Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen
International Center for
Information Technology

Master Thesis Proposal

Compliant Manipulation with Task Specification and Reinforcement Learning

Abhishek Padalkar

Supervised by

Prof. Dr. Paul G. Plöger
Second Supervisor
Sven Schneider

April 24, 2019

1. Introduction

Most of the real world robotic manipulation tasks present the need for compliant manipulation, where robot needs to respond to the contact forces while executing the task. Classical planning and control algorithms fail to perform satisfactorily due to the lack of precise models of contact forces and high computational complexity. Various approaches have been proposed to learn compliant manipulation skills with help of Reinforcement Learning (RL). RL, in theory, is very promising when it comes to learn intelligent behaviors in complex environment with high dimensions. But it requires high number of interactions with environment. In model free reinforcement learning, a agent learns the skills by exploring the environment and adopting the parameters which governs the trajectory of the agent in the environment. Learning all the parameters of the policy can be computationally very expensive and might require large number of interactions with the environment. Application of RL to robotics is limited by above reason because large number trials causes wear and tear in mechanical parts and even damage to robot and the environment apart from being time consuming. Model based RL provide the solution to the problem of required high number of interactions with the environment by modeling some part of environment and motions policy.

On the other hand, task specification approaches, proposed in [1, 7, 10] rely on instantaneous task specification in task frame where constraint on motion are specified in each direction. Such approaches are more practical because of their deterministic nature which can be used for ensuring safe robot operation. But some parameters used in task specification need to be tuned manually which is tedious task and require number of human interventions. Moreover, sometimes, the task is too complex to be specified fully using task specification.

Nemec et al. in [13] propose a control algorithm by joining reinforcement learning with intelligent control which learns force policy to open door on top of the specified motion for unlatching and opening door operation. We propose to extend above mentioned work by using task specification approach and using reinforcement learning for searching parameters good enough for executing the task maximizing given reward function. Our approach combines the features from model based manipulation solutions and reinforcement learning. We propose to bring together the feature of deterministic solution from model based control solutions and self learning capability from reinforcement learning in one solution.

1.1. Problem Statement

- To solve the problem of compliant manipulation where complete model of the task,and environment is not available.
- We plan to solve above problem by dividing the problem into two subproblems:
  - Model the available knowledge about task and environment using existing task specification framework.
  - Use reinforcement learning for parameter search or policy search which account for the unavailable knowledge os the task and environment.
- We will evaluate our approach by learning a demo tasks of
  - Opening door,
  - Cutting vegetables.

2. Related Work

2.1. Task specification

In industrial settings, task are usually specified in the form of sequence of primitives [7]. These primitives mostly contain point-to-point motion, motion to some default configuration and basic curves. For example, following script taken from FRI interface examples shows the example motion primitives as shown in listing 2.1. Though this kind of task specification works in controlled industrial environment, it leaves very little scope for external sensory feedback and the task needs to specified in terms of positional goals.

Listing 2.1: KUKA Robot Language code

```
INI
PTP HOME ; go to HOME joint configuration
LIN P1   ; linear motion to point P1
LIN P2   ; linear motion to point P2
```

In [7], Leidner presented a representation in the form of action templates describing robot action using symbolic representations and geometric process model as shown in listing 2.2. Symbolic representation of the task in PDDL allows classical planners to consider the action in the high level abstract task plan and geometric representation of the task specifies the sequence of low level movement sequences needed to complete the action. Here, geometric representation only specifies discrete motion primitives and not the parametric representation of the motion primitives. Actual parameterization and control is left to low level control module.

Listing 2.2: Action Template: _object.pick

```
I Symbolic Representation

 , , ,
(:action _object.pick:
:parameters (?o - _object ?m - _manipulator ?s - _surface)
:precondition (and(free ?m) (on ?o ?s))
:effect (and(bound ?o ?m) (not(free ?m)) (not(on ?o ?s)))
)
'''

II Geometric Representation

def pick(self, manipulator, surface):
''' approach, grasp, and lift an object from a surface '''

graspset = odb.get_property(self.type, 'graspset',
   manipulator)
for grasp_candidate in graspset:
if grasp_candidate in self.history:
continue
self.history.append(candidate)
grasp = grasp_candidate
break

if grasp is None:
raise RuntimeError('no more alternatives')
else:
self.grasp = grasp
operations = [
('move_hand', manipulator, g.approach_grasp),
('plan_to_frame', manipulator, g.approach_frame, self.frame
   ), ('plan_to_frame', manipulator, g.grasp_frame, self.
   frame),
('bind', manipulator, self.name),
('move_hand', manipulator, g.pre_grasp),
('move_hand', manipulator, g.grasp),
('move_straight', manipulator, txyz(0, 0, 0.1))
]
return operations
```

In [10], Mason et. al. presented the idea behind *Task Frame Formalization - TFF* for representing compliant task. Using hybrid control, various control modes are assigned to each axis of the *task frame* or *compliance frame*[12]. Listing 2.3 shows Open Door task taken from [1]. This framework doesn't consider the specification of task or motion quality related parameters like velocity damping or instantaneous sensory inputs. It also specify action to be executed *compliantly* but does not specify the how?

