

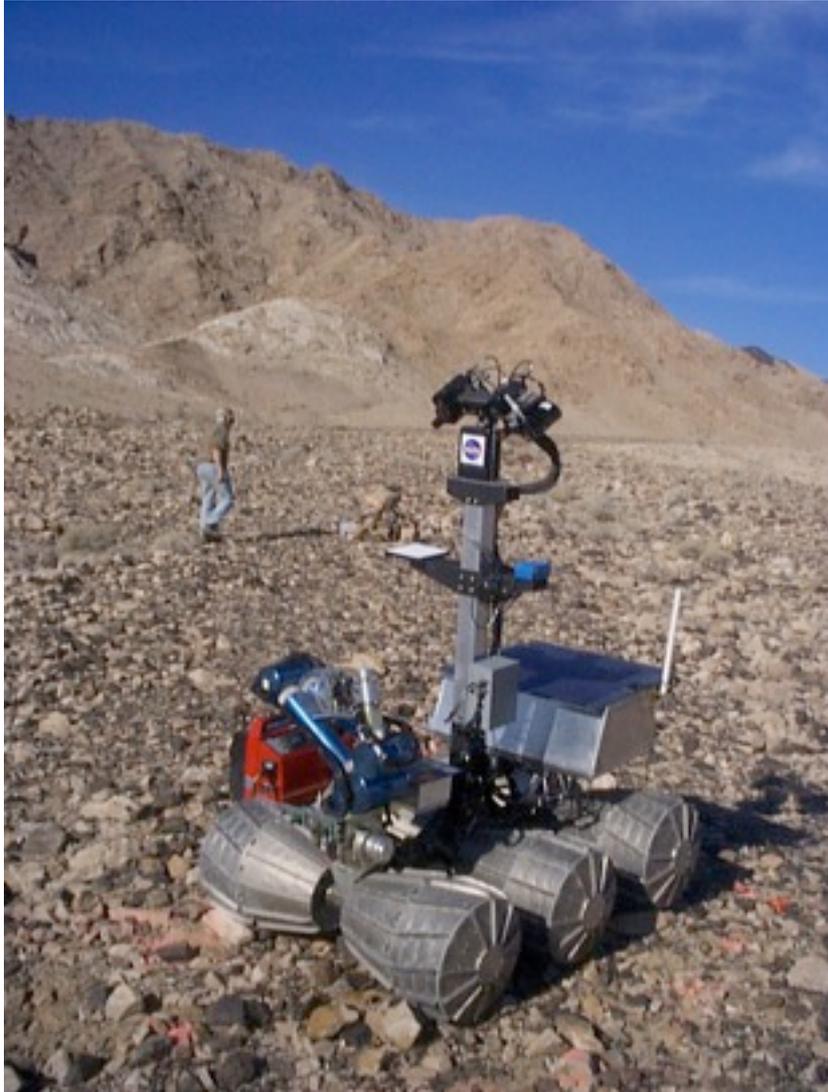
# Planning from Head to Toe

David E. Smith  
NASA Ames Research Center



??

# 1997-2000: Marsokhod



# 2001-2005: K9 rover



# 2007-2008: ATHLETE



# 2009-: A Different Kind of Rover



# Some Rover Peculiarities

- Hazardous environments
  - Slow rad-hardened processors (200 MHz)
  - Low power (125 Watts)
  - Limited memory (256 MB)
  - Limited storage (2 GB)
- Unstructured rough terrain
  - Navigation/localization difficult
- Limited autonomy
  - Local obstacle avoidance
  - Opportunistic pictures



# Some Rover Peculiarities

Ignore

- ~~Hazardous environments~~
  - Slow rad-hardened processors (200 MHz)
  - Low power (125 Watts)
  - Limited memory (256 MB)
  - Limited storage (2 GB)



Ignore

- Unstructured rough terrain
  - Navigation/localization difficult
- ~~Limited autonomy~~
  - Local obstacle avoidance
  - Opportunistic pictures



# The Planning Problem

- Temporal

- Action durations
  - Concurrency

- Time constraints

- Communication windows
  - Illumination of targets
  - Temperature

- Uncertainty

- Terrain & tracking
  - Duration of actions
  - Energy usage
  - Storage available

- Oversubscription

- Many conflicting goals
  - Goal dependence



# The Talk



- Temporal
  - Action durations
  - Concurrency
- Time constraints
  - Communication windows
  - Illumination of targets
  - Temperature
- Uncertainty
  - Terrain & tracking
  - Duration of actions
  - Energy usage
  - Storage available
- Oversubscription
  - Many conflicting goals
  - Goal dependence



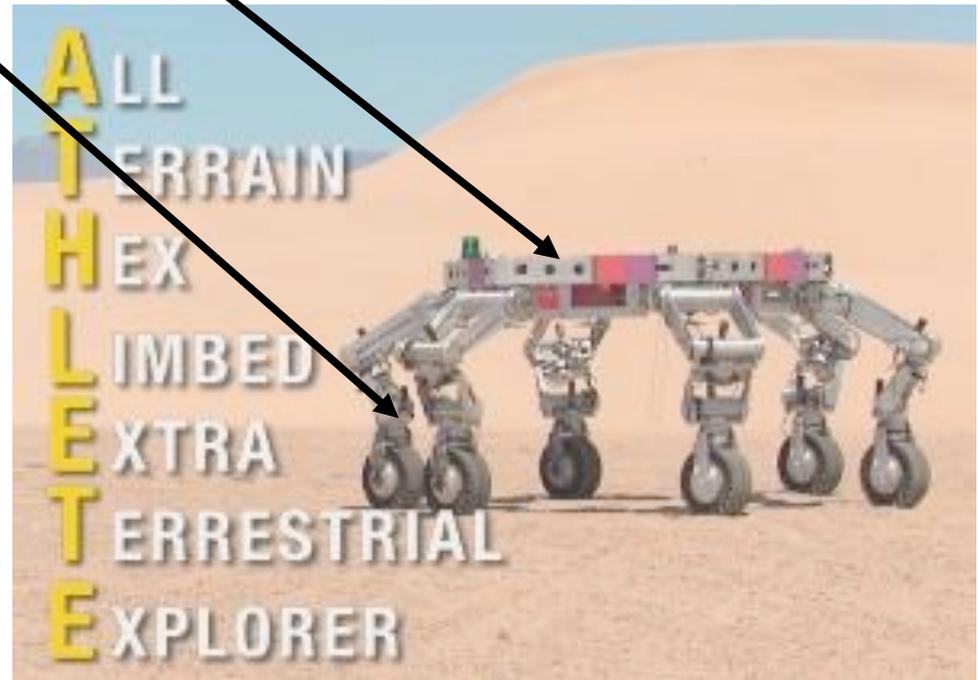
## Levels of planning



# ATHLETE

- 6 legs, 36 degrees of freedom
- Feet are wheels (walk and roll)
- 6 stereo camera pairs outward
- 3 stereo camera pairs inward
- 1 stereo pair on each foot
- 2.75m chassis
- 850 kg

