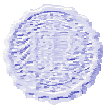
Hybrid System Design and Implementation Methodologies for Multi-Vehicle Multi-Modal Control

Shankar Sastry, Thomas Henzinger and EdwardLee
Alberto Sangiovanni Vincentelli

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley



Statement of Work

- **Thrust I: Experimental Evaluation of Multi-Vehicle Control System Designs. Run Time Executions:**
  1. Mode Switching in UAVs: flight envelop protection, survivability in normal modes of operation.
  2. Degraded Modes of Operation: loss of communication, loss of individual sensors, actuators.
  3. Multiple UAV Coordination: formation flying, pursuit-evasion scenarios.
- **Thrust II Multi-modal Control Derivation and Analysis. Design Tools. Design Tools:**
  1. Algorithmic Analysis for Nonlinear Hybrid Control
  2. Hierarchical Hybrid Control Design, Modular techniques
  3. Model Reduction and Conservative Approximatons

Statement of Work Part II

- **Thrust III: Hybrid Model Simulation and Implementation on the Open Control Platform. Run Time Implementation.**

  1. Hybrid Multi-Vehicle Model Simulation: mixed models of computation.

  2. Structuring Mechanisms for Hybrid Models: for managing complexity.

  3. Executability of Hybrid Models: determinacy, receptiveness.

  4. Architectural Mapping and Real time Analysis of Hybrid Control Designs: mapping "proven" designs onto OCP and to provide "guarantees" for different implementations: synchronous at low level, Corba/Tao at networked level?

  5. Robustness and Error Analysis of Hybrid Control Designs.

Statement of Work Part II

- **Thrust III: Hybrid Model Simulation and Implementation on the Open Control Platform. Run Time Implementation.**

  1. Hybrid Multi-Vehicle Model Simulation: mixed models of computation.

  2. Structuring Mechanisms for Hybrid Models: for managing complexity.

  3. Executability of Hybrid Models: determinacy, receptiveness.

  4. Architectural Mapping and Real time Analysis of Hybrid Control Designs: mapping "proven" designs onto OCP and to provide "guarantees" for different implementations: synchronous at low level, Corba/Tao at networked level?

  5. Robustness and Error Analysis of Hybrid Control Designs.

Statemement of Work Part III

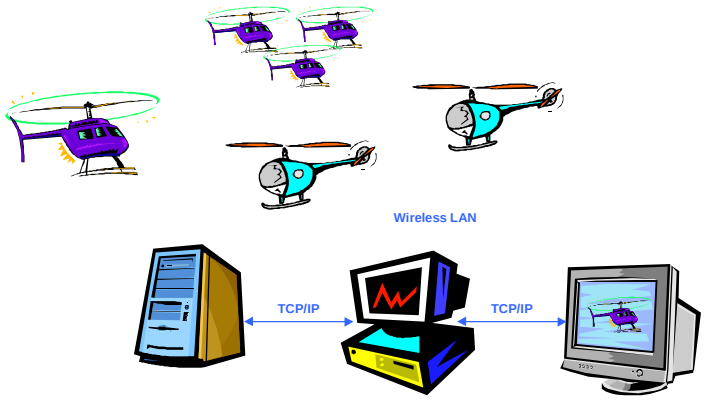
- **Thrust IV: Probabilistic Design and Active Fault Handling for Hybrid Systems. Design Time / Real Time.**

  1. Probabilistic Control: when specs cannot be met deterministically.

  2. Probabilistic Analysis: probabilistic estimates of safe and desired behavior.

  3. On-line Customization of Control: Active Hybrid Control: "adaptive control" during operation of system, embedding design abstractions.

