Chapter 10
Motion Planning

Part 1

10.1 Introduction

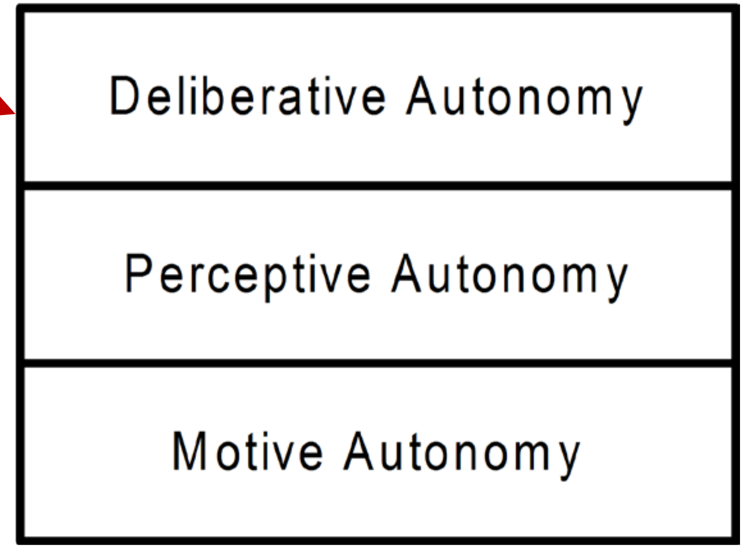
Carnegie Mellon
THE ROBOTICS INSTITUTE

Outline

- 10.1 Introduction
  - 10.1.1 Introducing Motion Planning
  - 10.1.2 Formulation of Motion Planning
  - 10.1.3 Obstacle Free Motion Planning
  - Summary

Hierarchy

- We are here now …
- Responsible for predicting consequences of actions.
- Requires some prior or learned representation of the state of the environment (e.g. a map).
- Usually needs absolute position estimates.
  - i.e referred to a single origin in the entire map.



10.1 Introduction

- Deliberation is related to Optimization
  - Deliberate: Consider many future actions.
  - Predict: Predict the outcomes for each.
  - Optimize: Pick best one.

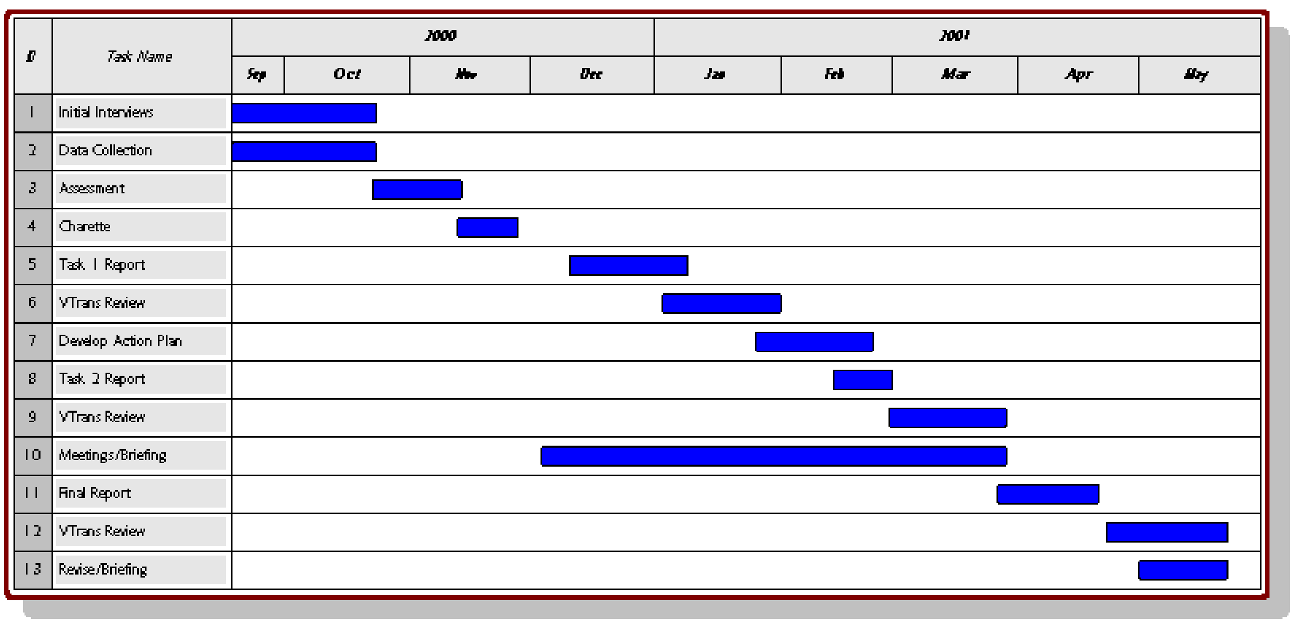
- An accompanying executive element executes plans.
  - c.f. Composer (planner) and orchestra (executive)
  - c.f. Playwright (planner) and players (executive)