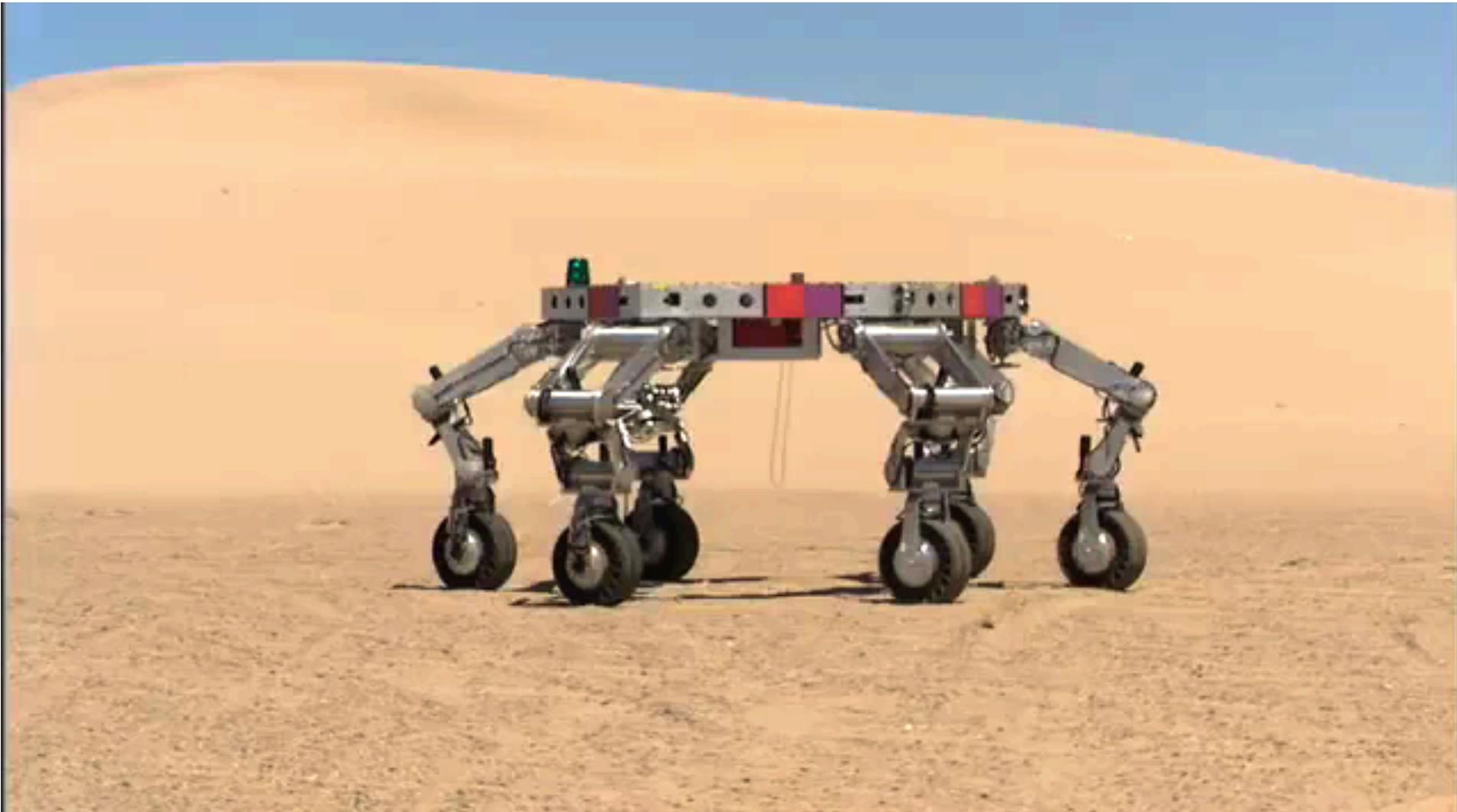
1/3 scale



# ATHLETE Capabilities



# ATHLETE in action

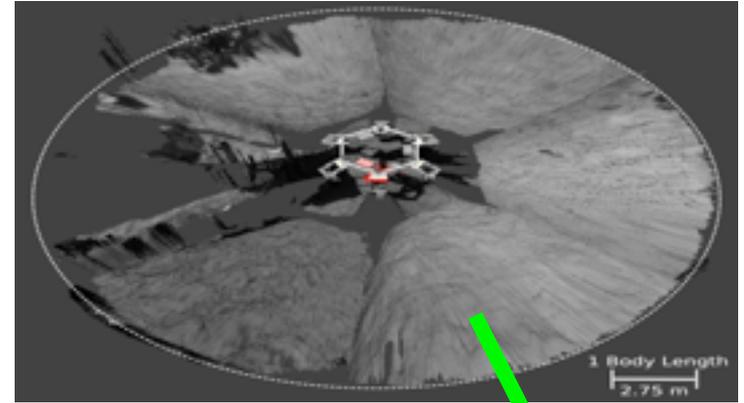


# ATHLETE in action

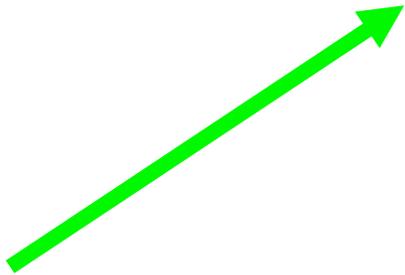
---

# Current Operation

- Remotely operated
- Rolling: **ok**
- Walking: **slow**

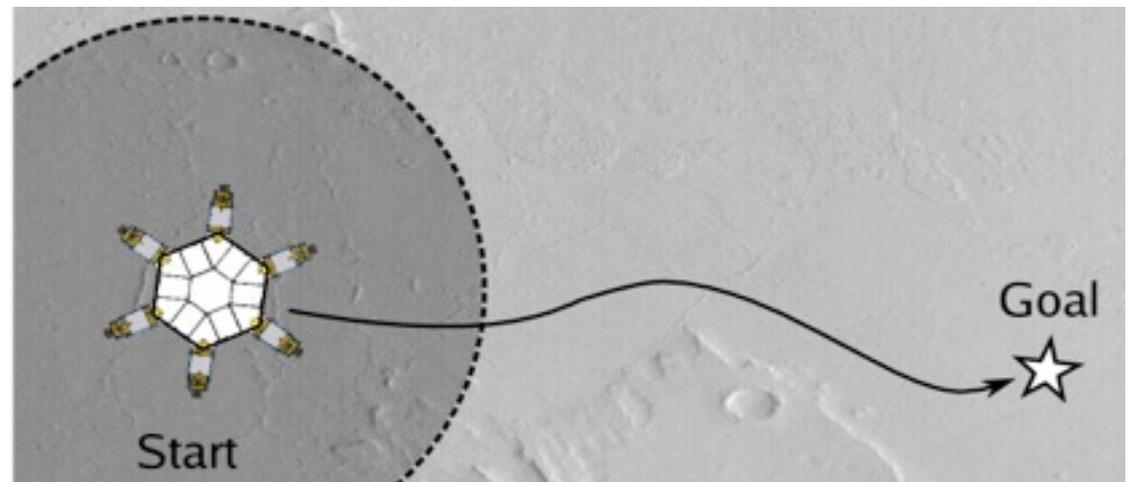


command  
center



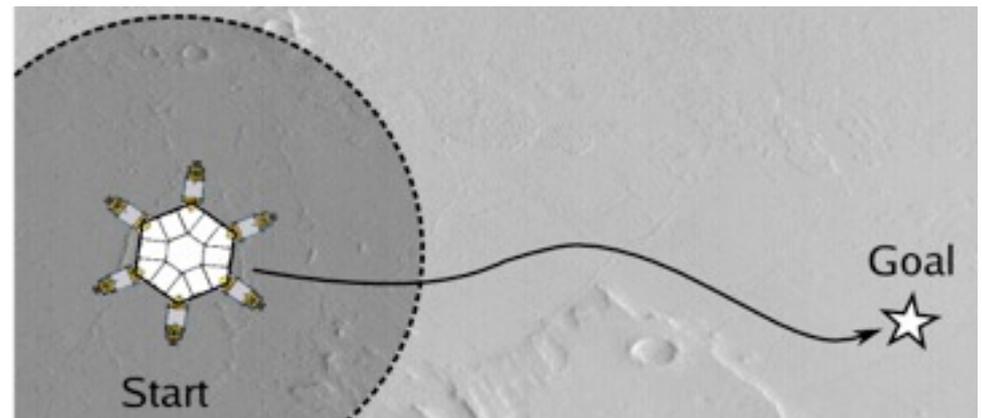
# Simple Planning Problem

- Given:
  - simple goal point
  - terrain map with varying resolution
    - detailed  $\leq 5$  meters
    - satellite  $> 5$  meters
- Find:
  - command sequence
  - prefer rolling to stepping

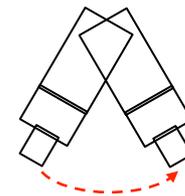
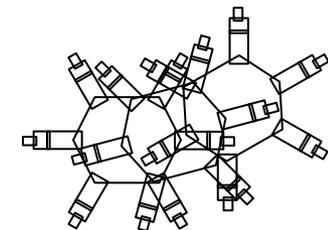
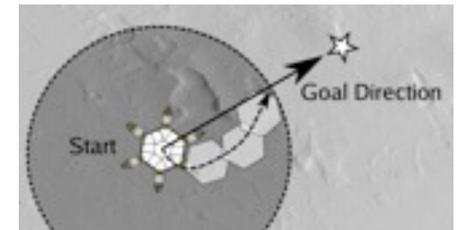
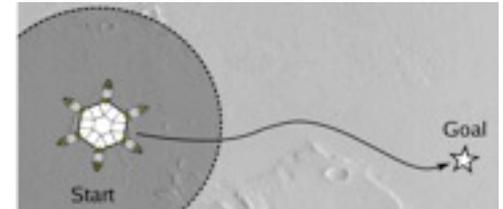
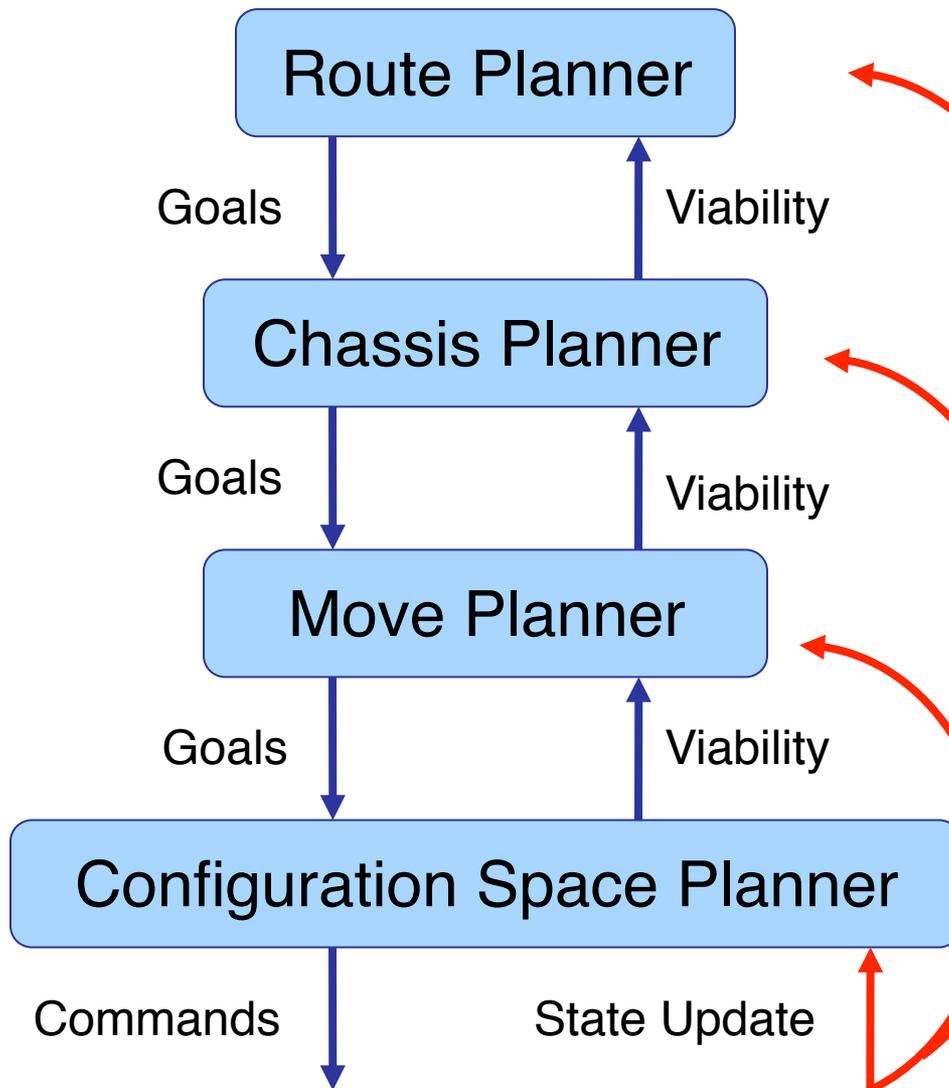


# Configuration Space Planning

- Implementation issues:
  - Computational
  - Rolling not included
  - Odd solutions
- Practical issues:
  - Data quality degrades quickly over distance
  - Uncertainty regarding future configurations
  - Active compliance (rolling)

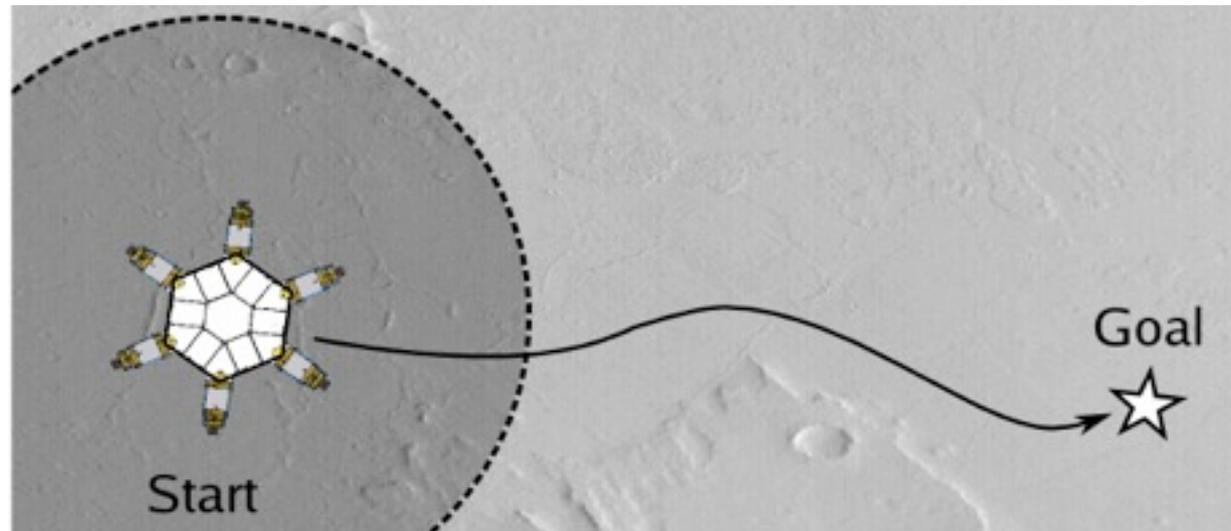


# Decomposing the problem



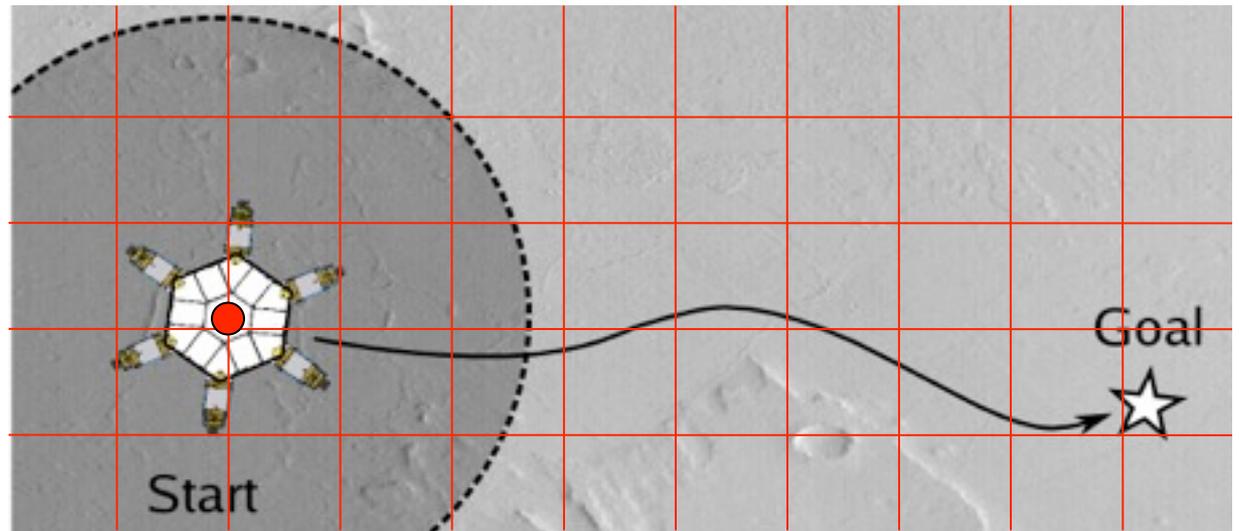
# Route Planner

- Given:
  - simple goal point
  - terrain map at varying resolution
- Find: route
- Simplifications:
  - robot is single point
  - terrain roughness as cost

