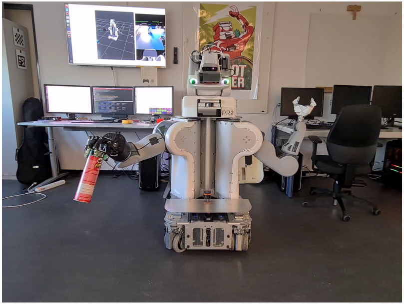
Bimanual Robot-To-Robot Handover Utilizing Multi-Modal Feedback

Björn Sygo

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

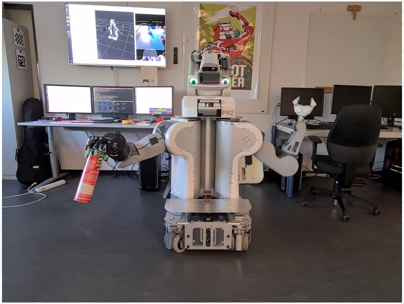
16. January 2024


PR2


PR2

Introduction

- Increased workspace by using both manipulators
- Allows easier re-orientation of objects
- Shadow hand is difficult to control for grasping objects
- PR2 has a limited workspace where both manipulators can act
- Run everything on the real robot without simulation



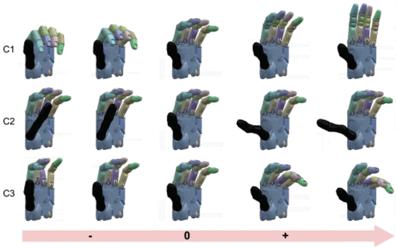
Manipulation

- Liang et al., Multifingered Grasping
- Use reinforcement learning to learn to grasp various objects
- Utilize synergies to control the Shadow hand
- Already available dataset for synergies
- Not tested for bimanual system

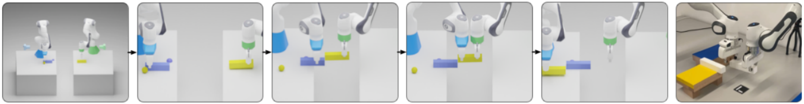
Liang et al., "Multifingered Grasping Based on Multimodal Reinforcement Learning", RA-L 2022

Synergies

- Human-inspired dimensionality reduction method for humanoid hands
- Record humans grasping various objects in different ways
- Run a Principal Component Analysis on recorded poses
- Use a weighted combination of first x eigenvectors to control hand



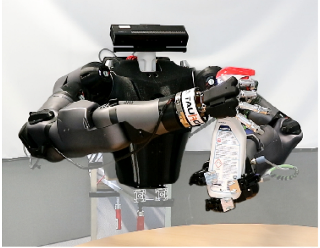
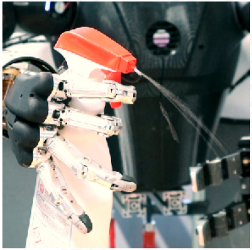
- Li et al., Efficient Bimanual Handover
- Multiple handovers using two panda arms
- Utilize symmetry between the two arms for efficient training
- SAC as backbone algorithm
- Only equipped with two finger grippers
- Simple block shapes as objects



Li et al., "Efficient Bimanual Handover and Rearrangement via Symmetry-Aware Actor-Critic Learning", ICRA 2023

Bimanual Manipulation

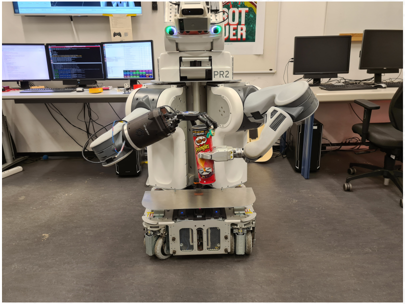
- Pavlichenko et al., Bimanual functional regrasping
- First generate and execute support grasp, then perform functional re-grasping
- Use mesh reshaping to handle objects from the same category
- Different multi-fingered manipulators used
- Functional grasp predetermined
- Difficult to expand to more object categories

