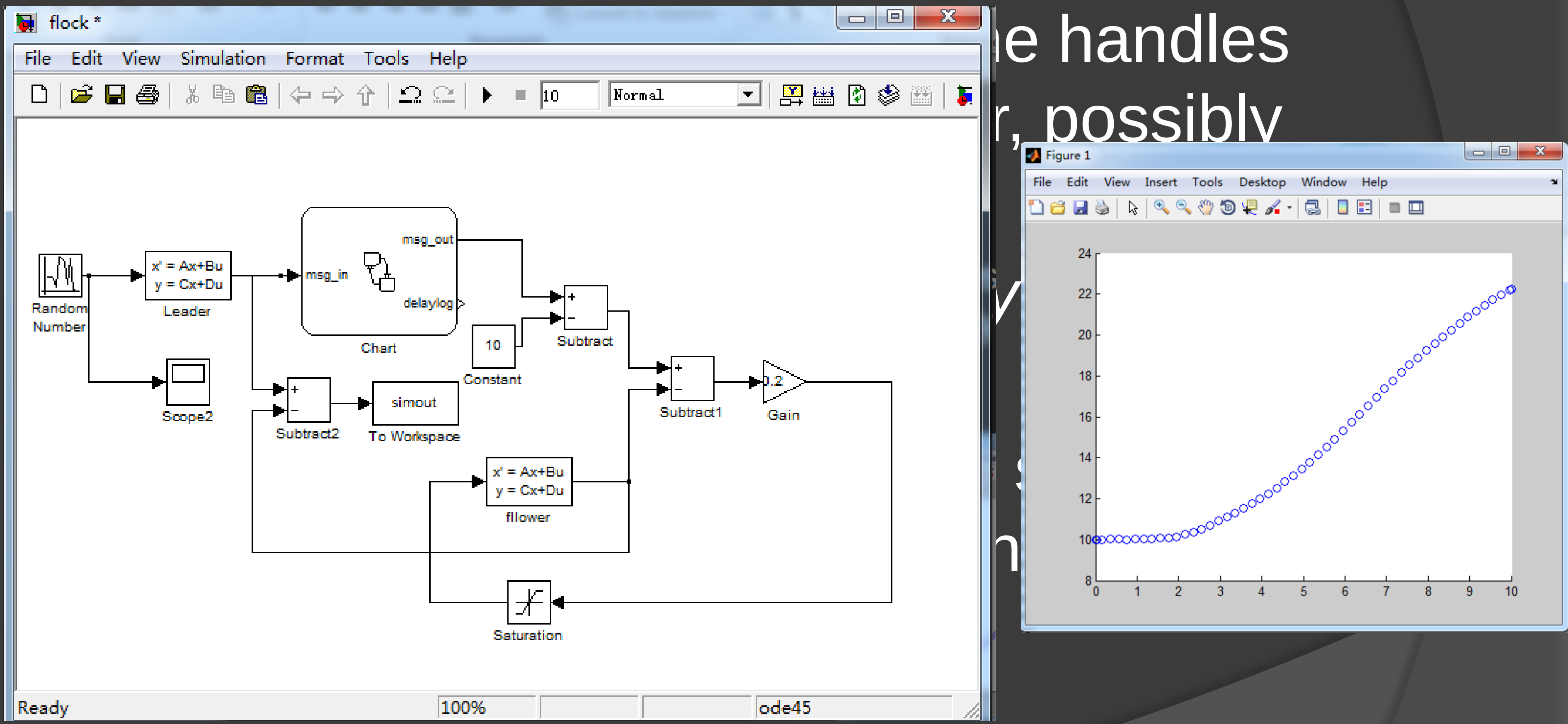
Bounded Verification of nondeterministic non-linear hybrid systems from Simulink/Stateflow Simulation

