Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Compliant Manipulation using Reinforcement Learning guided by Task Specification

Master Thesis

February 5, 2020

Abhishek Padalkar

Motivation

- Compliant manipulation is a must to have ability for collaborative and service robots.
- Robots need to deal with:
  - Inaccuracy in the geometric model of the world
  - Contact forces
  - External forces due to collision and other agents
- Limitations of model-based approaches:
  - Lack of precise model of contact forces
  - High computational complexity
- Limitations of learning based solutions:
  - Reinforcement learning requires high number of interactions.
  - Accuracy of simulation affects the learned behavior and transferability of solution.

Figure 1: Mother teaching child to cut vegetables.

- Humans learn multiple tasks through demonstrations and descriptions.
- Vital knowledge about motion is extracted and utilized.
- The task is then perfected with experience.
- Similarly, a task can be described or specified to a robot.

Problem Statement

Develop a learning based approach for achieving compliant manipulation that allows to exploit partial/incomplete models of the task and the environment provided by the task specification.

Two sub-problems:

- Model the easy-to-model knowledge about tasks using task specification framework.
- Use Reinforcement Learning for learning gradually more complete and robust action models for compliant manipulation tasks.

We demonstrate our approach on two demo tasks:

- Opening door, and
- Cutting vegetables.

Related Work

Task Specification Methods

- Industrial task specification languages like KUKA Robot Language
  - Reliable
  - Difficult to integrate sensory feedbacks
  - Limited expressive power [4]
- Task Frame Formalism specifies the task in task frame or compliance frame, by defining motion or force in each DoF[6, 5].
- Leidner presents symbolic representation of the task in PDDL along with the geometric representation of the task using sequence of low level movement sequences.
- iTaSC synthesizes control inputs by solving an optimization problem considering the task space constraints [1, 2, 3].

(1/2)
Reinforcement Learning

- Model-based reinforcement learning learns the model of environment as well as policy for performing the task.
- Number of interactions required can be very high for learning model of the environment.
- Model-free reinforcement learning learns value functions without learning the model of the environment.
- Policy search methods are model-free learning methods which directly learn policy to take action in current observable state without learning value function.

(2/2)

Reinforcement Learning

- Policy search methods are used heavily in robot control due to their ability to handle high dimensional state-action space.
- Algorithms:
  - Finite difference method : Learns by comparing returns with baseline after exploration in parameter space.
  - Likelihood ratio method / REINFORCE : Uses policy gradient theorem
  - Natural actor critic algorithm
  - Path integral policy improvement (PI^2)
  - Expectation-maximization method (PoWER) : Learns by exploration in the parameter space and by weighing the exploration by returns

Door opening task

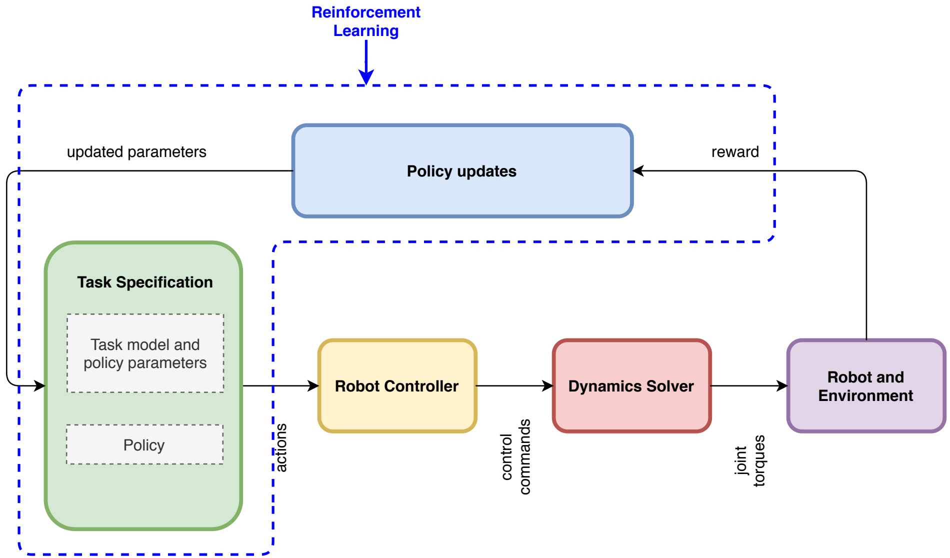
- Motion synthesis using the geometric model of the door : requires accurate model of door
- Adaptive control along with on-line estimation of the door parameters using force-torque sensors : sensor noise affects the parameter estimation
- Learning force policies using reinforcement learning : requires numerous interactions with environment
- Learning the task from human demonstrations using Dynamic Motion Primitives : not generalizable for different doors

