

Motion Planning with Dynamic Constraints

Dr. Thierry Fraichard

Motion Planning with Dynamic Constraints

Moving obstacles

Dynamic constraints

Moving obstacles and dynamic constraints

Moving Obstacles

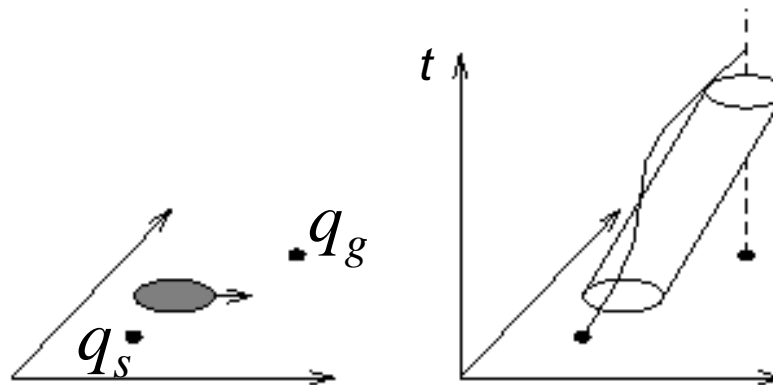
Moving Obstacles

Stationary obstacles: q is free iff $\forall i, A(q) \cap B_i = \emptyset$
 \Rightarrow Time-independent no-collision constraint:

Moving obstacles: B_i becomes time-dependent, $B_i(t)$
 q is free iff $\forall i, A(q) \cap B_i(t) = \emptyset$
 \Rightarrow *Time-dependent* no-collision constraint

Planning a path does not suffice, *time* has to be taken into account.
Motion plan \equiv time-parameterized path

General approach: configuration-time space [Erdmann & Lozano-Perez, 86]



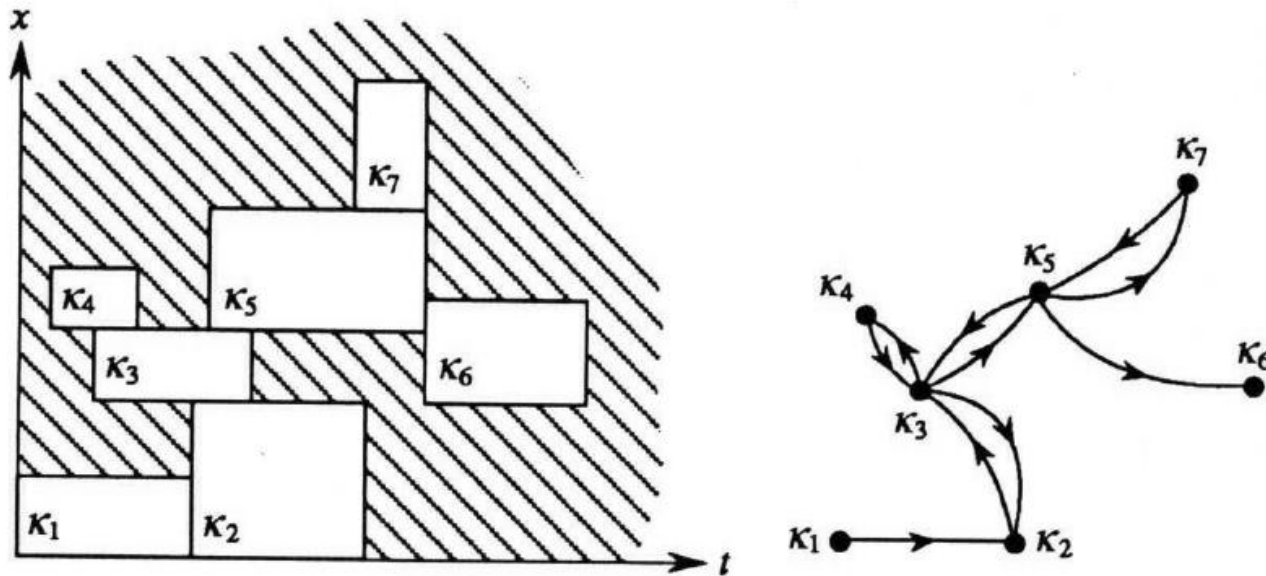
Moving Obstacles (C'ed)