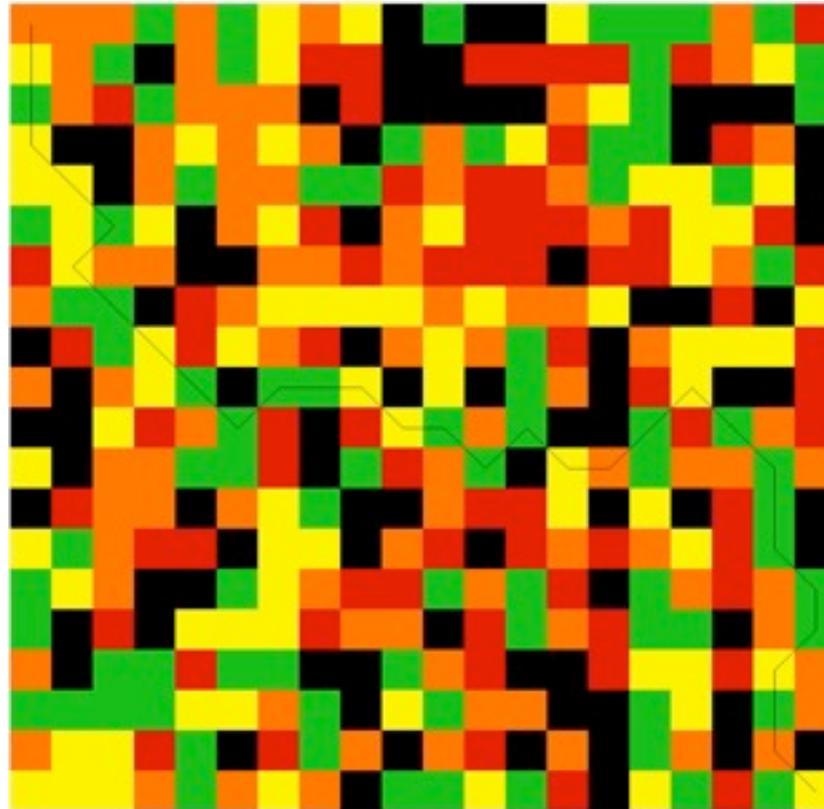


# Route Planning Approach

- Regular tessellation
- For each tile
  - steepness = max - min elevation
  - steepness < clearance
- Between tiles
  - roughness = std-deviation from mode
  - cost = roughness \* steepness

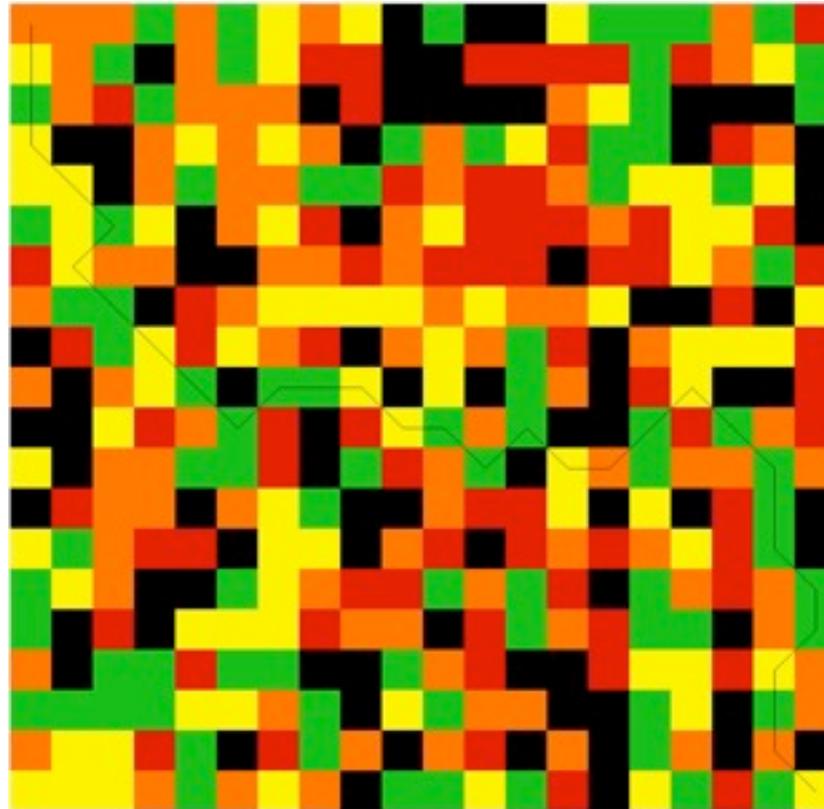


# Route Planning Search



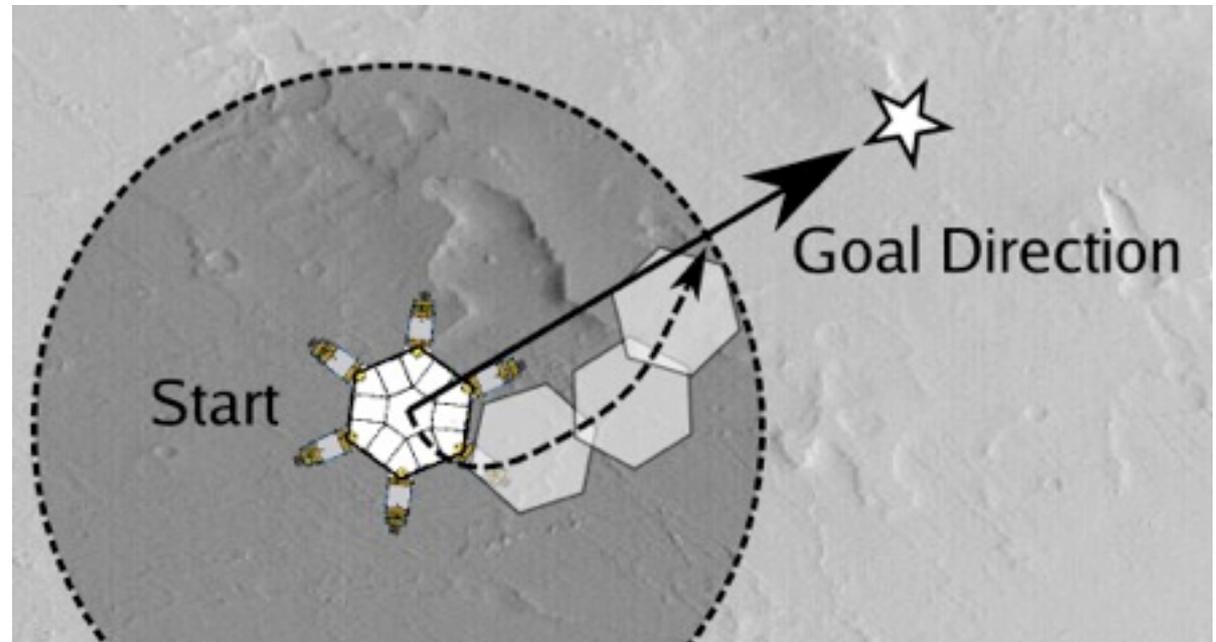
# Route Planning Search

- D\*-Lite

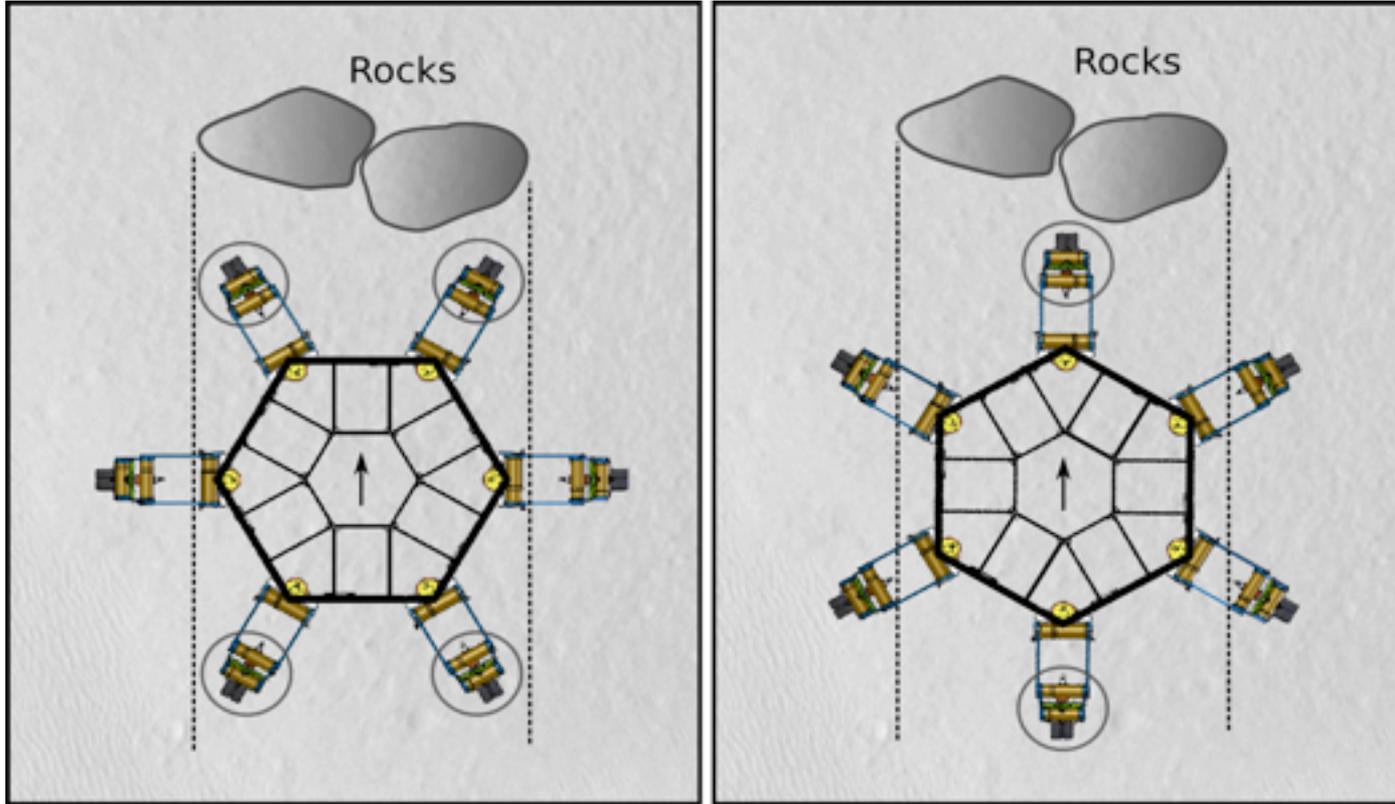


# Chassis Planner

- Given:
  - goal direction, horizon, detailed terrain map
- Find:
  - sequence of translations and rotations
  - minimize stepping
- Simplification:
  - fixed leg pose

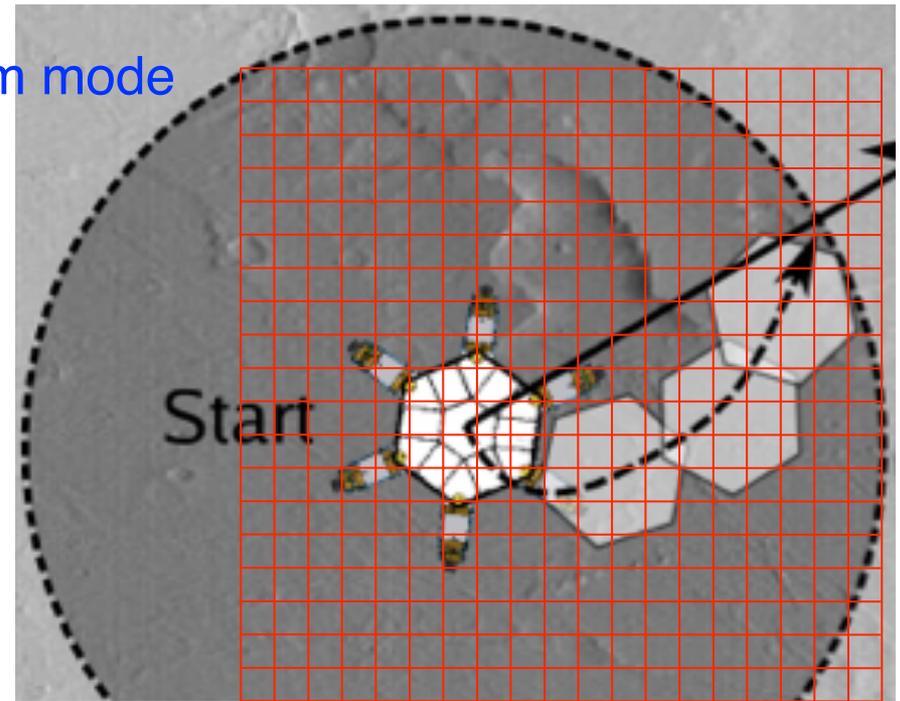


# Chassis Planner



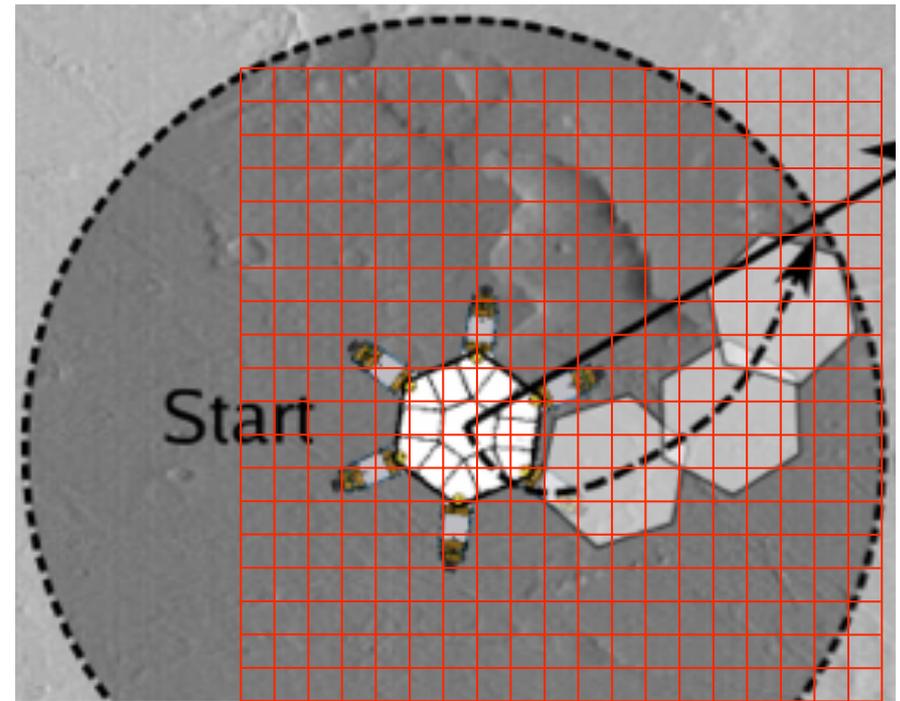
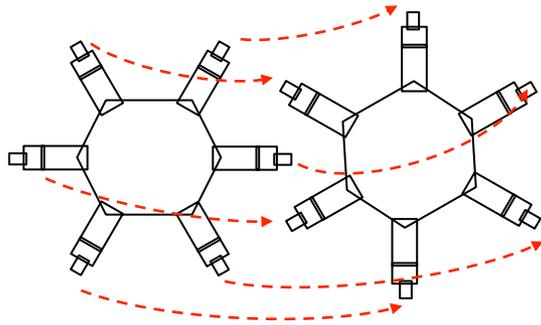
# Chassis Planner Approach

