Research Plan

Manipulation of Articulated Objects: Perception and Control

Giuseppe Maria Rizzi

April, 2021

Doctoral Candidate

Supervisor

Autonomous Systems Lab
Institute of Robotics and Intelligent Systems
8092 Zurich

Abstract

Max. $^1/_2$ page abstract ...

1 Introduction

There is a growing interest in deploying autonomous systems in everyday life. Robots have the power to enhance productivity while relieving human labor from repetitive and life threatening operations. Picking objects from warehouse shelves [1], assistive service in the healthcare domain [2], agrifoods [3], cleaning up disaster sites [4], search and rescue [5], industrial inspection and maintenance [6] are highly demanded applications for robots. The need for more automation in risk-sensitive scenario and the readiness level of robotics research and development has motivated the EU Horizon 2020 Piloting project [7]. The project aims to successfully exploit ground and aerial unmanned vehicles for industrial maintenance and inspection. As part of the project's consortium, we would like to take advantage of this collaboration to inspire this research proposal.

All these scenario involve physical interaction between the autonomous agent and the surrounding environment. Oftentimes, this interaction requires manipulation of objects such as doors, buttons, handles, cranks and switches just to name a few. In particular, articulated objects pose additional challenges as their motion is constrained (e.g by revolute and prismatic joints). Consider the task of autonomously opening a door along an aisle. The robot should be able to perceive the door and its components, model its articulation, elaborate a manipulation plan, e.g. navigate to the door, grasp its handle and finally move the end effector such that the door is pulled open. Furthermore, the robot does not access to an exact knowledge of the environment. Instead, manipulation should be robust under sensing and modeling uncertainty. Yet, do humans count on perfect knowledge of the environment for interaction? We observe that we do not but we are still able to perform a myriad of complex interaction tasks. We must therefore deduce that environment knowledge is improved "on-the-fly" and we use the unconscious knowledge of uncertainty to perceive, plan and control as a whole. Referring to the previous example, what would a human do when trying to perform the same task in the dark? She would probably reach the visible wall next to the door and follow the surface until sensing a handle-shaped object. We need perception and control algorithms that actively take uncertainty into consideration in order to achieve robust manipulation capabilities under hard sensing conditions. We see that multiple *perception*, *modeling*, *planning and control* problems must be solved to achieve the desired manipulation goal. In order to enhance the synergism between perception, modeling, planning and control, we need a better understanding of each module and existing gaps. In the remaining of this section we briefly introduce each sub-topic and highlight its relationship to autonomous manipulation tasks.

Perception Perception systems are often developed whose representation of the world is not optimal for motion planning. In the last decade, perception algorithms have focused on tasks such as classification [8], semantic segmentation [9], generative modeling [10] and pose estimation [11]. Autonomous manipulation requires a denser and interaction-rich information which is not provided by pure passive observation. We need perception to provide a broad appreciation of the scene but also to provide high-resolution information, including knowledge of the contact locations and forces exchanged [12].

Modeling The geometry of the scene can be measured only at the accuracy allowed by the visual sensors and perception pipeline. The modeling of the environment and its physical properties can also be inaccurate. Consider the apparently easy task of turning the door handle. The required motion is a simple circle. The problem is, where exactly should the robot produce the circle? We can do our best to estimate the handle's position, but we will never get it exactly right [12]. Uncertainty is often treated as a metric in the state estimation pipeline rather than a variable to actively account for during control. So generally, control validation is performed with ground-truth information or the architecture is designed such that it complies with a small degree of uncertainty.

Planning and Control Interaction is often decoupled in a planning and control stages. The high level plan, e.g a trajectory of end effector poses or manipulator joints is often produced beforehand and cannot be changed reactively [13]. Instead we need to produce an interaction plan that can adaptively change in real-time because of unmodeled disturbances, sensing noise and environmental uncertainty. Furthermore, planning should take into account point-to-point, point-to-plane, continuous and discontinuous contacts. Nevertheless planning for all possible interactions becomes soon intractable and an analytical solution can be restricted to few simple cases and geometries (put some citation here). An autonomous agent should leverage a physical understanding of the scene.

A unified framework that can address all these challenges is yet to be developed. Additionally, the research is mostly focused on developing independently each module. We hypothesize that a joint effort is needed to advance the current state of the art. In particular, open questions are:

- How is the scene best represented in order to plan a manipulation task? For instance, should the representation be limited to objects' segmentation and poses rather then to a more expressive and dense information? What would this representation be and which sensor modalities should be used?

- How can modeling mismatch and uncertainty be embed in an autonomous manipulation framework? Can manipulation plans be found such that achieving a goal and obtaining more information can be optimally combined according to some criteria such as time, risk or effort?

- How to plan for interaction in a closed-loop fashion so to be reactive to environmental changes while taking into account modeling and perceptual uncertainty? Can planning fully leverage the physical understanding of the scene while staying computationally feasible?

In the course of this thesis we will try to answer these questions. In particular we aim to develop a integrated autonomous manipulation framework which is able to robustly address the problem of manipulating articulated objects from perceiving them to successfully achieving the desired objective through forceful interaction. In the following sections we are going to briefly review the related work while focusing on the aspects that are more interesting for this research proposal. Thereafter the approach is presented. The main objective is subdivided into simpler sub-tasks which are grounded on clear research hypotheses. The research plan is concluded with a list of proposed publications and the accompanying time plan forecast to achieve the intended goals.

