Contents of the lecture:

- Intro: Incorporating continuous dynamics & sources of computational complexity
- Recall: Receding horizon control
- Receding horizon temporal logic planning (RHTLP)
  - Basic idea & main result
  - Discussion of the key details of implementation
- Autonomous driving examples
- Finite-state abstraction & hierarchical control architecture
Problem: Design control protocols, that...

Handle mixture of discrete and continuous dynamics

Account for both high-level specs and low-level constraints

Reactively respond to changes in environment,

... with “correctness certificates.”

\[
(\phi_{\text{init}} \land \phi_{\text{env}}) \rightarrow (\phi_{\text{safety}} \land \phi_{\text{goal}})
\]
Hierarchical control architecture

**TuLiP**: Temporal logic planning toolbox
(Open source at [http://tulip-control.sf.net](http://tulip-control.sf.net))

[Coming up in the next lecture]
This lecture focuses on two of the remaining issues:

- Incorporating continuous dynamics
- Computational complexity
• Each of these cells may be occupied by an obstacle.
• The vehicle can be in any of these cells.

\[(2L)(2^{2L})\text{ possible states!}\]
Receding Horizon Control

\[ \min_{u[t, t+T]} \int_{t}^{t+T} C(x(\tau), u(\tau)) d\tau + V(x(t + T)) \]

subject to:
\[ \dot{x} = f(x, u), \quad x(t) \text{ given} \]
\[ x(t + T) = x_f, \quad g(x, u) \leq 0 \]

• Reduces the computational cost by solving smaller problems.
• Real-time (re)computation improves robustness.
Receding Horizon Control

• If not implemented properly, global properties, e.g., stability, are not guaranteed.
• Increasing $T$ helps for stability at the expense of increased computational cost.

• If the terminal cost is chosen as a control Lyapunov function, i.e., $V$ is (locally) positive definite and satisfy (for some $r>0$)

$$\min_{u(t,t+T)} \left( \int_{t}^{t+T} C(x(\tau),u(\tau)) d\tau + V(x(t+T)) \right)$$

subject to:

$$\dot{x} = f(x,u), \quad x(t) \text{ given}$$

$$x(t+T) = x_f, \quad g(x,u) \leq 0$$

$$\min_{u} (\dot{V} + C)(x,u) < 0, \quad \forall x \in \{ x : V(x) \leq r^2 \}$$

then stability is guaranteed.

• Alternative (related) approach, imposed contractiveness constraints in short-horizon problems.
Receding Horizon for LTL Synthesis

Global (long-horizon) specification:

\[(\varphi_{\text{init}} \land \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \land \varphi_{\text{goal}})\]

Basic idea:

- Partition the state space into a partially ordered set \((\{W_j\}, \preceq_{\varphi_g})\)
- Goal-induced partial order

Short-horizon specification: For each \(i\),

\[(\nu \in W_i) \land \Phi \land \varphi_{\text{env}} \rightarrow (\Box \Phi \land \varphi_{\text{safety}} \land \Diamond (\nu \in F_i(W_i)))\]

Plan from the current cell on

Receding horizon invariant: rules out “corner” cases

Get closer to goal rather than reaching. \(F\): “horizon” length

Theorem: Receding horizon implementation of the short-horizon strategies ensures the correctness of the global specification.

Trade-offs:

- Computational cost vs. horizon length vs. strength of invariant vs. conservatism

state satisfying \(\varphi_{\text{goal}}\)
How to come up with a partial order, $\mathcal{F}$ and $\Phi$?

• In general, problem-dependent and requires user guidance.
• Partial automation is possible (discussed later).
• Partial order: “measure of closeness” to the goal, i.e., to the states satisfying.
• The map $\mathcal{F}$ determines the “horizon length.

- The invariant $\Phi$ (in this example) rules out the states that render the short horizon problems unrealizable.
- In the example above, it is the conjunction of the following propositional formulas on the initial states for each subproblem:
  • no collision in the initial state
  • vehicle cannot be in the left lane unless there is an obstacle in the right lane in the initial state
  • vehicle is able to progress from the initial state

\[
\mathcal{W}_0 \prec \ldots \prec \mathcal{W}_{L-1} \prec \mathcal{W}_L
\]\
\[
\mathcal{F}(\mathcal{W}_j) = \mathcal{W}_{j-2}, \quad j \geq 2
\]\
\[
\mathcal{F}(\mathcal{W}_j) = \mathcal{W}_0, \quad j < 2
\]
Navigation of point-mass omnidirectional vehicle

nondimensionalized dynamics:
\[ \ddot{x} + \dot{x} = q_x(t) \]
\[ \ddot{y} + \dot{y} = q_y(t) \]
\[ \dddot{\theta} + \frac{2mL^2}{J} \dot{\theta} = q_\theta \]

conservative bounds on control authority to decouple the dynamics:
\[ |q_x(t)|, |q_y(t)| \leq \sqrt{0.5} \]
\[ |q_\theta(t)| \leq 1 \]