In configuration-time space, the path planning techniques can be used except that...
Time is a particular dimension:

- (a) you cannot travel into the past
- (b) you cannot travel too fast (velocity bound)

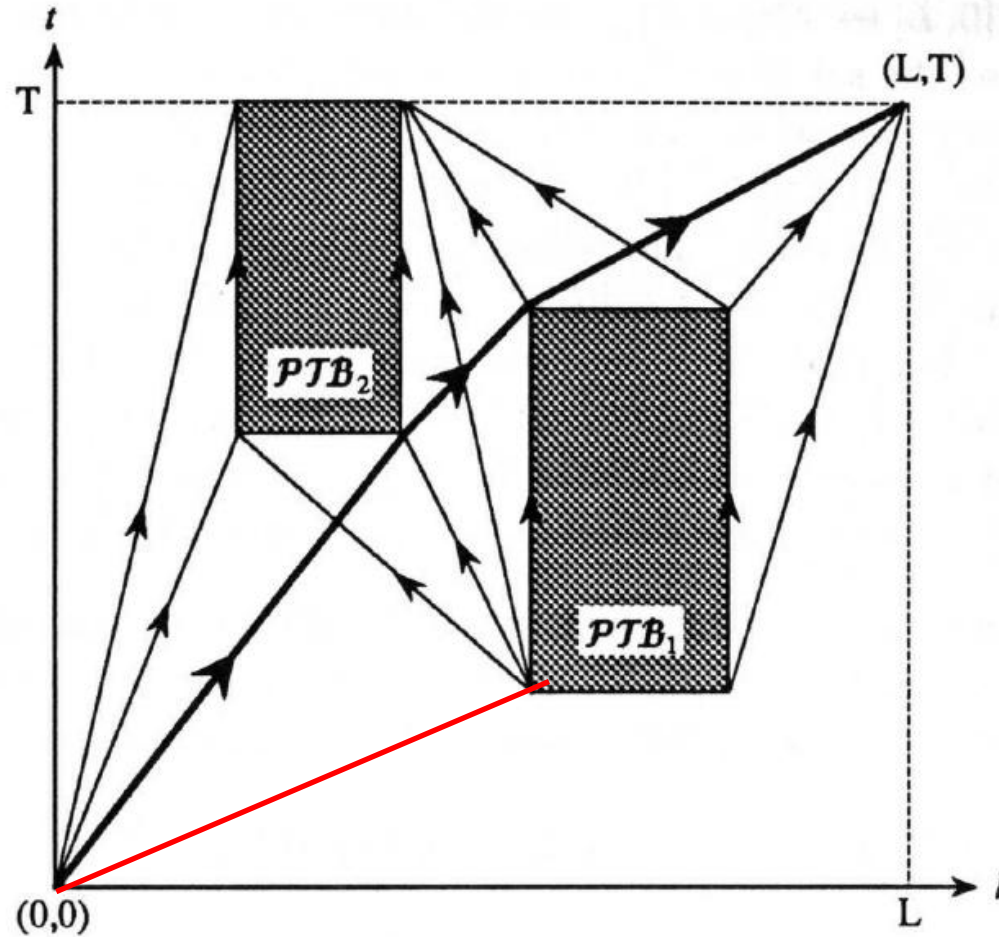
Regular path planning techniques must be adapted

e.g. cellular decomposition approach:



Moving Obstacles (C'ed)

e.g. visibility graph approach:



Moving Obstacles (C'ed)

Path-velocity decomposition [*Kant & Zucker 86*]

- (1) Path planning: stationary obstacles avoidance
- (2) Velocity planning: along the path to avoid the moving obstacles

Step 1: regular path planning problem

Step 2: one dimensional motion planning with moving obstacles
a configuration-time space-based approach can be used

⇒ the initial problem is decomposed into two subproblems of lesser complexity but...
at the expense of completeness

Partial answer to this problem: planning of a set of paths

Dynamic Constraints

Dynamic Constraints

Goal: compute a trajectory, e.g. time-parameterized path, taking into account the robot's dynamics, e.g. force, torque, acceleration, velocity bounds, inertia...