Planning vs Reaction

- Reaction is the opposite of planning,
  - Controllers decide what to do now based on locally sensed information.
  - Planners decide what to do later based on predicted information.
- Planning requires predictive models
  - Its impossible to sense the future.
- You sometimes do this in addition to path planning in order to be more responsive or perceptive. See Obstacle Avoidance.

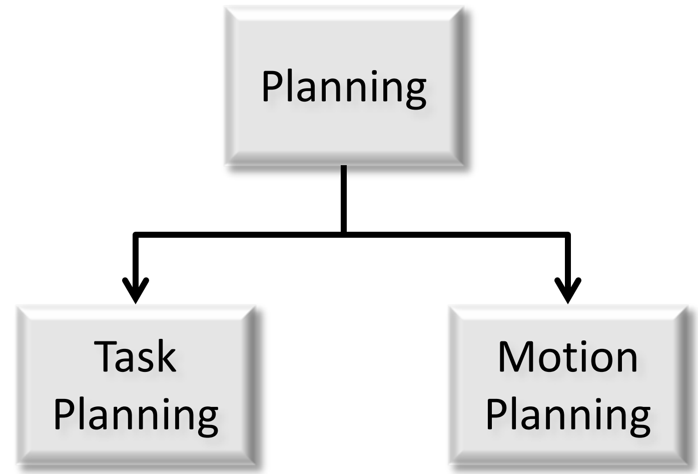
Planning vs Scheduling

- Planning:
  - concerns the sequence of performing actions
  - normally without considerations of time.
- Scheduling: dual of planning
  - concerns sequence and timing
  - normally without explicit considerations of space.