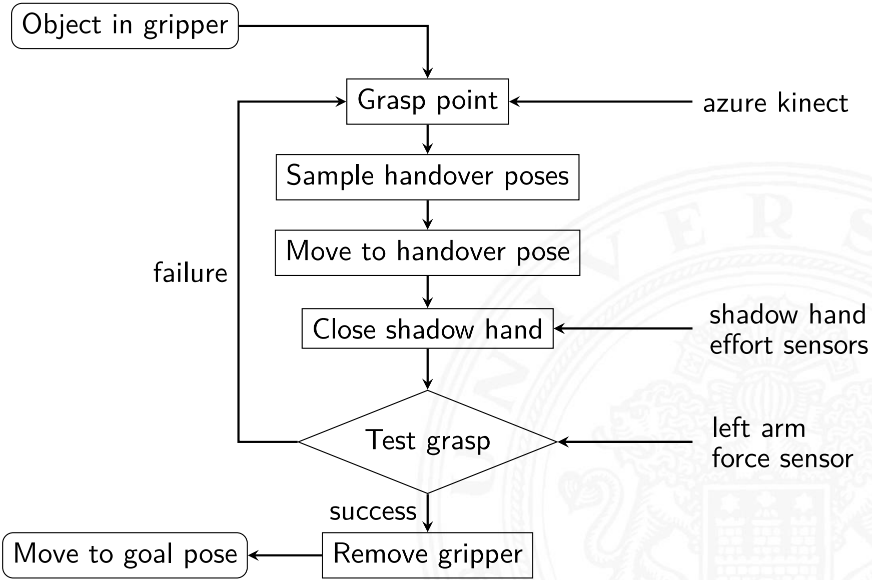
Pavlichenko et al., "Autonomous Bimanual Functional Regrasping of Novel Object Class Instances", Humanoids 2019

Setup

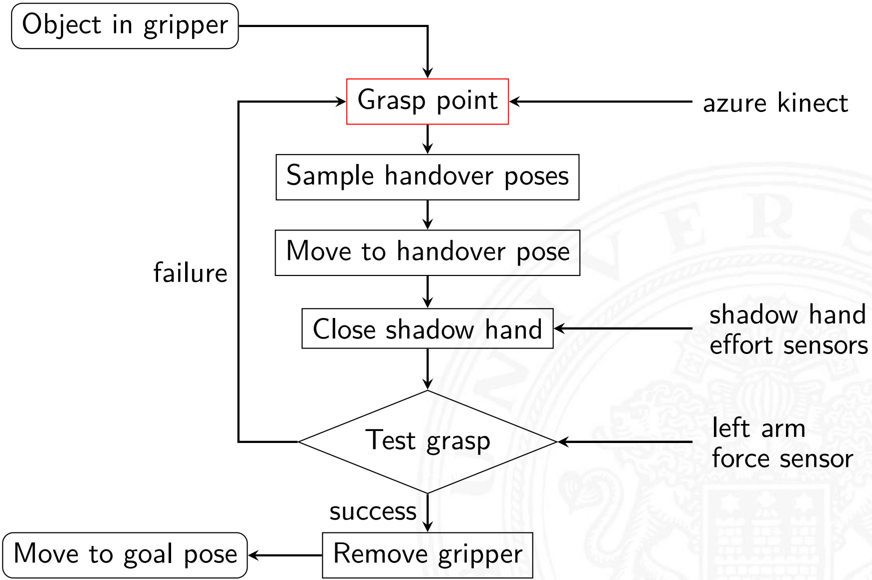
- Utilize PR2
- Enable the robot to hand over from left to right manipulator
- Different manipulators provide unique challenges in both directions
- Assumption to have an object already in the gripper at the start
- No external sensors