Configuration space & Lagrangian mechanics: $F_i(\ddot{q}, \dot{q}, q) = 0$

State space: $x = (\dot{q}, q)$
 $G(\dot{x}, x) = 0$ implicit representation
 $\dot{x} = f(x, u)$ parametric

Solution trajectory: (a) verifies the constraints,
(b) optimizes a given criterion,
e.g. execution time, fuel consumption...

(It is not always the case, e.g., if a robot with dynamic constraints is surrounded by moving obstacles, the main goal might be to avoid collision)

Planning Approaches

Direct planning:

Grid-based methods: grids defined in

Workspace [e.g. Shiller & Dubowsky 88]

Configuration space [e.g. Sahar & Hollerbach 85]

State space [e.g. Canny et al. 88; Donald & Xavier 90]

Rapidly-exploring random trees in state space [LaValle, 98]

...

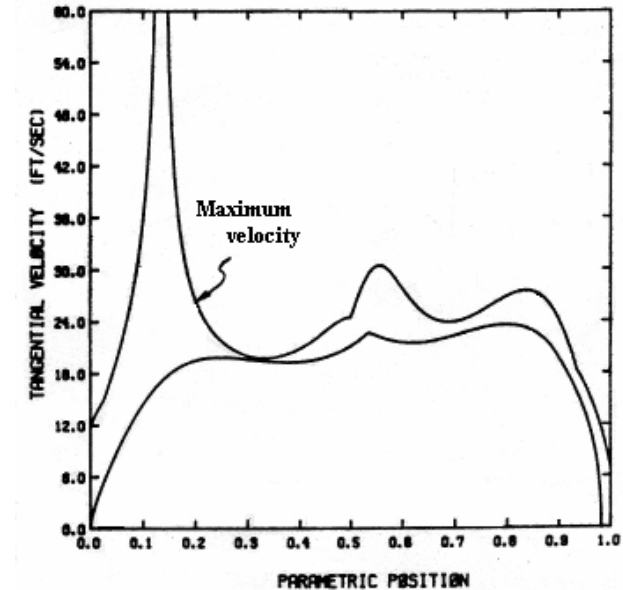
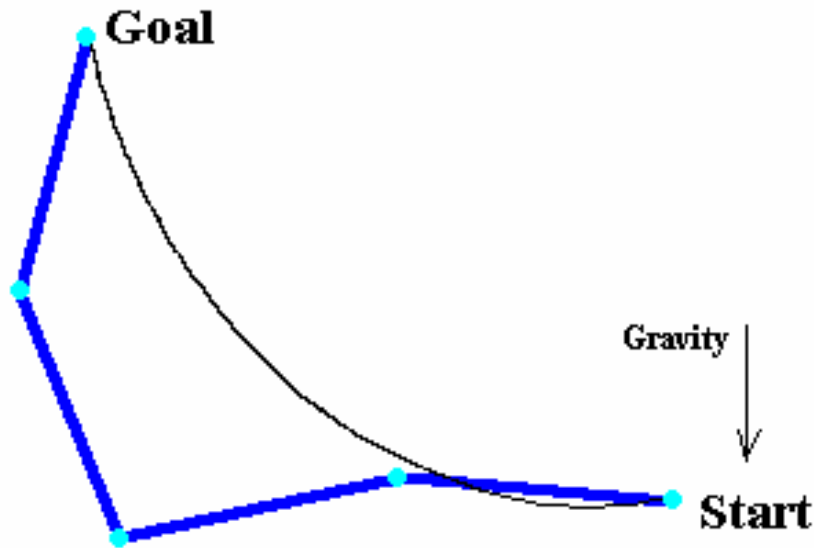
Two-phase planning:

Compute a collision-free path ignoring constraints

Transform this path into a trajectory satisfying the dynamic constraints

Two-Phase Planning

Control theory: results for robots with full dynamics moving along a given path



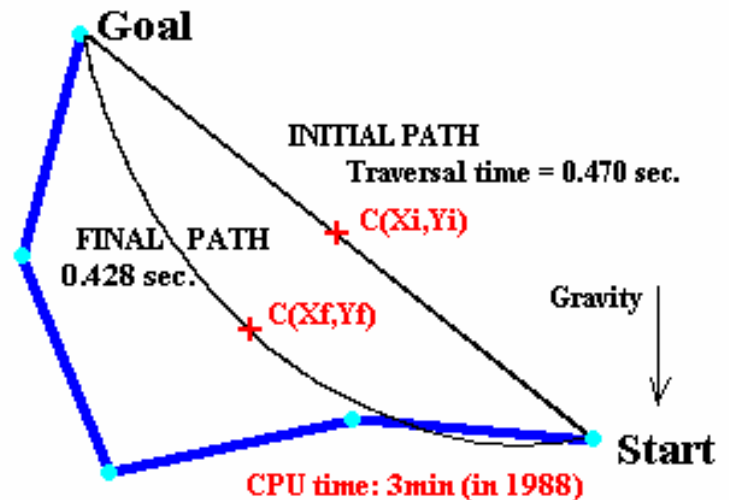
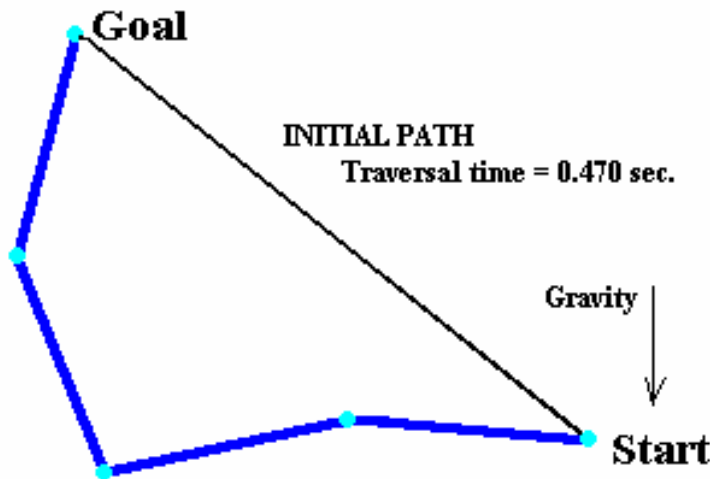
Two-phase optimization techniques: optimal motion time along a given path is used as a cost function for a local path optimization phase (variational technique)

Two-Phase Planning (C'ed)

Geometric path represented as a set of parameterized interpolation functions
e.g. B-splines polynomials [Bobrow, 88]