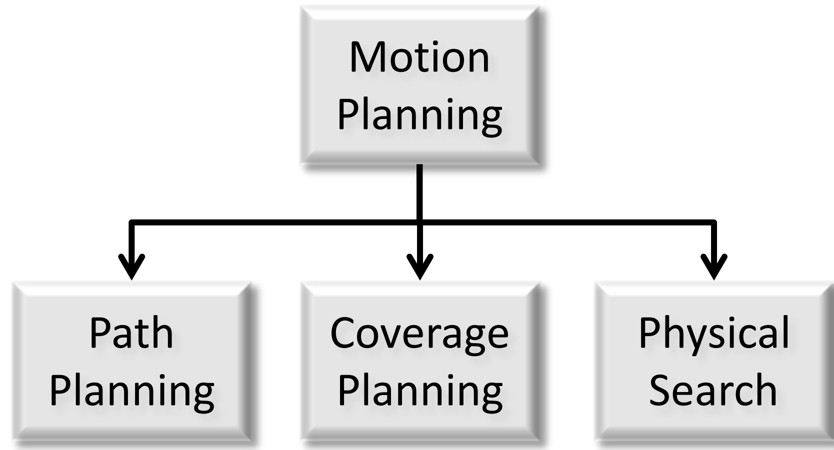
Kinds of Planning

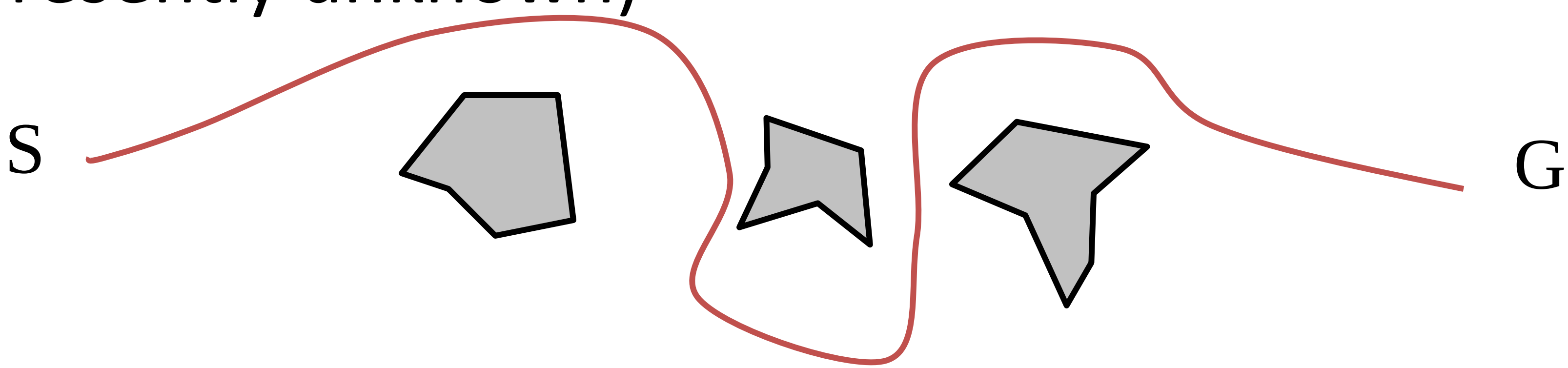
- Motion planning.
  - Finds a path through space.
- We also have task planning
  - e.g. assemble this auto part.



Kinds of Motion Planning

- Path planning (get from A to B)
- Coverage planning (get everywhere)
- Physical search (get somewhere that is presently unknown)





Outline

- 10.1 Introduction
  - 10.1.1 Introducing Motion Planning
  - 10.1.2 Formulation of Motion Planning
  - 10.1.3 Obstacle Free Motion Planning
  - Summary

10.1.1 Introducing Path Planning

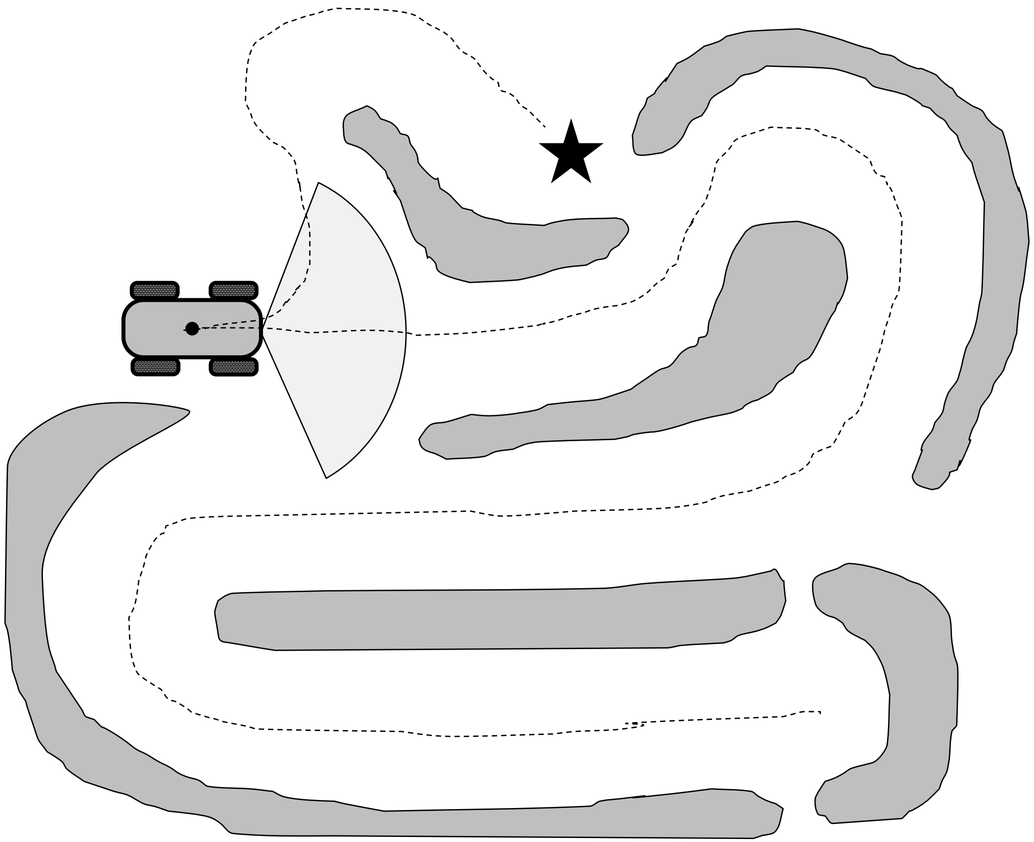
(Global vs Local Planning)

