MOTION PLANNING

Maria Isabel Ribeiro
Pedro Lima

mir@isr.ist.utl.pt pal@isr.ist.utl.pt
Instituto Superior Técnico (IST)
Instituto de Sistemas e Robótica (ISR)
Av.Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL

April.2002

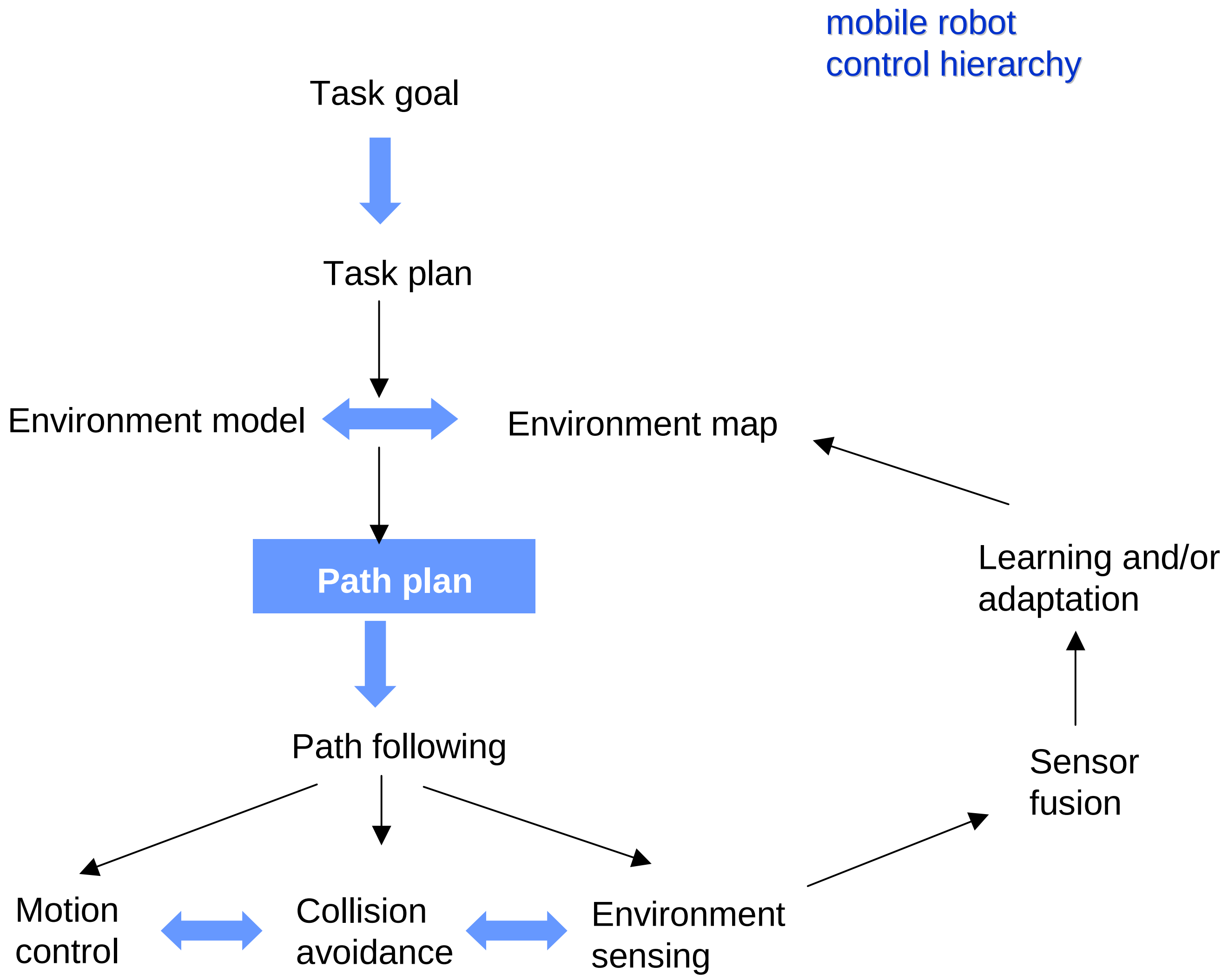
References

- J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991 (Chapter 1, Chapter 7, Chapter 9 –Kinematics Constraints)
- G. Dudek, M. Jenkin, *Computational Principles of Mobile Robots*, MIT Press, 2000 (Chapter 5)
- Additional references
  - Path Planning with A* algorithm
    - S. Kambhampati, L. Davis, “Multiresolution Path Planning for Mobile Robots”, IEEE Journal of Robotrics and Automation, Vol.RA-2, No.3, September 1986, pp.135-145.
  - Potential Field
    - O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, Int. Journal of Robotics Research, 5(1), pp.90-98, 1986.
    - P. Khosla, R. Volpe, “Superquadratic Artificial Potentials for Obstacle Avoidance and Approach” Proc. Of ICRA, 1988, pp.1178-1784.
    - B. Krogh, “A Generalized Potential Field Approach to Obstacle Avoidance Control” SME Paper MS84-484.
  - Smooth Paths
    - Y.Kanayama, B. Hartman, “Smooth Local Path Planning for Autonomous Vehicles”, in *Autonomous Robot Vehicles*, edited by Cox, Wilfong, Springer, 1990, pp.62-67.

Problem Statement

Navigation problem

- Find a path from a start location (S) to a final goal (G) and traverse it withouth collision
- Decomposition in three sub-tasks:
  - Mapping and modelling the environment
  - **Path planning and selection**
  - Path traversal and collision avoidance



Problem Statement

- MOTION PLANNING
  - Evaluate a collision free path from an initial pose and the goal, taking into account the constraints (geometric, physical, temporal)
  - In certain situations motion planning takes into account more advanced aspects as:
    - Robot cooperation
    - Manouvers (push and pull)
    - Uncertainty of available information

- Three different aspects in MOTION PLANNING
  - PATH PLANNING
  - MANOEUVRE PLANNING
  - TRAJECTORY GENERATION

PATH

A PATH is a geometric locus of the points – in a given space – where the robot has to pass

TRAJECTORY

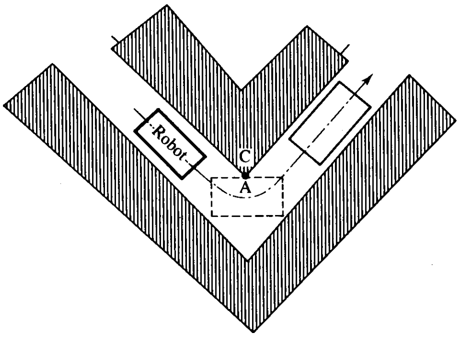
A TRAJECTORY is a path for which a temporal law is specified (e.g., acceleration and velocity in each point)

MANOUVERS

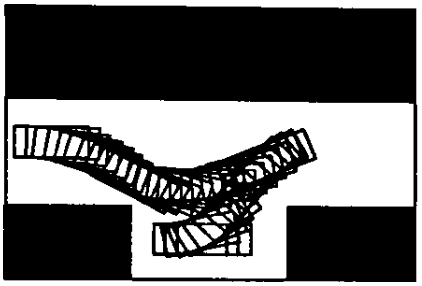
A mobile robot is not a point in the space

Piano-mover's problem

A path that the rectangular robot can negotiate only if it rotates around A as it turns the corner C



Parking a car in a narrow parking lot



Basic Motion Planning Problem

Notation

- **A**: single rigid object – (the robot)
- **W**: Euclidean space where **A** moves

$$W = \Re^2 \quad \text{or} \quad \Re^3$$

- $\mathbf{B_1, B_2, \ldots, B_m}$ fixed rigid objects distributed in **W**. These are the obstacles

Assumptions

- The geometry of **A** and $\mathbf{B_i}$ is known
- The localization of the $\mathbf{B_i}$ in **W** is accurately known
- There are no kinematic constraints in the motion of **A** (**A** is a free-flying object)

Problem

Given an initial pose and a goal pose of **A** in **W**, generate a path τ specifying a continuous sequence of poses of **A** avoiding contact with the $\mathbf{B_i}$, starting at the initial pose and terminating at the goal pose.

Report failure if no such path exists.

