Push Path Improvement with Policy based Reinforcement Learning

Junhu He

TAMS
Department of Informatics
University of Hamburg
Cross-modal Interaction In Natural and Artificial Cognitive Systems (CINACS)

06.12.2016

Outline

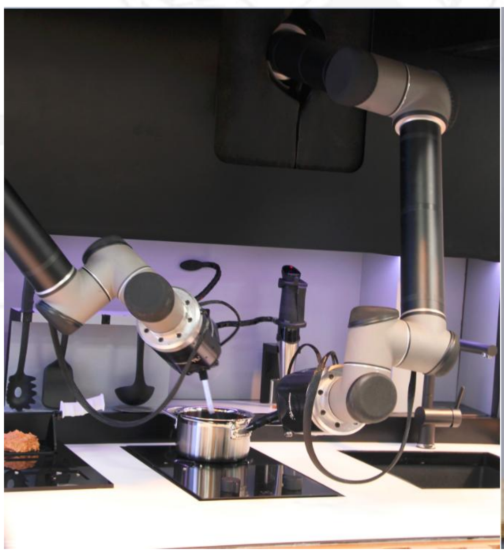
- *Motivation*
- *Research Concept*
- *Previous Works*
- *System Architecture*

- *Policy Based Reinforcement learning*
- *Simulator Training*
- *Manipulation learning*
- *Learning Result*

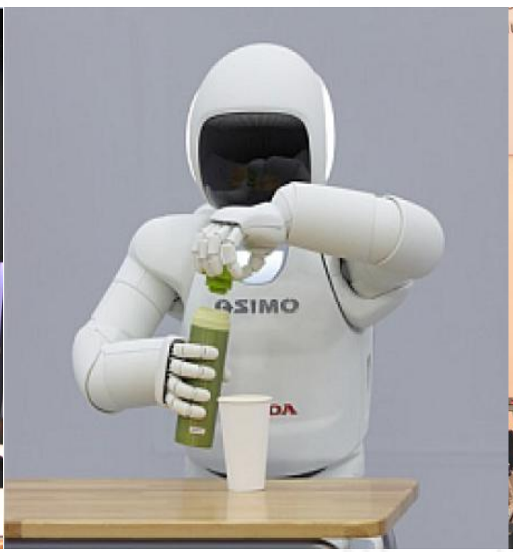
Motivation

To complete real world tasks intelligently

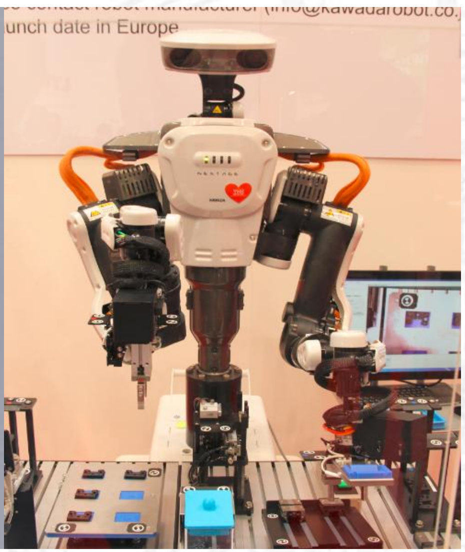
(in-hand manipulation/grasping)



Shadow Hand



ASIMO



NEXTAGE

Motivation

In-hand manipulation

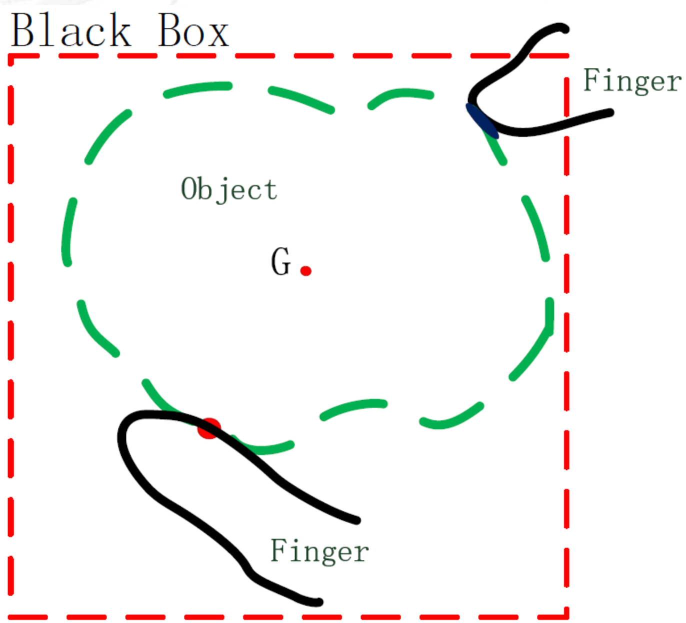
- An ability to move and position objects within one hand
- Fingers 'push' an object to generate motions.

Challenges

- A large number of joints (shadow: 19/24 DOFs)
- Complex interaction model (sensitive to errors)
- Limited perception capability (visual & tactile sensors)

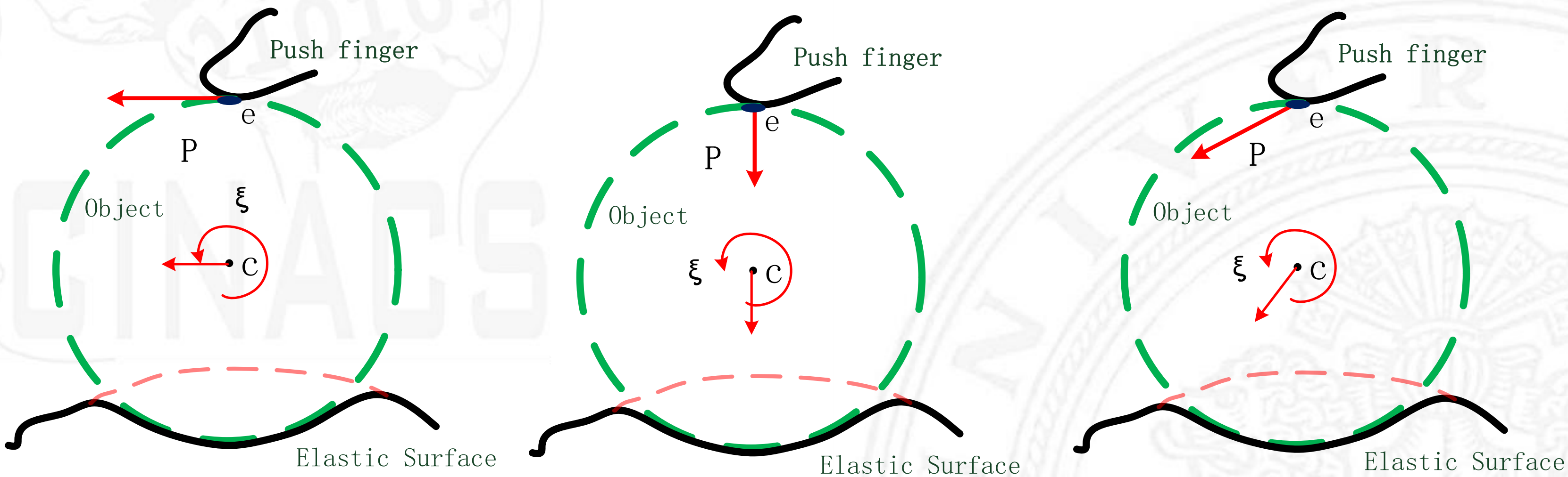
Research Concept

- In-hand interaction system is a black box
- Fingers push in the black box in different directions
- Perceive from trial and error



Research Concept

Push and Support Models



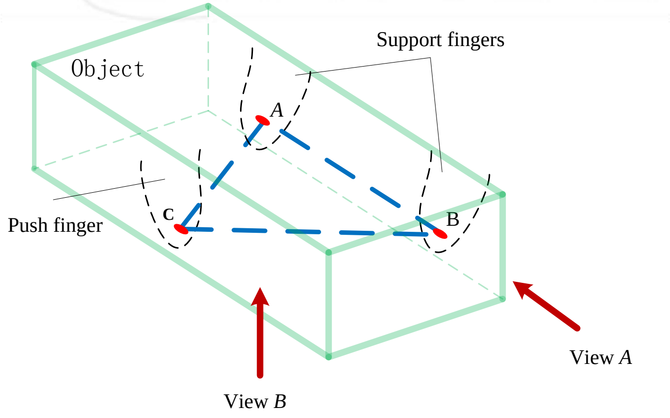
- To roll the object on an elastic surface
- Trade off down and forward motions