Zhenqi Huang
ECE 584 final project

MATLAB Simulink/Stateflow



e handles

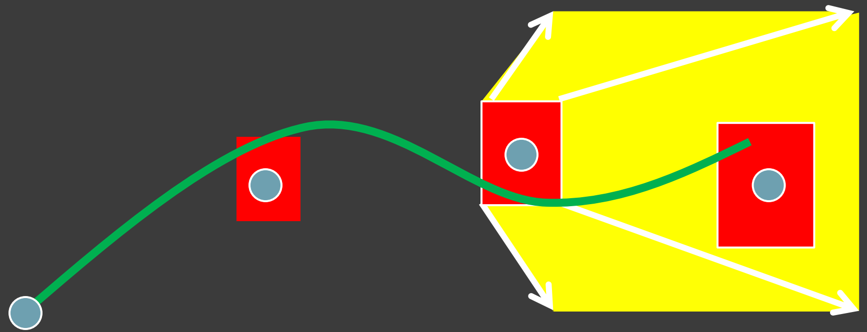
r, possibly

Simulation vs Verification

	Simulation	Verification
Sound	No	Yes
Coverage	One instance	All possible cases
Usability	Deterministic	Deterministic/Nondeterministic
Scalability	Good	Not as good
Cost	Low	High

- Simulation-based verification?

Simulation → Verification



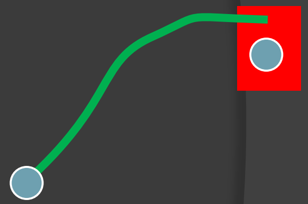
- Get a deterministic, inaccurate, and discrete simulation trace. $\beta = (v_0, t_0), (v_1, t_1), \ldots, (v_l, t_l)$
- Compute the accumulated error associated with each sample point.
  - Truncate error, approximation error, non-determinism…
- Bound the reach set between consecutive sample points.

Problem Formulation and Limitations

- System modeled as an Nondeterministic Hybrid Automaton $A =< V, L, Q, q_0, D, \mathrm{T} >$
  - $t \in V, \dot{t} = 1$ in whatever locations.
  - $loc \in L$ is associated with an Inv
  - Initial state is a single state.
  - Transition is specified with Grd and Res, guard and reset. $D = D_T \cup D_Q$, time-triggered and state-triggered transitions. For state-triggered transitions, $Res = id$ identity mapping
  - A trajectory $\tau \in \mathrm{T}$ follows a differential inclusion $\dot{\tau}.X \in F_{\tau.loc}(\tau.X)$, where $F_{loc}: \Re^n \to P(\Re^n)$.

Additional Assumptions

- Bounded stepwise numerical error.
  - $\beta = (v_0, t_0), (v_1, t_1), \ldots, (v_l, t_l)$. An execution fragment α starts at v_k, implies $|\alpha\ (t_{k+1} - t_k\ ) - v_{k+1}| \le e$.
- Bounded non-determinism.
  - $\forall loc, \forall x$, the diameter $\mathrm{D}(F_{loc}(x)) \le d$.



- Lipchitz dynamics.
  - $\exists L$, $\forall loc$, $\forall x, y$, $|F_{loc}(x) - F_{loc}(y)| \le L|x - y|$
- Bounded difference in dynamics between loc
  - $M = \sup_{x \in Inv(i) \cap Inv(j)} |F_i(x) - F_j(x)|$
- Minimum dwell time exists

Instantiation

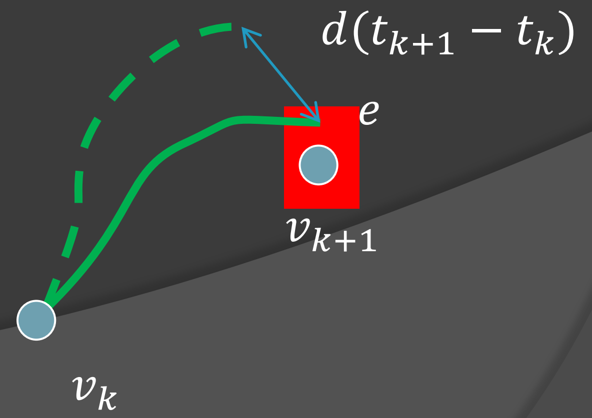
- $f_{loc}: \Re^n \rightarrow \Re^n$ is an instance of F_{loc} if $\forall x$, $f_{loc}(x) \in F_{loc}(x)$
- An deterministic hybrid automaton $A' =< V, L, Q, q_0, D, \mathrm{T}' >$ is an instance of a nondeterministic hybrid automaton $A =< V, L, Q, q_0, D, \mathrm{T} >$ if
  - A trajectory $\tau \in \mathrm{T}'$ follows a differential equation $\dot{\tau}.X = f_{\tau.loc}(\tau.X)$, where f_{loc} is an instance of F_{loc}.
- Simulation engines can handle A'

- So far we introduced the motivation and formulation of the problem, in addition with a set of assumptions on the model

- Next we will discuss the approach to compute the reach set of a nondeterministic hybrid system *A* given a simulation trace β of its instance *A'*.

