# Planning Surface Cleaning Tasks by Learning Uncertain Drag Actions Outcomes

*David Martínez, Guillem Alenyà and Carme Torras*

Institut de Robòtica i Informàtica Industrial

CSIC-UPC

23rd International Conference on Automated Planning and Scheduling
Workshop in Planning and Robotics (PlanRob)

June 11, 2013

# Index

# Index

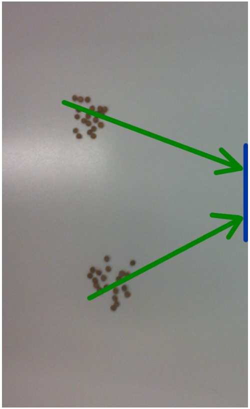
# Motivation

Robot to clean surfaces

- Moving lentils to a container
- Fast execution
  - Planning
  - Minimize cleaning actions

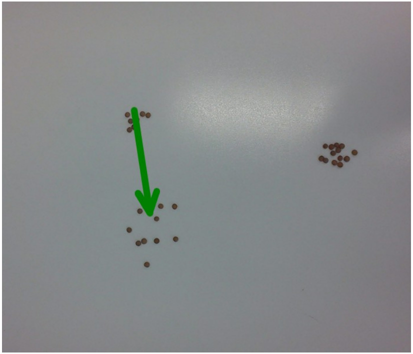
# Plan examples

# Plan examples

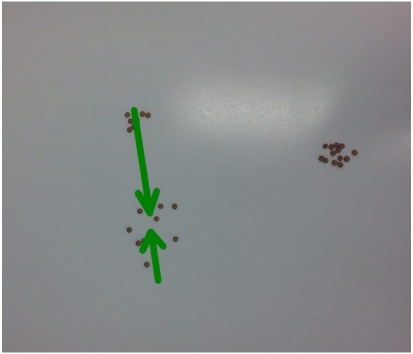


# Plan examples

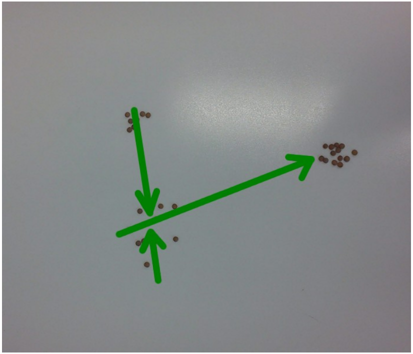
# Plan examples



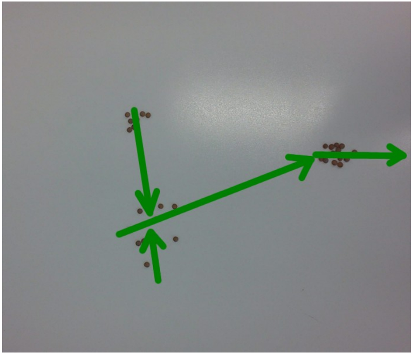
# Plan examples



# Plan examples



# Plan examples



# Motivation

Robot to clean surfaces

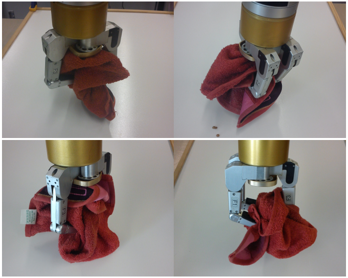
- Fast execution
  - Minimize cleaning actions
- Adapts to changes
  - Cloth grasping



# Motivation

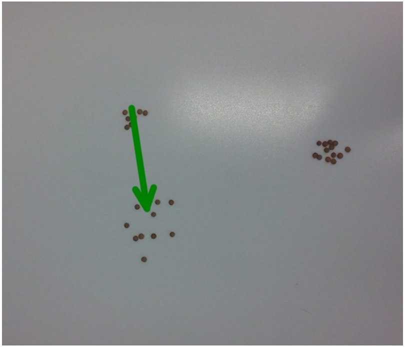
Robot to clean surfaces

- Fast execution
  - Minimize cleaning actions
- Adapts to changes
  - Cloth grasping