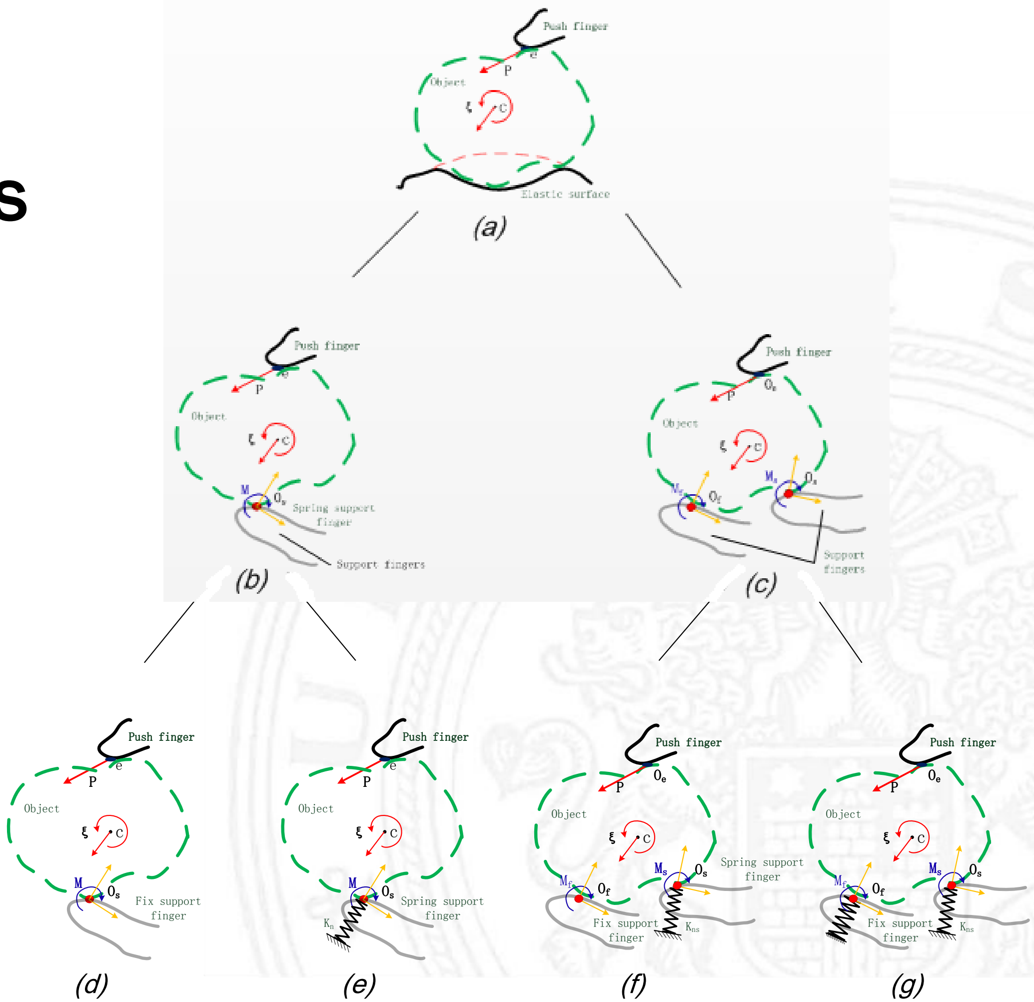
Research Concept

Model Evolution

Different support fingers

- Number (Different views)
- Type:
  - Fixed support finger
  - Spring support finger





