LaVallee Part II: Planning under Differential Constraints

4/22/18

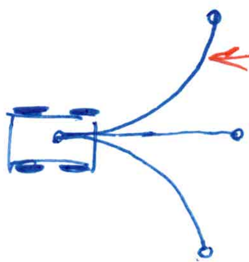
(1)

Ch 5

Contrast to Part II – mainly Sampling-Based Motion Planning

- Constraints were "global" – they applied to C-space.
- Assumed robot is easy to control to follow any ~~straight~~ smooth ~~line~~ path in C-space. (usually we chose a line)

Problem with that, e.g. Dubins Car.



curved path in {W} is not straight line segment in C-space (C-space = SE(2)) because of steering

- Think of differential constraints as "local" – applying to velocities, not configs.
- When practical, plan with a graph representing physially reachable states – in state space, not C-space.

Ch 13:

- To emphasize importance of differential constraints, Ch 13, discribe a number of different types. (Mostly the examples are of wheeled systems.)
- Generally they take the form

$$g(q, \dot{q}) \geq 0$$

which can be as complicated as Cobstacles.

- One special form discussed at length is $\dot{q} = f(q, u)$, where u is the input.

For manipulation, we have

$$M\dot{\nu} = \sum \text{wrenches}$$

$$\dot{q} = V\nu$$

and $0 \leq \lambda_n \perp \psi_n \geq 0$

friction complementarity

13.4.3.1 comes closest to this model, but does not has stick/slip friction

- In some cases, the differential constraints can be integrated, i.e. all $\dot{q}$ variable can be eliminated. In these cases, we can use methods for C-space, i.e. the differential constraints are not fundamental, or inherent.

4/22/18

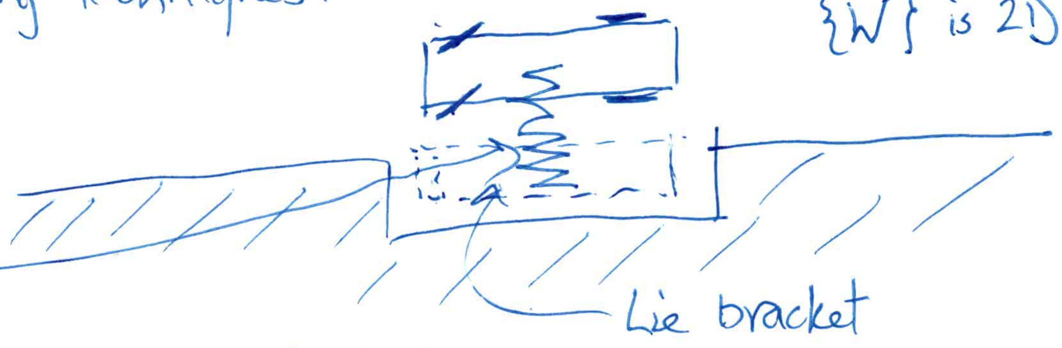
3

Problem class – nonholonomic –

differential variable cannot be integrated out!

Ex. Car can achieve any $q \in C_{free}$ by parallel parking techniques.

{W} is 2D



In W, car w/o wheel constraints has 3 DOF

With wheels, it is "under-actuated" since only 2 dofs can be (locally) controlled.

Officially non-holonomic constrs do NOT affect C_{free}, but they ~~aff~~ constrain trajectories in C_{free}.

Small-time local controllability – ability to follow a path in C_{free} arbitrarily closely even though the number of actuator is less than # DOF.

Lie Brackets provide the controls for motions not in the actuated directions.

Chapter 14: Sampling-Based Planning under Differential Constraints

4/22/18

Assume state transition function

$$\dot{x} = f(x, u)$$

where x is state & may include velocities & accelerations.

Def: Two-Point Boundary Value Problem

action space

Given x_I and x_G, determine $\tilde{u} : [0, \infty) \rightarrow U$ such that x_I is initial state and x_G is final state.

Here $\tilde{u} : [0, \infty) \rightarrow U$ is an action trajectory

$u(t)$ is the action at time t.

Note that $u(t)$ could be state dependent

$$u(t) \in U(x(t)) \subseteq U$$

U is typically actuator forces & torques, but could be changes in position, or new position.