2 Related Work

- *Which major works consider a similar context?*
- *Which works are addressing same problem?*
- *Why are these works insufficient (gaps)?*
- *Which works use a similar methodology?*

In this section we present a collection of recent works which are relevant to the problem of autonomous manipulation of articulated objects. Such manipulation tasks are often decomposed into a set of smaller sub-tasks which are solved independently and are unaware of each other. The door opening example already shows which components might be involved. An hypothetical implementation could read as follows. The perception pipeline outputs handle poses from an RGB-D video stream. Once the object pose is established, a manipulation plan is generated. This could be additionally decomposed into a grasping and end effector tracking stages. Usually, a stable handle grasp is seeked and then, a well engineered end effector trajectory is commanded. In this case, the trajectory could be the sequential composition of two arcs, one for unlocking the door mechanism and the other to pull the door open. A low-level end effector controller is then in charge of executing the trajectory by directly commanding the joint torques. As we see in this example, once the door (handle) has been perceived, the perception pipeline is unused. Furthermore a stable grasp assumption could easily fall under slipping and modeling uncertainty. How would then the system know about this failure mode? How would the plan (end effector trajectory) be changed reactively to cope with unforeseen collisions and slipping? While this approach has the advantage of decomposing the complex manipulation in simpler sub-tasks, the resulting framework is purely "open-loop": each module is called only once instead of repeatedly computing the best course of action. This implementation is often referred to as the *sense-plan-act* paradigm (figure 1).

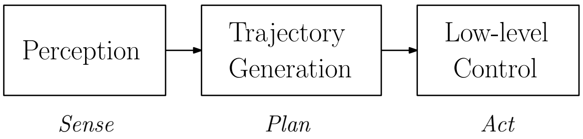

Figure 1: The *sense-plan-act* paradigm offers the unique advantage of breaking down the task complexity. On the other hand, information down the stream is never fed back to the sensing and planning modules.

In order to *close the loop* more advanced techniques are deemed. In recent years, the need for advancing platforms and algorithms for autonomous manipulation has motivated international robotic challenges such as the Amazon Picking Challenge [1] (APC) and MBZIRC International Competition [14]. The objective of the APC was to design an autonomous robots to pick up items from a warehouse shelf. In [2] it is reported that more then 80% of the 27 participating teams agreed with the statement *perception needs to be better integrated with motion planning.* More then half the participants also agreed with the statement *motion planning needs to be better integrated with reactive planning.* In the extensive and influential review [15] the authors claim that a need to bridge teh gap between perception and action in whole-body, multi-contact motion planning and controll is needed. This insightful feedback highlight the need for closing the loop and integrating more perception and planning. In the following we investigate which promising directions have been taken in the fields of perception, modeling, planning and control to address the aforementioned limitation. Furthermore, we envision that the recent technological advance in physical and photo-realistic rendering will play a decisive role in the development of new algorithms. We conclude this section with a brief overview on simulation engines and their potential as core technology for new solutions that span both perception and planning.

2.1 Perception

Perception algorithms are designed to convert raw sensory data (e.g RGBD images and point clouds) into a simpler representation of the scene which can be later used for planning. Ideally, one could directly plan in the space of RGB images [16]. Of course this would be extremely complex and is generally addressed by end to end reinforcement learning methods which either requires lots of training data and/or do not generalize well on unforeseen images. Therefore, one has to decide which representation better trades off complexity and representation power. Purely geometric representations of the object as poses [11], parts [17] or pixel-wise semantic segmentation [18], provide information about *where* and *what* but not *how*, namely how the object can be interacted with, its function as a part of the whole. Geometrical properties convey just a piece of information required to manipulate an object. We would like a manipulation-oriented perception system which answers the following questions:

- At which point/part does the interaction happen?
- What is the function of that point/part?
- Is the point/part/object movable? If so, how does it move?

Not all questions can be exactly answered by remotely perceiving a static scene. Nevertheless, a perception system could infer a reasonable prior over these properties. For instance, assume that we are looking at a slightly open door which we want to pull completely open. Grasping the handle is only one of the options to successfully executing the task. When we perceive the door, we perceive all the reachable door edges as *actionable points*, namely points we can interact with to successfully reach the desired goal. The perception system is *physically aware*, knows the *physical semantic* of each point on object of interest. This representation has potentially greater within-category invariance. In fact, all doors will share edges and handle shaped links that can be used to pull the door open, independently from their actual visual appearance. The formalization of the above concept goes under the name of *affordances* and was first introduced by the American psychologist James J. Gibson in [19]. In this seminal work, *affordances* are defined as what the environment *offers* the animal. The hypothesis of affordances can be summarized as follows [19]:

> *...what we perceive when we look at objects are their affordances, not their qualities; what an object affords us is what we pay attention to while the special combination of qualities into which an object can be analyzed is ordinarily not noticed.*

This idea has been applied in a number of recent works as an alternative object representation. In [20] the authors develop a perception algorithm which detects semantic 3d-keypoints as the object's representation. Although being a step forward in affordance-based perception, this representation lacks flexibility since it relies on a educated guess (annotation) of interaction hotspots and orientations. Part-based representation allows for a wide generalization across different types of objects as well. In [21] the proposed approach is able to predict that the bottom of the mug is useful for pounding, or the edge of a turner can used for cutting. A denser representation [22, 23] could be more suited to the task as opposed to lumped positions and orientations. For instance, the autonomous agent might be impaired from reaching the handle while it could easily interact with the door edges. Then it could exploit the knowledge that every reachable point on the edge is interactable in order to find the best interaction candidate. The aforementioned methods provide a dense mapping of affordances but miss the fundamental link with control or collapse the affordance map to poses or single points which are tracked by some off-the-shelf control algorithm. We then identify the following unanswered questions:

- What is the best affordance-based representation for closed-loop manipulation control?
- How can we directly plan in the space of affordances?