Manipulation Model

Enhanced Manipulation Model

Hybrid support model
for yaw manipulation

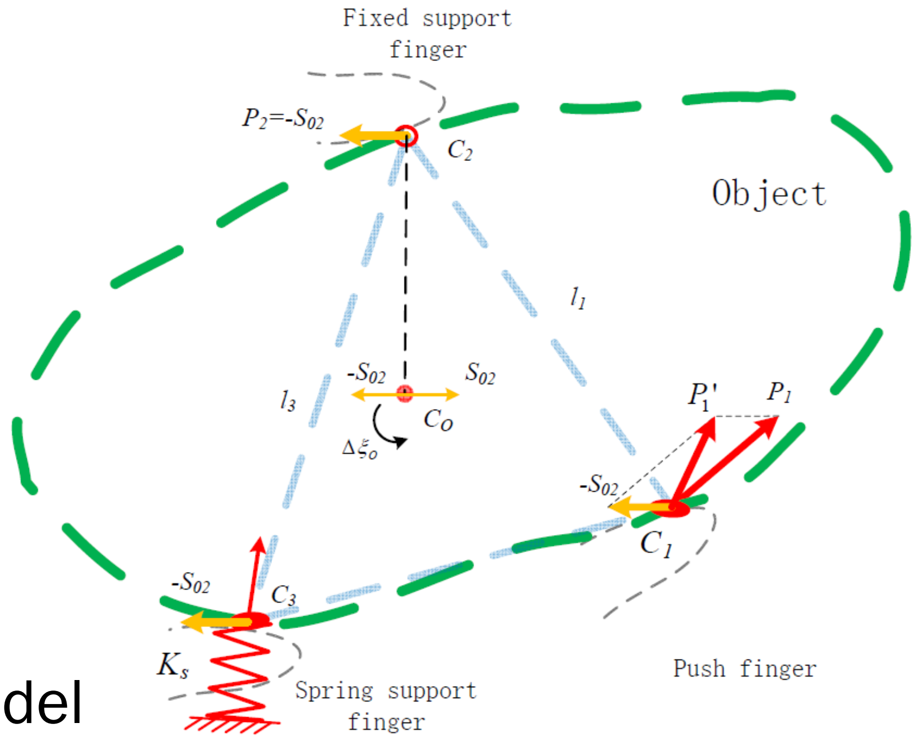
+

Opposite Velocity

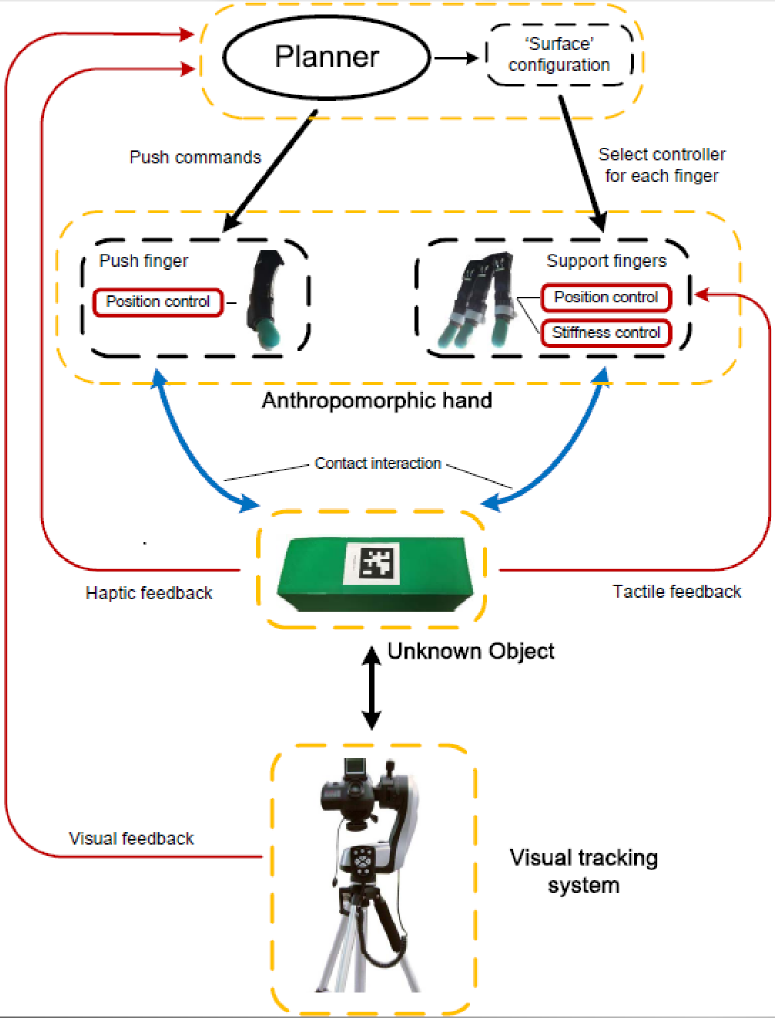
$$\mathrm{P}_2 = -\mathrm{S}_{O2}$$

=

Enhanced manipulation model



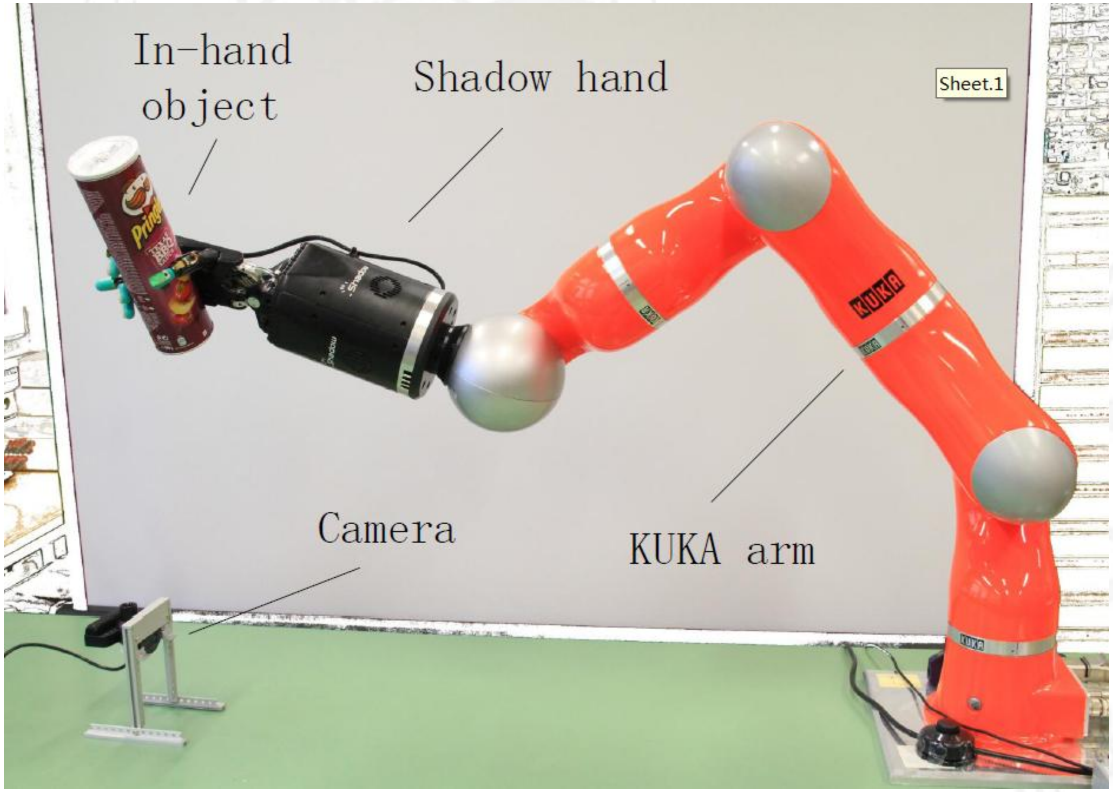
System Architecture

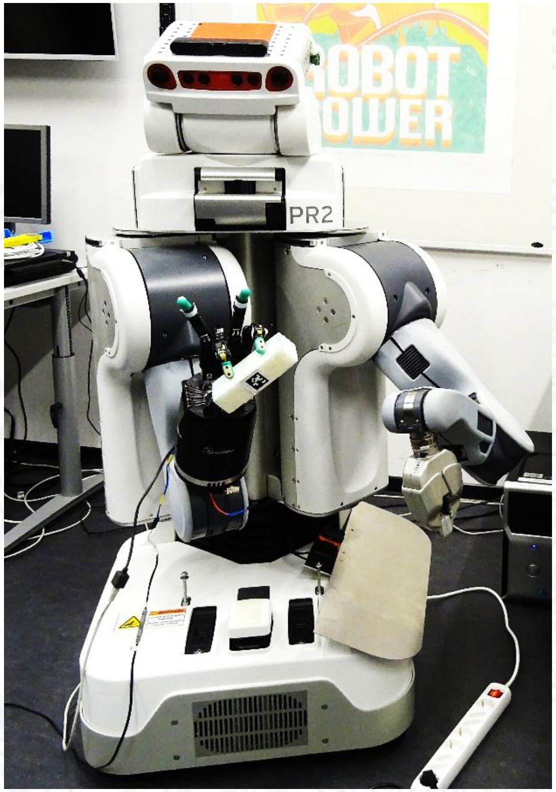


- Robot hand: Shadow hand (Anthropomorphic, 19 DOFs, tenden dirven)
- Haptic sensing: BioTac (force, vibration and temperature, etc.)
- Visual tacking: AprilTags (2D barcode)