Vegetable cutting task

- Deep recurrent neural network with model predictive control : requires large amount of data to learn the model
- Using sequence of Dynamic Motion Primitives : purely motion-based approach

Solution

- Provide task specification using Task Frame Formalism (TFF), which also contains a policy to be learned by reinforcement leaning for generating control commands.
- Design policy for control command generation in the task specification. This policy can be a engineered policy for a well understood problem or can be a generic parameterized policy e.g. neural network or radial basis function network.
- Design a reward function for the given task.
- Use existing reinforcement learning methods to learn parameters of the policy to maximize given reward function by carrying out trials in the environment.


Reinforcement Learning
updated parameters
Policy updates
reward
Task Specification
Task model and policy parameters
Policy
actions
Robot Controller
control commands
Dynamics Solver
joint torques
Robot and Environment

Task Specification using Task frame formalism

Task frame formalism provides guidelines for identifying reciprocal degrees of freedom and task specification using them.

The frame is chosen such that the motion does not generate power against reaction forces.

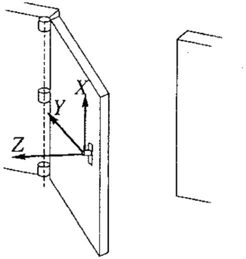


Figure 2: Task frame selection for door opening task

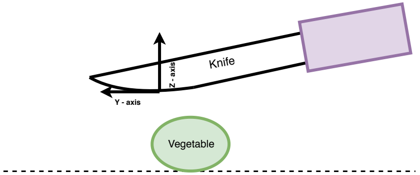


Figure 3: Task frame selection for vegetable cutting task

Task specification for opening door:

Listing 1: Task Specification using TFF: Open Door

```
    move compliantly {
    with task frame directions
    xt: force 0 N
    yt: force 0 N
    zt: velocity v m/sec
    axt: force 0 Nm
    ayt: force 0 Nm
    azt: force 0 Nm
  }until chord length > d m or force z > f N
```

Task specification for cutting vegetables:

Listing 2: Cut vegetables

```
    move compliantly {
    with task frame directions
    xt: velocity 0 m/s
    yt: velocity v(t) m/s
    zt: force f(environment, robot, task) N
    axt: velocity 0 rad/s
    ayt: velocity 0 rad/s
    azt: velocity 0 rad/s
  }until distance y > d m
```