Listing 2.3: Task Specification using TFF: Open Door

```
move compliantly {
    with task frame directions
    xt: force 0 N
    yt: force 0 N
    zt: velocity v mm/sec
    axt: force 0 Nmm
    ayt: force 0 Nmm
    azt: force 0 Nmm
} until distance > d mm
```

iTaSC developed in [2–4], synthesizes control inputs based on provided task space constraints. It formulates a optimization problem considering provided constraints in the environment. In case of conflicting constraints, constraints are weighted in the optimization problem.

3. Reinforcement Learning

Ideally, using reinforcement learning, an agent can find an optimal or near optimal behavior by interacting with the environment. Instead of providing a detailed engineered solution, a value function provides the feedback on the task which enables agent to learn a optimal solution. Based on use of prior knowledge, reinforcement learning approaches are categorized in two categories namely Model based and Model free reinforcement learning.

3.1. Model Free Reinforcement Learning

In model free reinforcement learning, no model of transition dynamics exist, hence optimal actions are derived by trial and error[15]. Learning all the parameters of the policy can be computationally very expensive and might require large number of interactions with the environment. Moreover, traditional RL methods utilize discrete state/action representation, which results into huge state-action space[13]. Application of model free reinforcement learning to robotics is limited by above reason, as large number of trials causes wear and tear in mechanical parts and even damage to robot and the environment apart from being time consuming.

3.2. Model Based Reinforcement Learning

In model based reinforcement learning, the model of transition dynamics exists. This transition model is used for deriving rewards and optimal actions. Optimal actions are derived using internal simulations reducing the number of interactions needed. This optimal policy is then executed in the environment. But this approach too heavily suffers from the accuracy of the transition dynamics model. Value function based RL methods learn the model by interacting with the envionment and then deriving the control policy using methods like *Differential Dynamic Programming, Temporal Difference Learning, Policy Iteration, Monte Carlo and Value Iteration*[15].

Some of the approaches implemented in robotics use reinforcement learning for learning direct control policy, rather than learning model and then deriving the policy[15]. Polydoros et al. give comprehensive survey on model based RL in [15].

Nemec et al. use reinforcement learning for learning force control policy for opening door[13]. They take into account the constraints on the door motion and cosed kinematic chain resulting from a firm grasp of robot hand on the door handle. While opening the door in such configuration, high internal forces are generated in the direction where motion is not possible. They use this knowledge to learn a compliant force control policy. Our approach is motivated by this work of Nemec et al.[13]. We try to formalize constraints in terms of already existing task specification methods, which can be generalized to more number of compliant manipulation tasks.

3.3. Door Opening Task

Opening door is one the most important task for service and rescue robots. Number of studies have been conducted to investigate this challenging problem using different techniques. Number of approaches rely on detail geometric modeling of the door and analytical trajectory generation. Nagatani et.al. [11] presented an early approach for opening door by modeling the geometry of the door and then synthesizing trajectory to open door by considering geometrical constraints. To cope with the errors in position control of the early days manipulator used in the experiment, they used a force torque sensor module to achieve compliance with the environment. But this approach highly relies on correct geometric model and accurate execution of designed trajectories.

Approaches presented in [6, 8, 14] use adaptive control algorithms which synthesize the motion by identifying the environmental constraints. In [6], authors proposed to find the radial direction based on force and torque readings. Door is opened by applying velocity in radial direction. This method is prone to failure in case of uncertain grasp position. Also it considers only the case where door is already unlatched.

Some of the methods for door opening motor primitives like Dynamic motion primitives for generating trajectories for opening door. Disadvantage of these approaches is that we need to know the goal pose to generate trajectory hence it confines the condition of completing the task to achieving a goal pose.

3.4. Vegetables Cutting Task

Vegetable cutting task in an interesting problem for robot manipulation considering the complexity of contact forces. Different vegetables requires different cutting forces. Also force profile changes as we cut through the vegetables, e.g. tomato requires much less force once we cut through the skin. Such a complicated force interactions are very hard to model and specified in the task specification. Reinforcement learning provides the solution by allowing robots to learn force control policy over time or phase of the cutting task. This task was chosen for demonstration because of its complex contact force profile. Moreover the task itself does not need any external sensors other than force torque sensors to for estimating rewards and completion of the task.

In [9], Lioutikov et al. solved the task by learning Dynamic Motion Primitives, which is a trajectory control policy. Authors does not consider the problem as compliant manipulation problem hence contact forces are not taken into account. Task is solved just by performing Cartesian trajectory, and hence requiring multiple attempts to cut through the vegetable.