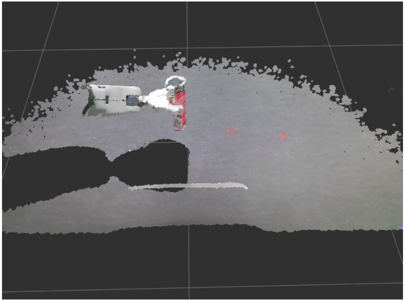
Structure

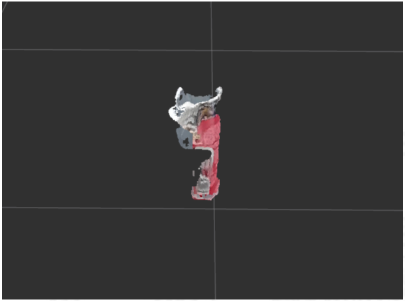

Object in gripper
Grasp point
azure kinect
Sample handover poses
Move to handover pose
failure
Close shadow hand
shadow hand
effort sensors
Test grasp
left arm
force sensor
success
Move to goal pose
Remove gripper

Grasp Point


Object in gripper
Grasp point
azure kinect
Sample handover poses
Move to handover pose
failure
Close shadow hand
shadow hand
effort sensors
Test grasp
left arm
force sensor
success
Move to goal pose
Remove gripper

Pointcloud Filter

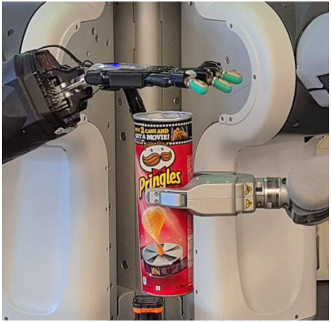




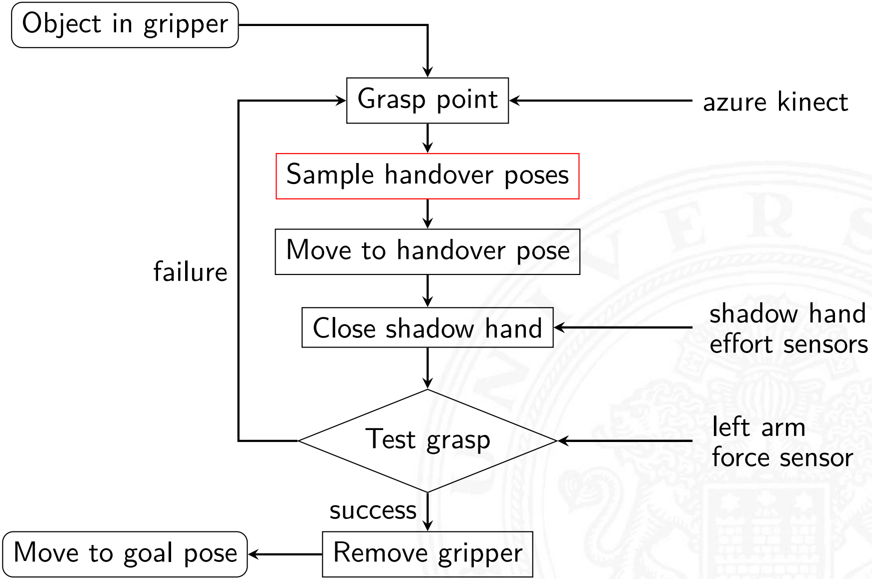
- Apply fixed cropbox filter around the gripper
- Filter remaining robot through robot self-filter package
- Reduction from 3145728 points to 37183 points

Grasp Point Generation

- Initial training used the highest point in point cloud
- For can use rotation invariant IK pose generation
- With changing handover poses switched to fixed translation relative to the gripper
- Possibly utilize GPD on filtered object point cloud in the future

