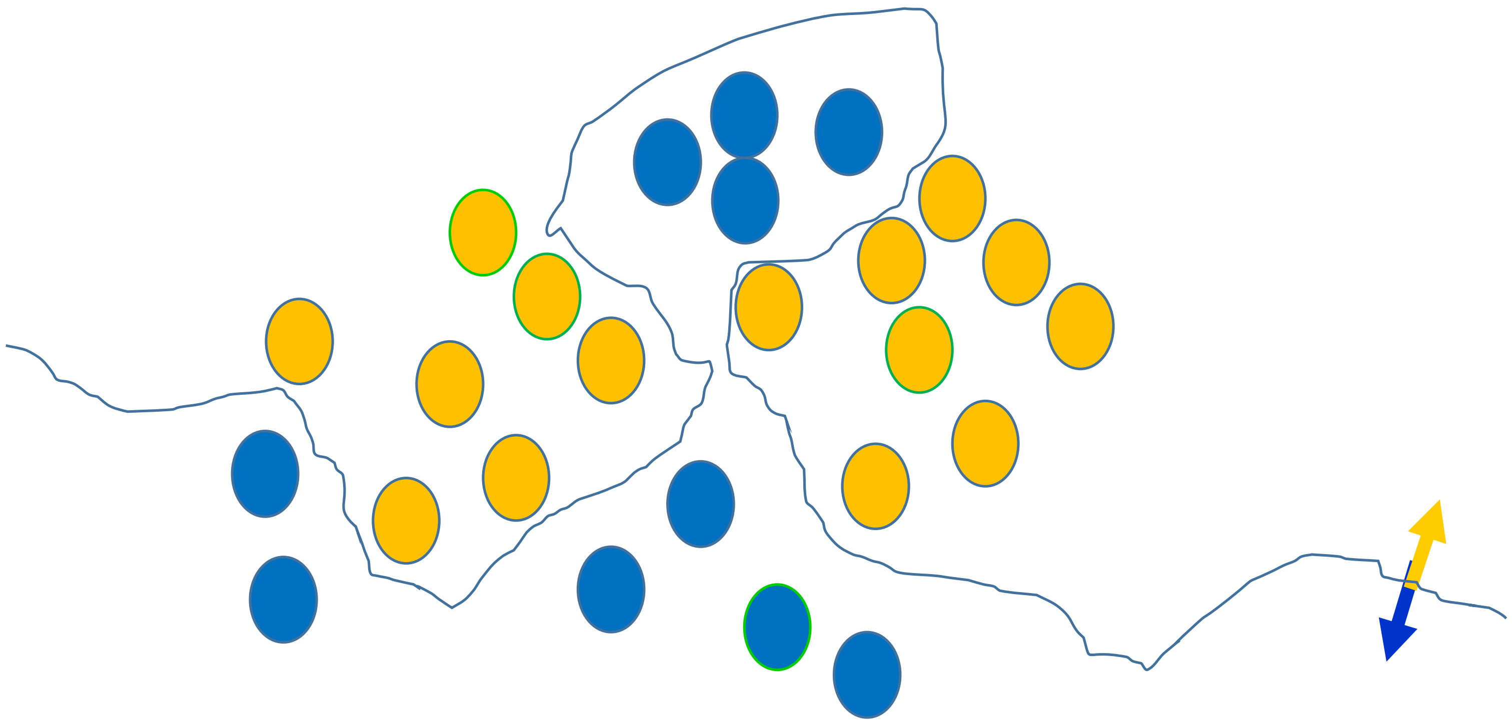
Decision Boundary (cont.)

Present a training instance / adjust the weights

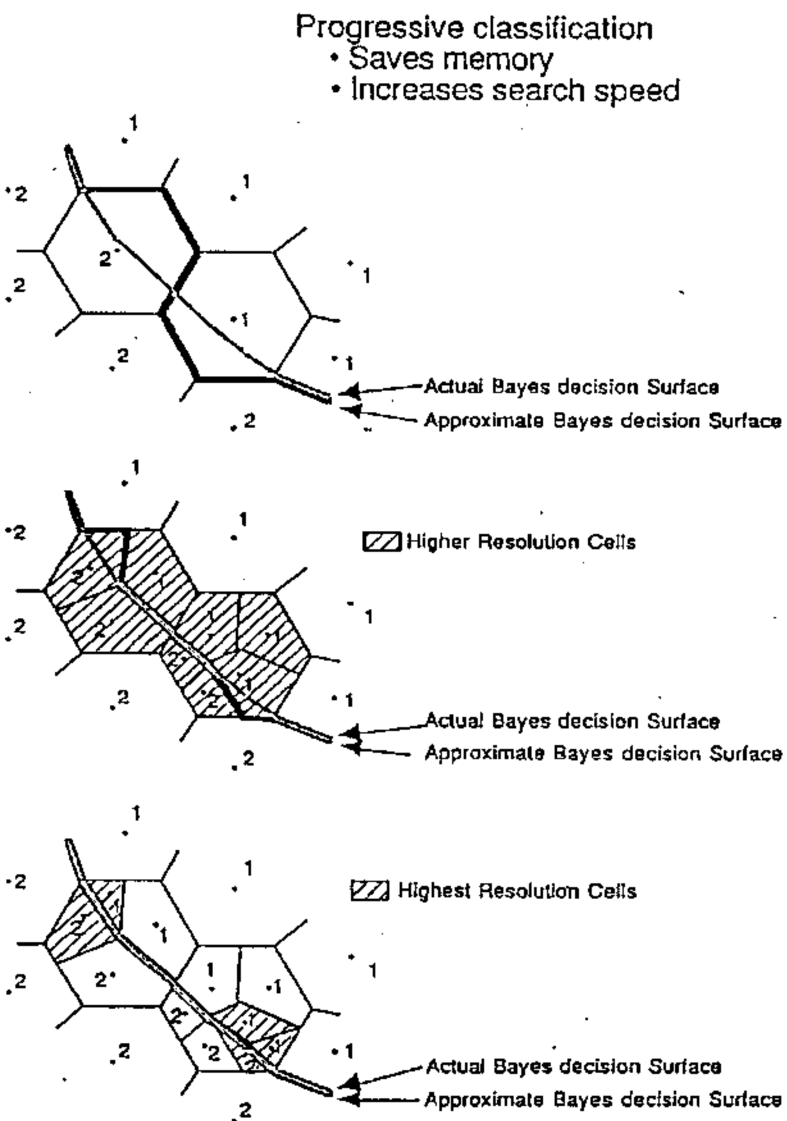


Decision Boundary (cont.)

Eventually ….



Extension to LTSVQ and Interpretation

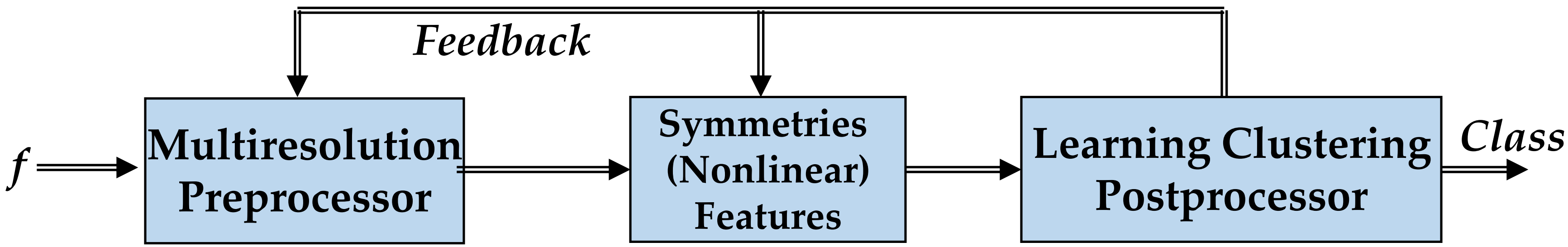


Extension to Learning TSVQ

- Combine LVQ with Deterministic Annealing -- Clustering error vs Purity of the Cluster (Entropy)
- Step needed for full analysis of WTSVQ and application in progressive classification (combined compression and classification framework)
- LTSVQ approximates directly the optimal Bayes surface with successive approximations and variable (along the surface) resolution
  - Split cells where approximation is not very good using finer resolution data
  - Akin to a multigrid numerical computation of the Bayes surface
- Application to state aggregation in control

Baras & Dey, IEEE T IT, 1999; Baras & Borkar, 2000 IEEE CDC

Multiresolution Learning Clustering



- Address hierarchical organization of signal databases, progressive classification:
  - **Combine a multiresolution preprocessor with a learning clustering postprocessor**
  - **Novel: Local and Global Feedback → Recurrent CNN (RCNN)**
- Resulting algorithms proved to have some "universal" qualities
- Found analogs of such algorithms in animals and humans:
  - **Hearing and sound classification**
  - **Vision and identification of objects by humans**
- Mathematical formulation: combined compression and classification for general signals
- "One Learning Algorithm" Hypothesis : Auditory cortex learns to see [Row et al, 1992], Somatosensory cortex learns to see [Metin & Frost, 1989]
- Our work on this problem started in the 90's working on multisensory ATR for the Navy!
- Parallelizable easily; Faster training; Insertion of new models easy
- Applied to compression and control (via state aggregation)