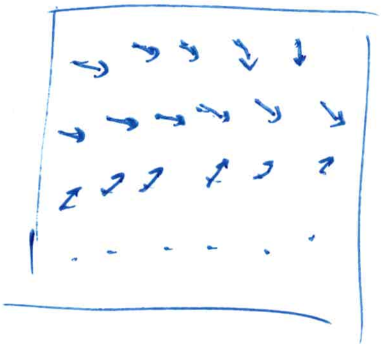
$$x(t) = x(0) + \int_0^t f(x(t'), u(t'))\, dt'$$

4/22/18

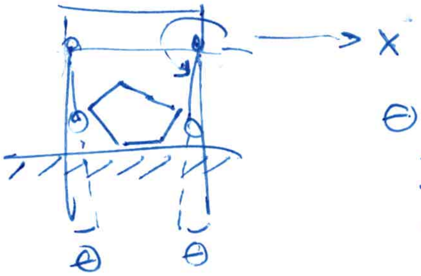
(5)

Note if u is fixed,

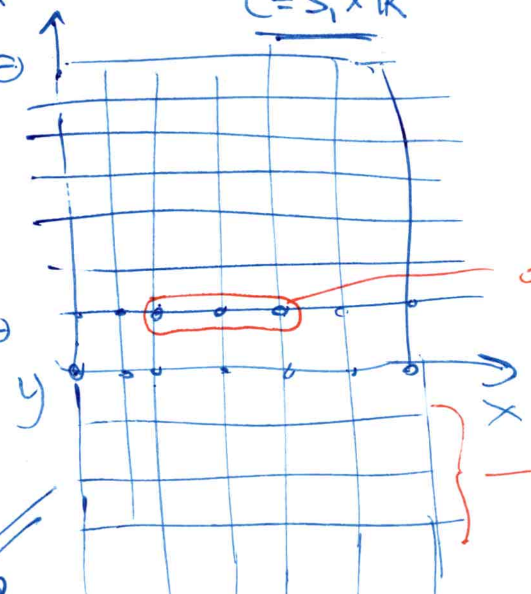
$\dot{x} = f(x,u)$ generates a vector field



For grasping (under actuated)



$C = S_1 \times \mathbb{R}^2$

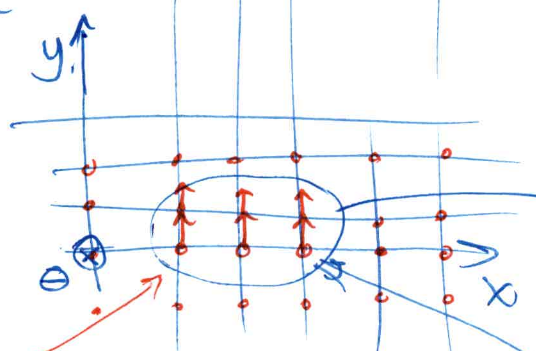


Let $u \Rightarrow \theta = \theta + \Delta\theta$

of interest in grasp acquisition

can't close enough to grab object

different view of C



object lifts as fingers squeeze.

This region defines an intermediate goal during grasp planning.

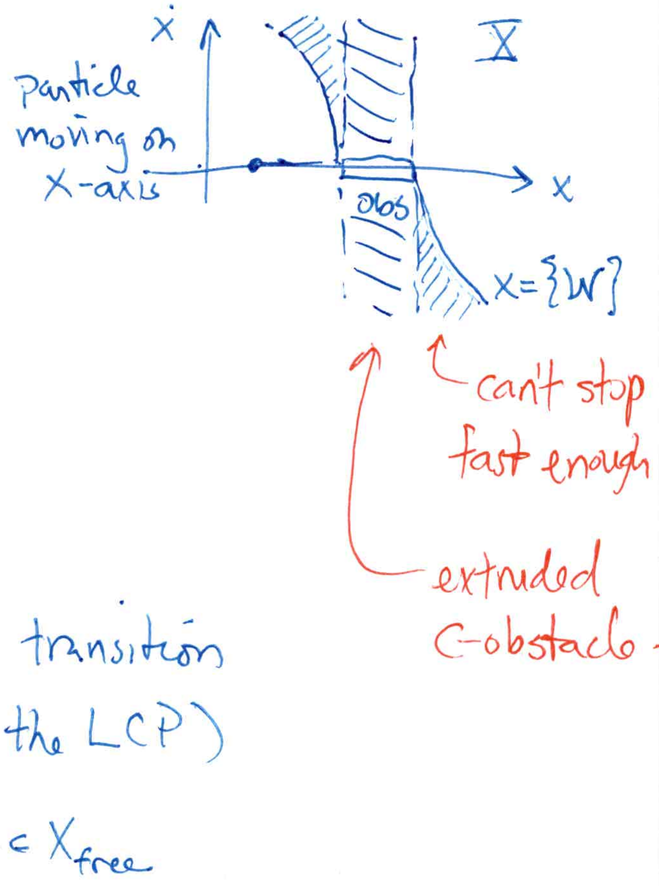
Solve the LCP to find the vector field.