pose = position + orientation

Example

Illustration of the motion planning problem

Rectangle shaped robot

L-shaped robot

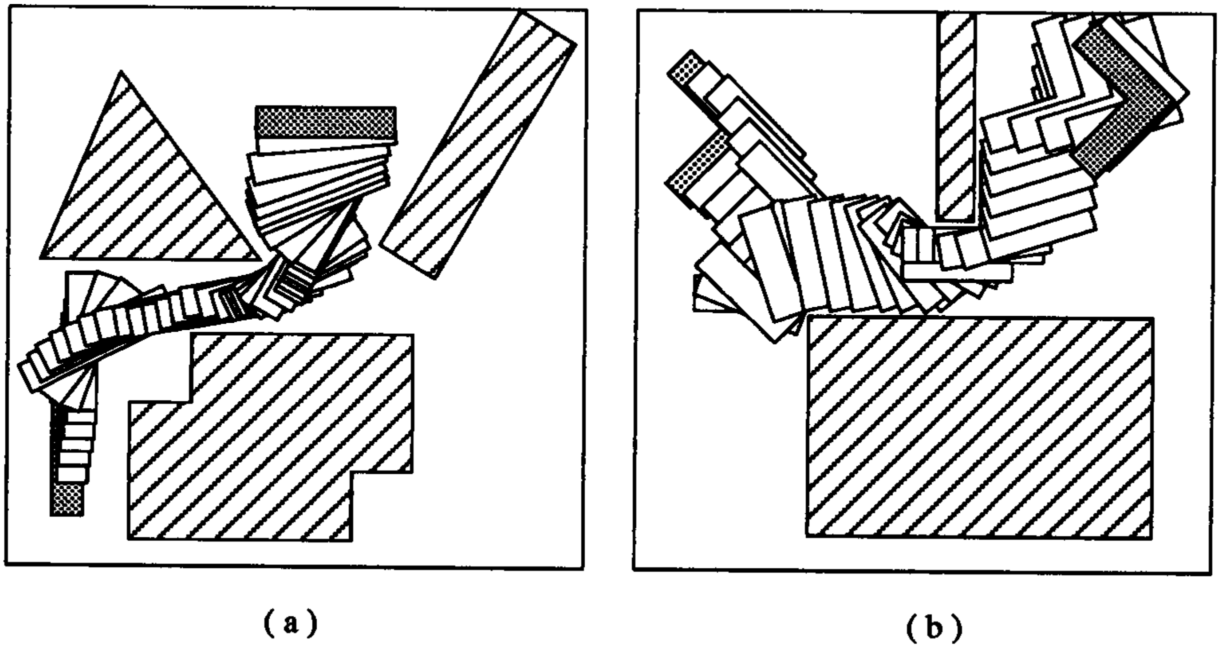
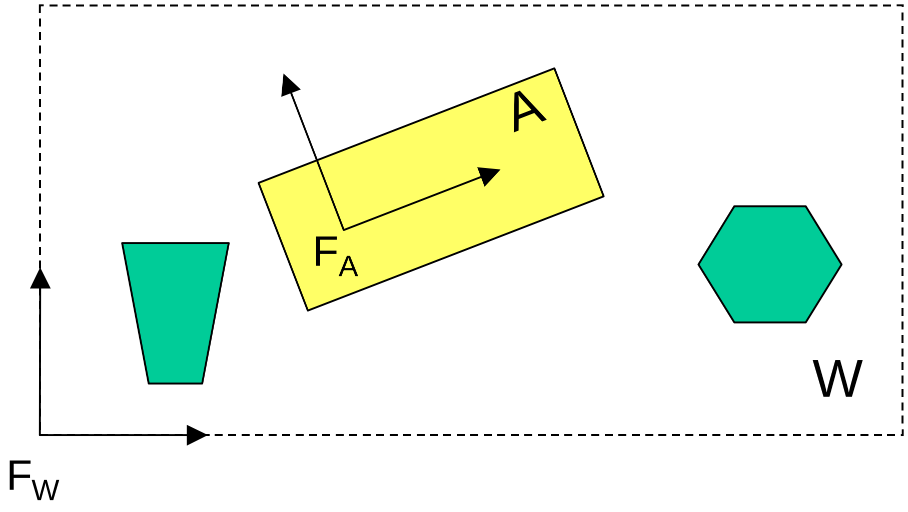


Figure 2. This figure illustrates the basic motion planning problem with two robots — a rectangle in Figure a and an L-shaped polygon in Figure b. The contact-free paths are displayed as discrete sequences of positions and orientations of the robot.

From
Robot Motion Planning
J.C. Latombe

Configuration Space



F_W – world frame
F_A – robot frame

- F_W – world frame – fixed frame
- F_A – robot frame –
  - moving frame (rigidly associated with the robot)
  - rigid frame
  - any point a in A has a fixed position in F_A, but its poisition in F_W depends on the position and orientation of F_A relative to F_W.
- The objects B_i are fixed = any point in B_i has a fixed position with respect to F_W.

A configuration q of A

pose (position and orientation) of F_A with respect to F_W.

Configuration space of A

space **C** of all configurations of **A**

A(**q**) = subspace of **W** occupied by **A** at configuration **q**

Configuration Space

path of A

Set of configurations from $\mathbf{q}_{init}$ to $\mathbf{q}_{goal}$, definied as a continuous map:

$$\tau : [0,1] \rightarrow C$$

$$\tau(0) = q_{init}$$

$$\tau(1) = q_{goal}$$

Continuity is defined using a topology in **C**.
The topology is induced by an Euclidean distance

- This definition of path considers that **A** is a free-flying object for the path to be feasible
- No objects present

the only constraints on its motions are due to obstacles. No kinematics nor dynamics constraints

OBSTACLES IN THE CONFIGURATION SPACE

- No pose of the robot **A** along a **path** can intersect any object (i.e., a path should be collision free)
- The robot has a given shape
  - Along its path the robot with its shape spans a region of **W**
  - This spanned region is a function of the robot shape and consecutive poses along the path
  - No point of this spanned area can intersect any object

$$W \longrightarrow C$$

$$B_i \longrightarrow CB_i = \{q \in C : A(q) \cap B_i \neq \phi\}$$

object $\longrightarrow$ **C-object**

Example

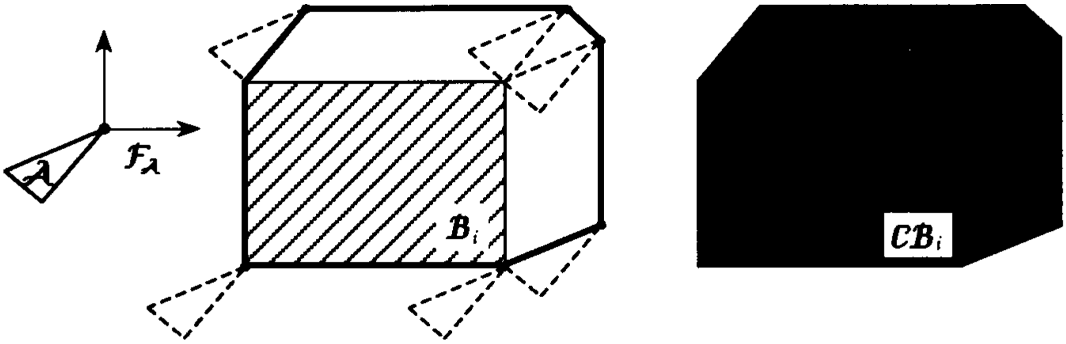


Figure 3. The robot $\mathcal{A}$ (a triangle) can translate freely in the plane at fixed orientation. Its configuration is represented as $\mathbf{q} = (x, y)$, the coordinates in $\mathcal{F}_{\mathcal{W}}$ of the vertex of $\mathcal{A}$ marked as a small circle (the origin of $\mathcal{F}_{\mathcal{A}}$). Hence, $\mathcal{A}$'s configuration space is $\mathcal{C} = \mathbf{R}^2$. The C-obstacle $\mathcal{CB}_i$ (shown dark) is obtained by "growing" the corresponding workspace obstacle $\mathcal{B}_i$ (a rectangle) by the shape of $\mathcal{A}$. Planning a motion of $\mathcal{A}$ relative to $\mathcal{B}_i$ is equivalent to planning a motion of the marked vertex of $\mathcal{A}$ relative to $\mathcal{CB}_i$.

From
Robot Motion Planning
J.C. Latombe

C-OBSTACLE REGION

$B_1, B_2, \ldots, B_m$ —— obstacles

CB_i —— C-obstacle

$\bigcup_{i=1}^{m} CB_i$ —— C-obstacle region

FREE SPACE

$$C_{free} = C \setminus \bigcup_{i=1}^{m} CB_i = \{q \in C : A(q) \cap \bigcup_{i=1}^{m} CB_i = \phi\}$$

Free configuration q iff $q \in C_{free}$

Path Planning in Configuration Space

Planning a path in C for a non-point robot
and in presence of obstacles

Planning a path in C_{free} for a point robot

$$\tau : [0,1] \rightarrow C_{free}$$

$$\tau(0) = q_{init}$$

$$\tau(1) = q_{goal}$$

Assumption: free-flying object assumption

In the presence of obstacles any path defined in such a way is feasible (continuous)

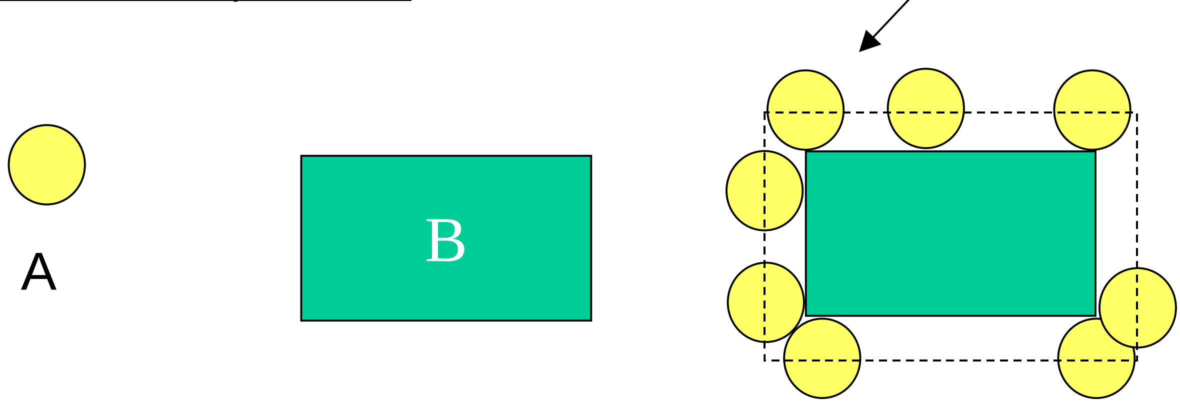
The motion of **A** is not limited by any kinematics or dynamic constraints

PARTICULAR SITUATIONS

A is a point

C-obstacle

A is a disk-shaped robot



A only has translational motion

Planning Approaches

- Methodololgies for Path Planning
  - Roadmap
  - Cell Decomposition
  - Potential Field