Experiment

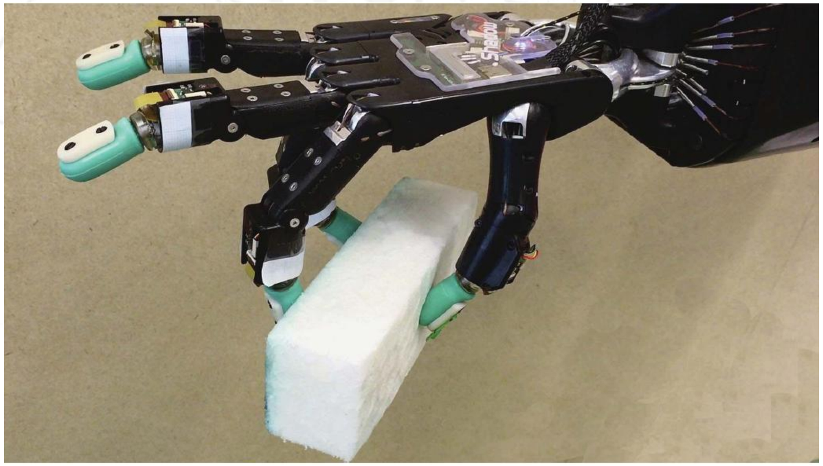
Experiment Setup

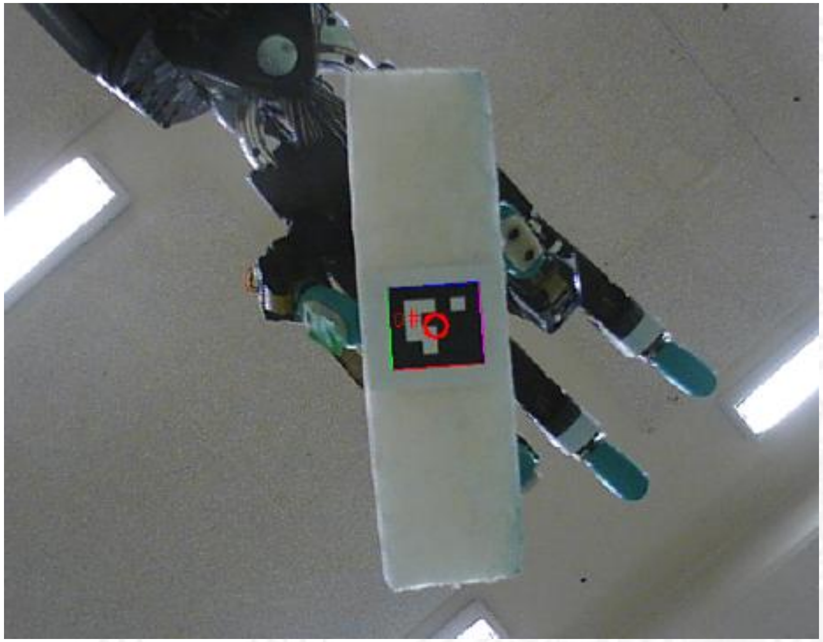




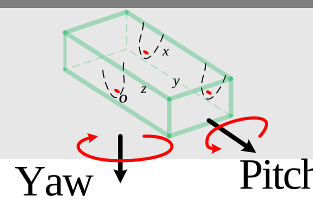
Experiments

Initial Grasping Configuration

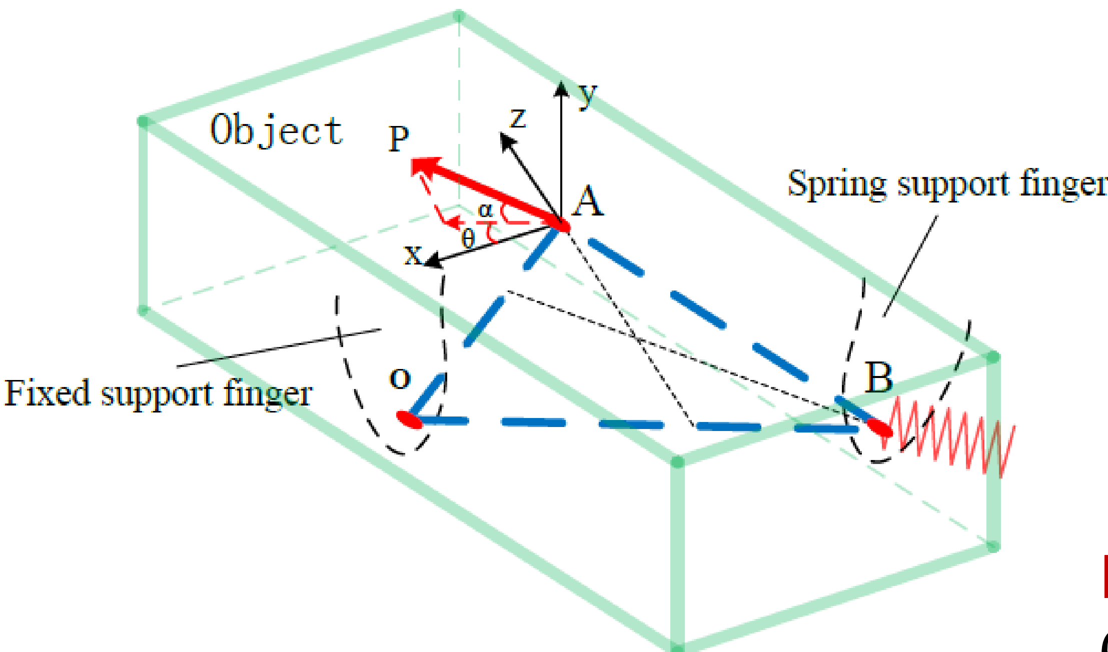


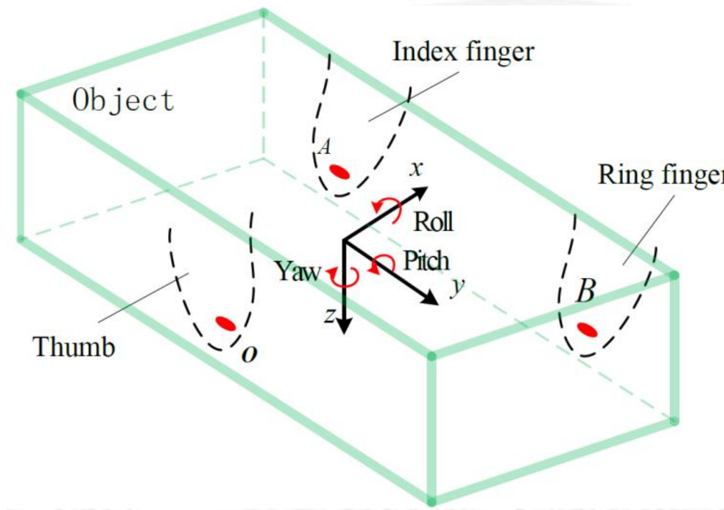


Experiments



Rotational Manipulation (Yaw)





Index finger pushes

θ from -60° to 60°

α from -60° to 60°

Experiments

Haptic feature

$$\mathbf{pK} = \begin{bmatrix} pk_1 \\ pk_2 \\ pk_4 \end{bmatrix}$$

Haptic reward

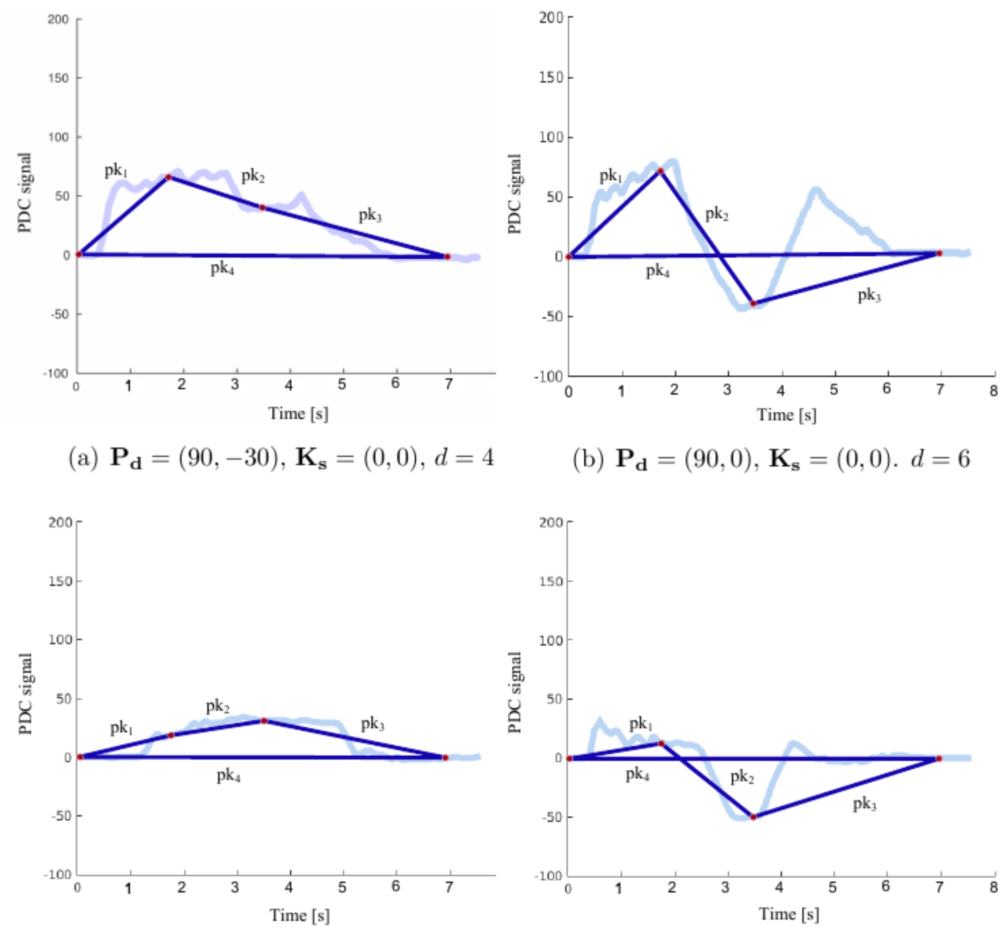
$$R_H = -d(\mathbf{pK}, \mathbf{pK}')$$

Visual feature

Object's Rotation: $\boldsymbol{V_r}^T$

Visual reward

$$R_V = \mathbf{V_r}^T \mathbf{V}'_r$$



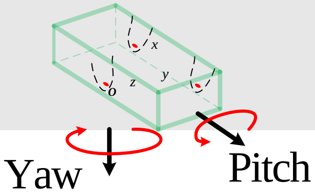
(a) $\mathbf{P_d} = (90, -30)$, $\mathbf{K_s} = (0, 0)$, $d = 4$

(b) $\mathbf{P_d} = (90, 0)$, $\mathbf{K_s} = (0, 0)$. $d = 6$

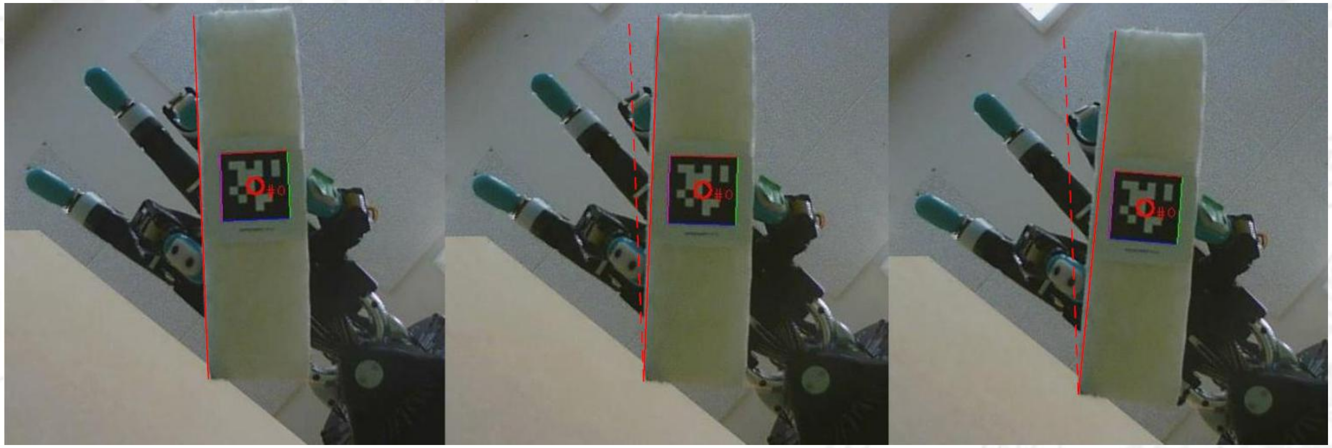
(c) $\mathbf{P_d} = (-30, -30)$, $\mathbf{K_s} = (0, 1)$. $d = 4$