- Two different scales
  - We are considering global path planning.
  - Obstacle avoidance and trajectory generation were local path planning.
- Impact of large scale (well beyond perceptive horizon):
  - Prior models must be used.
  - More abstract models are used.
  - Topologically more complex (more room for obstacles).

10.1.1 Introducing Path Planning

(Tradeoffs Related to Prediction)

- Prediction can help avoid a wrong decision..
- However, diminishing returns:
  - Predicting deeper costs more computation.
  - But … models may be useless as you predict deeper – too much error.
- Response vs Accuracy Tradeoff
  - Accurate models require too much processing. Inaccurate ones give wrong answers.
  - Fast answers or good ones. Pick any one.



10.1.1.2 Predictive Models for Motion Planning

- Entities:
  - Robot (volumetric, massive, kinematic, dynamic, competence)
  - Environment (slope, traversibility, hazard)
  - Objects (occupancy, shape, motion)
- Interactions:
  - Support of weight
  - Traction
  - Collision
  - Traversal cost

Outline

- 10.1 Introduction
  - 10.1.1 Introducing Motion Planning
  - 10.1.2 Formulation of Motion Planning
  - 10.1.3 Obstacle Free Motion Planning
  - Summary

10.1.2 Formulation Path Planning

- Alternatives are paths through space.
- Often the path is constrained in several ways.
  - Start somewhere in particular
  - Avoid obstacles
  - Finish somewhere in particular
- Compute a function or a sequence of actions or states.

$$\underline{x}(t_0) \in S$$

$$\underline{x}(t) \notin O$$

$$\underline{x}(t_f) \in G$$

10.1.2 Formulation Path Planning

(Different Goal Sets)

- Get to one point
  - (“piano movers problem” when obstacles are discrete)
- Get to any of a number of points
  - (e.g. get to high ground)
- Visit all of a set of points in a tour
  - (“travelling salesman problem”)

10.1.2.1 Relationship to Optimal Control

- Path planning searches for an unknown function.
- "Sequence of actions" is just the input u(t)
  - the control sometimes implicit. (i.e. derived from the state sequences)
- Objective function chooses best of many alternatives.
- Obstacles can be in cost function or constraints.
- Terminal state constraint is often active.
  - Conversely, this is often not so for OA.

10.1.2.2 Desiderata for Path Planners

- We like planners to have some or all of these properties:
  - Soundness: Every solution found is a true solution.
    - Feasible: Satisfies motion constraints
    - Admissible: Does not intersect obstacles.
  - Completeness:
    - If any solution exists, it will be found.
    - Otherwise, report failure.
  - Optimality. If more than one solution exists, the best will be generated.
- Achieving any of these goals tends to happen at the expense of efficiency.

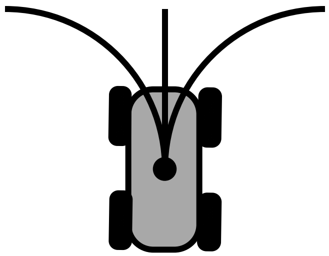
Outline

- 10.1 Introduction
  - 10.1.1 Introducing Motion Planning
  - 10.1.2 Formulation of Motion Planning
  - <u>10.1.3 Obstacle Free Motion Planning</u>
  - Summary