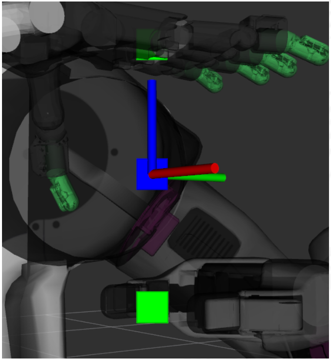


Handover Pose



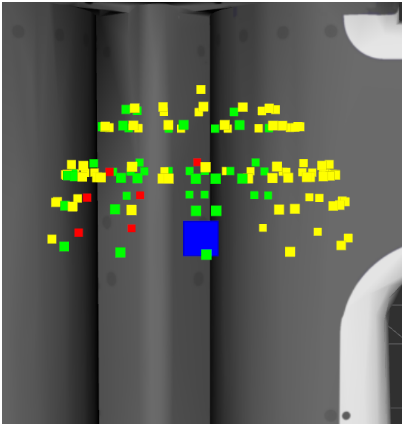
Handover Pose Workspace Analysis

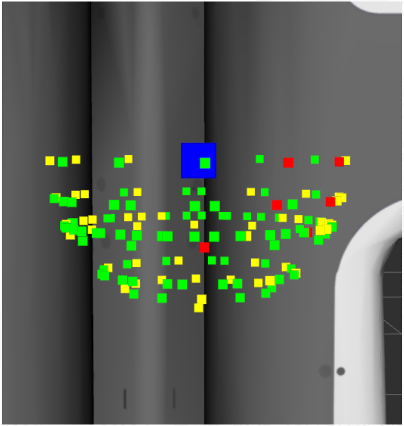
- Need to decide where to perform the handover process
- Investigate the best region to sample
- Workspace analysis to determine the optimal sampling area
- Use handover points as the middle point between the gripper and hand



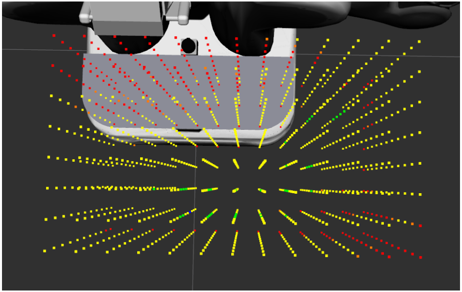
Handover Pose Workspace Analysis

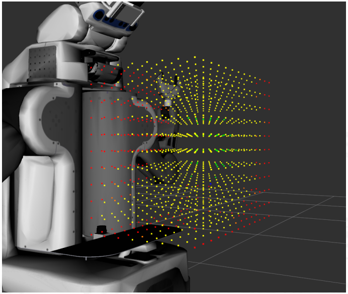
- Sample every 6cm along all axes
- 30°steps rotation around each axis from -90°to 90°for gripper and hand, resulting in 343 orientations per position
- Rotations relative to the handover point





Handover Pose Workspace Analysis Visualization





- From red to green increasing the number of valid configurations
- Shown positions are handover points
- For each position, all gripper and hand orientations were tested
- Best region towards gripper side

Handover Pose Cost Assignment

- Decide which valid pose to choose
- Cost function used by Pavlichenko et al.
- Find configuration far from joint limits
- Ensures enough flexibility for grasp testing/retracting gripper
- Use lowest cost pose for handover

$\delta(\theta) = min(|\theta_{upper} - \theta|, |\theta - \theta_{lower}|)$

θ : joint values

$\theta_{upper,lower}$: upper/lower joint limit

$$c(\theta) = \frac{1}{|\delta(\theta)|} \sum_{i=1}^{|\delta(\theta)|} \frac{1}{\epsilon_i^2}(\delta(\theta_i))^2 - \frac{2}{\epsilon_i}\delta(\theta_i) + 1$$

ϵ_i : $\frac{1}{2}(\theta_{upper} - \theta_{lower})$ for joint i

Handover Pose Cost Assignment

- Decide which valid pose to choose
- Cost function used by Pavlichenko et al.
- Find configuration far from joint limits
- Ensures enough flexibility for grasp testing/retracting gripper
- Use lowest cost pose for handover

$$\delta(\theta) = min(|\theta_{upper} - \theta|, |\theta - \theta_{lower}|)$$

θ : joint values

$\theta_{upper,lower}$: upper/lower joint limit

$$c(\theta) = \frac{1}{|\delta(\theta)|} \sum_{i=1}^{|\delta(\theta)|} \frac{1}{\epsilon_i^2}(\delta(\theta_i))^2 - \frac{2}{\epsilon_i}\delta(\theta_i) + 1$$