System Configuration



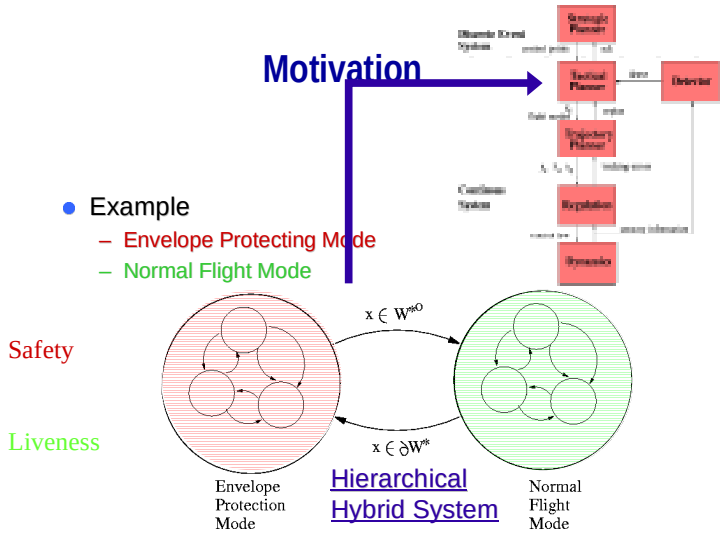
Motivation

- Goal
  - Design a multi-agent multi-modal control system for Unmanned Aerial Vehicles (UAVs)
    - Intelligent coordination among agents
    - Rapid adaptation to changing environments
    - Interaction of models of operation
  - Guarantee
    - Safety
    - Performance
    - Fault tolerance
    - Mission completion

Conflict Resolution
Coll
Env
Tracking Error
Fu
Re
Sensor Failure
Ac
Path Following
Object Searching
Pursuit-Evasion

Motivation

- Example
  - Envelope Protecting Mode
  - Normal Flight Mode



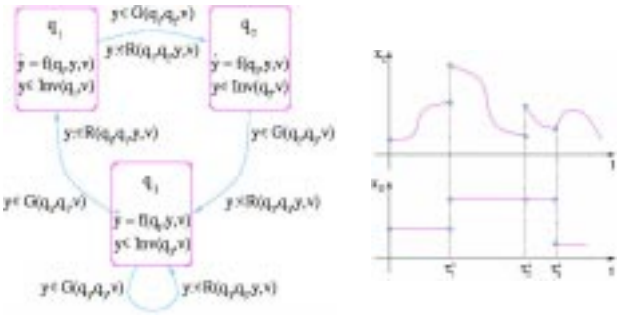
Hierarchical Hybrid System

System Design Flow

- **Mission Specifications** Path Following/Object Searching/Pursuit-Evasion
- **System Identification** Nonlinear Model/Linear Model
- **Controller Synthesis** Envelope Protection/Tracking/Regulation
- **Hybrid System Synthesis** Conflict Resolution/Collision Avoidance/Flight Mode Switching
- **Hierarchical Hybrid System Synthesis** Flight Management System
- **Verification** Safety/Mission Completion
- **Simulation** Hierarchical Hybrid System
- **Embedded System Synthesis** HW/SW+RTOS
- **Validation** Simulation/Emulation

What Are Hybrid Systems?

Dynamical systems with interacting continuous and discrete dynamics

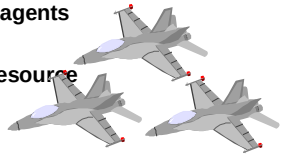


Why Hybrid Systems?

- Modeling abstraction of
  - Continuous systems with phased operation (e.g. walking robots, mechanical systems with collisions, circuits with diodes)
  - Continuous systems controlled by discrete inputs (e.g. switches, valves, digital computers)
  - Coordinating processes (multi-agent systems)
- Important in applications
  - Hardware verification/CAD, real time software
  - Manufacturing, communication networks, multimedia
- Large scale, multi-agent systems
  - Automated Highway Systems (AHS)
  - Air Traffic Management Systems (ATM)
  - Uninhabited Aerial Vehicles (UAV), Power Networks

Control Challenges

- Large number of semiautonomous agents
- Coordinate to
  - Make efficient use of common resource
  - Achieve a common goal



- Individual agents have various modes of operation
- Agents optimize locally, coordinate to resolve conflicts
- System architecture is hierarchical and distributed
- Safety critical systems

***Challenge**: Develop models, analysis, and synthesis tools for designing and verifying the safety of multi-agent systems*

Hybrid Automata

- **Hybrid Automaton** $H = (X; V; Init; f; Inv; R)$
  - **State space** $X = X_C \times X_D$
  - **Input space** $V = V_C \times V_D$
  - **Initial states** $Init \subseteq X$
  - **Vector field** $f : X \times V \to \mathbb{R}^n$
  - **Invariant set** $Inv \subseteq X \times V$
  - **Transition relation** $R : X \times V \to 2^X$
- **Remarks:**
  - $X_D; V_D$ **countable,** $X_C = \mathbb{R}^n; V_C \subseteq \mathbb{R}^m$
  - **State** $x = (q; y) \in X$
  - **Can add outputs, etc.**