10.1.3 Obstacle Free Motion Planning

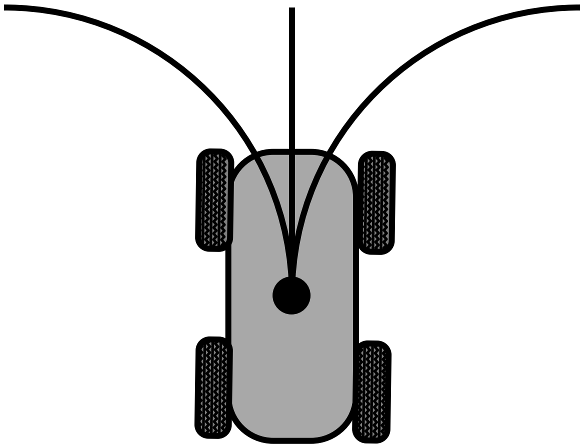
- Easy when there is no vehicle model
  - No “differential constraints”
  - You draw a straight line.
- Surprisingly hard when there is a vehicle model.
  - We’re back to optimal control again.
- Obstacle free solutions are excellent heuristics in motion planning with obstacles.

Example



10.1.3.1 Dubins Car

- Car can turn left or right.
- Must drive forward only.
- Optimal Control Problem.



minimize: $J[\underline{x}, \underline{u}] = \int_0^{s_f} ds = s_f$

where: $\underline{x} = [x \; y \; \theta]^T \quad \underline{u} = [\kappa \; u]^T$

subject to: $\frac{d\underline{x}}{ds} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \kappa \end{bmatrix} u \quad \underline{x}(s_0) = \underline{x}_0 \; ; \; \underline{x}(s_f) = \underline{x}_f$

$u = 1 \qquad |\kappa| \le \kappa_{max}$

- Can reach every point in the plane.
  - But not if there are obstacles (not STLC - small time locally controllable).

STLC Theory

- From http://www.roboticsschool.ethz.ch/sfly-summerschool/programme/4.2-Kumar-Control_Theory_Review.pdf

$$\dot{x} = f(x) + \sum_{j=1}^{m} g_j(x) u_j$$

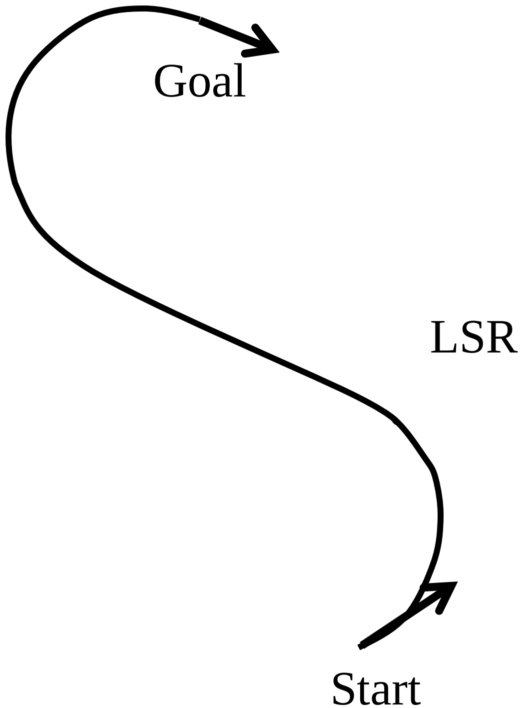
The system is said to be *small time locally controllable* (STLC) at x_0, if given an open subset V in R^n, and x_0, x_1 in V, if for all positive T, there exists an admissible control such that the system can be steered from x_0 to x_1 with $x(t)$ staying inside V for all time.

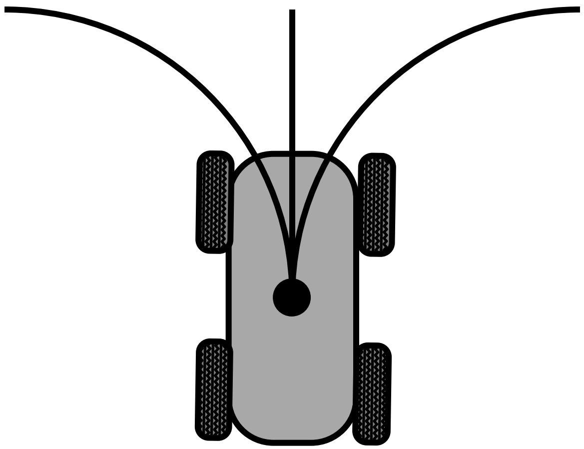
Why is the STLC property important?