ϵ_i : $\frac{1}{2}(\theta_{upper} - \theta_{lower})$ for joint i

Handover Pose Cost Assignment

- Decide which valid pose to choose
- Cost function used by Pavlichenko et al.
- Find configuration far from joint limits
- Ensures enough flexibility for grasp testing/retracting gripper
- Use lowest cost pose for handover

$\delta(\theta) = min(|\theta_{upper} - \theta|, |\theta - \theta_{lower}|)$

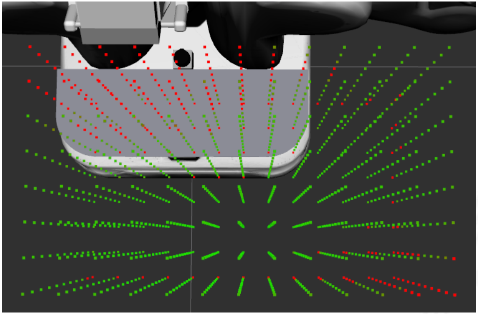
θ : joint values

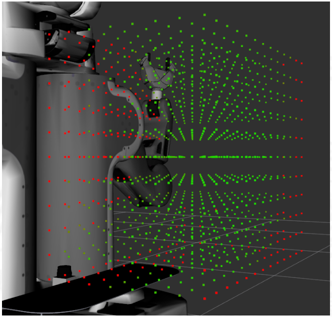
$\theta_{upper,lower}$: upper/lower joint limit

$$c(\theta) = \frac{1}{|\delta(\theta)|} \sum_{i=1}^{|\delta(\theta)|} \frac{1}{\epsilon_i^2} (\delta(\theta_i))^2 - \frac{2}{\epsilon_i} \delta(\theta_i) + 1$$

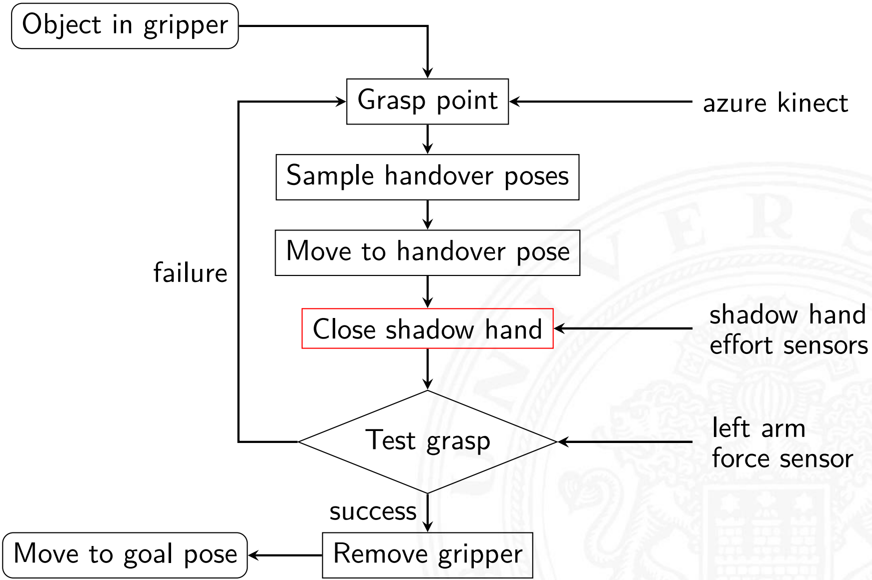
ϵ_i : $\frac{1}{2}(\theta_{upper} - \theta_{lower})$ for joint i