Global methods

Local methods

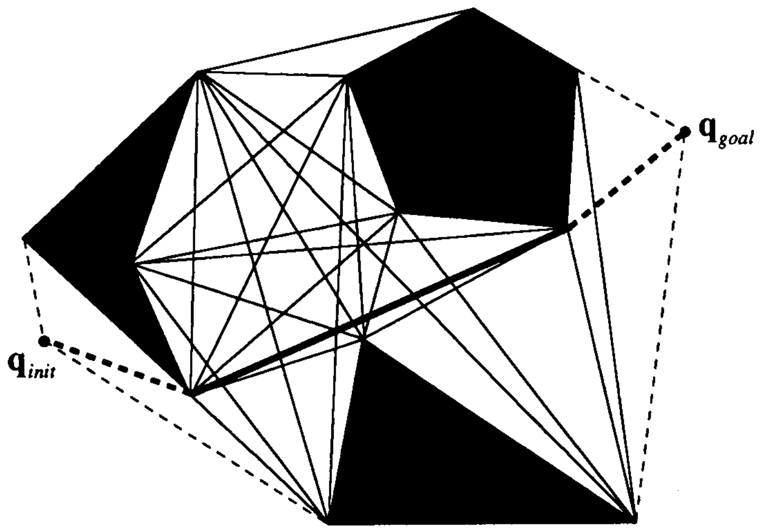
for free-flying robots

- **Roadmap**
  - From C_{free} a graph is defined (Roadmap)
  - Ways to obtain the Roadmap
    - Visibility graph
    - Voronoi diagram
    - Freeway
    - Silhouette
- **Cell Decomposition**
  - The robot free space (C_{free}) is decomposed into simple regions (cells)
  - The path in between two poses of a cell can be easily generated
- **Potential Field**
  - Path planning is based on a powerful analogy:
  - The robot is treated as a particle acting under the influence of a potential field U, where:
    - the attraction to the goal is modeled by an additive field
    - obstacles are avoided by acting with a repulsive force that yields a negative field

Roadmap: Visibility Graph

- **Hypothesis**: Polygonal C-obstacles
- **VISIBILITY GRAPH** = non directed graph whose nodes are:
  - The initial pose (q_{init})
  - The goal pose (q_{goal})
  - The vertices of the obstacles (in the C-obstacle space)

Two nodes are linked by a line segment if the straight line segment joining them does not intersect any obstacle



From
Robot Motion Planning
J.C. Latombe

The plain lines connect the obstacle vertices → Roadmap

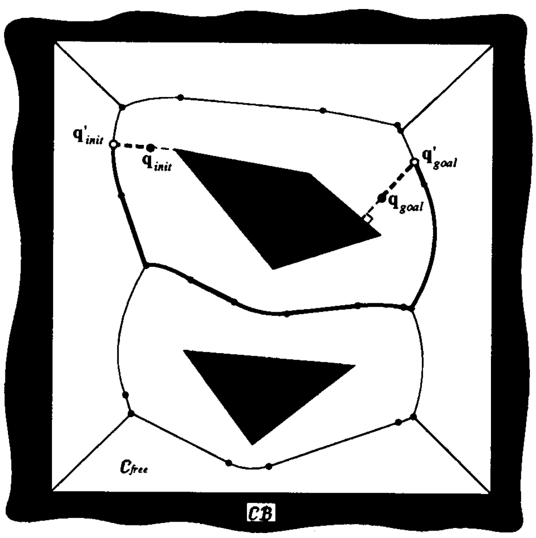
Independent of the **initial** and **goal** poses

The dashed lines connect q_{init} and q_{goalt} to the roadmap

- **HOW TO FIND THE SHORTEST PATH ?**
  - **Search the roadmap**
- **IS THIS PATH SMOOTH ?**
- **IS THIS PATH COLLISION FREE?**

Roadmap: Voronoi Diagram

- **Given:**
  - The Generalized Voronoi Diagram (in the C-obstacle space)
  - The initial pose
  - The goal pose
- **How to plan the shortest path that is far away from obstacles?**



From
Robot Motion Planning
J.C. Latombe

Define a graph whose nodes are:

- **Junction nodes** – where three or more arcs of the GVD intersect
- **Terminal nodes** – dead ends of the GVD
- **Initial** and **goal** configurations

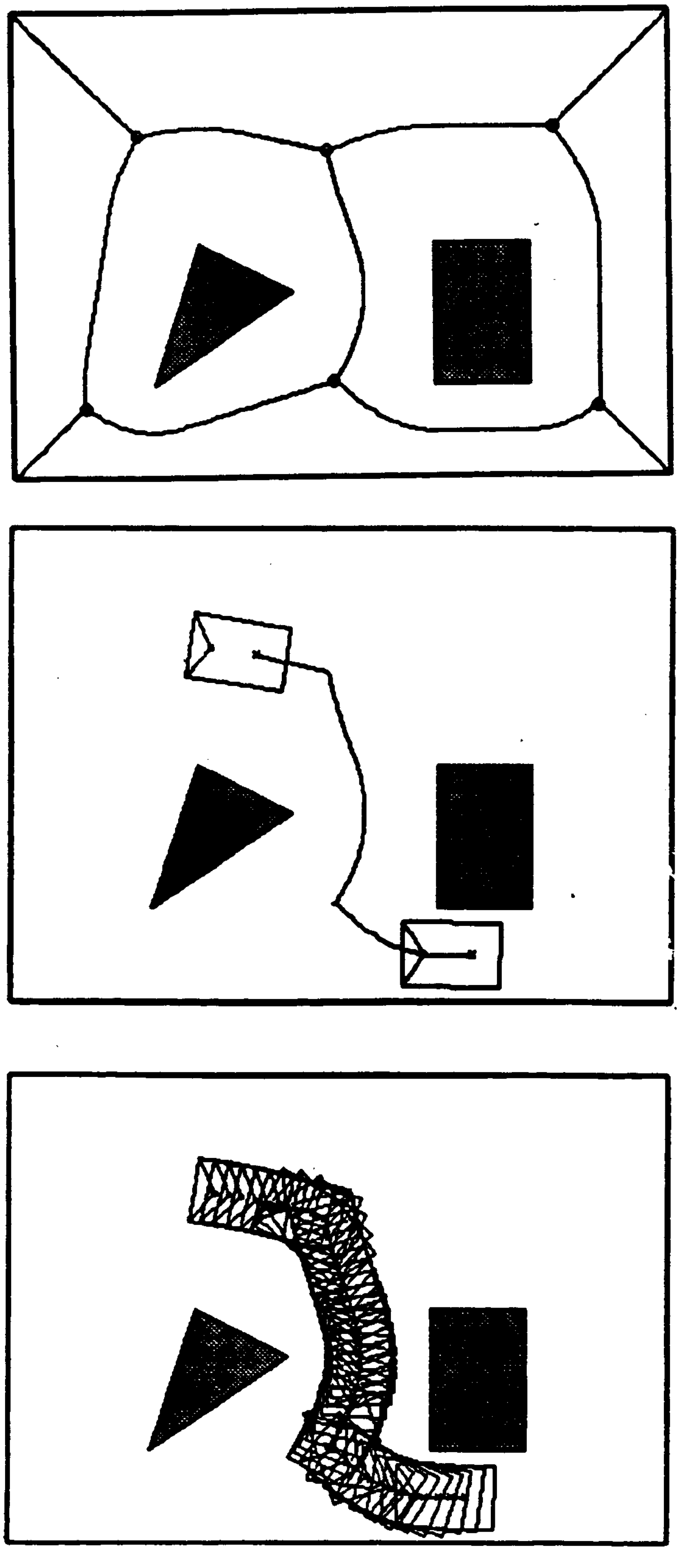
Use a graph search algorithm to find out the shortest path

- Length of the graph edges – use piecewise linear arcs to approximate arbitrarily close parabolic arcs of the GVD
- The initial and goal configurations are mapped into the closest Voronoi diagram arc.

Output of the graph search: **a set of nodes**

How to define a smooth curve that vists this set of points?

Roadmap: Voronoi Diagram

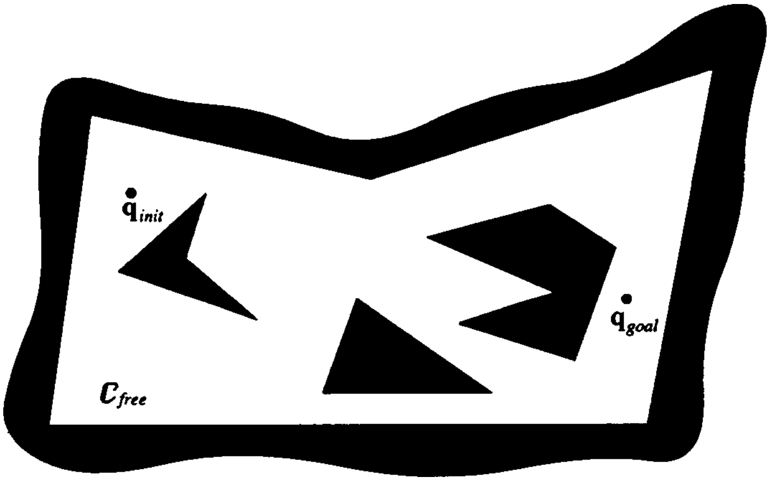


(In V.Sequeira)

- **Cell Decomposition** methods
  - The robot free space (C_{free}) is decomposed into simple regions (cells)
  - The path in between two poses of a cell can be easily generated
- Two distinct methods
  - Exact Cell Decomposition
    - The free space is decomposed into cells whose union is exactly the free space.
  - Approximate Cell Decomposition
    - The free space is decomposed in cells of pre-defined shape (e.g., squares) whose union is strictly included in the free space.