- So an STLC system can drive between two points while remaining within an arbitrary region containing both points.
- It implies that the system can:
  - locally maneuver in any direction.
  - approximate any motion of a system with no constraints arbitrarily closely.

10.1.3.1 Dubins Car

- Six 3-letter solutions:
  - LRL, RLR, LSL, LSR, RSL and RSR

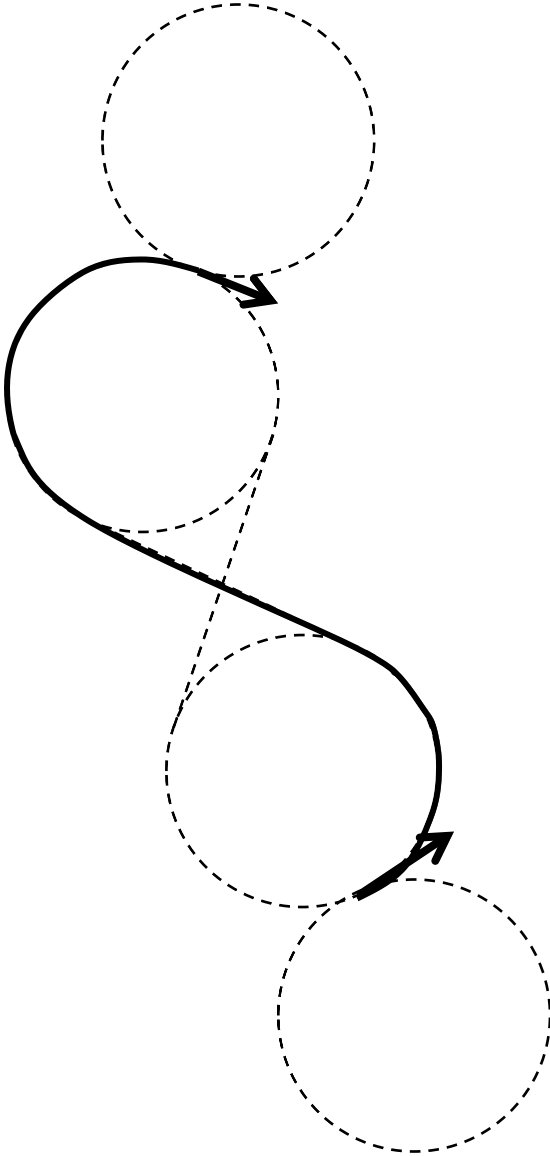




10.1.3.1 Dubins Car

(Extremals)

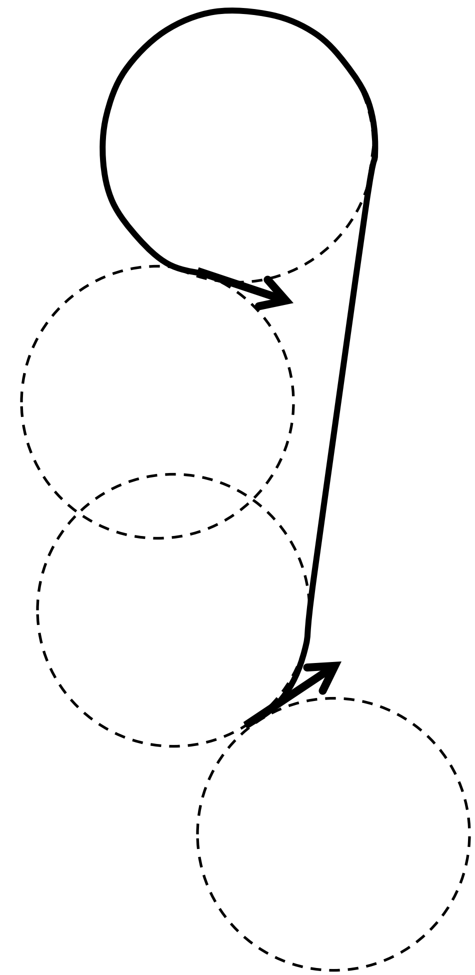
- Intuition (4 of 6 cases).
  - Draw two circles on each side of start and end.
  - Generate all feasible tangent lines from a start circle to an end circle.
  - Pick shortest 3 part path.



