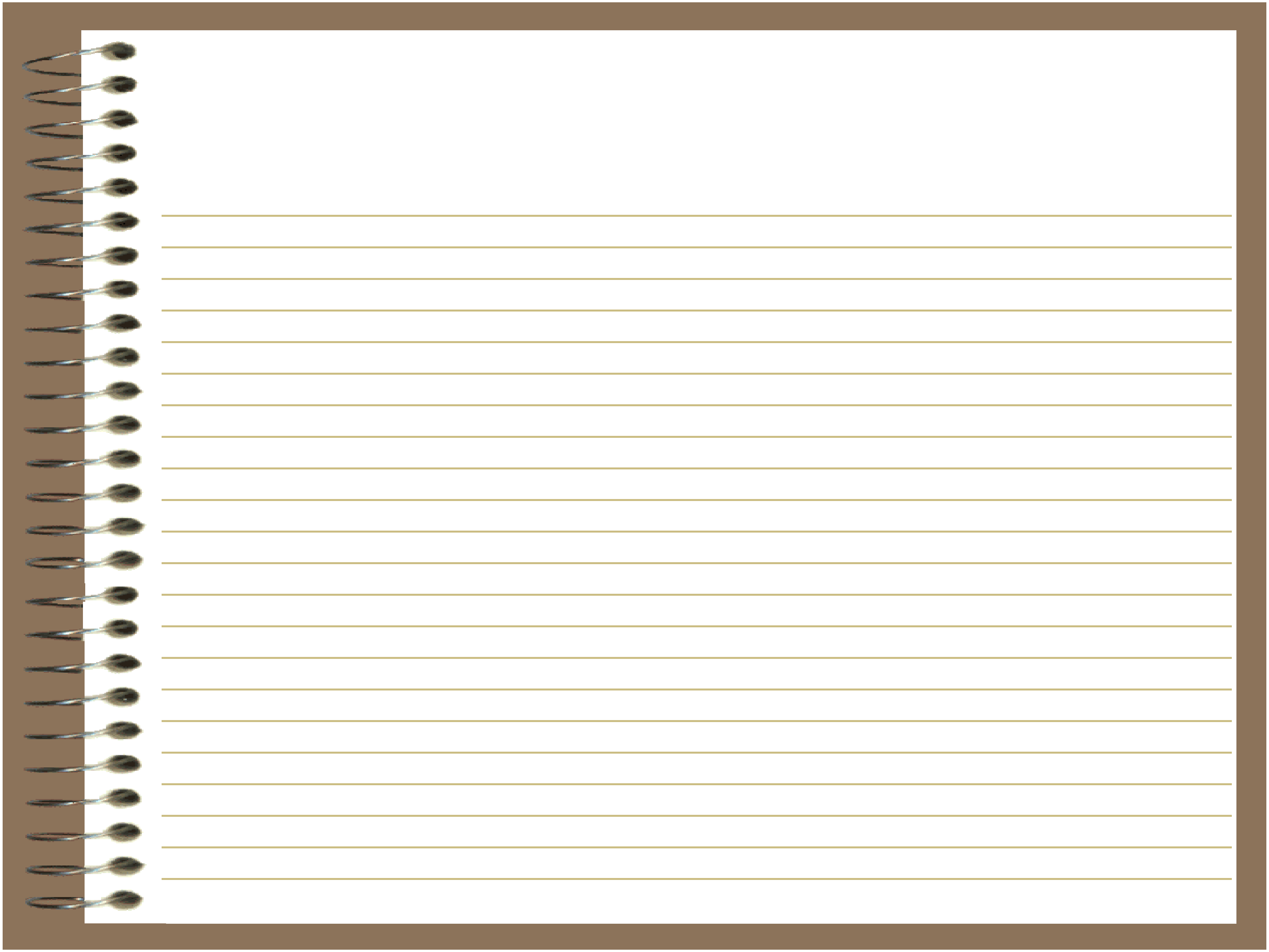


Task Planning

- High level planning is often required to achieve truly automated production
- “Planning is the process of deciding a course of action before acting”
- Controlling and actual execution is carried out only later
- Problem solving techniques for robotic task planning