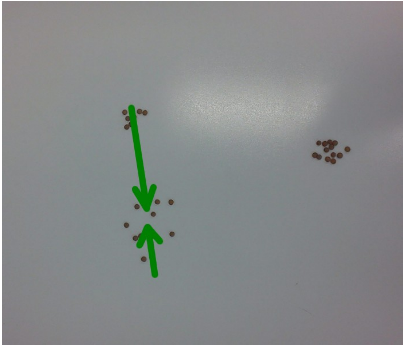
# Plan examples

Grasp 1



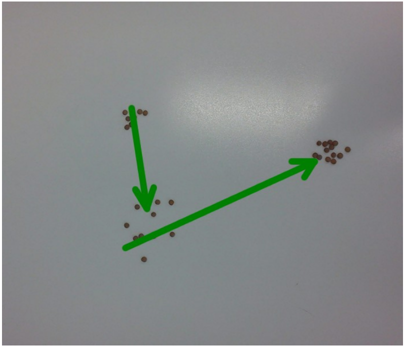
# Plan examples

Grasp 1



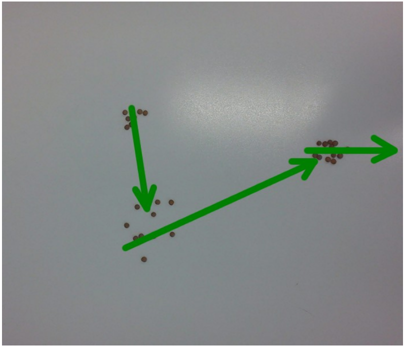
# Plan examples

Grasp 2



# Plan examples

Grasp 2



# Objectives

- Minimize execution time
  - Robot actions are expensive
  - Planning best sequences of actions
    - Rules to define actions
- Adapting to changes
  - Adapting rules to grasps
  - Learning

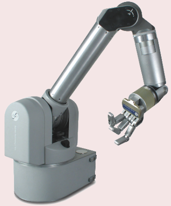
# Objectives

- Minimize execution time
  - Robot actions are expensive
  - Planning best sequences of actions
    - Rules to define actions
- Adapting to changes
  - Adapting rules to grasps
  - Learning