Executions

- **Hybrid time trajectory,** $\tau = \{[\tau_i; \tau_i']\}_{i=0}^{N}$**, finite or infinite with** $\tau_{i+1} = \tau_i \cap \tau_i'$
- **Execution** $\chi = (\tau; x; v)$ **with** $x : \tau \to X; v : \tau \to V$ **and**
  - **Initial Condition:** $x(\tau_0) \in Init$
  - **Discrete Evolution:** $x(\tau_{i+1}) \in R(x(\tau_i'); v(\tau_i'))$
  - **Continuous Evolution: over** $[\tau_i; \tau_i']$**,** x **continuous, piecewise continuous,** $\dot{x} = f(x; v)$ **and** $(x(t); v(t)) \in Inv; \forall t \in [\tau_i; \tau_i']$
- **Remarks:**
  - **not function, multiple transitions possible**
  - **constant along continuous evolution**
  - **Can study existence uniqueness**
  - **Use** $\mathcal{H}_H$ **to denote the set of executions of** H

Controller Synthesis

- Consider **plant** hybrid automaton, inputs partitioned to:
  - Controls, U
  - Disturbances, D
- Controls specified by "us"
- Disturbances specified by the "environment"
  - Unmodeled dynamics
  - Noise, reference signals
  - Actions of other agents
- **Memoryless controller** is a map $g : X \to 2^U$
- The **closed loop executions** are

$$E^H_g = \{(\tau; x; (u; d)) \in E^H \mid \forall \tau; u(t) \in g(x(t))\}$$

Controller Synthesis Problem

- Given H and $F \subseteq X$ find g such that

$$\forall (\tau; x; (u; d)) \in E^H_g; \forall \tau; x(t) \in F$$

- A set $W \subseteq X$ is **controlled invariant** if there exists a controller such that all executions starting in W remain in W

Proposition: The synthesis problem can be solved iff there exists a unique maximal controlled invariant set with

$$Init \subseteq W \subseteq F$$

- Seek maximal controlled invariant sets & (least restrictive) controllers that render them invariant
- **Proposed solution:** treat the synthesis problem as a **non-cooperative game** between the **control** and the **disturbance**

Gaming Synthesis Procedure

- **Discrete Systems: games on graphs, Bellman equation**
- **Continuous Systems: pursuit-evasion games, Isaacs PDE**
- **Hybrid Systems: for $K; L \subseteq F$ define**
  - $Pre_u(K) \subseteq X$ **states that can be forced to jump to K by u**
  - $Pre_d(K) \subseteq X$ **states that may jump out of K for some d**
  - $Reach(K; L) \subseteq X$ **states that whatever u does can be continuously driven to K avoiding L by d**

Initialization: $W^0 = F; W^{-1} = \emptyset; i = 0$

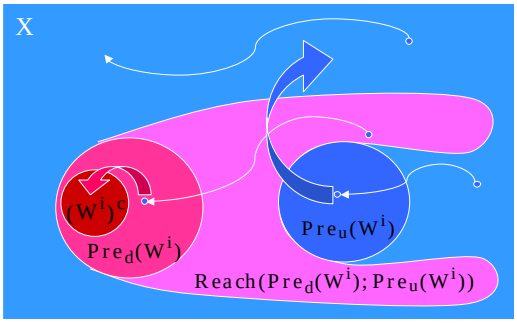
while $W^i \neq W^{i-1}$ ***do***

$W^{i+1} = W^i \cap Reach(Pre_u(W^i); Pre_d(W^i))$

$i = i + 1$

end

Algorithm Interpretation



Proposition: If the algorithm terminates, the fixed point is the maximal controlled invariant subset of F

Computation

- One needs to compute Pre_u, Pre_d and $Reach$
- Computation of the Pre is straight forward (conceptually!): invert the transition relation R

$$Pre_u(K) = \{x \in K \mid \exists u \in U, \forall d \in D, (x,(u,d)) \in Inv \wedge R(x,(u,d)) \in K\}$$

$$Pre_d(K) = \{x \in X \mid x \in K^c \wedge \forall u \in U, \exists d \in D, R(x,(u,d)) \setminus K^c \neq \emptyset\}$$