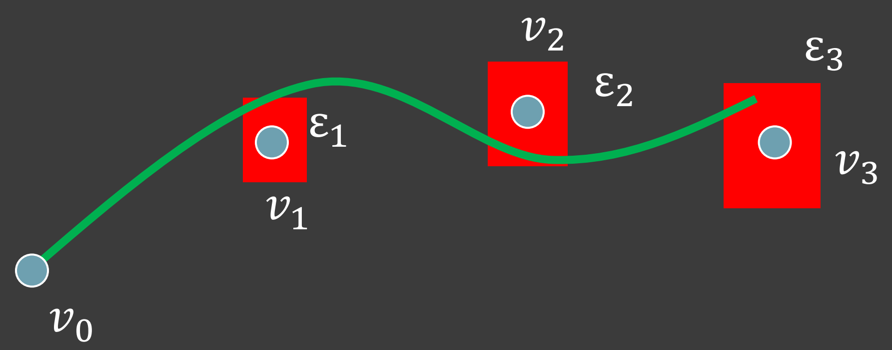
Stepwise Error

- From the assumptions, we can control the stepwise error.
- Encode the numerical error and non-determinism as stepwise error $c_k = e + d(t_{k+1} - t_k)$.
- All possible execution fragments start at v_k should be within distance c_k from v_k after a period $t_{k+1} - t_k$



Accumulated Error

- Denote $\varepsilon_k = \sup_{\alpha} |\alpha(t_k) - v_k|$ be the accumulated error between all admissible execution α and sample point v_k



Accumulated Error

- If no transition takes place in $[t_k, t_{k+1}]$,

$$\varepsilon_{k+1} = \varepsilon_k e^{L\,(t_{k+1}-t_k)} + c_k.$$

  - $\forall loc, \forall x, y, |F_{loc}(x) - F_{loc}(y)| \leq L|x - y|.$
- If one transition takes place in $[t_k, t_{k+1}]$

$$\varepsilon_{k+1} = \varepsilon_k e^{L\,(t_{k+1}-t_k)} + \frac{M}{L}(e^{L(t_{k+1}-t_k)} - 1) + c_k$$

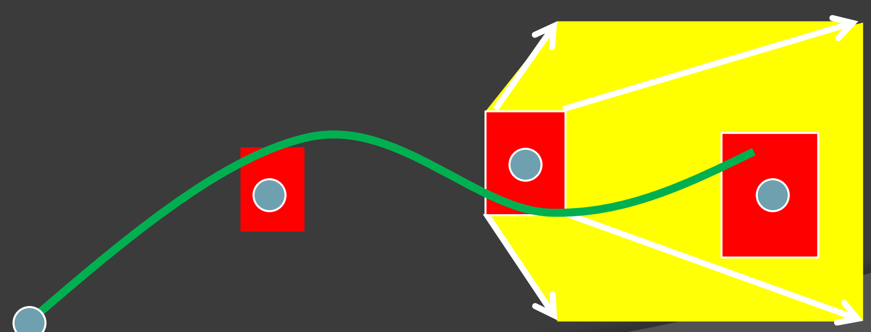
  - Where, $M = \sup_{x \in Inv(i) \cap Inv(j)} |F_i(x) - F_j(x)|$

- Proofs in [Computing Bounded Reachset from Sampled Simulation Trace] in proceedings of HSCC 2012'

Propagation between sample points

- Fixed point computation.