# Index

# Setup

## Arm manipulator: WAM



## Camera: Kinect



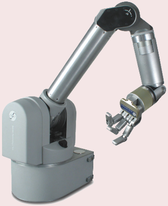
## Surface to clean



## Setup

# Setup

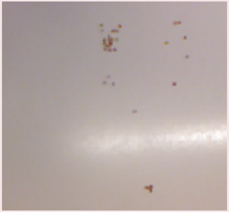
## Arm manipulator: WAM



## Camera: Kinect



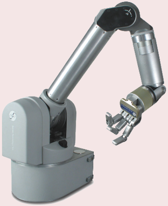
## Surface to clean



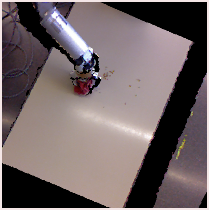
## Setup

# Setup

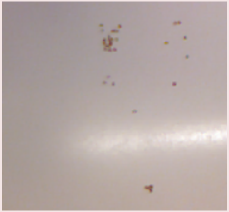
## Arm manipulator: WAM



## Camera: Kinect



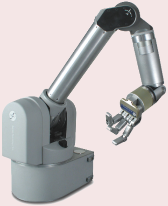
## Surface to clean



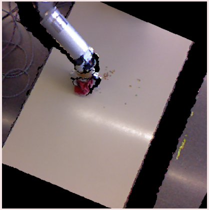
## Setup

# Setup

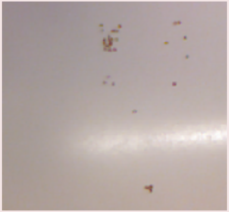
## Arm manipulator: WAM



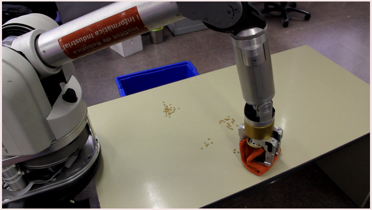
## Camera: Kinect



## Surface to clean



## Setup

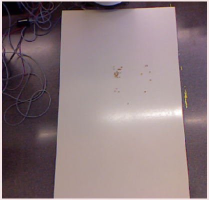


# Overview

- Observations $\rightarrow$ state
- Rules $\rightarrow$ actions
- Planning
- Learning

# Observations

## Kinect image



## Dirt segmentation



## Surface segmentation

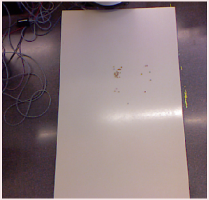


## Extract information about dirty areas

- Position
- Size
- Shape
- Scattered

# Observations

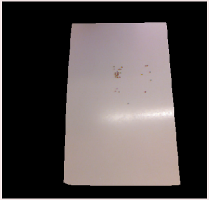
## Kinect image



## Dirt segmentation



## Surface segmentation

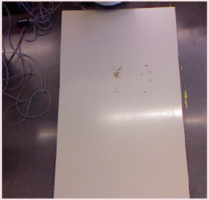


## Extract information about dirty areas

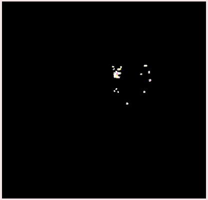
- Position
- Size
- Shape
- Scattered

# Observations

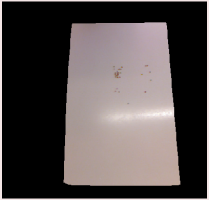
## Kinect image



## Dirt segmentation



## Surface segmentation

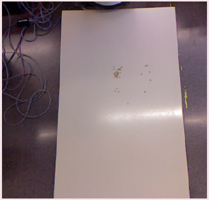


## Extract information about dirty areas

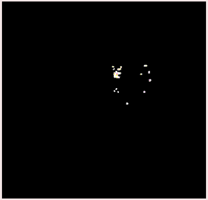
- Position
- Size
- Shape
- Scattered

# Observations

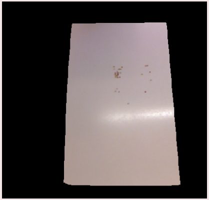
## Kinect image



## Dirt segmentation



## Surface segmentation



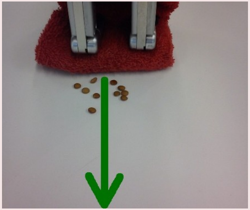
## Extract information about dirty areas

- Position
- Size
- Shape
- Scattered

# Actions

## Cleaning actions

- Straight move
- Fast move



## Grouping actions

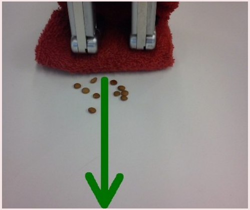
- Group scattered lentils

- Join 2 or 3 groups

# Actions

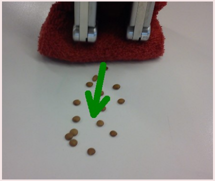
## Cleaning actions

- Straight move
- Fast move



## Grouping actions

- Group scattered lentils

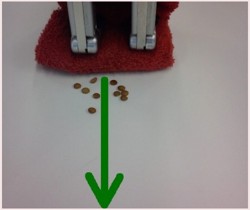


- Join 2 or 3 groups

# Actions

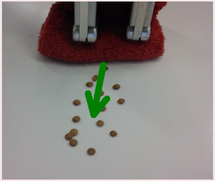
## Cleaning actions

- Straight move
- Fast move



## Grouping actions

- Group scattered lentils



- Join 2 or 3 groups

# Planning

Probabilistic planner

- Actions are stochastic
- All outcomes are important

### Example cleaning action

- Outcome 1: clean a group of lentils
- Outcome 2: clean a part of the group
- Outcome 3: scatters the group