Path Planning by Exact Cell Decomposition methods

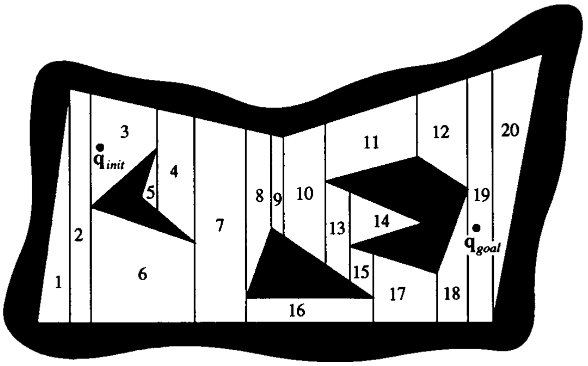
EXAMPLE



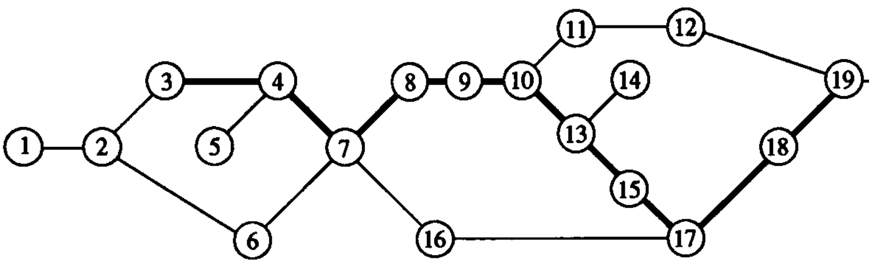
From
Robot Motion Planning
J.C. Latombe

Exact Cell Decomposition

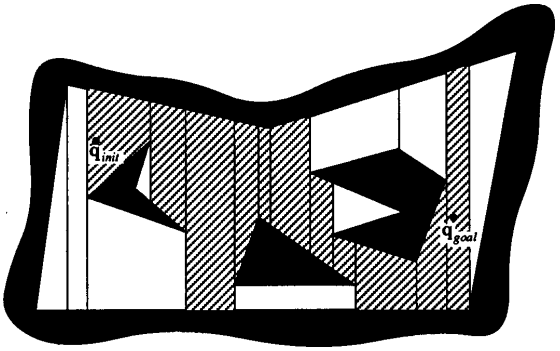
Decomposition of the free space into trapezoidal and triangular cells



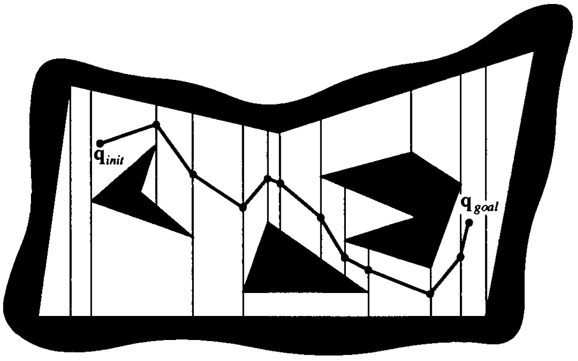
Connectivity graph representing the adjacency relation between the cells



Search the graph for a path (sequence of consecutive cells)



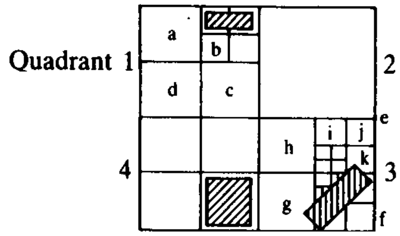
Transform the previously obtained sequence of cells into a free path (e.g., connecting the mid-points of the intersection of two consecutive cells)



Approximate Cell Decomposition

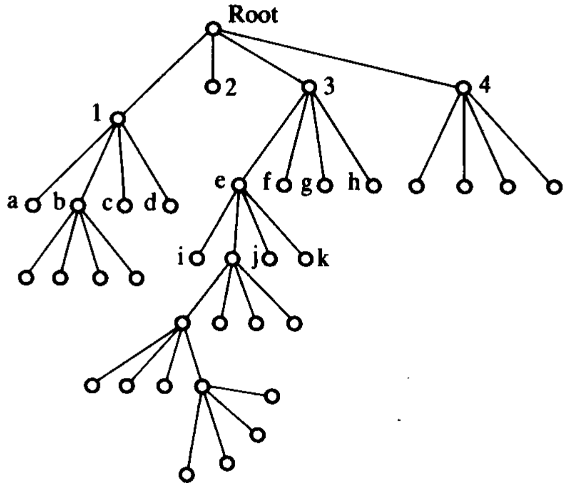
- The most usual approach: QUADTREE DECOMPOSITION

World geometric representation

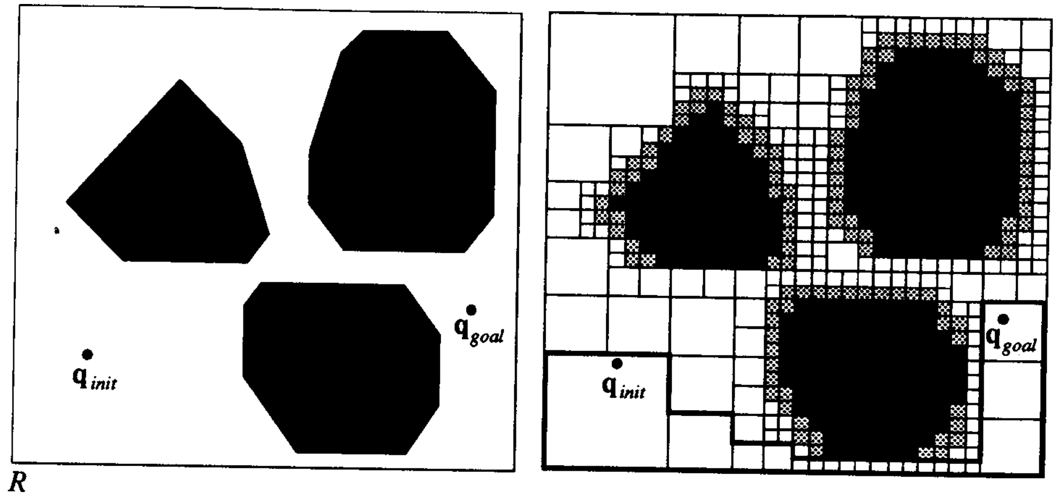


From
Introduction to Ribotics
McKerrow

Tree data structure



Each leaf is classified as empty or occupied. Representation ends in a leaf if it is occupied

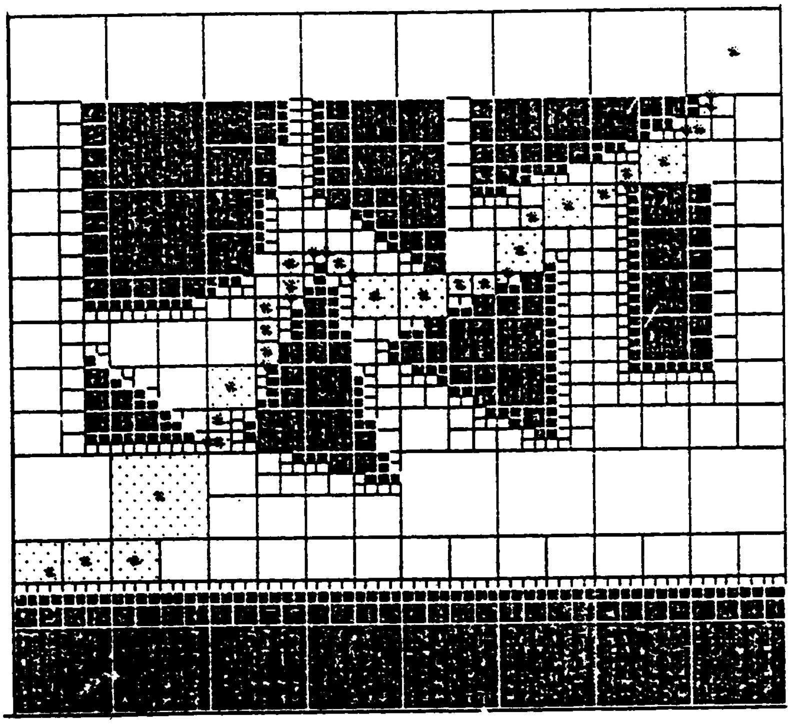


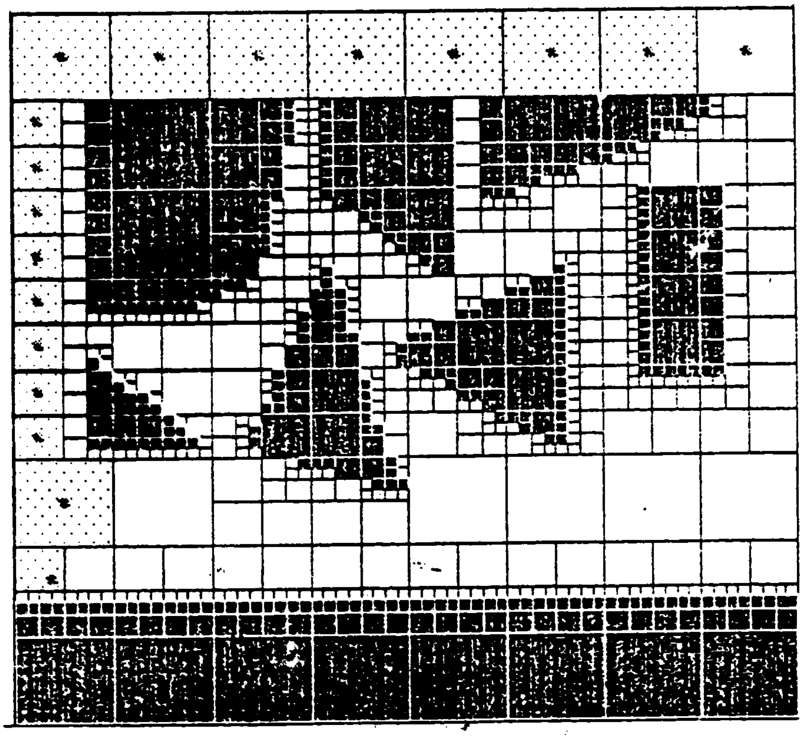
- The rectangle R is recursively decomposed into smaller rectangles
- At a certain level of resolution, only the cells whose interiores lie entirely in the free space are used
- A search in this graph yields a collision free path

From
Robot Motion Planning
J.C. Latombe

based on Quadtree Decomposition

- Represent each free cell by its central point
- Do a graph search, minimizing the total path lenght
- Result: a set of spaced points





