Lecture #10

Goals

1) Recap
2) Sampling Based Methods
3) Summary of State of the Art
4) Frontiers

I. Recap

Strategy

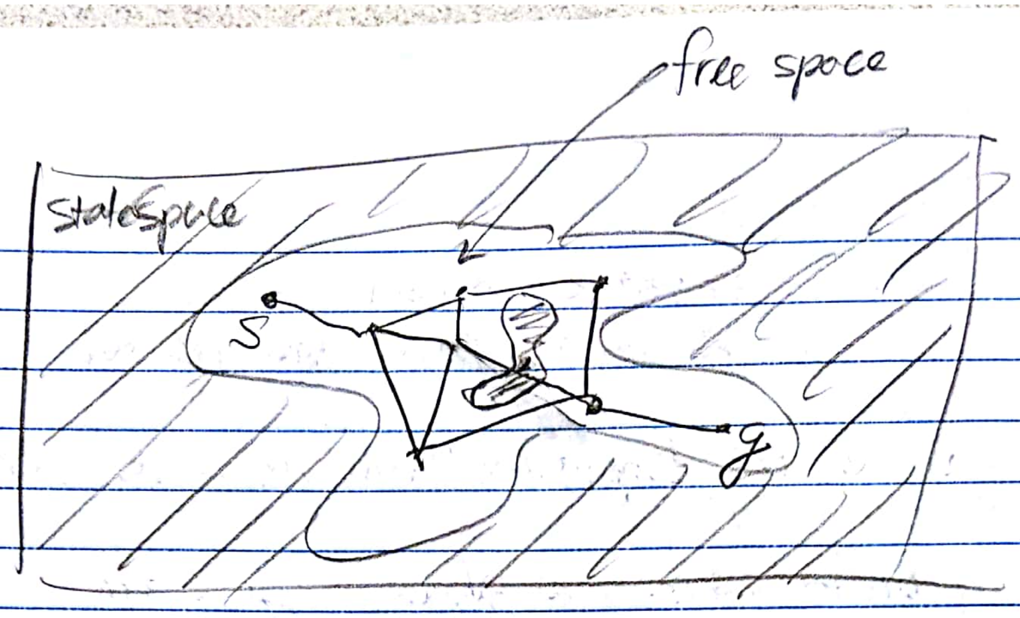
1) Given a map, apply nonlinear optimization to design a control around the track
2) Avoid obstacles only known at runtime, or deal w/ discretization error, by using model predictive control to design feedback about nominal trajectory.

Warning

1) ensure gradients are correct
2) avoid choosing large time discretization
3) SCALE CORRECTLY - gradients are larger than the cost, then solver will struggle

II. Sampling Based Methods

A. Probabilistic Road Maps (PRMs)



1) Sample states from state space at random
2) Remove those not in free space
Retained samples are called milestones
3) Each milestone is linked by straight line to nearest neighbor.
7) Only collission free paths are retained.
4) Apply graph search to find path from start to goal.

Note: straight line makes collision check easy

Challenges
1) When do we have enough samples?
a) need density to be high enough otherwise all straight line paths will intersect w/ obstacles.
b) sampling w/ same density as state space dimension increases requires exponentially more samples
2) How to generate control that draws straight line path between milestones?
a) samples are close, linearize and apply LQR/MPC?
b) samples are faraway, use nonlinear optimization.

3) How to check collisions?
Sample along path and check...

B. Rapidly Exploring Random Trees (RRTs)

Build a tree to generate next state
(rather than sample all of state space)

Algorithm:
Given: x_{init}, K ← # of samples
add x_{init} to G ← graph
for $k = 1$ to K
1. generate random state x_{rand}
2. find nearest point x_{near} in G to x_{rand}
3. Select an input u that gets as close to x_{rand} from x_{near}, if it does not intersect with anything then add x_{new} to graph
end.

Challenges
1. how do we sample to ensure we get to goal?
↳ sample in a biased fashion
2. how to find nearest neighbor?
↳ efficient data structures like k-dimensional trees.
3. how to select input in step 3?
↳ nonlinear optimization, LQR
↳ sample over input space?

4. how to check constraints satisfied?
↳ sample over path.

In general, sampling techniques are popular but have to tackle a hard control and constraint check problem so some usually still apply MPC

III Summary of Methods

	Dynamics	Constraints	Feedback	Real-Time?
Ackermann	LTI	no	yes	yes
LQR	LTV	no	yes	yes
MPC	LTV	yes* polytopes	yes	yes
Nonlinear Trajectory Opt.	Nonlinear	yes* discretized	no	no
Sampling	Nonlinear	yes* discretized	no	no* (faster than nonlinear optimization)

Applications

1. ACC, lane keeping – LQR or Ackermann
2. low speed lane change turns – (RRT or Nonlinear trajectory Opt) + MPC

IV. Frontiers.

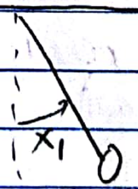
Challenges

speak about these two issues.

1. how do we know the MPC controller will work?
   e.g. we design a path and then have discretization errors or state estimation errors or obstacles, so we use MPC but it only works locally. What is local?
2. how to ensure constraints are truly satisfied?
   e.g. we only check at a finite set of time instances
3. how do we know that QP will operate fast enough?

A. Better understanding the region of attraction of MPC based Controllers.

Ex: Consider a pendulum w/ mass = 1 and gravity = 1, w/ damping



$\dot{x}_2 = x_1$

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ -\sin(x_1) - \gamma x_2 \end{pmatrix}$$

damping $\gamma > 0$

It has equilibria at $x_{2e} = 0$, $x_{1e} = 0, \pm\pi$
What is the region of attraction to the equilibria?