2.2 Modeling

Once an interaction is engaged, the robot needs to determine the outcome of its actions and how they change the environment. In the context of articulated objects this means that the it needs to estimate the object articulation mechanism, implemented as a kinematic constraint. Kinematic constraints are generally represented using different articulation models, e.g. revolute or prismatic. Many recent works have focused on extracting the articulation model category and its parameters from a static scene [24, 17]. Passively perceiving the environment can provide good prior knowledge for modeling the scene, however the accuracy of such model is limited by the sensor capabilities and can be deteriorated by sensor noise, occlusions, limited field of view and intrinsic ambiguities. While precise modeling is required to successfully act in the environment, acting can provide even more precise information about the environment itself. The latter approach is referred to as *active perception.* Following the definition in [25]:

> *An agent is an active perceiver if it knows why it wishes to sense, and then chooses what to perceive, and determines how, when and where to achieve that perception.*

As an example, an information gain criteria is often used to steer perception toward more informative area (citations). When the policy considers interactions and explicitly uses them to gain more information, then we talk about *interactive perception* [15, 26]. For example, the robot may not know whether two objects are rigidly connected or simply in contact; interactive perception allows to test each hypothesis by applying forces on the scene and observing the resulting object configuration [27].

Interactive perception often relies on priors in order to simplify the estimation problem. In fact, computing a posterior without any prior on the object category (deformable vs rigid) and articulation model (revolute vs prismatic) is instractable. Instead most systems rely on some previous knowledge. Often, the perception system is able to provide a good prior about the articulation model [28] and we know a priori that the object of interest is articulated and not deformable. Most approaches impose some structure on the type of the interaction. They often rely on a rigid grasp model [29] and a set of action primitives such as pull, push, grasp which are assumed to be executed without failure [30]. An implicit assumption that is made in this works consists in the access to nicely shaped handles that generally stand out from the object surface. It is not clear how these approaches would work for an embedded handle as the one shown in figure 2

During interactive perception, multiple sensor modalities can be combined: the already mentioned RGBD and time of flight sensors and also haptic devices such as force-torque cells. How to combine these modalities optimally is still an open question. In fact, more sensors introduce also more noise and complexity. Furthermore, we can only partially observe the state of the world. A reasonable question is then, how can we cope with uncertainty? If we maintain a full distribution over the quantity of interest, then computing an action that takes uncertainty into account is usually intractable [31].In general, we are not interested to identify the model parameters per se but rather to achieve the manipulation goal. A better estimate of

Figure 2: Traditional household furniture has handles which do not always allow to easily close the grasp on both sides. In this cases, it is harder to ensure a fixed grasp and planning interaction results consequently more complex.

the model and its parameters is then functional as long as the objective is executed better and faster.

2.3 Planning

Manipulating an object involves a wide range of motions that is hard to achieve with fixed base manipulators. For this reason, they are often combined with a mobile base for greater mobility and unconstrained workspace. As an example, the rotation range of the door could be too big to be handled by a fixed base arm while fulfilling joint limits constraints. However, to fully exploit these capabilities, systems require planning algorithms that can generate fast, accurate and coordinated reactive whole-body motions that account for multiple potential contacts with the environment. While traditional "plan-and-act" frameworks break down such tasks into subproblems that are easier to solve (e.g. reach, grasp, pull) [32], they do not offer fast and control-aware replanning, which is crucial for mobile manipulation in dynamic and uncertain environments. With the recent advancements in artificial intelligence, reinforcement learning is a promising method to solve a range of robotic control tasks, including manipulation [33], as they learn an end-to-end representation of the optimal policy. However, real-world applications of reinforcement learning typically require training times that are not practical for physical hardware and suffer from the well-known *sim-to-real* gap [34]. On the other side of the spectrum, Model Predictive Control (MPC) has gained broad interest in the robotics community thanks to its capability of dealing with input constraints and task objectives by solving a multivariate optimization problem. MPC has been successfully applied to aerial robots [35], autonomous racing [36], legged locomotion [37] and whole-body control [38]. Nevertheless, MPC requires a model that is locally differentiable with respect to the input and the state [39]. On the other hand, manipulation tasks involve changes in the contact state causing sharp discontinuities in both the cost and system dynamics, thus directly violating the differentiability requirements.

Recently, sampling-based methods have emerged and advanced in theory and applications [40, 41, 42]. In contrast to traditional MPC, sampling methods stem from a probabilistic interpretation of the control problem. Rather than solving a big optimization, they rely on sampling system trajectories. The only requirement is that it is possible to forward simulate the system evolution. This has been exploited to control camera motions for target tracking in drone racing [40], robot arm motions for manipulation tasks [41] and for generating aggressive driving maneuvers such as drifting [43].