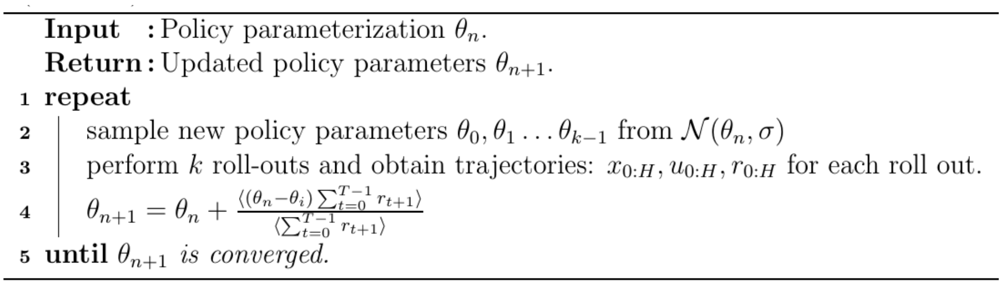
Reinforcement Learning Algorithms

REINFORCE

```
Input   : Policy parameterization $\theta_h$.
Return  : gradient estimate $g_{FD} = [g_1, \dots, g_h]$.
1 repeat
2     perform a trail and obtain $x_{0:H}, u_{0:H}, r_{0:H}$.
3     foreach gradient element $g_h$ do
4         estimate optimal baseline $b^h = \frac{\langle (\sum_{k=0}^{H} \nabla_{\theta_h} \log \pi_\theta(\vec{u}_k|\vec{x}_k))^2 \sum_{l=0}^{H} a_l r_l \rangle}{\langle (\sum_{k=0}^{H} \nabla_{\theta_h} \log \pi_\theta(\vec{u}_k|\vec{x}_k)) \rangle}$
5         estimate the gradient element
          $g_h = \langle (\sum_{k=0}^{H} \nabla_{\theta_h} \log \pi_\theta(\vec{u}_k|\vec{x}_k))(\sum_{l=0}^{H} a_l r_l - b^h) \rangle$
6 until gradient estimate $g_{FD} = [g_1, \dots, g_h]$ converged.
```

- Uses policy gradient theorem to estimate the gradient of the parameters.
- Adds exploration noise in the actions at every discrete time step.

Policy learning by Weighting Exploration with the Returns (PoWER)



```
Input  : Policy parameterization $\theta_n$.
Return : Updated policy parameters $\theta_{n+1}$.
1 repeat
2     sample new policy parameters $\theta_0, \theta_1 \dots \theta_{k-1}$ from $\mathcal{N}(\theta_n, \sigma)$
3     perform $k$ roll-outs and obtain trajectories: $x_{0:H}, u_{0:H}, r_{0:H}$ for each roll out.
4     $\theta_{n+1} = \theta_n + \frac{\langle (\theta_n - \theta_i) \sum_{t=0}^{T-1} r_{t+1} \rangle}{\langle \sum_{t=0}^{T-1} r_{t+1} \rangle}$
5 until $\theta_{n+1}$ is converged.
```

- Uses expectation-maximization method to estimate the gradient of the parameters.
- Adds exploration noise in the parameters at the beginning of the episode.

Policy Representation and Reward Function

- Linear policy

$$f = -(A\dot{y} + B(0.5^2 - (0.5 - \phi_y)^2) + C(1 - \phi_z)) \tag{1}$$

- Gaussian policy

$$f = -(A\dot{y} + B(0.5^2 - (0.5 - \phi_y^2)) + \sum_{i=0}^{N-1} W\psi_i(\phi_z)) \tag{2}$$

$$\psi_i(\phi_z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(c_i - \phi_z)}{2\sigma^2}} \tag{3}$$

- Where reward function used is :

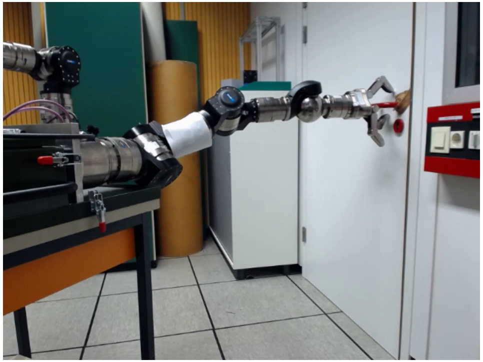
$$r = C_1 * \phi_z^2 - C_2 * f_z^2 \tag{4}$$

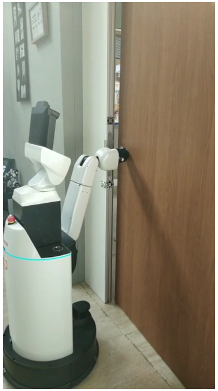
where A, B, C, and W (weight vector) are policy parameters, N is number of Gaussian functions, c_i is the center of i^{th} Gaussian, σ is width of Gaussian functions, ϕ_y and ϕ_z are cutting phases in Y and Z direction respectively.