Partition (in two consecutive cells):

Reasons for the non-intuitive trajectories:
- Synthesis: feasibility rather than “optimality.”
- Specifications are not rich enough.
Example: Navigation In Urban-Like Environment

Dynamics: \( \dot{x}(t) = u_x(t) + d_x(t), \quad \dot{y}(t) = u_y(t) + d_y(t) \)

Actuation limits: \( u_x(t), u_y(t) \in [-1, 1], \quad \forall t \geq 0 \)

Disturbances: \( d_x(t), d_y(t) \in [-1, 1], \quad \forall t \geq 0 \)

Traffic rules:
- No collision
- Stay in right lane unless blocked by obstacle
- Proceed through intersection only when clear

Environment assumptions:
- Obstacle may not block a road
- Obstacle is detected before it gets too close
- Limited sensing range (2 cells ahead)
- Obstacle does not disappear when the vehicle is in its vicinity
- Obstacles don’t span more than certain # of consecutive cells in the middle of the road
- Each intersection is clear infinitely often
- Cells marked by star and adjacent cells are not occupied by obstacle infinitely often

Goals: Visit the cells with *’s infinitely often.
Navigation In Urban-Like Environment

Setup:
- **Dynamics**: Fully actuated with actuation limits and bounded disturbances
- **Specifications**:
  - Traffic rules
  - Assumptions on obstacles, sensing range, intersections,...
- **Goals**: Visit the two stars infinitely often

Results:
- **Without receding horizon**: $1e87$ states (hence, not solvable)
- **Receding horizon**:
  - Partial order: From the top layer of the control hierarchy
  - Horizon length $= 2$ \( (F(\mathcal{W}_j^i) = \mathcal{W}_{j-2}^i) \)
  - Invariant: Not surrounded by obstacles. If started in left lane, obstacle in right lane.
  - $1e4$ states in the automaton.
  - $\sim 1.5$ sec for each short-horizon problem
  - Milliseconds for partial order generation
What is $\Phi$?

- A propositional formula (that we call receding horizon invariant).
- Used to exclude the initial states that render synthesis infeasible, e.g., states from which collision is unavoidable.

Short-horizon specification:

$$(\nu \in \mathcal{W}_i) \land \Phi \land \varphi_{env} \rightarrow (\square \Phi \land \varphi_{safety} \land \Diamond (\nu \in \mathcal{F}_i (\mathcal{W}_i)))$$

Given partial order and $\mathcal{F}$, computation of the invariant can be automated:

- Check realizability
- If realizable, done.
- If not, collect violating initiation conditions. Negate them and put in $\bar{\Phi}$.
- Repeat until all subproblems or all possible states are excluded (in the latter case, either the global problem is infeasible or RHTLP with given partial order and $\mathcal{F}$ is inconclusive.)
Generalization to multiple “goals”

Each partial order covers the discrete (system) state space. For each \( \nu \in \mathcal{W}_0^{i,j} \), one can find a cell in the “proceeding” partial order that \( \nu \) belongs to.

**Strategy:** While in \( \mathcal{W}_j^i \) implement (in a receding horizon fashion) the controller that realizes

\[
(\nu \in \mathcal{W}_j^i) \land \Phi \land \square \psi_e^c \land \bigwedge_{k \in I_f} \square \Diamond \psi_{f,k}^c, \quad \land_{\nu \in \mathcal{F}^i(\mathcal{W}_j^i)} \land \square \Phi
\]
Computational complexity & completeness

For Generalized Reactivity [1] formulas, the computation time of synthesis is $O(mn|\Sigma|^3)$, where $|\Sigma|$ is the number of discrete states.

Receding horizon implementation...
- reduces the computational complexity by restricting the state space considered in each subproblem; and
- is not complete, i.e., the global problem may be solvable but the choice of $\{W_j\}$, the partial order, the maps $F_i$, and $\Phi$ may not lead to a solution.

Choose $F_i$ to give “longer horizon”:
- Subproblems in RHTLP are more likely to be realizable.
- Computational cost is higher.

E.g., for urban-like driving example is infeasible with horizon length of one.