Handover Pose Cost Map



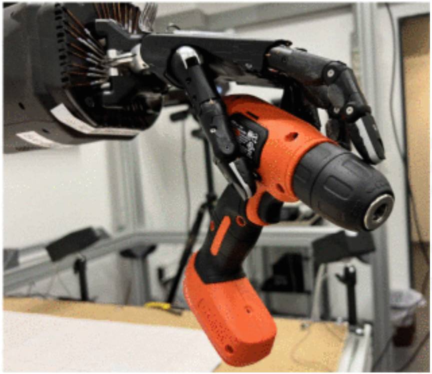


Reinforcement Learning



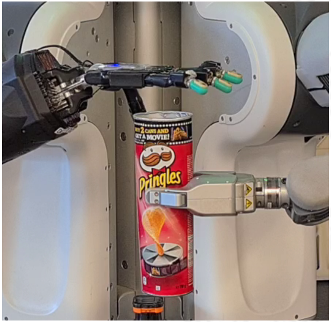
Reinforcement Learning Motivation

- Popular grasping method, also used by Liang et al.
- Maybe generalize/quickly adapt to new objects
- No hard coded grasps for individual objects
- Able to adapt to object movement during handover



Reinforcement Learning Implementation

- Training is done only on the real robot
- The goal is to have the shadow hand grasp the currently held object
- Each step adds a predicted synergy step to the current joint state
- Episode stop after all fingers have contact or step limit



Reinforcement Learning Formulation

State Space:

$s_t = \{pca_{3t}, eff_t, oh\}$

pca_{3t} : first three synergy values of joint state at time t
eff_t : effort values of closing joints at time t
closing joints : joints 2,3 for fingers and joint 5 for thumb
oh : one-hot encoding of three objects

Action Space:

$a_t = \{pca_3\}$

pca_3 : first three synergy values by which to change joint values

Reward:

$$r_t = \begin{cases} r_b + r_{con}, & \text{if } t = T_{final} \\ r_c, & \text{otherwise} \end{cases}$$

r_b : binary reward $\{-1, 1\}$ depending if grasp successful
r_c : closing reward, sum of change in closing joints
r_{con} : $0.1 \times$ number of finger contacts

Reinforcement Learning Formulation

State Space:

$s_t = \{pca_{3t}, eff_t, oh\}$

pca_{3t} : first three synergy values of joint state at time t

eff_t : effort values of closing joints at time t

closing joints : joints 2,3 for fingers and joint 5 for thumb

oh : one-hot encoding of three objects

Action Space:

$a_t = \{pca_3\}$

pca_3 : first three synergy values by which to change joint values

Reward:

$$r_t = \begin{cases} r_b + r_{con}, & \text{if } t = T_{final} \\ r_c, & \text{otherwise} \end{cases}$$

r_b : binary reward $\{-1, 1\}$ depending if grasp successful

r_c : closing reward, sum of change in closing joints

r_{con} : $0.1 \times$ number of finger contacts

Reinforcement Learning Formulation

State Space:

$s_t = \{pca_{3t}, eff_t, oh\}$

pca_{3t} : first three synergy values of joint state at time t

eff_t : effort values of closing joints at time t

closing joints : joints 2,3 for fingers and joint 5 for thumb

oh : one-hot encoding of three objects

Action Space:

$a_t = \{pca_3\}$

pca_3 : first three synergy values by which to change joint values

Reward:

$$r_t = \begin{cases} r_b + r_{con}, & \text{if } t = T_{final} \\ r_c, & \text{otherwise} \end{cases}$$

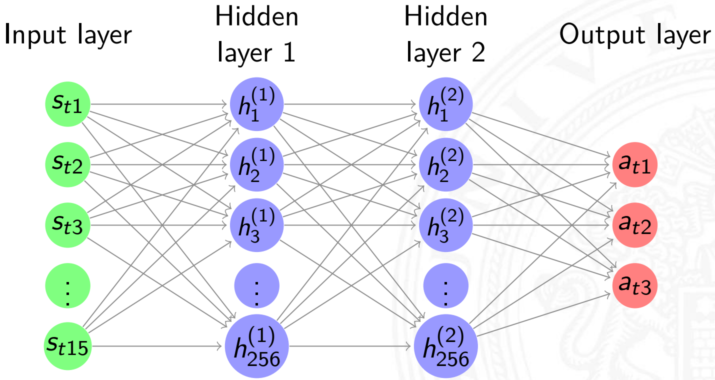
r_b : binary reward $\{-1, 1\}$ depending if grasp successful

r_c : closing reward, sum of change in closing joints

r_{con} : $0.1 \times$ number of finger contacts

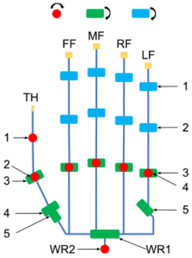
Reinforcement Learning Training Overview

