EN.530.678: Nonlinear Control and Planning in Robotics

Lecture# 13
Receding Horizon Planning

April 28, 2022

Planning/Control Architecture

Global Planning Methods:

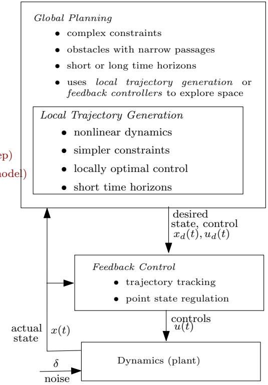
- discrete space search methods (e.g. A*)
- tree/graph sampling-based methods
- stochastic trajectory optimization (global models)
- value function approximation methods
- relaxation methods (linearize/convexify)

Local Planning Methods:

- exploit structure: nonholonomy, symmetries, flatness
- gradient-based trajectory optimization (SQP, IP,sweep)
- stochastic trajectory optimization (local stochastic model)

Feedback Control Methods:

- feedback linearization
- Lyapunov design
- linearization-based control



Update Speed

slow (e.g. 1-10Hz)

medium (e.g. 10-100 Hz)

real-time (e.g. 100Hz)

- *Global planning*: handle complex constraints, long-time horizons; generate subgoals
- *Local trajectory generation*: optimally achieve subgoals and satisfy dynamics
- *Feedback control*: handle noise/disturbances and execute desired trajectory
- *Receding Horizon Control*: recompute reference trajectory in real-time

Receding Horizon Control

Global Planning Methods:

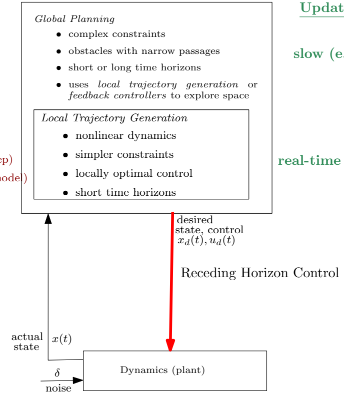
- discrete space search methods (e.g. A*)
- tree/graph sampling-based methods
- stochastic trajectory optimization (global models)
- value function approximation methods
- relaxation methods (linearize/convexify)

Local Planning Methods:

- exploit structure: nonholonomy, symmetries, flatness
- gradient-based trajectory optimization (SQP, IP,sweep)
- stochastic trajectory optimization (local stochastic model)

Feedback Control Methods:

- feedback linearization
- Lyapunov design
- linearization-based control



Update Speed

slow (e.g. 1-10Hz)

real-time (e.g. 100Hz)

- *Global planning*: handle complex constraints, long-time horizons; generate subgoals
- *Local trajectory generation*: optimally achieve subgoals and satisfy dynamics
- *Feedback control*: handle noise/disturbances and execute desired trajectory
- *Receding Horizon Control*: recompute reference trajectory in real-time

RHC: methodology

Consider a task involving

- a long time-horizon, or long-time periodic tasks
- unmodeled disturbances move system away from desired path
- performance can be greatly improved by recomputing reference

RHC approach

- Solve optimization over a short horizon T, the cost is:

$$J_T^*(x(t), u(\cdot)) = \min_{u(\cdot)} \int_t^{t+T} L(x(\tau), u(\tau))d\tau + V(x(t+T)),$$

  where $L(x, u)$ is the *incremental cost* and $V(x)$ is the *terminal cost*
- The cost V accounts for the "tail" of the hirozon
- RHC idea: optimize J over small T but carefully choose V to guarantee stability!

Stability Issues

- If T is the original long-horizon then optimization is too expensive
- For shorter T, J can be optimized in real-time, then stability depends on V

V is an estimate of the optimal cost-to-go

- it is generally unavailable, i.e. we do not know $V(x) = J^*_\infty(x)$
- V should "measure" the total accrued cost $L(x)$ along the "tail"
- that cost must be driven to zero

RHC approach: *V is chosen as an appropriate Lyapunov Function*, i.e. RHC subsumes a tracking/regulation problem inside the optimal control formulation.

RHC Stability Theorem

[Jadbabaie and Hauser, 2002] Suppose that the terminal cost $V(x)$ is a CLF s.t.

$$\min_u(\dot{V} + L)(x, u) \leq 0$$

for each x in $\Omega_r = \{x : V(x) < r^2\}$. Then, for every $T > 0$ and $\delta \in (0, T]$, the RHC trajectories reach the goal exponentially fast.

- Meaning: V should decrease at least as fast as the accrued cost L
- V is difficult to find: currently standard appraoch is to linearize around reference and use LQR, i.e. set $V = \frac{1}{2}x^T Px$, where P is the solution to the Ricatti equation.

Proof

- Let $x^u(\tau, x)$ denote the state trajectory at time τ starting from x after applying control $u(\cdot)$
- Let $(x^*_T, u^*_T)(\cdot, x)$ denote the optimal trajectory of the finite horizon OC problem with hirozon T
- Assume $x^*_T(T, x) \in \Omega_r = \{x : V(x) < r^2\}$ for some $r > 0$. Then, for each $\delta \in (0, T]$, our notion of stability is understood as the following condition: the optimal cost from $x^*_T(\delta, x)$ must satisfy

$$J^*_T(x^*_T(\delta; x)) \leq J^*_T(x) - \int_0^\delta L\left(x^*_T(\tau; x), u^*_T(\tau; x)\right) d\tau$$

- In other words, the optimal cost is constantly decreasing (converging) so that the state will remain in the region of attraction of V.
- Proving this condition is equivalent to proving stability.

Proof (cont)

- Let $(\widetilde{x}(t), \widetilde{u}(t))$, $t \in [0, 2T]$ obtained by concatenating $(x_T^*, u_T^*)(t, ; x)$, $t \in [0, T]$ and $(x^k, u^k)(t - T; x_T^*(T; x))$, $t \in [T, 2T]$ which are the closed-loop trajectories with $u = k(x)$ such that $(\dot{V} + L)(x, k(x)) \leq 0$.
- Consider the cost of using $\widetilde{u}(\cdot)$ for time T, starting at $x_T^*(\delta; x)$, $\delta \in [0, T]$

$$
\begin{aligned}
J_T(x_T^*(\delta; x), \widetilde{u}(\cdot)) &= \int_\delta^{T+\delta} L(\widetilde{x}(\tau), \widetilde{u}(\tau)) d\tau + V(\widetilde{x}(T+\delta)) \\
&= J_T^*(x) - \int_0^\delta L\left(x_T^*(\tau; x), u_T^*(\tau; x)\right) d\tau - V\left(x_T^*(T; x)\right) \\
&\quad + \int_T^{T+\delta} L\left(\widetilde{x}(\tau), \widetilde{u}(\tau)\right) d\tau + V(\widetilde{x}(T+\delta)) \\
&\leq J_T^*(x) - \int_0^\delta L\left(x_T^*(\tau; x), u_T^*(\tau; x)\right) d\tau,
\end{aligned}
$$

using the fact that

$$
L(\widetilde{x}(\tau), \widetilde{u}(\tau)) \leq -\dot{V}(\widetilde{x}(\tau), \widetilde{u}(\tau)), \text{ for all } \tau \in [T, 2T]
$$

The proof then follows, since $J_T^*(x_T^*(\delta; x)) \leq J_T(x_T^*(\delta; x), \widetilde{u}(\cdot))$.