Additional constraint:

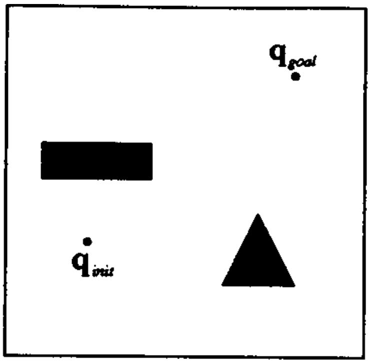
Minimum distance to an obstacle is set

(Results from R.Cabral, C.Resendes, M.Isabel Ribeiro)

Potential Field

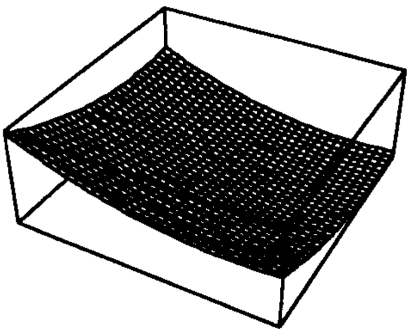
- **Potential Field** (working principle)
  - The goal location generates an **attractive potential** – pulling the robot towards the goal
  - The obstacles generate a **repulsive potential** – pushing the robot far away from the obstacles
  - The **negative gradient of the total potential** is treated as an artificial force applied to the robot

C-obstacles



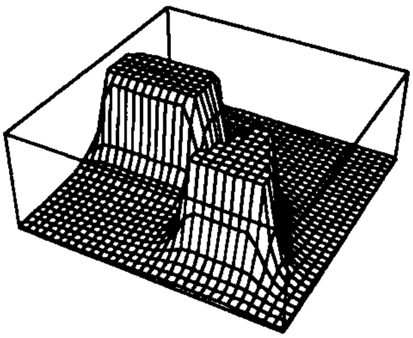
(a)

Attractive potential



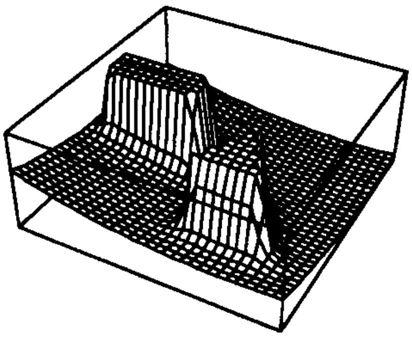
(b)

Repulsive potential



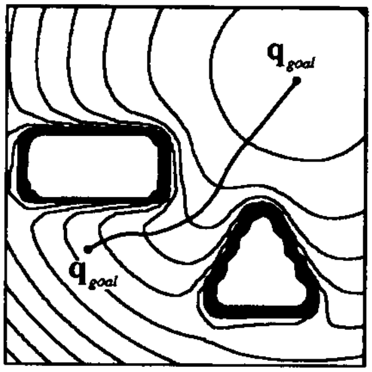
(c)

Sum of potentials



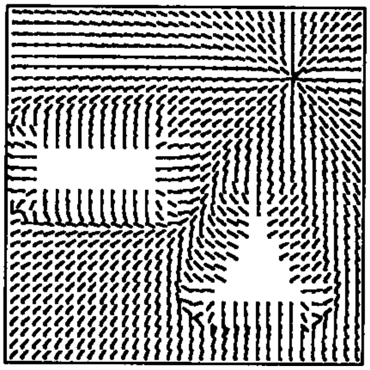
(d)

Equipotential contours



(e)

Negative gradient



(f)

From
Robot Motion Planning
J.C. Latombe

Artificial **Potential**

$$U(q) = \underbrace{U_{goal}(q)}_{\text{attractive potential}} + \underbrace{\sum U_{obstacles}(q)}_{\text{repulsive potential}}$$

Artificial **Force Field**

$$F(q) = -\nabla U(q)$$

Negative gradient

Example: free-flying robot modeled as a point

$$F(q) = -\nabla U(q) = -\begin{bmatrix} \partial U / \partial x \\ \partial U / \partial y \end{bmatrix}$$

Robot motion can then be executed by taking small steps driven by the local force

- Attractive Potential

$$U_{goal}(q) = \frac{1}{2}\xi \| q - q_{goal} \|^2$$

$$F_{att}(q) = -\xi (q - q_{goal})$$

Parabolic

Positive or null

Minimum at q_{goal}

Tends to zero when the robot gets closer to the goal configuration

Different functions have also been used

- Repulsive Potential
  - Create a potential barrier around the C-obstacle region that cannot be traversed by the robot's configuration
  - It is usually desirable that the repulsive potential does not affect the motion of the robot when it is sufficiently far away from C-obstacles

- **Repulsive Potential** (ctn)

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2 & \text{if} \quad \rho(q) \le \rho_0 \\ 0 & \text{if} \quad \rho(q) > \rho_0 \end{cases}$$

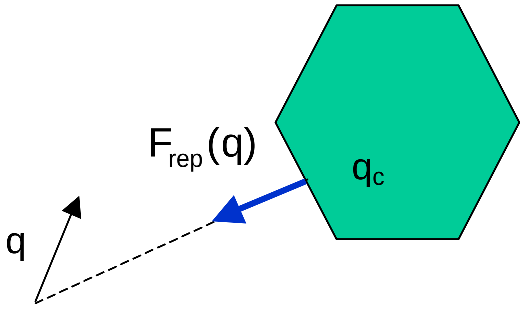
η → positive scaling factor

$\rho(q) = \min_{q' \in CB} \| q - q' \|$ → Distance from the actual configuration q to the C-obstacle region CB

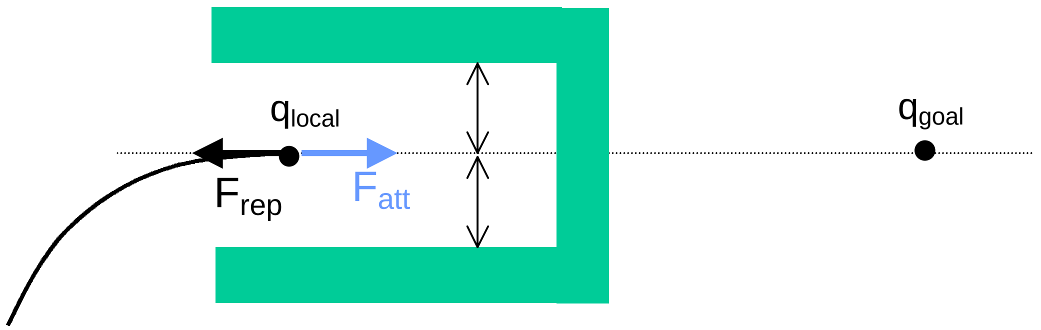
ρ_0 → Positive constant (**distance of influence**) of the C-obstacles

$$F_{rep}(q) = \begin{cases} \eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2(q)}\nabla\rho(q) & \text{if} \quad \rho(q) \le \rho_0 \\ 0 & \text{if} \quad \rho(q) > \rho_0 \end{cases}$$

Example with a single obstacle



- Potential Fields method suffers from **LOCAL MINIMUM problem**



$F_{rep} = -F_{att}$ ⟶ Local Minimum

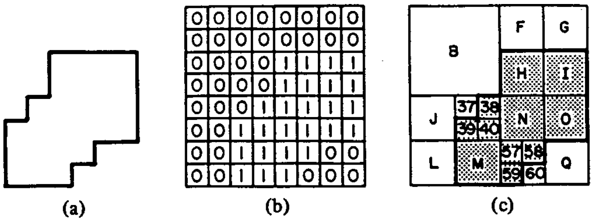
Local Minimum results, usually, from simmetry situations on CB

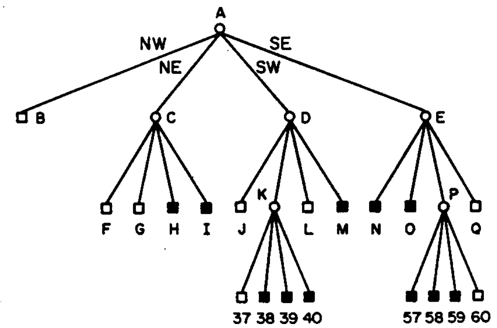
- Solutions to overcome Local Minimum problems

(see in J. Latombe

The A* algorithm in Path Planning

- Multiresolution Planners (an example)
  - S.K.Kambhampati, L.S.Davis, “Multiresolution Path Planning for Mobile Robots” IEEE Journal of Robotics and Automation, RA-2, 3, pp.135-145, 1986.





Quadtree representation of world model

ALGORITHM

1. Robot ---→ point. The objects are extended to include the robot dimensions.
2. Givent the Start and Goal points, determine the quadtree leaf nodes S and G, representing the regions containing these points.
3. Plan a minimum cost path between S and G in the graph formed by the non-obstacle leaf nodes, using the A* algorithm.

<u>Cost function at an arbitary node n</u>

$$f(c) = g(c) + h(c)$$

Heuristic estimate of the cost of the remaining path from n to G

Cost of the path from S to c

$$g(c) = g(p) + \tilde{g}(p,c)$$

Cost of the path segment between p and c

Cost of the path from S to the c’s predecessor p on the path

Cost function at an arbitary node n

$$f(c) = g(c) + h(c)$$

Heuristic estimate of the cost of the remaining path from n to G

Euclidean distance between the midpoints of the regions represented by c and G

Cost of the path from S to c

$$g(c) = g(p) + \tilde{g}(p,c)$$

Cost of the path segment between p and c

Cost of the path from S to the c's predecessor p on the path

$$\tilde{g}(p,c) = D(p,c) + \alpha\, d(c)$$

Cost incurred by including node c on the path. d(c) depends upon clearance of the node c from the nearby obstcales

Distance between nodes p and c (half the sum of the nodes size)

Extensions to the Basic Problem

The basic motion planning makes assumptions that significantly limit the praticability of its solutions.