Yet despite their appealing features, sampling-based methods can be costly to execute and solution quality is highly dependent on sampling quality. Previous works argue that thousands of trajectories need to be sampled in real time for the effectiveness of the proposed sampling-based algorithm and therefore GPU-based simulation is needed for fast parallel computation [44]. Unfortunately, GPUs are not common on mobile robots (e.g., because of limited power and payload) and the overall computation times are often not adequate for feedback control. Furthermore, demonstrated solutions for tasks involving different physical interactions (e.g. a robot arm opening a drawer [41]) have been shown on a real system only by breaking down the multi-contact task into stages and enforcing constraints when switching between them. This can limit the control envelope of the system and sacrifices solution optimality. For example, it is common practice to fix the gripper orientation between successive reach and pull stages and perform manipulation under a rigid grasp. In the presence of uncertainty and tracking errors, this often leads to high contact forces that might damage the robot as well as the manipulated object. Ultimately, because of the lack of practical implemented solutions, sampling-based control methods are yet to be applied on real mobile manipulators for whole-body control of complex multi-contact tasks.

2.4 The Rise of Simulators

We expect that perception, planning and model estimation algorithms could greatly profit from the readiness level of physics and sensor rendering capabilities reached nowadays. In recent years, simulators have covered a fundamental role in robotics research. Simulators allow to validate control strategies without access to real-world hardware. Many instantiations of a simulation can run in parallel and are often faster than real-time. Data-hungry deep learning approaches heavily employ simulators as a cheap source of data for training while obviating the potential damage to the robot [45]. However, there are discrepancies between simulations and the real world. Certain physical phenomena such as gear backlash, sensor noise and latencies are approximated or removed. This prevents control systems created in simulation from performing to the same standard as in reality [46]. This disparity is known as the *reality gap* [47]. In order to reduce the impact of the reality gap, physical parameters are then sampled and modified between multiple simulation runs. As a result, the control policy is more robust to real world variations when transferring it to the real-robot [48]. This technique, called *domain randomization* has pushed simulators to expose in their API methods to dynamically change physical and geometrical properties such as friction, damping, mass, joint axis position and orientation, collision and visual meshes. (express that this capability can be deployed in a real time setting)

(more citations, second part is especially unclear) Nevertheless, contemporary simulators have achieved a level of accuracy and speed that make them not only appealing for offline data collection but also for real-time deployment as a model of the system dynamics. This is particularly important for manipulation as physics engines can reliably predict the outcome of complex interactions in faster-than-real-time. Nevertheless, most approaches still use simulation only for validation or data generation. (move this to the rl part)From the learning perspective this means that the learning agent can reach at most the same accuracy as the simulator. Furthermore, the bad performance of these algorithms on the real platform is a result of the intertwined controller behavior, which can be by its own suboptimal, and modeling mismatch.

3 Approach

TODO Giuseppe : this paragraph should go restructured here as it is more the approach then the related work
Solving a complex manipulation task (e.g. tidying up a kitchen) requires a comprehensive understanding of the scene. This task can be further decomposed into a set of sub-tasks of increasing granularity. We restrict our view to the atomic manipulation of a single object.
A solution to robust manipulation of articulated objects consists of perception and control algorithms designed to work together. The elements of the proposed approach build upon the following core components:

- closed-loop physics-aware control
- affordance-based perception
- active model estimation and uncertainty reduction

In the rest of this section, an overview of the methodology, underlying assumptions and a description of each subgoal are provided.

Methodology Each proposed solution is first evaluated in simulation and then on the real hardware consisting of the RoyalPanda mobile manipulator platform. The robot consists of a 7-DoF Franka Emika arm mounted on the holonomic Clearpath Ridgeback base.

Assumptions In this research, the following assumptions are made:

- the object category is recognized prior to manipulation
- manipulation can be solved in a non-prehensile manner: grasping is not required
- sensing modalities are RGB-D images, point clouds, proprioceptive data and wrist-mounted wrench measurements
- articulated objects are restricted to the class of open loop kinematic chains consisting of a combination of revolute and/or prismatic joints

Leveraging simulation We believe that the potential of simulation engines (both for physical interaction and rendering) has not been fully exploited in the current research. Contemporary simulators are fast and accurate enough to replace analytical models while rendering engines afford photorealistic results. Recent behavioral and computation studies of human physical scene understanding push forward an account that people's judgement are best explained as probabilistic simulations of a realistic, albeit noisy, physics engine [49]. They can not only be used for offline synthesis and validation but also to generate manipulation behaviors and robust perception in real-time. Real-time perception offers the following advantages:

- Closed-loop control keeps the deviation of the simulation from the current observed state small
- The improved real-time estimate of model parameters (e.g joint positions and orientations) can be dynamically changed in the simulation
- Detection algorithms can directly act in a virtually rendered scene which replicates the current simulated environment zeroing the *sim-to-real* gap.

On the other hand, a full-scene simulation and rendering is computationally expensive. Therefore it is of utmost importance to come up with new algorithmic solutions that require little data or can cope with limited computational resources. As an example, one could decide to focus perception only on the object to be interacted with and reduce collision checks to the hand (gripper) and the few interaction points.

