Action Learning in IRFO

Leon Bodenhagen

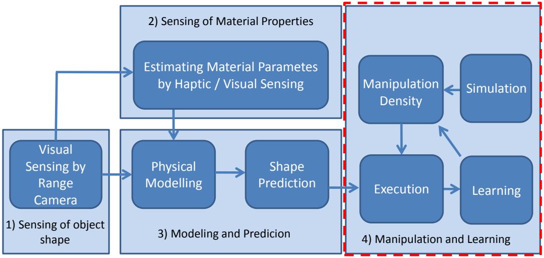
Maersk McKinney Moller Institute
University of Southern Denmark

25. November 2011

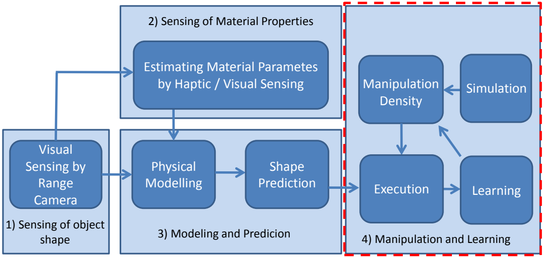
INTERREG4A
SYDDANMARK-SCHLESWIG-K.E.R.N.

European Union
European Regional
Development Fund
Investing in your future

Outline


2) Sensing of Material Properties
Estimating Material Parametes by Haptic / Visual Sensing
Visual Sensing by Range Camera
1) Sensing of object shape
Physical Modelling
Shape Prediction
3) Modeling and Predicion
Manipulation Density
Simulation
Execution
Learning
4) Manipulation and Learning

Outline



Outline:

1. Introduction / background
2. Grasping
3. Peg In Hole operations
4. Laying down operations

Introduction

Grasp learning: association of actions to an object:

- Object relative gripper poses

Introduction

Grasp learning: association of actions to an object:

- Object relative gripper poses

Modelling all successful grasps:

- Grasp Density [1]

[1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater.
Learning grasp affordance densities.
Paladyn Journal of Behavioral Robotics, 2:1–17, 2011.

Introduction

Grasp learning: association of actions to an object:

- Object relative gripper poses

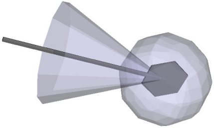
Modelling all successful grasps:

- Grasp Density [1]

Full 6D Representation:

$$d(x) = \sum_{i=0}^{n} w_i K_{\mu_i,\sigma}(x) \text{ for } x \in SE(3)$$

General, but high-dimensional



[1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater. Learning grasp affordance densities. *Paladyn Journal of Behavioral Robotics*, 2:1–17, 2011.

Introduction

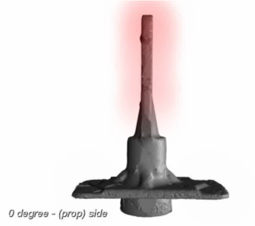
Grasp learning: association of actions to an object:

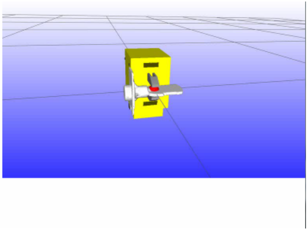
- Object relative gripper poses

Modelling all successful grasps:

- Grasp Density [1]

Example of a learned density:





[1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater.
Learning grasp affordance densities.
Paladyn Journal of Behavioral Robotics, 2:1–17, 2011.

Introduction

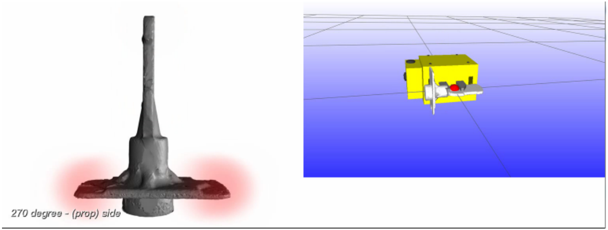
Grasp learning: association of actions to an object:

- Object relative gripper poses

Modelling all successful grasps:

- Grasp Density [1]

Example of a learned density:



[1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater.
Learning grasp affordance densities.
Paladyn Journal of Behavioral Robotics, 2:1–17, 2011.

Introduction

Grasp learning: association of actions to an object:

- Object relative gripper poses

Modelling all successful grasps:

- Grasp Density [1]

Within IRFO:

- Grasping of flexible objects
- Apply density-concept to other actions

[1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater.
Learning grasp affordance densities.
Paladyn Journal of Behavioral Robotics, 2:1–17, 2011.