Implementation

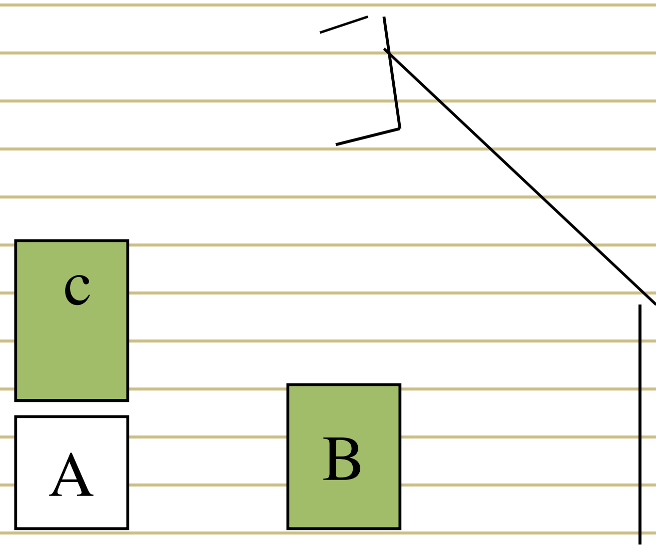
- On-line vs Off-line programming
  - Programming languages, Simulation software
  - Teach pendant
- Goal driven planning
  - Robot independent
  - World Model - Robot database, environment, manipulation task etc needed for task planning
  - Automated plan, optimum solutions

State Search

- Finding possible solutions until the desired results are achieved - Trial and error
- Problem state
  - Set of all possible configuration
- Operator
  - When applied changes the state from one to another
- Solution - sequence of *operator* that changes one initial state to the final

Block world example

- Example of Blocks



- Stack A on top, B in middle and C at the bottom

Block

- The operator that robot can use is MOVE X from Y to Z
  - MOVE(X,Y,Z)
- Requirements:
  - X, the object to be moved, be a block with nothing on top of it
  - if Z is a block, there must be nothing on it
- Draw a graph of the state

Solution state

- MOVE(C,A,T)
- MOVE(B,T,C)
- MOVE(A,T,B)
  - What is the state space for this problem ?

Problem

Homework - submit it on October15.

- Suppose that three missionaries and three cannibals seek to cross a river from the right bank to the left bank by a boat. The maximum capacity of the boat is two persons. If the missionaries are outnumbered at any time by the cannibals, the cannibals will eat the missionaries. Propose a solution state for safe crossing of all six persons.
  - Continued next page

State space representation

- Using state descriptors (Nm, Nc) where Nm is number of missionaries and Nc the number of cannibals solve the problem. The initial state is (0,0) on the left bank and the final state is (3,3).

Robotic Task Planning

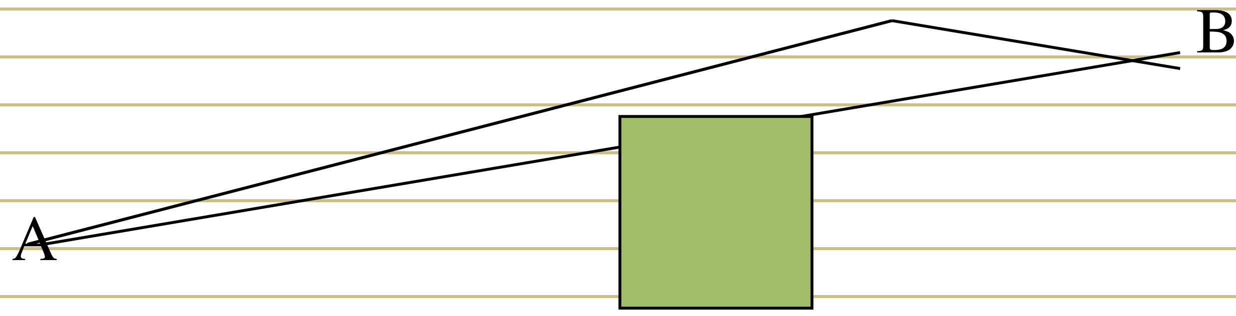
- Pickup (a) and Stack (x,y) without specifying robot path: Inputs, outputs and environment?
  - Geometric description
  - Physical description
  - Kinematics
  - Sensors
  - Methods to execute path based on the goal state.