- *What approach will be used?*
- *Why is the approach promising?*
- *What are the expected results?*

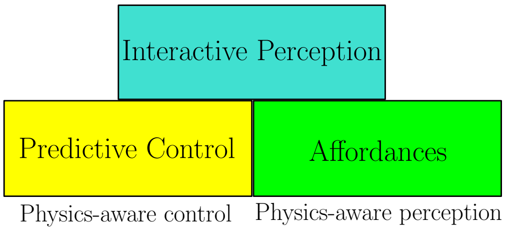

Figure 3: Building blocks of the proposed approach: . . .

3.1 Predictive Control

The literature generally distinguishes between planning and control algorithms. Here we consider control as the process of generating a reference signal that the robot can track and that brings the system's state (robot and object) to the desired one. A typical example is a velocity-controlled manipulator that has to open a door. A control algorithm is then in charge of generating the sequence of velocity values that allows the robot to interact with the door and move it to the open state. This definition is general enough to encompass various approaches. The velocity, for example, can be tracked by another low-level controller which compensates for gravity and Coriolis effects and commands robot torques. The controller could be split into an intermediate step where a reference end-effector trajectory is computed and this is then converted into low-level velocity signals.

The proposed approach consists of planning directly in the joint velocity space as this allows us to consider additional objectives such as joint limits and self-collisions. Furthermore, a low-level control layer which converts velocity references to torques is always used. In fact, tuning the low-level PD gains lets us define a desired compliant behavior avoiding high interaction wrenches with the stiff environment. A cost function is then used as a proxy to inform the controller about the completeness of the task. In the running example, this could be the squared distance from the current door's joint angle to the one corresponding to the open position. Directly optimizing such a cost in the robot action space is extremely complex. In fact, the mapping between the optimization variables (velocities) and the cost is highly non-linear. The mapping goes through the full whole-body dynamics and switching interactions that happen between the robot and the door. For this reason, we resort to gradient-free methods. Gradient-free methods rely on sampling input-state trajectories in simulation in order to drive the system towards optimality (see Sec. ??). However, these come at the cost of being less sample efficient compared to gradient-based methods.

3.1.1 Research gap and relation to previous works

Using a fully fledged simulation backend exploits the fact that basic physical concepts (e.g., distinct objects cannot occupy the same space, gravity applies mass-dependent force on objects, friction and kinematic constraints) provide strong prior knowledge for manipulation tasks [27]. In the first part of this work, a practical

receding horizon algorithm that achieves real-time whole-body control of a mobile manipulator using only a laptop CPU is proposed. This work is based on Model Predictive Path Integral control (MPPI) [43], which is the underlying sampling-based controller for whole-body coordination and control.
We developed several algorithmic elements that enable the controller to achieve performant results, often requiring fewer than 100 rollouts. The main contribution is a regularization of our sampling exploration by means of a momentum update. This is inspired by the Bayesian inference viewpoint of [50], which shows that the choice of exploration noise has the effect of tuning the gradient step size of the path integral update. The benefits of this sampling scheme are twofold, it can aid in escaping local minima while also damping strong oscillations in the optimal policy. The Raisim physics engine is used for fast and accurate simulation [51] while the Pinocchio library [52] is used for rigid body dynamics modeling. The goal of this subtask is to verity the following hypothesis:

> *Gradient-free methods enable real-time reactive whole-body mobile manipulation for multi-contact manipulation of articulated objects on a real platform with limited computational resources.*

3.1.2 Results

The developed algorithm is able to control a complex system in real-time without the need for massive parallel computation. To demonstrate the applicability and effectiveness of this approach, several ablation studies are performed in simulation on kinematic and dynamical manipulators. The full algorithm is then deployed on the RoyalPanda platform for a target reaching and door opening task. An open source implementation of the solution is readily available at `https://git.io/Jtda7`. The results support the original hypothesis and encourage the deployment of the algorithm to more challenging systems such as aerial manipulators.

3.2 Affordances

As mentioned in the previous section, control requires a reward signal to know which states are closer to the target state. In practice, the true objective is to manipulate the object to a target configuration but we do not care too much about the intermediate robot and object states. The cost then is generally an educated guess about this reward signal and can generally impose a structure not needed or even subpotimal for the manipulation problem. Consider the problem of turning a handwheel valve. The final objective (valve being turned) can be achieved in several different ways. For example one could grasp the wheel from the side or just apply a tangential push force to the ribs that connect the wheel to the shaft. It is argued here that designing a cost function that enforces a preference for one or the other approach is suboptimal for autonomous manipulation. Instead, the cost could be related to the concept of *affordances*. The controller can then be informed about more generic interesting interaction regions.