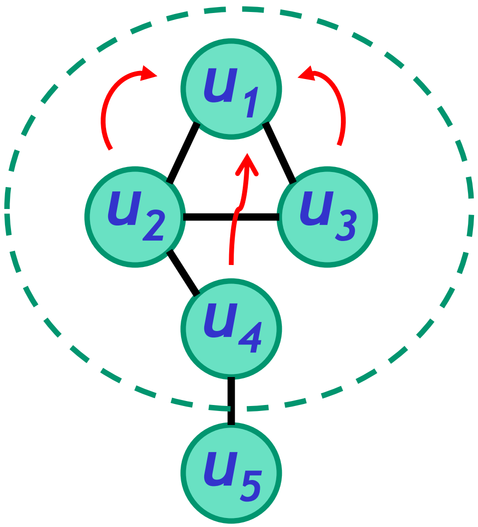
Baras & Dey, IEEE T IT, 1999; Baras & Borkar, 2000 IEEE CDC

Need to Integrate Semantic Vector Space Models and Knowledge Graphs (Hypergraphs) Models

- Convert the h-hop neighborhood of each node u in G into a multi-dimensional vector $R_G(u)=\{\langle u', w_u(u')\rangle\}$, based on the distance of neighbor nodes u′ from u.

$$w_u(u') = \begin{cases} \alpha^{d(u,u')}, & \underline{d(u,u')} \leq h; \\ 0, & \text{otherwise.} \end{cases}$$

Distance between u and u’



Information Propagation

Previous Applications of Information Propagation: Semi-supervised Learning [AI’ 08], Concept Propagation [CIKM ’06]

Advancing AI and ML for Autonomy: our Approach

- Rigorous Mathematics for Deep Networks – Universal Architecture emerging
- Non von-Neumann computing – do not separate CPU from Memory – Synaptic NN, in-memory processing -- HTM
- Universal ML -- Integrate Deep NN and Synaptic NN
- Knowledge Representation and Reasoning: Integrate Knowledge Graphs and Semantic Vector Spaces
- Progressive Learning, Knowledge Compacting
- Link Machine Learning with Knowledge Representation and Reasoning
- Inspirations from neuroscience

Composable Autonomy V&V
Proposed Novel Approach

AKA Trusted Autonomy:

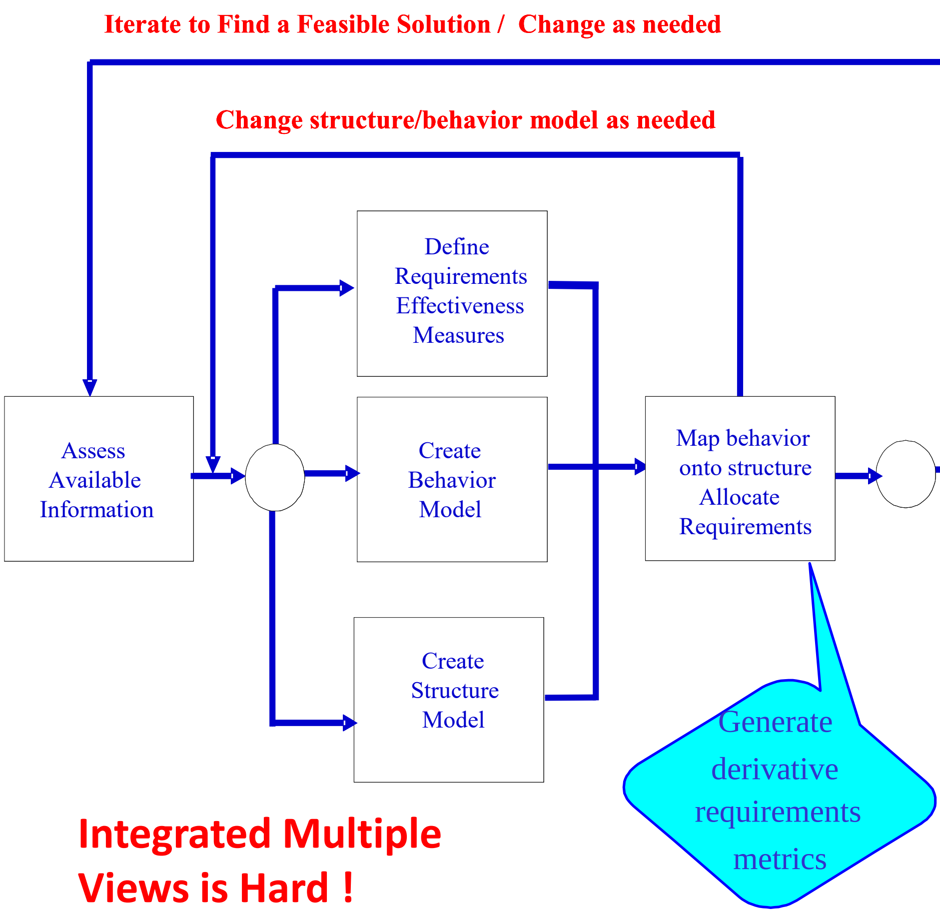
- **Formal models of tasks and missions combining spatial and temporal tolerances**
- **Composability: Requirements, Models, Tasks, Formal Models (Timed Automata, MITL, STL, contracts), Sensing, Control, Optimization**
- **Self-monitoring, self-learning and self-adjustment for correct autonomous execution of tasks**
- **Integrate composability methods and algorithms, with the rigorous model-based systems engineering methodology and framework we have developed (Baras)**

MODEL-BASED SYSTEMS ENGINEERING COMPONENTS -- ARCHITECTURE

Integrated System Synthesis Tools - & Environments missing

Model - based
- **UML - SysML - GME - eMFLON**
- **Rapsody**
- **UPPAAL**
- **Artist Tools**
- **MATLAB, MAPLE**
- **Modelica / Dymola**

DOORS, etc
CONSOL-OPTCAD
CPLEX, ILOG SOLVER,
SIEMENS, PLM, NX, TEAM CENTER



Model - Based Information - Centric Abstractions

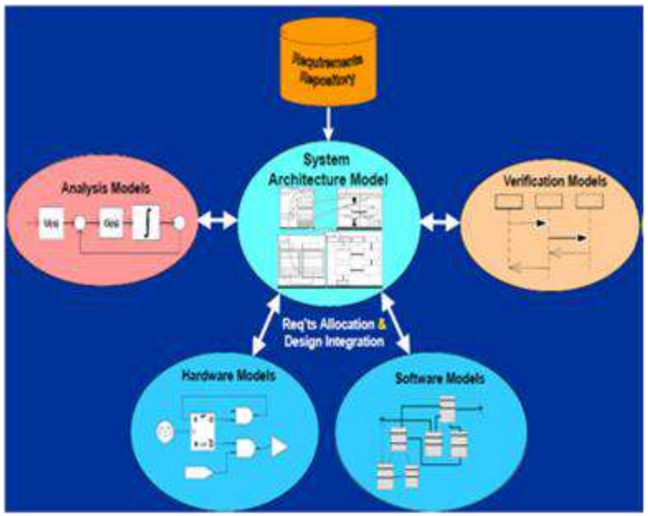
Integrated Multiple Views is Hard !

A Rigorous Framework for Model-based Systems Engineering

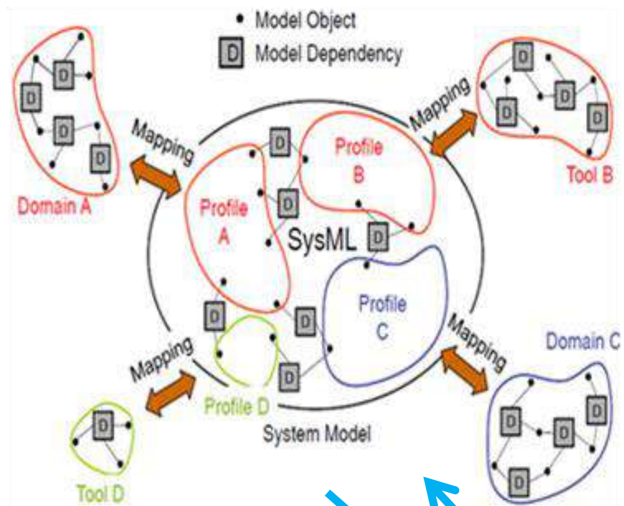
The Challenge & Need:

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

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



System Modeling Transformations



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**

“ 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

APPLICATIONS

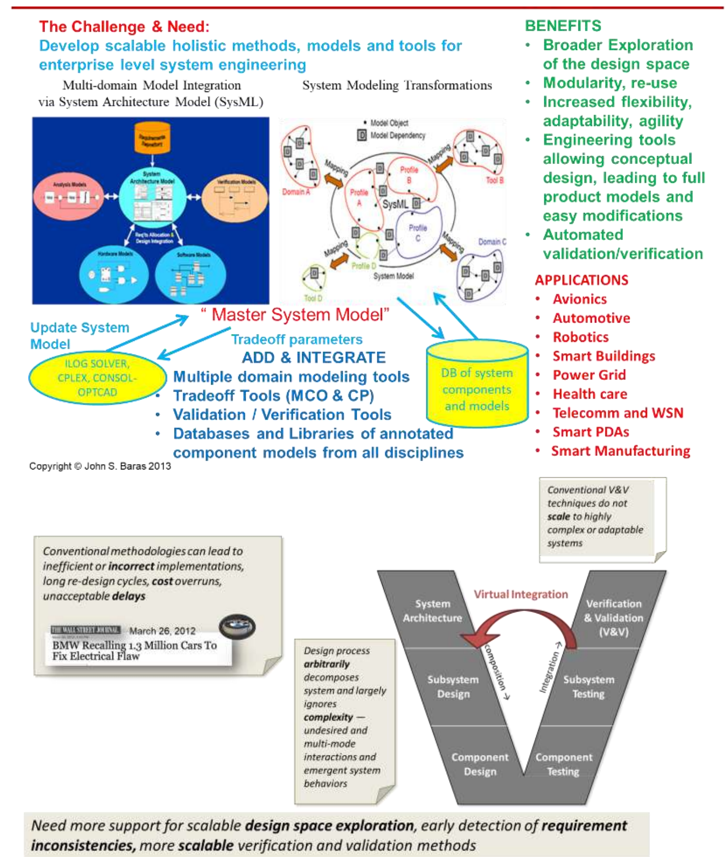
- **Avionics**
- **Automotive**
- **Robotics**
- **Smart Buildings**
- **Power Grid**
- **Health care**
- **Telecomm and WSN**
- **Smart PDAs**
- **Smart Manufacturing**

Requirements Engineering

- How to represent requirements?
  - Automata, Timed-Automata, Timed Petri-Nets
  - Dependence-Influence graphs for traceability
  - Set-valued systems, reachability, … for the continuous parts
- How to automatically allocate requirements to components?
- How to automatically check requirements?
  - **Approach**: Integrate contract-based design, model-checking, automatic theorem proving
- How to integrate automatic and experimental verification?
- How to do V&V at various granularities and progressively as the design proceeds – not at the end?
- The front-end challenge: Make it easy to the broad engineering user?

MBSE and the need for Contract Based Design

- The state-of-the-art Model Based Systems Engineering (MBSE) framework allows multi-domain model integration via SysML, trade-off analysis and verification/validation tools in for the development of complex systems.
- However, the current iteration of SysML does not permit the formalization of requirements which can then be tied together to model driven engineering to enable Correct-by-Construction designs.
- On the other hand, compositional approaches such as contract based design offer a comprehensive framework for early requirement validation with tight safety, reliability and performance guarantees and for scalable, system-level design space exploration under a set of heterogenous constraints.



[Nuzzo and ASV, "Let's Get Physical: Computer Science Meets Systems", FPS'14]

Contracts for Learning-Enabled Systems

- Enhance system with mathematically rigorous learning components, including ML and AI components
- Add assurance on these learning components
- A/G contracts encode the behavior of individual components
- Designer can model the behavior of the learning-enabled components working side-by-side with other components
- Incorporate assurance states, risk states
- Strong connections with risk-based stochastic control and prediction with the addition of temporal constraints.
- Leads to efficient use of model-checking systems

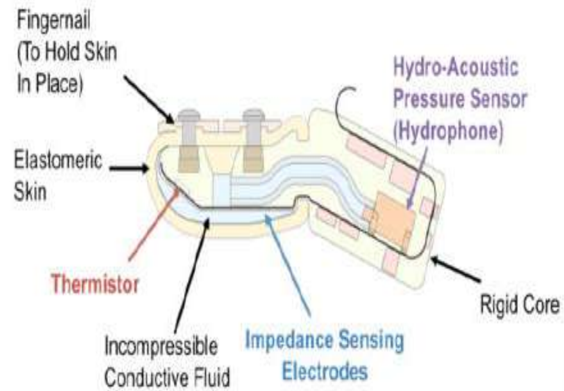
Dexterous Robotic Hand Grasping with Slippage Detection, Learning and Self-correction

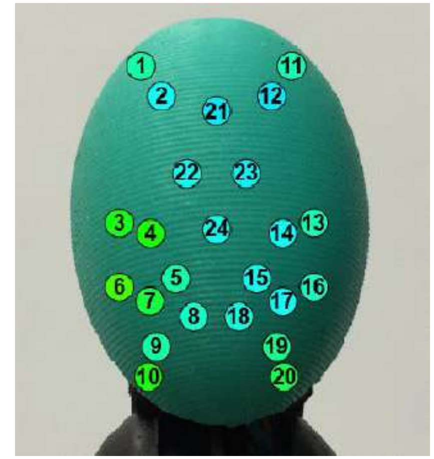
Zhenyu Lin, Charles Meehan, John Baras

Haptic-Vision Dataset

Haptic-Vision Dataset:

- Collected **synchronized** haptic and vision data for 80 experiments
- Haptic data is sampled at 100HZ with five modalities
- Vision data with resolution 1080P and 30fps framerate
- Slippage moment t^* is labeled for each experiment

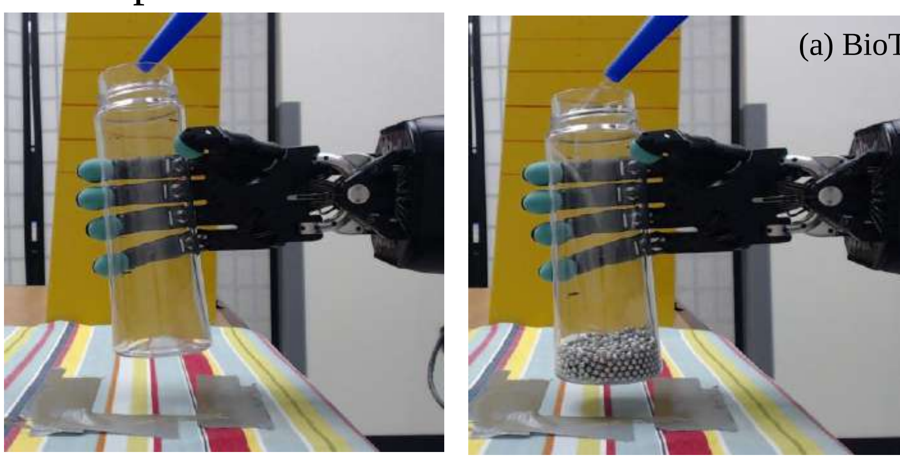




(a) BioTac S

Sensory Modality	Symbol	Range	Resolution	Frequency Response
Impedance	E_n	0 - 3.3V	3.2 mV	0 - 100 Hz
Fluid Pressure	P_{DC}	0 - 100 kPa	36.5 Pa	0 - 1040 Hz
Microvibration	P_{AC}	+/-0.76 kPa	0.37 Pa	10 - 1040 Hz
Temperature	T_{DC}	0 - 75 C	0.1 C	0 – 22.6 Hz
Thermal Flux	T_{AC}	0 - 1 C/s	0.001 C/s	0.45 – 22.6 Hz