Formulation 14.1 (M.P. under Diff. Constr.)

1. World and Obstacle region defined as in Chapter 4.
2. Unbounded time interval $T = [0, \infty)$
3. Plan on smooth manifold called state space, X.
4. X_{obs} analogous to C_{obs}, but includes states that cannot avoid collision
5. Each $x \in X$ has an associated bounded action space $U(x) \subseteq \mathbb{R}^m \cup \{u_T\}$ where

   m is # of actuated dofs &

   u_T is a terminal action.



6. System is specified by a state transition equation $\dot{x} = f(x, u)$ (and the LCP)
7. x_I is given initial state. $x_I \in X_{free}$
8. $X_G \subset X_{free}$ is goal region
9. A complete alg must compute an action trajectory $\tilde{u} : T \rightarrow U$ for which the resulting state traj. $\tilde{x}$ satisfies: $x(0) = x_I$, $x(T) \in X_G$.

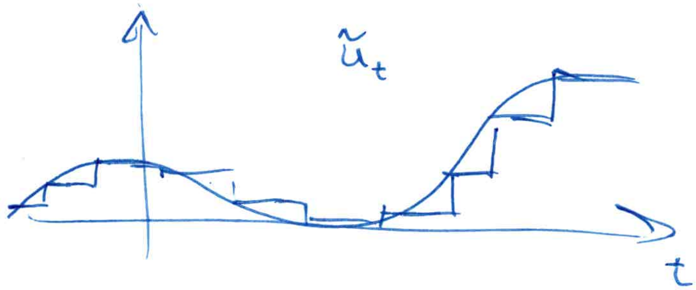
Reachability & Completeness

$\mathcal{R}(x_0, \mathcal{U}) \subseteq \underline{X}$ is reachable from x_0

$$\mathcal{R}(x_0, \mathcal{U}) = \{ x_1 \in \underline{X} \mid \exists\ \tilde{u} \in \mathcal{U} \text{ and } \exists\ t \in [0, \infty) \text{ such that } x(t) = x_1 \}$$

Hard to search continuous space. One approach is to discretize time & $\tilde{u}$

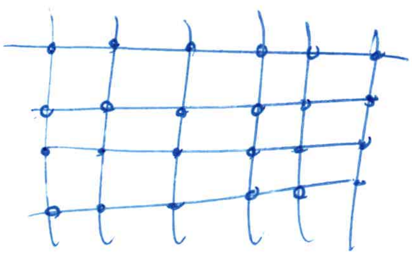
This leads to a reachability graph that is revealed incrementally during search.



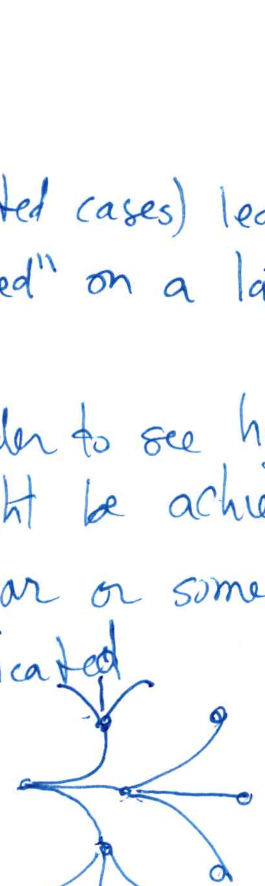
Careful discretization could (in limited cases) lead to reachable states being "trapped" on a lattice

e.g. Use the following:

$$U_d = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\} \Delta t$$



Harder to see how this might be achieved with a car or something more complicated



etc.

4/22/18

8

Resolution Completeness

Sample so that lattice points in reachable graph are dense on the reachable set.

U_d ~~must be~~ ? if a dense sampling of U, then as dispersion of U_d on $U \to 0$ and $\Delta t \to 0$, the graph becomes the reachable set.