Experimental Results: Door opening task

Experiments were performed with two robots:

- Schunk LWA4D equipped with Schunk FTM115 force torque module and Schunk Dexterous Hand
- Toyota Human Service Robot (HSR)





Schunk LWA4D opening unlatched door

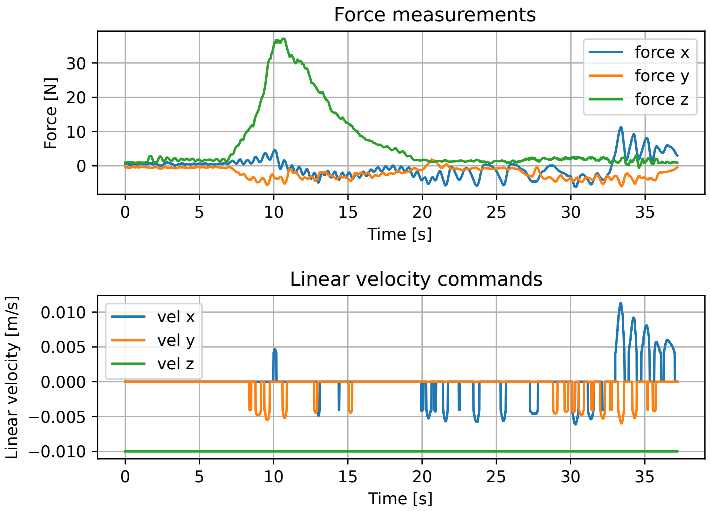


Figure 4: Force readings at TCP and linear velocity applied to TCP

Schunk LWA4D opening unlatched door

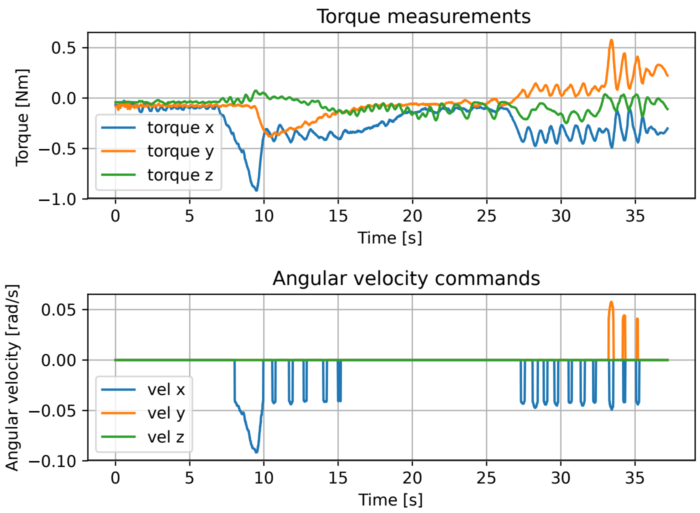


Figure 5: Torque readings at TCP and velocity angular velocity applied at TCP

Schunk LWA4D opening unlatched door

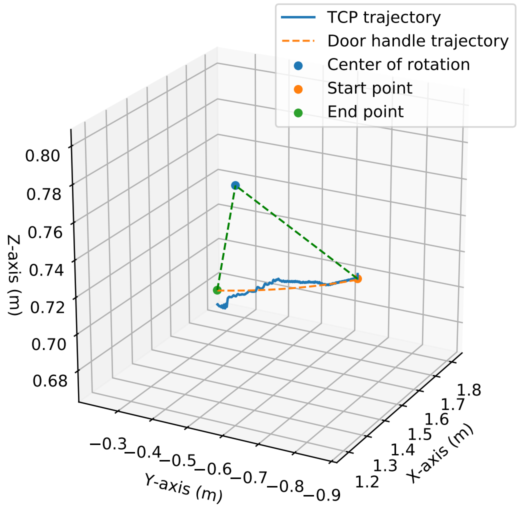


Figure 6: Geometric analysis of trajectory traced by TCP

Toyota HSR trying to open jammed door

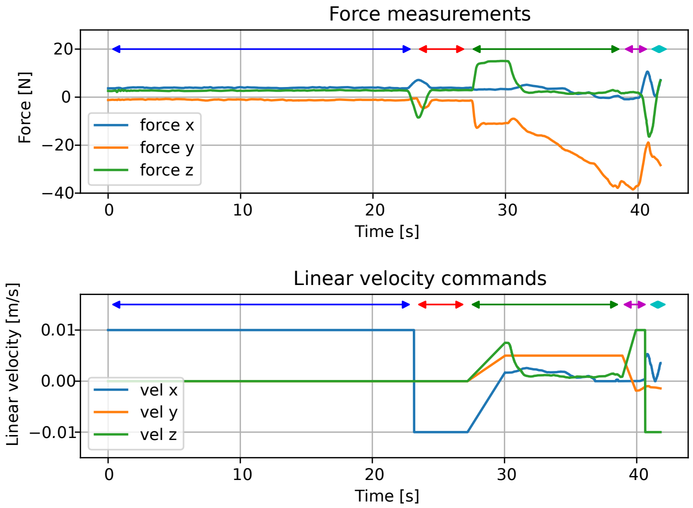


Figure 7: Force readings at TCP and linear velocity applied to TCP

Toyota HSR trying to open jammed door