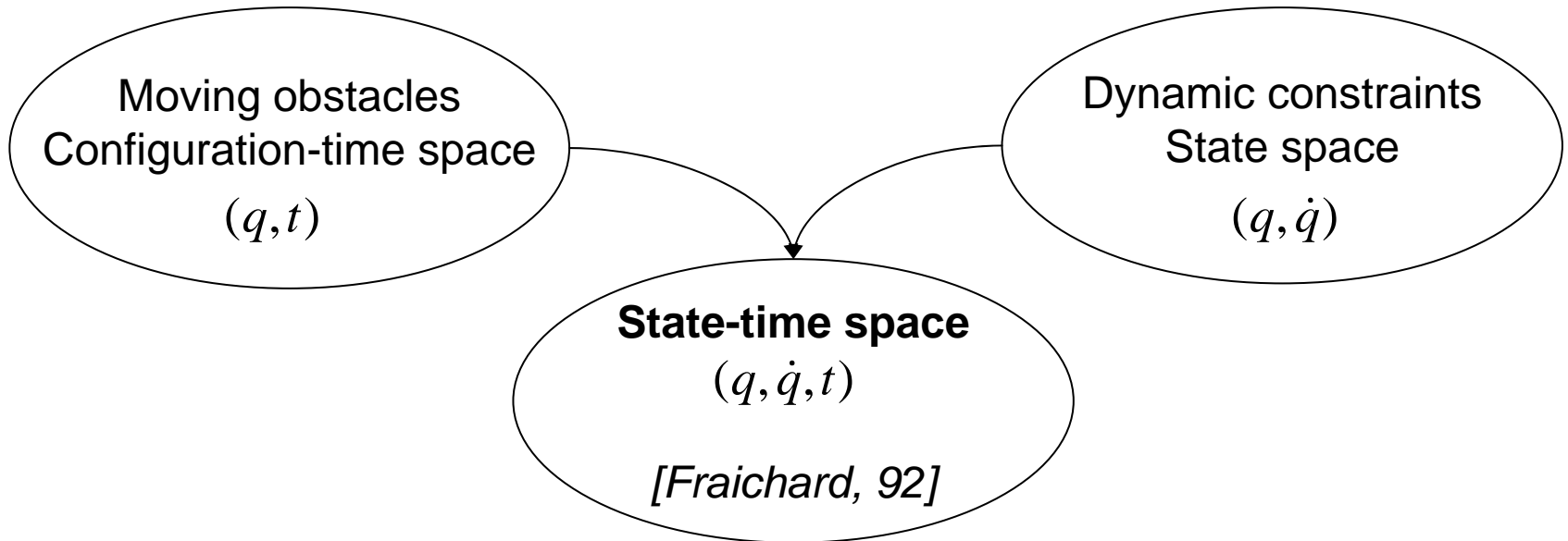
Parameters are varied to minimize the path traversal time
Local control of the path shape

Such techniques computes local time-optimal motions, they are costly though ...



Moving Obstacles and Dynamic Constraints

Principle



Unified modelling of different types of constraints:
State-time space obstacles

Trajectory = curve in State-time space