General framework

1. Init $G(V,E)$ with $V = x_I$ and $E = \emptyset$.

2. Swath Selection Method (SSM) – choose $x \in S(G)$ for expansion instead of (vertex selection method)

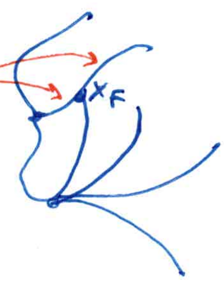
3. Local planning method (LPM) – generate $\tilde{u} : [0, t_F] \to \bar{X}_{free}$

   i.e., choose $\tilde{u}$ for a short time interval & integrate system motion to find x_F.

   Check to be sure $x_F \notin X_{obs}$, if not, go to step 2.

   ~~Then insert edge into G.~~

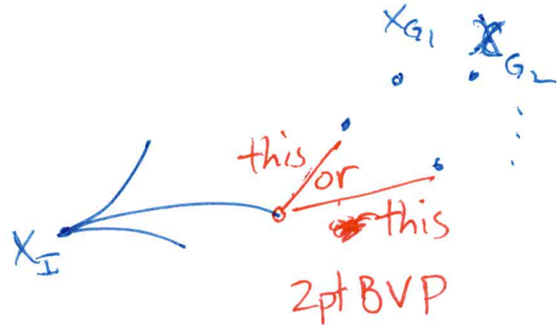
4. Insert edge into G. If x_F on swath, split edge
5. Check for solution
6. Return to step 2 or terminate.



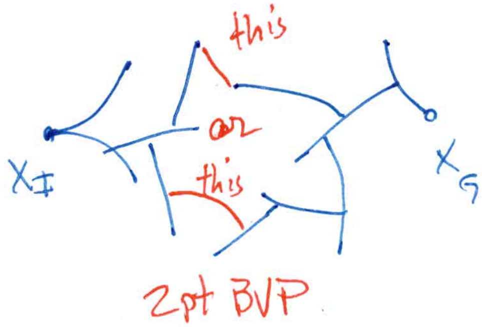
4/22/18

9

At some point, some methods must solve 2pt BVP's



- If X_G = isolated points
- If multi-tree search needs to connect trees.



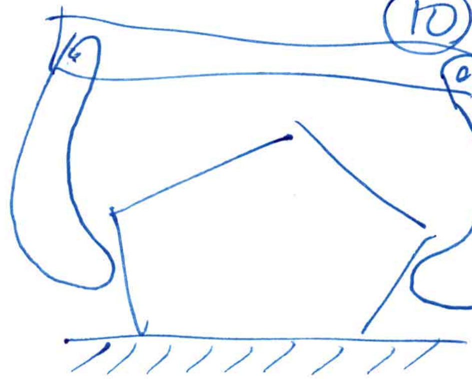
Note if X_G has same dimension as X, then, if large enough, 2pt BVP not needed to be solved.

14.5 Feedback planning.

4/22/18

How to set $\overline{X}_G$ for grasping.

Example,
form closure

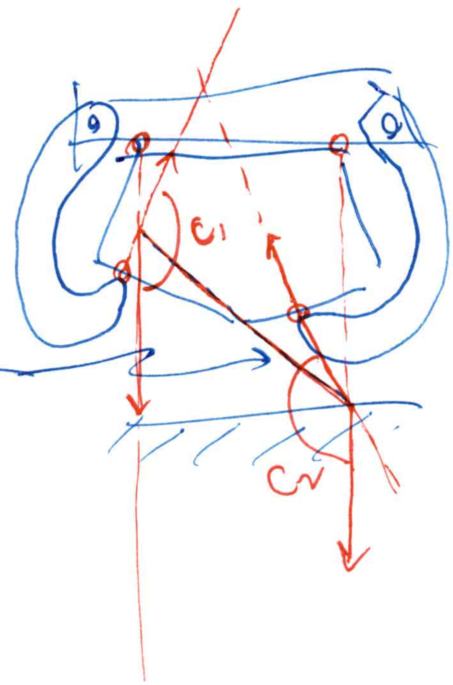


Perhaps you can't set it explicitly!

e.g.

- require any 2 vertices of object on palm and no contacts on ground

cones see each other



- require at least 4 contacts such that the form cl. conditions are satisfied.

Perhap start from form closure grasp and work backward in time to create a <u>reverse</u> reachability graph.