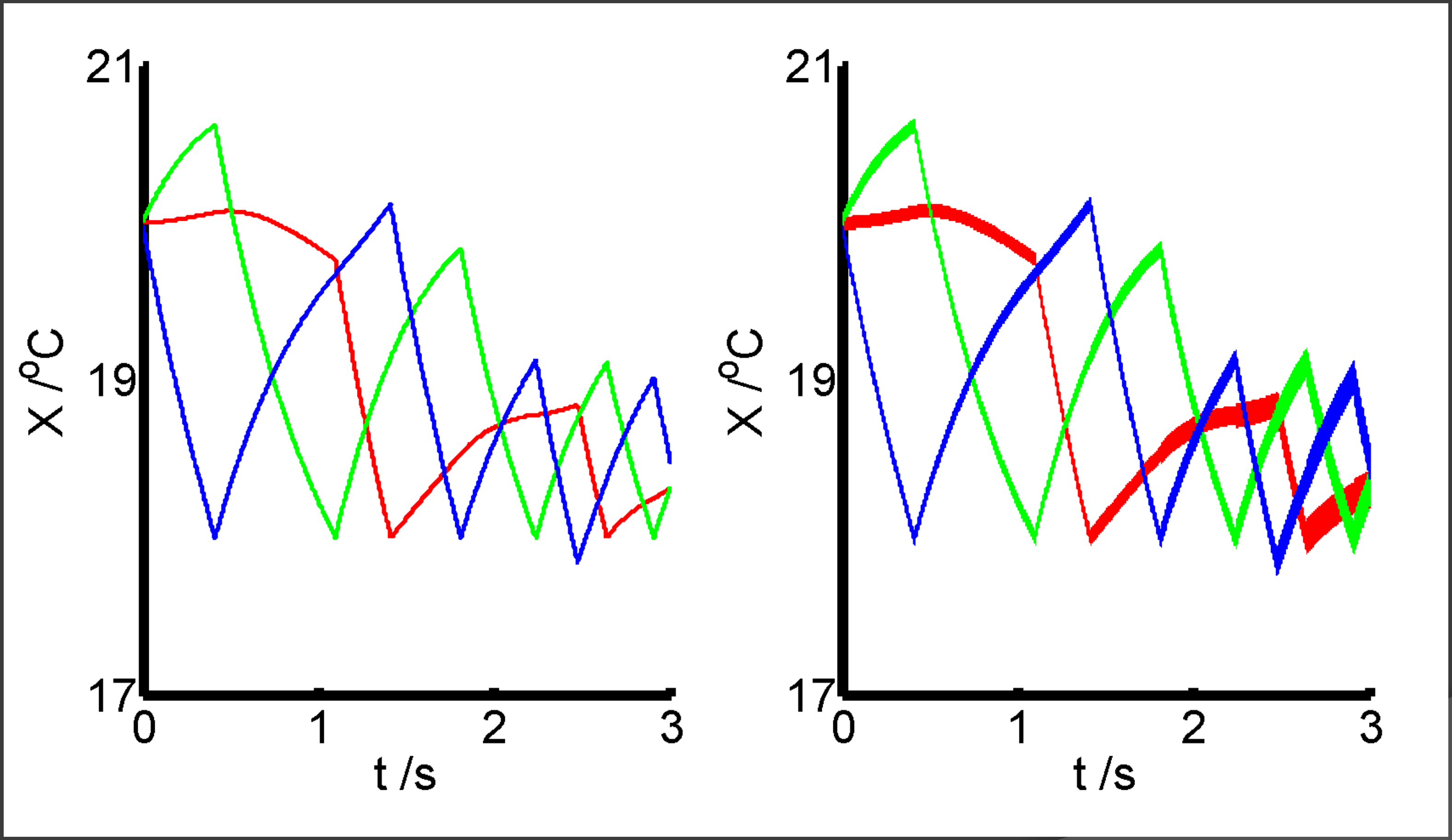
$\sigma \leftarrow \epsilon_k$;
do
  $\sigma \leftarrow b\sigma \quad \backslash\backslash b > 1$ is a constant;
  $B \leftarrow Ball(\mathbf{v}_k.X, \sigma)$;
  $m \leftarrow \sup_{X \in B} ||f(X)||$;
while $\sigma - m\delta < \epsilon_k$;



Case Study I: Room Heating

- There are 3 rooms heated by 2 heater.
- Heaters can move from one room to another.
- The continuous variables (x_1, x_2, x_3) capture the temperature of the three rooms.
- The discrete transitions capture how heaters move. A heater moves from room i to room j if
  - If room i has a heater and room j does not,
  - $x_i - x_j > 1$, and
  - $x_j \leq 18$
- The safety property of interest is that the temperature of all rooms stay above a threshold, say 17C.