3.2.1 Research gap and relation to previous works

The concept of *affordances* for manipulation has been explored in many recent perception works. They mainly differ in the representation of object affordance as interaction points [20], interaction regions [22] or dense interaction likelihood and orientation maps [23]. Most of these works do not highlight the importance and applicability of the resulting perception pipeline for manipulation control. Furthermore, an agreement on what is the optimal representation of affordances for object manipulation is still missing. In this part of the research, the aim is to combine affordances with control. The goal of this subtask is to verify the following hypothesis:

> *That affordances allow a robotic agent to better exploit the structure in manipulation tasks with respect to fixed manipulation strategies.*

The controller can try to bring the robot close to *high-affordance* points. Note that this task can be hard or even infeasible for traditional gradient-based methods, while the sampling-based algorithm we introduced in section 3.1 lends itself well to this objective. Furthermore, a novel concept of affordances will be explored by introducing haptic information. Intuitively, human not only know by experience *where* to act on object but also *how.* The latter can be expressed in terms of interaction orientations and expected wrench profiles (e.g. pulling vs pushing force). The goal of this second part of the project is to verify the following research hypothesis:

> *That affordances describing haptic information are more descriptive than purely geometry-based affordances.*

In this task the SAPIEN simulator will be used since it integrates a comprehensive set of articulated objects normally found in households, provides ground truth segmentation and photo-realistic rendering [53].

3.3 Interactive Perception

A fundamental aspect to take into consideration for object manipulation is model estimation and the associated *epistemic uncertainty*, which is the uncertainty about the model parameters. For instance, we might not know the geometrical and physical properties of the object such as size, weight, friction, joint positions and orientations. Furthermore, uncertainty can be extended to other types of scene representations such as keypoints and interaction hotspots. Active perception will be investigated to reduce such uncertainty by exploiting the robot's capacity to move sensors in space and changing the environment state.

The concept of affordances developed in the previous section provides information that is decoupled from the specific articulation model. If we are able to perceive which parts of the object are more likely to be interacted with and how, we are also imposing some prior on the type of the articulated object and vice versa. It is then possible to use affordances as a proxy for *explorative actions* when the model is not available. In this part of the work the goal is to integrate interaction into the perception process.

Forceful interaction, also known as *perceptive manipulation* [15] can be used to generate motion and information-rich signals that validate hypotheses about the articulation model. This validation can be performed in simulation which serves as a likelihood model. The plan is to integrate *information seeking* actions in the developed stochastic control and perception framework. We assume that the object has already been categorized and a coarse estimate of the articulation model (category and model parameters) is available. The components so far developed will serve as building blocks of a simultaneous estimation and control pipeline which works in a closed-loop fashion. We will then investigate the following research hypothesis:

> *That simulation and affordance-guided perceptive manipulation is robust to large uncertainty while simultaneously executing the task.*

The sensor modality used to develop the proposed algorithm will be RGB-D cameras and wrist mounted force torque sensors. End-effector mounted cameras are generally good for passive perception but are unusable during manipulation because of heavy occlusions. Therefore, a shoulder or base mounted camera will be evaluated as additional sensory input.

3.3.1 Research gap and relation to previous works

Object classification and articulation model estimation from visual data has already been addressed in previous works [54, 17]. Although showing promising results, the error from single-shot detection is still too large to deploy this estimate for modeling and control. The idea of refining the initial estimate using a physics engine to validate exploratory interactive actions has been investigated in [55]. The proposed method relies on a reactive sampling strategy and human intervention for an accurate initial estimate of the articulation model. Frequently, the grasp model is fixed as well as the action parameterization. These assumptions restrict the type of possible interactions and the generalizability of the approach. Furthermore, methods generally split the problem of task execution and model estimation [55]. Simultaneous parameter estimation and control has been not fully explored for manipulation tasks and has been only applied to low dimensional systems and low dimensional parameter spaces [56].

4 Time Schedule and Planned Publications

- *How can each topic be subdivided into meaningful tasks?*
- *What is the time schedule?*
- *Which publications are planned (list of planned publications)?*

4.1 Planned Publications

Stochastic Control for Reactive Whole-Body Manipulation We demonstrate the applicability of our sampling-based method to whole-body coordination for the task of opening a cabinet door. We provide an optimized open-source algorithm which can be used to solve a general set of robotic tasks. We introduce system and algorithmic adaptations that enable us to successfully transfer the algorithm for reactive control on the real robot.

Affordance-based Control We investigate how to use affordances for control. Following the research thread about dense affordance predictions, we use these to create a cost map to be used by the sampling-based controller to drive the manipulator towards interaction hotspots, thus removing the need for highly engineered cost functions and interaction priors. We evaluate the approach in simulation and on the real system.

Interactive Affordances We enrich pure geometrical affordances with wrench information. We will develop a wrench interaction scheme to provide supervision to a learning agent about successful interaction points, orientations and wrenches. The method will be evaluated on a simulated dataset and on real data of various articulated objects such as cabinet doors, drawers, switches, and handwheel valves. Data sources consists of RGB-D images and point clouds.

Interactive Perception We investigate how to leverage the previous work to perform robust manipulation under modeling uncertainty. We rely on off-the-shelf methods to get a initial coarse estimate of the articulated model and its parameters. This initial estimate is fed to the estimation and control pipeline as a prior which is then refined in real time through interaction while performing the task.

The detailed time plan is shown in the Gantt chart in fig. 4.