Extensions (some of the most usual)

- **Multiple Moving Obstacles**
  - Moving obstacles (non static environment)
  - Multiple robots in the same working space
  - Articulated robots
- **Kinematics Constraints**
  - Holonomic Constraints
  - Nonholonomic Constraints
- **Uncertainty**

Question: how to extend the path planning methodologies to cope with these conditions ?

Extensions to the Basic Problem

Moving Obstacles

- The motion planning problem can no longer be solved by merely constructing a geometric path
- Time has to be taken into account
- **Add dimension time to the configuration space**

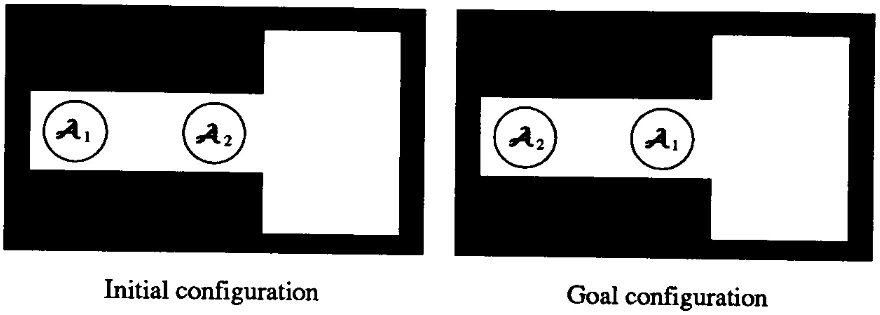
CT - configuration time space

CT - obstacles

- **Motion planning = finding a path among the CT-obstacles in CT.**

Multiple Robots

- Different from planning with moving obstacles in that the motions of the robots have to be planned while the motions of the moving obstacles are not under control



From
Robot Motion Planning
J.C. Latombe

- Two discs have to interchange their positions in a narrow corridor where they cannot pass each other
- There is enough room space for permutation at one end of the corridor
- Decoupled planning, would very likely fail to solve this planning problem

- Consider a single multi-bodied robot **$A=\{A_1,A_2\}$**
- Solve the Path Planning problem in a **composite configuration space $C=C_1 x C_2$**

Kinematic Constraints

- In the basic path planning problem the only constraint of robot motion is due to obstacles
- There may occur other contraints – kinematic constraints (objects that cannot translate and rotate freely in the workspace)
- Two types of kinematic constraints
  - **Holonomic Constraints**
    - Do not fundamentally change the path planning problem
  - **Nonholonomic Constraints**
    - Much harder to deal in terms of path planning

C – configuration space – dimension m

$$q \in C, \quad q = [q_1 \quad q_2 \quad \ldots \quad q_m]^T$$

Holonomic Constraints

$$F(q,t) = 0$$

F is a smooth function with non-zero derivative

$$F(q_1, q_2, \ldots, q_m, t) = 0$$

1 holonomic constraint = Relation (equality or inequality) among the parameters of **C** that can be solved for one of them as a function of the others

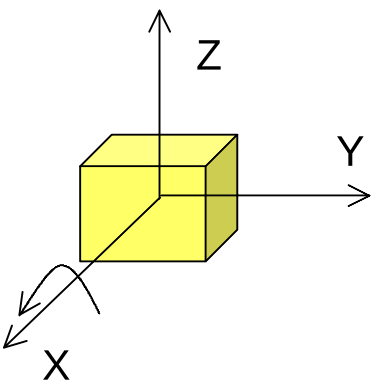
k independent holonomic constraints → **dim C=m-k**

Path planning is done in a submanifold of C with dimension m-k as if there were no constraints

Holonomic Constraints

Holonomic Constraints - Example

- A – tridimensional object that:
  - Can freely translate
  - Has a rotation along a fixed axis (relative to F_A)



- Pitch angle = yaw angle = 0
- These two independent equations constraints the configuration

dim C = 6

$$q = (x, y, z, \theta, \varsigma, \psi)$$

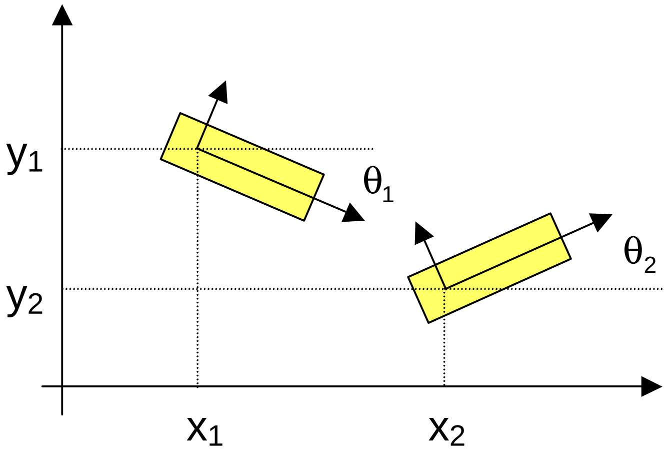
$$\zeta = 0, \;\; \psi = 0$$ holonomic constraints

dim = 4

Holonomic Constraints

Articulated Robots

- 2 Planar Robots

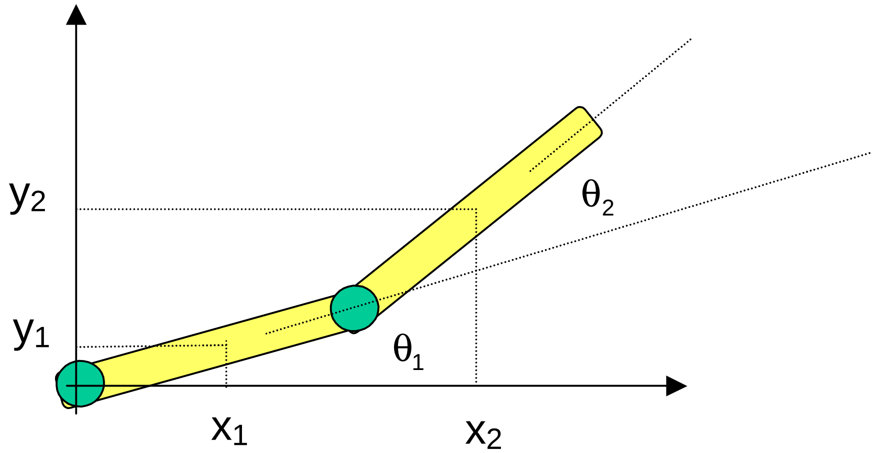


$$q = (x_1, x_2, y_1, y_2, \theta_1, \theta_2)$$

$$\dim\ C = 6$$

there are no holonomic constraints

- Planar Manipulator with two links



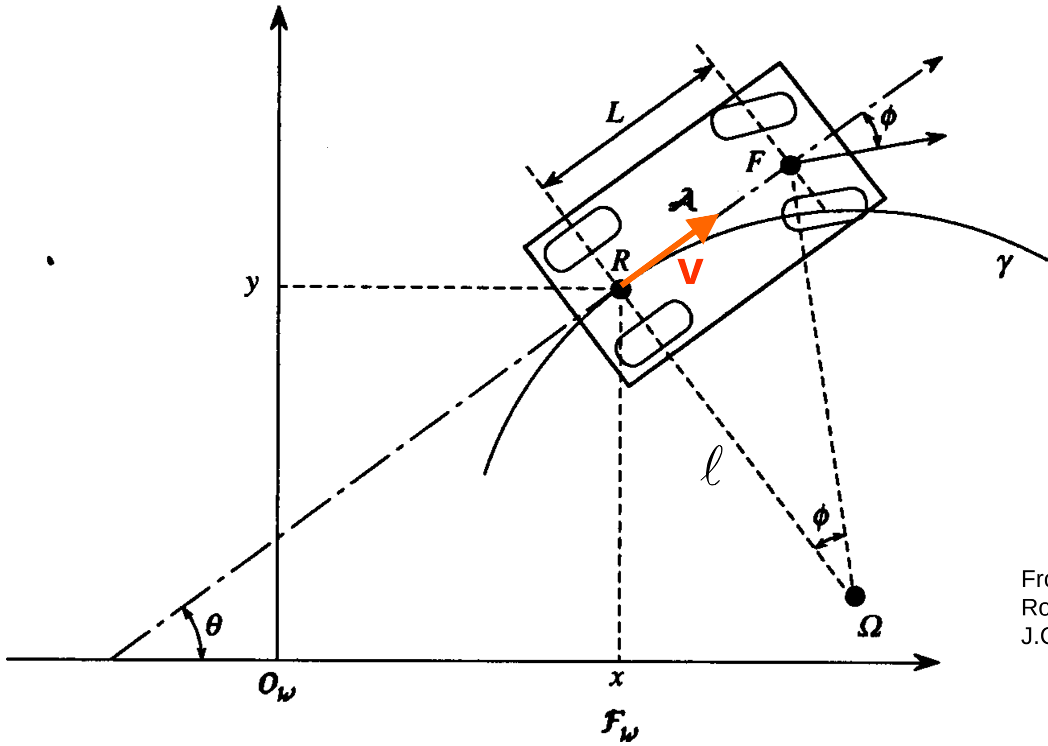
$$C' = (x_1, x_2, \theta_1, \theta_2)$$

$$\dim\ C' = \dim C - 2 = 4$$

there are two holonomic constraints

Nonholonomic Constraints

Nonholonomic Constraints – Example (car like robot)



From
Robot Motion Planning
J.C. Latombe

- In an empty space we can drive the robot to any position and orientation
- q=(x,y,θ) – configuration (position + orientation)
  ϕ – steering angle
- C (configuration space) has dimension 3

- If there is no slipping, the velocity of point R has to point along the main axis of **A**

$$\frac{dy}{dx} = \mathrm{tg}\theta \quad \Rightarrow \quad \frac{dy}{dx} = \frac{\sin\theta}{\cos\theta}$$

$$-\sin\theta\, dx + \cos\theta\, dy = 0$$

Contraint
non-integrable equation

Nonholonomic constraint

Nonholonomic Constraints

$$-\sin\theta\, dx + \cos\theta\, dy = 0$$

$$\dot{x} = v\cos\theta \qquad \dot{y} = v\sin\theta$$

$$\dot{\theta} = \frac{v}{L}\mathrm{tg}\phi = \frac{v}{\ell}$$

ℓ - Curvature of the path $\qquad \ell = \dfrac{L}{\mathrm{tg}\phi}$

- The space of differential motions (dx, dy, dθ) of the robot at any configuration is a two-dimensional space.
- If the robot was a free-flying object this space would be three-dimensional.