Obstacle Avoidance

- Non-collision Path Planning
- Three methods
  - Hypothesize and test
  - Penalty function
  - Explicit Free space
    - Configuration space, GVD representations

Hypothesize and test

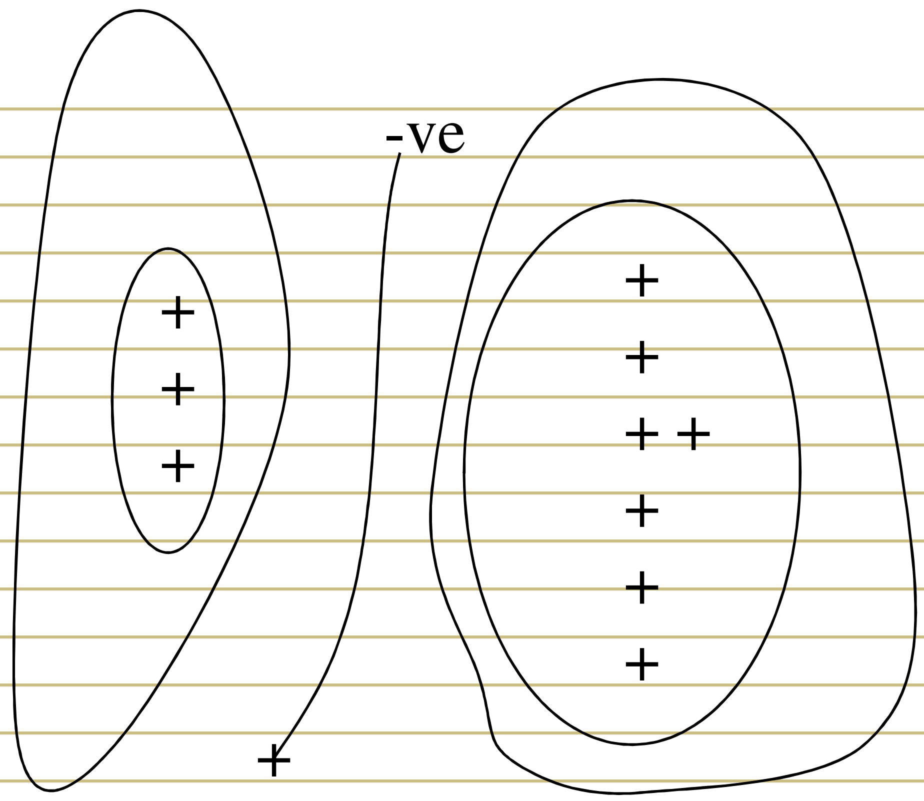
- Select a planned path
- Check for possible collisions
- Correct the path