Avoids State-time space obstacles
Verifies additional shape properties: \ddot{q}, t

Case Study

Car-like robot: kinematic constraints (perfect rolling, limited steering) and dynamic constraints (engine force, no sliding, maximum velocity)

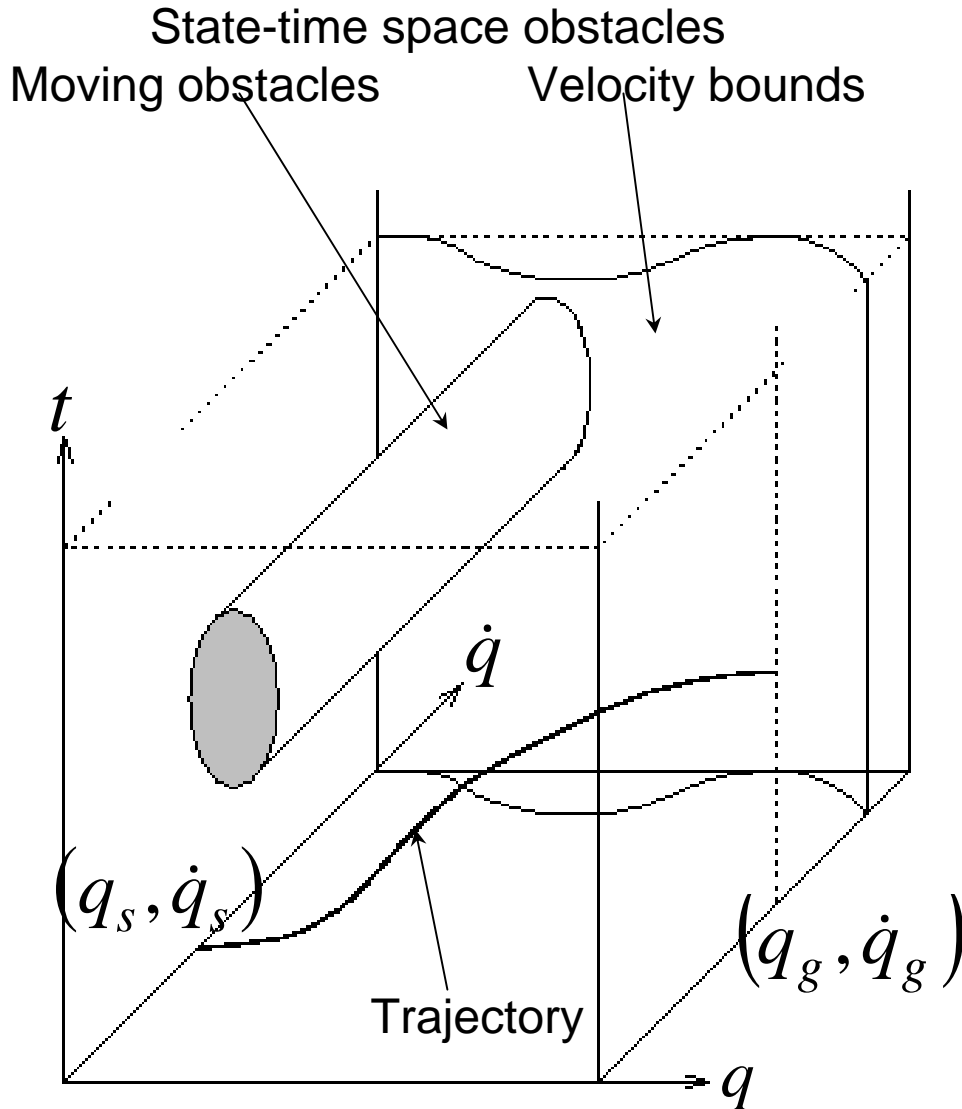
Dynamic constraints \Rightarrow bounds on accelerations + velocities