Haptic data sensory modality

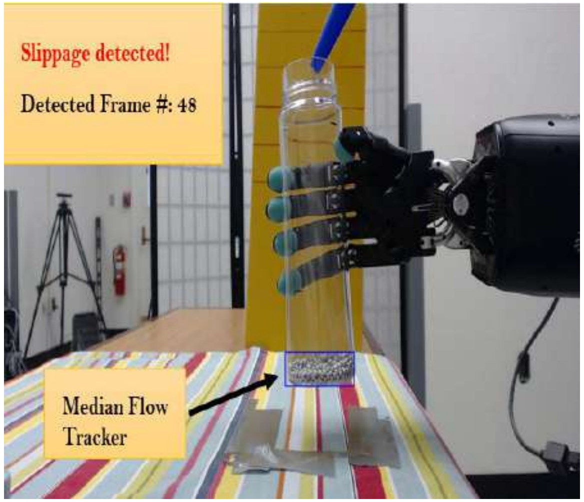


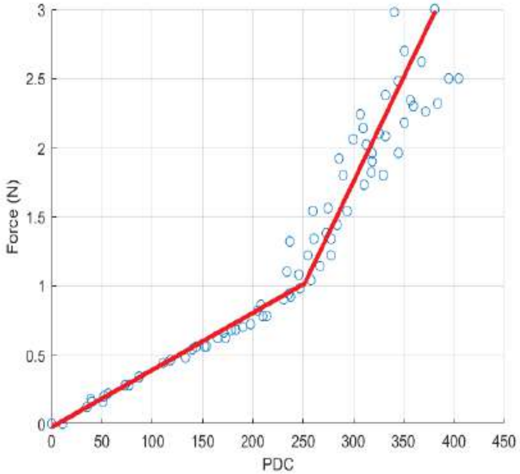
Sample vision data from experiment

Lin, Meehan, Baras, 2019 DGR Symp., IJRR, IEEE JRA, INCOSE SE

Median Flow Tracker: Slippage Detection Using Tactile Sensor

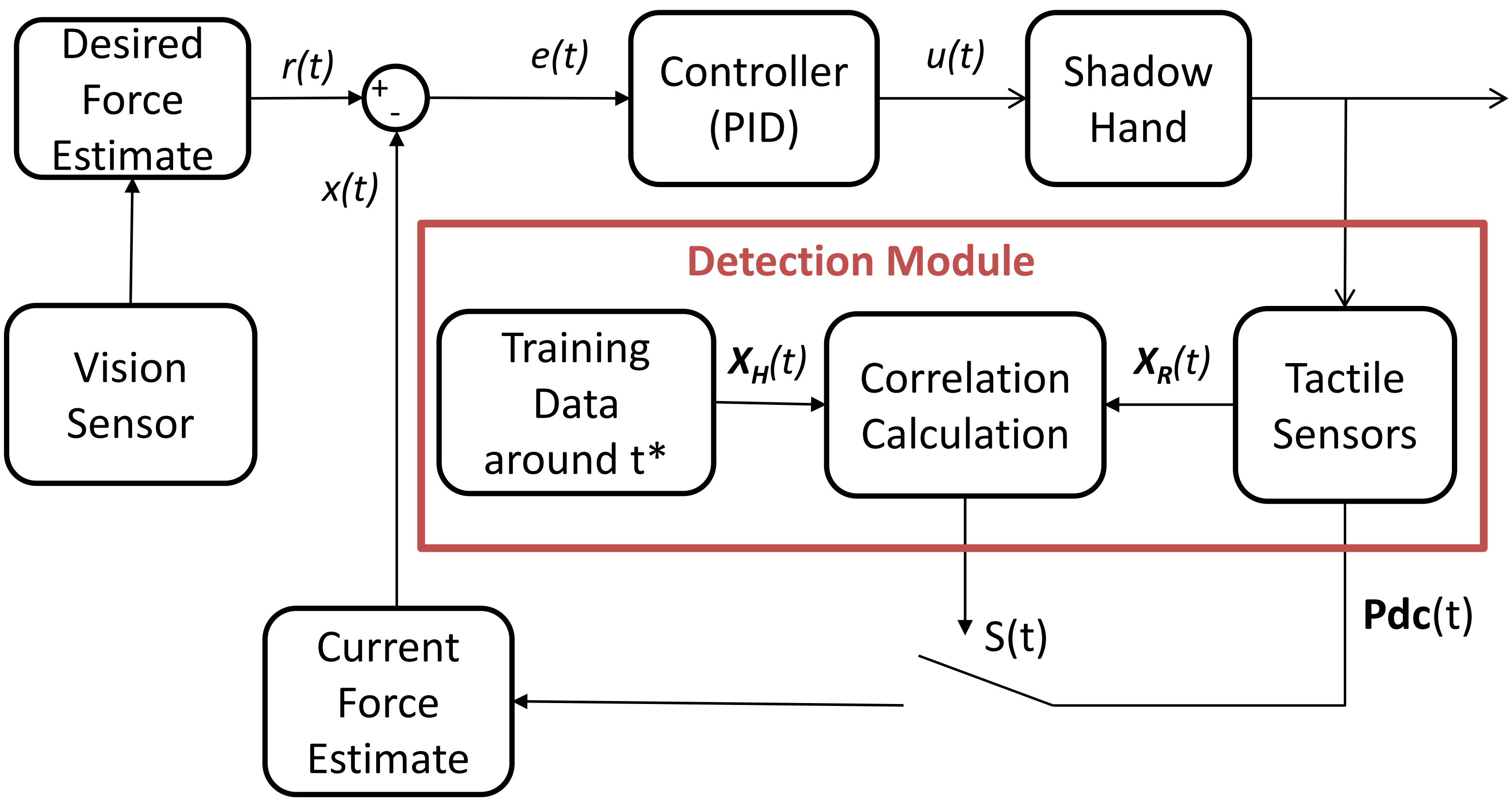
- Median flow tracker is used to detect the time *t** that slippage occurs for each experiment.
- Since there is only one possible direction of movement, the tracker has a high accuracy.
- Analyze the statistics of the haptic data around *t**
- Found peak correlation around moment of slippage
- Fluid pressure (PDC) used to estimate the force applied on the object
- Relationship between PDC and force is piece-wise linear (Coulomb model)
- Force estimation is very accurate within [0,1]N range
- During self-correction, the robot will apply suitable force based on weight estimation of grasped object





Lin, Meehan, Baras, 2019 DGR Symp., IJRR, IEEE JRA, INCOSE SE

Slippage Detection and Self-Correction Framework



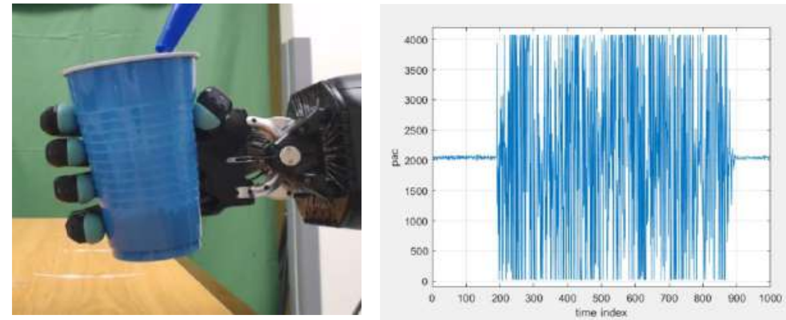
Lin, Meehan, Baras, 2019 DGR Symp., IJRR, IEEE JRA, INCOSE SE

Object Classification

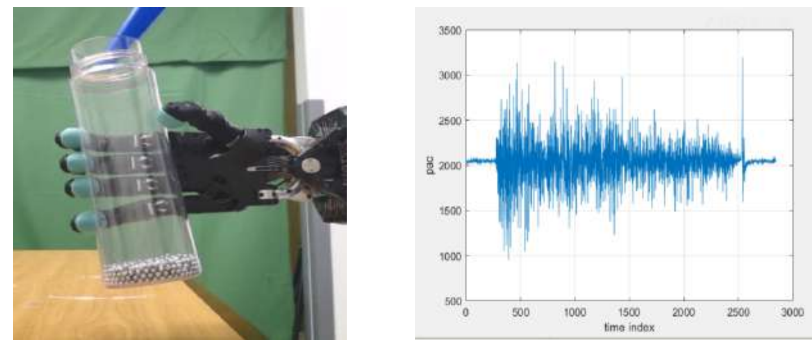
- We observe that the vibration sensor (PAC) data is very different between soft container and rigid container
- If an object is classified as soft, we will set another empirical threshold for the force applied on the object
- Performance based on detection time error: $e = \frac{|t_{detect} - t^*|}{t_{total}}$

TABLE I: Comparison slippage prediction using different methods and window size

Method Used	Window Size	Average Error
Corr-MAD-single	20	7.41
Corr-MAD-ALL	20	6.67
Corr-MAD-single	50	8.23
Corr-MAD-ALL	50	5.47
Corr-MAD-single	100	7.75
Corr-MAD-ALL	100	2.77
DNN-LSTM	100	15.57