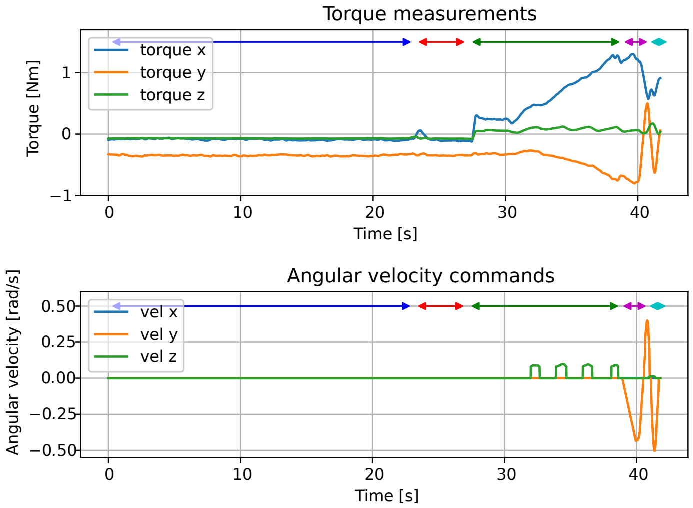


Figure 8: Torque readings at TCP and velocity angular velocity applied at TCP

Toyota HSR trying to open jammed door

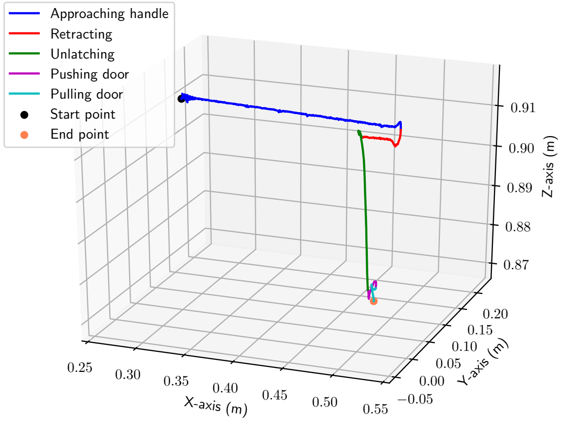


Figure 9: Geometric analysis of trajectory traced by TCP

Discussion

- Schunk LWA4D was able to open the door every time in 27 trials.
- Robot was able to open the door in the presence of a bump on the floor.
- Toyota HSR robot was successful in detecting the opening direction of door and in determining whether the door is jammed or not.
- The chord length traced while opening door varies because :
  - One or more joint limits reached
  - Operation was stopped to avoid self-collision
- Oscillations were observed in the trajectory because of :
  - Drift in the sensor readings over time
  - Delay in motion command update