- Default network structure from stable baselines 3
- Currently training for 10000 steps
- Object change after 1000 steps
- SAC as RL algorithm



Reinforcement Learning Restrictions

- Training is done on the robot instead of simulation
- Finger joints 4 fixed to limit self-collisions during training
- Wrist joints don't get moved
- Thumb joint 4 remains at the initial configuration
- Change normalized to largest joint change maximum 9 degree



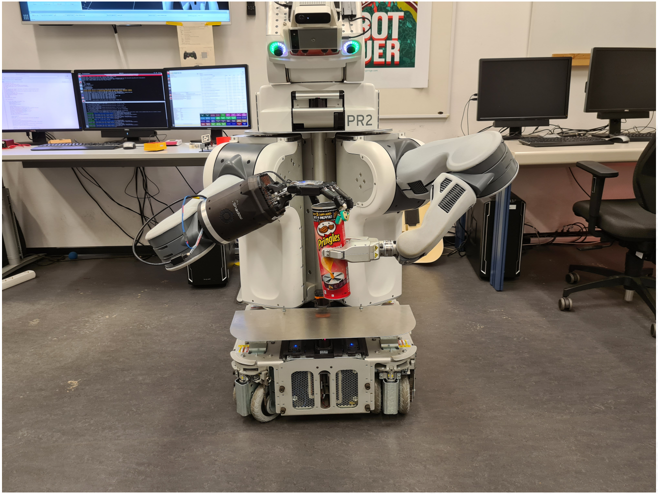
Reinforcement Learning Training

- Trained with effort feedback and two objects
- Learned to grasp can but failed with book
- Only fixed handover and grasped pose
- Showed validity of effort values as input but needs parameter adjustments

Training Video Effort

Full handover

Video full handover



Interim Findings

Accomplishments:

- Implemented a bimanual object handover pipeline
- Managed to train a grasping model using only the real robot
- Analyzed the bimanual workspace of the PR2 regarding object handover

Limitations:

- Limited to one object
- Still uses hard-coded poses for grasp pose
- Training not yet done with sampled handover poses
- Requires further evaluation of chosen parameters

Ongoing Work

Structure:

- Increase to multiple (YCB) objects
- Implement grasp point generation
- Investigate the possibility of a second object handover to the gripper

Evaluation:

- Investigate different state space configurations
- Evaluate success rate
- (Investigate different cost functions)

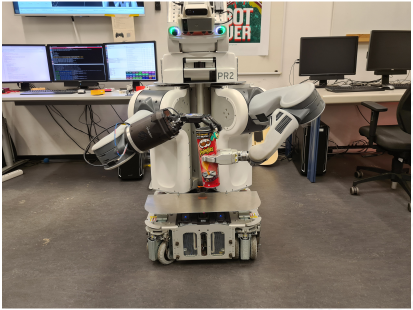
Bibliography

Li, Yunfei et al. "Efficient Bimanual Handover and Rearrangement via Symmetry-Aware Actor-Critic Learning". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 3867–3874. DOI: 10.1109/ICRA48891.2023.10160739.

Liang, Hongzhuo et al. "Multifingered Grasping Based on Multimodal Reinforcement Learning". In: *IEEE Robotics and Automation Letters (RA-L)* 7.2 (2022), pp. 1174–1181. DOI: 10.1109/LRA.2021.3138545.