Ex: Lets look at the energy of the system to understand the states which converge to $x_{2e} = 0$, $x_{1e} = 0$.

energy of a pendulum $= V(x)$

$$= mgl(1 - \cos(x_1)) + \tfrac{1}{2} x_2^2$$

$$= (1 - \cos(x_1)) + \tfrac{1}{2} x_2^2 \geq 0 \quad \forall x_1, x_2$$

is equal to zero at $x_{2e} = x_{1e} = 0$.

Lets look at energy evolution:

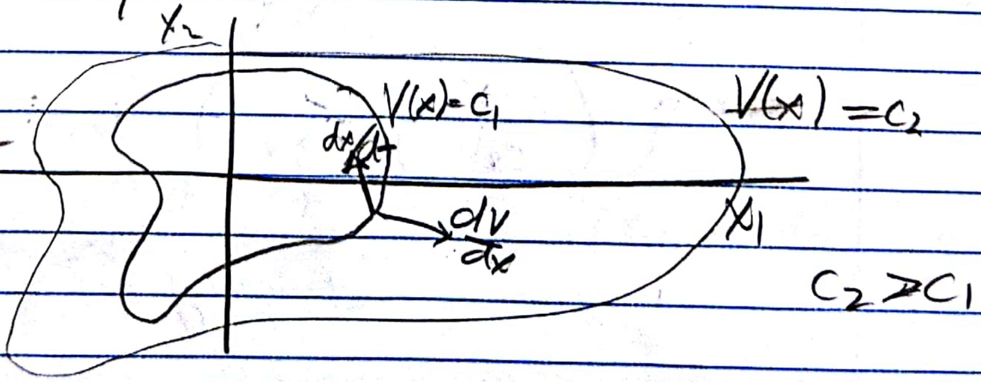
$$\frac{dV}{dt} = \frac{dV}{dx} \cdot \frac{dx}{dt} = \sin(x_1)\dot{x}_1 + x_2 \dot{x}_2$$

$$= \sin(x_1) x_2 + x_2(-\sin(x_1) - \gamma x_2)$$

$$= -\gamma x_2^2 \leq 0 \quad \forall x_1, x_2$$

Note: 1) V non-negative everywhere, zero only at equilibria, dV/dt is non-positive everywhere

Notice dx/dt cannot point out since $\dot{V} \leq 0$.



2) The points where $V \geq 0$ and $\dot{V} \leq 0$ are telling us about the region of convergence to equilibria.

<u>Theorem</u>: Let $B_r := \{x \in \mathbb{R}^n \mid \|x\|_2 \leq r\}$.
Let $V: \mathbb{R}^n \to \mathbb{R}$

a) If $V(x) \geq 0$ and $\dot{V}(x) \leq 0$ for all $x \in B_r$, then all solutions that start in B_r stay in B_r.

b) If $V(x) \geq 0$ and $\dot{V}(x) < 0$ for all $x \in B_r$ except at $x = 0$ which is an equilibria, where $V(0) = 0$ and $\dot{V}(0) = 0$. Then $x = 0$ is locally asymptotically stable w/ nbhd of convergence B_r.

<u>Note</u>: 1) Such a V is called a <u>Lyapunov Function</u>.

<u>Ex</u>: Consider the system

$$\frac{dx}{dt} = \frac{2}{1+x} - x$$

System has an equilibria at $x = 1$, What is the region of convergence?
Shift dynamics to put $x-1$ at origin, by definining $z = x - 1$:

$$\frac{dz}{dt} = \frac{dx}{dt} = \frac{2}{1+x} - x = \frac{2}{2+z} - z - 1$$

↓

Consider the Lyapunov Function:

$V(z) = \frac{1}{2} z^2 \geq 0$ and is zero at $z = 0$

$$\frac{dV}{dt} = \frac{dV}{dz} \cdot \frac{dz}{dt} = z \cdot \dot{z} = \frac{2z}{2+z} - z^2 - z$$

Lets look at $B_2 = \{ z \mid |z| \leq 2 \}$ so

$-2 \leq z \leq 2$ so $2 + z \geq 0$

$$\frac{dV}{dt}(2+z) = 2z - (z^2 + z)(2+z)$$

$$= -z^3 - 3z^2 = -z^2(z+3)$$

So on B_2 $\frac{dV}{dt}(2+z) < 0$ if $z \neq 0$

Therefore on B_2, $z = 0$ is locally asymptotically stable.

Note: 1) how to generate V?
2) how to generate B_r?

B. Generating V

How to ensure positivity of a function?
↳ sampling alone is insufficient
↳ how about only optimizing over functions which are known to be positive

Idea: Focus on sums of square functions

For this discussion functions are polynomials

Theorem: A polynomial $p(x_1, \ldots, x_n)$ of degree $2d$ is non-negative on $\mathbb{R}^n$ if it can be written as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1^2 \\ \vdots \\ \prod x_i^d \end{bmatrix} Q \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1^2 \\ \vdots \\ \prod x_i^d \end{bmatrix}^T$$

where Q is positive semidefinite

Note: 1. The space of psd matrices is convex!
2. Can apply semidefinite programming which is a convex program.
3. How to ensure positivity on a set?

Ex: Suppose we want V to be positive only on a set $A = \{x \in \mathbb{R}^n \mid g(x) \geq 0\}$ where g is a polynomial.

$$\left.\begin{aligned} V(x) - s(x)\,g(x) &\geq 0 \\ s(x) &\geq 0 \end{aligned}\right\} \text{both psd constraints}$$

Why does this work?

Case 1:

x s.t. $g(x) \geq 0$, then $V(x) \geq s(x)g(x) \geq 0$

so $V(x) \geq 0$

Case 2:

x s.t. $g(x) \leq 0$ then $V(x) > s(x)g(x) \geq ??$

so $V(x) \geq ??$

Scanned by CamScanner