2D workspace: stationary and moving obstacles

+ **Nominal path:** collision-free (stationary obstacle)
and feasible (kinematic constraints)

\Rightarrow one-dimensional kinodynamic motion planning

State-Time Space

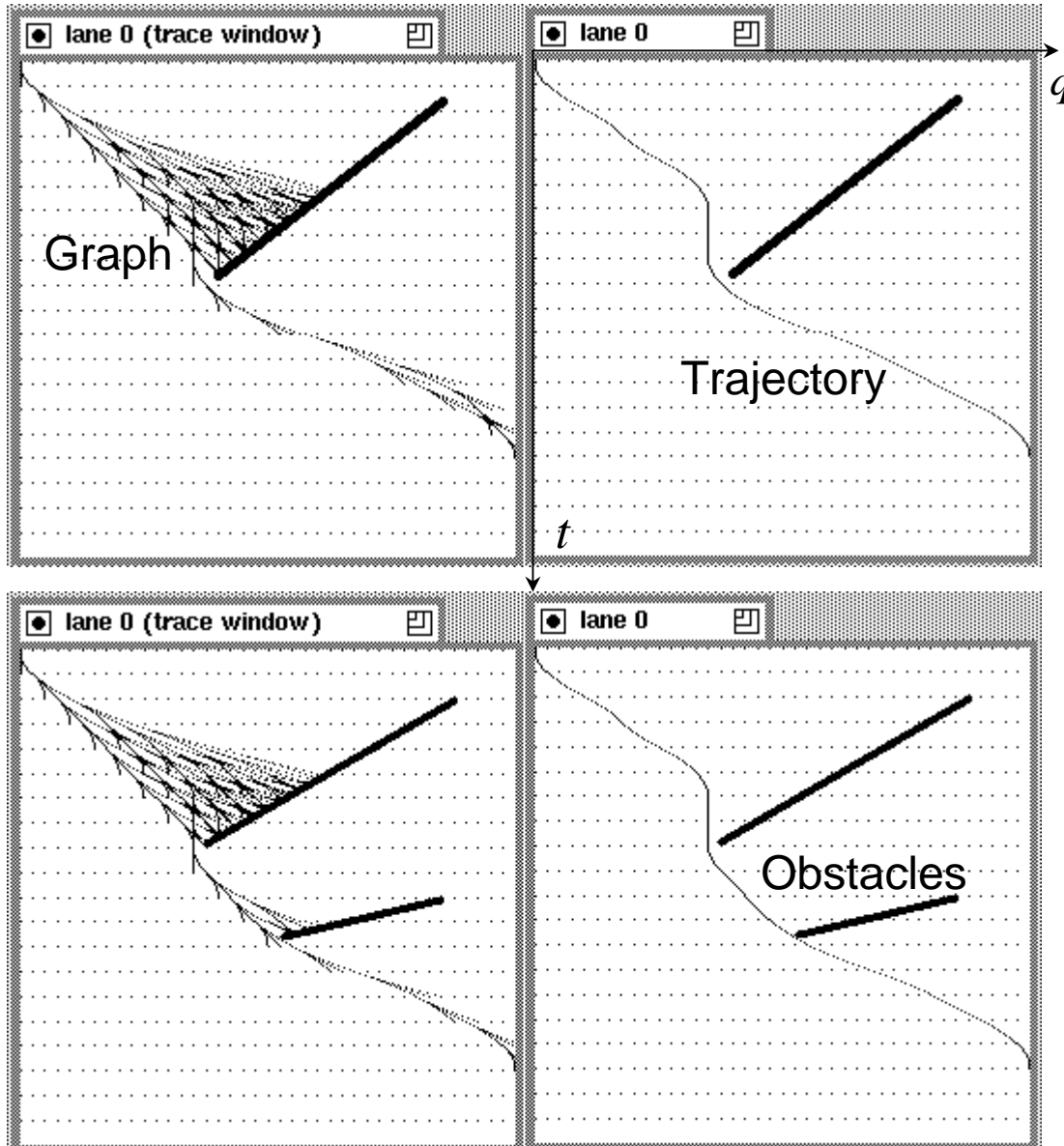


Velocity bounds + moving obstacles
 \Rightarrow State-time space obstacles

Trajectory = curve in State-time space
from $(q_s, \dot{q}_s, 0)$ to (q_g, \dot{q}_g, T)

Acceleration bounds + time constraint
 \Rightarrow curve shape constraints

Results



Moving obstacles:
a priori knowledge / prediction
 \Rightarrow time horizon: T_{\max}

Running time: $P(\tau, \delta, T_{\max})$

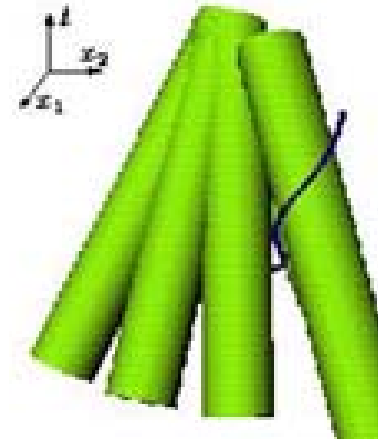
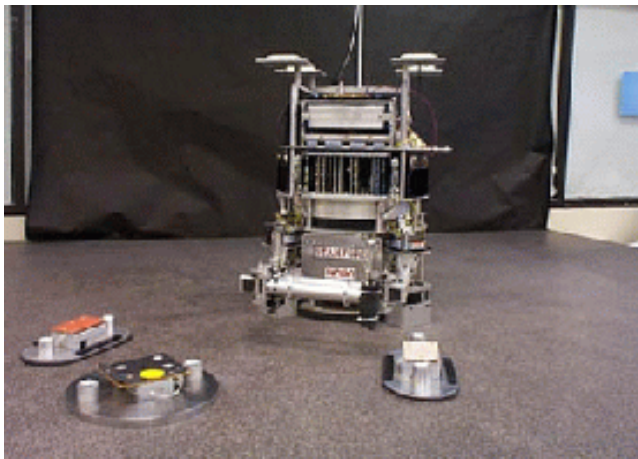
2D workspace:
[Fraichard & Scheuer, 94]

Other Planning Approach

Point mass robot moving on a plane

$s' = G(s, u)$, $s = (q, q')$, $q = (x, y) \rightarrow 5D$ state-time space

Randomized search method [Kindel et al., 00]



T = 0.0 secs

T = 9.0 secs

T = 20.0 secs

T = 30.0 secs

T = 45.5 secs