- Fine tessellation of horizon
- For each tile
  - steepness = max - min elevation
  - steepness < clearance (within entire chassis)
- Between tiles
  - roughness = std-deviation from mode
  - cost = roughness\*steepness



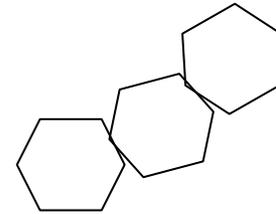
# Chassis Planner Approach

- For successive chassis positions, cost is:
  - sum over leg paths of tile transition costs
- Additional penalties when
  - adjacent legs have significant elevation change at same time
  - legs move into steep squares.



# Move Planner

- Given: fixed path for chassis
- Find: sequence of moves
  - Roll
  - Shift chassis
  - Step
- Simplification: ignore transitions



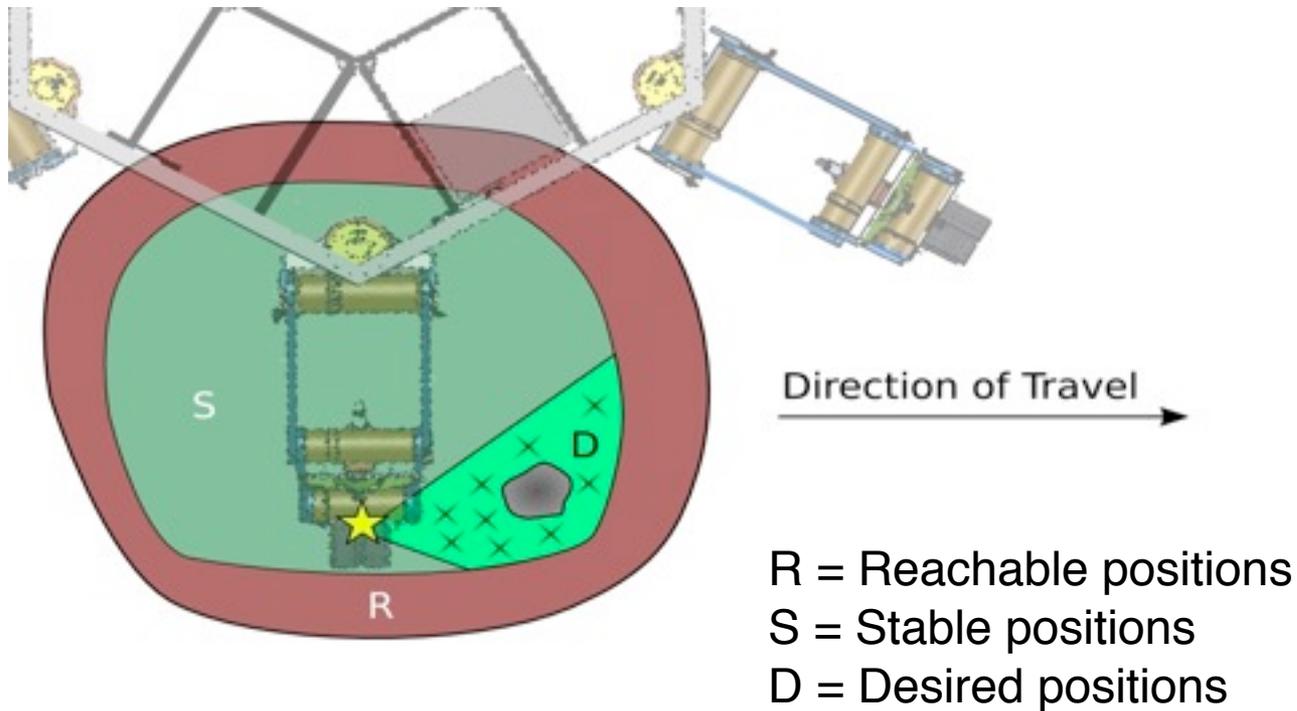
drive 090,5m  
rotate -20  
move-leg 2,pos  
raise-leg 1  
drive 070,1m  
lower-leg 1  
...

# Finding the best move

---

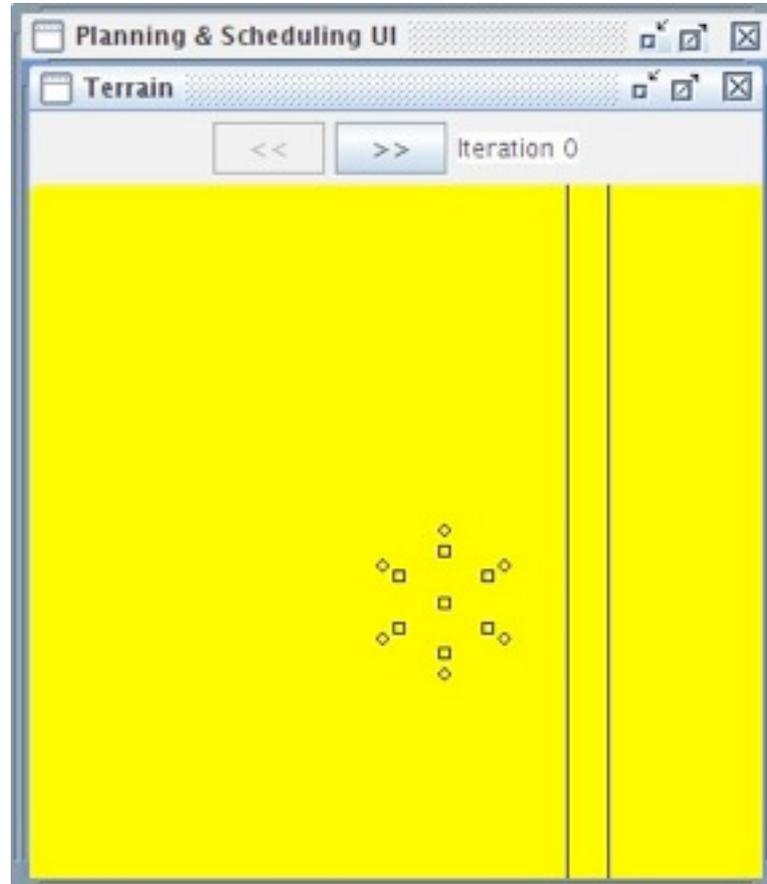
- Using depth-first search
  1. Roll if possible in the direction dictated by the chassis plan
  2. If lifting a leg will allow further rolling, prefer it
  3. For each leg and the chassis:
    - compute the max progress that the leg/chassis can be advanced in the direction of the chassis plan
    - order the leg/chassis moves according to progress along the chassis plan

# Steps Considered

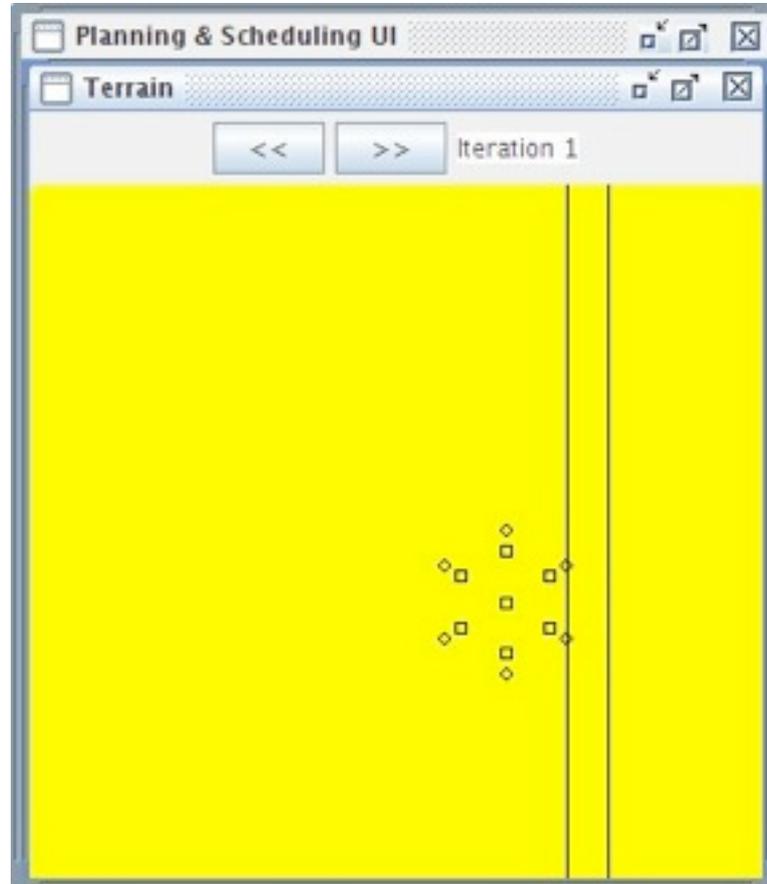


Reachable and stable regions are computed quickly by  
the Configuration Space routines