Vegetable Cutting Experiment

Testing algorithm in simulation

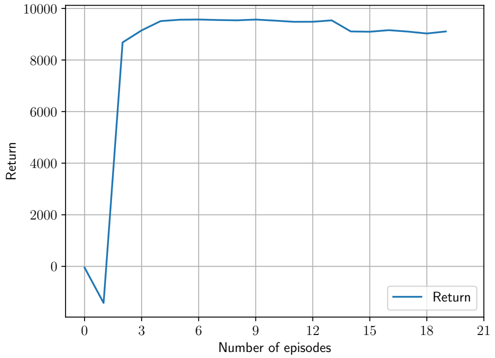


Figure 10: Performance of REINFORCE algorithm in simulation

- Vegetable cutting task was simulated as motion of object through a viscous medium.
- REINFORCE algorithm successfully learned cutting policy.

Experiments with KUKA LWR 4+

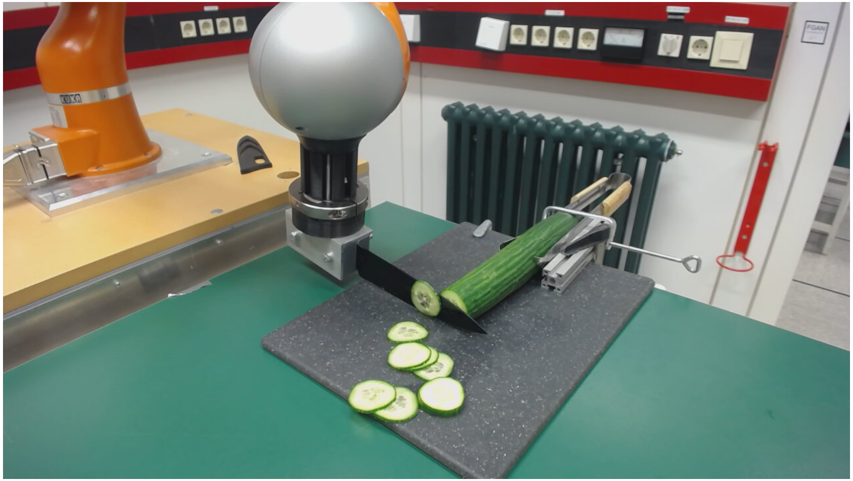


Figure 11: KUKA LWR 4+ cutting vegetable

Learning Force Policy with REINFOCE Algorithm

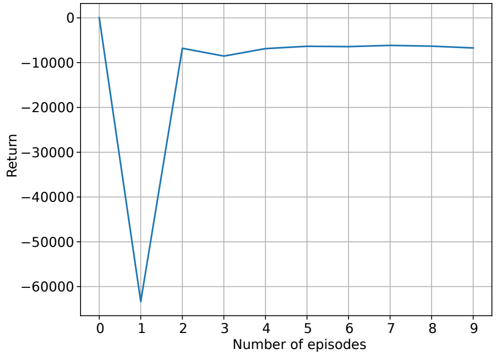


Figure 12: Performance of REINFORCE algorithm with KUKA LWR 4+

REINFORCE algorithm fails to learn force policy:

- RINFORCE uses high frequency noise for exploration in action space.
- The task and robot dynamics acts as a low-pass filter.
- Contact dynamics also acts as low pass filter.
- Hence noise is damped and REINFORCE does not converge.

Learning Force Policy with PoWER Algorithm

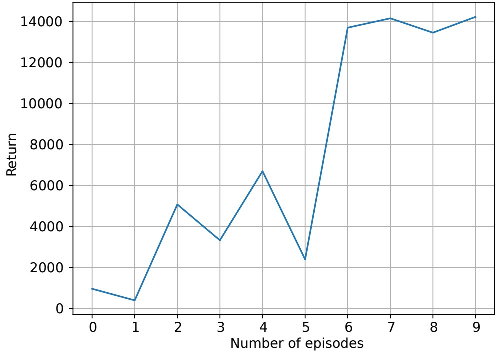


Figure 13: Performance of PoWER algorithm with KUKA LWR 4+ using linear policy

PoWER algorithm successfully learns force policy:

- Policy was learned after 6 episodes.
- Effective exploration was possible because of low frequency noise.