Limited steering angle
An additional constraint

$$\left.\begin{array}{r}\phi \in [-\phi_{max}, \phi_{max}] \\ \phi_{max} < \pi/2\end{array}\right\} \Rightarrow |\phi| \le \phi_{max} < \pi/2$$

$$\ell_{min} = \frac{L}{\mathrm{tg}\phi_{max}}$$

$$|\dot{\theta}| \le \frac{|v|}{\ell_{min}} \rightarrow \dot{x}^2 + \dot{y}^2 - \ell_{min}^2\dot{\theta}^2 \ge 0$$

$$F\left(\frac{dx}{dt}, \frac{dy}{dt}, \frac{d\theta}{dt}\right) \ge 0$$

Nonholonomic Constraints

Nonholonomic Constraints

- Non-integrable equation involving the configuration parameters and their derivatives (velocity parameters)
- Nonholonomic constraints
  - **do not reduce** the dimension of the configuration space attainable by the robot
  - **reduce** the dimension of the possible differential motions (i.e., the space of the velocity directions) at any given configuration

$$F(q, \dot{q}, t) = 0$$

$$F(q_1, q_2, \ldots, q_m, \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_m, t) = 0$$

F is a smooth function with non-zero derivative

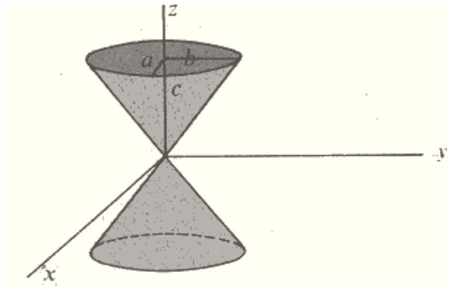
- If F is integrable – the equation can be written as a holonomic constraint
- If F is non-integrable – the constraint is nonholonomic

- dim of **C** does not change
- dim (velocity or differential motion space) = dim C – nº of independent restrictions

Nonholonomic Constraints
Car-like robot with minimum turning radius

$$\begin{cases} -\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \\ \dot{x}^2 + \dot{y}^2 - \ell_{min}^2\dot{\theta}^2 \geq 0 \end{cases}$$

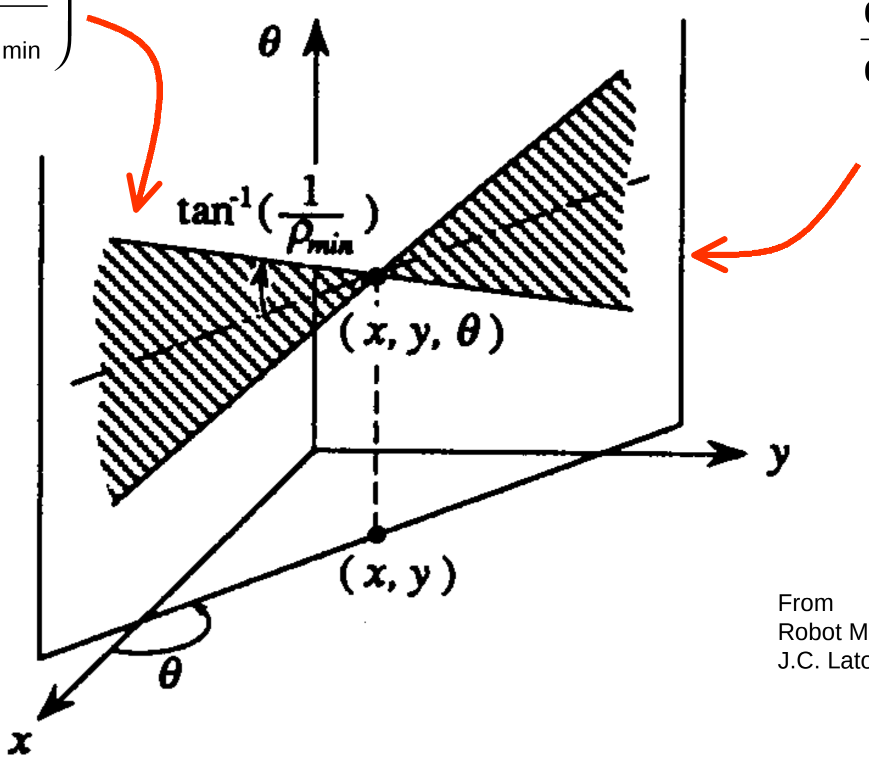
Velocity vector $(\dot{x}, \dot{y}, \dot{\theta})$ satisfies this inequality due to a limited turning angle



$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$$

$$\phi_{max} = \operatorname{arctg}\left(\frac{L}{\ell_{min}}\right)$$

$$\frac{dy}{dx} = \operatorname{tg}\theta$$



From
Robot Motion Planning
J.C. Latombe

Nonholonomic Constraints

$$F(q,\dot{q}) = 0$$

- Do non-holonomic constraints restrict the set of configurations achivable by the robot ?
- A non-integrable contraint $F(q,\dot{q}) = 0$ restricts the set of possible velocities of the robot to a vector subspace of the tangent space of q, T_q.
- But does this restricts the set of achievable configurations?

- Control vector (the vector that creates motion) = Robot velocity vector = $\dot{q}$
- In the holonomic case, at any configuration q, the control space coincides with the tangent space $T_q(C)$.
  - Every configuration q' in a small neighborhood of q can be achieved from q by selecting a vector $\dot{q}$ appropriately.
  - This is no longer true in the non-holonomic case, where the dimenson of the control space is smaller than that of the tangent space.
- Controllability concept

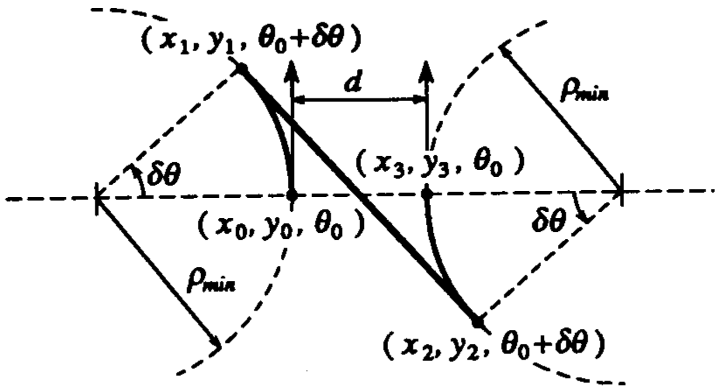


Figure 5. A maneuver of type 1 allows the car-like robot to translate sidewise. Under the condition $|\delta\theta| < \pi/2$, the net length d of the translation is equal to $2\rho_{min}(\frac{1}{\cos\delta\theta} - 1)$, which is strictly positive for any $|\delta\theta| > 0$. The maneuver shown in this figure results in a translation of the robot toward its right. A similar maneuver would produce a leftwise translation.

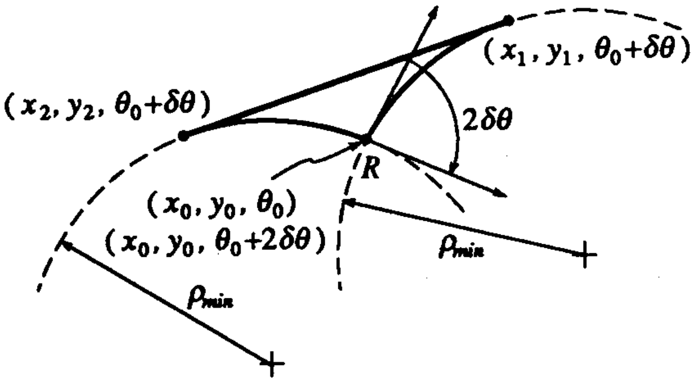


Figure 6. A maneuver of type 2 allows the car-like robot to rotate around the point R, as if it had a zero turning radius. The maneuver shown in the figure makes the robot rotate clockwise. A similar maneuver would result in a counterclockwise rotation.

From
Robot Motion Planning
J.C. Latombe

Manouvers for a car-like robot

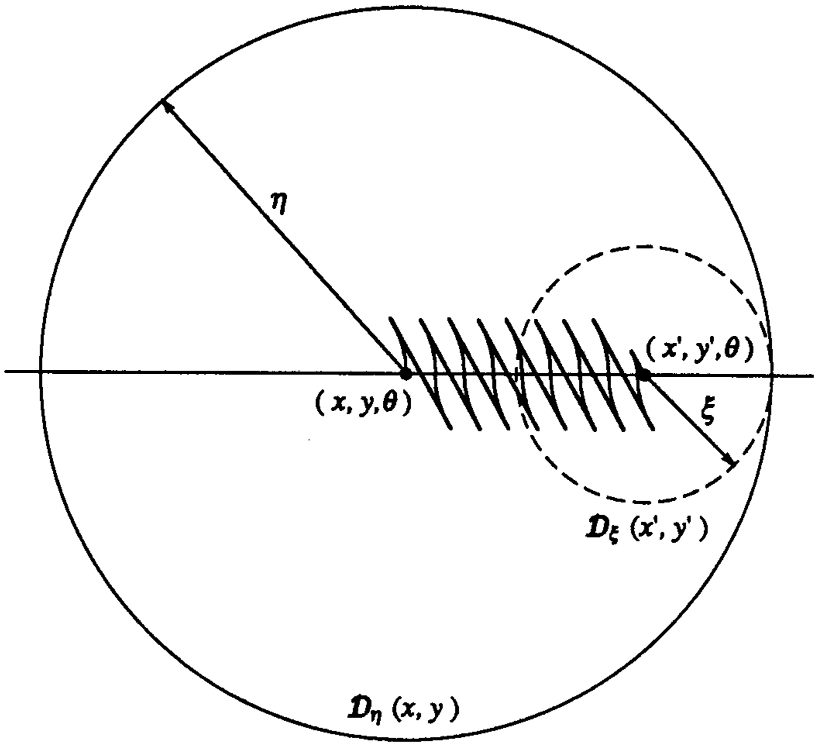


Figure 7. The two configurations (x, y, θ) and (x', y', θ) can be connected by a feasible path completely contained in the cylinder $CYL(x, y, \theta, \eta, \varepsilon)$ by executing a finite number of maneuvers of type 1 (see Lemma 1).