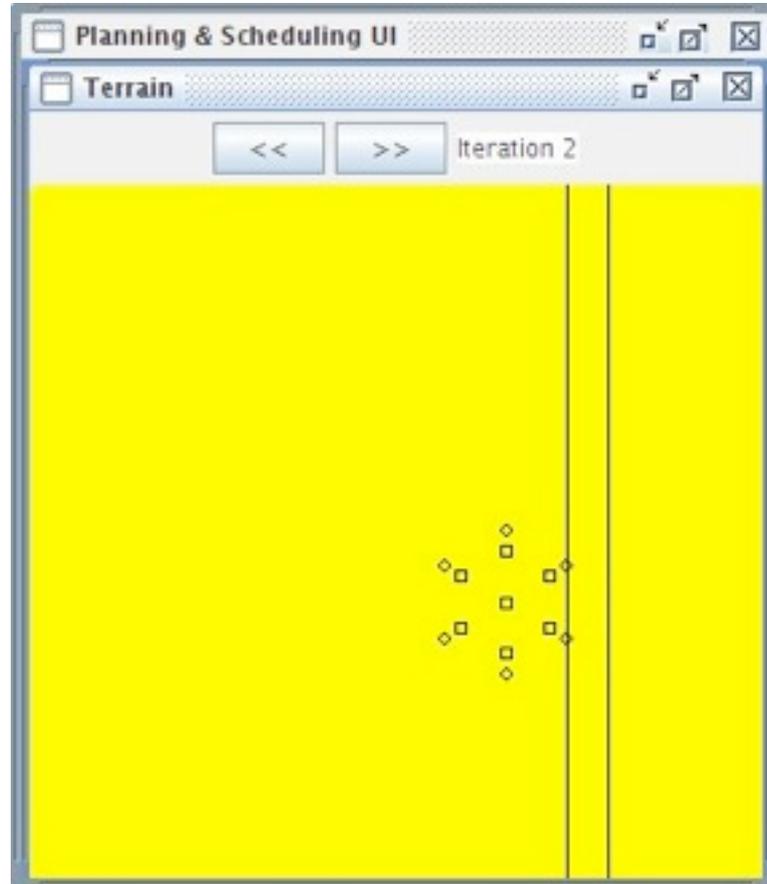
# Move Planner



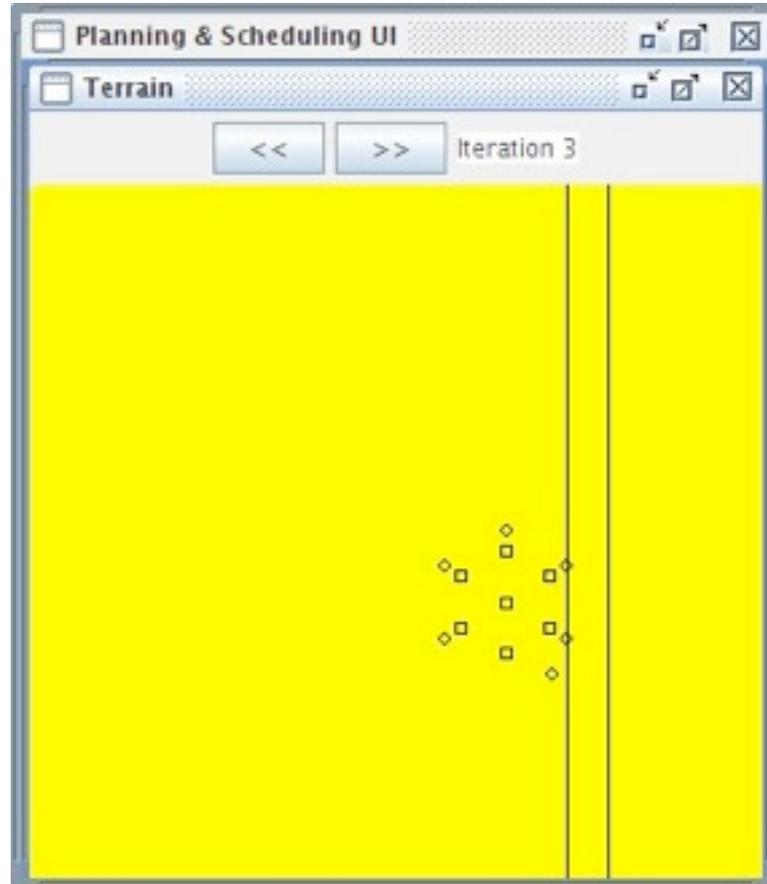
# Move Planner



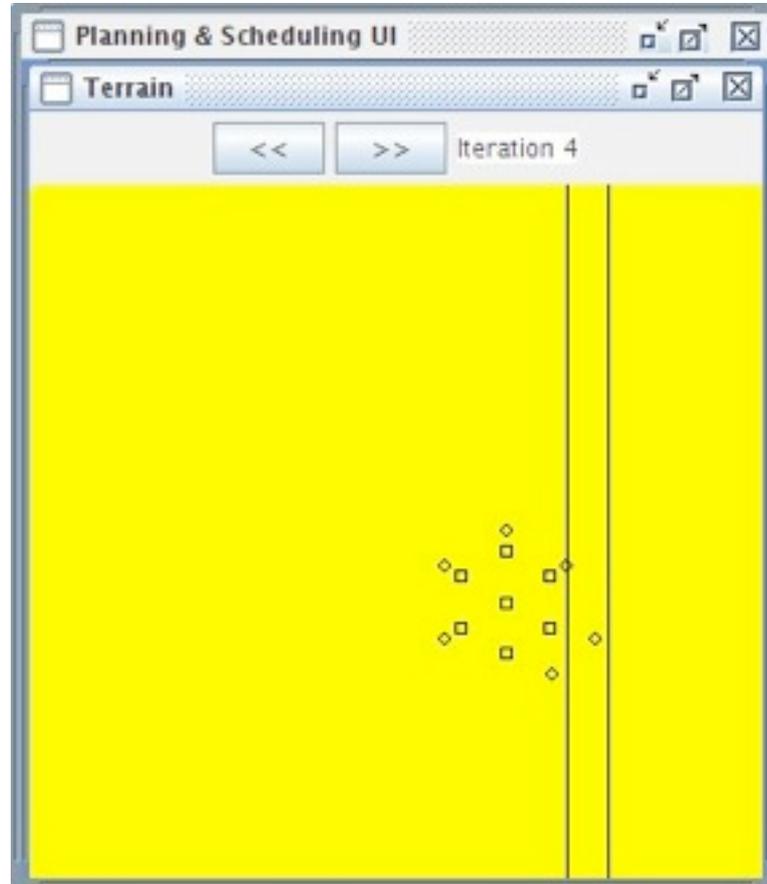
# Move Planner



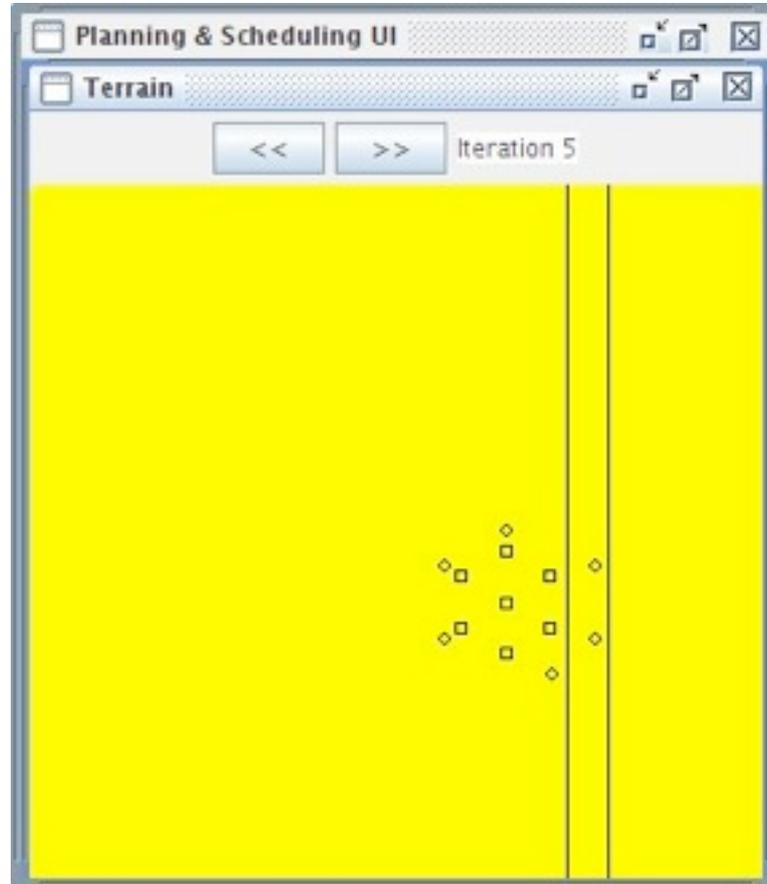
# Move Planner



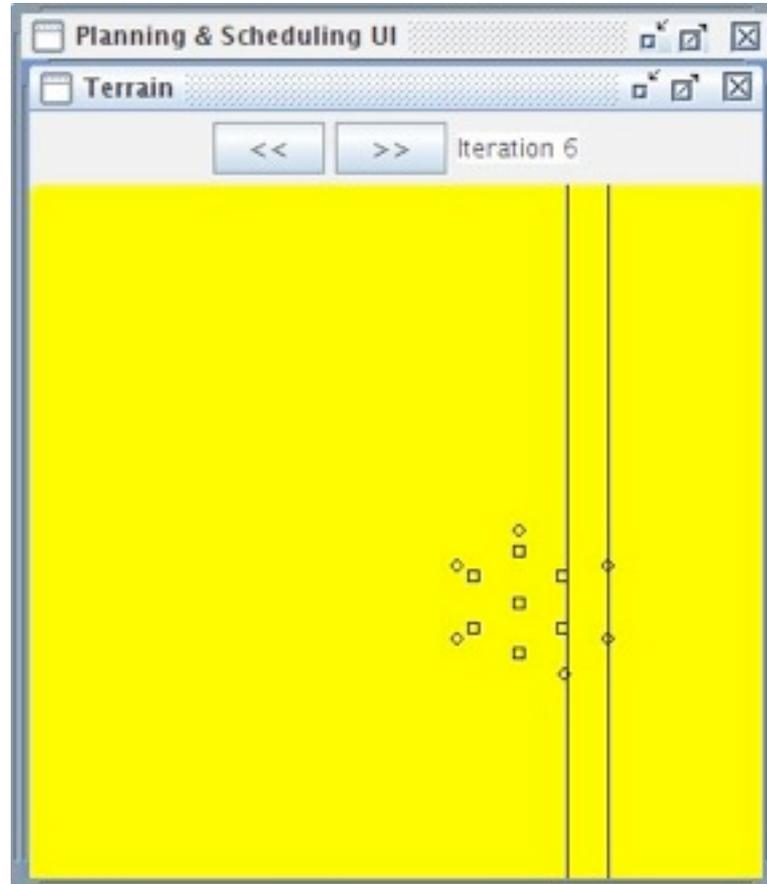
# Move Planner



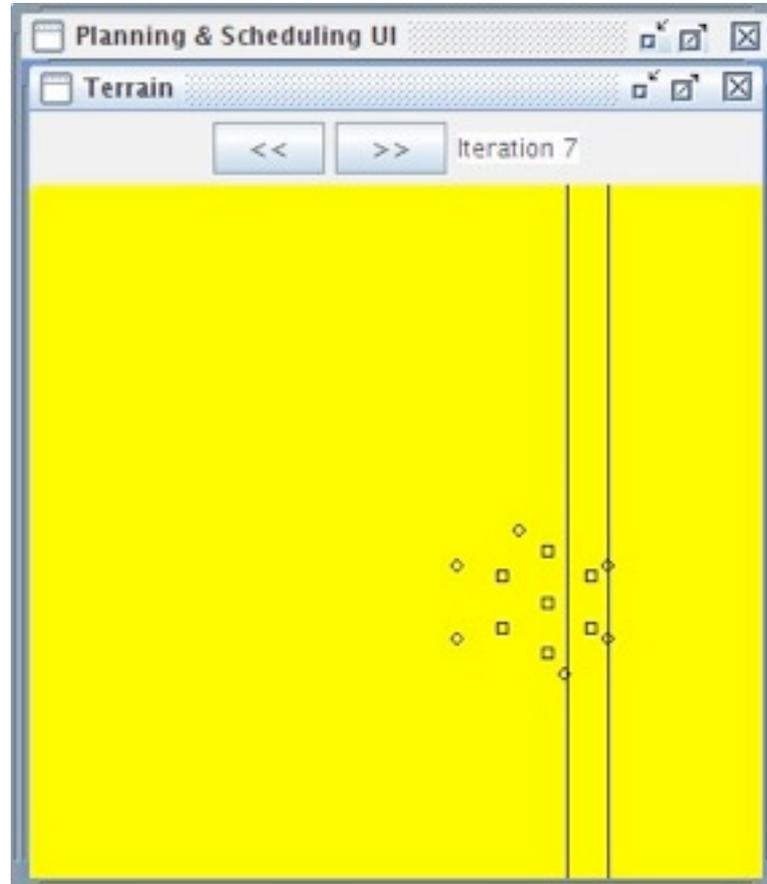
# Move Planner



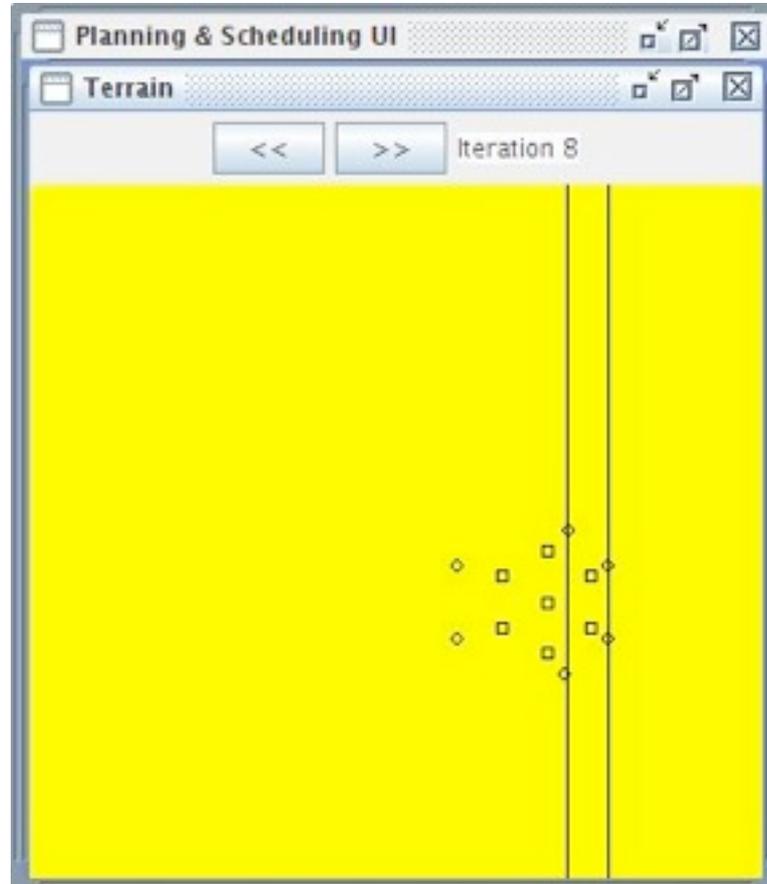
# Move Planner



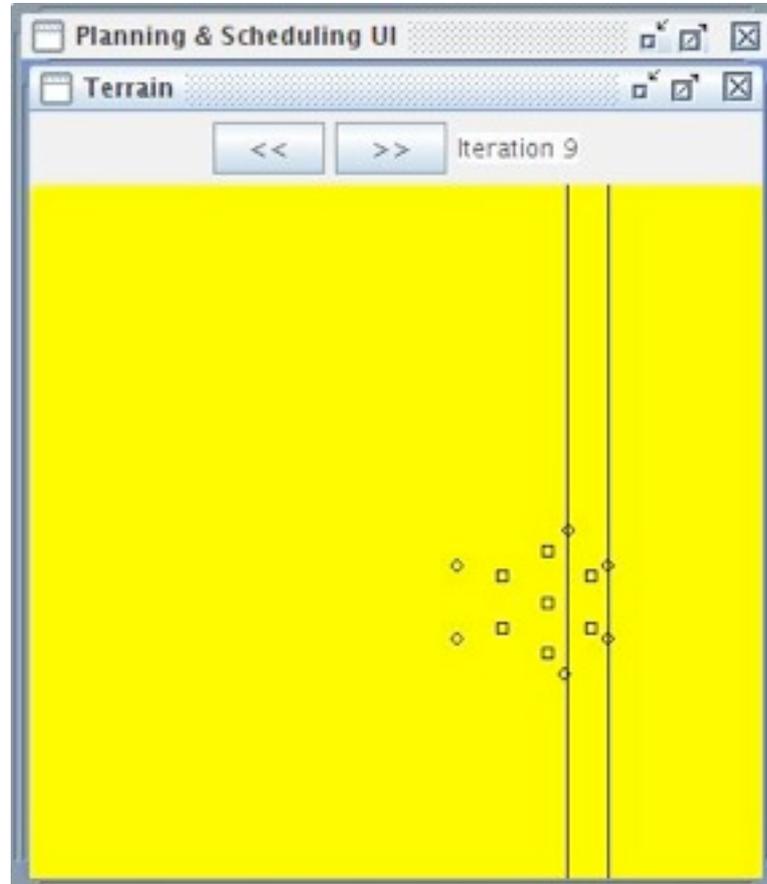
# Move Planner



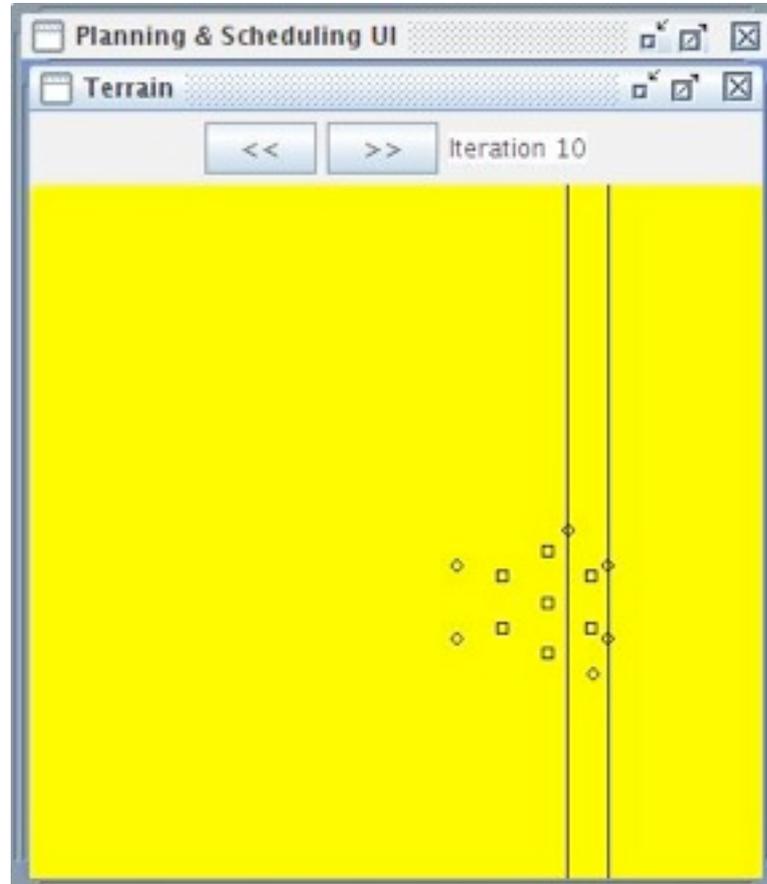
# Move Planner



# Move Planner

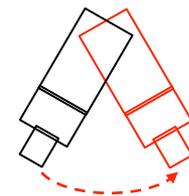
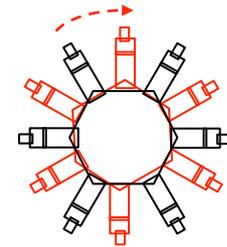
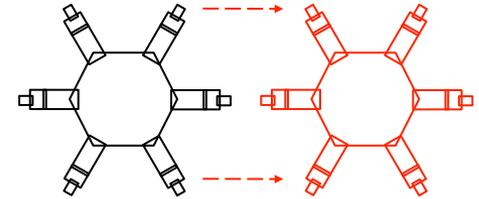