- Computation of Reach through a pair of coupled Hamilton-Jacobi partial differential equations

Reach Set Computation

Can be done one discrete "location", $q \in X_D$, at a time

Assume there exist real valued functions k, l such that

$$K = \{y \in X_C \mid k(y) < 0\} \quad L = \{y \in X_C \mid l(y) \geq 0\}$$

Solve the partial differential equations:

$$\partial J_K/\partial t = \min\left(0, H_K(y, \partial J_K/\partial y)\right)$$

$$\partial J_L/\partial t = \min\left(0, H_L(y, \partial J_L/\partial y)\right)$$

with initial condition $J_K(y,0) = k(y)$ and $J_L(y,0) = l(y)$

where the equations are coupled through their Hamiltonian

$$H_K(y,p) = \min_{u \in U} \max_{d \in D} p^T f(q,y,u,d) \quad \text{if} \quad J_L(y,t) > 0$$

$$H_K(y,p) = 0 \quad \text{if} \quad J_L(y,t) \leq 0$$

(and likewise for $H_L(y,p)$)

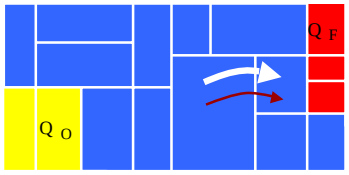
Transition Systems

- **Transition System** $T = (Q; \hat{I}; \rightarrow; Q_O; Q_F)$
- **Define for** $\hat{u} \subseteq \hat{I}; P \subseteq Q$

$$Pre_{\hat{u}}(P) = \{q \subseteq Q \mid \exists p \subseteq P \text{ and } q \rightarrow_{\hat{u}} p\}$$

- **Given equivalence relation** $\sim \subseteq Q \times Q$ **define**

$$T/\sim = (Q/\sim; \hat{I}; \rightarrow_{\sim}; Q_O/\sim; Q_F/\sim)$$



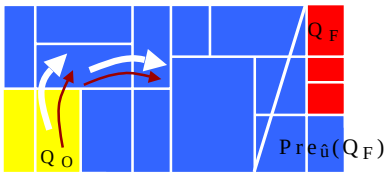
- **A ~ block is a union of equivalence classes**

Bisimulations of Transition Systems

A partition ~ is a bisimulation iff

- $Q_O; Q_F$ ***are ~ blocks***
- ***For all*** $\hat{u} \subseteq \hat{I}$ ***and all ~ blocks*** $P; Pre_{\hat{u}}(P)$ ***is a ~ block***

- **Alternatively, for** $P_1; P_2 \subseteq Q/\sim; P_1 \setminus Pre_{\hat{u}}(P_2) = \emptyset$ or P_1



- **Why are bisimulations important?**

Bisimulation Algorithm

initialize: $Q = \{Q_O, Q_F, Q \setminus (Q_O \cup Q_F)\}$
while $\exists P_1, P_2 \in Q = \{\dots, \hat{u} \in \hat{I}$ such that
$\emptyset \subset P_1 \cap Pre_{\hat{u}}(P_2) \subset P_1$
define $Q = \{\dots\} = (Q = \{\dots\} \setminus \{P_1\}) \cup \{R_1, R_2\}$
refine $R_1 = P_1 \cap Pre_{\hat{u}}(P_2); R_2 = P_1 \setminus Pre_{\hat{u}}(P_2)$

- If algorithm terminates, we obtain a finite bisimulation

Bisimulation Algorithm

- Refinement process is therefore decoupled
- Consider for each discrete state the finite collection of sets
  $A_q = \{I(q), (X_O)_q, (X_F)_q\} \cup \{G(e), R(e) | e \in E\}$
- Let S_q be a partition compatible with A_q

Initialize $X = \{\dots\} = \cup_q S_q$
for each $q \in X_D$
while $\exists P_1, P_2 \in S_q$ such that $\emptyset \subset P_1 \cap Pre_{\hat{u}}(P_2) \subset P_1$
  define $R_1 = P_1 \setminus Pre_{\hat{u}}(P_2); R_2 = P_1 \cap Pre_{\hat{u}}(P_2)$
  refine $S_q = (S_q \setminus \{P_1\}) \cup \{R_1, R_2\}$
end while; end for