4. Proposed Solution

We plan to solve a compliant manipulation task by providing a incomplete task specification and using Reinforcement Learning to determine un-modeled components in task specification. We propose to create a framework based on task specification and model based reinforcement learning to solve the problem of compliant manipulation for given task in deterministic manner with fewer number of interactions with environment.

4.1. Composition

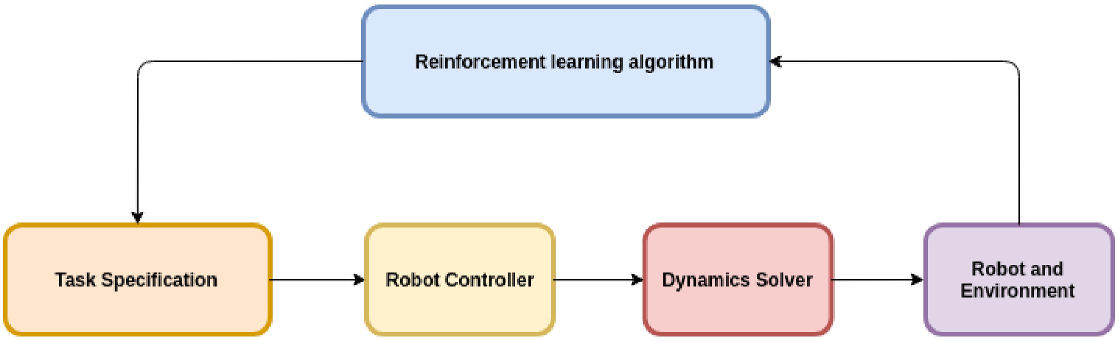

Figure 4.1: Composition

Above figure shows the composition of the proposed solution.

4.2. Task Specification

We propose to use approach given by Mason et al.[10], with further modification considering limitation on robot and environment. These parameters will also be used for ensuring quality of the task. e.g. smoothness in opening door. Also we introduce force and velocities as functions of robot and environmental parameters and phase of the task.

Examples of task specification:

Listing 4.1: Task specification for opening door

```
open_door
{
    with task frame directions
    xt: force 0 N
    yt: force 0 N
    zt: force f(environment, robot, task) N
    axt: force 0 Nmm
    ayt: force 0 Nmm
    azt: force 0 Nmm
    max_acc = a_max m/ss
    max_dec = a_dec m/ss
}until(g(...))
```

In above mentioned door opening task, task is represented in door frame where z-axis is perpendicular to door frame. Here, instead of specifying force in the z-direction, a constant velocity can also be specified for simplicity. But such specification can result in oscillations in case of self closing doors where initially huge amount force is required to move the door. It can be a huge problem especially when velocity control loop is running at low frequency. Parameters max_acc and max_dec ensure safe operation of the robot as well as quality of task.

In vegetable cutting task stated in Listing 4.2, task is represented in chopping board frame such that z-axis is perpendicular to the chopping board and x-axis is in cutting direction. In this task, velocity in x-axis and force in z-axis are not easy to be modeled or parameterized. Here both of these entities can be represented by and time parameterized policy. Moreover velocity can be learned from demonstrations using methods like Dynamic Motions Primitives[9].

Listing 4.2: Task specification for cutting vegetables

```
cut_vegetables
{
    with task frame directions
    xt: velocity v(t) m/s
    yt: force 0 N
    zt: force f(environment, robot, task, xt) N
    axt: force 0 Nmm
    ayt: force 0 Nmm
    azt: force 0 Nmm
    max_acc = a_max m/ss
    max_dec = a_dec m/ss
}until(g(...))
```

4.3. Reinforcement Learning Algorithm

As already discussed,

1. Task specification approaches are reliable but they need manual tuning of many parameters

2. Sometimes force interactions in compliant manipulation tasks are so complex that parameterization of task and force control policy is not possible.

Both of the above mentioned problems can be tackled by reinforcement learning. In first case, where the parameterization is possible, model based RL methods can be used for searching near optimal model parameters. Then policy can be derived by methods mentioned in section 3.2 and in [15]. It is even possible to specify a engineered policy which uses the learned parameters.

Tasks where policy parameterization is not possible, model based RL can be used directly to learn a generic parametrized policy by using methods like *Gradient-based* *approaches, Sampling based approaches, Information theory methods, Evolutionary computation, etc*[15].

In our case, we plan to represent control policy in terms of weighted Gaussian functions similar as in [13]. We plan to use Policy Improvement with Path Integral (PI^2) algorithm[17] for optimizing the policy parameters. (PI^2) is a sampling based algorithm which uses zero mean Gaussian noise to generate new parameters.