(a) Soft Container



(b) Rigid Container

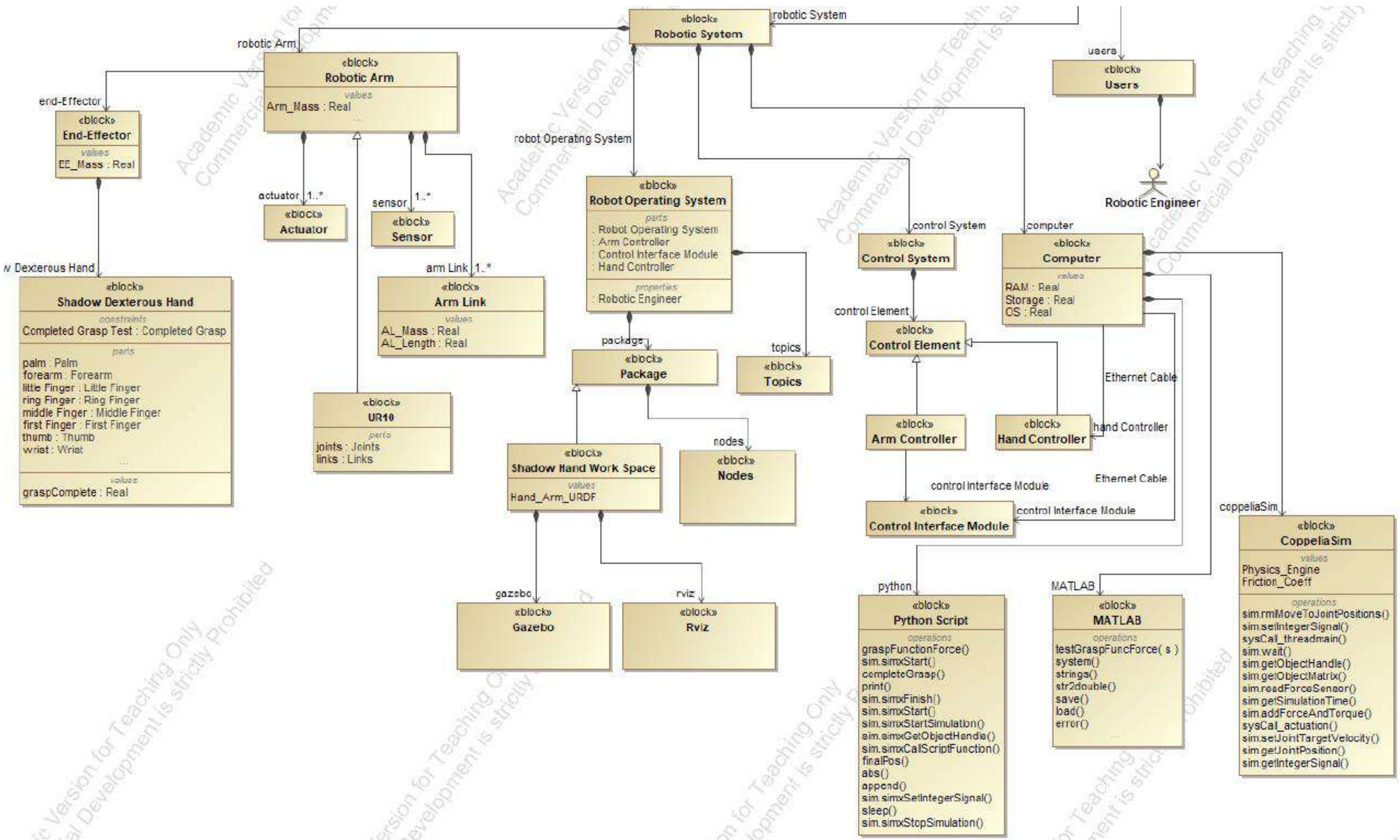
- Our LSTM approach similar to (Wyk, Falco, 2018)
- Our approach has higher accuracy

Lin, Meehan, Baras, 2019 DGR Symp., IJRR, IEEE JRA, INCOSE SE

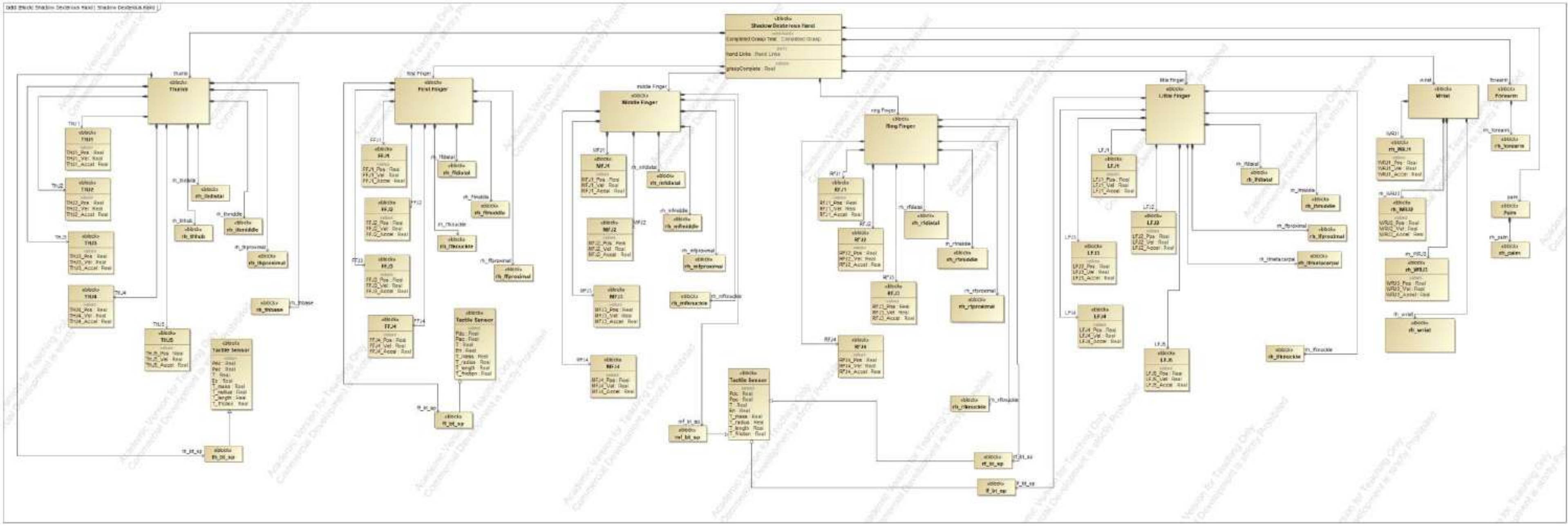
MBSE Framework for Dexterous Robotic Hand

- Created SysML architecture diagrams to model the structure and behavior of the robotic system (included UR10 arm, Shadowhand, BioTac sensor)
- Implemented Lua, MATLAB, and Python scripts to control the CoppeliaSim simulation. Vortex Studio used for stable grasping.
- Connected Cameo Systems Modeler (SysML) to the robotic simulation. Essentially creating a “Digital Twin” Framework.
- Designed and evaluated a slippage detection and correction algorithm for the experiments in the lab and simulation.
- Validated stakeholder requirements, verified the simulation requirements.
- Many behavior diagrams were created, including those modeling the slippage detection and correction problem in simulation.
- Main goal was to start simulation from Cameo Systems Modeler with certain input parameters, conduct simulation in a robotic simulator (CoppeliaSim), and receive the output metrics back into Cameo Systems Modeler once the simulation has ended.
- Used framework to optimize the rotational velocity of the finger and thumb joints used during the correction stage.

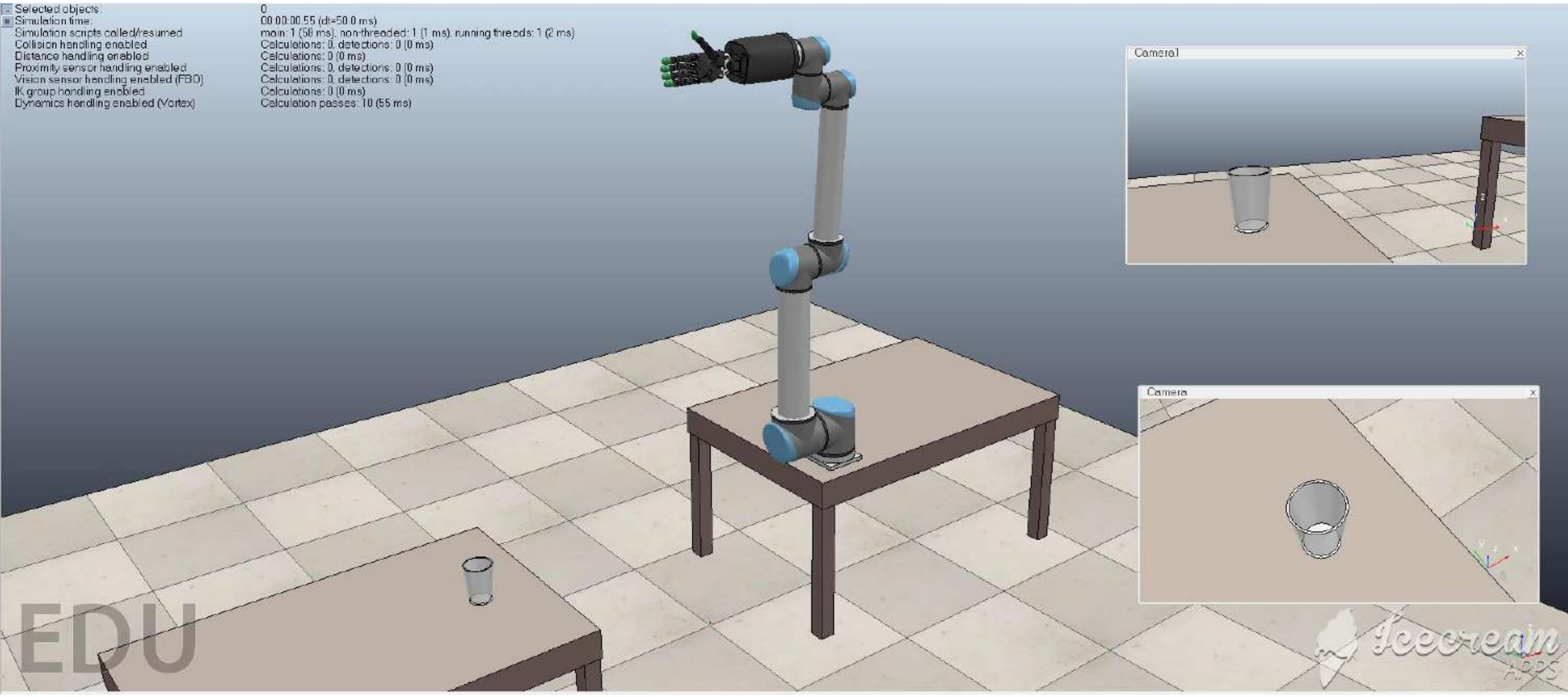
Robotic System Context-Level BDD: System