- Algorithm must terminate for each discrete location

Computability & Finitiness

- Decidability requires the bisimulation algorithm to
  - Terminate in finite number of steps and
  - Be computable
- For the bisimulation algorithm to be computable we need to
  - Represent sets symbollically,
  - Perform boolean combinations on sets
  - Check emptyness of a set,
  - Compute Pre(P) of a set P
- Class of sets and vector fields must be topologically simple
  - Set operations must not produce pathological sets
  - Sets must have desirable finiteness properties

O-Minimal Theories

- A definable set is $\{(x_1,\ldots,x_n) \in \mathbb{R}^n \mid p(x_1,\ldots,x_n)\}$

A theory of the reals is called o-minimal if every definable subset of the reals is a finite union of points and intervals

Example: $\{x \in \mathbb{R} \mid p(x) > 0\}$ **for polynomial** $p(x)$

Recent o-minimal theories

$(\mathbb{R}; <; +; 0; 1)$	**Semilinear Sets**
$(\mathbb{R}; <; +; \cdot; 0; 1)$	**Semialgebraic Sets**
$(\mathbb{R}; <; +; \cdot; e^x; 0; 1)$	**Exponential Flows**
$(\mathbb{R}; <; +; \cdot; f^{|}; 0; 1)$	**Subanalytic Sets (bounded)**
$(\mathbb{R}; <; +; \cdot; e^x; f^{|}; 0; 1)$	**Spirals ???**

O-Minimal Hybrid Systems

A hybrid system H is said to be o-minimal if

- ***the continuous state lives in $\mathbb{R}^n$***
- ***For each discrete state, the flow of the vector field is complete***
- ***For each discrete state, all relevant sets and the flow of the vector field are definable in the same o-minimal theory***

Main Theorem

Every o-minimal hybrid system admits a finite bisimulation.

- **Bisimulation alg. terminates for o-minimal hybrid systems**
- **Various corollaries for each o-minimal theory**

Controlled Invariance Problem

- **Discrete Time System : collection *H=(X,V,Init,f)***
  - ***X*** **set of state variables**
  - ***V = (U,D)*** **set of input and disturbance variables**
  - ***Init*** **set of initial states**
  - ***f :*** $X \times V \rightarrow 2^X$ **reset relation**
- **Controlled Invariance Problem: Given a discrete time system *H*, and a set *F*** $\subset X$**, compute *W*, the maximal controlled invariant subset of *F*, and *g(x)*, the least restrictive controller**

Controlled Invariance Algorithm

initialization $W^0 = F,\ W^{-1} = \mathbf{X},\ l = 0$

while $W^{l+1} \cap \left(W^l\right)^C \neq \phi$ do

$$W^{l+1} = \mathrm{Pre}\left(W^l\right) = \left\{x \in W^l \mid \exists u \in \mathbf{U}\ \forall d \in \mathbf{D},\ f(x,u,d) \cap \left(\hat{W}\right)^C = \phi\right\}$$

$$l = l + 1$$

end while

set $\hat{W} = \bigcap_{l \geq 0} W^l$

$$\hat{g}(x) = \begin{cases} \left\{u \in \mathbf{U} \mid \forall d \in \mathbf{D}\ \ f(x,u,d) \cap \left(\hat{W}\right)^C = \phi\right\} & x \in \hat{W} \\ \mathrm{U} & x \notin \hat{W} \end{cases}$$

Implementation for Linear DTS

- $X = \Re^n$, $\mathsf{U} = \{u \mid Eu \leq \eta\}$, $\mathsf{D} = \{d \mid Gd \leq \gamma\}$, $f = \{Ax+Bu+Cd\}$, $F = \{x \mid Mx \leq \beta\}$.
- $\mathrm{Pre}(W^l) = \{x \mid \varphi^l(x)\}$

  $\varphi^l(x) = \exists u\ \forall d \mid [M^l x \leq \beta^l] \mathrm{c} [Eu \leq \eta] \wedge$