(d) $\mathbf{P_d} = (-30, 0)$, $\mathbf{K_s} = (0, 1)$, $d = 4$

Experiments

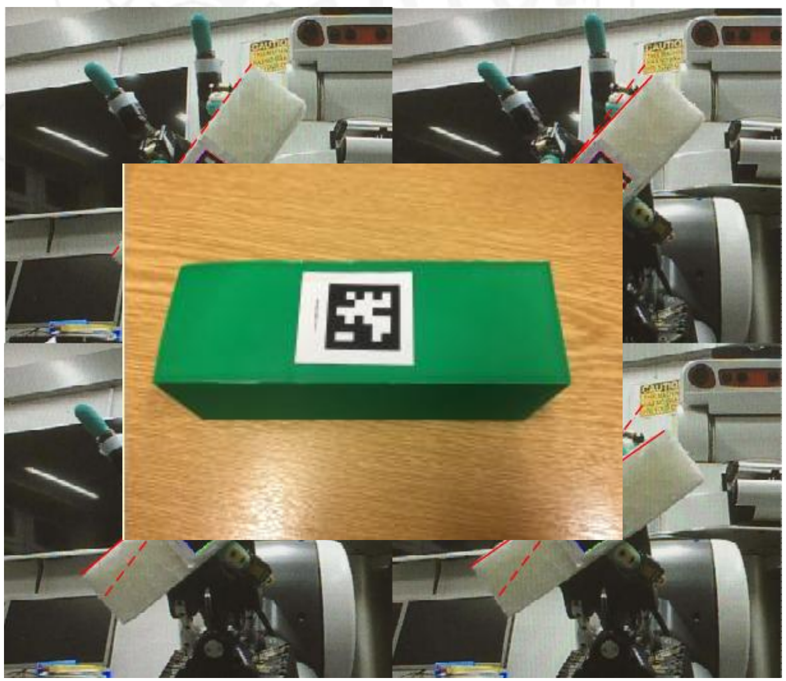


Snapshots (rotational manipulation)



Experiments

Enhanced Manipulation



Rigid object

(a) A plastic bottle.

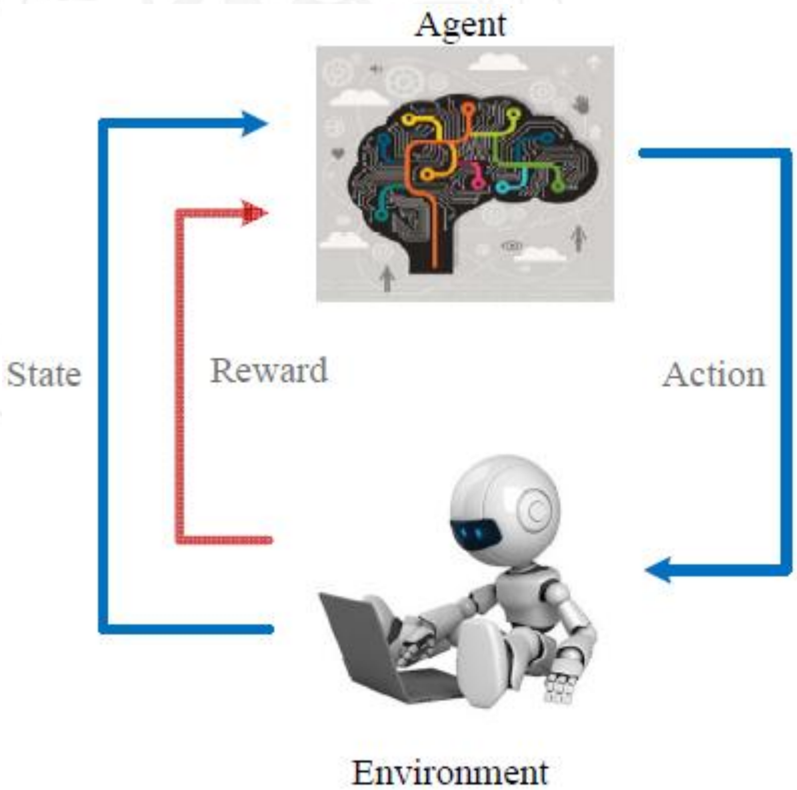
(b) A remote control.

(c) A caffee pack.

(d) A square foam piece.

Policy Based Reinforcement Learning

Reinforcement Learning



Markov Decision Process (MDP)

$$\langle X, U, f, \rho \rangle$$

State: x

Action: u

Reward: r

Policy Based Reinforcement Learning

Cost function (cost function): J

The gradient of the cost function:

$$\nabla_\theta J(\theta) = \int_X d^\pi(x) \int_U \nabla_\theta \pi(x,u)(Q^\pi(x,u) - b^\pi(x))du dx.$$

Stationary distribution of the state

$$d^\pi(x) = lim_{t\to\infty} P\{x_t = x | x_0, \pi\}$$

Q function $$Q^\pi(x) = E\{\sum_{k=0}^{\infty} \gamma^k r_{k+1} | x_0 = x, u_0 = u, \pi\}.$$

Baseline $b^\pi(x)$

Policy Based Reinforcement Learning

Williams' Episodic REINFORCE algorithm

$$\nabla_\theta J(\theta) = \langle (\sum_{k=0}^{n} \nabla_\theta \pi_\theta(u_k|x_k))(\sum_{k=0}^{n} a_k r_k - b)\rangle .$$

Peters' Episodic Actor-Critic algorithm

$$\begin{aligned}\nabla_\theta J(\theta) &= \int_X d^\pi(x) \int_U \nabla_\theta \pi(x,u)(\nabla_\theta log\pi(u|x))^T du dx \mathbf{w} \\ &= F_\theta \mathbf{w},\end{aligned}$$

With compatible function:

$$f_w^\pi(x,u) = (\nabla_\theta log\pi(u|x))^T \mathbf{w} \equiv Q^\pi(x,u) - b^\pi(x).$$

Policy Based Reinforcement Learning

Algorithm 1 Episodic REINFORCE algorithm

Input: policy parameter θ, learning rate α, policy standard deviation σ, and baseline b;
for episode j **do**
  Initialization: $\pi_\theta \leftarrow \theta$, get initial state X_0;
  for each step i **do**
    $u_k \leftarrow \pi(\theta)$ and do action u_k;
    get next state X_{k+1} and reward r_k;
    $r = r + r_k$;
    $\mathrm{e} = \mathrm{e} + \frac{\partial ln(\pi_\theta(X_k))}{\partial\theta}$;
  end for
  b = b + (r - b) / j;
  $\theta_{k+1} = \theta_k + \alpha_k(r - b)\mathrm{e}$;
end for