Penalty Function

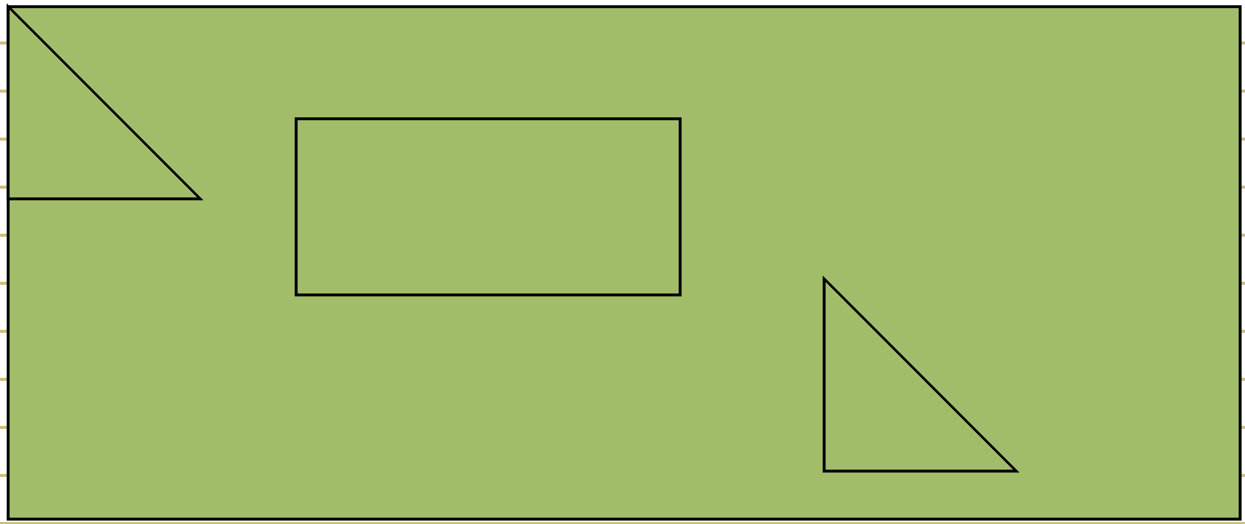
- Infinite penalty for obstacle
- Low penalty for path away from obstacle
- High penalty for non-optimal path
- Potential field method (Khatib '80)
  - Repelling forces between robot and obstacle
  - Attracting forces between goal and robot