4.4. Robot Controller

We propose to use robot specific impedance controller for controlling the robot. For KUKA LWR robot this controller is already provided by the manufacturer which is stated by following equation taken from [16], an official documentation for KUKA Fast Research Interface.

$$\tau_{cmd} = J^T(K_c(x_{FRI} - x_{msr}) + D(d_c) + F_{FRI}) + f_{dynamics}(q, \dot{q}, \ddot{q}) \tag{4.1}$$

Where, τ_{cmd} is commanded joint torque, J is jacobian of manipulator, K_c is spring constant or stiffness constant, x_{FRI} is commanded Cartesian position, x_{msr} is current measured Cartesian position, d_c is Cartesian normalized damping parameter, F_{FRI} is commanded Cartesian force vector. The terms for the damping design $D(d_c)$ and the dynamic model $f_{dynamics}(q, \dot{q}, \ddot{q})$ are taken care of in the KUKA motion kernel.

For manipulators which don't have internal torque controllers, but are equipped with the wrist force-torque sensors, impedance control can be achieved by a linear PID feedback controller which in turn controls velocity of end effector to control force and torque at end effector. Here the feedback is force and torque measurements by wrist force torque sensor. Such controller can be given by equation below:

$$v = P \cdot f_e + I \cdot \int f_e + D \cdot \dot{f}_e \tag{4.2}$$

$$f_e = f_{desired} - f_{measured} \tag{4.3}$$

Where, v is velocity vector of end effector.

5. Project Plan

5.1. Work Packages and Milestones

Work Packages		**Targeted date**
WP01	Literature search	10-05-2019
WP02	Analysis of state of the art	
MS01	*Documentation of the state of the art*	
WP03	Modeling demo tasks using task specification approach	20-06-2019
WP04	Design and implementation of the robot controller	
MS02	*Implementing task specification approach for door opening*	
WP05	Integrating RL algorithm in task specification	30-07-2019
WP06	Design and implementation of RL algorithm	
MS03	*Implementing task specification approach with RL for door opening task*	
WP05	Extending implemented solution for vegetable cutting task	10-09-2019
WP06	Experimental evaluation of implemented algorithms	
MS03	*Implementing task specification approach with RL for cutting vegetable task*	

5.2. Project Schedule

5.3. Deliverables

5.3.1. Minimum Viable

- Literature survey and analysis of the state of the art.

- Design and implementation of task specification framework with reinforcement learning.
- Demonstration of task specification framework for door opening task without using reinforcement learning.

5.3.2. Expected

- Demonstration of task specification framework with reinforcement learning for door opening task.

5.3.3. Desired

- Extending proposed approach for cutting vegetable task.
- Demonstration of task specification framework with reinforcement learning for door opening task.

References

[1] Herman Bruyninckx and Joris De Schutter. Specification of force-controlled actions in the" task frame formalism"-a synthesis. *IEEE Transactions on robotics and Automation*, 12(4), 1996.

[2] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes, and Herman 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.

[3] Wilm Decré, Ruben Smits, Herman Bruyninckx, and Joris 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.

[4] Wilm Decré, , Herman Bruyninckx, and Joris 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.

[5] Mrinal Kalakrishnan, Ludovic Righetti, Peter Pastor, and Stefan Schaal. Learning force control policies for compliant manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011.

[6] Yiannis Karayiannidis, Christian Smith, Petter Ögren, and Danica Kragic. Adaptive force/velocity control for opening unknown doors1. *IFAC Proceedings Volumes*, 45(22), 2012.

[7] Daniel Sebastian Leidner. *Cognitive Reasoning for Compliant Robot Manipulation*. PhD thesis, Universität Bremen, 2017.

[8] Martin Levihn and Mike Stilman. Using environment objects as tools: Unconventional door opening. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.

[9] Rudolf Lioutikov, Oliver Kroemer, Guilherme Maeda, and Jan Peters. Learning manipulation by sequencing motor primitives with a two-armed robot. In *Intelligent Autonomous Systems 13*. Springer, 2016.

[10] Matthew T Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6), 1981.

[11] Keiji Nagatani and SI Yuta. An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2. IEEE, 1995.

[12] Frank Nägele, Lorenz Halt, Philipp Tenbrock, and Andreas Pott. A prototype-based skill model for specifying robotic assembly tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[13] Bojan Nemec, Leon Žlajpah, and Aleš Ude. Door opening by joining reinforcement learning and intelligent control. In *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017.

[14] Günter Niemeyer and J-JE Slotine. A simple strategy for opening an unknown door. In *Proceedings of International Conference on Robotics and Automation*, volume 2. IEEE, 1997.

[15] Athanasios S Polydoros and Lazaros Nalpantidis. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems*, 86(2), 2017.

[16] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*. Citeseer, 2010.

[17] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. Learning policy improvements with path integrals. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.