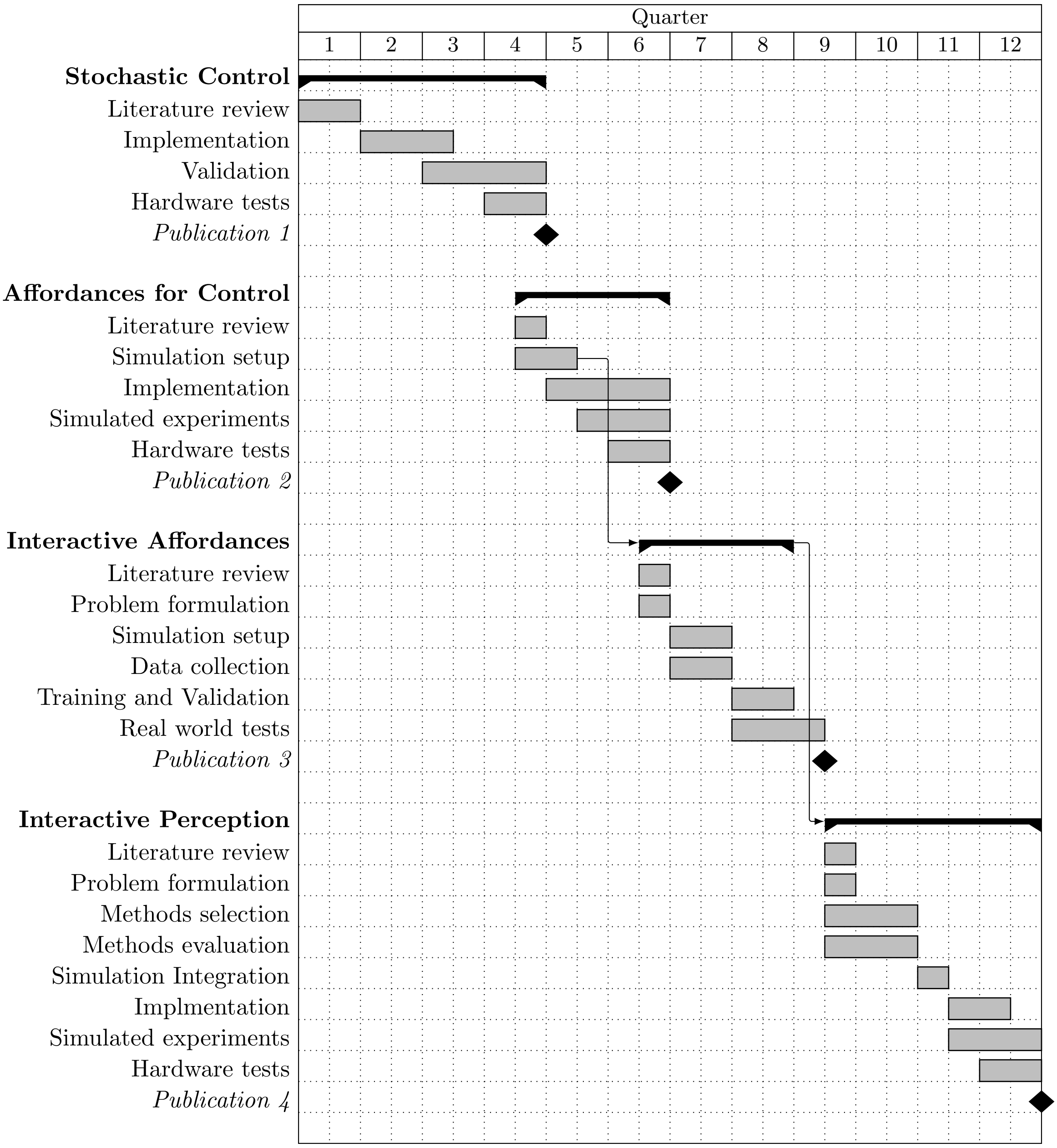

Figure 4: Time plan

References

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2016.

[2] S. Cooper, A. Di Fava, C. Vivas, L. Marchionni, and F. Ferro, "ARI: the social assistive robot and companion," in *2020 29th IEEE Int. Conf. on Robot and Human Interactive Commun.*, 2020, pp. 745–751.

[3] T. Duckett, S. Pearson, S. Blackmore, B. Grieve, W.-H. Chen, G. Cielniak, J. Cleaversmith, J. Dai, S. Davis, C. Fox *et al.*, "Agricultural robotics: The future of robotic agriculture," UK-RAS Network, Tech. Rep., 2018.

[4] K. Nishikawa, A. Imai, K. Miyakawa, T. Kanda, T. Matsuzawa, K. Hashimoto, A. Takanishi, H. Ogata, and J. Ohya, "Disaster response robot's autonomous manipulation of valves in disaster sites based on visual analyses of rgbd images," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4790–4797.

[5] F. Negrello, A. Settimi, D. Caporale, G. Lentini, M. Poggiani, D. Kanoulas, L. Muratore, E. Luberto, G. Santaera, L. Ciarleglio *et al.*, "Walk-man humanoid robot: Field experiments in a post-earthquake scenario," *IEEE Robotics & Automation Magazine*, no. 99, pp. 1–1, 2018.

[6] D. Lattanzi and G. Miller, "Review of robotic infrastructure inspection systems," *J. of Infrastructure Systems*, vol. 23, no. 3, p. 04017004, 2017.

[7] E. Commission. (2020) Horizon 2020 piloting.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2019.

[11] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[12] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 1–28, 2018.