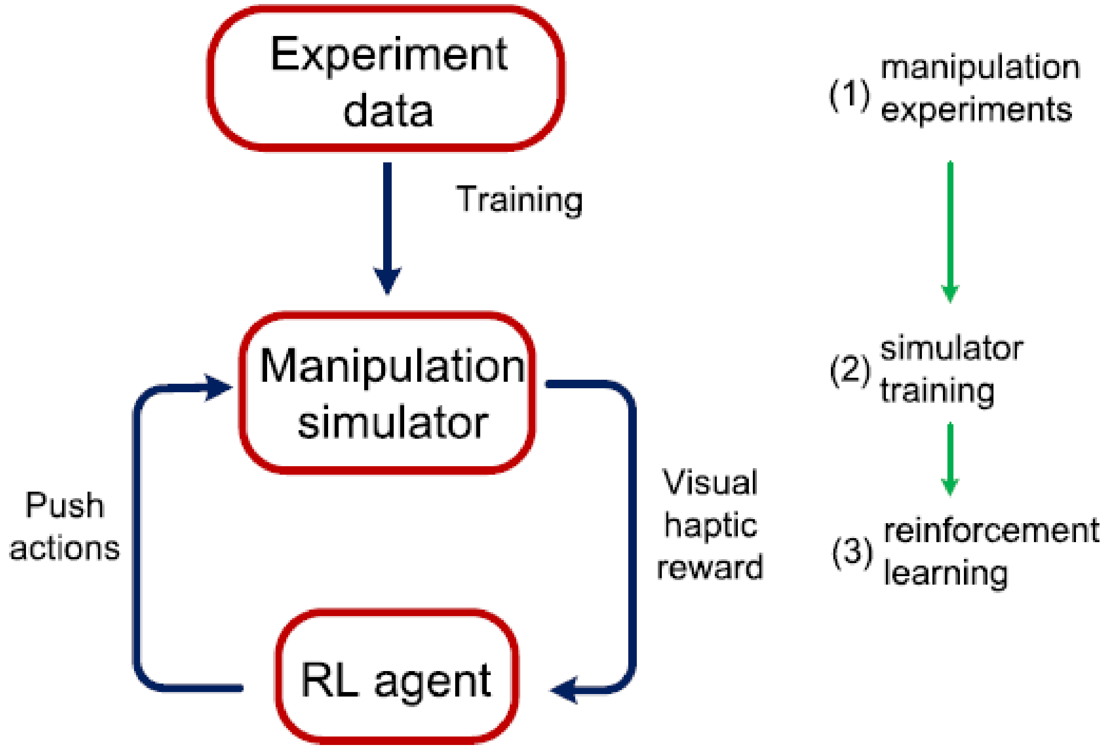
Policy Based Reinforcement Learning

Algorithm 2 Peters' Episodic Actor Critic algorithm

Input: policy parameters θ, learning rate α, policy standard deviation σ;
repeat
  for m episodes **do**
    Initialization: $\pi_\theta \leftarrow \theta$, get initial state $\mathbf{x}_0$;
    Calculate:
    policy derivatives: $\psi_k = \nabla_\theta \log \pi_\theta(\mathbf{u}_k|\mathbf{x}_k)$
    fisher matrix $\mathbf{F}_\theta = \langle(\sum_{k=0}^{H} \psi_k)(\sum_{l=0}^{H} \psi_l)^T\rangle$.
    vanilla gradient $\mathbf{g} = \langle(\sum_{H}^{k=0} \psi_\mathbf{k})(\gamma^{(H-k)}r)\rangle$.
    average reward $\overline{r} = \langle\sum_{k=0}^{H} \gamma^{H-k}r\rangle$.
    eligibility $\phi = \langle\sum_{k=0}^{H} \psi\rangle$.
    natural gradient:
    baseline $b = \mathbf{Q}(\overline{r} - \phi\mathbf{F}_\theta^{-1}\mathbf{g})$
    where $\mathbf{Q} = \frac{1}{m}(1 + \phi^T(m\mathbf{F}_\theta - \phi\phi^T)^{-1}\phi)$
    natural gradient $g_n = \mathbf{F}_\theta^{-1}(\mathbf{g} - \phi b)$
  end for
  policy update $\theta = (1 - \frac{\alpha}{n})\theta + \frac{\alpha}{n}\mathbf{g}_n$
until θ converged

Policy Based Reinforcement Learning

Learning Frame



Simulator Training

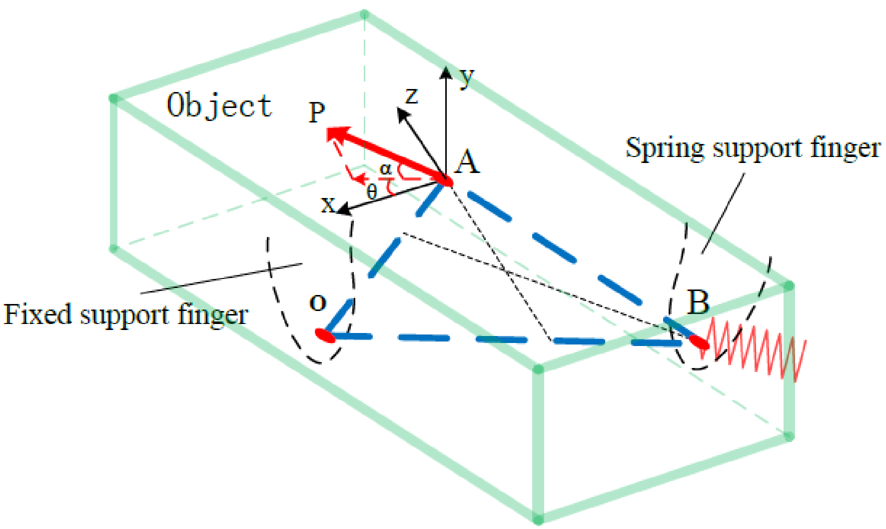
Density Push experiment:

Push action: $\mathbf{P} = [\theta, \alpha, P_l]^T$

Algorithm 1 The push sequence in the density push experiment.

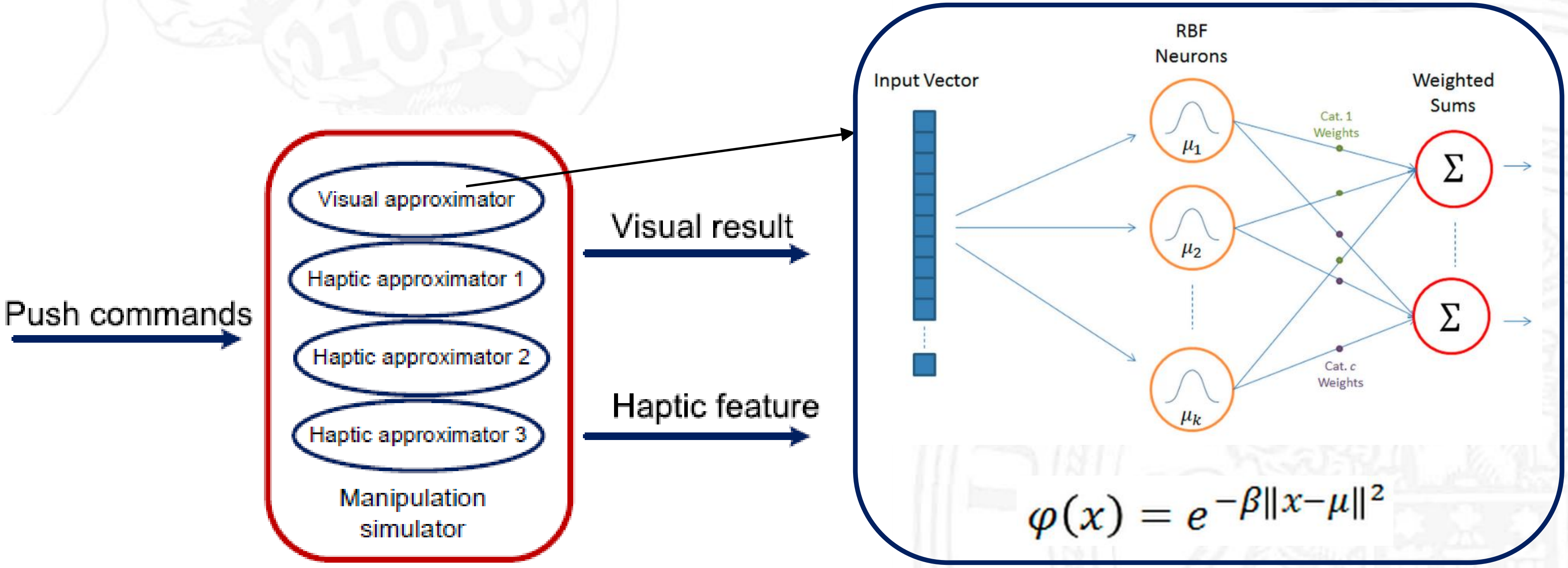
```
for P_l = 2 to 10 mm do
    for θ = −60° to 60° by 15° do
        for α = −60° to 60° by 15° do
            Push execution with P = [θ, α, P_l]^T;
        end for
    end for
end for
```