Global synthesis problem
\[(\varphi_{\text{init}} \land \varphi_{\text{env}}) \rightarrow (\varphi_{\text{safety}} \land \varphi_{\text{goal}})\]

Subproblems in RHTLP
\[((v \in W_i) \land \Phi \land \varphi_{\text{end}}) \rightarrow (\varphi_{\text{safety}} \land \Box (v \in F_i(W_i) \land \Box \Phi))\]
Response mechanism is introduced to compensate for mismatch between the system and its model and between the actual behavior of the environment and its assumptions.
Theorem: For any discrete run satisfying the specification, there exists an admissible control signal leading to a continuous trajectory satisfying the specification.

Proof: Constructive → Finite-state model + Continuous control signals.

Abstraction refinement for reducing potential conservatism.
**Finite state abstraction**

**Given:**
- A system with controlled variables $s \in S$ in domain $\text{dom}(S)$ and environment variables $e \in E$ in domain $\text{dom}(E)$.
- Define $v = (s, e)$, $V = S \cup E$ and $\text{dom}(V) = \text{dom}(S) \times \text{dom}(E)$.

- Controlled variables evolve with (for $t = 0, 1, 2, ...$):

  \[
  s[t + 1] = A s[t] + B_u u[t] + B_d d[t] \quad \text{state evolution}
  \]

  \[
  u[t] \in U \quad \text{admissible control inputs}
  \]

  \[
  d[t] \in D \quad \text{exogenous disturbances}
  \]

  \[
  s[0] \in \text{dom}(S) \quad \text{set that states take values in}
  \]

  \[
  s[t + 1] \in \text{dom}(S)
  \]

- System specification $\varphi$

**Find:** A finite transition system with discrete states $\nu$ such that for any sequence $\nu_0 \nu_1 \ldots$ satisfying $\varphi$, (very roughly speaking) there exists a sequence of admissible control signals leading to an infinite sequence $\nu_0 \nu_1 \nu_2 \ldots$ that satisfies $\varphi$.

(stated more precisely later...)
Proposition preserving partition

Given $\text{dom}(V)$ and atomic propositions in $\Pi$.

A partition of $\text{dom}(V)$ is said to be proposition preserving if, for any atomic proposition $\pi \in \Pi$ and any states $\nu$ and $\nu'$ that belong to the same cell of the partition, $\nu$ satisfies $\pi$ if and only if $\nu'$ satisfies $\pi$.

Example: $\Pi = \{x \leq 1, y \geq 0, x + y \geq 0, \ldots\}$

A discrete state $\nu$ is said to satisfy $\pi$ if and only if there exists a continuous state $\nu'$, in the cell labeled, that satisfies $\pi$.

\[
\nu_5 \models_d \pi \iff \exists \nu \in \nu_5 \text{ s.t. } \nu \models \pi
\]

proposition preserving:

\[
\nu \models \pi \iff \nu' \models \pi
\]
Finite-time reachability

A discrete state \( \nu_j \) is finite-time reachable from a discrete state \( \nu_i \), only if starting from any \( s[0] \in T_{s}^{-1}(\nu_i) \), there exists - a finite horizon length \( N \in \{0, 1, \ldots\} \)
- for any allowable disturbance, there exists
  \( u[0], u[1], \ldots, u[N - 1] \in U \) such that

\[
\begin{align*}
  s[N] &\in T_{s}^{-1}(\nu_j) \\
  s[t] &\in T_{s}^{-1}(\nu_i) \cup T_{s}^{-1}(\nu_j), \quad \forall t \in \{0, \ldots, N\}
\end{align*}
\]

Verifying the reachability relation:
• Compute the set \( S_0 \) of \( s[0] \) from which \( T_{s}(\nu_j) \) can be reached under the system dynamics in a pre-specified time \( N \).
• Check whether \( T_{s}^{-1}(\nu_i) \subseteq S_0 \).

\[
\text{system dynamics } \left\{ \begin{array}{l}
  s[t + 1] = As[t] + Bu[t] + Bd[t] \\
  u[t] \in U \\
  d[t] \in D \\
  s[0] \in \text{dom}(S) \\
  s[t + 1] \in \text{dom}(S)
\end{array} \right.
\]
Computing $S_0$

Given $N$ and polyhedral sets
\[ T_{s}^{-1}(\nu_i) = \{ s \in \mathbb{R}^n : L_1 s \leq M_1 \} \]
\[ U = \{ u \in \mathbb{R}^m : L_2 u \leq M_2 \} \]
\[ T_{s}^{-1}(\nu_j) = \{ s \in \mathbb{R}^n : L_3 s \leq M_3 \} \].