From
Robot Motion Planning
J.C. Latombe

Examples

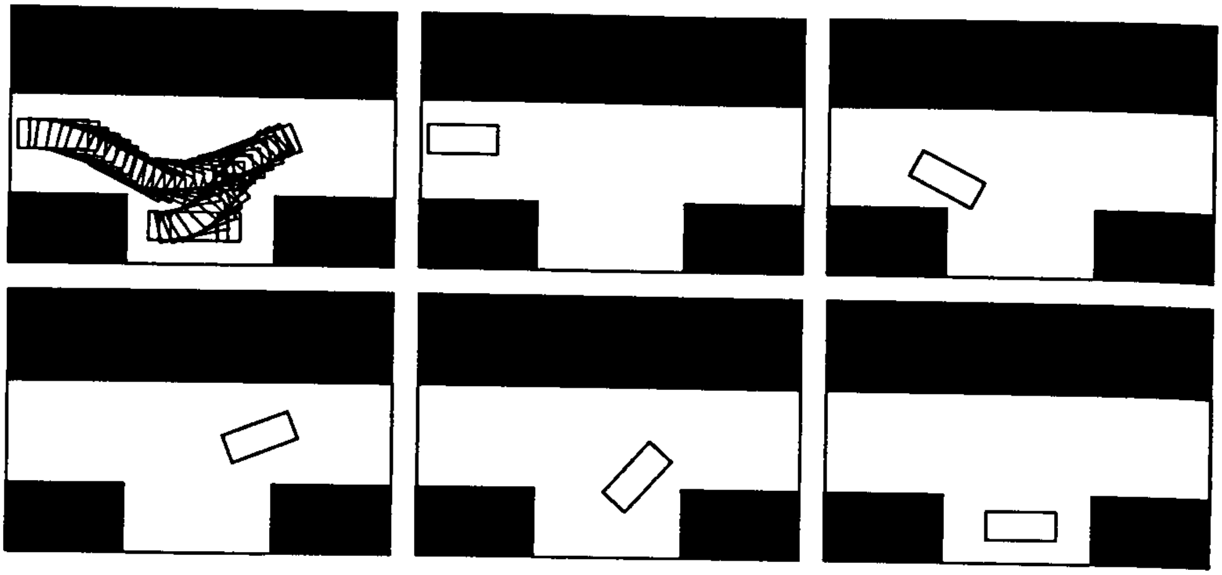


Figure 9. This figure shows a path constructed by the planner described in Subsection 6.1 for the classical parallel parking problem. The steering angle of the car-like robot is limited to 30 degrees.

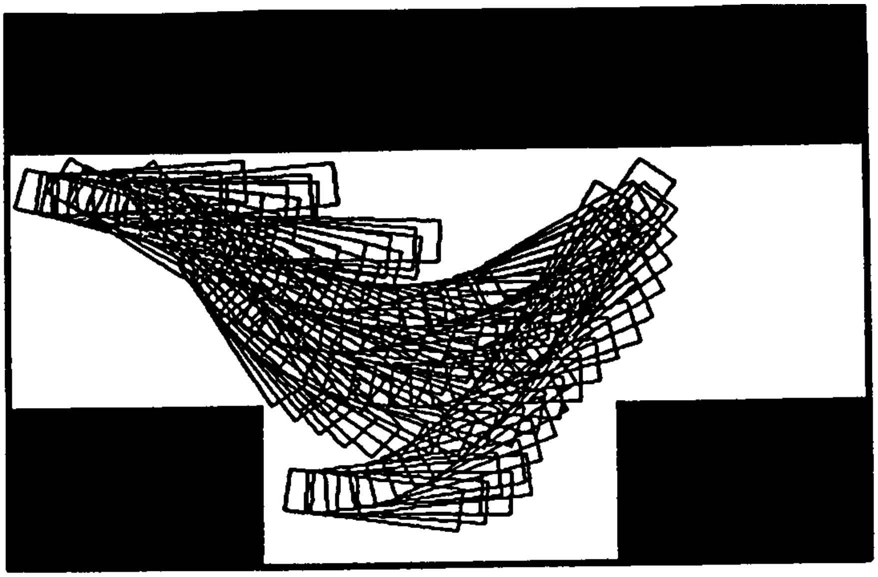


Figure 10. This figure shows a path constructed by the planner described in Subsection 6.1 for the parallel parking problem with a car that can only turn left (22.5 degrees $\leq \phi \leq$ 45 degrees).

From
Robot Motion Planning
J.C. Latombe

Examples

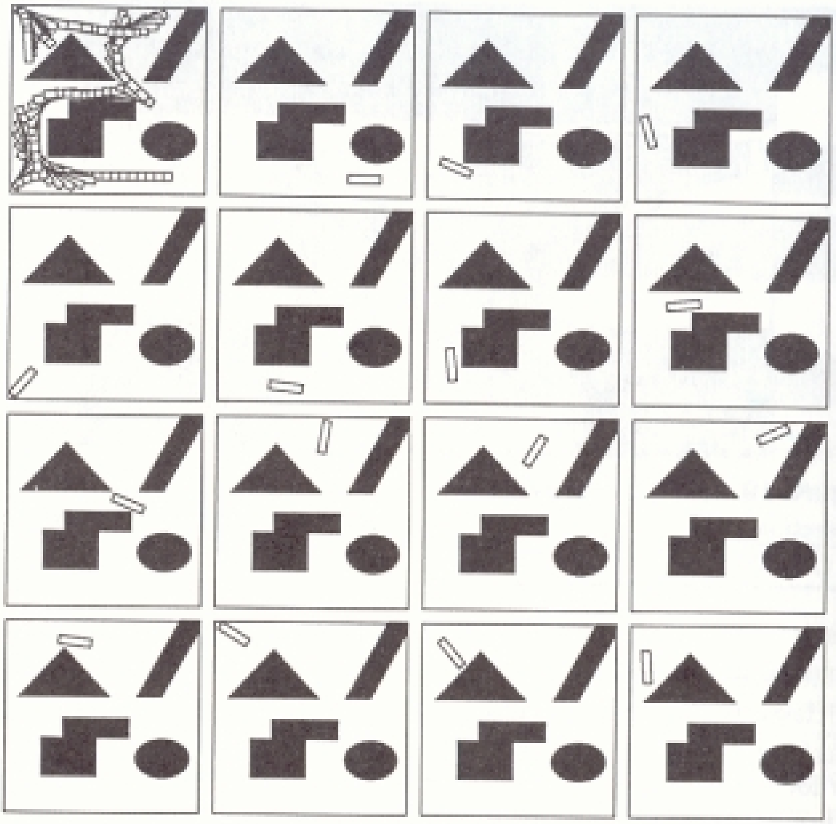


Figure 11. This figure shows another path constructed by the planner of Subsection 6.1. The cluttered environment and the limited steering angle (45 degrees) require the car-like robot to perform multiple reversals.

From
Robot Motion Planning
J.C. Latombe

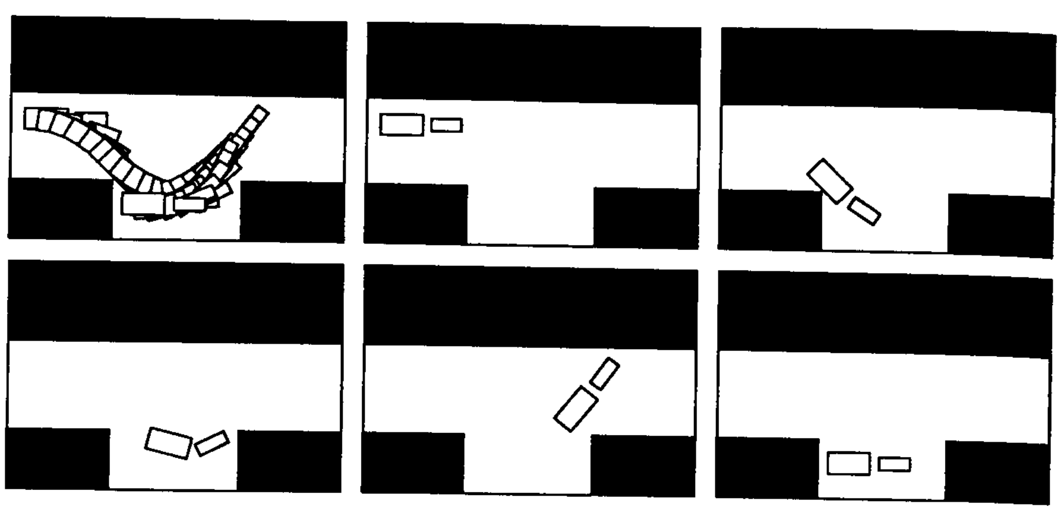


Figure 13. This figure illustrates the parallel parking problem for a tractor-trailer robot. It shows a path generated by the planner described in Subsection 6.2. The steering angle is limited to 30 degrees.