380 push actions



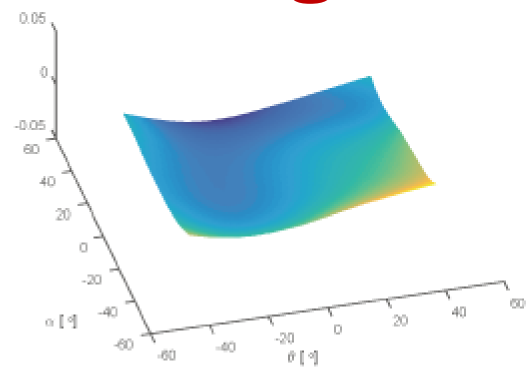
Simulator Training

RBFNs: Radial Basis Function Networks

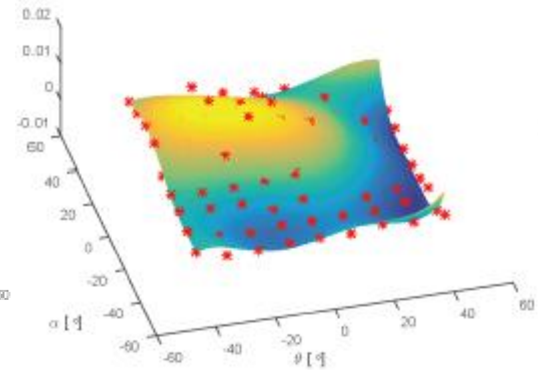


Simulator Training

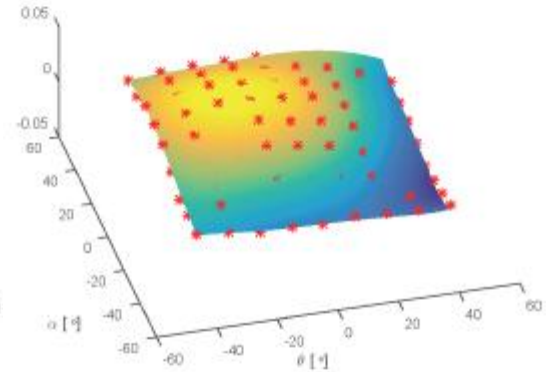
Visual regression with RBFN



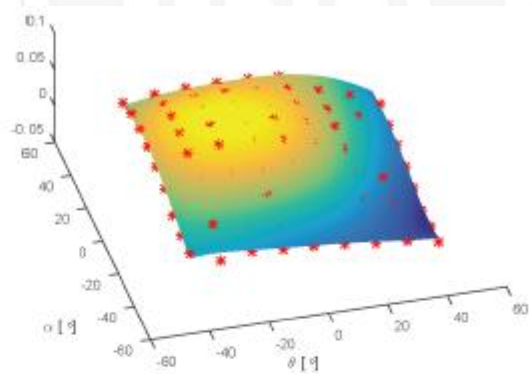
(a) $d = 0\ mm$



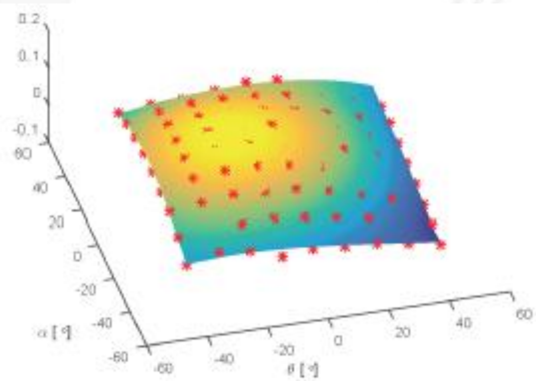
(b) $d = 2\ mm$



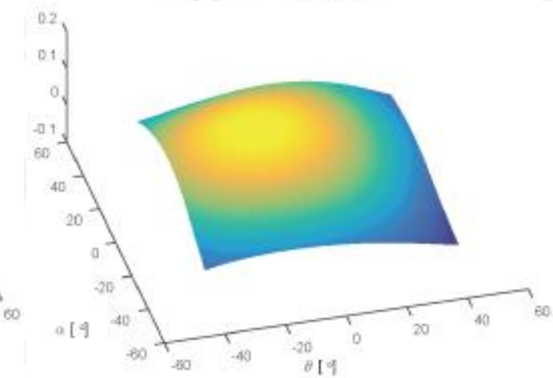
(c) $d = 4\ mm$



(d) $d = 6\ mm$



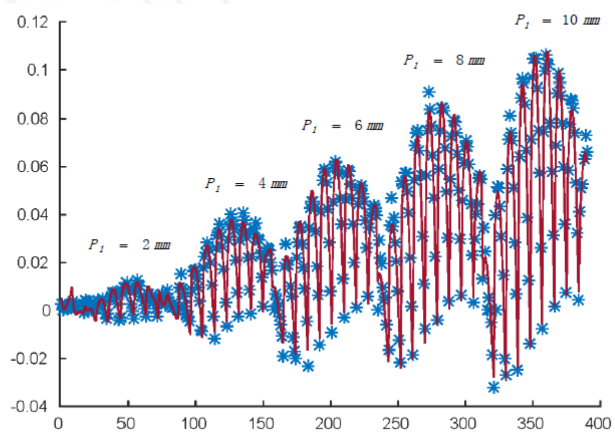
(f) $d = 10\ mm$



(h) $d = 14\ mm$

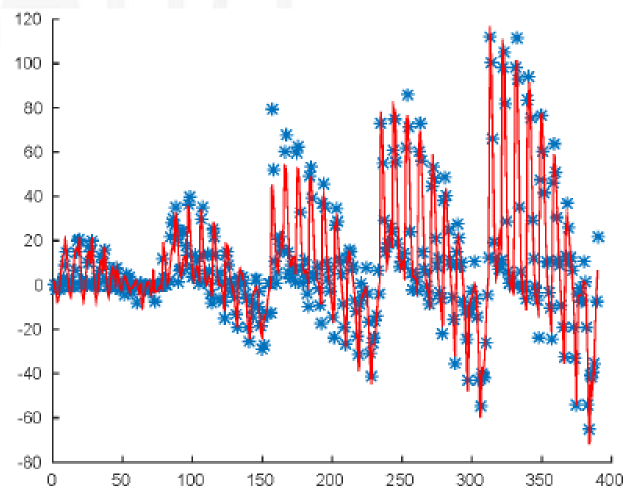
Simulator Training

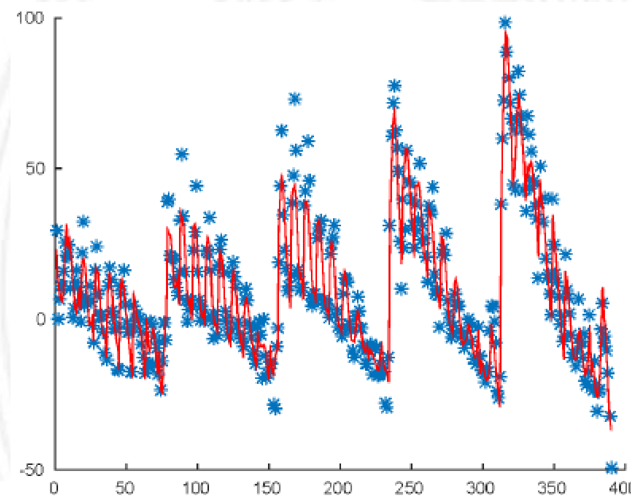
Visual:



- Approximate visual result with RBFNs
- Conduct as a simulator for the learning agents

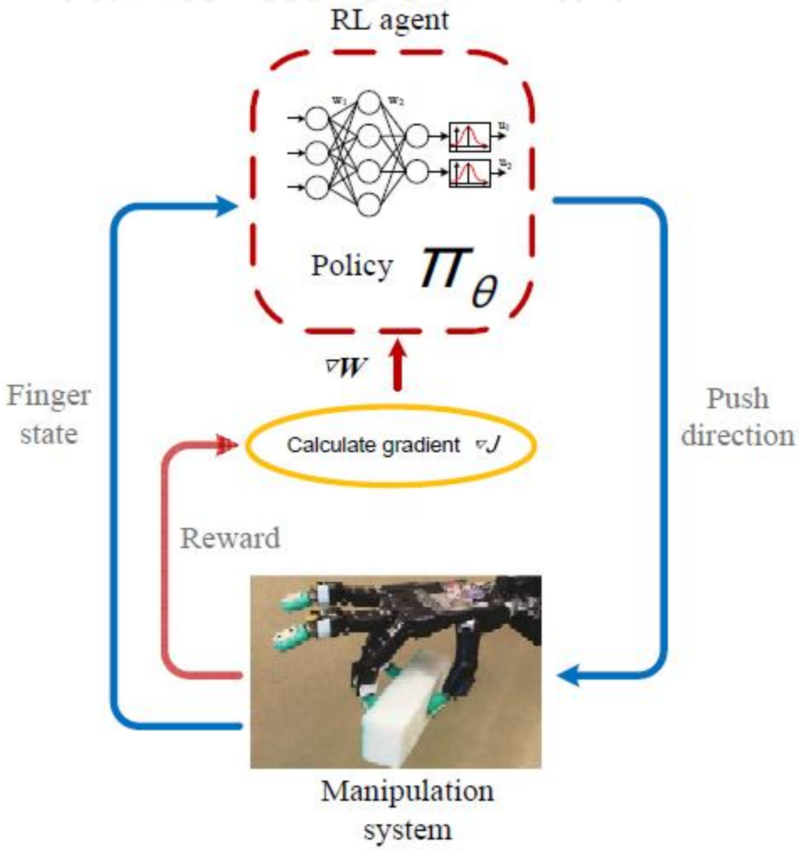
Haptic:





Manipulation learning with simulators

Manipulation learning with simulators



- Learn to push object in a right direction
- Interact with visual and haptic simulators

Manipulation learning with simulators

Reward

$$r = \sum_{k=1}^{H-1} \gamma^{H-k} \mathbf{k}^T \mathbf{r}_{step}(\mathbf{x}_t) + r_{stepV}(\mathbf{x}_H)$$

$$\mathbf{r}_{step}(\mathbf{x}_t) = [r_{stepV}(\mathbf{x}_t)\ r_{stepH}(\mathbf{x}_t)]^T$$