$S_0$ is computed as the set of $s_0$ such that there exist $u[0], \ldots, u[N-1]$ satisfying $L_2 u[t] \leq M_2$, for $t \in \{0, \ldots, N-1\}$, leading to
\[ L_1 s[t] \leq M_1 \text{ for } t = 0, \ldots, N-1 \]
\[ L_3 s[N] \leq M_3, \]
where
\[ s[t] = A^t s_0 + \sum_{k=0}^{t-1} (A^k B_u u[t-1-k] + A^k B_d d[t-1-k]) \]
for all $d[0], \ldots, d[N-1] \in D$ ($D$ polyhedral).

Put together: $S_0$ is computed as a polytope projection:
\[ S_0 = \left\{ s_0 \in \mathbb{R}^n : \exists \hat{u} \in \mathbb{R}^{mN} \text{ s.t. } L \begin{bmatrix} s_0 \\ \hat{u} \end{bmatrix} \leq M - G \hat{d}, \ \forall \hat{d} \in \bar{D}^N \right\} \]
stacking of $u$ and $d$
set of vertices of $D^N = D \times \cdots \times D$.
Define the finite transition system $\mathcal{D}$, an abstraction of $\mathcal{S}$ as:
- $\mathcal{V} := \mathcal{S} \times \mathcal{E}$, set of discrete states (both controller and environment)
- $\nu_i = (\varsigma_i, \epsilon_i) \rightarrow \nu_j = (\varsigma_j, \epsilon_j)$ only if $\varsigma_j$ is reachable from $\varsigma_i$.

**Refining the partition**

While checking the reachability from $T_s^{-1}(\nu_i)$ to $T_s^{-1}(\nu_j)$, if $T_s^{-1}(\nu_i) \not\subseteq S_0$, then
- Split $T_s^{-1}(\nu_i) \cap S_0$ and $T_s^{-1}(\nu_i) \cap S_0^c$
- Remove $\nu_i$ from the set of discrete states
- Add two new discrete states corresponding to $T_s^{-1}(\nu_i) \cap S_0$ and $T_s^{-1}(\nu_i) \cap S_0^c$

- Repeat until no cell can be sub-partitioned s.t. the volumes of the two resulting new cells both greater than $Vol_{\min}$.
- Smaller $Vol_{\min}$ leads to more cells in the partition and more allowable transitions.
- If the initial partition is proposition preserving, so is the resulting.
Correctness of the hierarchical implementation

**Using**

- Proposition preserving property of the partition
- \( \mathbb{D} \) only includes the transitions that are implemented by the control signal \( u \) within some finite time (by construction through the reachability formulation)
- Stutter invariance of the specification \( \varphi \), ...

Two words \( \sigma_1 \) and \( \sigma_2 \) over \( 2^{AP} \) are stutter equivalent, if there exists an infinite sequence \( A_0 A_1 A_2 \ldots \) of sets of atomic propositions and natural numbers \( n_0, n_1, n_2, \ldots \) and \( m_0, m_1, m_2, \ldots \) such that \( \sigma_1 \) and \( \sigma_2 \) are of the form

\[
\sigma_1 = A_0^{n_0} A_1^{n_1} A_2^{n_2} \ldots \quad \sigma_2 = A_0^{m_0} A_1^{m_1} A_2^{m_2} \ldots
\]

An LT property \( P \) is stutter-invariant if for any word \( \sigma \in P \) all stutter-equivalent words are also contained in \( P \).

Example: \( \nu_0 \nu_1 \ldots \nu_8 \ldots \) and \( \nu_0 \nu_1 \ldots \) are stutter-equivalent.

**...we can prove:**

Let \( \sigma_d = \nu_0 \nu_1 \ldots \) be a sequence in \( \mathbb{D} \) with \( \nu_k \rightarrow \nu_{k+1}, \nu_k = (\varsigma_k, \epsilon_k), \varsigma_k \in S \) and \( \epsilon_k \in \mathcal{E} \). If \( \sigma_d \models_d \varphi \), then by applying a sequence of control signals from the Reachability Problem with initial set \( T_s^{-1}(\varsigma_k) \) and final set \( T_s^{-1}(\varsigma_{k+1}) \), the sequence of continuous states \( \sigma = \nu_0 \nu_1 \nu_2 \ldots \) satisfies \( \varphi \).
TuLiP: Temporal logic planning toolbox
(Open source at http://tulip-control.sf.net)

[Coming up in the next lecture]