Cutting cucumber with generated policy

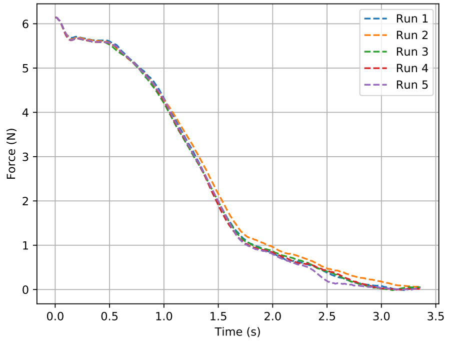


Figure 14: Force generated by linear policy while cutting cucumber

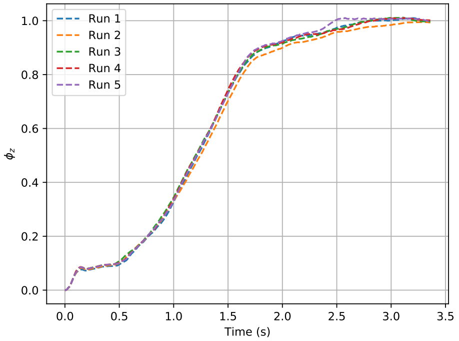


Figure 15: Cutting phase ϕ_z in Z-direction

Learning Force Policy with PoWER Algorithm

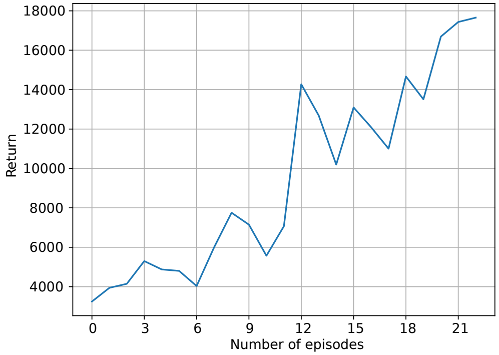


Figure 16: Performance of PoWER algorithm with KUKA LWR 4+ using linear policy

PoWER algorithm successfully learns force policy:

- Policy was learned after 20 episodes
- Increase in number parameters requires more trials to learn the policy.

Learning Force Policy with PoWER Algorithm

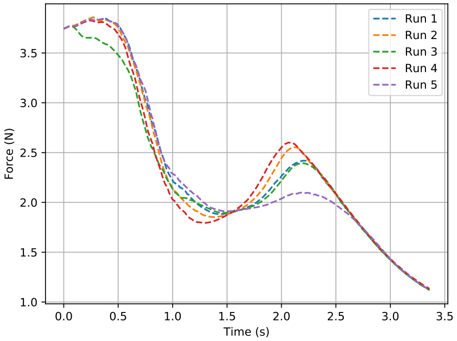


Figure 17: Force generated by Gaussian policy while cutting vegetable

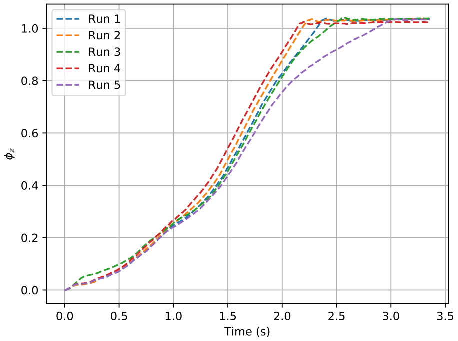
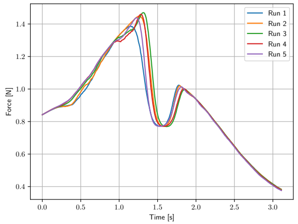
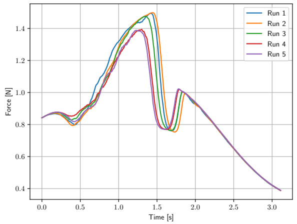