  $[(Gd > \gamma) \vee (M^l Ax + M^l Bu + M^l Cd \leq \beta^l)]$
- **Implementation**
  - **Quantifier Elimination on *d*:** **Linear Programming**
  - **Quantifier Elimination on *u*:** **Linear Algebra**
  - **Emptiness:** **Linear Programming**
  - **Redundancy:** **Linear Programming**

Implementation for Linear DTS

- **Q.E. on *d*:** $[(Gd>\gamma)\vee(M^lAx+M^lBu+M^lCd \le \beta^l)] \Leftrightarrow$
  $[M^lAx+M^lBu+\max\{M^lCd \mid Gd\le\gamma\}\le\beta^l)]$
- **Q.E. on *u*:** $[Eu\le\eta] \wedge [M^lAx+M^lBu+\delta(M^lC) \le \beta^l)] \Leftrightarrow$
  $[\Lambda^l(M^lAx+\delta(M^lC)) \le \Lambda^l\beta^l]$ where $\Lambda^lM^lB=0$, $\Lambda^lE=0$, $\Lambda^l\eta\ge0$, $\Lambda^l\ge0$
- **Emptiness** $\min\{t \mid M^lx \le \beta^l+(1...1)^Tt\} > 0$ where
  $M^l = [M^l ; \Lambda^lM^lA]$ and $\beta^l = [\beta^l ; \Lambda^l(\beta^l-\delta(M^lC))]$
- **Redundancy** $\max\{m_i^Tx \mid M^lx \le \beta^l\} \le \beta_i^l$

Decidability Results for Algorithm

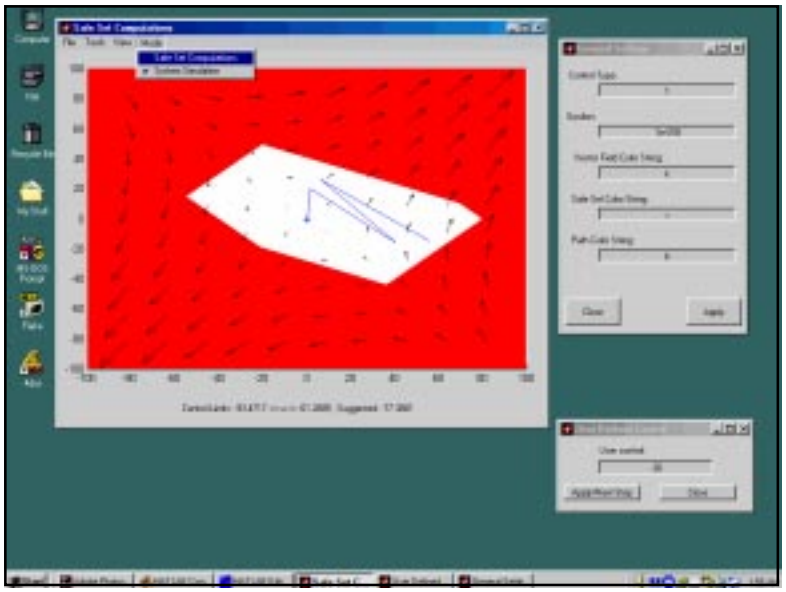
The controlled invariant set calculation problem is

- **Semi-decidable in general.**
- **Decidable when** F **is a rectangle, and** A,b **is in controllable canonical form for single input single disturbance.**

Extensions:

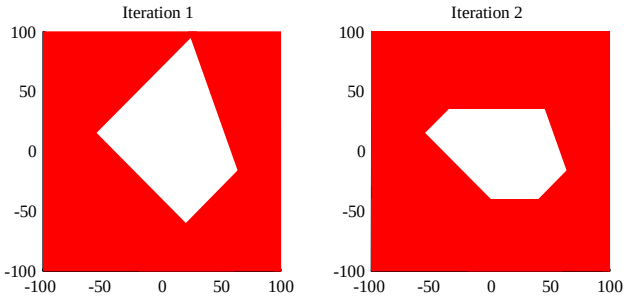
Hybrid systems with continuous state evolving according to discrete time dynamics: difficulties arise because sets may not be convex or connected.

There are other classes of decidable systems which need to be identified.



Example 1

2 states, 1 input, 1 disturbance, 4 constraints
Converges in 2 iterations



Example 2

2 states, 1 input, 1 disturbance, 4 constraints
converges in an infinite number of iterations