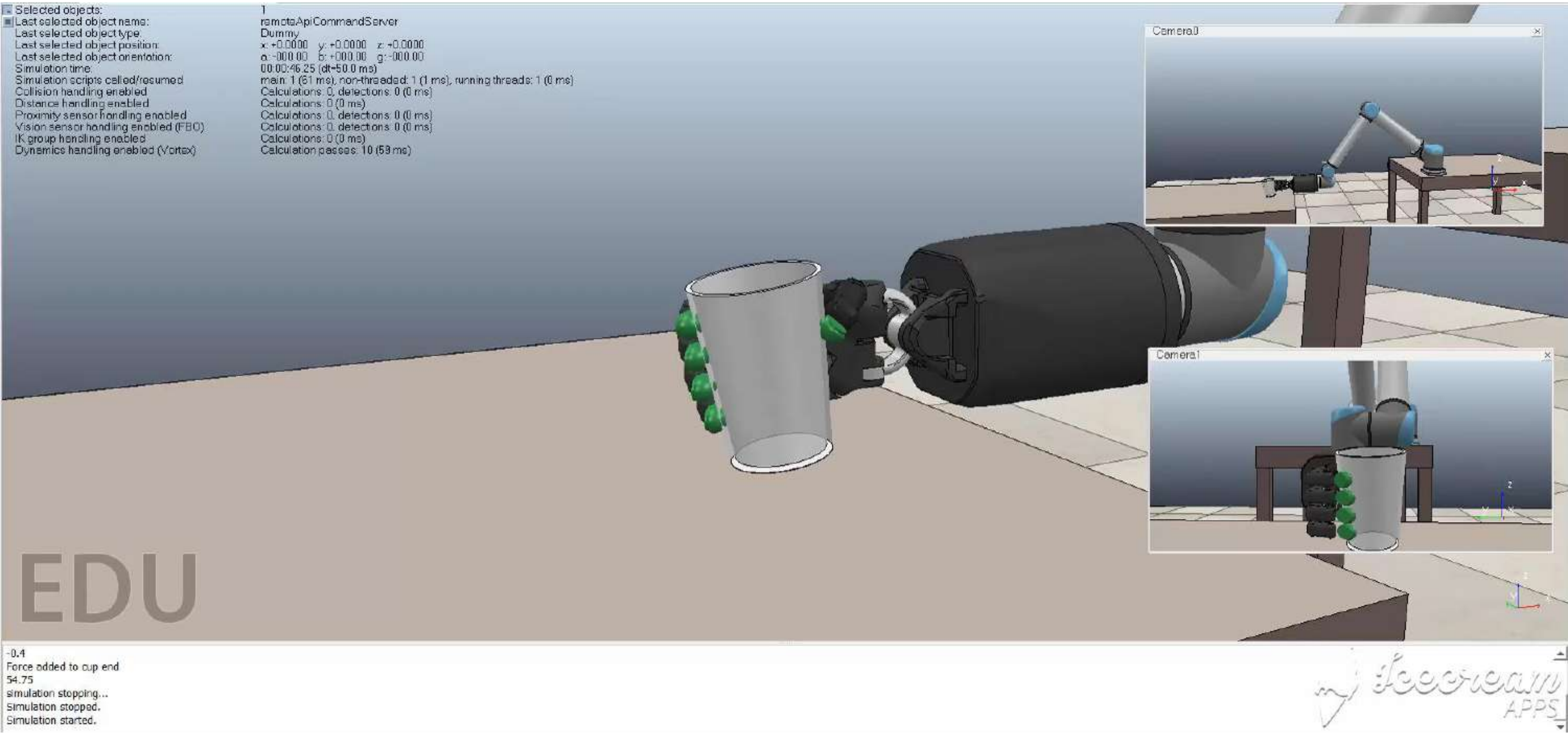
Shadow Hand BDD



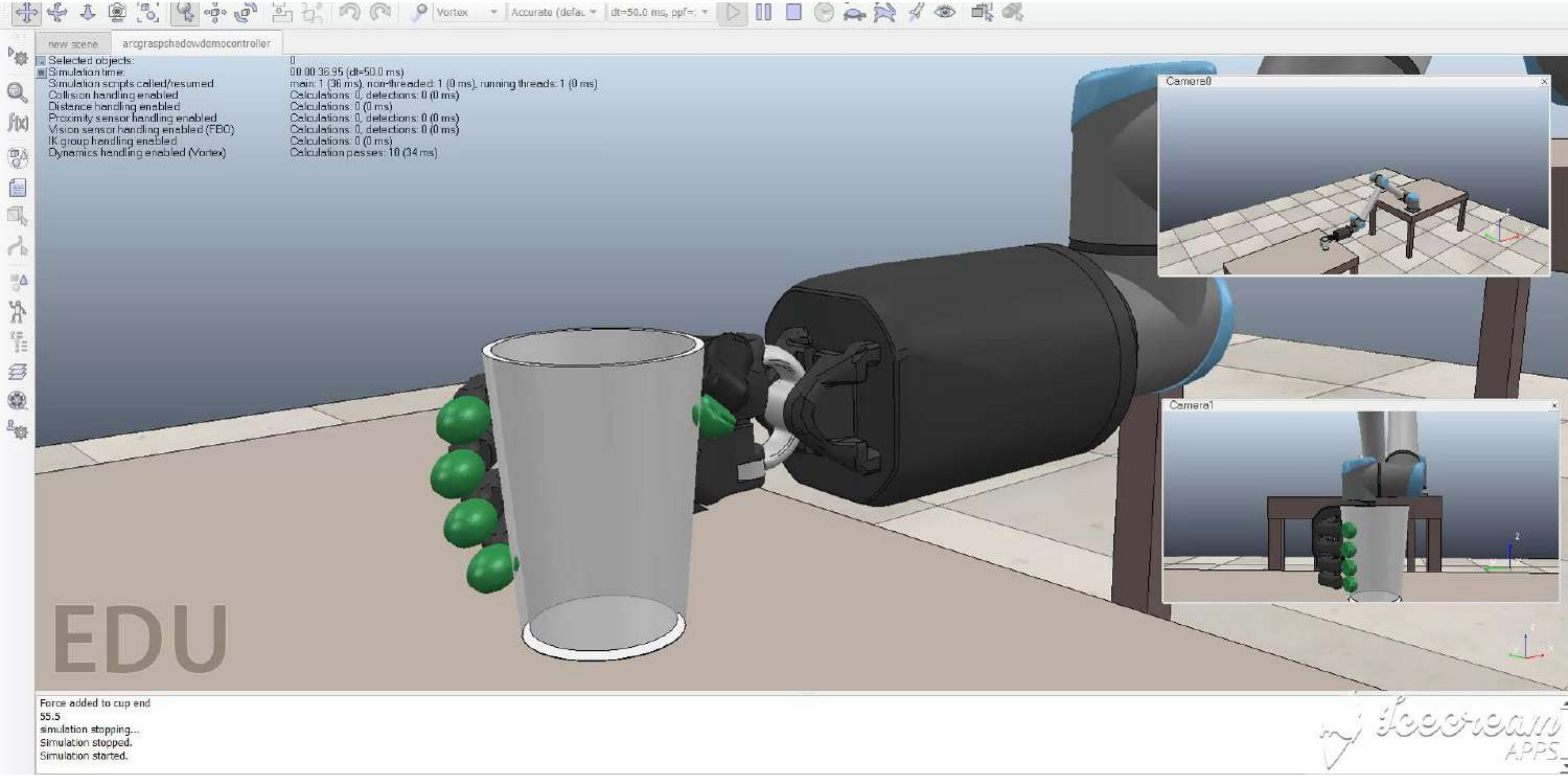
“Digital Twin” (Simulation) Demo



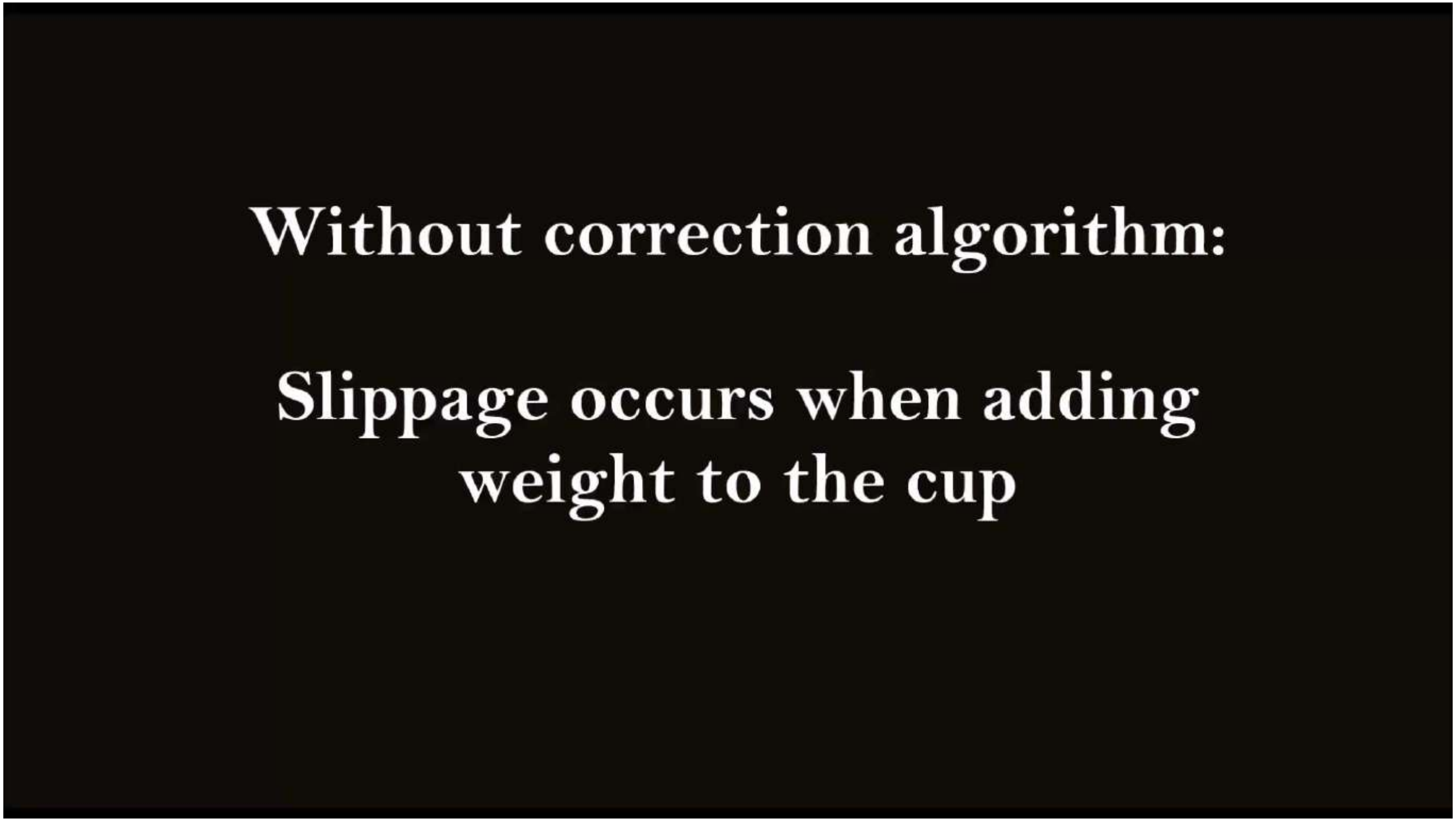
"Digital Twin" (Simulation) Demo: Shadow Hand Grasping Simulation with Force and No Correction



"Digital Twin" (Simulation) Demo: Simulation of Slippage Detection and Correction Problem



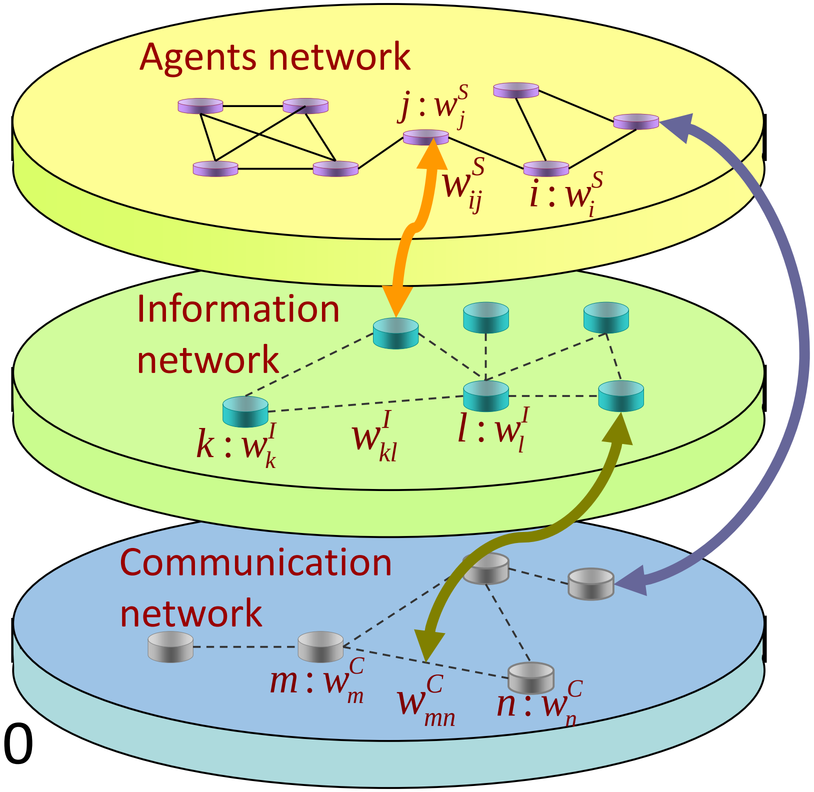
Demo: Grasping without and with self-correction



Lin, Meehan, Baras, 2019 DGR Symp., IJRR, IEEE JRA, INCOSE SE

Multi-Agent Autonomous Systems: Multiple Coevolving Multigraphs

- Multiple Interacting Graphs
  - *Nodes*: agents, individuals, groups
  - Directed graphs
  - *Links*: ties, relationships
  - Weights on links : value (strength, significance) of tie
  - Weights on nodes : importance of node (agent)
- **Real-life problems: Dynamic, time varying graphs, relations, weights, policies**



- We introduced these models -- 2010
- Used them recently to model Net-CPS, Net-CHPS
- Investigated effects of topology: proved Small World Graphs speed up consensus (probabilistic argument)

Multi-Agent Autonomous Systems: Multiple Coevolving Multigraphs

- Investigated coalition formation via constrained coalitional games
- Showed expander graphs are “best” topologies for communication
- Showed effects of communication topology on convergence speed and robustness of MA algorithms
- Proved the stability of car platooning (long standing problem)
- Developed and analyzed extremely simple distributed algorithms achieving system optimality – with no communication or with just one bit communication -- currently studying ways to speed up