[13] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[14] T. Baca, R. Penicka, P. Stepan, M. Petrlik, V. Spurny, D. Hert, and M. Saska, "Autonomous cooperative wall building by a team of unmanned aerial vehicles in the mbzirc 2020 competition," *arXiv preprint arXiv:2012.05946*, 2020.

[15] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[17] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3706–3715.

[18] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," *arXiv preprint arXiv:1707.01932*, 2017.

[19] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[20] W. Gao and R. Tedrake, "kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[21] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1374–1381.

[22] T. Nagarajan, C. Feichtenhofer, and K. Grauman, "Grounded human-object interaction hotspots from video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8688–8697.

[23] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," *arXiv preprint arXiv:2101.02692*, 2021.

[24] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *Proceedings of the 3rd Conference on Robot Learning*, 2019.

[25] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[26] D. Katz, A. Orthey, and O. Brock, "Interactive perception of articulated objects," in *Experimental Robotics*. Springer, 2014, pp. 301–315.

[27] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *arXiv preprint arXiv:1907.03146*, 2019.

[28] D. Katz and O. Brock, "Interactive segmentation of articulated objects in 3d," in *Workshop on mobile manipulation at ICRA*, vol. 2011, 2011.

[29] Y. Karayiannidis, C. Smith, F. E. Vina, P. Ögren, and D. Kragic, "Model-free robot manipulation of doors and drawers by means of fixed-grasps," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4485–4492.

[30] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, "Active articulation model estimation through interactive perception," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3305–3312.

[31] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[32] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, “6-DOF grasping for target-driven object manipulation in clutter,” in *2020 IEEE Int. Conf. on Robot. and Autom.*, 5 2020, pp. 6232–6238.

[33] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, “Deep spatial autoencoders for visuomotor learning,” in *2016 IEEE Int. Conf. on Robot. and Autom.*, 2016, pp. 512–519.

[34] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, “Closing the sim-to-real loop: Adapting simulation randomization with real world experience,” in *2019 Int. Conf. on Robot. and Autom.*, 2019, pp. 8973–8979.

[35] M. Brunner, K. Bodie, M. Kamel, M. Pantic, W. Zhang, J. Nieto, and R. Siegwart, “Trajectory tracking nonlinear model predictive control for an overactuated mav,” in *2020 IEEE Int. Conf. on Robot. and Autom.*, 2020, pp. 5342–5348.

[36] A. Liniger, A. Domahidi, and M. Morari, “Optimization-based autonomous racing of 1:43 scale RC cars,” *Optimal Control Appl. and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[37] R. Grandia, F. Farshidian, A. Dosovitskiy, R. Ranftl, and M. Hutter, “Frequency-aware model predictive control,” *IEEE Robot. and Autom. Lett.*, vol. 4, no. 2, pp. 1517–1524, 2019.

[38] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, “Whole-body MPC for a dynamically stable mobile manipulator,” *IEEE Robot. and Autom. Lett.*, vol. 4, no. 4, pp. 3687–3694, 2019.

[39] J. Buchli, F. Farshidian, A. Winkler, T. Sandy, and M. Giftthaler, “Optimal and learning control for autonomous robots: Lecture notes,” arXiv preprint arXiv:1708.09342, 2017.

[40] K. Lee, J. Gibson, and E. A. Theodorou, “Aggressive perception-aware navigation using deep optical flow dynamics and PixelMPC,” *IEEE Robot. and Autom. Lett.*, vol. 5, no. 2, 2020.

[41] I. Abraham, A. Handa, N. Ratliff, K. Lowrey, T. D. Murphey, and D. Fox, “Model-based generalization under parameter uncertainty using path integral control,” *IEEE Robot. and Autom. Lett.*, vol. 5, no. 2, pp. 2864–2871, 2020.

[42] J. Rajamäki and P. Hämäläinen, “Augmenting sampling based controllers with machine learning,” in *Proc. of the ACM SIGGRAPH / Eurographics Symp. on Comput. Animation*, 2017, pp. 1–9.

[43] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, “Information theoretic MPC for model-based reinforcement learning,” in *2017 IEEE Int. Conf. on Robot. and Autom.*, 2017, pp. 1714–1721.

[44] G. Williams, A. Aldrich, and E. A. Theodorou, “Model predictive path integral control: From theory to parallel computation,” *J. of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[45] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, “GPU-accelerated robotic simulation for distributed reinforcement learning,” 00016.

[46] J. Collins, J. McVicar, D. Wedlock, R. Brown, D. Howard, and J. Leitner, "Benchmarking simulated robotic manipulation through a real world dataset," vol. 5, no. 1, pp. 250–257, 00002.

[47] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[48] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[49] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," *Advances in neural information processing systems*, vol. 28, pp. 127–135, 2015.

[50] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, "Stein variational model predictive control," in *Conf. on Robot Learn.*, 2020.

[51] R. Tech. (2021) Raisim: a cross-platform multi-body physics engine for robotics and ai.

[52] J. Carpentier, F. Valenza, N. Mansard *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2021.

[53] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[54] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[55] C. Eppner, R. Martín-Martín, and O. Brock, "Physics-based selection of informative actions for interactive perception," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7427–7432.

[56] L. Barcelos, A. Lambert, R. Oliveira, P. Borges, B. Boots, and F. Ramos, "Dual online stein variational inference for control and dynamics," *arXiv preprint arXiv:2103.12890*, 2021.