# Move Planner

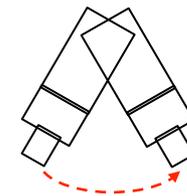
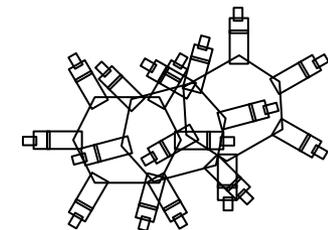
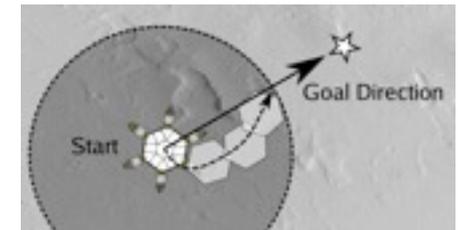
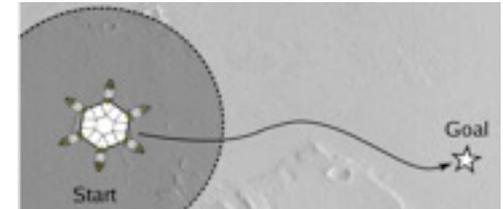
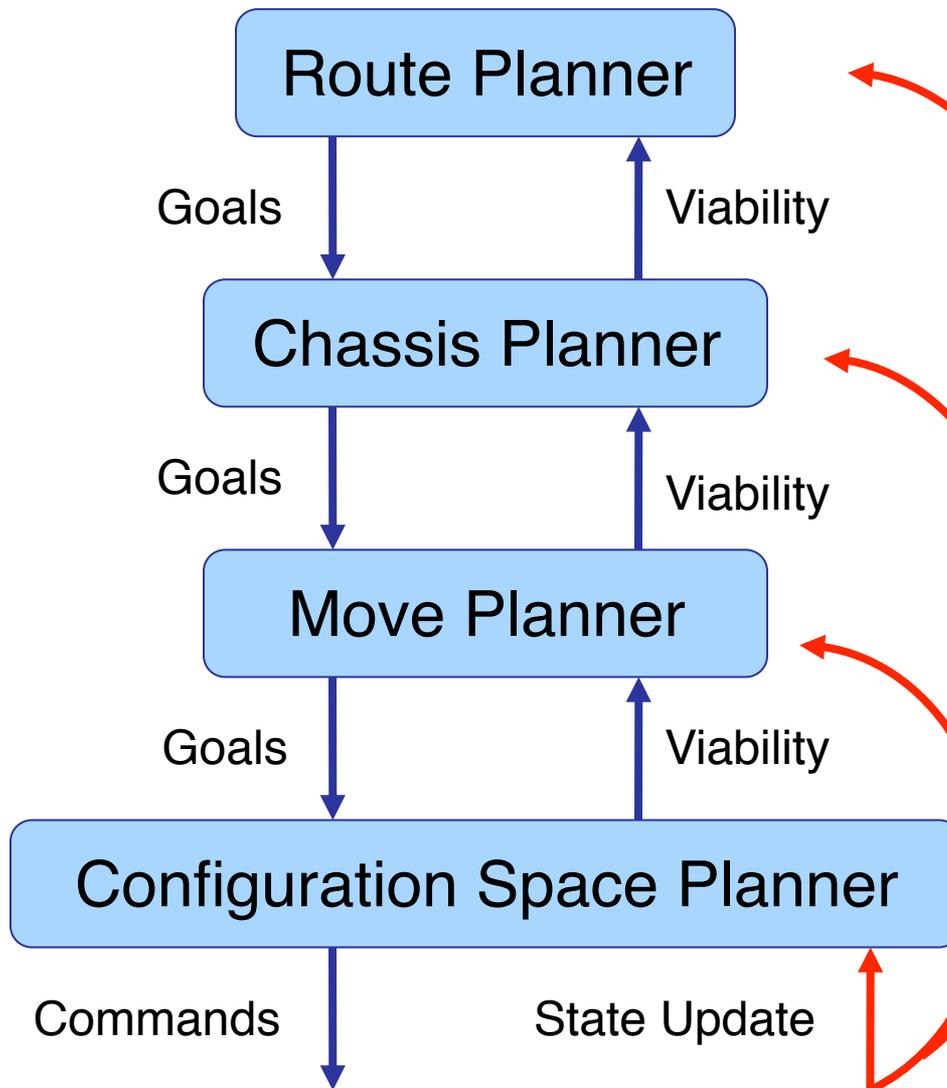


# Configuration Space Planner

- Given: specific move
- Find: path in configuration space
- Only a few steps
- Simplification: none!
  - check self-collisions
  - check environment collisions
  - check torque



# Decomposing the problem



# Example

## Path Planner

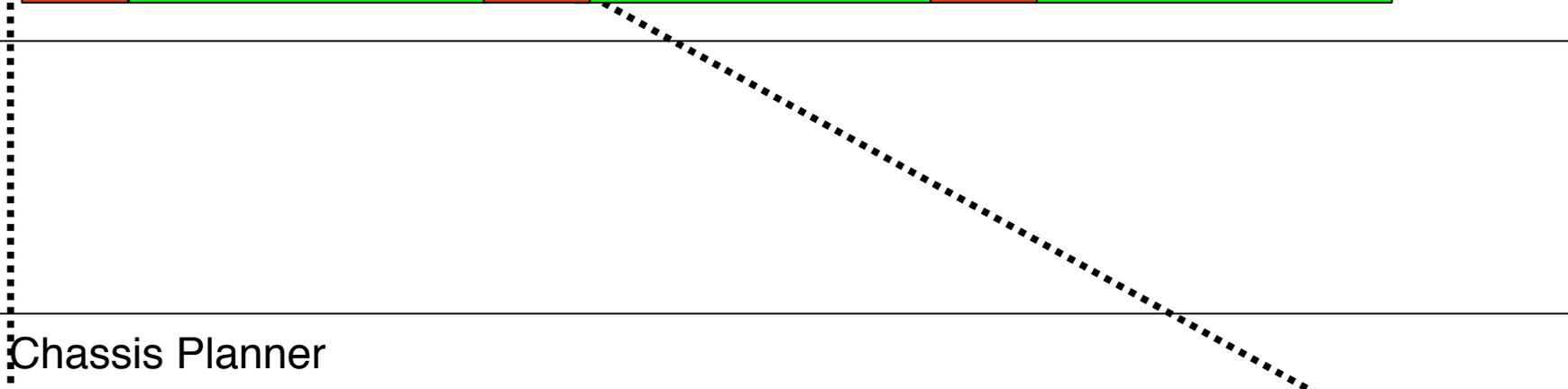


# Example

## Path Planner



## Chassis Planner

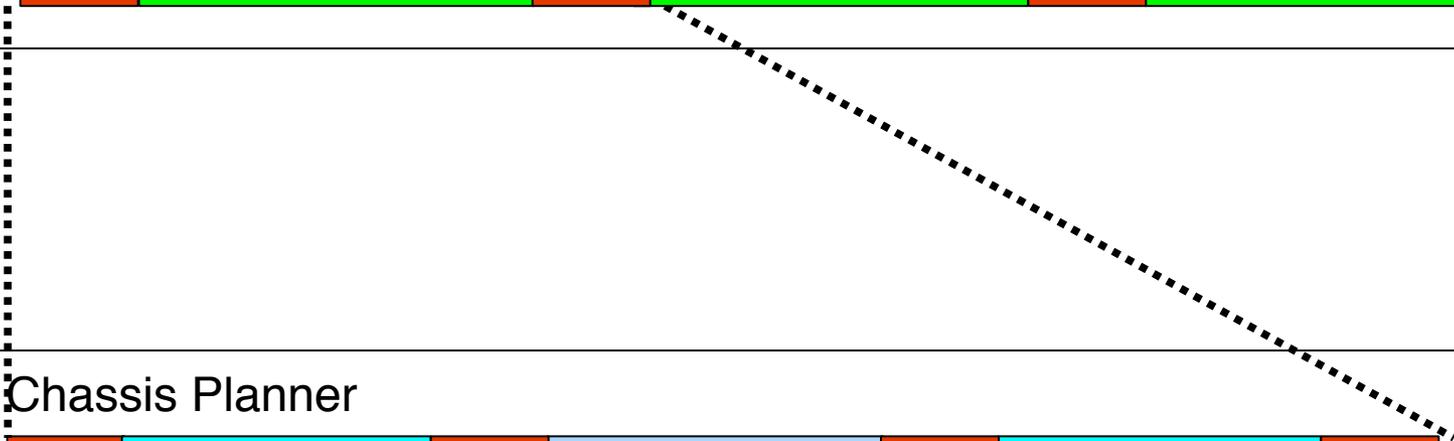


# Example

## Path Planner



## Chassis Planner



# Example

Chassis Planner



...