Grasping flexible objects

Objective: Grasp learning

- E.g. by learning promising object relative gripper poses with KDE

Grasping flexible objects

Objective: Grasp learning

- E.g. by learning promising object relative gripper poses with KDE

Requirements:

1. Define a reference frame

Grasping flexible objects

Objective: Grasp learning

- E.g. by learning promising object relative gripper poses with KDE

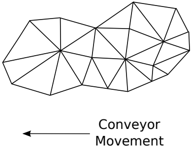
Requirements:

1. Define a reference frame
2. Parametrization of grasps

Definition of reference frame

Objects are:

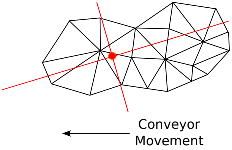
- similar, not identical
- defined by a 3D mesh



Definition of reference frame

Objects are:

- similar, not identical
- defined by a 3D mesh

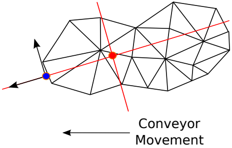


- Let *CM* be their center of mass
- Main axes are found using PCA

Definition of reference frame

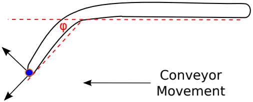
Objects are:

- similar, not identical
- defined by a 3D mesh



- Let CM be their center of mass
- Main axes are found using PCA

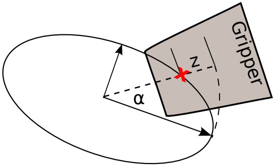
- Outmost point is found
- Object orientation + deflection φ



Concise Parametrization of Grasps

Parametrization using 2 coordinates

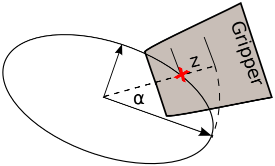
- position based on α and z
- orientation based on α or constant



Concise Parametrization of Grasps

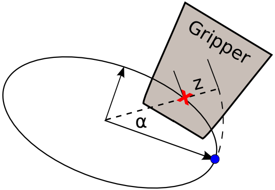
Parametrization using 2 coordinates

- position based on α and z
- orientation based on α or constant



Parametrization using 3 coordinates

- position based on α and z
- orientation based on come angle β



Results

Currently used:

- $\alpha = 0$, $z = constant$

Results

Currently used:

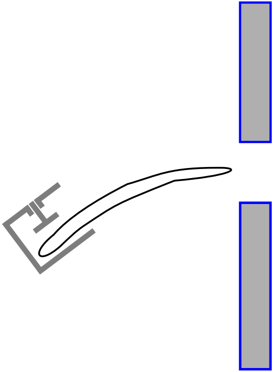
- $\alpha = 0$, $z = \mathit{constant}$

Performance on the real setup:

	Object orientation	
Object	$0° \leq \lvert\theta\rvert \leq 10°$	$10° < \lvert\theta\rvert \leq 45°$
1	100% (10)	100% (7)
⋮	⋮	⋮

Table: Success rates for different objects and orientations.

PegInHole Action



PegInHole Action

Prerequisites:

1. 3D-mesh of the object
2. Description of the scene
3. Predicted deflections (static situation)

PegInHole Action

Prerequisites:

1. 3D-mesh of the object
2. Description of the scene
3. Predicted deflections (static situation)

We want:

1. to insert the object into the hole
2. not to learn full 6D-paths by heart

PegInHole Action

Prerequisites:

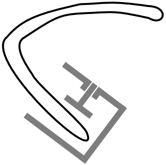
1. 3D-mesh of the object
2. Description of the scene
3. Predicted deflections (static situation)

We want:

1. to insert the object into the hole
2. not to learn full 6D-paths by heart

One assumption:

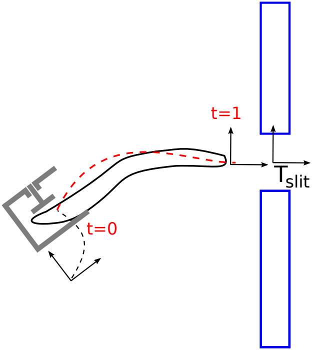
1. deflection is less than 90 degree



PegInHole Curve Approach

Suggestion:

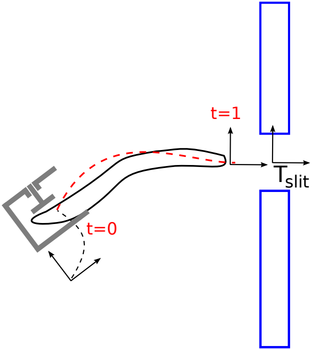
- Define a parameterized curve



PegInHole Curve Approach

Suggestion:

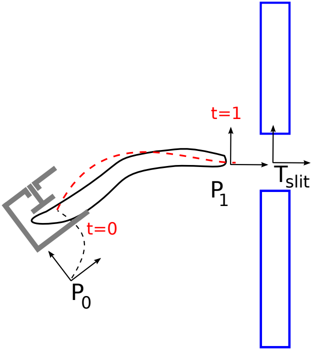
- Define a parameterized curve
- The path may consist of two dimensions for position and one for orientation.