10.1.3.1 Dubins Car

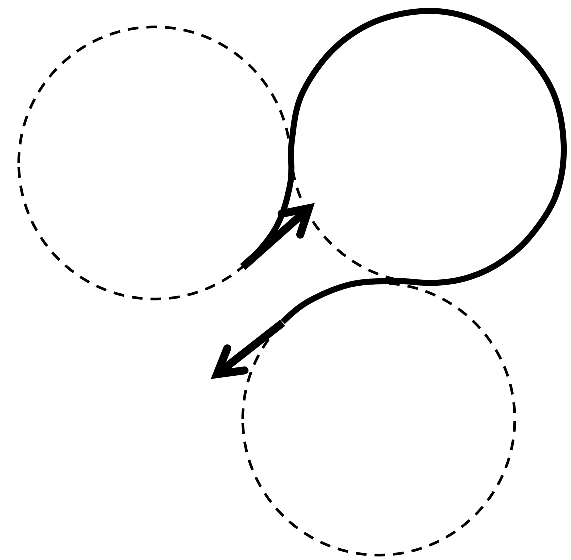
(Extremals)

- Intuition (4 of 6 cases).
  - Draw two circles on each side of start and end.
  - Generate all feasible tangent lines from a start circle to an end circle.
  - Pick shortest 3 part path.



10.1.3.1 Dubins Car
(Extremals)

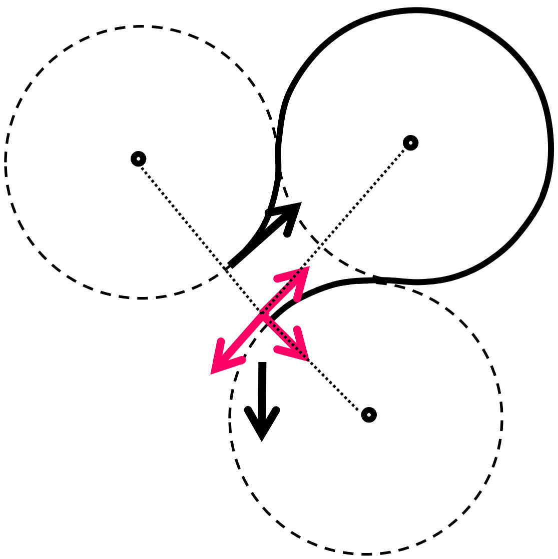
- Intuition (2 of 6 cases).
  - Draw two circles on each side of start and end.
  - Generate all feasible tangent circles from a start circle to an end circle.
  - Pick shortest 3 part path.



Computing Dubins Extremals

- Set up a parameter optimization problem
- E.g. for LSR case:

$$J[\underline{x}, \underline{u}] = \int_{0}^{s_1} ds|_L + \int_{s_1}^{(s_1+s_2)} ds|_S + \int_{(s_1+s2)}^{(s_1+s_2+s_3)} ds|_R = s_f$$



- Three (red) elements of gradient opposite.
- Imagine moving circles around until they fit the terminal pose.

10.1.3.2 Reeds-Shepp Car

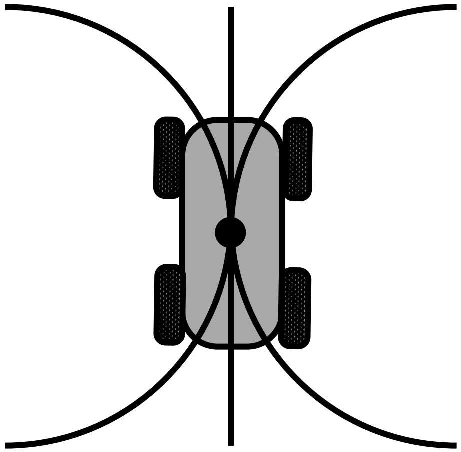
- Car can turn left or right.
- Can go fwd or bkwd.
- Optimal Control Problem.

minimize: $J[\underline{x}, \underline{u}] = \int_0^{s_f} |u| ds = s_f$

where: $\underline{x} = [x \; y \; \theta]^T \quad \underline{u} = [\kappa \; u]^T$

subject to: $\frac{d\underline{x}}{ds} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \kappa \end{bmatrix} u \qquad \underline{x}(s_0) = \underline{x}_0 \; ; \; \underline{x}(s_f) = \underline{x}_f$