- Showed emergence of motifs in communication graph for simpler controls
- Proved a simple version of Arrow’s 1974 conjecture: Trust is a catalyst for collaboration
- Investigated dynamics of trust and mistrust
- Investigated dynamics of various types of signed networks

Multi-Agent Autonomous Systems: Multiple Coevolving Multigraphs

- Developed novel partially ordered semiring model for indirect trust dynamics
- Showed that dynamic trust mechanisms are essential for achieving consensus in the presence of adversaries (applications to various distributed algorithms)
- Showed the need for noncommutative probability models (von Neumann like) for MA systems
- Recently initiated investigation of joint optimization of information and control – led to discovery of new values of information (non Shannon)
- Showed the interactions between information and control lead to constrained event algebras that can only be modeled by "Independence Friendly Logic", which has game theoretic semantics
- Investigation of the effects of resource constraints, stress, etc. on human decision making using these noncommutative probability models
- Investigation of the emergence of noncommutative probability models in asynchronous autonomous networked systems, and human machine teams – connections to neuropsychology and human behavior studies
- Investigation of connections to human cognition, decision making, risk

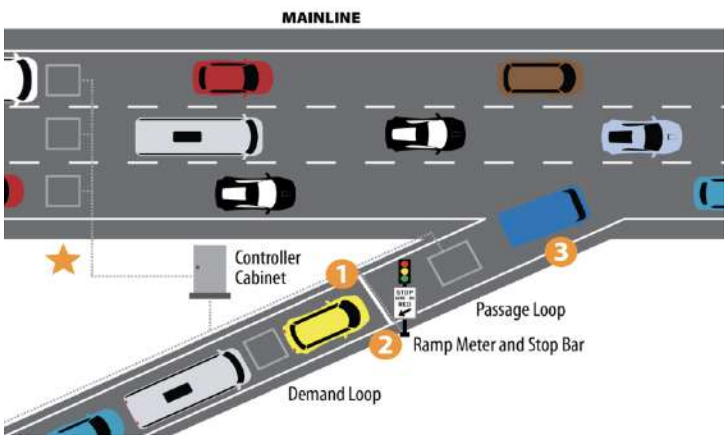
Outreach and Collaborations:
On-Going Applications