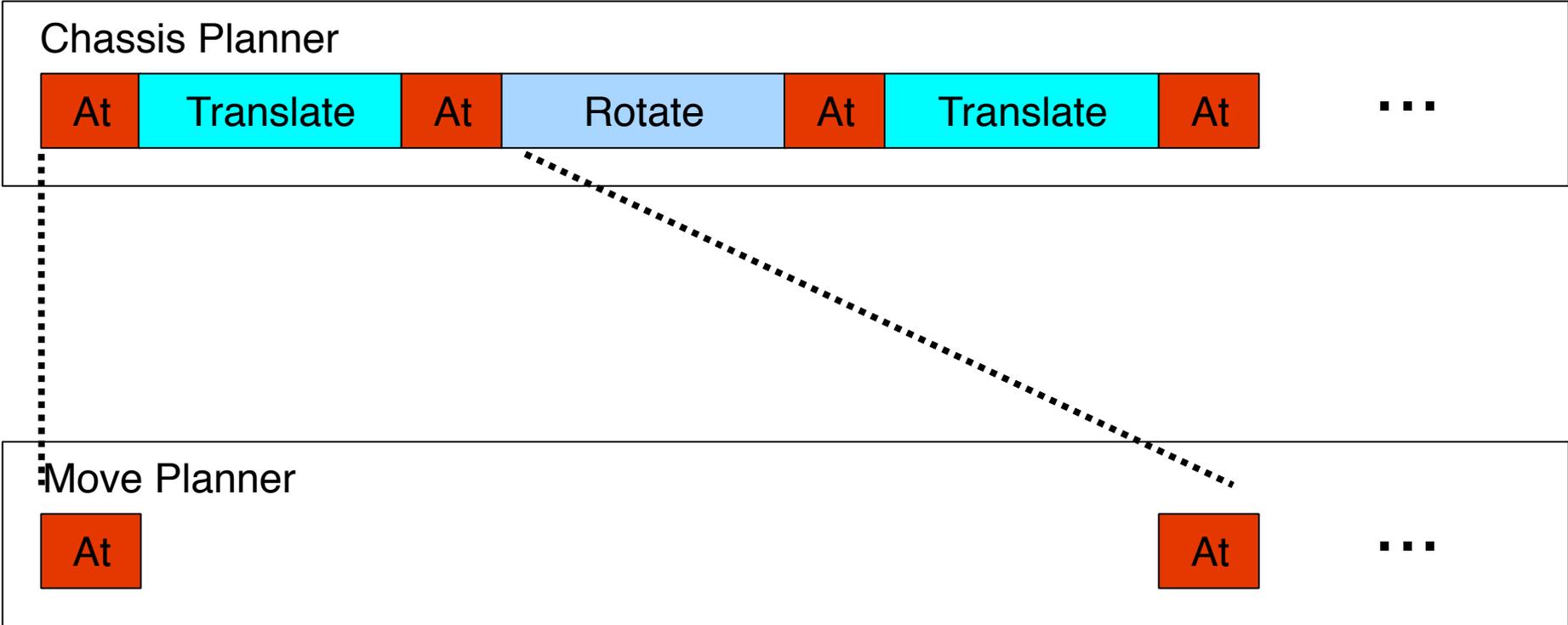
# Example

## Chassis Planner



...

# Example

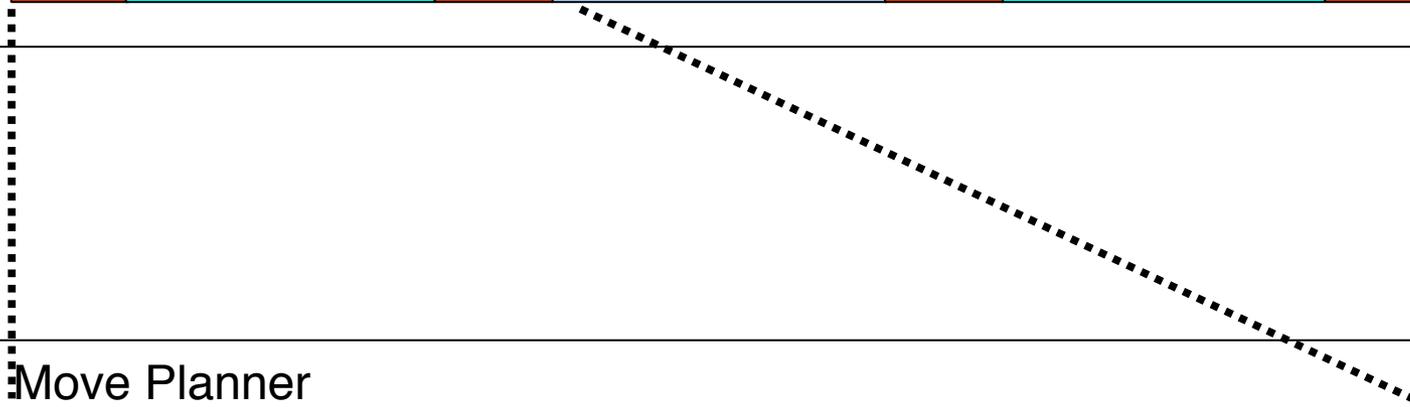
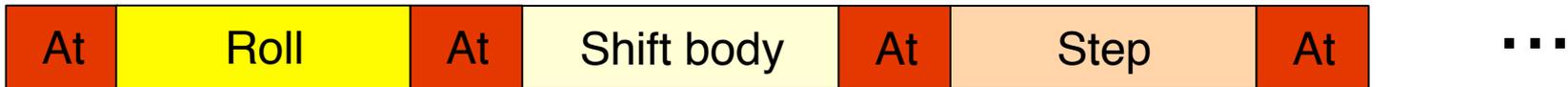


# Example

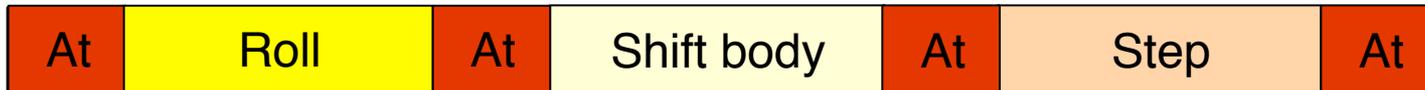
## Chassis Planner



## Move Planner



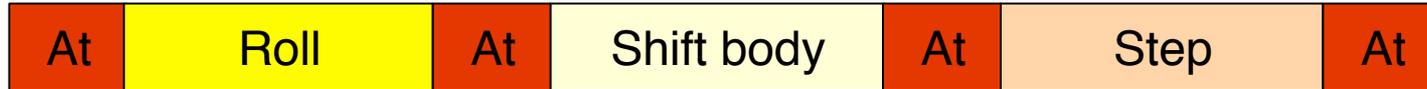
## Move Planner



...

# Example

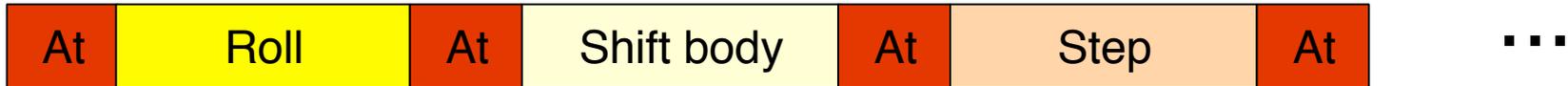
## Move Planner



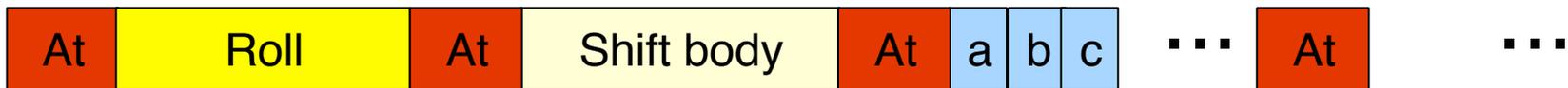
...