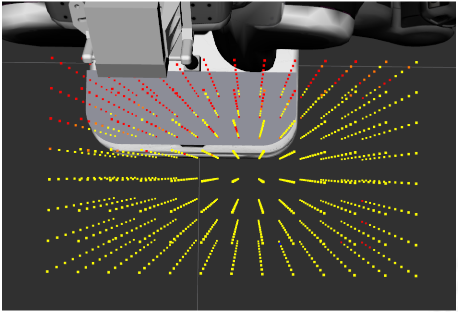
Pavlichenko, Dmytro et al. "Autonomous Bimanual Functional Regrasping of Novel Object Class Instances". In: *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. 2019, pp. 351–358. DOI: 10.1109/Humanoids43949.2019.9035030.

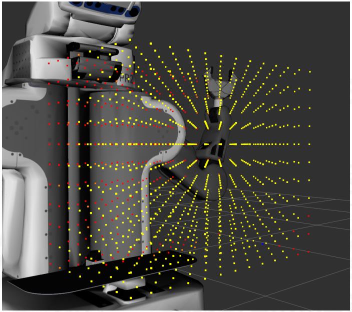
Thank You For Listening!



Any questions or feedback are very welcome.

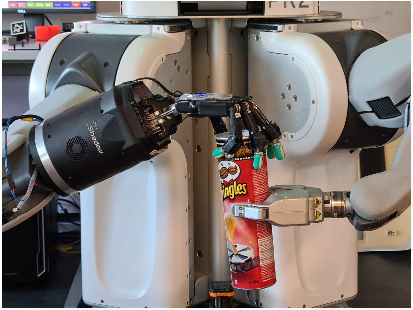
Workspace Analysis Book



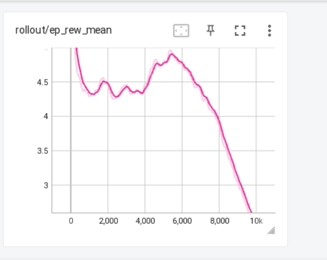


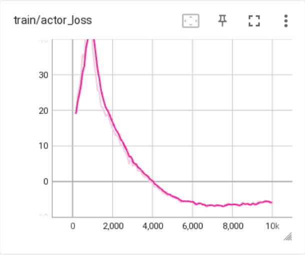
Reinforcement Learning Initial Version

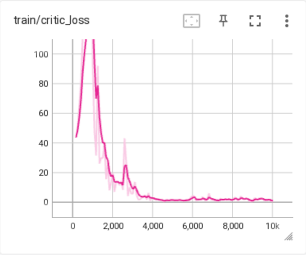
- Initially trained with biotac sensor feedback
- Can object could be grasped reliably
- Only one object and one pose
- Initial indicator for validity of approach



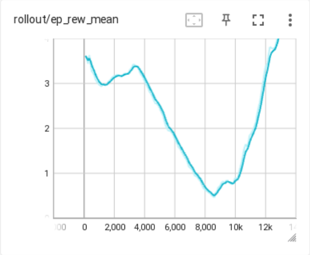
Effort Training Graphs

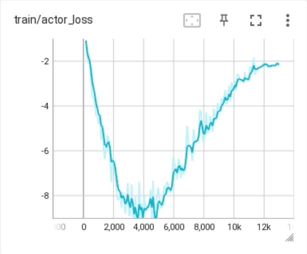


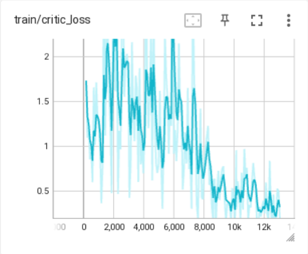




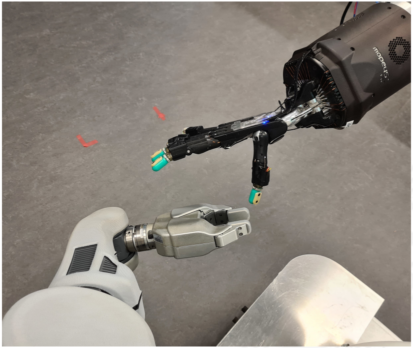
Tactile Training Graphs







Handover Sample



GPD Initial Test