PegInHole Curve Approach

Suggestion:

- Define a parameterized curve
- The path may consist of two dimensions for position and one for orientation.
- Let P_0 and P_1 be the coordinates corresponding to the start and target configuration, both $\in R^2 \times SO(1)$

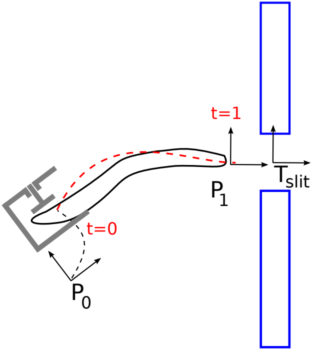


PegInHole Curve Approach

Parametric interpolation:

$$P(t) = P_0 + f(t) \circ (P_1 - P_0)$$

with $t \in [0,1]$, $f(0) = \bar{0}$ and $f(1) = \bar{1}$



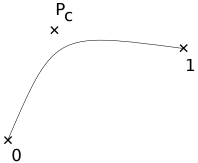
PegInHole Curve Approach

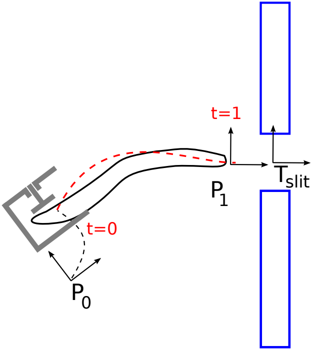
Parametric interpolation:

$$P(t) = P_0 + f(t) \circ (P_1 - P_0)$$

with $t \in [0,1]$, $f(0) = \bar{0}$ and $f(1) = \bar{1}$

Could be defined by a rational Bézier-curve, so that a control-point P_c and a scalar weight is learned:





PegInHole Curve Approach

Overall system action representation might become

$$K_{\mu,\sigma}(T_G, f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

PegInHole Curve Approach

Overall system action representation might become

$$K_{\mu,\sigma}(T_G, f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

- Kernel for PiH parameters, e.g. Gaussian

PegInHole Curve Approach

Overall system action representation might become

$$K_{\mu,\sigma}(T_G, f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

- Kernel for PiH parameters, e.g. Gaussian
- Destinctive object features

PegInHole Curve Approach

Overall system action representation might become

$$K_{\mu,\sigma}(f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

- Kernel for PiH parameters, e.g. Gaussian
- Destinctive object features

Idea: describe how the object behaves, not how it looks.

$$f = F(T_G, 3Dmesh)$$

PegInHole Curve Approach

Overall system action representation might become

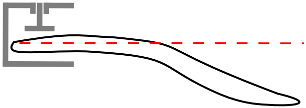
$$K_{\mu,\sigma}(f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

- Kernel for PiH parameters, e.g. Gaussian
- Destinctive object features

Idea: describe how the object behaves, not how it looks.

$$f = F(T_G, 3Dmesh)$$

We could estimate the deflections at multiple locations...



PegInHole Curve Approach

Overall system action representation might become

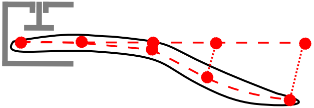
$$K_{\mu,\sigma}(f, P_c) = K^{G}_{\mu^G,\sigma^G}(T_G)\, K^{O}_{\mu^O,\sigma^O}(f)\, K^{PiH}_{\mu^P,\sigma^P}(P_c)$$

- Kernel for PiH parameters, e.g. Gaussian
- Destinctive object features

Idea: describe how the object behaves, not how it looks.

$$f = F(T_G, 3Dmesh)$$

We could estimate the deflections at multiple locations...



Learning PegInHole actions

Based on simulation, learn a density.

- Allows for exploration
- Provides detailed feedback, e.g. clearance

Learning PegInHole actions

Based on simulation, learn a density.