Highway on-ramp merging control with V2X information

- **Focus on the problem of highway on ramp merging with cooperation and communication between vehicles and infrastructure**
- Key Issues:
    - Overflow effect caused by merge delay in which outer lanes of highway becomes congested
    - Delay caused in both merging traffic and highway traffic
    - Unfair allocation of merging rights (Varying priorities)
    - Fairness and robustness of algorithms under changing traffic densities
- Sensor suite:
- Radar sensors, lidar sensors, odometry sensors and speed traps
- V2I communication and V2V communication

Nilesh Suriyarachchi and John Baras, with Maryland Transportation Institute

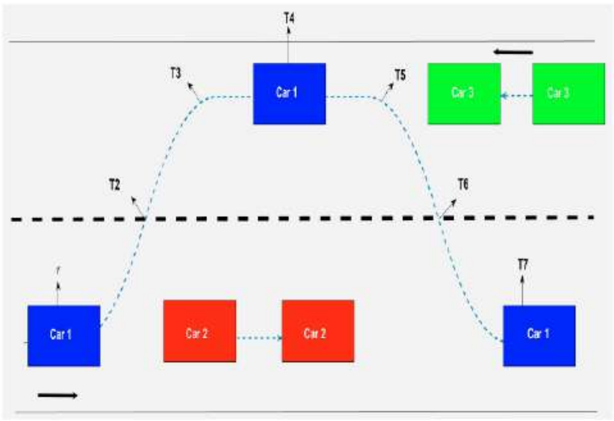
Integration of Ramp Metering Control and Route Guidance Strategy for Beltway Networks Using Model Predictive Control

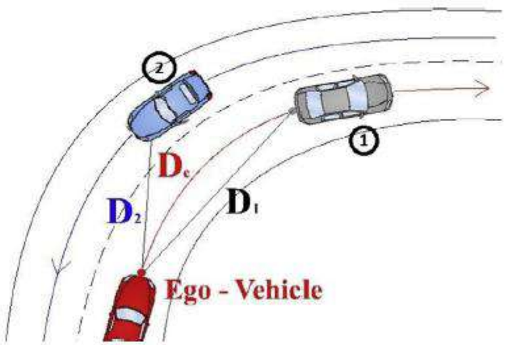


Fatima Alimardani and John Baras, with Maryland Transportation Institute

Car Overtake Problem

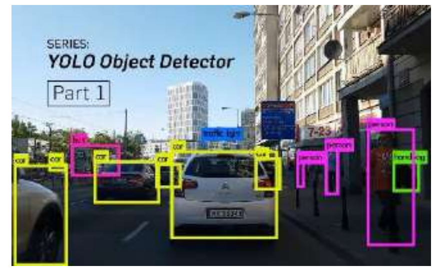
- **Problem:** Generation of trajectories and control laws for the autonomous ego vehicle to safely overtake a human-driven vehicle travelling in the same direction while avoiding the oncoming traffic in the adjacent lane
- **Objective:** Stochastic control formulation of the problem employing metric temporal logic constraints in order to satisfy the safety guarantees
- **Outcome:** A robust and safe algorithm which will be implemented and tested in a simulation environment

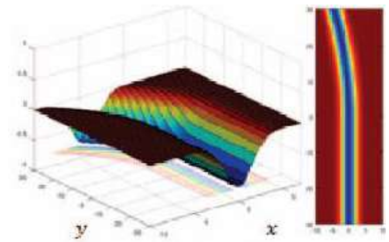


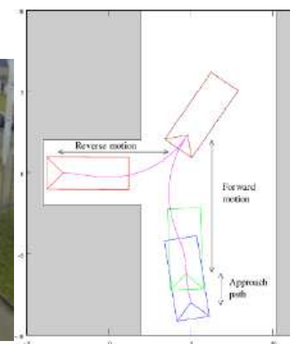


Autonomous Vehicle Parking Control

- **Key Research Aspects**
- Velocity control
  - Dynamic velocity adjustment for obstacles
  - Velocity adjustment along curves
- Obstacle Avoidance with Neural Nets
- Reverse Parking Logic Control
  - Importance of States
  - Transition Logic accuracy
  - Accuracy and repeatability of algorithm
- Parking Spot Detection
  - Minimal cost/complexity solution
  - Robustness for dynamic environments
  - Noise rejection



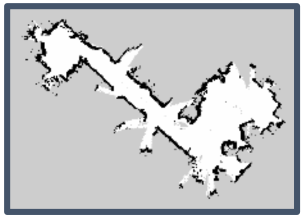




Nilesh Suriyarachchi and John Baras, with STEER

Autonomous Indoor Navigation using Visual SLAM

- **GOAL:** To perform ***robust, cost efficient*** autonomous navigation on off the shelf robotic platforms using Visual SLAM
- **Key Research Aspects**
- Cloud based computing
  - Reduces computation load in the robot
  - Can use low cost simple robotic platforms
  - Leverages 5G and advanced networking capabilities
- Multiple sensor fusion
  - Critical for dealing with multiple input data streams
  - Importance of time synchronization (Real-time applications)
  - Stabilized data frame output
- Visual feature-based localization
  - Graph SLAM optimization
  - Hardware and software upgrades for robustness
  - Multiple scenario testing

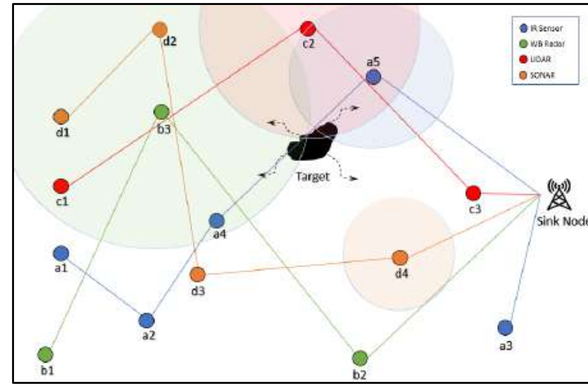


NOKIA Bell Labs

Composable Autonomy and V&V for Distributed Heterogenous Sensor Networks

- **<u>Objective:</u>**
  - Develop a novel methodology for the modeling, design and V&V of distributed ISR systems with heterogeneous sensors.
  - Methodology will be implemented into a prototype Model Based Systems Engineering (MBSE) framework for these systems.
  - Such a rigorous MBSE methodology and framework for distributed sensor ISR systems does not exist today.

- **<u>Tasks Description:</u>**
  - ***<u>Task 1</u>:*** Select a small set of sensor types for the Wireless Sensor Network (WSN) model. Develop simple performance models for the performance of each sensor type in relation to the ISR use case.
  - ***<u>Task 2</u>:*** Select a small number of sensors and a network architecture for modeling an initial distributed sensor ISR system. Specify the performance metrics and requirements.
  - ***<u>Task 3</u>:*** Apply the new MBSE methodology to the resulting networked system. Ensure that a simple executable simulation works. Model specifications using metrics, constraints, contracts and timed automata as needed. Keep the overall system simple as time is limited.




(a) LIDAR


(b) IRST


(c) RADAR

SENSORS


(i) UAV


(ii) UGV

PLATFORMS

Sandeep Damera, Marilyn Duong, and John Baras, with Lockheed Martin and NRL

Other Collaborations

- Marilyn Duong – NRL Internship on human robot collaboration
- Charles Meehan – NRL Internship on autonomous sensor networks
- John Baras – with Lynn Ewart and Scott Sideleau of USN NUWC Newport RI, on MBSE for hybrid autonomous UAS teams mission planning and operation
- David Hartman, Erfaun Noorari, John Baras with Brian Sadler of ARL on collaborative autonomous teams and RL
- Christos Mavridis and John Baras – with Alex Duda and Neta Ezer of Northrop Grumman on LVQ, SOM, and application to autonomous acoustic sensor networks

Future Directions

- Develop further the framework for synthesis of autonomous systems with safety
- Develop further composability theory and methods (divide and conquer)
- Self-monitoring, Self-adjustment, for Correctness
- Incorporate advanced learning, safe learning, sensor fusion
- New “Value of Information” metrics and Control-Information duality
- Incorporate time and complexity metrics
- Investigate further new logics for autonomous teams (machine and human-machine)
- Risk sensitive decision making, Prospect theory, autonomy, RL
- Reverse engineer flocks and biological collectives
- Further demonstrations in UAV, robotic, autonomous ground vehicle examples
- Unmanned ship and underwater vehicle applications
- Further demonstrations in UAV, robotic, autonomous ground vehicle examples
- Unmanned ship and underwater vehicle applications
- Link to advanced simulators and on-line data input

Thank you!

baras@umd.edu

301-405-6606

https://johnbaras.com/

Questions?