# Planning

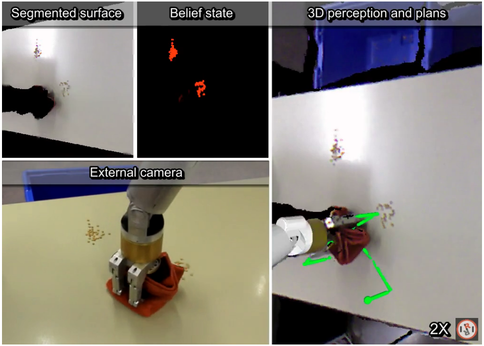
Probabilistic planner

- Actions are stochastic
- All outcomes are important

## Example cleaning action

- Outcome 1: clean a group of lentils
- Outcome 2: clean a part of the group
- Outcome 3: scatters the group

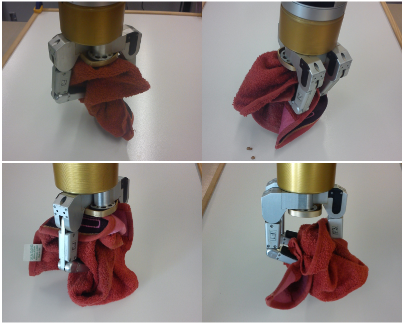
# Video




Cleaning lentils

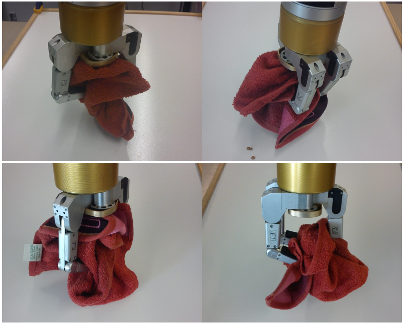
# Changing conditions

- Problem
  - Cloth grasps change rules
- Solution
  - Learn rules for new grasps
  - Good performance



# Changing conditions

- Problem
  - Cloth grasps change rules
- Solution
  - Learn rules for new grasps
  - Good performance



# Index

# Learning requirements

- Robot actions are slow
    - Learn with few actions
- Observability
    1. Partial observability requires more experience
    2. Accurate observations
        - Problem with occlusions

# Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions

# Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions

# Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions

# Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. ~~Partial observability requires more experience~~
  2. Accurate observations
     - Problem with occlusions

# Learning

Learning requirements

- Learning a model
- State has many objects
  - Symbolic domain
- Robot actions are stochastic
  - Action uncertainty

Different approaches

- Model-based RL
- Object-oriented RL
  - Diuk et al, ICML 08
- RL in Relational world
  - Lang et al, JMLR 13

# Learning

Learning requirements

- Learning a model
- State has many objects
  - Symbolic domain
- Robot actions are stochastic
  - Action uncertainty

Different approaches

- Model-based RL
- Object-oriented RL
  - Diuk et al, ICML 08
- RL in Relational world
  - Lang et al, JMLR 13

# Learning

Learning requirements

- Learning a model
- State has many objects
    - Symbolic domain
- Robot actions are stochastic
    - Action uncertainty

Different approaches

- Model-based RL
- Object-oriented RL
    - Diuk et al, ICML 08
- RL in Relational world
    - Lang et al, JMLR 13

# Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps

# Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps

# Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps

# Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps

# Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps

# Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

```
Action: Fast clean
Preconditions:
  dirt(X)
Outcomes:
  1.0 -dirt(X)
  0.0 Nothing
```

# Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

```
Action: Fast clean
Preconditions:
  dirt(X)
Outcomes:
  1.0 -dirt(X)
  0.0 Nothing
```

# Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

```
Action: Fast clean
Preconditions:
  dirt(X)
Outcomes :
  1.0 -dirt(X)
  0.0 Nothing
```

# Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

```
Action: Fast clean
Preconditions:
  dirt(X)
Outcomes:
  1.0 -dirt(X)
  0.0 Nothing
```

# Initial rules refinement

- Requirements
  - Few experiences available
  - Rule refinement after every execution
  - Fast

**Decreasing-m-estimate**

- Learning heuristic to update probabilities
- Based on m-estimate
- Very fast

# Initial rules refinement

- Requirements
  - Few experiences available
  - Rule refinement after every execution
  - Fast

## Decreasing-m-estimate

- Learning heuristic to update probabilities
- Based on m-estimate
- Very fast

## m-estimate

$$P = \frac{p + mP_0}{p + n + m}. \qquad (1)$$

- Parameters:
  - $m$ - Learning parameter
  - $P$ - Probability
  - $P_0$ - Probability
  - $p$ - Positive examples
  - $n$ - Negative examples
- Initial probability has much influence

# m-estimate

## m-estimate

$$P = \frac{p + mP_0}{p + n + m}. \tag{1}$$

- Parameters:
  - $m$ - Learning parameter
  - $P$ - Probability
  - $P_0$ - Probability
  - $p$ - Positive examples
  - $n$ - Negative examples
- Initial probability has much influence

# Decreasing m-estimate

## Decreasing m-estimate

$$P = \frac{p + (m/\sqrt{p+n})P_0}{p + n + (m/\sqrt{p+n})}. \qquad (2)$$

- $m$ decreases as experience is obtained
- Parameters:
  - $m$ - Learning parameter
  - $P$ - Probability
  - $P_0$ - Probability
  - $p$ - Positive examples
  - $n$ - Negative examples

# Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
- Using
  - Optimistic initial rules
  - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system

# Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
- Using
  - Optimistic initial rules
  - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system

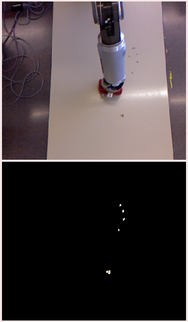
# Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
- Using
  - Optimistic initial rules
  - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system

# Learning overhead

- Learning requires
  - Accurate perception
  - No occlussions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound

**Occlusion**

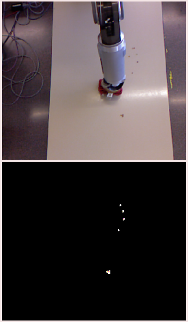


**Moved arm**

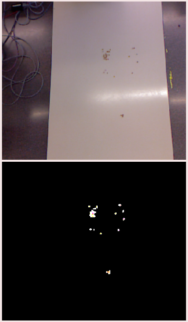
# Learning overhead

- Learning requires
  - Accurate perception
  - No occlussions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound

**Occlusion**



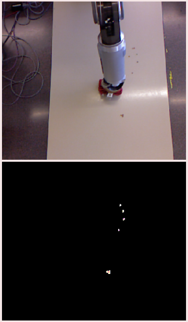
**Moved arm**



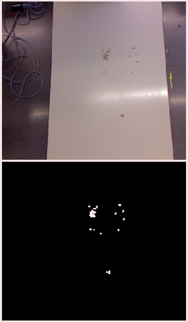
# Learning overhead

- Learning requires
  - Accurate perception
  - No occlussions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound

**Occlusion**



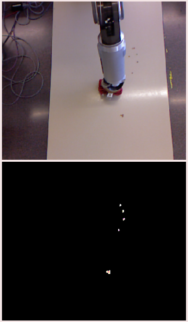
**Moved arm**



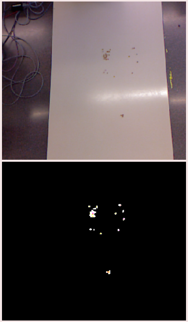
# Learning overhead

- Learning requires
  - Accurate perception
  - No occlussions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound

**Occlusion**

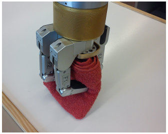


**Moved arm**

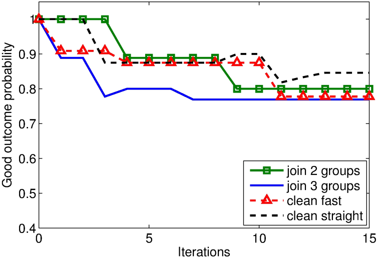


# Index

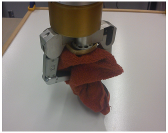
# Pre-trained rules



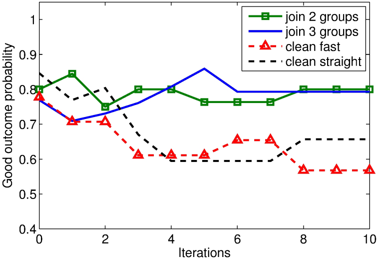
Generating pre-trained rules

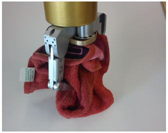


# Rule adaptation

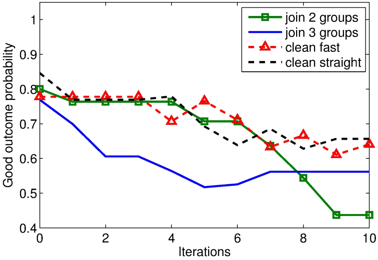


Refining rules with grasp 1



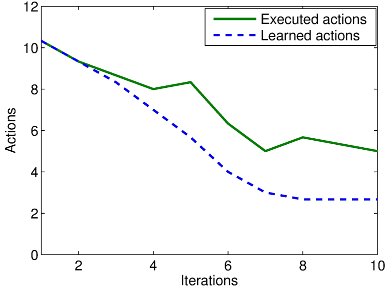



Refining rules with grasp 2

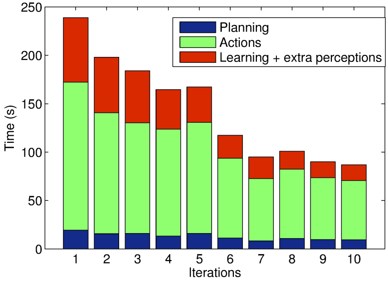


# Performance tests

## Actions executed and learned

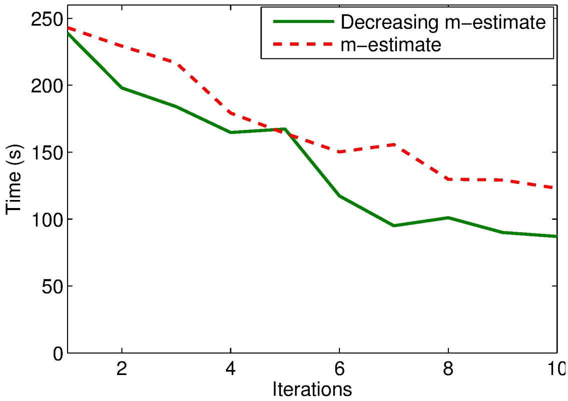



## Execution time



# Performance tests (decreasing m-estimate)

Comparing m-estimate vs decreasing m-estimate



# Index

# Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning

# Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning

# Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning

# Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability

# Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability

# Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability

# Thanks!

**Questions?**

# Planning Surface Cleaning Tasks by Learning Uncertain Drag Actions Outcomes

*David Martínez, Guillem Alenyà and Carme Torras*

Institut de Robòtica i Informàtica Industrial

CSIC-UPC

23rd International Conference on Automated Planning and Scheduling
Workshop in Planning and Robotics (PlanRob)

June 11, 2013