Visual only $\mathbf{k} = [1\ \ 0]^T$

Visual-haptic $\mathbf{k} = [1\ \ 1/100]^T$

Final reward $r_{end} = 10 r_{stepV}(\mathbf{x}_{end})$

Episodic REINFORCE Algorithm

Learning Parameters

Williams' Episodic REINFORCE Algorithm

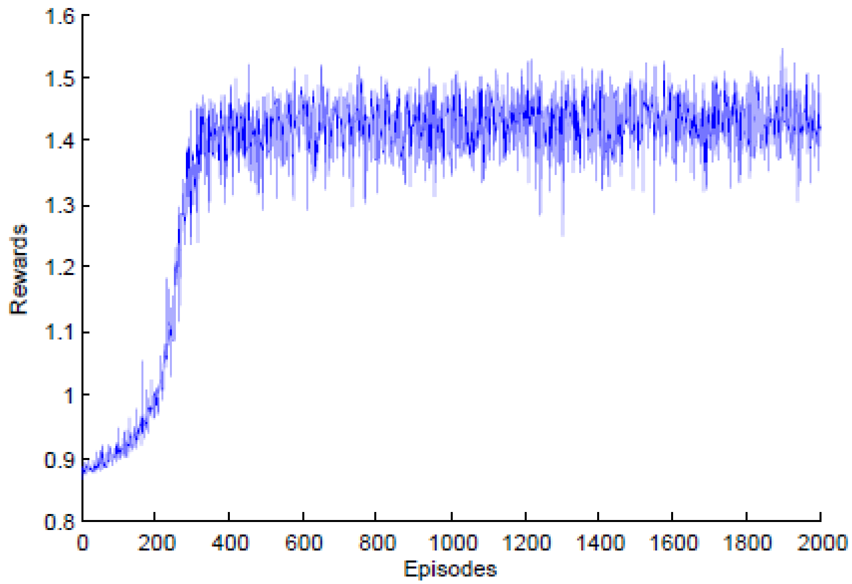
Parameters	Notation	value
State dimensions	n_s	3
Action dimensions	n_u	2
Hidden layers units	n_h	8
Learning rate	α	0.005
Discount factor	γ	0.8
Policy standard deviation	σ	0.4
Steps in one episode	H	12
Maximum episode number	n_{ep}	2000

Peter's Episodic Natural Actor-Critic

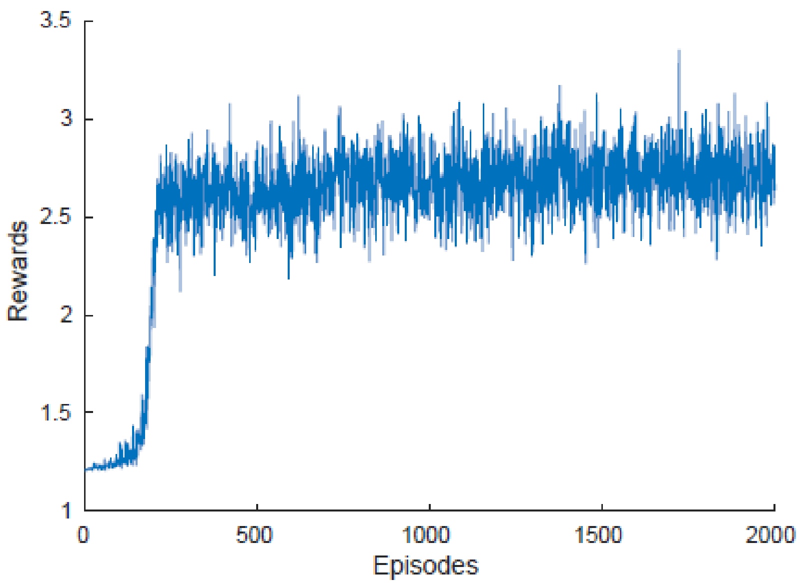
Parameters	Notation	value
State dimensions	n_s	3
State feature dimensions	n_{sf}	10
Action dimensions	n_u	2
Learning rate	α	0.5
Discount factor	γ	0.8
Policy standard deviation	σ	0.2
Steps in one episode	n_{step}	12
Maximum episode number	n_{ep}	1000

Learning Results

Episodic REINFORCE Algorithm



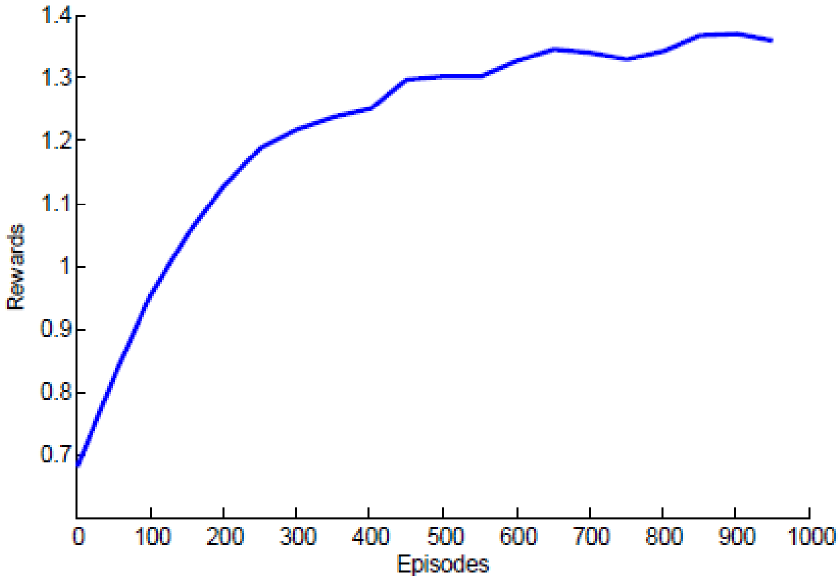
Visual-Only



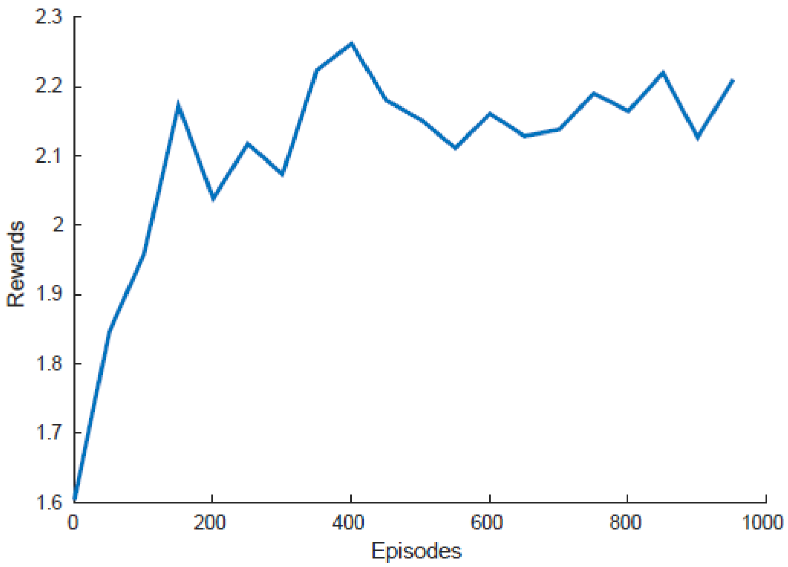
Visual-Haptic

Learning Results

Episodic Natural Actor-Critic



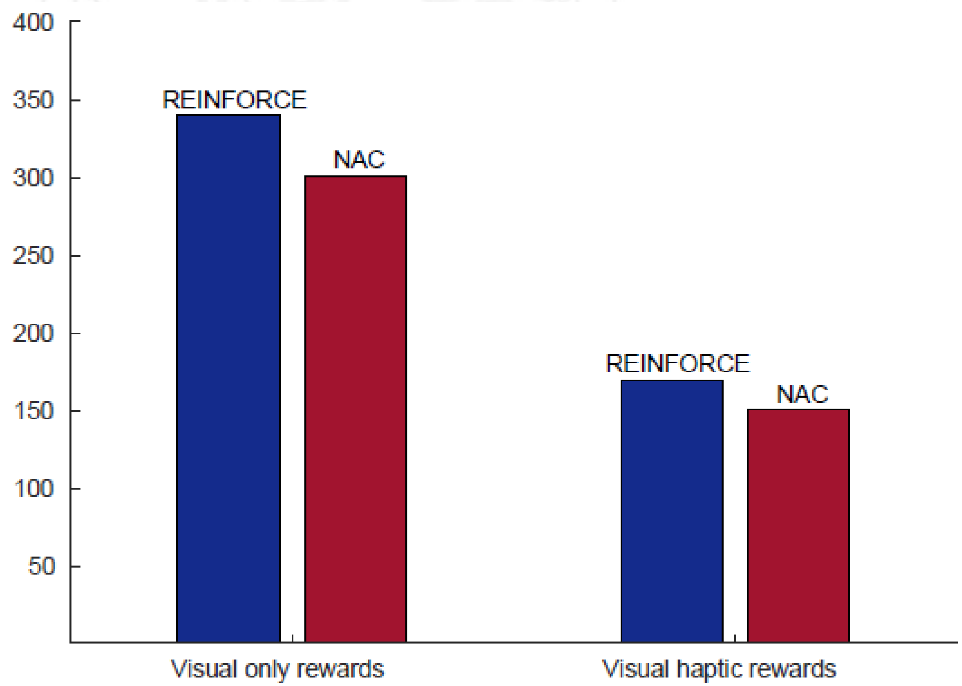
Visual-Only



Visual-Haptic

Learning Results

Episode Number before Learned



- NAC is little faster than REINFORCE
- Multimodal(Visual-Haptic) speeds up learning speed than unimodal (Visual-Only)

Thank You!

Junhu He
he@informatik.uni-hamburg.de

TAMS
Department of Informatics
University of Hamburg
Cross-modal Interaction In Natural and Artificial Cognitive Systems(CINACS)