- Allows for exploration
- Provides detailed feedback, e.g. clearance

The weights w_i are scaled wrt. minimum clearance of the corresponding action:

$$d(P_c, f) = \sum_{i=0}^{n} w_i K_{\mu_i,\sigma}(P_c, f)$$

The Peg In Hole action with the highest clearance:

$$\arg\max_{P_c \in R^3} d(P_c, f)$$

Learning PegInHole actions

Based on simulation, learn a density.

- Allows for exploration
- Provides detailed feedback, e.g. clearance

The weights w_i are scaled wrt. minimum clearance of the corresponding action:

$$d(P_c, f) = \sum_{i=0}^{n} w_i K_{\mu_i,\sigma}(P_c, f)$$

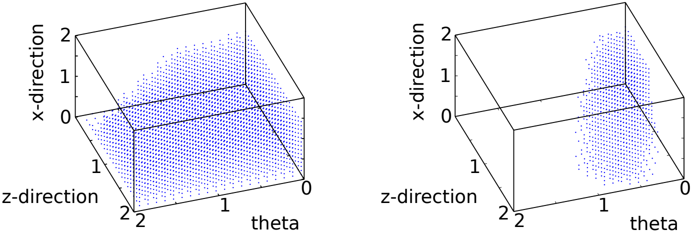
The Peg In Hole action with the highest clearance:

$$\arg \max_{P_c \in R^3} d(P_c, f)$$

Based on a set of samples from the density, evaluated on the real setup, a refined density can be achieved.

Initial results

Learned controlpoints for two different objects:



Object: $40 \times 15 \times 2mm$ Object: $80 \times 15 \times 2mm$

- Needs additional experiments

Laying Down operation

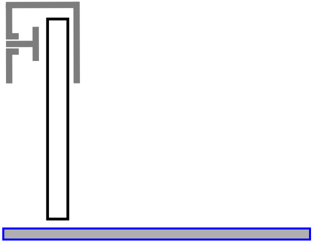
Objective:

- place object with minimum stress

Laying Down operation

Objective:

- place object with minimum stress

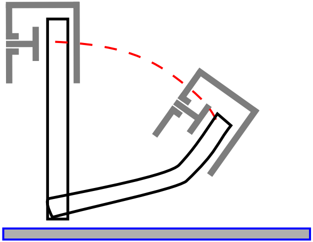


Start: no additional force applied from gripper.

Laying Down operation

Objective:

- place object with minimum stress

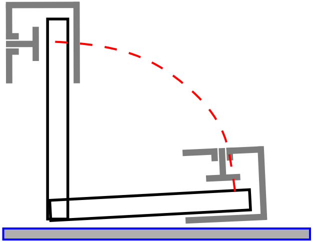


During action: potentially increased force applied by gripper

Laying Down operation

Objective:

- place object with minimum stress

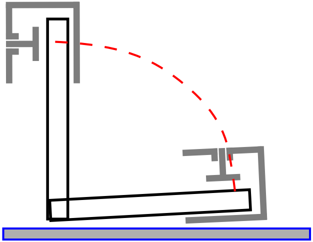


Target: Object is resting on the table.

Laying Down operation

Objective:

- place object with minimum stress



Target: Object is resting on the table.

Concept:

- reuse methodology from PiH-actions

Laying Down experiment

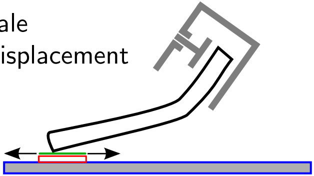
Real setup:

- Outcome is difficult to measure

Laying Down experiment

Real setup:

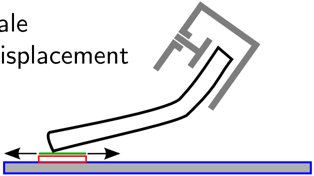
- Outcome is difficult to measure
  - Measure vertical forces with scale
  - Estimate horizontal forces by displacement
  - Estimate stress via shape(?)



Laying Down experiment

Real setup:

- Outcome is difficult to measure
  - Measure vertical forces with scale
  - Estimate horizontal forces by displacement
  - Estimate stress via shape(?)



Design of simulated experiments becomes complex:

- Supporting surface from table
- Impact of gripper
- Impact of gravity

Conclusion

Conclusions:

- Grasping objects is solved
- Learning PiH actions is developped, needs further experiments
- Strategy for Laying Down actions is proposed

Conclusion

Conclusions:

- Grasping objects is solved
- Learning PiH actions is developped, needs further experiments
- Strategy for Laying Down actions is proposed

Future task:

- Introduce feedback loop for error correction

Thank you for your attention