Potential function



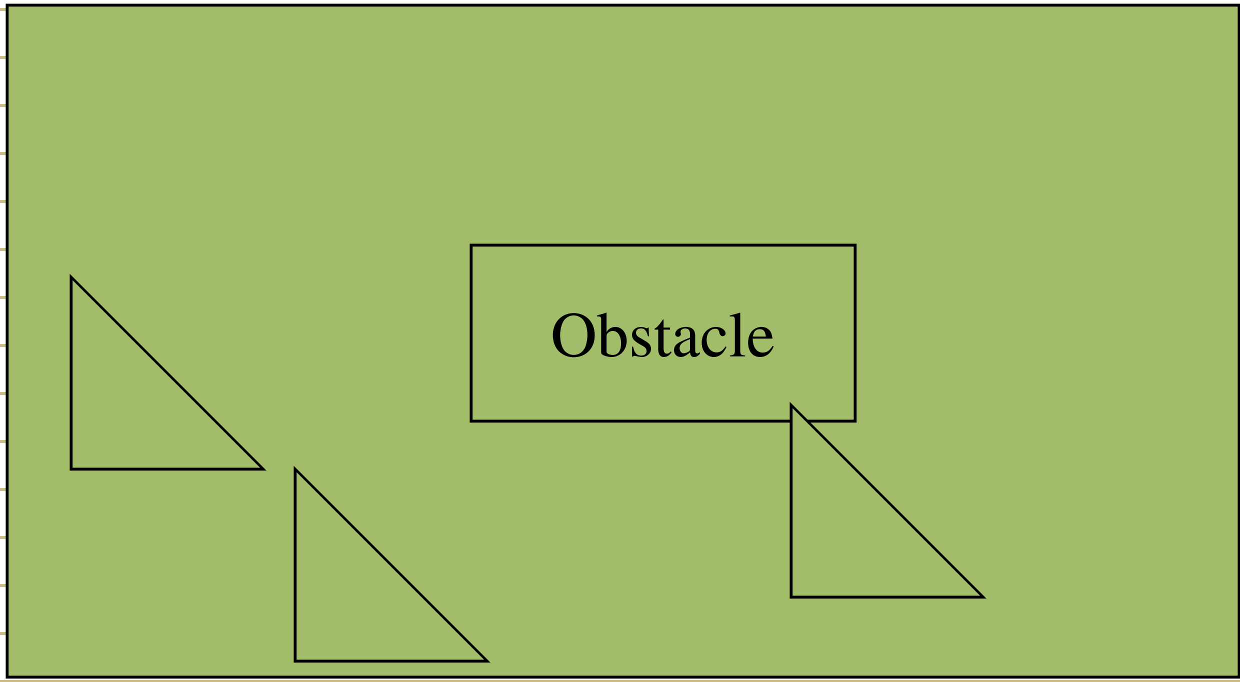
Configuration space

- Translation only
- Translation and rotation



Moving object

- Configuration space ?



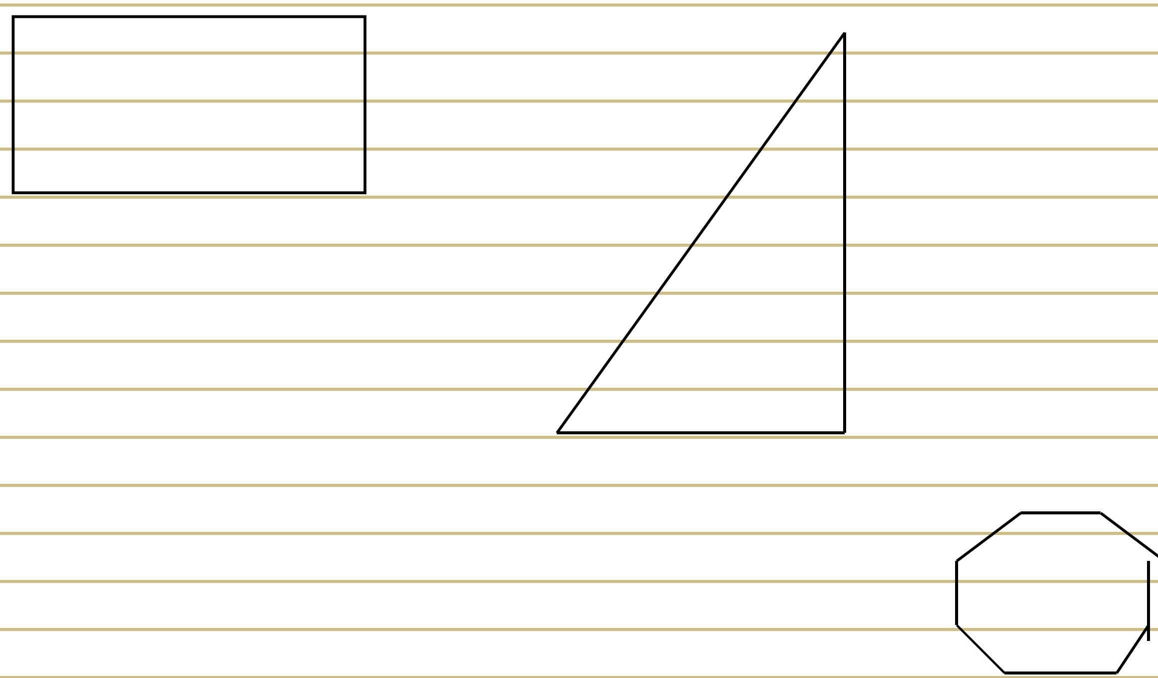
Gross motion plan for convex object

- Finding shortest path
  - with only translation
  - with rotation and translation
    - Enclose the rotating part in a convex hull
- Motion plan using Generalized Voronoi diagram (GVD)

Grasp Planning

- Safe grasp
- Reachable grasp
- Look Ahead
- Intermediate configuration
- Secure grasp
  - Parallel jaw gripper

How to grasp?



Other conditions?

- Rotation?
- Gravity?
- Irregular surface?
- Opposing surfaces
- Friction on jaws etc

Announcements

- Reminder: Your mid term test will be held on October 16, 2007.