Case Study I: Room Heating



Case Study II: delayed flocking

- Two robots move on a line . One leader one follower.
  - The leader moves with acceleration in [-0.2, 0.2]. The follower tries to maintain the separation to be 10.
  - Every 0.2s, the leader send a message containing its current position and velocity to the follower.
  - The message get delayed by $d \in [0.05, 0.1]$.
  - The follower updates its controller once a msg arrives.
- We want to check whether the two robots collide, say $x_1 - x_2 \le 5$.

$$\dot{x_2} = v_2$$
$$\dot{v_2} = f(msg, x_1, x_2)$$

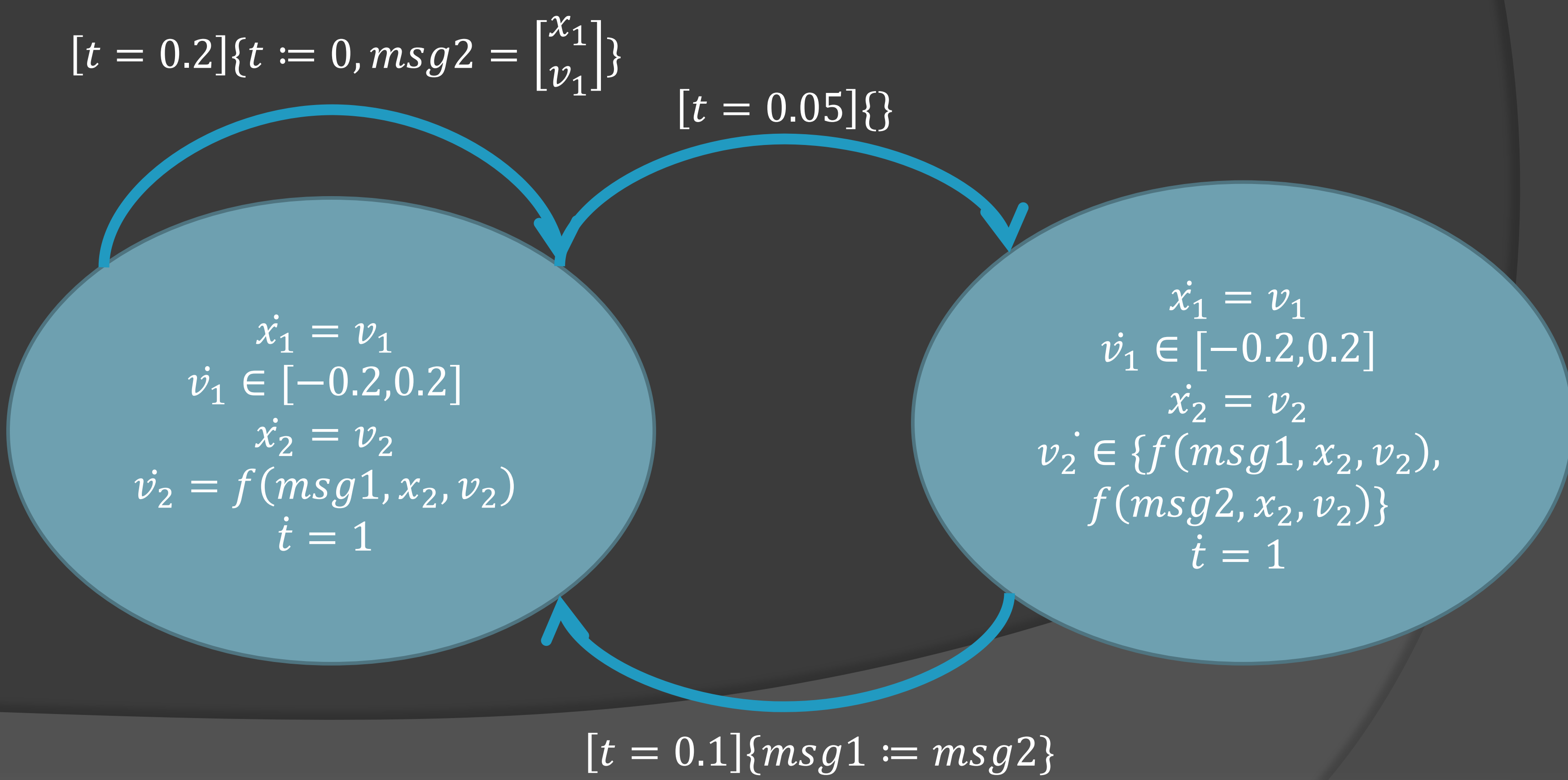
$$\dot{x_1} = v_1$$
$$\dot{v_1} \in [-0.2, 0.2]$$