# Example

## Move Planner



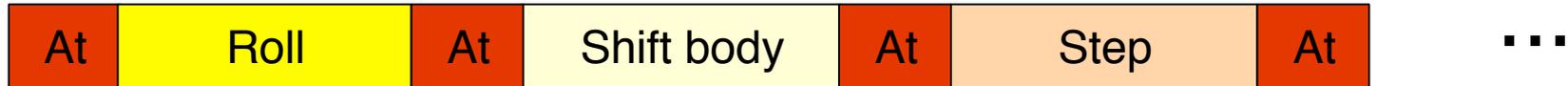
## Configuration Space Planner



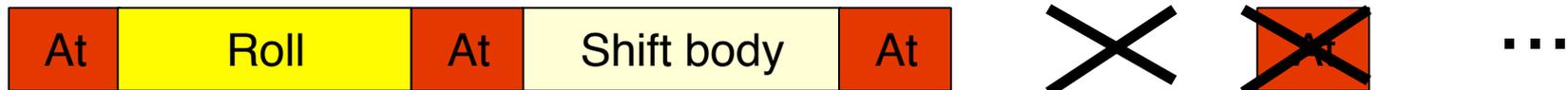
# Problem 1: Planning



## Move Planner



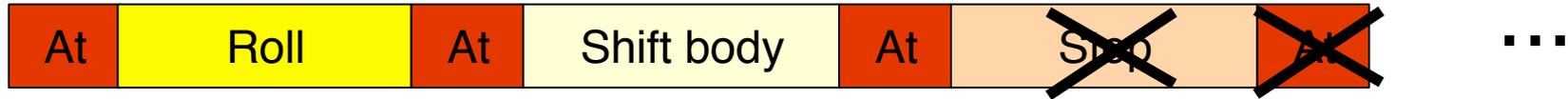
## Configuration Space Planner



# Problem 1: Planning



## Move Planner



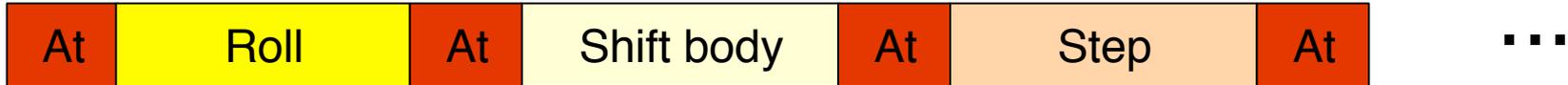
## Configuration Space Planner



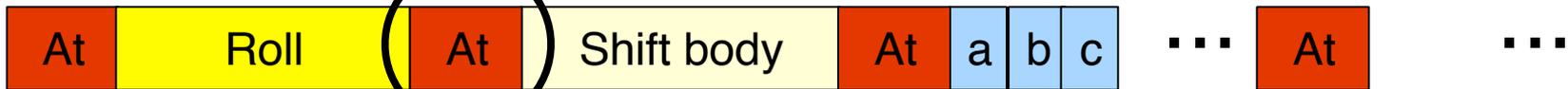
# Problem 2: Execution



## Move Planner



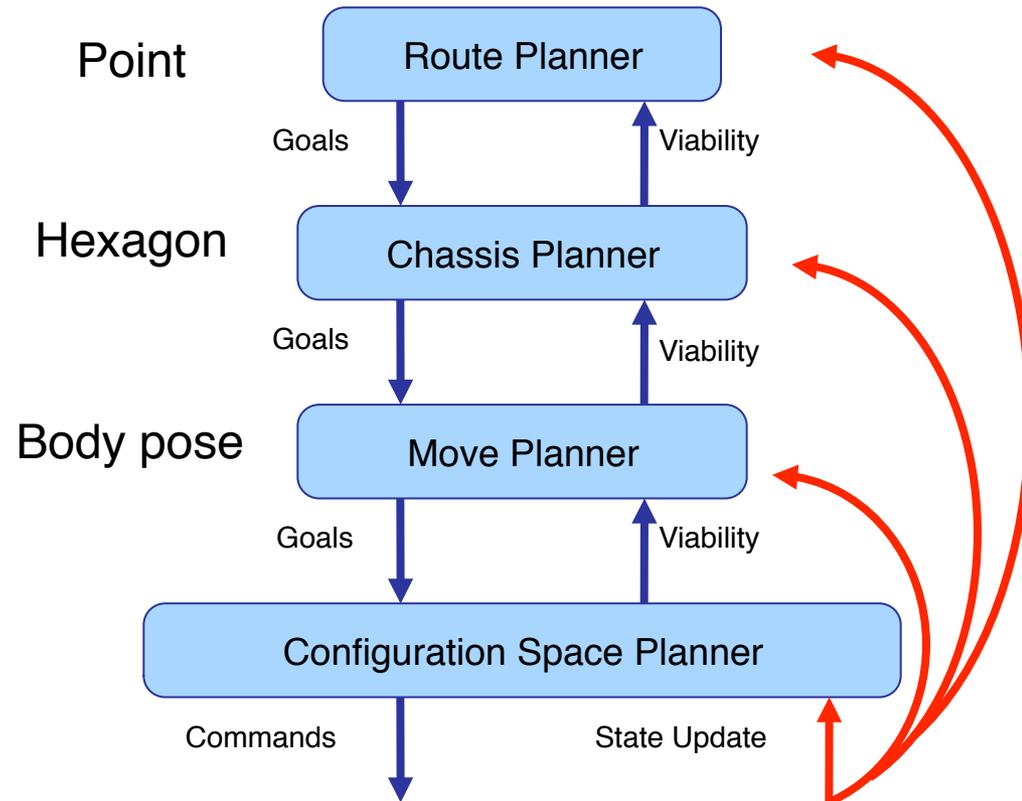
## Configuration Space Planner



End up in different place or  
configuration

# Some Questions

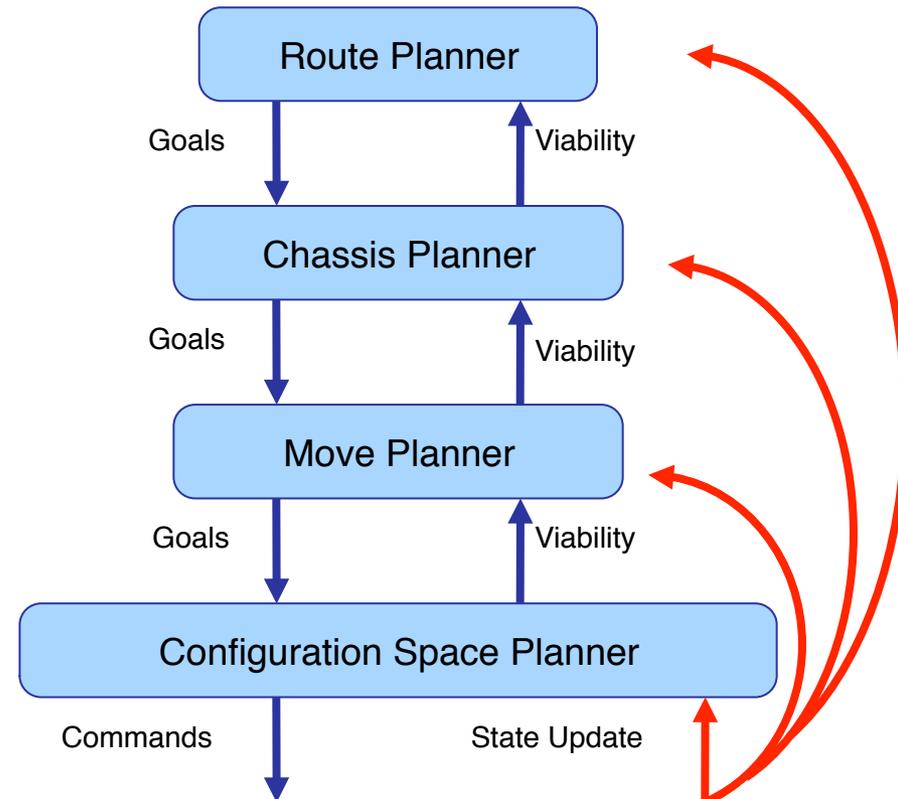
- Level breakdown
  - More than usual
  - Boundaries?



# Some Questions

- Horizon
  - Route planner
  - Chassis planner
    - visual horizon ~ 5 meters
  - Move Planner
    - 2-5 meters
  - Config space planner
    - a few moves

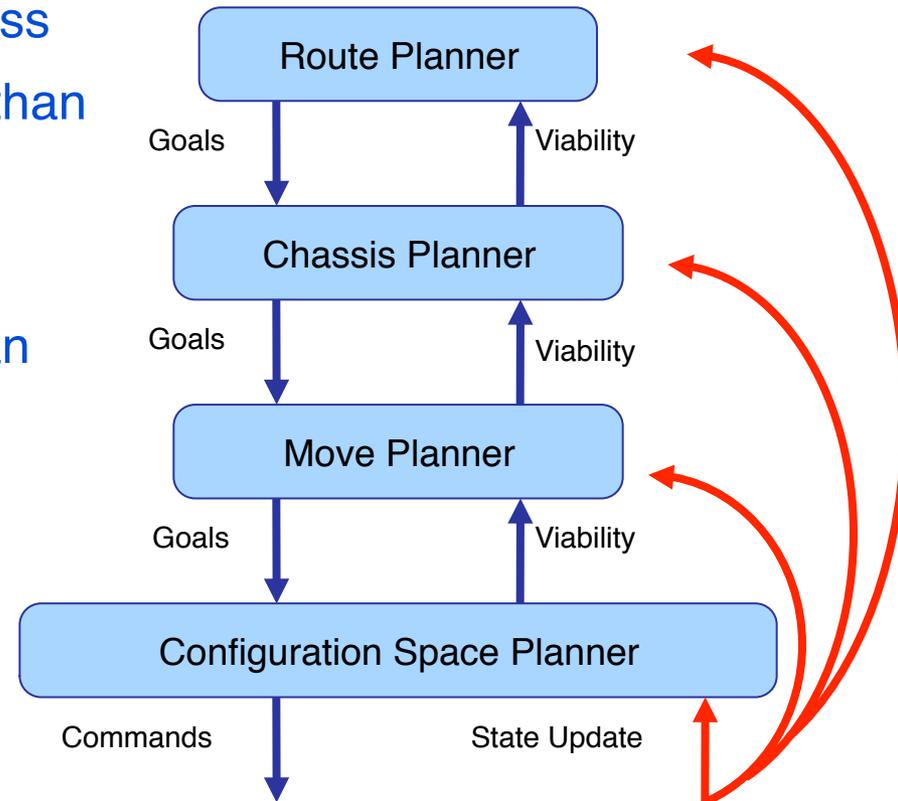
Dependent on terrain difficulty ?



# Some Questions

- How often to replan at levels
  - Route planner
    - terrain detail changes roughness
    - cost of Chassis plan is higher than predicted
  - Chassis planner
    - cost of move plan is higher than predicted
    - advancement by more than 2 meters
  - Move Planner
    - after each command

Dependent on terrain difficulty ?

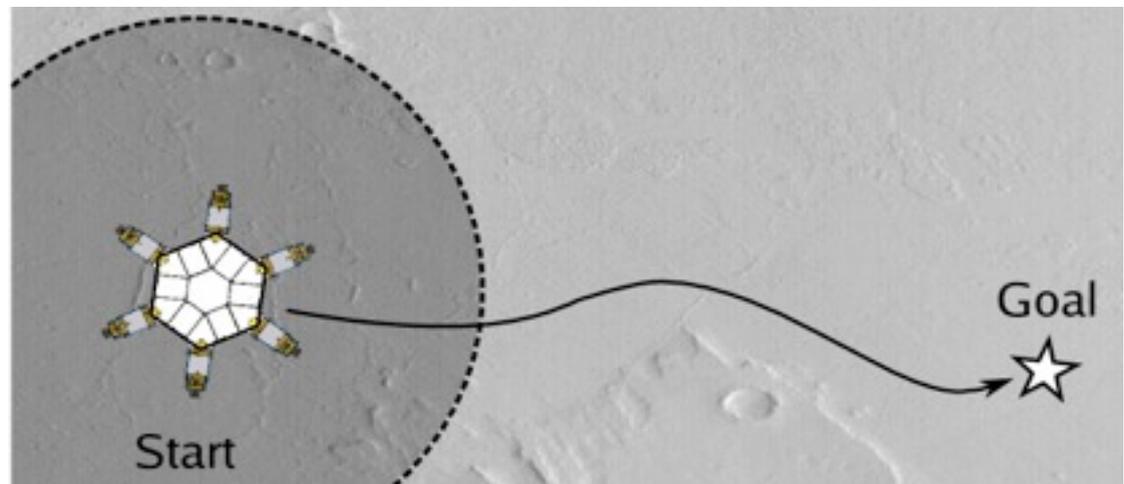


- Temporal
  - Action durations
  - Concurrency
- **Time constraints**
  - Communication windows
  - Illumination of targets
  - Temperature
- Uncertainty
  - Terrain & tracking
  - Duration of actions**
  - Energy usage
  - Storage available
- Oversubscription
  - Many conflicting goals**
  - Goal dependence



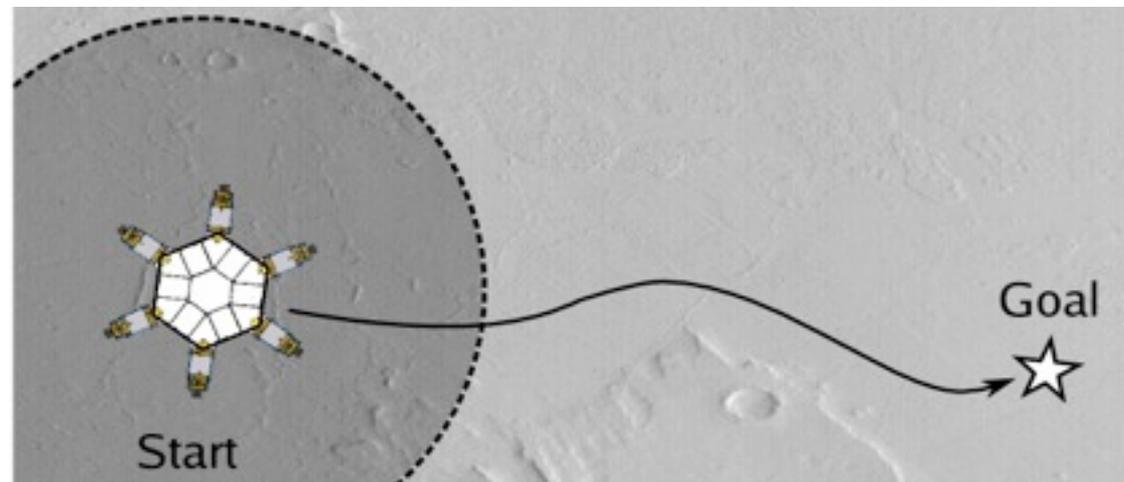
# Complicating the Planning Problem

- Given:
  - collection of goals with utilities
  - time constraints
  - uncertain durations
- Find:
  - command sequence
  - prefer rolling to stepping



# Impact

- Good news: only affects the top-level route planner:
  - need oversubscription planner
- Bad news: duration uncertainty
  - impacts time constraints
  - constantly simulate expected value
  - if timing is tight, contingency planning





- Temporal
  - Action durations
  - Concurrency
- **Time constraints**
  - Communication windows
  - Illumination of targets
  - Temperature
- Uncertainty
  - Terrain & tracking**
  - Duration of actions
  - Energy usage
  - Storage available
- Oversubscription
  - Many conflicting goals**
  - Goal dependence



# The Tracking Problem



- Navigation and localization difficult
  - beyond horizon - only gross features from satellite images
  - choose paths near trackable features

# Take Home Message

- Multiple levels of planning
  - 4 levels of path planning
  - 3T+++
- Good abstraction is key
  - allows feedback from lower level failures
  - minimizes backtracking between layers
- Task planning interacts primarily with highest layer
  - more serious with time constraints and duration uncertainty
  - breaks down w/tool usage