Figure 18: Cutting phase ϕ_z in Z-direction

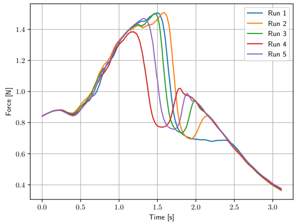
Learning Force Policy with PoWER Algorithm



(a) 1 day old

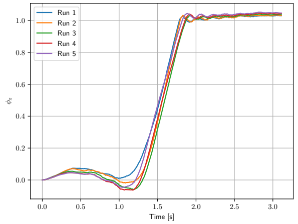


(b) 2 days old

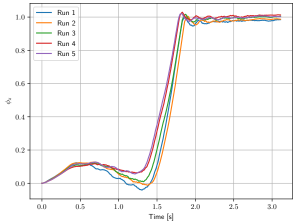


(c) 3 days old

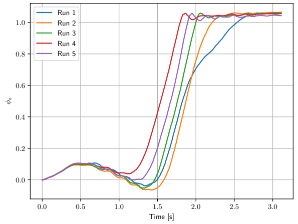
Figure 19: Force applied at TCP in Z-direction for cutting bananas with different ripeness.



(a) 1 days old



(b) 2 days old



(c) 3 days old

Figure 20: Phase ϕ_z of the cutting motion in Z-direction for cutting bananas with different ripeness.

Discussion

- REINFORCE algorithm failed to learn force policy for cutting vegetables because of ineffective exploration with high frequency noise in low-pass filter like environment.
- Increase in policy parameters resulted in increase in number of trials required for learning.
- It was also observed that the cutting depth achieved in previous trial affected the return of the next trial due to more resistance provided by vegetable.
- Different noise variance should be used for different parameters.

Conclusions

- Generalization ability of door opening approach - tested on two different robots.
- Extension for detecting the opening direction of the door and if the door is jammed.
- Oscillations in the end-effector trajectory while opening door due to sensor noise and low command update rate.
- Force policies for cutting vegetable were learned using PoWER algorithm with linear and Gaussian policy representation.
- Task specification framework reduced the dimensions of the problem from six to one, which resulted in faster learning directly on the robot without using simulation.
- Simple policies can be very effective when one is interested in learning to perform the task rather than learning the complex dynamics of the environment in which the task is being performed.
- The choice of the reinforcement learning algorithm is affected by the task dynamics.

Future Work

- Auto-tuning methods for tuning of admittance controller parameters.
- Extending the approach for tasks such as sweeping the floor, cutting curtains and bags during an investigation of the potentially dangerous environment.
- Extending for tasks involving multi-directional force interactions with the environment such as collaborating with human to transport objects from one place to another.

References (1/3)

J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx.
Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty.
The International Journal of Robotics Research, 26(5), 2007.

W. Decré, , H. Bruyninckx, and J. De Schutter.
Extending the Itasc constraint-based robot task specification framework to time- independent trajectories and user-configurable task horizons.
In Proceedings of the IEEE International Conference on Robotics and Automation, 2013.

References (2/3)

W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter.
Extending iTaSC to support inequality constraints and non-instantaneous task specification.
In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009.

D. S. Leidner.
Cognitive Reasoning for Compliant Robot Manipulation.
PhD thesis, Universität Bremen, 2017.

M. T. Mason.
Compliance and force control for computer controlled manipulators.
IEEE Transactions on Systems, Man, and Cybernetics, 11(6), 1981.

References (3/3)

F. Nägele, L. Halt, P. Tenbrock, and A. Pott.
A prototype-based skill model for specifying robotic assembly tasks.
In 2018 IEEE International Conference on Robotics and Automation (ICRA).
IEEE, 2018.