Every 0.2s send msg

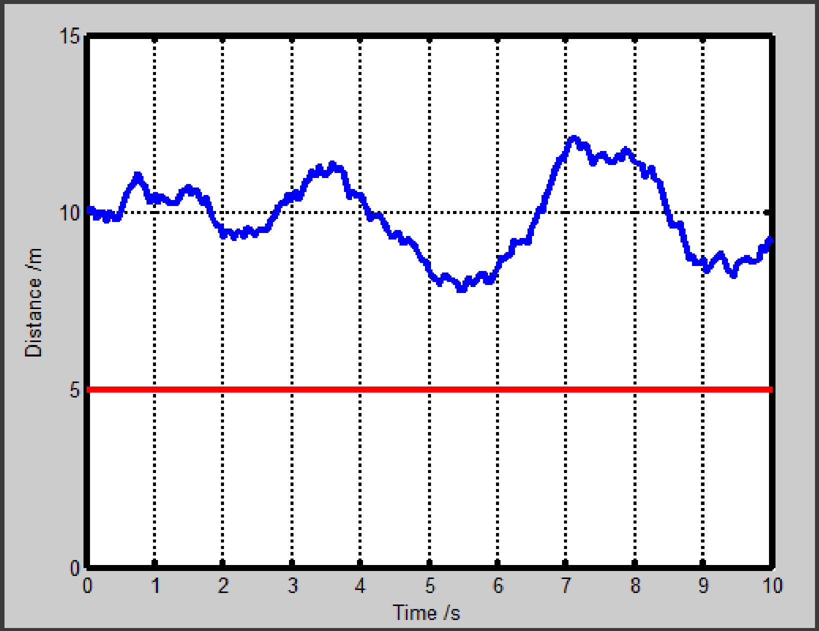
Get a delay in $[0.05, 0.1]$

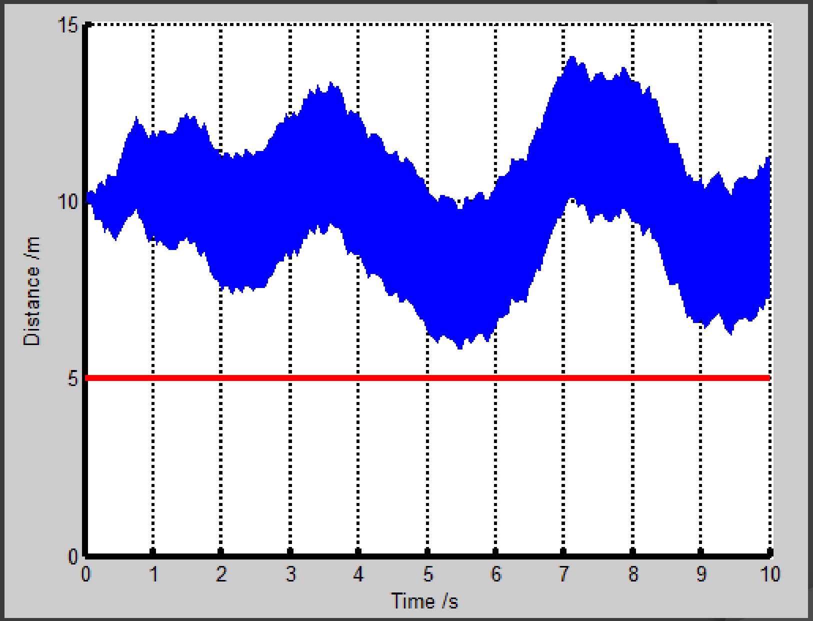
Case Study II: delayed flocking

- We encode the problem as the following hybrid automaton
  - Variable includes $x_1, v_1, x_2, v_2, t, msg1, msg2$



Case Study II: delayed flocking





Conclusion

- A approach to verify safety given simulation trace and model specification
- Handles nondeterministic nonlinear hybrid systems

- I am glad to answer any of your questions.