$u \in \{1, -1\} \quad |\kappa| \le \kappa_{max}$

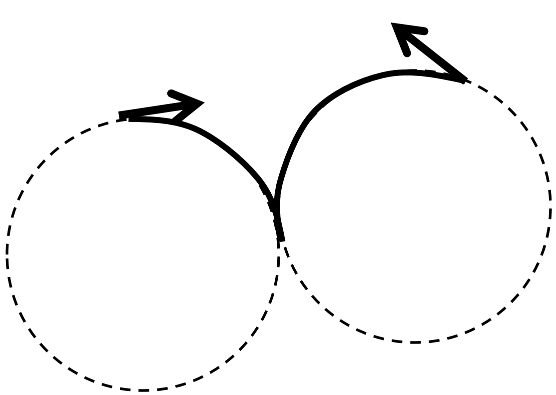


- Can reach every point in the plane.
  - Even if there are obstacles (not STLC - small time locally controllable).

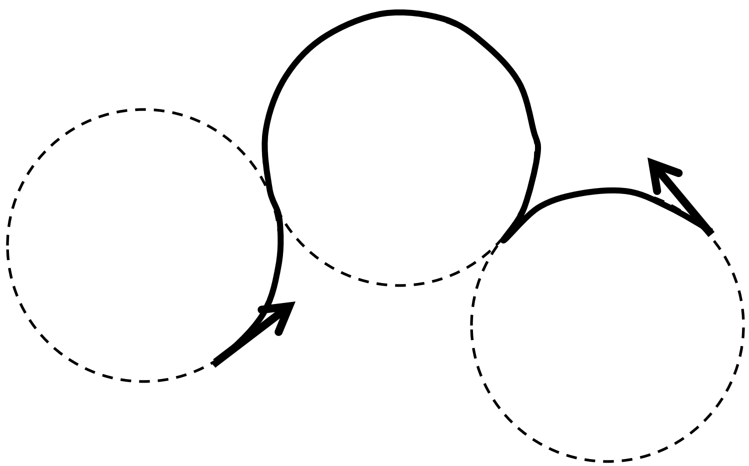
10.1.3.2 Reeds-Shepp Car

- Same Six 3-letter solutions:
  - LRL, RLR, LSL, LSR, RSL and RSR
- Ten 4-letter solutions:
  - LRLR, RLRL, LRSR, RLSL, LRSL, RLSR, LSLR, RSRL, RSLR, and LSRL
- Two 5-letter solutions
  - LRSLR and RLSRL
- 46 in all when you add the signs of velocity.
  - E.g. L+R-L+

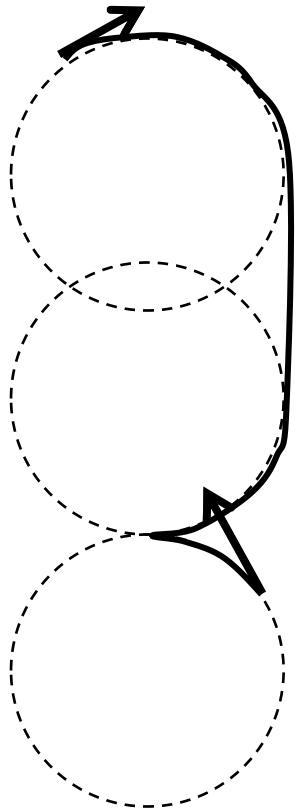
Reeds-Shepp Examples



R^+L^-



$L^+R^+L^-$



$R^+L^-S^-L^-$

+/- Annotation is
For Direction of Motion

Outline

- 10.1 Introduction
  - 10.1.1 Introducing Motion Planning
  - 10.1.2 Formulation of Motion Planning
  - 10.1.3 Obstacle Free Motion Planning
  - Summary

Summary

- Planning is where the most sophisticated deliberative intelligence lies.
- Alternative courses of action are evaluated based on models of environmental interaction.
- An optimal control formulation applies.
  - Dynamics are constraints
  - Obstacles are constraints or costs.
  - Feasible paths are evaluated for utility.
- A large number of design decisions exist, and there are important tradeoffs.
- Even planning with no obstacles is surprisingly difficult.